

Рабочая программа  
по учебному предмету «Информатика»  
на уровень среднего общего образования  
(10-11 класс)  
Углубленный уровень

**Содержание**

Пояснительная записка	2
1. Планируемые результаты освоения учебного предмета «Информатика»	2
2. Содержание учебного предмета «Информатика»	
3. Тематическое планирование с указанием количества часов, отводимых на освоение каждой темы с учетом реализации Рабочей программы воспитания	

## **Пояснительная записка**

Рабочая программа «Информатика» на уровень среднего общего образования (10-11 классы) по информатике составлена на основе ФГОС среднего общего образования, Примерной программы и направлена на реализацию УМК Полякова К.Ю. «Информатика».

Программа рассчитана на реализацию в объеме **4 часа в неделю.**

### **1. Планируемые результаты освоения учебного предмета «Информатика»**

#### **Личностные и метапредметные результаты освоения учебного предмета**

1) российскую гражданскую идентичность, патриотизм, уважение к своему народу, чувства ответственности перед Родиной, гордости за свой край, свою Родину, прошлое и настоящее многонационального народа России, уважение государственных символов (герб, флаг, гимн);

2) гражданскую позицию как активного и ответственного члена российского общества, осознающего свои конституционные права и обязанности, уважающего закон и правопорядок, обладающего чувством собственного достоинства, осознанно принимающего традиционные национальные и общечеловеческие гуманистические и демократические ценности;

3) готовность к служению Отечеству, его защите;

4) сформированность мировоззрения, соответствующего современному уровню развития науки и общественной практики, основанного на диалоге культур, а также различных форм общественного сознания, осознание своего места в поликультурном мире;

5) сформированность основ саморазвития и самовоспитания в соответствии с общечеловеческими ценностями и идеалами гражданского общества; готовность и способность к самостоятельной, творческой и ответственной деятельности;

6) толерантное сознание и поведение в поликультурном мире, готовность и способность вести диалог с другими людьми, достигать в нем взаимопонимания, находить общие цели и сотрудничать для их достижения, способность противостоять идеологии экстремизма, национализма, ксенофобии, дискриминации по социальным, религиозным, расовым, национальным признакам и другим негативным социальным явлениям;

7) навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, общественно полезной, учебно-исследовательской, проектной и других видах деятельности;

8) нравственное сознание и поведение на основе усвоения общечеловеческих ценностей;

9) готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;

10) эстетическое отношение к миру, включая эстетику быта, научного и технического творчества, спорта, общественных отношений;

11) принятие и реализацию ценностей здорового и безопасного образа жизни, потребности в физическом самосовершенствовании, занятиях спортивно-оздоровительной деятельностью, неприятие вредных привычек: курения, употребления алкоголя, наркотиков;

12) бережное, ответственное и компетентное отношение к физическому и психологическому здоровью, как собственному, так и других людей, умение оказывать первую помощь;

13) осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем;

14) сформированность экологического мышления, понимания влияния социально-экономических процессов на состояние природной и социальной среды; приобретение опыта эколого-направленной деятельности;

15) ответственное отношение к созданию семьи на основе осознанного принятия ценностей семейной жизни.

При изучении предмета «Информатика» в соответствии с требованиями ФГОС формируются следующие **метапредметные результаты**.

1) умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

2) умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;

3) владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;

4) готовность и способность к самостоятельной информационно-познавательной деятельности, владение навыками получения необходимой информации из словарей разных типов, умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

5) умение использовать средства информационных и коммуникационных технологий (далее - ИКТ) в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;

6) умение определять назначение и функции различных социальных институтов;

7) умение самостоятельно оценивать и принимать решения, определяющие стратегию поведения, с учетом гражданских и нравственных ценностей;

8) владение языковыми средствами - умение ясно, логично и точно излагать свою точку зрения, использовать адекватные языковые средства;

9) владение навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований, границ своего знания и незнания, новых познавательных задач и средств их достижения.

При изучении курса «Информатика» в соответствии с требованиями ФГОС формируются следующие предметные результаты, которые ориентированы на обеспечение, преимущественно, общеобразовательной и общекультурной подготовки.

### **Планируемые предметные результаты освоения учебного предмета «Информатика»**

<b>Требования ФГОС</b>	<b>Чем достигается</b>
1. Сформированность представлений о роли информации и связанных с ней процессов в окружающем мире	10 класс. Глава 1. Информация и информационные процессы
2. Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов	10 класс. Глава 8. Алгоритмизация и программирование. § 51. Алгоритмы. § 53. Анализ алгоритмов с ветвлениями и циклами
3. Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня. Владение знанием основных конструкций программирования. Владение умением анализировать алгоритмы с использованием таблиц	10 класс. Глава 8. Алгоритмизация и программирование
4. Владение стандартными приемами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных	10 класс. Глава 8. Алгоритмизация и программирование

конструкций программирования и отладки таких программ	
Использование готовых прикладных компьютерных программ по выбранной специализации	10 класс. Глава 6. Программное обеспечение 11 класс. Глава 3. Базы данных. Глава 8. Обработка изображений. Глава 9. Трёхмерная графика
5. Сформированность представлений о компьютерно-математических моделях и необходимости анализа соответствия модели и моделируемого объекта (процесса) Сформированность представлений о способах хранения и простейшей обработке данных. Сформированность понятия о базах данных и средствах доступа к ним, умений работать с ними	11 класс. Глава 2. Моделирование. 11 класс. Глава 3. Базы данных. 11 класс. Глава 3. Базы данных
6. Владение компьютерными средствами представления и анализа данных	10 класс. Глава 9. Вычислительные задачи. 11 класс. Глава 2. Моделирование
7. Сформированность базовых навыков и умений по соблюдению требований техники безопасности, гигиены и ресурсосбережения при работе со средствами информатизации. Сформированность понимания основ правовых аспектов использования компьютерных программ и работы в Интернете	10 класс. Техника безопасности. 10 класс. Глава 6. Программное обеспечение

## 2. Содержание учебного предмета «Информатика»

Программа учебного предмета «Информатика» на уровне среднего общего образования составлена в соответствии с требованиями ФГОС СОО; требованиями к результатам освоения основной образовательной программы. В ней соблюдается преемственность с ФГОС ООО и учитываются межпредметные связи.

Цель изучения учебного предмета «Информатика» на базовом и углубленном уровнях среднего общего образования – обеспечение дальнейшего развития информационных компетенций выпускника, готового к работе в условиях развивающегося информационного общества и возрастающей конкуренции на рынке труда.

Базовый уровень

Введение. Информация и информационные процессы

Роль информации и связанных с ней процессов в окружающем мире.

Различия в представлении данных, предназначенных для хранения и обработки в автоматизированных компьютерных системах, и данных, предназначенных для восприятия человеком.

Системы. Компоненты системы и их взаимодействие.

Универсальность дискретного представления информации.

Математические основы информатики

Тексты и кодирование

Равномерные и неравномерные коды. Условие Фано.

Системы счисления

Сравнение чисел, записанных в двоичной, восьмеричной и шестнадцатеричной системах счисления. Сложение и вычитание чисел, записанных в этих системах счисления.

Элементы комбинаторики, теории множеств и математической логики

Операции «импликация», «эквивалентность». Примеры законов алгебры логики. Эквивалентные преобразования логических выражений.

Построение логического выражения с данной таблицей истинности. Решение простейших логических уравнений.

Нормальные формы: дизъюнктивная и конъюнктивная нормальная форма.

Дискретные объекты

Решение алгоритмических задач, связанных с анализом графов (примеры: построения оптимального пути между вершинами ориентированного ациклического графа; определения количества различных путей между вершинами). Использование графов, деревьев, списков при описании объектов и процессов окружающего мира. Бинарное дерево.

Алгоритмы и элементы программирования

Алгоритмические конструкции

Подпрограммы. Рекурсивные алгоритмы.

Табличные величины (массивы).

Запись алгоритмических конструкций в выбранном языке программирования.

Составление алгоритмов и их программная реализация

Этапы решения задач на компьютере.

Операторы языка программирования, основные конструкции языка программирования. Типы и структуры данных. Кодирование базовых алгоритмических конструкций на выбранном языке программирования.

Интегрированная среда разработки программ на выбранном языке программирования. Интерфейс выбранной среды. Составление алгоритмов и программ в выбранной среде программирования. Приемы отладки программ. Проверка работоспособности программ с использованием трассировочных таблиц.

Разработка и программная реализация алгоритмов решения типовых задач базового уровня из различных предметных областей. Примеры задач:

– алгоритмы нахождения наибольшего (или наименьшего) из двух, трех, четырех заданных чисел без использования массивов и циклов, а также сумм



(или произведений) элементов конечной числовой последовательности (или массива);

- алгоритмы анализа записей чисел в позиционной системе счисления;
- алгоритмы решения задач методом перебора (поиск НОД данного натурального числа, проверка числа на простоту и т.д.);
- алгоритмы работы с элементами массива с однократным просмотром массива: линейный поиск элемента, вставка и удаление элементов в массиве, перестановка элементов данного массива в обратном порядке, суммирование элементов массива, проверка соответствия элементов массива некоторому условию, нахождение второго по величине наибольшего (или наименьшего) значения.

Алгоритмы редактирования текстов (замена символа/фрагмента, удаление и вставка символа/фрагмента, поиск вхождения заданного образца).

Постановка задачи сортировки.

Анализ алгоритмов

Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.

Сложность вычисления: количество выполненных операций, размер используемой памяти; зависимость вычислений от размера исходных данных.

Математическое моделирование

Представление результатов моделирования в виде, удобном для восприятия человеком. Графическое представление данных (схемы, таблицы, графики).

Практическая работа с компьютерной моделью по выбранной теме. Анализ достоверности (правдоподобия) результатов экспериментов. Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности.

Использование программных систем и сервисов

Компьютер – универсальное устройство обработки данных

Программная и аппаратная организация компьютеров и компьютерных систем. Архитектура современных компьютеров. Персональный компьютер. Многопроцессорные системы. Суперкомпьютеры. Распределенные вычислительные системы и обработка больших данных. Мобильные цифровые устройства и их роль в коммуникациях. Встроенные компьютеры. Микроконтроллеры. Роботизированные производства.

Выбор конфигурации компьютера в зависимости от решаемой задачи. Тенденции развития аппаратного обеспечения компьютеров.

Программное обеспечение (ПО) компьютеров и компьютерных систем. Различные виды ПО и их назначение. Особенности программного обеспечения мобильных устройств.

Организация хранения и обработки данных, в том числе с использованием интернет-сервисов, облачных технологий и мобильных устройств. Прикладные компьютерные программы, используемые в соответствии с типом решаемых задач и по выбранной специализации. Параллельное программирование.

Инсталляция и деинсталляция программных средств, необходимых для решения учебных задач и задач по выбранной специализации. Законодательство Российской Федерации в области программного обеспечения.

Способы и средства обеспечения надежного функционирования средств ИКТ. Применение специализированных программ для обеспечения стабильной работы средств ИКТ.

Безопасность, гигиена, эргономика, ресурсосбережение, технологические требования при эксплуатации компьютерного рабочего места. Проектирование автоматизированного рабочего места в соответствии с целями его использования.

Подготовка текстов и демонстрационных материалов

Средства поиска и автозамены. История изменений. Использование готовых шаблонов и создание собственных. Разработка структуры документа, создание гипертекстового документа. Стандарты библиографических описаний.

Деловая переписка, научная публикация. Реферат и аннотация. Оформление списка литературы.

Коллективная работа с документами. Рецензирование текста. Облачные сервисы.

Знакомство с компьютерной версткой текста. Технические средства ввода текста. Программы распознавания текста, введенного с использованием сканера, планшетного ПК или графического планшета. Программы синтеза и распознавания устной речи.

Работа с аудиовизуальными данными

Создание и преобразование аудиовизуальных объектов. Ввод изображений с использованием различных цифровых устройств (цифровых фотоаппаратов и микроскопов, видеокамер, сканеров и т. д.). Обработка изображения и звука с использованием интернет- и мобильных приложений.

Использование мультимедийных онлайн-сервисов для разработки презентаций проектных работ. Работа в группе, технология публикации готового материала в сети.

Электронные (динамические) таблицы

Примеры использования динамических (электронных) таблиц на практике (в том числе – в задачах математического моделирования).

Базы данных

Реляционные (табличные) базы данных. Таблица – представление сведений об однотипных объектах. Поле, запись. Ключевые поля таблицы. Связи между таблицами. Схема данных. Поиск и выбор в базах данных. Сортировка данных.

Создание, ведение и использование баз данных при решении учебных и практических задач.

## Автоматизированное проектирование

Представление о системах автоматизированного проектирования. Системы автоматизированного проектирования. Создание чертежей типовых деталей и объектов.

## 3D-моделирование

Принципы построения и редактирования трехмерных моделей. Сеточные модели. Материалы. Моделирование источников освещения. Камеры.

## Аддитивные технологии (3D-принтеры).

## Системы искусственного интеллекта и машинное обучение

Машинное обучение – решение задач распознавания, классификации и предсказания. Искусственный интеллект.

Информационно-коммуникационные технологии. Работа в информационном пространстве

## Компьютерные сети

Принципы построения компьютерных сетей. Сетевые протоколы. Интернет. Адресация в сети Интернет. Система доменных имен. Браузеры.

## Аппаратные компоненты компьютерных сетей.

Веб-сайт. Страница. Взаимодействие веб-страницы с сервером. Динамические страницы. Разработка интернет-приложений (сайты).

## Сетевое хранение данных. Облачные сервисы.

## Деятельность в сети Интернет

Расширенный поиск информации в сети Интернет. Использование языков построения запросов.

Другие виды деятельности в сети Интернет. Геолокационные сервисы реального времени (локация мобильных телефонов, определение загруженности автомагистралей и т.п.); интернет-торговля; бронирование билетов и гостиниц и т.п.

## Социальная информатика

Социальные сети – организация коллективного взаимодействия и обмена данными. Сетевой этикет: правила поведения в киберпространстве.

Проблема подлинности полученной информации. Информационная культура. Государственные электронные сервисы и услуги. Мобильные приложения. Открытые образовательные ресурсы.

Информационная безопасность

Средства защиты информации в автоматизированных информационных системах (АИС), компьютерных сетях и компьютерах. Общие проблемы защиты информации и информационной безопасности АИС. Электронная подпись, сертифицированные сайты и документы.

Техногенные и экономические угрозы, связанные с использованием ИКТ. Правовое обеспечение информационной безопасности.

**3. Тематическое планирование с указанием количества часов, отводимых на освоение каждой темы с учетом реализации**

**Рабочей программы воспитания**

Тематическое планирование ориентировано на положения модуля 4.2. «Школьный урок» Рабочей программы воспитания, являющейся частью содержательного раздела данной образовательной программы

Объем углублённого курса 272 учебных часа (по 4 часа в неделю в 10 и 11 класс).

№	Тема	Количество часов/класс		
		Всего	10кл.	11кл.
<b>Основы информатики</b>				
1	Техника безопасности. Организация рабочего места	1	1	
2	Информация и информационные процессы	16	5	11
3	Кодирование информации (урок-викторина)	14	14	
4	Логические основы компьютера	13	13	

5	Компьютерная арифметика (урок-турнир)	6	6	
6	Устройство компьютера (урок-брифинг)	6	6	
7	Программное обеспечение	19	19	
8	Компьютерные сети	9	9	
9	Информационная безопасность	6	6	
	Итого:	90	79	11
<b>Алгоритмы и программирование</b>				
10	Алгоритмизация и программирование	69	44	25
11	Решение вычислительных задач	8	8	
12	Элементы теории алгоритмов (урок-конкурс)	6		6
13	Объектно-ориентированное программирование (деловая игра)	12		12
	Итого:	95	52	43
<b>Информационно-коммуникационные технологии</b>				
14	Моделирование (урок-соревнование)	13		13
15	Базы данных	11		11
16	Создание веб-сайтов	15		15
17	Графика и анимация (урок-КВН)	9		9
18	3D-моделирование	10		10
	Итого:	58	0	58
	Резерв	29	5	24
	Итого по всем разделам:	272	136	136

**Приложение**  
**Оценочный модуль**  
**Контрольные работы**  
**10 класс**

**Контрольная работа «Системы счисления»**

- 1) Перевести число из двоичной системы в системы с основанием 8, 10, 16.
- 2) Перевести число из восьмеричной системы в системы с основанием 2, 10, 16.
- 3) Перевести число из десятичной системы в системы с основанием 2, 8, 16.
- 4) Перевести число из шестнадцатеричной системы в системы с основанием 2, 8, 10.
- 5) Сложить двоичные числа.
- 6) Сложить восьмеричные числа.
- 7) Сложить шестнадцатеричные числа.
- 8) Вычесть двоичные числа.
- 9) Вычесть восьмеричные числа.
- 10) Вычесть шестнадцатеричные числа.

----- Вариант 1 -----

- 1)  $10001100_2$  2)  $205_8$  3)  $155$  4)  $89_{16}$   
5)  $10010110_2$  6)  $6271_8$  7)  $B06_{16}$  8)  $1011010_2$  9)  $1521_8$  10)  $401_{16}$   
+  $11001011_2$  +  $7024_8$  +  $C64_{16}$  -  $11111001_2$  -  $7041_8$  -  $E1A_{16}$

----- Вариант 2 -----

- 1)  $11000111_2$  2)  $346_8$  3)  $137$  4)  $D6_{16}$   
5)  $11011110_2$  6)  $6310_8$  7)  $93D_{16}$  8)  $1010110_2$  9)  $1611_8$  10)  $40C_{16}$   
+  $10001010_2$  +  $5514_8$  +  $BD6_{16}$  -  $10111100_2$  -  $7760_8$  -  $C92_{16}$

----- Вариант 3 -----

- 1)  $10101010_2$  2)  $301_8$  3)  $229$  4)  $8D_{16}$   
5)  $10111011_2$  6)  $4401_8$  7)  $BB8_{16}$  8)  $110011_2$  9)  $1745_8$  10)  $34A_{16}$   
+  $1101001_2$  +  $4077_8$  +  $FC0_{16}$  -  $11101010_2$  -  $7633_8$  -  $D13_{16}$

----- Вариант 4 -----

- 1)  $11000110_2$  2)  $304_8$  3)  $200$  4)  $7C_{16}$   
5)  $11001001_2$  6)  $4002_8$  7)  $8EC_{16}$  8)  $1000100_2$  9)  $1676_8$  10)  $39B_{16}$   
+  $10001101_2$  +  $5634_8$  +  $865_{16}$  -  $11101101_2$  -  $6676_8$  -  $C38_{16}$

----- Вариант 5 -----

- 1)  $10000001_2$  2)  $156_8$  3)  $107$  4)  $D6_{16}$   
5)  $11011010_2$  6)  $6717_8$  7)  $D8D_{16}$  8)  $1001100_2$  9)  $2005_8$  10)  $33D_{16}$   
+  $11010011_2$  +  $4706_8$  +  $A1E_{16}$  -  $11111100_2$  -  $7442_8$  -  $DAC_{16}$

----- Вариант 6 -----

- 1)  $11000111_2$  2)  $160_8$  3)  $194$  4)  $7F_{16}$   
5)  $11001010_2$  6)  $7321_8$  7)  $F97_{16}$  8)  $1110100_2$  9)  $2071_8$  10)  $42D_{16}$   
+  $10111010_2$  +  $5725_8$  +  $C55_{16}$  -  $11001111_2$  -  $7543_8$  -  $ECD_{16}$

----- Вариант 7 -----

- 1)  $10010100_2$  2)  $300_8$  3)  $172$  4)  $A5_{16}$   
5)  $10111000_2$  6)  $5323_8$  7)  $946_{16}$  8)  $1000101_2$  9)  $1776_8$  10)  $350_{16}$   
+  $10001110_2$  +  $7224_8$  +  $88B_{16}$  -  $11010001_2$  -  $6471_8$  -  $EAE_{16}$

----- Вариант 8 -----

- 1)  $11011100_2$  2)  $214_8$  3)  $214$  4)  $66_{16}$   
5)  $10101000_2$  6)  $4005_8$  7)  $C1E_{16}$  8)  $1101110_2$  9)  $1615_8$  10)  $356_{16}$   
+  $10110100_2$  +  $4603_8$  +  $955_{16}$  -  $11010100_2$  -  $7073_8$  -  $D90_{16}$

----- Вариант 9 -----

- 1)  $10000000_2$  2)  $222_8$  3)  $229$  4)  $66_{16}$   
5)  $10111011_2$  6)  $6771_8$  7)  $BE4_{16}$  8)  $1100000_2$  9)  $2044_8$  10)  $3DF_{16}$

$$+ 11111001_2 + 6725_8 + 959_{16} - 11011001_2 - 6464_8 - C58_{16}$$

----- Вариант 10 -----

- 1)  $11001011_2$  2)  $235_8$  3)  $239$  4)  $E5_{16}$   
5)  $11001101_2$  6)  $4251_8$  7)  $9A3_{16}$  8)  $1011010_2$  9)  $1721_8$  10)  $43B_{16}$   
 $+ 10000100_2 + 6561_8 + A55_{16} - 11111110_2 - 7402_8 - C08_{16}$

----- Вариант 11 -----

- 1)  $11010001_2$  2)  $254_8$  3)  $123$  4)  $75_{16}$   
5)  $10110001_2$  6)  $5560_8$  7)  $EAA_{16}$  8)  $1100101_2$  9)  $1656_8$  10)  $3DD_{16}$   
 $+ 11001011_2 + 4764_8 + C17_{16} - 10100011_2 - 6246_8 - D35_{16}$

----- Вариант 12 -----

- 1)  $10101110_2$  2)  $332_8$  3)  $140$  4)  $CA_{16}$   
5)  $1101101_2$  6)  $6015_8$  7)  $F89_{16}$  8)  $1110100_2$  9)  $1653_8$  10)  $3F0_{16}$   
 $+ 10011101_2 + 5654_8 + B84_{16} - 11010101_2 - 6152_8 - F7D_{16}$

----- Вариант 13 -----

- 1)  $10010010_2$  2)  $265_8$  3)  $213$  4)  $86_{16}$   
5)  $1101111_2$  6)  $6644_8$  7)  $F17_{16}$  8)  $1111000_2$  9)  $2052_8$  10)  $324_{16}$   
 $+ 11110101_2 + 4235_8 + 993_{16} - 11110100_2 - 5761_8 - CB6_{16}$

----- Вариант 14 -----

- 1)  $11110010_2$  2)  $370_8$  3)  $140$  4)  $AB_{16}$   
5)  $10010101_2$  6)  $5674_8$  7)  $96F_{16}$  8)  $111001_2$  9)  $2032_8$  10)  $3EB_{16}$   
 $+ 10011100_2 + 5576_8 + A52_{16} - 10110010_2 - 6205_8 - DD1_{16}$

----- Вариант 15 -----

- 1)  $11110110_2$  2)  $157_8$  3)  $122$  4)  $D8_{16}$   
5)  $1101000_2$  6)  $6502_8$  7)  $EEF_{16}$  8)  $1000001_2$  9)  $1671_8$  10)  $3A0_{16}$   
 $+ 11111100_2 + 5273_8 + BDE_{16} - 11011111_2 - 7107_8 - C53_{16}$

----- Вариант 16 -----

- 1)  $10000010_2$  2)  $303_8$  3)  $196$  4)  $72_{16}$   
5)  $10000111_2$  6)  $7606_8$  7)  $F38_{16}$  8)  $1010011_2$  9)  $1730_8$  10)  $43B_{16}$   
 $+ 10001000_2 + 7563_8 + D6E_{16} - 11011110_2 - 7645_8 - FC7_{16}$

----- Вариант 17 -----

- 1)  $1110100_2$  2)  $352_8$  3)  $140$  4)  $64_{16}$   
5)  $1111001_2$  6)  $4472_8$  7)  $C32_{16}$  8)  $1010011_2$  9)  $1774_8$  10)  $37F_{16}$   
 $+ 11001010_2 + 6171_8 + E48_{16} - 10101010_2 - 6320_8 - F3D_{16}$

----- Вариант 18 -----

- 1)  $1101111_2$  2)  $275_8$  3)  $153$  4)  $98_{16}$   
5)  $10111100_2$  6)  $6566_8$  7)  $D06_{16}$  8)  $1100111_2$  9)  $1766_8$  10)  $383_{16}$   
 $+ 11001101_2 + 6526_8 + CC8_{16} - 11000111_2 - 6020_8 - EBB_{16}$

----- Вариант 19 -----

- 1)  $11100110_2$  2)  $341_8$  3)  $134$  4)  $83_{16}$   
5)  $10000111_2$  6)  $7267_8$  7)  $922_{16}$  8)  $111000_2$  9)  $1446_8$  10)  $42D_{16}$   
 $+ 10110010_2 + 6504_8 + D5E_{16} - 11111011_2 - 6025_8 - D8F_{16}$

----- Вариант 20 -----

- 1)  $11101001_2$  2)  $343_8$  3)  $211$  4)  $E8_{16}$   
5)  $1111101_2$  6)  $4734_8$  7)  $B39_{16}$  8)  $1110010_2$  9)  $1500_8$  10)  $3A1_{16}$   
 $+ 1100111_2 + 7063_8 + 98F_{16} - 10100110_2 - 6000_8 - E5A_{16}$



## Ответы к контрольной работе «Системы счисления»

----- Вариант 1 -----

- 1)  $214_8$      $140_{16}$      $8C_{16}$     2)  $10000101_2$      $133_{16}$      $85_{16}$   
3)  $10011011_2$      $233_8$      $9B_{16}$     4)  $10001001_2$      $211_8$      $137_{16}$   
5)  $101100001_2$     6)  $15315_8$     7)  $176A_{16}$     8)  $-10011111_2$     9)  $-5320_8$     10)  $-A19_{16}$

----- Вариант 2 -----

- 1)  $307_8$      $199_{16}$      $C7_{16}$     2)  $11100110_2$      $230_{16}$      $E6_{16}$   
3)  $10001001_2$      $211_8$      $89_{16}$     4)  $11010110_2$      $326_8$      $214_{16}$   
5)  $101101000_2$     6)  $14024_8$     7)  $1513_{16}$     8)  $-1100110_2$     9)  $-6147_8$     10)  $-886_{16}$

----- Вариант 3 -----

- 1)  $252_8$      $170_{16}$      $AA_{16}$     2)  $11000001_2$      $193_{16}$      $C1_{16}$   
3)  $11100101_2$      $345_8$      $E5_{16}$     4)  $10001101_2$      $215_8$      $141_{16}$   
5)  $100100100_2$     6)  $10500_8$     7)  $1B78_{16}$     8)  $-10110111_2$     9)  $-5666_8$     10)  $-9C9_{16}$

----- Вариант 4 -----

- 1)  $306_8$      $198_{16}$      $C6_{16}$     2)  $11000100_2$      $196_{16}$      $C4_{16}$   
3)  $11001000_2$      $310_8$      $C8_{16}$     4)  $1111100_2$      $174_8$      $124_{16}$   
5)  $101010110_2$     6)  $11636_8$     7)  $1151_{16}$     8)  $-10101001_2$     9)  $-5000_8$     10)  $-89D_{16}$

----- Вариант 5 -----

- 1)  $201_8$      $129_{16}$      $81_{16}$     2)  $1101110_2$      $110_{16}$      $6E_{16}$   
3)  $1101011_2$      $153_8$      $6B_{16}$     4)  $11010110_2$      $326_8$      $214_{16}$   
5)  $110101101_2$     6)  $13625_8$     7)  $17AB_{16}$     8)  $-10110000_2$     9)  $-5435_8$     10)  $-A6F_{16}$

----- Вариант 6 -----

- 1)  $307_8$      $199_{16}$      $C7_{16}$     2)  $1110000_2$      $112_{16}$      $70_{16}$   
3)  $11000010_2$      $302_8$      $C2_{16}$     4)  $1111111_2$      $177_8$      $127_{16}$   
5)  $110000100_2$     6)  $15246_8$     7)  $1BEC_{16}$     8)  $-1011011_2$     9)  $-5452_8$     10)  $-AA0_{16}$

----- Вариант 7 -----

- 1)  $224_8$      $148_{16}$      $94_{16}$     2)  $11000000_2$      $192_{16}$      $C0_{16}$   
3)  $10101100_2$      $254_8$      $AC_{16}$     4)  $10100101_2$      $245_8$      $165_{16}$   
5)  $101000110_2$     6)  $14547_8$     7)  $11D1_{16}$     8)  $-10001100_2$     9)  $-4473_8$     10)  $-B5E_{16}$

----- Вариант 8 -----

- 1)  $334_8$      $220_{16}$      $DC_{16}$     2)  $10001100_2$      $140_{16}$      $8C_{16}$   
3)  $11010110_2$      $326_8$      $D6_{16}$     4)  $1100110_2$      $146_8$      $102_{16}$   
5)  $101011100_2$     6)  $10610_8$     7)  $1573_{16}$     8)  $-1100110_2$     9)  $-5256_8$     10)  $-A3A_{16}$

----- Вариант 9 -----

- 1)  $200_8$      $128_{16}$      $80_{16}$     2)  $10010010_2$      $146_{16}$      $92_{16}$   
3)  $11100101_2$      $345_8$      $E5_{16}$     4)  $1100110_2$      $146_8$      $102_{16}$   
5)  $110110100_2$     6)  $15716_8$     7)  $153D_{16}$     8)  $-1111001_2$     9)  $-4420_8$     10)  $-879_{16}$

----- Вариант 10 -----

- 1)  $313_8$      $203_{16}$      $CB_{16}$     2)  $10011101_2$      $157_{16}$      $9D_{16}$   
3)  $11101111_2$      $357_8$      $EF_{16}$     4)  $11100101_2$      $345_8$      $229_{16}$   
5)  $101010001_2$     6)  $13032_8$     7)  $13F8_{16}$     8)  $-10100100_2$     9)  $-5461_8$     10)  $-7CD_{16}$

----- Вариант 11 -----

- 1)  $321_8$      $209_{16}$      $D1_{16}$     2)  $10101100_2$      $172_{16}$      $AC_{16}$   
3)  $1111011_2$      $173_8$      $7B_{16}$     4)  $1110101_2$      $165_8$      $117_{16}$   
5)  $101111100_2$     6)  $12544_8$     7)  $1AC1_{16}$     8)  $-111110_2$     9)  $-4370_8$     10)  $-958_{16}$

----- Вариант 12 -----

- 1)  $256_8$      $174_{16}$      $AE_{16}$     2)  $11011010_2$      $218_{16}$      $DA_{16}$   
3)  $10001100_2$      $214_8$      $8C_{16}$     4)  $11001010_2$      $312_8$      $202_{16}$   
5)  $100001010_2$     6)  $13671_8$     7)  $1B0D_{16}$     8)  $-1100001_2$     9)  $-4277_8$     10)  $-B8D_{16}$

----- Вариант 13 -----

- 1)  $222_8$      $146_{16}$      $92_{16}$     2)  $10110101_2$      $181_{16}$      $B5_{16}$

- 3)  $11010101_2$   $325_8$   $D5_{16}$  4)  $10000110_2$   $206_8$   $134$   
5)  $101100100_2$  6)  $13101_8$  7)  $18AA_{16}$  8)  $-1111100_2$  9)  $-3707_8$  10)  $-992_{16}$

----- Вариант 14 -----

- 1)  $362_8$   $242$   $F2_{16}$  2)  $11111000_2$   $248$   $F8_{16}$   
3)  $10001100_2$   $214_8$   $8C_{16}$  4)  $10101011_2$   $253_8$   $171$   
5)  $100110001_2$  6)  $13472_8$  7)  $13C1_{16}$  8)  $-1111001_2$  9)  $-4153_8$  10)  $-9E6_{16}$

----- Вариант 15 -----

- 1)  $366_8$   $246$   $F6_{16}$  2)  $1101111_2$   $111$   $6F_{16}$   
3)  $1111010_2$   $172_8$   $7A_{16}$  4)  $11011000_2$   $330_8$   $216$   
5)  $101100100_2$  6)  $13775_8$  7)  $1ACD_{16}$  8)  $-10011110_2$  9)  $-5216_8$  10)  $-8B3_{16}$

----- Вариант 16 -----

- 1)  $202_8$   $130$   $82_{16}$  2)  $11000011_2$   $195$   $C3_{16}$   
3)  $11000100_2$   $304_8$   $C4_{16}$  4)  $1110010_2$   $162_8$   $114$   
5)  $100001111_2$  6)  $17371_8$  7)  $1CA6_{16}$  8)  $-10001011_2$  9)  $-5715_8$  10)  $-B8C_{16}$

----- Вариант 17 -----

- 1)  $164_8$   $116$   $74_{16}$  2)  $11101010_2$   $234$   $EA_{16}$   
3)  $10001100_2$   $214_8$   $8C_{16}$  4)  $1100100_2$   $144_8$   $100$   
5)  $101000011_2$  6)  $12663_8$  7)  $1A7A_{16}$  8)  $-1010111_2$  9)  $-4324_8$  10)  $-BBE_{16}$

----- Вариант 18 -----

- 1)  $157_8$   $111$   $6F_{16}$  2)  $10111101_2$   $189$   $BD_{16}$   
3)  $10011001_2$   $231_8$   $99_{16}$  4)  $10011000_2$   $230_8$   $152$   
5)  $110001001_2$  6)  $15314_8$  7)  $19CE_{16}$  8)  $-1100000_2$  9)  $-4032_8$  10)  $-B38_{16}$

----- Вариант 19 -----

- 1)  $346_8$   $230$   $E6_{16}$  2)  $11100001_2$   $225$   $E1_{16}$   
3)  $10000110_2$   $206_8$   $86_{16}$  4)  $10000011_2$   $203_8$   $131$   
5)  $100111001_2$  6)  $15773_8$  7)  $1680_{16}$  8)  $-11000011_2$  9)  $-4357_8$  10)  $-962_{16}$

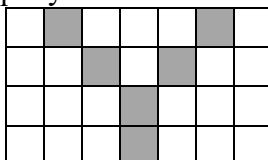
----- Вариант 20 -----

- 1)  $351_8$   $233$   $E9_{16}$  2)  $11100011_2$   $227$   $E3_{16}$   
3)  $11010011_2$   $323_8$   $D3_{16}$  4)  $11101000_2$   $350_8$   $232$   
5)  $11100100_2$  6)  $14017_8$  7)  $14C8_{16}$  8)  $-110100_2$  9)  $-4300_8$  10)  $-AB9_{16}$

## Контрольная работа «Кодирование информации»

### Вариант 1

1. Текст длиной 32768 символов закодирован с помощью алфавита, содержащего 64 символа. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 28672 символа занимает в памяти 21 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 92 на 2048 пикселей закодирован с палитрой 2 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 512 на 384 пикселей занимает в памяти 168 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:

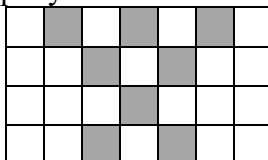


Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 2 минуты оцифрован с частотой 11000 Гц. Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 2

1. Текст длиной 57344 символов закодирован с помощью алфавита, содержащего 32 символа. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 10240 символов занимает в памяти 5 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 448 на 128 пикселей закодирован с палитрой 2 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 128 на 128 пикселей занимает в памяти 6 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:

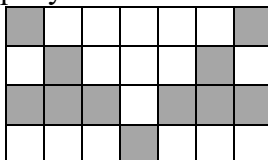


Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 11000 Гц. Разрядность кодирования - 16 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 3

1. Текст длиной 49152 символов закодирован с помощью алфавита, содержащего 2048 символов. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 32768 символа занимает в памяти 40 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 192 на 128 пикселей закодирован с палитрой 32 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 1280 на 160 пикселей занимает в памяти 175 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 11000 Гц. Разрядность кодирования - 16 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 4

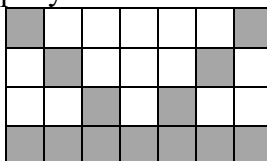
1. Текст длиной 36864 символов закодирован с помощью алфавита, содержащего 1024 символа. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 49152 символов занимает в памяти 42 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 512 на 64 пикселей закодирован с палитрой 2 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 288 на 256 пикселей занимает в памяти 18 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 2 минуты оцифрован с частотой 44100 Гц. Разрядность кодирования - 16 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 5

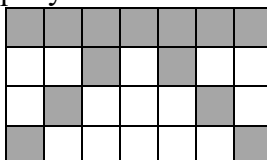
1. Текст длиной 65536 символов закодирован с помощью алфавита, содержащего 128 символов. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 40960 символов занимает в памяти 55 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 80 на 192 пикселей закодирован с палитрой 256 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 1536 на 64 пикселей занимает в памяти 24 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 11000 Гц. Разрядность кодирования - 32 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 6

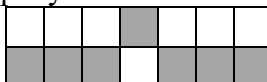
1. Текст длиной 57344 символов закодирован с помощью алфавита, содержащего 8 символов. Сколько килобайт занимает в памяти этот текст?

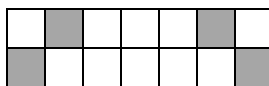
2. Сообщение длиной 20480 символа занимает в памяти 25 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 1024 на 48 пикселей закодирован с палитрой 32 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 256 на 240 пикселей занимает в памяти 45 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:





Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 2 минуты оцифрован с частотой 44100 Гц. Разрядность кодирования - 22 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 7

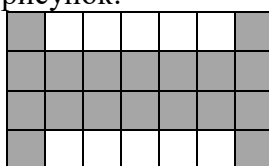
1. Текст длиной 24576 символов закодирован с помощью алфавита, содержащего 1024 символа. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 7168 символов занимает в памяти 7 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 64 на 272 пикселей закодирован в режиме истинного цвета (*True Color*). Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 768 на 160 пикселей занимает в памяти 75 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 22000 Гц. Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 8

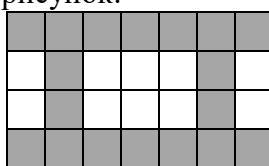
1. Текст длиной 14336 символов закодирован с помощью алфавита, содержащего 16 символов. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 28672 символа занимает в памяти 35 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 192 на 288 пикселей закодирован с палитрой 16 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 56 на 128 пикселей занимает в памяти 7 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 8000 Гц. Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 9

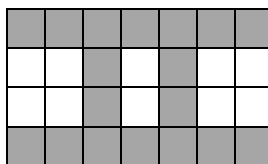
1. Текст длиной 10240 символов закодирован с помощью алфавита, содержащего 16 символов. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 65536 символов занимает в памяти 72 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 128 на 704 пикселей закодирован с палитрой 32 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 256 на 112 пикселей занимает в памяти 14 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 2 минуты оцифрован с частотой 11000 Гц. Разрядность кодирования - 22 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 10

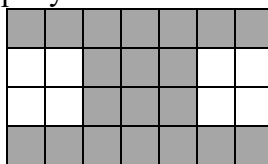
1. Текст длиной 10240 символов закодирован с помощью алфавита, содержащего 16 символов. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 36864 символа занимает в памяти 45 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 1408 на 64 пикселей закодирован с палитрой 8 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 256 на 104 пикселей занимает в памяти 13 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 3 минуты оцифрован с частотой 44100 Гц. Разрядность кодирования - 20 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 11

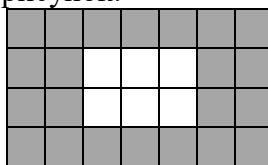
1. Текст длиной 73728 символов закодирован с помощью алфавита, содержащего 8 символов. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 49152 символов занимает в памяти 42 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 1792 на 64 пикселей закодирован с палитрой 4 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 160 на 256 пикселей занимает в памяти 15 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 11000 Гц. Разрядность кодирования - 8 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

#### Вариант 12

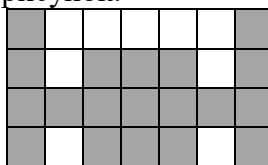
1. Текст длиной 28672 символов закодирован с помощью алфавита, содержащего 64 символа. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 73728 символов занимает в памяти 63 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 640 на 64 пикселей закодирован с палитрой 64 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 176 на 512 пикселей занимает в памяти 33 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 2 минуты оцифрован с частотой 11000 Гц. Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 13

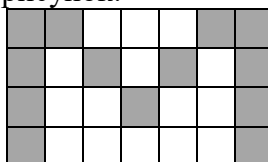
1. Текст длиной 73728 символов закодирован с помощью алфавита, содержащего 128 символов. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 24576 символа занимает в памяти 18 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 192 на 512 пикселей закодирован с палитрой 8 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 256 на 272 пикселей занимает в памяти 51 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 44100 Гц. Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 14

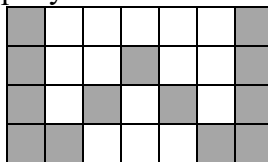
1. Текст длиной 57344 символов закодирован с помощью алфавита, содержащего 8 символов. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 73728 символов занимает в памяти 27 Кбайт. Найдите мощность алфавита, который использовался при кодировании.

3. Рисунок размером 512 на 272 пикселей закодирован с палитрой 2 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?

4. Рисунок размером 80 на 128 пикселей занимает в памяти 10 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.

5. Дан черно-белый растровый рисунок:



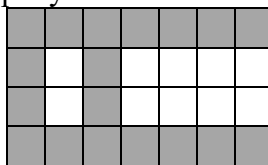
Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 44100 Гц. Разрядность кодирования - 8 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 15

1. Текст длиной 49152 символов закодирован с помощью алфавита, содержащего 32 символа. Сколько килобайт занимает в памяти этот текст?

2. Сообщение длиной 6144 символов занимает в памяти 6 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 4096 на 58 пикселей закодирован с палитрой 128 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 2048 на 64 пикселей занимает в памяти 16 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:

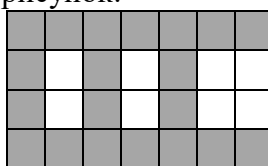


Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 2 минуты оцифрован с частотой 11000 Гц. Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 16

1. Текст длиной 49152 символов закодирован с помощью алфавита, содержащего 32 символа. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 73728 символа занимает в памяти 45 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 64 на 256 пикселей закодирован с палитрой 4 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 512 на 256 пикселей занимает в памяти 48 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:

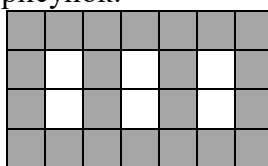


Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 22000 Гц. Разрядность кодирования - 22 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 17

1. Текст длиной 20480 символов закодирован с помощью алфавита, содержащего 64 символа. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 65536 символов занимает в памяти 88 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 64 на 64 пикселей закодирован с палитрой 256 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 68 на 2048 пикселей занимает в памяти 119 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:



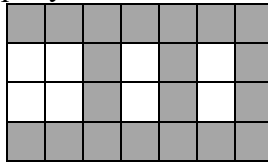
Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 3 минуты оцифрован с частотой 11000 Гц. Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.



### Вариант 18

1. Текст длиной 20480 символов закодирован с помощью алфавита, содержащего 1024 символа. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 49152 символов занимает в памяти 66 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 256 на 76 пикселей закодирован с палитрой 256 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 448 на 384 пикселей занимает в памяти 63 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:

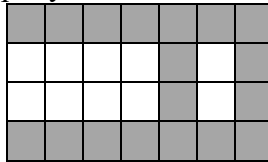


Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 22000 Гц. Разрядность кодирования - 16 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 19

1. Текст длиной 57344 символов закодирован с помощью алфавита, содержащего 2048 символов. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 65536 символов занимает в памяти 24 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 64 на 3584 пикселей закодирован с палитрой 8 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 2048 на 48 пикселей занимает в памяти 84 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:

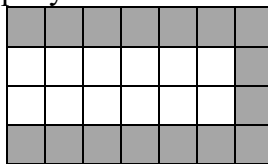


Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 1 минута оцифрован с частотой 44100 Гц. Разрядность кодирования - 20 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

### Вариант 20

1. Текст длиной 57344 символов закодирован с помощью алфавита, содержащего 512 символов. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 49152 символов занимает в памяти 54 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 768 на 96 пикселей закодирован с палитрой 8 цветов. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 736 на 128 пикселей занимает в памяти 69 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

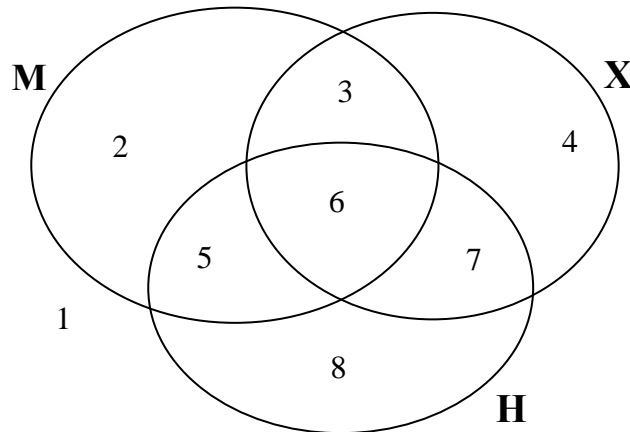
6. Звук длительностью 3 минуты оцифрован с частотой 11000 Гц. Разрядность кодирования - 16 бит. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

**Ответы к контрольной работе «Кодирование информации»**

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>1</b>	<b>24</b>	<b>64</b>	<b>23</b>	<b>128</b>	<b>4450408</b>	<b>3,78</b>
<b>2</b>	<b>35</b>	<b>16</b>	<b>7</b>	<b>8</b>	<b>5450414</b>	<b>1,26</b>
<b>3</b>	<b>66</b>	<b>1024</b>	<b>15</b>	<b>128</b>	<b>828BB88</b>	<b>1,26</b>
<b>4</b>	<b>45</b>	<b>128</b>	<b>4</b>	<b>4</b>	<b>8288A7F</b>	<b>10,09</b>
<b>5</b>	<b>56</b>	<b>2048</b>	<b>15</b>	<b>4</b>	<b>FE51141</b>	<b>2,52</b>
<b>6</b>	<b>21</b>	<b>1024</b>	<b>30</b>	<b>64</b>	<b>11DD141</b>	<b>13,88</b>
<b>7</b>	<b>30</b>	<b>256</b>	<b>51</b>	<b>32</b>	<b>83FFFC1</b>	<b>3,78</b>
<b>8</b>	<b>7</b>	<b>1024</b>	<b>27</b>	<b>256</b>	<b>FE8917F</b>	<b>1,37</b>
<b>9</b>	<b>5</b>	<b>512</b>	<b>55</b>	<b>16</b>	<b>FE50A7F</b>	<b>3,46</b>
<b>10</b>	<b>5</b>	<b>1024</b>	<b>33</b>	<b>16</b>	<b>FE70E7F</b>	<b>18,93</b>
<b>11</b>	<b>27</b>	<b>128</b>	<b>28</b>	<b>8</b>	<b>FF8F1FF</b>	<b>0,63</b>
<b>12</b>	<b>21</b>	<b>128</b>	<b>30</b>	<b>8</b>	<b>8377FDD</b>	<b>3,78</b>
<b>13</b>	<b>63</b>	<b>64</b>	<b>36</b>	<b>64</b>	<b>C7564C1</b>	<b>7,57</b>
<b>14</b>	<b>21</b>	<b>8</b>	<b>17</b>	<b>256</b>	<b>8326AE3</b>	<b>2,52</b>
<b>15</b>	<b>30</b>	<b>256</b>	<b>203</b>	<b>2</b>	<b>FF4287F</b>	<b>3,78</b>
<b>16</b>	<b>30</b>	<b>32</b>	<b>4</b>	<b>8</b>	<b>FF52A7F</b>	<b>3,46</b>
<b>17</b>	<b>15</b>	<b>2048</b>	<b>4</b>	<b>128</b>	<b>FF56AFF</b>	<b>5,66</b>
<b>18</b>	<b>25</b>	<b>2048</b>	<b>19</b>	<b>8</b>	<b>FE54AFF</b>	<b>2,52</b>
<b>19</b>	<b>77</b>	<b>8</b>	<b>84</b>	<b>128</b>	<b>FE142FF</b>	<b>6,31</b>
<b>20</b>	<b>63</b>	<b>512</b>	<b>27</b>	<b>64</b>	<b>FE040FF</b>	<b>3,78</b>

### Контрольная работа «Логические основы компьютеров»

1. Построить таблицу истинности для заданного логического выражения (можно сначала упростить).
2. Записать и упростить выражение для объединения областей на диаграмме



3. Записать логическое высказывание, обратное данному.
4. Построить логическое выражение по таблице истинности.
5. Построить схему на логических элементах. Упрощать выражение не нужно.

Вариант 1	Вариант 2	Вариант 3																																																																																																												
<p>1. <math>X = (B \rightarrow A) \cdot (\overline{B} \cdot \overline{C})</math></p> <p>2. 3+5+6</p> <p>3. Вася высокий, и Петя ушел за хлебом.</p> <p>4.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (\overline{A+B}) \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C}</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1	<p>1. <math>X = (\overline{A \rightarrow B}) + (\overline{C \rightarrow B})</math></p> <p>2. 2+5+6</p> <p>3. Семен пришел поздно, или Вася опоздал на поезд.</p> <p>4.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = \overline{A \cdot C} + \overline{B \cdot C} + \overline{A} \cdot \overline{B} \cdot C</math></p>	A	B	C	X	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1	<p>1. <math>X = (\overline{A \rightarrow B}) \cdot (\overline{B \rightarrow C})</math></p> <p>2. 2+3+6</p> <p>3. Машина стоит у подъезда, и идет дождь.</p> <p>4.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (\overline{B+C}) \cdot \overline{A} + A \cdot \overline{B} \cdot C</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	1																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	1																																																																																																											
0	1	0	0																																																																																																											
0	1	1	0																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	1																																																																																																											
0	1	1	0																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
Вариант 4	Вариант 5	Вариант 6																																																																																																												
<p>1. <math>X = (\overline{A \rightarrow C}) + (\overline{A \rightarrow B})</math></p> <p>2. 2+3+5</p> <p>3. Самолет летит в Москву или в Париж.</p> <p>4.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = A + B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C</math></p>	A	B	C	X	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	1	1	1	1	1	<p>1. <math>X = (\overline{B \rightarrow C}) \cdot (\overline{A \rightarrow C})</math></p> <p>2. 4+7+8</p> <p>3. Окунь живет около дна, и сегодня вторник.</p> <p>4.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (\overline{A+B}) \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	<p>1. <math>X = (\overline{B \rightarrow C}) + (\overline{A \rightarrow C})</math></p> <p>2. 1+7+8</p> <p>3. Маша блондинка, или ее соседка - брюнетка.</p> <p>4.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = \overline{A} \cdot (\overline{B+C}) + \overline{A} \cdot B</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1
A	B	C	X																																																																																																											
0	0	0	1																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	0																																																																																																											
1	0	0	1																																																																																																											
1	0	1	0																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	1																																																																																																											
0	1	1	1																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	0																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	1																																																																																																											
1	0	0	1																																																																																																											
1	0	1	1																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
Вариант 7	Вариант 8	Вариант 9																																																																																																												

<p>1. <math>X = (\overline{A} \rightarrow \overline{B}) + (\overline{A} \rightarrow \overline{C})</math></p> <p>2. 1+4+7</p> <p>3. Стол сделан из дерева, и жираф летит на север.</p> <p>4.</p> <table border="1" data-bbox="256 349 488 696"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (A+B) \cdot (B+\overline{C}) + C \cdot B</math></p>	A	B	C	X	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	<p>1. <math>X = (\overline{A} \rightarrow \overline{B}) + (\overline{A} \rightarrow \overline{C})</math></p> <p>2. 1+4+8</p> <p>3. Мопед быстрее носорога, или кролик крупнее удава.</p> <p>4.</p> <table border="1" data-bbox="699 333 930 680"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (B+\overline{C}) \cdot (A+\overline{C}) + \overline{A} \cdot B</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	1	1	1	0	1	1	1	1	1	<p>1. <math>X = (B \oplus A) \cdot (\overline{B} \cdot \overline{C})</math></p> <p>2. 3+6+7</p> <p>3. Монитор стоит в подвале, и мышка убежала.</p> <p>4.</p> <table border="1" data-bbox="1150 353 1382 701"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (B+\overline{C}) \cdot A + A \cdot \overline{B}</math></p>	A	B	C	X	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1
A	B	C	X																																																																																																											
0	0	0	1																																																																																																											
0	0	1	1																																																																																																											
0	1	0	0																																																																																																											
0	1	1	1																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	0																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	0																																																																																																											
1	0	0	0																																																																																																											
1	0	1	1																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	1																																																																																																											
0	1	0	0																																																																																																											
0	1	1	1																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
<b>Вариант 10</b>																																																																																																														
<p>1. <math>X = (\overline{A} \oplus \overline{B}) + (\overline{C} \rightarrow \overline{B})</math></p> <p>2. 4+6+7</p> <p>3. Все слоны серые, или дважды два – пять.</p> <p>4.</p> <table border="1" data-bbox="256 992 488 1339"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (A+\overline{C}) \cdot B + \overline{A} \cdot \overline{B} \cdot C</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	1	1	1	1	1	<p>1. <math>X = (\overline{A} \rightarrow \overline{B}) \cdot (\overline{B} \oplus \overline{C})</math></p> <p>2. 3+4+6</p> <p>3. Клара – артистка, и Фаина – инженер.</p> <p>4.</p> <table border="1" data-bbox="699 1003 930 1350"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (\overline{A}+\overline{C}) \cdot B + A \cdot \overline{B} \cdot \overline{C}</math></p>	A	B	C	X	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1	<p>1. <math>X = (\overline{A} \rightarrow \overline{C}) + (\overline{A} \oplus \overline{B})</math></p> <p>2. 3+4+7</p> <p>3. Вчера шел снег, или сегодня солнечно.</p> <p>4.</p> <table border="1" data-bbox="1150 1005 1382 1352"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> <p>5. <math>X = \overline{A} + B \cdot C + A \cdot B \cdot \overline{C}</math></p>	A	B	C	X	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	0
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	0																																																																																																											
1	0	0	1																																																																																																											
1	0	1	0																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	1																																																																																																											
0	0	1	0																																																																																																											
0	1	0	1																																																																																																											
0	1	1	0																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	1																																																																																																											
0	0	1	1																																																																																																											
0	1	0	1																																																																																																											
0	1	1	0																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	0																																																																																																											
1	1	1	0																																																																																																											
<b>Вариант 13</b>																																																																																																														
<p>1. <math>X = (\overline{B} \rightarrow \overline{C}) \cdot (\overline{A} \oplus C)</math></p> <p>2. 2+5+8</p> <p>3. Лампа светит ярко, и форточка закрыта.</p> <p>4.</p> <table border="1" data-bbox="256 1626 488 1973"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (B+\overline{C}) \cdot A + A \cdot B \cdot \overline{C}</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	<p>1. <math>X = (\overline{B} \rightarrow C) + (\overline{A} \oplus C)</math></p> <p>2. 1+5+8</p> <p>3. Звук идет из-под земли, или Пятачок – волшебник.</p> <p>4.</p> <table border="1" data-bbox="687 1655 919 2002"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (B+A) \cdot C + \overline{A} \cdot B \cdot \overline{C}</math></p>	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	1	1	<p>1. <math>X = (\overline{A} \oplus \overline{B}) + (\overline{A} \rightarrow \overline{C})</math></p> <p>2. 1+2+5</p> <p>3. На карте есть река, и этот дом – старый.</p> <p>4.</p> <table border="1" data-bbox="1142 1637 1374 1984"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>5. <math>X = (\overline{C} + A) \cdot B + A \cdot B \cdot \overline{C}</math></p>	A	B	C	X	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	1																																																																																																											
0	1	1	1																																																																																																											
1	0	0	0																																																																																																											
1	0	1	1																																																																																																											
1	1	0	0																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	0																																																																																																											
1	0	0	1																																																																																																											
1	0	1	1																																																																																																											
1	1	0	0																																																																																																											
1	1	1	1																																																																																																											
A	B	C	X																																																																																																											
0	0	0	1																																																																																																											
0	0	1	1																																																																																																											
0	1	0	0																																																																																																											
0	1	1	1																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	0																																																																																																											
1	1	1	1																																																																																																											
<b>Вариант 14</b>																																																																																																														
<b>Вариант 15</b>																																																																																																														

Вариант 16	Вариант 17	Вариант 18																																																																																																																																																
1. $X = (\overline{A} \oplus B) + (A \rightarrow \overline{C})$ 2. 1+2+8 3. Это двухэтажный дом, или Карлсон ест варенье. 4. <table border="1" style="margin-left: 20px;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> 5. $X = (\overline{C} + \overline{A}) \cdot \overline{B} + A \cdot B$	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1. $X = (\overline{A} \rightarrow \overline{C}) \cdot (\overline{A} \oplus \overline{B})$ 2. 5+6+7 3. Сканер вводит рисунки, и принтер печатает. 4. <table border="1" style="margin-left: 20px;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> 5. $X = (B + \overline{A}) \cdot \overline{C} + A \cdot B$	A	B	C	X	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1. $X = (\overline{B} \rightarrow \overline{C}) + (\overline{A} \oplus C)$ 2. 5+6+8 3. Если Иванов – школьник, то Петров – милиционер. 4. <table border="1" style="margin-left: 20px;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> 5. $X = A \cdot (B + \overline{C}) + A \cdot \overline{B}$	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	1	1	1	0	1	1	1	1	1																																				
A	B	C	X																																																																																																																																															
0	0	0	0																																																																																																																																															
0	0	1	0																																																																																																																																															
0	1	0	0																																																																																																																																															
0	1	1	1																																																																																																																																															
1	0	0	0																																																																																																																																															
1	0	1	1																																																																																																																																															
1	1	0	0																																																																																																																																															
1	1	1	1																																																																																																																																															
A	B	C	X																																																																																																																																															
0	0	0	0																																																																																																																																															
0	0	1	1																																																																																																																																															
0	1	0	0																																																																																																																																															
0	1	1	0																																																																																																																																															
1	0	0	0																																																																																																																																															
1	0	1	1																																																																																																																																															
1	1	0	0																																																																																																																																															
1	1	1	1																																																																																																																																															
A	B	C	X																																																																																																																																															
0	0	0	0																																																																																																																																															
0	0	1	0																																																																																																																																															
0	1	0	1																																																																																																																																															
0	1	1	0																																																																																																																																															
1	0	0	0																																																																																																																																															
1	0	1	1																																																																																																																																															
1	1	0	1																																																																																																																																															
1	1	1	1																																																																																																																																															
Вариант 19	Вариант 20	Вариант 21																																																																																																																																																
1. $X = (\overline{B} \rightarrow C) \cdot (\overline{A} \oplus C)$ 2. 6+7+8 3. Если клякса, большая, то Земля маленькая. 4. <table border="1" style="margin-left: 20px;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> 5. <table border="1" style="margin-left: 20px;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> 6. $X = C \cdot (A + \overline{B}) + A \cdot \overline{B}$	A	B	C	X	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	1	1	A	B	C	X	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	1	1	1. $X = (\overline{A} \oplus \overline{B}) \cdot (\overline{A} \rightarrow \overline{C})$ 2. 5+7+8 3. Вася моет раму, или Петя поливает цветы. 4. <table border="1" style="margin-left: 20px;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> 5. $X = C \cdot (A + \overline{B}) + B \cdot \overline{C}$	A	B	C	X	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	1. $X = (\overline{A} \oplus B) \cdot (\overline{A} \rightarrow \overline{C})$ 2. 2+3+4 3. Мама моет Васю, и кошка пьет молоко. 4. <table border="1" style="margin-left: 20px;"> <thead> <tr><th>A</th><th>B</th><th>C</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> 5. $X = (A + C) \cdot (B + C) + \overline{B} \cdot \overline{C}$	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	1	1	0	1	0	1	1	0	1	1	1	1	1
A	B	C	X																																																																																																																																															
0	0	0	1																																																																																																																																															
0	0	1	0																																																																																																																																															
0	1	0	0																																																																																																																																															
0	1	1	0																																																																																																																																															
1	0	0	1																																																																																																																																															
1	0	1	1																																																																																																																																															
1	1	0	0																																																																																																																																															
1	1	1	1																																																																																																																																															
A	B	C	X																																																																																																																																															
0	0	0	1																																																																																																																																															
0	0	1	0																																																																																																																																															
0	1	0	0																																																																																																																																															
0	1	1	0																																																																																																																																															
1	0	0	1																																																																																																																																															
1	0	1	1																																																																																																																																															
1	1	0	0																																																																																																																																															
1	1	1	1																																																																																																																																															
A	B	C	X																																																																																																																																															
0	0	0	0																																																																																																																																															
0	0	1	1																																																																																																																																															
0	1	0	0																																																																																																																																															
0	1	1	1																																																																																																																																															
1	0	0	0																																																																																																																																															
1	0	1	0																																																																																																																																															
1	1	0	0																																																																																																																																															
1	1	1	1																																																																																																																																															
A	B	C	X																																																																																																																																															
0	0	0	0																																																																																																																																															
0	0	1	0																																																																																																																																															
0	1	0	0																																																																																																																																															
0	1	1	1																																																																																																																																															
1	0	0	1																																																																																																																																															
1	0	1	0																																																																																																																																															
1	1	0	1																																																																																																																																															
1	1	1	1																																																																																																																																															

### Ответы к контрольной работе

#### Задание 1.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0
1	0	0	0	0	1	1	1	0	1	0	0	0	1	0	1	0	0	1	0	0
0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	1	0	0	1
0	1	0	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	1	0	1
0	1	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	0	0	0	1
1	1	0	0	0	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	0
1	0	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	0	1	0	0
1	1	0	1	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0	0

#### Задание 2.

1	$M(X + H)$	7	$\overline{M}(X + \overline{H})$	13	$\overline{X}(M + H)$	19	$H(X + \overline{M})$
2	$M(\overline{X} + H)$	8	$\overline{M}(\overline{X} + \overline{H})$	14	$\overline{X}(\overline{M} + H)$	20	$H(\overline{X} + \overline{M})$
3	$M(X + \overline{H})$	9	$X(M + H)$	15	$\overline{X}(M + \overline{H})$	21	$\overline{H}(X + M)$

4	$M(\bar{X} + \bar{H})$	10	$X(\bar{M} + H)$	16	$\bar{X}(\bar{M} + \bar{H})$		
5	$\bar{M}(X + H)$	11	$X(M + \bar{H})$	17	$H(X + M)$		
6	$\bar{M}(\bar{X} + H)$	12	$X(\bar{M} + \bar{H})$	18	$H(\bar{X} + M)$		

Задание 4.

1	$AB + BC$	7	$\bar{A}\bar{B} + BC$	13	$\bar{A}B + AC$	19	$AC + \bar{B}\bar{C}$
2	$AB + \bar{B}C$	8	$AB + AC$	14	$A\bar{B} + AC$	20	$\bar{A}C + BC$
3	$AB + B\bar{C}$	9	$AB + \bar{A}C$	15	$\bar{A}\bar{B} + BC$	21	$A\bar{C} + BC$
4	$AB + \bar{B}\bar{C}$	10	$AB + A\bar{C}$	16	$AC + BC$		
5	$\bar{A}B + BC$	11	$AB + \bar{A}\bar{C}$	17	$AC + \bar{B}C$		
6	$A\bar{B} + BC$	12	$\bar{A}\bar{B} + \bar{A}\bar{C}$	18	$AC + B\bar{C}$		

## Контрольные работы «Основы языка Python»

### Уровень А.

1. С клавиатуры вводится трёхзначное число. Нужно вывести ответ «Да», если все его цифры четные, и ответ «Нет» в остальных случаях.

2. Определите значение переменной *s* после выполнения фрагмента программы:

```
1. s = 0
```

```
2. k = 0
```

```
3. while s < 1024:
```

```
4. s = s + 10
```

```
5. k += 1
```

```
6. s:=0;
```

```
7. k:=0;
```

```
8. while s < 1024 do begin
```

```
9. s:=s+10;
```

```
10. k:=k+1;
```

```
11. end;
```

3. Определите значение переменной *s* после выполнения фрагмента программы:

```
12. s = 0
```

```
13. for k in range(-3,6):
```

```
14. s = s + k
```

```
15. s:=0;
```

```
16. for k:=-3 to 5 do begin
```

```
17. s:=s+k;
```

```
18. end;
```

4. Определите результат работы функции при входном значении, равном 123456:

```
19. def qq(x):
```

```
20. s = 0
```

```
21. for i in range(1, 4):
```

```
22. s = s + x % 10
```

```
23. x = x // 10
```

```
24. return s
```

```
25. function qq(x: integer):integer;
```

```
26. var i, s: integer;
```

```
27. begin
```

```
28. s:=0;
```

```
29. for i:=1 to 3 do begin
```

```
30. s := s + x mod 10;
```

```
31. x := x div 10;
```

```
32. end;
```

```
33. qq := s;
```

```
34. end;
```

### Уровень В.

1) С клавиатуры вводится трёхзначное число. Нужно вывести ответ «Да», если все его цифры четные, и ответ «Нет» в остальных случаях. Если введено не трёхзначное число, должно быть выведено сообщение «Неверное число».

2) Укажите наименьшее и наибольшее из таких чисел *x*, при вводе которых алгоритм печатает сначала 3, а потом 6:

```
x = int(input())
```

```
L = 0; M = 0
```

```
while x > 0:
```

```
    L = L + 1
```

```
    M = M + x % 10
```

```
    x = x // 10
```

```
print(L)
```

```
print(M)
```

```
var x, L, M: integer;
```

```
begin
```

```
    readln(x);
```

```
    L:=0; M:=0;
```

```
    while x > 0 do begin
```

```
        L:= L + 1;
```

```
        M:= M + x mod 10;
```

```
        x:= x div 10;
```

```
    end;
```

```
    writeln(L); write(M);
```

```
end.
```

3) Определите значение переменной *s* после выполнения фрагмента программы:

```
s = 0
```

```
for k in range(1,6):
```

```
    for j in range(1,k+1):
```

```
        s = s + k
```

```
s:=0;
```

```
for k:=1 to 5 do
```

```
    for j:=1 to k do
```

```
        s:=s+k;
```

4) Определите результат работы функции входном значении, равном 123456:

```
def qq(x):
```

```
    s = 0
```

```
function qq(x: integer):integer;
```

```
var i, s: integer;
```

```

while x > 0:
    s = 10*s + x % 10
    x = x // 10
return s

```

```

begin
    s:=0;
    while x > 0 do begin
        s:= 10*s + x mod 10;
        x:= x div 10;
    end;
    qq:= s;
end;

```

### Уровень С.

- 1) С клавиатуры вводится целое число от 1 до 1000, обозначающая сумму в рублях. Программа должна напечатать эту сумму прописью, например:

**двадцать один рубль**  
**сто семьдесят три рубля**  
**девятьсот пятнадцать рублей**

- 2) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите максимальное и минимальное числа, при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
L = 0; M = 0
while x > 0:
    L = L + 1
    if x % 2 == 1:
        M = M + (x % 10) // 2
    x = x // 10
print(L)
print(M)

```

```

var x, L, M: integer;
begin
    readln(x);
    L:=0; M:=0;
    while x > 0 do begin
        L:= L + 1;
        if x mod 2 = 1 then
            M:= M +
                (x mod 10) div 2;
        x:= x div 10;
    end;
    writeln(L); write(M);
end.

```

- 3) Определите значение переменной  $s$  после выполнения фрагмента программы:

```

s = 0
for k in range(1,6):
    for j in range(1,k+1):
        for m in range(1,j+1):
            s = s + k + j + m

```

```

s:=0;
for k:=1 to 5 do
    for j:=1 to k do
        for m:=1 to j do
            s:=s+k+j+m;

```

- 4) Определите результат работы функции при входных значениях, равных 123456 и 78901:

```

def qq(x, y):
    s = 0
    while x > 0 and y > 0:
        s = 10*s + (x+y) % 10
        x = x // 10
        y = y // 10
    return s

```

```

function qq(x,y: integer)
    :integer;
var i, s: integer;
begin
    s:=0;
    while (x > 0) and (y > 0) do
        begin
            s:= 10*s + (x+y) mod 10;
            x:= x div 10;
            y:= y div 10;
        end;
    qq:= s;
end;

```



## **Контрольная работа «Массивы»**

### **Уровень А.**

- 1) Введите массив из 5 элементов с клавиатуры и найдите количество элементов, которые оканчиваются на 8 и делятся на 3.
- 2) Заполните массив случайными числами в интервале [100,999] и переставьте их по возрастанию средней цифры.
- 3) Заполните массив случайными числами в интервале [0,1000] и выведите номера всех элементов, которые равны последнему элементу.

### **Уровень В.**

- 1) Введите массив из 5 элементов с клавиатуры и найдите количество элементов, старшая цифра которых равна 8.
- 2) Заполните массив случайными числами в интервале [100,999] и переставьте их по возрастанию суммы цифр.
- 3) Заполните массив случайными числами в интервале [0,1000] и выведите номера двух соседних элементов, произведение которых наибольшее.

### **Уровень С.**

- 1) Введите массив из 5 элементов с клавиатуры и найдите количество элементов, запись которых в двоичной системе – палиндром (например,  $10101_2 = 21$ ).
- 2) Заполните массив случайными числами в интервале [100,999] и переставьте их по возрастанию количества делителей.
- 3) Заполните массив случайными числами в интервале [0,1000] и выведите номера трёх различных минимальных элементов. Разрешается сделать только один проход по массиву.

## **Контрольная работа «Символьные строки»**

### **Уровень А.**

- 1) С клавиатуры вводится символьная строка. Определите, сколько в ней цифр.
- 2) С клавиатуры вводится предложение, в котором каждые два слова разделены ровно пробелом. Определите, сколько слов в этом предложении.
- 3) Напишите функцию, которая удаляет из переданной ей символьной строки все символы '<' и '>'.

### **Уровень В.**

- 1) С клавиатуры вводится символьная строка. Определите, какой символ встречается в ней чаще всего.
- 2) С клавиатуры вводится предложение, в котором слова разделены пробелами. Определите, сколько слов в этом предложении начинается на букву «а».
- 3) Напишите функцию, которая удаляет из переданной ей символьной строки все лишние пробелы, заменяя повторяющиеся пробелы на один пробел.

### **Уровень С.**

- 1) С клавиатуры вводится символьная строка. Определите, сколько раз встречается в строке каждый из входящих в нее символов.
- 2) С клавиатуры вводится предложение, в котором слова разделены пробелами. Определите, сколько слов в этом предложении начинается и заканчивается на одну и ту же букву.
- 3) Напишите функцию, которая удаляет из переданной ей символьной строки все символы, заключенные в угловые скобки '<' и '>' (вместе с этими скобками).

## **Контрольная работа «Файлы»**

**Уровень А.** В каждой строке файла записано три числа, каждое из которых находится в интервале  $[0,255]$ . Эти числа обозначают красную, зелёную и синюю составляющие цвета в цветовой модели RGB. Запишите в другой файл полные коды заданных цветов в виде целых чисел (код цвета =  $256 \cdot 256 \cdot R + 256 \cdot G + B$ ).

**Уровень В.** В каждой строке файла записано три числа, каждое из которых находится в интервале  $[0,255]$ . Эти числа обозначают красную, зелёную и синюю составляющие цвета в цветовой модели RGB. Запишите в другой файл полные коды заданных цветов в виде целых чисел в шестнадцатеричной системе счисления (цвет (128,255,255) запишется как 80FFFF).

**Уровень С.** В файле записаны данные о результатах сдачи экзаменов по математике, русскому языку и физике. Каждая строка содержит фамилию, имя и три целых числа – количество баллов, набранных участником по каждому предмету:

**<Фамилия> <Имя> <Математика> <Русский язык> <Физика>**

Вывести в другой файл данные учеников, которые получили в сумме больше 240 баллов. Список должен быть отсортирован по убыванию балла. Формат выходных данных:

- 1) П. Иванов 284
- 2) И. Петров 275

## 11 класс

### Контрольная работа № 1 по теме «Информация и информационные процессы»

#### Вариант №1

##### Часть А.

1. Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=1, Б=000, В=001. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?  
1) 00 2) 01 3) 11 4) 010
2. По каналу связи передаются сообщения, содержащие только 4 буквы: А, Б, В, Г. Для кодирования букв А, Б, В используются 5-битовые кодовые слова: А - 11111, Б - 11000, В - 00100. Для этого набора кодовых слов выполнено такое свойство: любые два слова из набора отличаются не менее чем в трёх позициях. Это свойство важно для расшифровки сообщений при наличии помех. Какое из перечисленных ниже кодовых слов можно использовать для буквы Г, чтобы указанное свойство выполнялось для всех четырёх кодовых слов?  
1) 00000 2) 00011 3) 11100 4) не подходит ни одно из указанных выше слов
3. За 33 секунд передаётся сообщение занимающие 40 страниц по 7920 символов на каждой, при условии что оно закодировано при помощи символов 256-символьного алфавита. Какую скорость передачи информации имеет данный модем?
4. Сколько секунд потребуется модему, передающему сообщения со скоростью 51200 бит/с, чтобы передать 64-цветное растровое изображение размером 640 × 480 пикселей, при условии, что в каждом байте закодировано максимально возможное число пикселей?

##### Часть В.

5. После кодирования методом RLE получилась следующая последовательность байтов (первый байт – управляющий):  
00000010 10101010 00110011 10000111 00001111 00000110 10000011 10000001 11111111 10011001 11111110 11101110  
Сколько байт будет содержать данная последовательность после распаковки?
6. Для регистрации на сайте некоторой страны пользователю необходимо придумать пароль длиной ровно 11 символов. В пароле можно использовать десятичные цифры и 32 различных символа местного алфавита, причем все буквы используются в двух начертаниях – строчные и прописные. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый пароль – одинаковым и минимально возможным целым количеством байт. Определите объем памяти в байтах, необходимый для хранения 50 паролей.
7. У Кати есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации  $2^{20}$  бит в секунду. У Сергея нет скоростного доступа в Интернет, но есть возможность получать информацию от Кати по телефонному каналу со средней скоростью  $2^{13}$  бит в секунду. Сергей договорился с Катей, что она скачает для него данные объёмом 9 Мбайт по высокоскоростному каналу и ретранслирует их Сергею по низкоскоростному каналу. Компьютер Кати может начать ретрансляцию данных не раньше, чем им будут получены первые 1024 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Катей данных до полного их получения Сергеем?

##### Часть С.

8. Постройте дерево Хаффмана для фразы: ШЛА САША ПО ШОССЕ И СОСАЛА СУШКУ. Найдите коды всех входящих в неё символов. Чему равен коэффициент сжатия в сравнении с равномерным кодом минимальной длины? С однобайтовым кодом?

## Контрольная работа № 1 по теме «Информация и информационные процессы»

### Вариант №2

#### Часть А.

1. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А–111, Б–110, В–100, Г–101. Укажите, каким кодовым словом может быть закодирована буква Д. Код должен удовлетворять свойству однозначного декодирования. Если можно использовать более одного кодового слова, укажите кратчайшее из них.  
1) 0 2) 01 3) 00 4) 000
2. По каналу связи передаются сообщения, содержащие только 4 буквы: А, Б, В, Г. Для кодирования букв А, Б, В используются 5-битовые кодовые слова: А - 11100, Б - 00110, В - 01011. Для этого набора кодовых слов выполнено такое свойство: любые два слова из набора отличаются не менее чем в трёх позициях. Это свойство важно для расшифровки сообщений при наличии помех. Какое из перечисленных ниже кодовых слов можно использовать для буквы Г, чтобы указанное свойство выполнялось для всех четырёх кодовых слов?  
1) 11001 2) 10010 3) 10001 4) не подходит ни одно из указанных выше слов
3. Скорость передачи информации 12000 бит/с. Объем передаваемой информации 3 страницы. На одной странице содержится 180 символов на двух других по 3000 символов. Мощность алфавита – 32. Сколько понадобится времени для передачи данной информации?
4. Сколько секунд потребуется модему, передающему сообщения со скоростью 30000 бит/с, чтобы передать растровое изображение размером 800 × 600 пикселей, при условии, что в палитре 128 цветов?

#### Часть В.

5. После кодирования методом RLE получилась следующая последовательность байтов (первый байт – управляющий):  
10000101 10101010 00000011 10000101 00001111 00000110 10000100 10000001 00000010  
10011001 10000010  
Сколько байт будет содержать данная последовательность после распаковки?
6. В некоторой стране автомобильный номер длиной 5 символов составляется из заглавных букв (всего используется 26 букв) и десятичных цифр в любом порядке. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый номер – одинаковым и минимально возможным целым количеством байт. Определите объем памяти в байтах, необходимый для хранения 40 автомобильных номеров.
7. У Толи есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации  $2^{18}$  бит в секунду. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Толи по низкоскоростному телефонному каналу со средней скоростью  $2^{15}$  бит в секунду. Миша договорился с Толей, что тот будет скачивать для него данные объемом 11 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу. Компьютер Толи может начать ретрансляцию данных не раньше, чем им будут получены первые 512 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Толей данных до полного их получения Мишей?

#### Часть С.

8. Постройте дерево Хаффмана для фразы: ШЛА САША ПО ШОССЕ И СОСАЛА СУШКУ. Найдите коды всех входящих в неё символов. Чему равен коэффициент сжатия в сравнении с равномерным кодом минимальной длины? С однобайтовым кодом?

**Контрольная работа по теме «Моделирование и формализация» (11 класс ). Вариант – 1.**

**1. Какие пары объектов не находятся в отношении "объект - модель"?**

- А) компьютер - его фотография;
- Б) компьютер - его функциональная схема;
- В) компьютер - его процессор;
- Г) компьютер - его техническое описание.

**2. Информационной моделью, которая имеет иерархическую структуру является ...**

- А) файловая система компьютера;
- Б) расписание уроков;
- В) таблица Менделеева;
- Г) программа телепередач.

**3. Какая модель является статической (описывающей состояние объекта)?**

- А) формула химического соединения;
- Б) формулы равноускоренного движения;
- В) формула химической реакции;
- Г) второй закон Ньютона.

**4. Информационной моделью, которая имеет сетевую структуру является ...**

- А) файловая система компьютера;
- Б) таблица Менделеева;
- В) генеалогическое дерево семьи;
- Г) модель компьютерной сети Интернет.

**5. Информационной (знаковой) моделью является ...**

- А) анатомический муляж;
- Б) макет здания;
- В) модель корабля;
- Г) химическая формула.

**6. В информационных моделях разомкнутых систем управления отсутствует ...**

- А) управляющий объект;
- Б) управляемый объект;
- В) канал управления;
- Г) канал обратной связи.

**7. Какие из приведенных ниже определений понятия «модель» верные? Отметить все правильные на ваш взгляд ответы.**

- А) модель - это некое вспомогательное средство, объект, который в определенной ситуации заменяет другой объект;
- Б) модель - это новый объект, который отражает некоторые стороны изучаемого объекта или явления, существенные с точки зрения цели моделирования;
- В) модель - это физический или информационный аналог объекта, функционирование которого - по определенным параметрам - подобно функционированию реального объекта;
- Г) модель некоторого объекта - это другой объект (реальный, знаковый или воображаемый), отличный от исходного, он обладает существенными для целей моделирования свойствами и в рамках этих целей полностью заменяет исходный объект.

**8. Вставьте в предложение наиболее точный термин из предложенного ниже списка. Если материальная модель объекта - это его физическое подобие, то информационная модель объекта - это его ...**

- А) описание;
- Б) точное воспроизведение;
- В) схематичное представление;
- Г) преобразование.

**9. Какое из утверждений верно?**

- А) информационные модели одного и того же объекта, пусть даже предназначенные для разных целей, должны быть во многом сходны;
- Б) информационные модели одного и того же объекта, предназначенные для разных целей, могут быть совершенно разными.

**10. Может ли передаваться информация от человека к человеку и от поколения к поколению без использования моделей?**

- А) нет, без моделей никогда не обойтись;
- Б) да, иногда, например, генетическая информация;
- В) да, чаще всего знания передаются без использования каких-либо моделей.

**11. Верно ли, что моделирование представляет собой один из основных методов познания, способ существования знаний?**

- А) нет; Б) да.

**12. Какие из приведенных ниже моделей являются вероятностными? Выбрать три правильных ответа.**

- А) прогноз погоды;
- Б) отчет о деятельности предприятия;
- В) схема функционирования устройства;
- Г) научная гипотеза;
- Д) оглавление книги;
- Е) план мероприятий, посвященных Дню Победы.

**13. Правильно ли определен вид следующей модели: «Компьютерная модель полета мяча, брошенного вертикально вверх, - динамическая формализованная модель, имитирующая поведение данного объекта»?**

- А) нет; Б) да.

**Контрольная работа по теме «Моделирование и формализация» (11 класс). Вариант – 2.**

**1. Какие пары объектов находятся в отношении "объект - модель"?**

- А) компьютер – данные;
- Б) компьютер - его функциональная схема;
- В) компьютер – программа;
- Г) компьютер – алгоритм.

**2. Какая модель компьютера является формальной (полученной в результате формализации)?**

- А) техническое описание компьютера;
- Б) фотография компьютера;
- В) логическая схема компьютера;
- Г) рисунок компьютера.

**3. Информационной моделью, которая имеет табличную структуру является ...**

- А) файловая система компьютера;
- Б) таблица Менделеева;
- В) генеалогическое дерево семьи;
- Г) функциональная схема компьютера.

**4. Какая модель является динамической (описывающей изменение состояния объекта)?**

- А) формула химического соединения;
- Б) формула закона Ома;
- В) формула химической реакции;
- Г) закон Всемирного тяготения.

**5. Формальной информационной моделью является ...**

- А) анатомический муляж;
- Б) техническое описание компьютера;
- В) рисунок функциональной схемы компьютера;
- Г) программа на языке программирования.

**6. Компьютерный эксперимент может быть проведен, если информационная модель представлена в форме ...**

- А) программы на языке программирования;
- Б) изображения в растровом графическом редакторе;
- В) изображения в векторном графическом редакторе;
- Г) текста в текстовом редакторе.

**7. Вставьте пропущенное слово, выбрав его из предложенного ниже списка.**

**Информационная модель - это целенаправленно отобранная информация об объекте, которая отражает наиболее существенные для исследователя ... этого объекта.**

- А) информация;
- Б) законы функционирования;
- В) отличительные особенности;
- Г) свойства.

**8. Вставьте пропущенное слово, выбрав его из предложенного ниже списка.**

**Компьютерная модель - это ... модель, выполненная с помощью компьютерных технологий.**

- А) информационная; Б) схематичная; В) электронная.

**9. Могут ли у разных объектов быть одинаковыми модели?**

- А) нет;
- Б) да, но только для конструктивных (искусственных, созданных людьми) объектов;
- В) да.

**10. Построение любой модели начинается ...**

- А) с выделения свойств и признаков объекта-оригинала;



Б) с определения цели моделирования;

В) с выбора вида будущей модели?

**11. Вставьте в предложение наиболее точный термин из предложенного ниже списка.**

***Если материальная модель объекта - это его....., то информационная модель объекта - это его описание.***

А) физическое подобие;

Б) точное воспроизведение;

В) схематичное представление;

Г) преобразование.

**12. Какие из приведенных ниже моделей являются статическими? Выбрать три правильных ответа.**

А) карта местности;

Б) дружеский шарж;

В) программа, имитирующая движение стрелок циферблата на экране дисплея;

Г) план сочинения;

Д) график изменения температуры воздуха в течение дня.

**13. Какие из утверждений являются верными? Выбрать два правильных ответа.**

А) математическая формула является информационной моделью;

Б) график движения поезда - табличная статическая модель;

В) план дома - графическая детерминированная модель, описывающая структуру объекта;

Г) турнирная таблица чемпионата по футболу - эталонная динамическая модель.

## Контрольная работа по информатике и ИКТ в 11 классе (ИТП).

### 1. Базы данных — это:

1. информационные модели, позволяющие в упорядоченном виде хранить данные о группе объектов, обладающих одинаковым набором свойств;
2. программные средства, позволяющие организовывать информацию в виде таблиц;
3. программные средства, обрабатывающие табличные данные;
4. программные средства, осуществляющие поиск информации.

### 2. Запись – это:

1. Столбец в базе данных;
2. Поле базы данных.
3. Строка в базе данных.
4. Отдельное значение в базе данных.

### 3. Поле – это:

1. Столбец в базе данных;
2. Объект базы данных.
3. Строка в базе данных.
4. Отдельное значение в базе данных.

### 4. В коробке меньше 9, но больше 3 шаров. Сколько шаров может быть в коробке?

А) 3; В) 9; С) 2; D) 5; E) 10.

### 5. Какие атрибуты (признаки) объекта должны быть отражены в информационной модели, описывающей хобби ваших одноклассников, если эта модель позволяет получить ответы на следующие вопросы:

- Каков возраст всех детей, увлекающихся компьютером?
- Каковы имена девочек, увлекающихся пением?
- Каковы фамилии мальчиков, увлекающихся хоккеем?

А) имя, пол, хобби;

В) фамилия, пол, хоккей, пение, возраст;

С) имя, пол, хобби, возраст;

D) имя, возраст, хобби;

E) фамилия, имя, пол, возраст, хобби?

### 6. Реляционная база данных задана таблицей:

	Ф.И.О	Пол	Возраст	Клуб	Спорт
1	Панько Л.П.	жен	22	Спартак	футбол
2	Арбузов А.А.	муж	20	Динамо	лыжи
3	Жиганова П.Н.	жен	19	Ротор	футбол
4	Иванов О.Г.	муж	21	Звезда	лыжи
5	Седова О.Л.	жен	18	Спартак	биатлон
6	Багаева СИ.	жен	23	Звезда	лыжи

Какие записи будут выбраны по условию: Спорт= "лыжи" И Пол= "жен" ИЛИ Возраст<20?

А) 2, 3, 4, 5, 6; В) 3, 5, 6; С) 1, 3, 5, 6; D) 2, 3, 5, 6; E) таких записей нет.

### 7. Реляционная БД задана таблицей:

	Название	Категория	Кинотеатр	Начало сеанса
1	Буратино	х/ф	Рубин	14
2	Кортик	х/ф	Искра	12
3	Винни-Пух	м/ф	Экран	9
4	Дюймовочка	м/ф	Россия	10
5	Буратино	х/ф	Искра	14
6	Ну, погоди	м/ф	Экран	14
7	Два капитана	х/ф	Россия	16

Выбрать первичный ключ для таблицы (допуская, что в кинотеатре один зал):

- А) Название+Кинотеатр;
- В) Кинотеатр+Начало сеанса;
- С) Название+Начало сеанса;
- Д) Кинотеатр;
- Е) Начало сеанса.

**8. Структура реляционной базы данных изменяется при:**

- А) удалении любой записи;
- В) удалении любого поля;
- С) изменении любой записи;
- Д) добавлении записи;
- Е) удалении всех записей.

**9. Реляционная база данных задана таблицей. Записи в таблице пронумерованы.**

	Код дистанции	Код соревнований	Дата	Время спортсмена (с)
1	101	Д02	11.12.2004	56,6
2	104	Д01	12.10.2005	37
3	102	Д02	11.12.2005	56,1
4	103	Д05	11.12.2005	242,8
5	101	Д04	13.01.2005	181,1
6	102	Д01	12.10.2005	35,45

Сформулировать условие поиска, дающее сведения о спортсменах, принимавших участие в соревнованиях на дистанциях с кодами Д01 и Д03 не позднее 10.12.2004.

- А) Код\_дистанции="Д01" и Код\_дистанции="Д03" и Дата\_соревнования>10.12.2004
- В) (Код\_дистанции="Д01" или Код\_дистанции="Д03") и Дата\_соревнования>10.12.2004
- С) Код\_дистанции="Д01" и (Код\_дистанции="Д03" или Дата\_соревнования<=10.12.2004)
- Д) Код\_дистанции="Д01" и Код\_дистанции="Д03" и Дата\_соревнования<=10.12.2004
- Е) (Код\_дистанции="Д01" или Код\_дистанции="Д03") и Дата\_соревнования<=10.12.2004

**10. Дана однотабличная база данных «Автомобилисты»:**

	Владелец	Модель	Номер	Дата регистрации
1	Левченко Н.	Волга	И537ИГ-59	15.08.2001
2	Сидоров А.	Жигули	Ф131ФП-59	14.02.2000
3	Горохов И.	Форд	Б171БП-59	27.10.2000
4	Федоров К.	Волга	И138ИП-59	20.05.2001
5	Сидоров А.	Жигули	И321ИП-59	27.10.2000

Отсортировать таблицу в порядке возрастания по двум полям: Модель+Номер. А) 1; 4; 2; 5; 3; ; В) 3; 4; 5; 1; 2; С) 4; 1; 5; 2; 3 Д) 3; 5; 2; 4; 1; Е) 2; 1; 5; 4; 3.

**11. Основные типы полей:**

- 1. Дата, числовой, звуковой, логический;
- 2. Символьный, табличный, дата, логический;
- 3. Логический, числовой, дата, символьный;
- 4. Числовой, логический, ключевой, табличный.

12. В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных, сколько родных сестер есть у Лесных П.А

**Таблица 1**

ИД	Фамилия_И.О.	Пол
2011	Ковач Л.П.	Ж
2012	Данзас К.К.	М
2024	Павлова В.А.	Ж
2045	Лесных Л.А.	Ж
2056	Данзас Е.Ф.	Ж
2077	Ларина Т.Д.	Ж
2083	Данзас И.К.	М
2094	Данзас Е.К.	Ж
2115	Лесных А.П.	М
2140	Данзас Т.И.	Ж
2162	Данзас П.И.	М
2171	Гиппиус З.А.	Ж
2186	Молчалина С.А.	Ж
2201	Лесных П.А.	М

1) 1

2) 2

3) 3

4) 4

**Таблица 2**

ИД_Родителя	ИД_Ребенка
2094	2045
2115	2045
2011	2083
2012	2083
2011	2094
2012	2094
2056	2140
2083	2140
2056	2162
2083	2162
2094	2186
2115	2186
2094	2201
2115	2201

## Контрольная работа по теме «Алгоритмизация и программирование»

### Вариант № 1

A1. Что представляет алгоритм, который изображён на рис.1.

- 1) команда ветвления в полной форме, в которую вложена команда ветвления в сокращенной форме
- 2) две команды ветвления в полной форме, одна из которых вложена в другую
- 3) две команды ветвления в сокращенной форме, одна из которых вложена в другую
- 4) команда ветвления в сокращенной форме, в которую вложена команда ветвления в полной форме.

A2. Определите значение переменной S после выполнения фрагмента алгоритма изображённого на рис. 2 1) 9 2) 16 3) 25 4) 36

A3. Определите значение переменной a после выполнения следующего фрагмента программы. a:=4;b:=2\*a+8;

c:=b+a;a:=c\*a;

- 1) 20 2) 80 3) 64 4) 32

A4. Определите значение целочисленных переменных a и b после выполнения следующего фрагмента программы.

a:=1235;b:=(a div 100)+25;

a:=(a mod 100)-b;

- 1) a=-2, b=37 2) a=37, b=35 3) a=5, b=123 4) a=-12, b=35

A5. Найдите, какое числовое значение выведется на экран оператором

Writeln((5+2)\*2+(4+8)\*2-22/2:2:0); 1) 12 2) 16 3) 27 4) 23

A6. С помощью какого условного выражения можно определить попадает ли точка с координатами (x,y) внутрь

прямоугольника с координатами левого верхнего угла (2,10), правого нижнего угла (8, 3).

- 1) If (x>2) or (x<8) or (y<10) or (y>3) then write('попадает');
- 2) If (x>2) and (x<8) or (y<10) and (y>3) then write('попадает');
- 3) If (x>2) or (x<8) and (y<10) or (y>3) then write('попадает');
- 4) If (x>2) and (x<8) and (y<10) and (y>3) then write('попадает');

A7. Сколько раз выполнится оператор в теле цикла: For i:=3 to 20 do j:=i;

- 1) 19 2) 16 3) 17 4) 18

A8. Следующий фрагмент программы обрабатывает массив A размером  $n \times n$ .

for i:=1 to n do begin

c:=a[i,i]; a[i,i]:=a[n-i+1,n-i+1]; a[n-i+1,n-

амостоятельная работа в форме ЕГЭ для учащихся 11 класса

по теме «Алгоритмизация и программирование»

### Вариант № 2

A1. Что представляет алгоритм, который изображён на рис.1.

- 1) команда ветвления в полной форме, в которую вложена команда ветвления в сокращенной форме
- 2) две команды ветвления в полной форме, одна из которых вложена в другую
- 3) две команды ветвления в сокращенной форме, одна из которых вложена в другую
- 4) команда ветвления в сокращенной форме, в которую вложена команда ветвления в полной форме.

A2. Определите значение переменной S после выполнения фрагмента алгоритма изображённого на рис. 2 1) 29 2) 37 3) 46 4) 56

A3. Определите значение переменной a после выполнения следующего фрагмента программы. a:=-24;a:=a+a/8;

c:=12; b:=c-a; a:=-b;

1) 24 2) 15 3) □36 4) □39

A4. Определите значение целочисленных переменных c и b после выполнения следующего фрагмента программы.

a:=14;b:=a mod 3;

a:=39;c:=a div b;

1) c=2, b=19 2) c=39, b=2 3) c=19, b=2 4) c=42, b=19

A5. Найдите, какое числовое значение выведется на экран оператором Writeln(128/4/2/2/2+5:1:0);

1) 37 2) 13 3) 31 4) 9

A6. С помощью какого условного выражения можно определить, что точка с координатами (x,y) не попадает внутрь прямоугольника с координатами левого верхнего угла (2,10), правого нижнего угла (8, 3).

1) If (x<=2) or (x>=8) or (y>=10) or (y<=3) then write('не попадает');

2) If ((x<=2) and (x>=8)) or ((y>=10) and (y<=3)) then write('не попадает');

3) If ((x<=2) or (x>=8)) and ((y>=10) or (y<=3)) then write('не попадает');

4) If (x<=2) and (x>=8) and (y>=10) and (y<=3) then write('не попадает');

A7. Сколько раз выполнится оператор в теле цикла: For j:=10 downto 5 do i:=i+1;

1) 5 2) 6 3) 7 4) 4

A8. Следующий фрагмент программы обрабатывает массив A размером n□n.

k:=4; for i:=1 to n do begin

c:=a[1,i]; a[1,i]:=a[k,i];a[k,i]:=c;end;

В результате его выполнения меняются местами:

1) две строки в таблице 2) два столбца в таблице

3) элементы диагонали в таблице 4) элементы строки на элементы столбца

A9. В результате выполнения фрагмента программы на экран долж

Самостоятельная работа в форме ЕГЭ для учащихся 11 класса по теме «Алгоритмизация и программирование»

Вариант № 3

A1. Что представляет алгоритм, который изображён на рис1.

1) команда ветвления в полной форме, в которую вложена

команда ветвления в сокращенной форме

2) две команды ветвления в полной форме, одна из которых вложена в другую

3) две команды ветвления в сокращенной форме, одна из которых вложена в другую

4) команда ветвления в сокращенной форме, в которую вложена команда ветвления в полной форме.

A2. Определите значение переменной B после выполнения фрагмента алгоритма изображённого на рис. 2 1) 3 2) □4 3) 4 4) □3

A3. Определите значение переменной y после выполнения следующего фрагмента программы. X:=3;

x:=x\*3+2;y:=1-x\*x;

1) □120 2) □110 3) □8 4) □6

A4. Определите значение целочисленных переменных a и b после выполнения следующего фрагмента программы.

a:=543;b:=(a div 100)+15;

a:=(a mod 4)-b;

1) a=-44, b=58 2) a=20, b=20 3) a=□17, b=20 4) a=□17, b=5

A5. Найдите, какое числовое значение выведется на экран оператором Writeln(128/4/(2/2/2)+5.0:2:0);

1) 37 2) 13 3) 69 4) 9

A6. Какие из предложенных условных выражений будут ложными, если точка A с координатами (x,y) попадает внутрь прямоугольника с координатами левого нижнего угла (-10,5), правого верхнего угла (-5, 10).

- 1)  $(x < -5) \text{ or } (x > -10) \text{ or } (y > 10) \text{ or } (y < 5)$ ;
- 2)  $(x < -5) \text{ or } (x > -10) \text{ and } (y > 10) \text{ or } (y < 5)$ ;
- 3)  $(x > -5) \text{ and } (x < -10) \text{ or } (y < 5)$ ;
- 4)  $(x < 5) \text{ and } (y > 5)$ .

A7. Сколько раз выполняются операторы в теле цикла: For i:=1 to 6 do begin i:=i+1; SS:=SS+1.0; end;

1) 5 2) 6 3) 3 4) 4

A8. Следующий фрагмент программы обрабатывает массив A размером  $n \times n$ .

```
K:=2; for i:=1 to n do begin  
b:=a[i,i]; a[i,i]:=a[i,k]; a[i,k]:=b;end;
```

В результате его выполнения меняются местами:

- 1) элементы диагонали и k-ой строки таблицы
- 2) две строки в таблице
- 3) элементы диагонали и k-го столбца таблицы
- 4) два столбца в таблице

A9. В результате выполнения фрагмента программы на экран должно быть выведено значение n!. Выберите

правильный фрагмент.

- 1) Begin Read(n); For I:=1 to 10 do begin n:=n\*I; writeln(n); end; end.
- 2) Begin Read(n); P:=1; For I:=2 to n do p:=p\*I; writeln(p); end.
- 3) Begin Read(n); For I:=2 to n do p:=p\*I; writeln(p); end.
- 4) Begin Read(n); P:=1; For I:=1 to n do begin p:=p\*I; writeln(p); end; end.

B1. Найдите значение переменной P при выводе на экран.

```
Program Primer4;
```

```
Var P,i:integer;
```

```
begin
```

```
P:=1; For i:=1 to 5 do P:=P*i; Writeln(P);
```

```
end.
```

B2. Чему равно значение переменной S2 при выводе ее на экран

Самостоятельная работа в форме ЕГЭ для учащихся 11 класса по теме «Алгоритмизация и программирование»

Вариант № 4

A1. Что представляет алгоритм, который изображен на рис1.

- 1) команда ветвления в полной форме, в которую вложена команда ветвления в сокращенной форме
- 2) две команды ветвления в полной форме, одна из которых вложена в другую
- 3) две команды ветвления в сокращенной форме, одна из которых вложена в другую
- 4) команда ветвления в сокращенной форме, в которую вложена команда ветвления в полной форме.

A2. Определите значение переменной S после выполнения фрагмента алгоритма изображенного на рис. 2 1) 15 2) 11 3) 20 4) 16

A3. Определите значение переменной c после выполнения следующего фрагмента программы.  $a := \square 3$ ;  $a := \square a \square a$ ;

```
b:=a+2;c:= $\square$ b+a*b;
```

1) 40 2) 32 3)  $\square$ 32 4) 48

A4. Определите значение целочисленных переменных c и b после выполнения следующего фрагмента программы.

a:=23;b:=a mod 3;

a:=b+10;c:=a div b;

1) c=2, b=12 2) c=6, b=2 3) c=12, b=6 4) c=2, b=1

A5. Найдите, какое числовое значение выведется на экран оператором  
Writeln((24+9)/3+(9+3)/3:2:0);

1) 16 2) 15 3) 31 4) 37

A6. Какое из предложенных условных выражений будет истинным, если точка A с координатами (x,y) попадает  
внутри прямоугольника с координатами левого нижнего угла (-10,5), правого верхнего угла (-5, 10).

1) (x<-5) and (y>10) or (x>-5) and (y>-10);

2) (x>-5) or (x>-10) and (y>10) or (y<5);

3) (x>-5) and (y<10) or (y>-10);

4) (x<5) and (y>10).

A7. Сколько раз выполнится оператор в теле цикла: For i:=5 to 3 do j:=i;

1) 0 2) 1 3) 2 4) 3

A8. Следующий фрагмент программы осуществляет поиск в одномерном массиве A длины m.

```
x:=a[1]; for i:=1 to m do
```

```
if a[i]<=x then x:=a[i];
```

После выполнения данного фрагмента программы в переменной x будет храниться:

1) индекс элемента массива, имеющего максимальное значение 2) значение максимального элемента массива

3) индекс элемента массива, имеющего минимальное значение 4) значение минимального элемента массива

A9. В результате выполнения фрагмента программы на экран должно быть выведено значение x в степени n, x

и n целые числа. Выберите правильный фрагмент.

1) Begin Read(x); Read(n); For I:=1 to n do x:=x\*i; writeln(x); end.

2) Begin Read(x); Read(n); For I:=1 to n do x:=x\*n; writeln(x); end.

3) Begin Read(x); Read(n); P:=1; For I:=1 to n do x:=x\*p; writeln(x); end.

4) Begin Read(x); Read(n); P:=1; For I:=1 to n do p:=p\*x; writeln(p); end.

B1. Чему равно значение переменной S при выводе ее на экран?

```
Program Primer8;
```

```
Var S,i:integer;
```

```
begin
```

```
s:=0; For i:=1 to 5 do S:=S+2*i; Writeln(S:5);
```

```
end.
```

B2. Чему равно значение переменной S при выводе ее на экран?

```
Program Prim4;
```

```
Var S:real;
```

```
begin
```

```
S:=1;
```

```
while S<=10 do
```

```
S:=S+S; Writeln(S);
```

```
end.
```

Рис.1

Условие 1

оператор1

Да

Нет

Условие 2 Да Нет



```

оператор2
оператор1
переменной A, которое будет выведено на экран.
Var A, n:integer;
Begin
n:=50; A:=0;
while n >1 do begin
n:=n div 2; A:=A+1;
End;
writeln(A);
end.

```

В4. Значения двумерного массива A размером  $7 \times 7$  задаются с помощью следующего фрагмента

```

программы:
For i:=1 to 7 do
For j:=1 to 7 do A[i,j]:=i*j-j;

```

Сколько строк в массиве будут иметь нулевые значения?

С1. Требовалось написать программу, но программист торопился и сделал в программе ошибки.

Последовательно выполните задания:

- 1) Приведите пример входных данных, при которых программа неверно решает поставленную задачу.
- 2) Доработайте программу, чтобы не было случаев её неправильной работы.

Задача: Определить можно ли построить треугольник из отрезков с длинами x,y,z.

Программа:

```

Var x,y,z:real;
begin
readln(x,y,z);
if (x+y>z) then begin
if (x+z>y) then
if (y+z>x) then
writeln('треугольник построить можно');
end;
writeln('треугольник построить нельзя');
end.

```

С2. Дана последовательность целых чисел. Опишите на русском языке или на одном из языков

программирования алгоритм, который позволяет указать, сколько раз в последовательности встречается каждый элемент.

Критерии оценки. Часть А каждое задание 1 балл. Часть В каждое задание 1 балл. Часть С задание С1

максимум 2 балла, задание С2 максимум 3 балла. Максимальный балл 18.

От 9 до 11 баллов - оценка «3». От 12 до 14 баллов - оценка «4». От 15 до 18 баллов - оценка «5»

## Ответы.

Вопрос	Вариант 1	Вариант 2	Вариант 3	Вариант 4
A1.	3	4	1	2
A2.	2	1	3	4
A3.	2	4	1	1
A4.	1	3	3	2
A5.	3	4	3	2
A6.	4	3	3	3
A7.	4	2	2	1
A8.	4	1	3	4
A9.	1	3	2	4
B1.	86	37	120	30
B2.	34	17	9	16
B3.	14	66	31	5
B4.	62	5	6	1

## Итоговая контрольная работа

### 1 вариант

1. Какие виды памяти используются в ПК?
  - a) только постоянная память
  - b) только оперативная память
  - c) только внешняя память
  - d) все перечисленные ответы
2. Какие функции выполняет центральный процессор?
  - a) руководит всей работой ПК, осуществляя связь между всеми частями компьютера
  - b) осуществляет связь между персональным компьютером и периферийными устройствами
  - c) выполняет только низкоуровневые команды
  - d) используется для осуществления связи между электронными компонентами ПК
3. Скорость работы ПК в целом определяется:
  - a) скоростью вычислений процессора
  - b) скоростью обмена данными с периферийными устройствами и оперативной памятью
  - c) только разрядностью системной шины
  - d) скоростью процессора и обмена данными с периферийными устройствами и оперативной памятью
4. Что из перечисленного не является внешней памятью?
  - a) накопитель на жестком магнитном диске
  - b) BIOS
  - c) накопитель на лазерном диске
  - d) накопитель на гибком магнитном диске
5. С помощью сканера можно:
  - a) выводить в цифровом виде фотографии, рисунки и тексты
  - b) вводить в ПК фотографии, тексты и рисунки
  - c) вводить только фотографии
  - d) печатать текстовую информацию
6. Какое из перечисленных устройств не является устройством вывода?
  - a) монитор
  - b) принтер
  - c) сканер
  - d) проектор
7. Что является характеристикой монитора?
  - a) тактовая частота
  - b) цветное разрешение
  - c) дискретность
  - d) время доступа к информации
8. Операционные системы представляют собой программные продукты, входящие в состав:
  - a) прикладного программного обеспечения
  - b) системного программного обеспечения
  - c) системы управления базами данных
  - d) систем программирования
  - e) уникального программного обеспечения
9. В некотором каталоге хранится файл Задачи\_по\_программированию.txt. В этом каталоге создали подкаталог и переместили в него файл Задачи\_по\_программированию.txt. После этого полное имя файла стало D:\INFORM\LESSONS\10\_CLASS\Задачи\_по\_программированию.txt. Каково полное имя каталога, в котором хранился файл до перемещения?

- a) D:\INFORM
- b) D:\INFORM\LESSONS
- c) 10\_CLASS
- d) LESSONS\10\_CLASS

**10.** Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «\*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «\*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: A?ce\*s.m\*

- a) Acess.md
- b) Accesst.dbf
- c) Access.mdb
- d) Akcces.m1

**11.** В ячейке C2 записана формула =\$E\$3+D2. Какой вид приобретет формула, после того как ячейку C2 скопируют в ячейку B1?

- a) =\$E\$3+C1
- b) =\$D\$3+D2
- c) =\$E\$3+E3
- d) =\$F\$4+D2

**12.** Выполните действия:

- 1)  $1375_8 + 1D4_{16}$
- 2)  $10111001_2 + 111001001_2$

**13.** Сообщение, записанное буквами из 32-символьного алфавит, содержит 30 символов. Какой объём информации оно несёт?

**14.** Сколько бит в 1 байте, килобайте, мегабайте?

**15.** Назовите известные вам браузеры (минимум четыре).

2 вариант

**1.** Драйверы – это:

- a) техническое устройство
- b) носители информации
- c) программы для согласования работы внешних устройств и компьютера
- d) программы для ознакомления пользователя с принципами устройства компьютера

**2.** Информация внутри ПК хранится в виде:

- a) файлов, находящихся в папках
- b) только в текстовом виде
- c) в виде цифр
- d) рисунков, текстов, звуков и видео

**3.** Ярлык – это:

- a) значок со стрелкой, позволяющий изменять свойства объектов
- b) значок со стрелкой, осуществляющий переход к ресурсам ПК и служащий для запуска программ
- c) ссылка на удаленный ресурс
- d) стандартный элемент ОС Windows

**4.** Заражение компьютерными вирусами может произойти в процессе:

- a) работы с файлами
- b) выключения компьютера
- c) форматирования дискеты
- d) печати на принтере

**5.** Какая программа не является антивирусной?

- a) Avast
- b) Norton Antivirus
- c) Corel Draw
- d) DrWeb

**6.** Основными характеристиками процессора являются:

- a) емкость ОЗУ, тактовая частота, разрядность

- b) разрядность, тактовая частота, адресное пространство  
 c) адресное пространство, разрядность, BIOS  
 d) BIOS, емкость ОЗУ, тактовая частота
- 7. Компьютерные вирусы:**  
 a) возникают в связи со сбоями в аппаратных средствах компьютера  
 b) создаются людьми специально для нанесения ущерба пользователям ПК  
 c) зарождаются при работе неверно написанных программных продуктов  
 d) являются следствием ошибок в операционной системе  
 e) имеют биологическое происхождение
- 8. Программой – архиватором называют:**  
 a) программу резервного копирования файлов  
 b) программу для уплотнения информационного объема (сжатия) файлов  
 c) интерпретатор  
 d) транслятор  
 e) систему управления базами данных
- 9. В некотором каталоге хранится файл Список\_литературы.txt. В этом каталоге создали подкаталог с именем 10\_CLASS и переместили в него файл Список\_литературы.txt. После чего полное имя файла стало D:\SCHOOL\PHYSICS\10\_CLASS\Список\_литературы.txt. Каково полное имя каталога, в котором хранился файл до перемещения?**  
 a) D:\SCHOOL\PHYSICS\10\_CLASS  
 b) D:\SCHOOL\PHYSICS  
 c) D:\SCHOOL  
 d) SCHOOL
- 10. Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «\*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «\*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов не удовлетворяет маске: ??pri\*.\***  
 a) saprika.wow    b) weprik.cpp    c) otopri.c    d) reprint.be
- 11. В ячейке B1 записана формула =2\*\$A1. Какой вид приобретет формула, после того как ячейку B1 скопируют в ячейку C2?**  
 a) =2\*\$B1    b) =2\*\$A2    c) =3\*\$A2    d) =3\*\$B2H
- 12. Выполните действия:**  
 1)  $1212_8 + 3F1_{16}$   
 2)  $10011101_2 + 100001111_2$
- 13. Сообщение, записанное буквами из 256-символьного алфавит, содержит 15 символов. Какой объём информации оно несёт?**
- 14. Сколько бит в 1 байте, килобайте, мегабайте?**
- 15. Назовите известные вам поисковые серверы (минимум четыре).**

Ответы:

I вариант	II вариант
1) D	1) B
2) A	2) B
3) A	3) C
4) D	4) A
5) A	5) B
6) C	6) A
7) A	7) B
8) B	8) A

9) B 10) A 11) A 12) $7-15 = -8$ 13) A: $701+692=1393$ Б: 110010010 14) $5*50=250$	9) D 10) A 11) C 12) -3 13) A: $738+1441=2179$ Б: $100111100_2=316_{10}$ 14) $7*25=175$
1-11 – 1 балл 12-14 – 2 балла 15,16 – 1 балл	Всего 21 балл 18-21 – 5 13-17 – 4 9-12 – 3

# **Практические работы**

## Глава 1. Информация и информационные процессы

### Практические работы

#### Практическая работа № 1.


##### Оформление документа

Скопируйте в свой каталог документ *ЛинусТорвальдс.doc* и оформите его следующим образом:

1. Левое поле страницы – 2 см, правое поле – 1,5 см.
2. Заголовок: шрифт *Arial*, 12 пт, жирный, выравнивание по центру .
3. Эпиграф: шрифт *Times New Roman*, 12 пт, курсив, левая граница абзаца (левый отступ) 10 см.
4. Остальной текст: шрифт *Times New Roman*, 12 пт, выравнивание по ширине, абзацный отступ (отступ первой строки) 1 см, интервалы до и после абзацев – 0.
5. Удалите пустые абзацы перед первым абзацем основного текста, для первого абзаца добавьте интервал сверху 12 пт.
6. Замените кавычки " " на «».
7. Выделите названия фирм и операционных систем курсивом. Для этого создайте новый стиль *Имя*.
8. Установите режим обтекания рисунка и разместите его так, как на образце.
9. Правильно расставьте пробелы около знаков препинания (перед знаком препинания пробел не ставится, после знака – ставится).
10. Для последнего абзаца установите выравнивание вправо. Адрес сайта сделайте гиперссылкой, проверьте её работу.

**Линус Торвальдс**

*«Я делаю (бесплатную) операционную систему (это всего лишь хобби, и она не будет большой и профессиональной как gnu) для клонов 386(486) AT».*  
Линус



Приступая к разработке своей операционной системы (ОС), Линус Торвальдс не помышлял о какой-либо конкуренции с фирмами типа *Microsoft* — это было просто хобби студента Хельсинского университета. Затем это хобби незаметно превратилось в самую многообещающую ОС, причем на это превращение потребовалось восемь лет.

С детства Линус привык делать все собственными руками. Однажды мальчик не стал дожидаться отца, купившего ему сложную модель корабля, чтобы вместе с сыном склеить ее. Линус сделал все сам в первую же ночь. А позже это проявилось, как говорили университетские преподаватели, в «сумасшествии» — 20-летний студент решил самостоятельно создать самую сложную из компьютерных программ — операционную систему.

История ОС *Linux*, в принципе, началась еще до поступления Линуса в Хельсинский университет. Профессор Амстердамского университета Эндрю Танненбаум написал ОС *Minitx*, усовершенствованную версию операционной системы *UNIX*, которая могла работать на персональном компьютере минимальной конфигурации.

Торвальдс решил заняться переработкой *Minitx*, устав от бесплодных попыток получить машинное время на принадлежащей университету машине *Micro VAX* корпорации *Digital Equipment*. Однако с *Minitx*, незаменимой как средство обучения, было все-таки невозможно работать как с полнофункциональной ОС.

Так или иначе, в итоге возникло ядро, содержащее все основные компоненты *UNIX* — переклечение задач, файловую систему и драйверы устройств. Другими словами, на свет появилась операционная система *Linux Version 0.02*. Это произошло весной 1991 года. Система, разработанная Линусом, получила название, образованное от имени создателя и операционной системы *UNIX*.

Многие крупные производители программного обеспечения, такие как *Oracle* и *Netscape*, объявили о поддержке данной ОС; растет и число коммерческих предложений для этой платформы. Целый ряд компаний, например *Red Hat Software* и *Caldera Systems*, продают собственные дистрибутивы *Linux* с обязательствами технической поддержки.

По материалам сайта <http://chemvkh.net>



## Практическая работа № 2.

### Таблицы и списки<sup>1</sup>

1. Представьте информацию в структурном виде так, чтобы ее было легче воспринимать:  
*«Озеро Байкал находится на юг Восточной Сибири. Длина озера 636 км, ширина 80 км, площадь 31 722 км<sup>2</sup>, объём 23 615,390 км<sup>3</sup>. Длина береговой линии Байкала примерно 2100 км. Байкал – самое глубокое озеро на Земле, его глубина 1642 м. В Байкал впадает 336 рек, в том числе Селенга, Верхняя Ангара, Баргузин. Вытекает из Байкала единственная река – Ангара».*
2. Представьте информацию в структурном виде так, чтобы ее было легче воспринимать:  
*«Василий Пупкин летел из Перми в Ростов-на-Дону через Москву. Самолет из Перми в Москву летит 2 часа 15 мин, билет стоит 4100 рублей. Полет из Москвы в Ростов-на-Дону стоит 6500 рублей и длится 1 час 50 мин. Из Ростова-на-Дону Вася на пригородном такси за 100 рублей приехал в порт Азов (время поездки – 1 час 10 минут), а оттуда на катере – в село Круглое (время поездки – 1,5 часа, стоимость 500 рублей)».*
3. Представьте информацию в структурном виде так, чтобы ее было легче воспринимать:  
*«В поход решили взять палатку, котелок, топор, хлеб, тушенку, гречу, макароны, чай».*
4. Представьте информацию в структурном виде так, чтобы ее было легче воспринимать:  
*«Разработка программы обычно выполняется в несколько этапов. Сначала необходимо четко поставить задачу, сформулировать техническое задание – требования к программе. После этого разрабатывается алгоритм решения задачи. Следующий этап - кодирование, то есть написание программы на каком-то языке программирования. Затем выполняется отладка, а когда программа отлажена – тестирование, то есть проверка ее работы в разных режимах. Завершающие этапы – разработка документации и сопровождение программы после сдачи заказчику (обучение пользователей, ответы на вопросы, исправление найденных ошибок)».*
5. Представьте информацию в структурном виде так, чтобы ее было легче воспринимать:  
*«В подсемействе Енотовые биологи выделяют 4 рода: еноты (енот-ракоед, тремарьясский енот, барбадосский енот, енот-полоскун, багаманский енот, гваделупский енот, козумельский енот); коати или носатые еноты (белоносый коати, южноамериканский коати); горные коати и кольцехвостые еноты (какомицли, центральноамериканский какомицли)».*

---

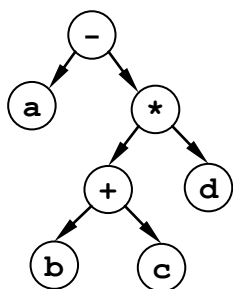
<sup>1</sup> В этой и следующих работах представлено довольно много заданий, из которых учитель может выбрать набор по вкусу.

### Практическая работа № 3.

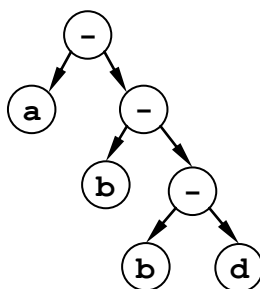
#### Деревья

- Представьте эту информацию в виде соответствующей структуры:  
«В каталоге все ссылки делятся на 4 раздела: *Образование, Программное обеспечение, Интернет* и *Остальное*. В разделе *Образование* есть подразделы *Школы, Вузы, Детские сады* и *Курсы*. Раздел *Программное обеспечение* включает подразделы *Операционные системы, Трансляторы, Языки программирования, Базы данных*. В разделе *Интернет* есть подразделы *Создание сайтов* и *Социальные сети*».
- Представьте эту информацию в виде соответствующей структуры:  
«Фирма *Рога и Копыта* продает компьютерную технику: принтеры (фирм *HP, Epson, Canon, Brother*), сканеры (фирм *Epson, Canon* и *Mustek*) и мониторы (фирм *Sony, Samsung, Philips, Acer*)».
- Представьте эту информацию в виде структуры вида «дерево»:  
«В каталоге *Фото* выделены отдельные подкаталоги для каждого года с 2008 по 2010. В каталоге 2008 есть вложенные каталоги *Ладога, Байкал* и *Волга*. Каталог 2009 содержит подкаталоги *Турция, Испания* и *Египет*, а каталог 2010 – подкаталоги *Москва* и *Санкт-Петербург*. В каталоге *Москва* есть подкаталоги *январь* и *июнь*».
- Определите выражения, соответствующие каждому из деревьев, в «нормальном» виде со скобками (эту форму называют *инфиксной* – операция записывается между данными). Постройте для каждого из них постфиксную форму.

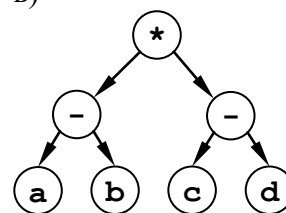
а)



б)



в)



Инфиксная форма:

Постфиксная форма:

а)	а)
б)	б)
в)	в)

- Постройте деревья, соответствующие следующим арифметическим выражениям:

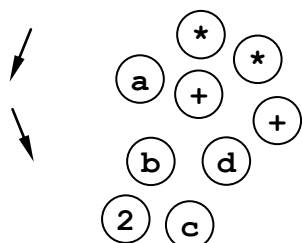
а)  $(a+b)*(c+2*d)$

в)  $(a+b+2*c)*d$

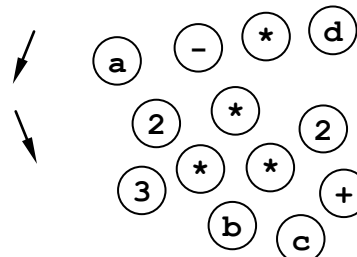
б)  $(2*a-3*d)*c+2*b$

г)  $3*a-(2*b+c)*d$

а)

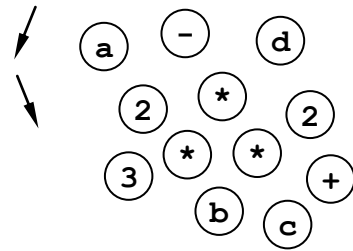
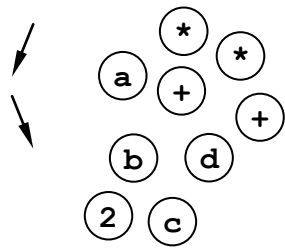


б)



в)

г)



Запишите эти выражения в префиксной и постфиксной формах.

*Префиксная форма:*

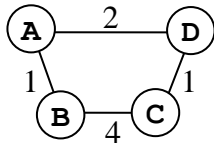
*Постфиксная форма:*

a)	a)
б)	б)
в)	в)
г)	а)

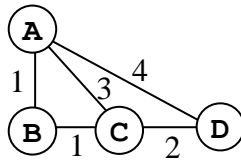
## Практическая работа № 4. Графы

1. Постройте матрицы смежности и весовые матрицы для каждого графа:

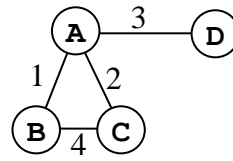
а)



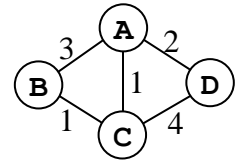
б)



в)



г)



Матрицы смежности:

а)

	A	B	C	D
A				
B				
C				
D				

б)

	A	B	C	D
A				
B				
C				
D				

в)

	A	B	C	D
A				
B				
C				
D				

г)

	A	B	C	D
A				
B				
C				
D				

Весовые матрицы

а)

	A	B	C	D
A				
B				
C				
D				

б)

	A	B	C	D
A				
B				
C				
D				

в)

	A	B	C	D
A				
B				
C				
D				

г)

	A	B	C	D
A				
B				
C				
D				

2. Постройте графы, соответствующие каждой из матриц смежности:

а)

	A	B	C	D	E
A		0	1	1	0
B	0		1	0	1
C	1	1		0	1
D	1	0	0		0
E	0	1	1	0	

б)

	A	B	C	D	E
A		0	1	1	1
B	0		1	0	0
C	1	1		0	1
D	1	0	0		0
E	1	0	1	0	

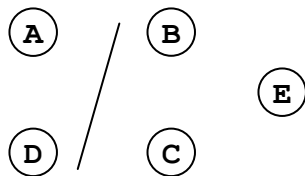
в)

	A	B	C	D	E
A		0	1	1	1
B	0		1	0	1
C	1	1		0	1
D	1	0	0		0
E	1	1	1	0	

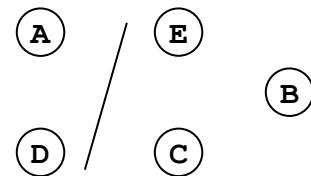
г)

	A	B	C	D	E
A		0	0	1	0
B	0		1	0	1
C	0	1		1	1
D	1	0	1		0
E	0	1	1	0	

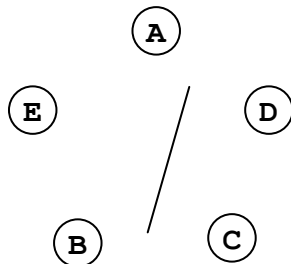
а)



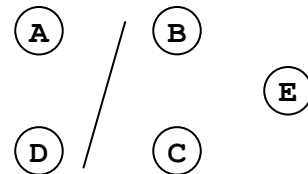
б)



в)



г)



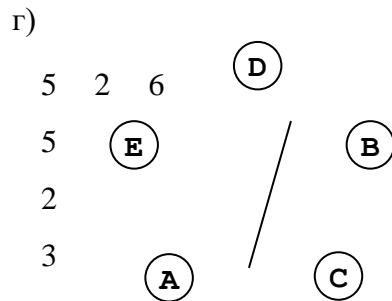
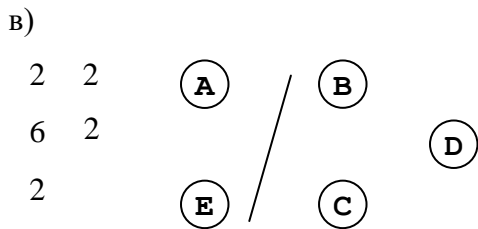
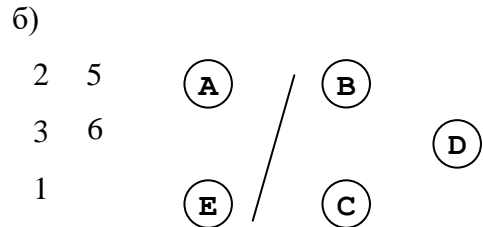
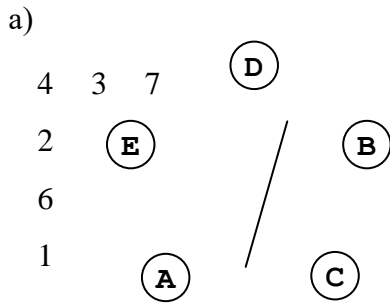
3. Постройте графы, соответствующие каждой из весовых матриц:

	A	B	C	D	E
A		4	3		7
B	4			2	
C	3			6	
D		2	6		1
E	7			1	

	A	B	C	D	E
A		2	5		6
B	2			3	
C	5				
D		3			1
E	6			1	

	A	B	C	D	E
A			2	2	6
B				2	
C	2			2	
D	2	2	2		
E	6				

	A	B	C	D	E
A		5	2		6
B	5			5	
C	2			2	
D		5	2		3
E	6			3	



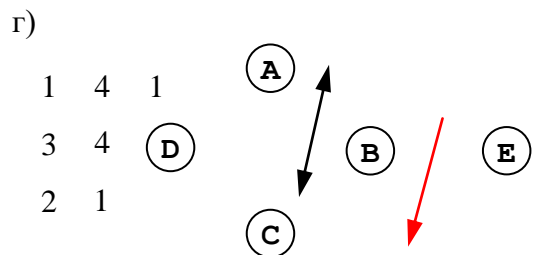
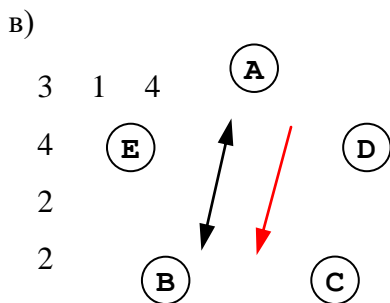
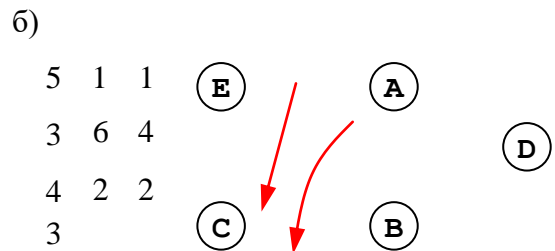
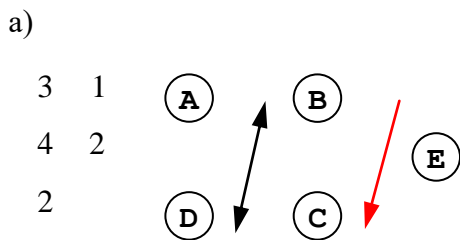
4. Постройте орграф, соответствующий каждой из весовых матриц.

	A	B	C	D	E
A			3	1	
B	2		4		2
C	3				
D	1				
E		2			

	A	B	C	D	E
A			5	1	1
B			6	4	
C	3	4			2
D		2			
E			3		

	A	B	C	D	E
A			3	1	4
B			4		2
C		4			2
D					
E	4		2		

	A	B	C	D	E
A				1	
B			4		1
C	3	4		4	2
D	1	2	4		
E	1	1	2		



## Глава 2. Кодирование информации

### Практические работы

#### Практическая работа № 5. Декодирование

1. Для кодирования сообщения используется таблица

Вариант 1:

А	Б	В	Г	Д
10	11	001	010	01

Сообщение: 0101110010110 (Ответы: ГБАДДА, ДДБВДА)

Вариант 2:

А	Б	В	Г	Д
01	11	110	010	101

Сообщение: 01011100101101 (Ответы: ААВААД, ААВГБА)

Вариант 3:

А	Б	В	Г	Д
000	001	00	01	100

Сообщение: 0010001001001 (Ответы: БВГББ, ВДГББ)

Вариант 4:

А	Б	В	Г	Д
000	010	011	01	11

Сообщение: 0100001101000010 (Ответы: БАДГАБ, ГАВГАБ)

Вариант 5:

А	Б	В	Г	Д
10	100	011	000	11

Сообщение: 1010000011011000 (Ответы: ААГВВГ, АБГДВГ)

Используя средства текстового процессора, изобразите двоичное дерево, соответствующее этому коду.

2. Выполняется ли для этой кодовой таблицы условие Фано? Обратное условие Фано? Почему?

Ответ:

3. Найдите все способы декодирования сообщения, записанное под таблицей:

Ответ:

Проверьте свой ответ с помощью программы *decode*.

4. Замените код одного символа так, чтобы выполнялось условие Фано (или обратное условие Фано). Выделите зеленым фоном ячейку таблицы с измененным кодом символа.

А	Б	В	Г	Д

5. Сократите код одного символа в таблице, полученной в п. 4 так, чтобы условие Фано (или обратное условие Фано) по-прежнему выполнялось. Выделите фиолетовым фоном ячейку таблицы с измененным кодом символа.

А	Б	В	Г	Д

## Практическая работа № 6. Необычные системы счисления

1. Найдите в Интернете информацию о факториальной системе счисления. Для этого можно использовать веб-страницы

<http://inf.1september.ru/article.php?ID=200801905>

<http://festival.1september.ru/articles/550322/>

<http://school.dtv.su/neobyichnyie-sistemyi-schisleniya/>

2. Переведите в десятичную систему числа, записанные в факториальной системе

Факториальная система	Десятичная система
2121 <sub>ф</sub>	
2201 <sub>ф</sub>	
2220 <sub>ф</sub>	
3020 <sub>ф</sub>	
3211 <sub>ф</sub>	

3. Переведите числа из десятичной системы счисления в факториальную :

Десятичная система	Факториальная система
91	
67	
84	
50	
69	

4. Найдите в Интернете информацию о фибоначчевой системе счисления. Для этого можно использовать веб-страницы

<http://inf.1september.ru/article.php?ID=200801905>

<http://pi.314159.ru/butler1.htm>

[http://ru.wikipedia.org/wiki/Фибоначчьева\\_система\\_счисления](http://ru.wikipedia.org/wiki/Фибоначчьева_система_счисления)

[http://www.tgl.net.ru:8080/wiki/index.php/Фибоначчьева\\_система\\_счисления](http://www.tgl.net.ru:8080/wiki/index.php/Фибоначчьева_система_счисления)

[http://www.e-maxx.ru/algo/fibonacci\\_numbers](http://www.e-maxx.ru/algo/fibonacci_numbers)

<http://festival.1september.ru/articles/550322/>

5. Переведите в десятичную систему числа, записанные в фибоначчевой системе

Фибоначчьева система	Десятичная система
10100 <sub>фиб</sub>	
10011 <sub>фиб</sub>	
10101 <sub>фиб</sub>	
100010 <sub>фиб</sub>	
11010 <sub>фиб</sub>	

6. Найдите все способы перевода следующих чисел из десятичной системы счисления в фибоначчьевую:

Десятичная система	Фибоначчьева система
14	
26	
40	

## Глава 3. Логические основы компьютеров

### Практические работы

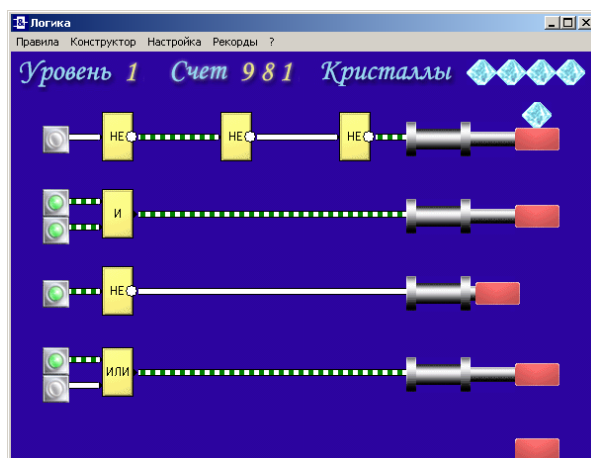
#### Практическая работа № 7.

#### Тренажер «Логика»

Тренажер «Логика» – это бесплатная программа, которую можно скачать с сайта автора:

<http://kpolyakov.spb.ru/prog/logic.htm>

6. Запустите тренажер «Логика». На уровне 1 познакомьтесь с действием логических элементов «И», «ИЛИ» и «НЕ».
7. **Правила игры.** Используя выключатели (кнопки) в левой части окна нужно сбрасывать кристалл строго на 1 «этаж» вниз (на следующую площадку). Когда кристалл опустится на самую нижнюю площадку, происходит переход на следующий уровень (всего 10 уровней). Если «уронить» кристалл (через 2 этажа или более), он разобьется, и этот уровень придется начать заново. Когда разбиты все 5 кристаллов, игра заканчивается.
8. Критерии выставления отметок:
  - «3» – выйти на уровень 6
  - «4» – выйти на уровень 9
  - «5» – пройти все 10 уровней





## Практическая работа № 8.

### Исследование запросов для поисковых систем

1. Используя поисковые системы *Яндекс* (<http://www.yandex.ru>) и *Google* (<http://www.google.ru>), определите количество веб-страниц, хранящихся в базе данных по следующим запросам:

Проверьте, насколько отклоняются результаты от теоретической формулы

$$N_{A/B} = N_A + N_B - N_{A \& B}$$

Вычислите отклонение от формулы:  $\Delta = N_{A/B} - (N_A + N_B - N_{A \& B})$

Запрос	Выдача <i>Яндекса</i> , млн страниц	Выдача <i>Google</i> , млн страниц
яблоки		
груши		
яблоки & груши		
яблоки   груши		
Отклонение от формулы $\Delta$		

2. Повторите исследование, взяв слова в запросе в кавычки (требуется только точная форма, без учета других падежей):

Запрос	Выдача <i>Яндекса</i> , млн страниц	Выдача <i>Google</i> , млн страниц
"яблоки"		
"груши"		
"яблоки" & "груши"		
"яблоки"   "груши"		
Отклонение от формулы $\Delta$		

3. Используя возможности расширенного поиска, ограничьте область действия предыдущих запросов доменом <http://narod.ru>:

Запрос	Выдача <i>Яндекса</i> , тыс. страниц	Выдача <i>Google</i> , тыс. страниц
"яблоки"		
"груши"		
"яблоки" & "груши"		
"яблоки"   "груши"		
Отклонение от формулы $\Delta$		

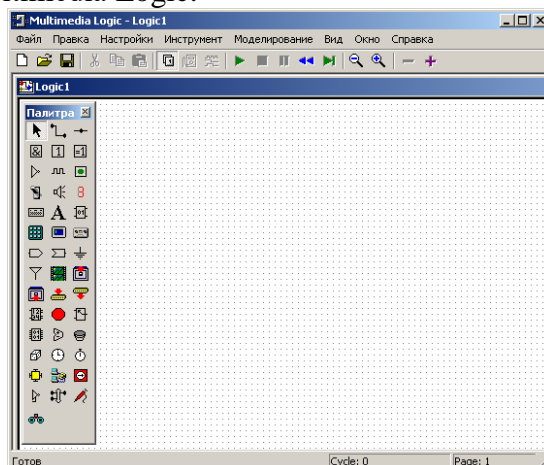
4. Сделайте выводы по результатам исследования двух поисковых систем.

--

## Практическая работа № 9.

### Логические элементы компьютера

Запустите программу Multimedia Logic.



В палитре, которая находится в левой части окна, мы будем использовать следующие инструменты:

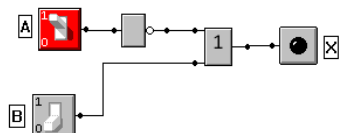
	перемещение		соединение		новый узел
	элемент НЕ		элемент И		элемент ИЛИ
	лампочка (светодиод)		выключатель (0/1)		текст

Названия других инструментов появляются при наведении мыши на кнопки этих инструментов в палитре, а информацию о них можно найти в справочной системе (F1).

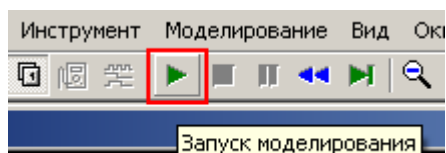
Чтобы добавить на схему какой-то элемент, нужно щёлчком левой кнопки мыши включить нужный инструмент и щёлкнуть мышью на поле там, где будет левый нижний угол элемента. Если элемент сначала был размещён не там, где хочется, его всегда можно подвинуть: нужно включить инструмент и перетащить картинку мышкой.

**Уровень А.** Соберём схему, которая соответствует логическому выражению  $X = \bar{A} + B$ .

- Из формулы видно, что нам будут нужны два входа-выключателя (А и В), один выход-лампочка (Х) и два логических элемента (НЕ и ИЛИ). Вот схема, которая должна получиться в результате этой части работы:



- Включим в палитре инструмент и двумя щелчками мыши установим на поле два выключателя. Затем включим инструмент и поставим их туда, куда нужно.
- С помощью инструмента добавим две надписи. Вызовем для каждой из них контекстное меню, щёлкнув правой кнопкой мыши, выберем пункт *Свойства* и изменим тексты на А и В соответственно.
- Добавим на поле один элемент НЕ, один элемент ИЛИ и лампочку. Расположим их так, как удобно для составления схемы. Около лампочки добавьте метку Х.
- Теперь нужно соединить элементы так, как показано на схеме выше. Включаем инструмент и рисуем линии соединения. Чтобы нарисовать такую линию, нужно нажать левую кнопку мыши на узле-источнике сигнала и протащить мышью (при нажатой левой кнопке) до узла, который получает сигнал. Если линия получилась не так, как нужно, её можно удалить. Для этого включите инструмент , выделите любой из узлов этой линии и нажмите клавишу *Delete*.
- Теперь щёлкаем по кнопке «Запуск моделирования» и включаем таким образом режим моделирования, в котором можно изучать работу собранной схемы:



7. Заполните таблицу истинности для собранной схемы:

A	B	X
0	0	
0	1	
1	0	
1	1	

8. Определите по таблице истинности, как называется операция, которую выполняет эта схема.

Ответ:

### Уровень В.

9. Соберите схему, которая соответствует логическому выражению  $X = \bar{A} \cdot \bar{B} + A \cdot B$ .  
Вставьте скриншот этой схемы в отчёт:

Скриншот:

10. Заполните таблицу истинности для собранной схемы:

A	B	X
0	0	
0	1	
1	0	
1	1	

11. Определите по таблице истинности, как называется операция, которую выполняет эта схема.

Ответ:

### Уровень С.

12. Найдите в сети схему, которая выполняет суммирование двух битов (без учёта переноса) с помощью **четырёх** логических элементов, соберите и испытайте её. Убедитесь, что она работает правильно. Вставьте скриншот этой схемы в отчёт:

Скриншот:

### Уровень D.

13. Постройте двухразрядный сумматор – схему, которая складывает двухбитные числа – и убедитесь, что он работает правильно. Вставьте скриншот этой схемы в отчёт:

Скриншот:

## Глава 4. Компьютерная арифметика

### Практические работы

Для выполнения этих работ используется учебный компьютер «ЛамПанель», который можно загрузить со страницы <http://kpolyakov.spb.ru/prog/lamp.htm>.

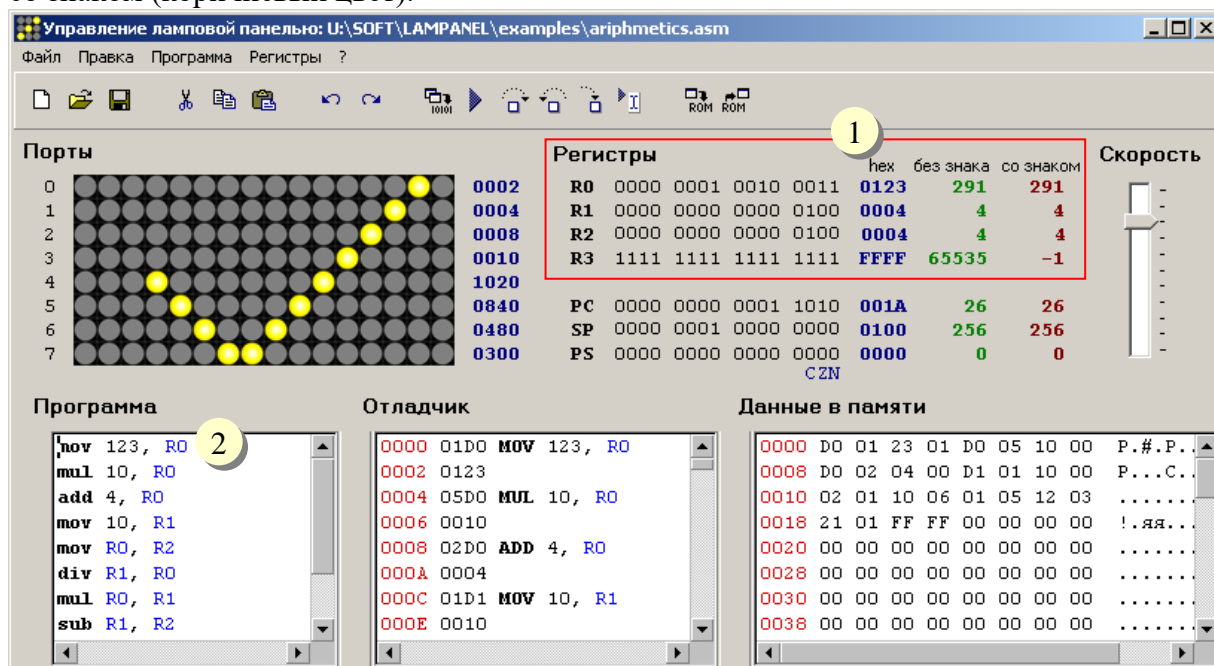
#### Практическая работа № 10.

#### Представление целых чисел

Знакомство с программой «ЛамПанель»

Программа «ЛамПанель» – это модель процессора, который управляет ламповой панелью, то есть, может с помощью специальных команд зажигать и гасить определенные лампочки.

Процессор обрабатывает данные, используя специальные сверхбыстродействующие ячейки собственной памяти – *регистры*. В этой работе мы будем использовать только четыре 16-битных *регистра общего назначения*, которые называются **R0**, **R1**, **R2** и **R3**. В области 1 на рисунке вы видите двоичные значения этих регистров (показаны черным цветом), шестнадцатеричные (синий цвет) и десятичные, без учета знака (зеленый цвет) и со знаком (коричневый цвет).



Область 2 – это текстовый редактор, в котором набирается программа для процессора на специальном языке, который называется *языком ассемблера*. Для того, чтобы программа выполнялась, нужно нажать клавишу F9 (выполнение без остановок) или F8 (выполнение по шагам). Чтобы программа остановилась, процессор должен выполнить команду **STOP**. Таким образом, простейшая программа состоит из одной команды **STOP**.

Для того, чтобы записать число в регистр, используют команду **MOV** (от англ. *move* – переместить). Числа записываются в шестнадцатеричной системе счисления. Например, команда

**MOV 12, R0**

запишет число  $12_{16} = 18$  в регистр **R0**. Каждая команда записывается в отдельной строке. Поэтому полная программа будет выглядеть так:

**MOV 12, R0**

**STOP**

Для того, чтобы добавить число к регистру, применяют команду **ADD** (от англ. *add* – сложить). Например, команда

**ADD 15, R0**

добавляет число  $15_{16} = 23$  к регистру **R0**. Есть и аналогичная команда вычитания – **SUB** (от англ. *subtract* – вычесть). Нам будет нужна еще одна команда:

**NOT R0**

которая выполняет *инверсию* всех битов регистра, то есть меняет все нули на единицы, а единицы – на нули.

*Задание на практическую работу*

**Уровень Е.**

11. Запустите тренажер «Лампанель».

12. Используя команду **MOV**, напишите программу, которая заполнит регистры так, как на рисунке:

Регистры				
<b>R0</b>	1111	0000	0000	0000
<b>R1</b>	1111	1111	0000	0000
<b>R2</b>	1111	1111	1111	0000
<b>R3</b>	1111	1111	1111	1111

Не забудьте закончить программу командой **STOP**. Выполните программу.

Программа:

13. Запишите, какие десятичные числа были только что записаны в регистры:

Регистр	Десятичные значения	
	без учета знака	с учетом знака
<b>R0</b>		
<b>R1</b>		
<b>R2</b>		
<b>R3</b>		

14. Выполните программу

```
sub 1, R0
not R0
stop
```

при различных начальных значениях регистра **R0** и запишите десятичные значения, полученные в **R0** после выполнения программы (не забудьте перевести числа в шестнадцатеричную систему):

Начальное значение	Значение R0 после выполнения программы	
	без учета знака	с учетом знака
<b>5</b>		
<b>10</b>		
<b>25</b>		

Какую операцию выполняет этот алгоритм? Найдите описание этого алгоритма в учебнике.

Ответ:

#### Уровень F.

15. При тех же начальных значениях регистра **R0** выполните программу

```
not R0
add 1, R0
stop
```

и заполните таблицу (не забудьте перевести числа в шестнадцатеричную систему):

Начальное значение	Значение R0 после выполнения программы	
	без учета знака	с учетом знака
<b>5</b>		
<b>10</b>		
<b>25</b>		

Сделайте выводы, сравнивая результаты пп. 4 и 5.

Выводы:

#### Уровень G.

16. Используя программу «ЛамПанель», вычислите арифметические выражения и

запишите результаты в таблицу (не забудьте перевести числа в шестнадцатеричную систему):

Выражение	Результат	
	без учета знака	с учетом знака
<b>65530 + 9</b>		
<b>32760 + 9</b>		
<b>8 - 10</b>		

Подсказка:  $65535 = \text{FFFF}_{16}$ ,  $32767 = 7\text{FFF}_{16}$

Объясните, почему при сложении в первой строке получилось именно такое число.

Как его можно рассчитать без программы?

Объясните, почему при сложении во второй строке в варианте «с учётом знака» получилось именно такое число. Как его можно рассчитать без программы?

Объясните, почему при сложении в третьей строке в варианте «без учёта знака» получилось именно такое число. Как его можно рассчитать без программы?

## Практическая работа № 11.

### Арифметические операции с целыми числами

#### Знакомство с программой «ЛамПанель»

Кроме регистров общего назначения, с которыми мы уже работали, в процессоре есть служебные регистры. В этой работе мы будем использовать один из них – регистр состояния **PS**. Используются только три младших бита этого регистра (см. рисунок), которые показывают, какой результат был получен в результате последней операции:

Регистры				
<b>R0</b>	1000	0000	0000	0001
<b>R1</b>	0000	0000	0000	0100
<b>R2</b>	0000	0000	0000	0100
<b>R3</b>	1111	1111	1111	1111
<b>PC</b>	0000	0000	0000	1000
<b>SP</b>	0000	0001	0000	0000
<b>PS</b>	0000	0000	0000	0001
				<b>CZN</b>

- **бит C** (от англ. *carry* – перенос) установлен (равен 1), если произошел перенос; в остальных случаях сброшен (равен 0);
- **бит Z** (от англ. *zero* – ноль) установлен, если результат последней операции – ноль; в остальных случаях сброшен;
- **бит N** (от англ. *negative* – отрицательный) установлен, если результат последней операции отрицательный; в остальных случаях сброшен.
- **бит O** (от англ. *overflow* – переполнение) установлен, если в результате последней операции произошло переполнение и результат неверен; в остальных случаях сброшен.

Эти биты могут учитываться при выполнении следующей команды. Например, для организации цикла используем команду перехода

#### **JNZ метка**

Если результат предыдущей операции НЕ равен нулю, то происходит переход на указанную метку. Вот пример программы, которая вычисляет сумму натуральных чисел от 1 до 5:

```

MOV 0, R0      ; начальное значение суммы
MOV 5, R1      ; количество шагов цикла
m:             ; метка обозначает начало цикла
  ADD R1, R0   ; R0:= R0 + R1
  SUB 1, R1    ; уменьшить R1 – оставшееся число шагов
  JNZ m        ; переход, если получился не ноль
STOP
    
```

Здесь в каждой строчке после точки с запятой записан комментарий, объясняющий ее действие. Давайте проследим, как выполняется эта программа:

Команда	R0	R1	бит Z	переход
MOV 0,R0	0		1	
MOV 5,R1		5	0	
ADD R1,R0	5		0	
SUB 1,R1		4	0	
JNZ m			0	да
ADD R1,R0	9		0	
SUB 1,R1		3	0	
JNZ m			0	да
ADD R1,R0	12		0	
SUB 1,R1		2	0	
JNZ m			0	да
ADD R1,R0	14		0	
SUB 1,R1		1	0	

JNZ m			0	да
ADD R1,R0	15		0	
SUB 1,R1		0	1	
JNZ m			1	нет
STOP				

Цикл заканчивается, потому что бит **Z** равен 1 (результат последней операции вычитания – ноль) и перехода по команде **JNZ** не происходит.

Задание на практическую работу

#### Уровень А.

1. Запустите тренажер «Лампанель». Вычислите приведенные выражения с помощью программы. Запишите в таблицу результаты, значение знакового бита и битов **O**, **C**, **N** и **Z**.

*Замечание: в программу нужно вводить числа в шестнадцатеричной системе счисления!*

Выражение	Результат		Знаковый бит	Биты состояния			
	без учета знака	с учетом знака		O	C	Z	N
32760+32752							
-32760-32752							
256-256							

Объясните полученные результаты:

2. Определите, в каких случаях произошло переполнение. Выделите эти строки таблицы желтым фоном.

#### Уровень В.

3. С помощью программы, приведенной в теоретической части, вычислите сумму натуральных чисел от 1 до 100.

Программа:

*Замечание: в программу нужно вводить числа в шестнадцатеричной системе счисления!*

Ответ:

#### Уровень С.

4. Напишите программу, которая вычисляет значение *факториала* – произведения всех натуральных чисел от 1 до заданного числа. Например, факториал числа 5 равен  $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$ .

Для выполнения умножения используйте команду **MUL** (см. справочную систему, клавиша F1).

Программа:

С помощью программы заполните таблицу:

N	N!	
	без учета знака	с учетом знака
5		
6		
7		
8		
9		

Объясните результаты, полученные в последних двух строчках:



## Практическая работа № 12.

### Логические операции и сдвиги

Знакомство с программой «ЛамПанель»

В программе «ЛамПанель» можно использовать логические операции «НЕ» (уже знакомая нам команда **NOT**), «И» (команда **AND**), «ИЛИ» (команда **OR**) и «исключающее ИЛИ» (команда **XOR**). В последних трех командах после названия команды сначала указывается маска, а затем через запятую – регистр, к которому применяется логическая операция. Например, команда

**AND FF,R0**

обнуляет старшие 8 бит (старший байт) регистра **R0**. Маска может находиться в регистре, например, последовательность команд

**MOV FF,R1**

**OR R1,R0**

устанавливает в единицу 8 младших бит регистра **R0**, а остальные оставляет без изменений.

Для выполнения сдвигов используются следующие команды

**SHL 1,R0** ; логический сдвиг влево на 1 бит

**SHR 2,R0** ; логический сдвиг вправо на 2 бита

**SAR 1,R0** ; арифметический сдвиг вправо на 1 бит

**ROL 2,R0** ; циклический сдвиг влево на 2 бита

**ROR 3,R0** ; циклический сдвиг вправо на 3 бита

Конечно, сдвиг может применяться к любому регистру общего назначения, а не только к **R0**.

Задание на практическую работу

#### Уровень А.

1. Запустите тренажер «ЛамПанель». Напишите программу, которая решает следующую задачу, используя логические операции:

*В регистрах R1, R2 и R3 записаны коды трех десятичных цифр, составляющих трехзначное число (соответственно сотни, десятки и единицы). Построить в регистре R0 это число. Например, если  $R1=31_{16}$ ,  $R2=32_{16}$  и  $R3=33_{16}$ , в регистре R0 должно получиться десятичное число 123.*

Программа:

2. Используя программу «ЛамПанель», определите и запишите в таблицу значения регистра R0 после выполнения каждой из следующих команд:

	Команда	R0
1	<b>MOV 1234, R0</b>	
2	<b>XOR ABCD, R0</b>	
3	<b>XOR ABCD, R0</b>	

Ответьте на вопросы:

- как изменится результат выполнения программы, если в команде 1 записать в R0 другое число?

- как изменится результат выполнения программы, если в командах 2 и 3 заменить маску на другую, например, на  $CB24_{16}$ ?

- как изменится результат выполнения программы, если маску в команде 2 изменить, а маску в команде 3 не менять?

#### Уровень В.

3. Запишите в таблицу десятичные числа, которые будут получены в регистре **R0** после

выполнения каждой команды этой программы при разных начальных значениях **R0** (две команды выполняются последовательно одна за другой):

Начальное значение	255	254	252	-255	-254	-252
<b>SHR 2,R0</b>						
<b>SHL 2,R0</b>						

*Замечание: не забудьте перевести числа в шестнадцатеричную систему!  
Когда последовательное выполнение этих двух команд не изменяет данные?*

### Уровень С.

4. Напишите программу, которая решает следующую задачу, используя логические операции и сдвиги:

*При кодировании цвета используются 4-битные значения составляющих R (красная), G (зеленая) и B (синяя). Коды этих составляющих записаны в регистрах R1, R2 и R3. Построить в регистре R0 полный код цвета. Например, если  $R1=A_{16}$ ,  $R2=B_{16}$  и  $R3=C_{16}$ , в регистре R0 должно получиться число  $ABC_{16}$ .*

*Программа:*

5. Напишите программу, которая умножает число в регистре **R0** на 10, не применяя команду умножения. Используйте арифметические операции и сдвиги.

*Программа:*

## Глава 5. Устройство компьютера

### Практические работы

Для выполнения этих работ используется учебный компьютер «ЛамПанель», который можно загрузить со страницы <http://kpolyakov.spb.ru/prog/lamp.htm>.

#### Практическая работа № 13.

##### Выбор конфигурации компьютера

Используя данные любого интернет-магазина, подберите конфигурацию компьютера в соответствии с заданием учителя. Варианты для выполнения работы:

- 1) офисный компьютер;
- 2) домашний компьютер;
- 3) игровой компьютер;
- 4) компьютер для программиста;
- 5) компьютер для дизайнера;
- 6) компьютер для разработчика веб-сайтов;
- 7) компьютер для пенсионера.

Выпишите задачи, которые пользователь будет решать с помощью компьютера.

Ответ:

--

Оцените возможные расходы, постарайтесь их сократить. Обсудите результаты в классе.

	Марка	Цена
Центральный процессор		
Объём оперативной памяти		
Объём жёсткого диска		
Видеокарта		
Звуковая карта		
Монитор		
DVD-дисковод (если нужно)		
Операционная система		
Клавиатура		
Мышь		
Принтер (если нужно)		
Сканер (если нужно)		

#### Практическая работа № 14.

##### Исследование компьютера

##### Аппаратное обеспечение

Используя любые программы, предоставляющие информацию о компьютере (по указанию учителя), определите для вашего компьютера:

Тип центрального процессора	
Размер оперативной памяти	
Тип видеоадаптера	
Объём видеопамати	
Тип сетевой карты	
Тип звуковой карты	
Объём жёсткого диска	

В операционной системе Windows можно использовать бесплатные версии утилит **CPU-Z, SiSoft Sandra, SIW, Speccy**; в операционной системе Linux – программы **Inex, Hardinfo** и другие.

### Программное обеспечение

Используя любые программы и средства, предоставляющие информацию о программном обеспечении компьютера (по указанию учителя), определите для вашего компьютера:

Название и версию операционной системы	
Офисный пакет	
Текстовый процессор	
Электронные таблицы	
Графический редактор	
Браузеры	
Среды программирования	

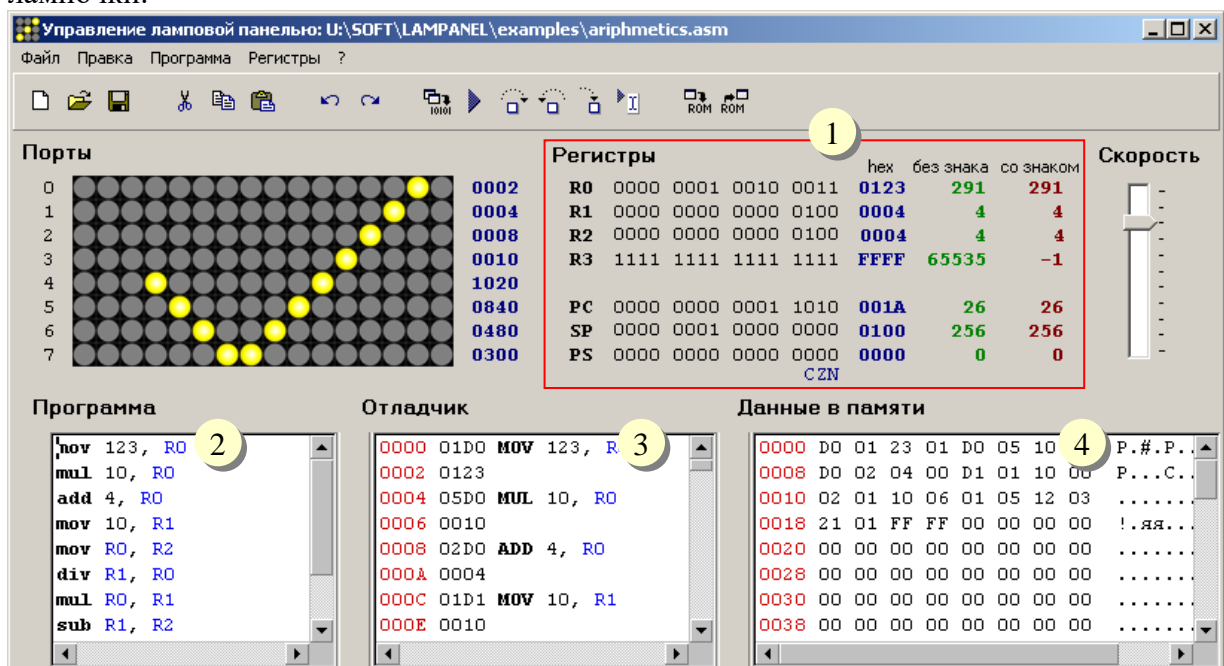
В операционной системе Windows для выполнения работы можно использовать *Панель управления* или специальные утилиты, в операционной системе Linux – Центр приложений или подобные средства.

### Практическая работа № 15.

#### Моделирование работы компьютера

*Возможности программы «ЛамПанель»*

Программа «ЛамПанель» – это модель процессора, который управляет ламповой панелью, то есть, может с помощью специальных команд зажигать и гасить определенные лампочки.



Вы уже знаете, что в области 1 на рисунке показано содержание регистров, в области 2 записан текст программы. Сегодня мы познакомимся еще с двумя областями:

- 3 – отладчик
- 4 – оперативная память

#### Коды в памяти

**Память** (область 4) разбита на ячейки размером 1 байт = 8 бит. Значение каждой ячейки записывается в виде двух шестнадцатеричных цифр – каждая из них кодируется ровно четырьмя битами.

Каждая строка в окне 4 содержит 8 байт памяти. Число слева, выделенное красным цветом – это *адрес* (номер) первой ячейки, показанной в этой строке. Справа от

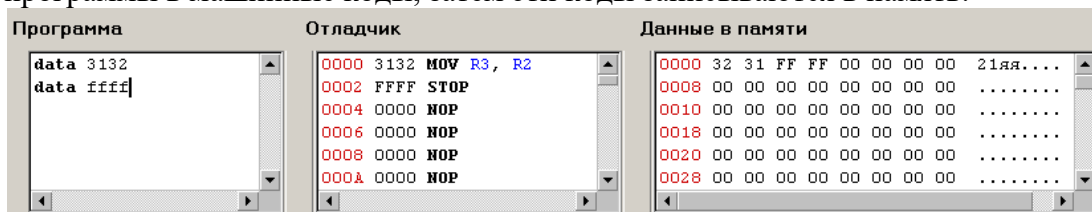
шестнадцатеричных кодов показана символьная строка из 8 символов – те же данные, только представленные как символы.

Данные можно записывать в память напрямую, используя команду DATA, например, можно набрать такую программу:

```
DATA 3132
```

```
DATA FFFF
```

Если теперь нажать клавиши **Ctrl+F9**, происходит *ассемблирование* («сборка») – перевод программы в машинные коды, затем эти коды записываются в память:



Посмотрим на окно отладчика. Каждый байт памяти имеет собственный адрес, адреса соседних байтов отличаются на единицу. Однако байтовые ячейки, как правило, слишком малы для хранения чисел (как целых, так и дробных) и машинных команд. Поэтому процессор должен уметь работать с более крупными блоками данных, которые часто называют *машинными словами*. Программа «ЛамПанель» умеет работать с 16-битными словами, то есть может сразу читать из памяти (в регистр) и записывать в память двухбайтный блок.

Любая машинная команда состоит из целого числа 16-битных слов, то есть из четного числа байтов. По договоренности адресом двухбайтового слова считается меньший из адресов входящих в него байтов, причем адрес этот обязательно должен быть четным.

В нашем случае память записаны два 16-битных слова (4 байта),  $3132_{16}$  и  $FFFF_{16}$ , причем эти слова процессор распознал как две команды:

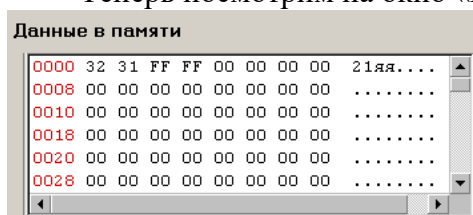
```
MOV R3,R2
```

```
STOP
```

Такой перевод из кодов команд в их символьное обозначение называется *дисассемблирование* (обратное ассемблирование, «разборка»).

Эту программу можно запустить, нажав на клавишу **F9**, и убедиться, что она действительно копирует содержимое регистра **R3** в регистр **R2**.

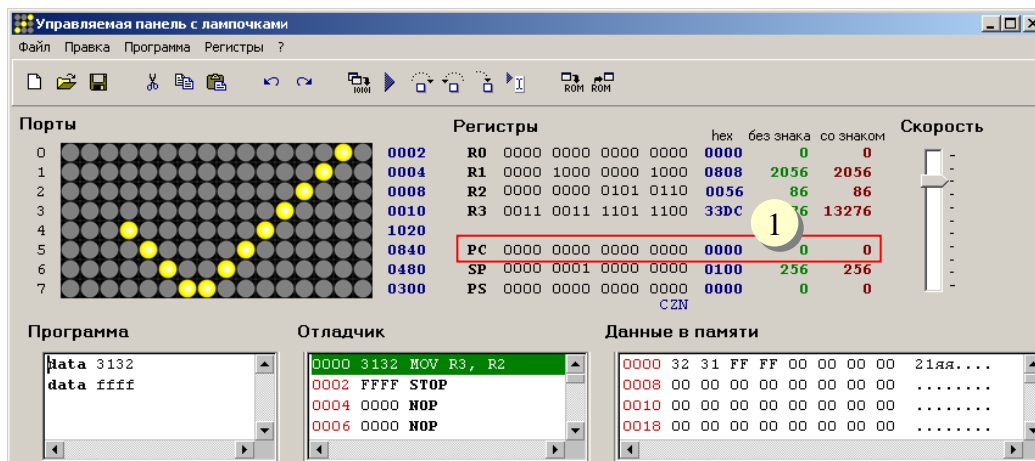
Теперь посмотрим на окно «Память»:



Видим, что байты 16-битного слова расположены «наоборот» – сначала младший байт  $32_{16}$ , а затем – старший  $31_{16}$ . Кроме того, в правой части окна видно, что эти коды соответствуют символам «21яя». Все специальные коды (не соответствующие каким-то принятым изображениям символов) обозначены точками. Таким образом, компьютер, основанный на архитектуре фон Неймана, не может самостоятельно различить, где данные, а где команды.

### Выполнение программы

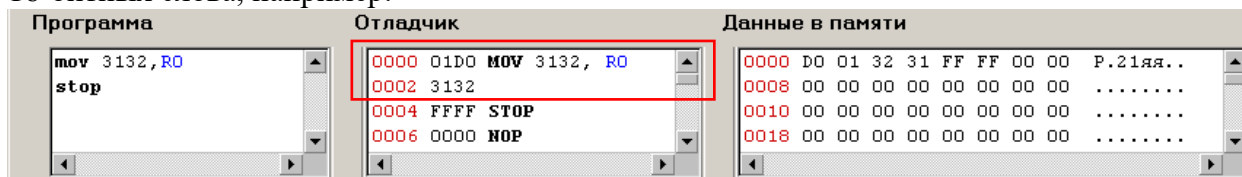
Теперь выполним программу в пошаговом режиме, нажав на клавишу **F8**. После этого в регистр **PC** (англ. *program counter* – программный счётчик) записывается стартовый адрес 0, с которого начинается выполнение программы. В окне *Отладчик* зелёным цветом выделена первая команда. Она еще не выполнялась, но будет выполнена при повторном нажатии **F8**. При этом регистр **PC**, будет указывать на начало следующей команды (которая еще не выполнялась).



Таким образом:

- регистр **PC** содержит адрес команды, которая будет выполнена следующей; как только эта команда будет выбрана из памяти, регистр-счетчик автоматически будет увеличен так, чтобы снова указывать на очередную команду;
- процессор воспринимает байты, расположенные по этому адресу, как код команды (а не как данные);
- программа всегда начинает выполняться с некоторого известного (в данном случае – нулевого) адреса, который «вшит» в компьютер и автоматически заносится в регистр **PC** при его включении;
- программа останавливается, когда будет выполнена команда **STOP** с кодом FFFF<sub>16</sub>.

Заметим, что команды, содержащие числовые данные, могут занимать в памяти два 16-битных слова, например:



### Работа с памятью

Для того, чтобы обрабатывать данные из оперативной памяти, процессор должен загрузить их в регистры. Поскольку программа и данные расположены в одной области памяти, размещать данные можно сразу после команды **STOP**:

```

...
STOP
D:
DATA 1234

```

Метка **D** нужна для того, чтобы удобно было загружать адрес блока данных в регистр, например, так:

**MOV @D, R0 ; загрузить адрес метки D в R0**

Можно считать, что **D** – это переменная программы. После этого легко загрузить в регистр данные из памяти:

**MOV (R0),R1 ; загрузить в R1 данные, адрес которых записан в R0**

Аналогичной командой можно изменить содержимое ячейки памяти:

**MOV R2,(R0) ; записать данные из R2 в ячейку, адрес которой ; записан в R0**

Заметим, что с помощью этого метода можно сразу обратиться к любой ячейке памяти, поэтому такой вид памяти называется *память с произвольным доступом* (англ. *RAM = random access memory*).

### Обработка отдельных байтов

Как вы знаете, минимальная ячейка памяти, имеющая собственный адрес, называется байтом. В современных компьютерах 1 байт составляют 8 бит. Поэтому

компьютер должен иметь возможность работать не только с 16-битными словами, но и с отдельными байтами.

Во-первых, в команде **DATA** можно задавать не только шестнадцатеричные числа, но и символьные строки, взятые в кавычки.

**D:**  
**DATA "ABCDEFGH"**

В этом случае символы заданной строки записываются в память последовательно, начиная с первого. Теперь запишем адрес строки в регистр в какой-нибудь регистр, например, в **R0**:

**MOV @D,R0**

Чтобы работать с отдельными байтами, используют байтовые версии команд, которые заканчиваются на букву **B**. Например, байтовый вариант команды **MOV** называется **MOVB**. Команда

**MOVB (R0),R2**

загружает один байт из памяти (расположенный по адресу, который записан в **R0**) в регистр **R2**. Теперь рассмотрим такую задачу – преобразовать все заглавные латинские буквы в строчные. Для этого нужно посмотреть, чем отличаются коды заглавных и строчных букв:

<b>A:</b> 01000001	<b>B:</b> 01000010	<b>C:</b> 01000011	<b>D:</b> 01000100
<b>a:</b> 01100001	<b>b:</b> 01100010	<b>c:</b> 01100011	<b>d:</b> 01100100

Оказывается, коды заглавных и соответствующих строчных букв отличаются одним битом (этот пятый бит выделен желтым фоном). Поэтому для того, чтобы получить из заглавной буквы строчную букву, нужно установить 5-й бит (сделать его равным 1). Для этого можно, например, использовать логическую операцию «ИЛИ» с маской 0020<sub>16</sub>, в которой 5-й бит установлен, а остальные – сброшены:

**OR 20,R2**

Затем нужно записать результат обратно в память, по адресу, находящемуся в **R0**:

**MOVB R2,(R0)**

Для перехода к следующему символу просто увеличиваем **R0** на единицу

**ADD 1,R0**

и выполняем те же самые команды. Отметим, что две команды

**MOVB R2,(R0) ; записать байт в память**

**ADD 1,R0 ; к следующему байту**

можно заменить на одну, которая делает то же самое:

**MOVB R2,(R0)+ ; записать байт в память и перейти**  
**; к следующему байту**

Чтобы обработать 6 символов, можно организовать цикл со счётчиком в регистре **R1**:

**MOV @D,R0 ; адрес данных – в R0**

**MOV 6,R1 ; счётчик шагов цикла (сделать 6 раз)**

**M:**

**MOVB (R0),R2 ; прочитать байт из памяти**

**OR 20,R2 ; заглавную – в строчную**

**MOVB R2,(R0)+ ; записать байт в память и перейти к следующему**

**SUB 1,R1 ; уменьшить счетчик**

**JNZ M ; если не все сделали – переход на метку M**

**STOP**

**D:**

**DATA "ABCDEFGH"**

*Задание на практическую работу*

**Уровень H.**

17. Запустите тренажёр «ЛамПанель». Введите программу

**DATA 01D0**

**DATA 3536**  
**DATA 0101**  
**DATA FFFF**

Используя дизассемблер программы «ЛамПанель», запишите эту программу на языке ассемблера:

*Программа:*

Запишите код команды **STOP**:

Запишите содержимое памяти, в которой располагается эта программы, в виде последовательности символов:

18. Как вы думаете, какой код будет иметь команда **MOV R1,R3**? Проверьте свой ответ с помощью программы.

### Уровень I.

19. Блок данных программы выглядит так.

**A:**  
**DATA 1234**  
**B:**  
**DATA 4321**  
**SUM:**  
**DATA 0**

Напишите программу, которая складывает переменные **A** и **B** и записывает результат в переменную **SUM**:

*Программа:*

### Уровень J.

20. Напишите программу, которая преобразует строчные буквы в заглавные, используя байтовые операции. Блок данных может выглядеть так:

**D:**  
**DATA "abcdefgh"**

*Программа:*

### Уровень K.

21. Усовершенствуйте программу так, чтобы цикл останавливался не после заданного количества букв, а тогда, когда очередной прочитанный байт равен 0. Возможно, вам понадобятся другие команды условного или безусловного перехода – изучите их по справочной системе (клавиша **F1**).

*Программа:*

22. Поскольку в компьютере с архитектурой фон Неймана программа и данные расположены в одной области памяти, программа может менять свой собственный код. Напишите какую-нибудь программу, которая изменяет сама себя во время работы.

*Программа:*



## Практическая работа № 16.

### Использование облачных хранилищ данных

Для выполнения работы можно использовать одно из бесплатных облачных хранилищ данных, например, Облако Mail.Ru (<https://cloud.mail.ru>), Яндекс.Диск (<https://disk.yandex.ru>) или Google Диск (<https://drive.google.com>).

*Примечание для учителя:* Ссылку в задании 1 рекомендуется заменить на свою собственную, предназначенную именно вашим ученикам.

#### Уровень А.

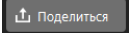
5. Скачайте файл из облачного хранилища по ссылке

<https://yadi.sk/i/mC2ajBgzdYsVYA>.

6. Определите данные файла:

Название файла	
Тип данных	
Размер файла	
Дата последнего изменения	
Адрес сайта, где есть похожие материалы	

#### Уровень В. (выполнение задания в парах)

7. Зарегистрируйтесь в одном из облачных хранилищ.  
8. Загрузите в облачное хранилище любой файл по вашему выбору.  
9. Создайте ссылку для получения доступа к этому файлу (например, в Яндекс.Диске для этого нужно щёлкнуть по кнопке  или выбрать этот пункт из контекстного меню).  
10. Перешлите ссылку своему напарнику и предложите ему скачать файл и определить его характеристики, как в п. 2.

Название файла	
Тип данных	
Размер файла	
Дата последнего изменения	

#### Уровень С. (выполнение задания в парах)

11. В вашем хранилище создайте новую папку для совместной работы.  
12. Настройте общий доступ для этой папки, вышлите приглашение напарнику на электронную почту.  
13. Загрузите в папку какой-либо файл. Предложите напарнику скачать его, используя общий доступ к папке (а не прямую ссылку на файл).  
14. Скачайте файл из папки напарника и определите его характеристики:

Название файла	
Тип данных	
Размер файла	
Дата последнего изменения	

15. Предложите напарнику загрузить файл в вашу папку с общим доступом. Определите его характеристики:

Название файла	
Тип данных	
Размер файла	
Дата последнего изменения	

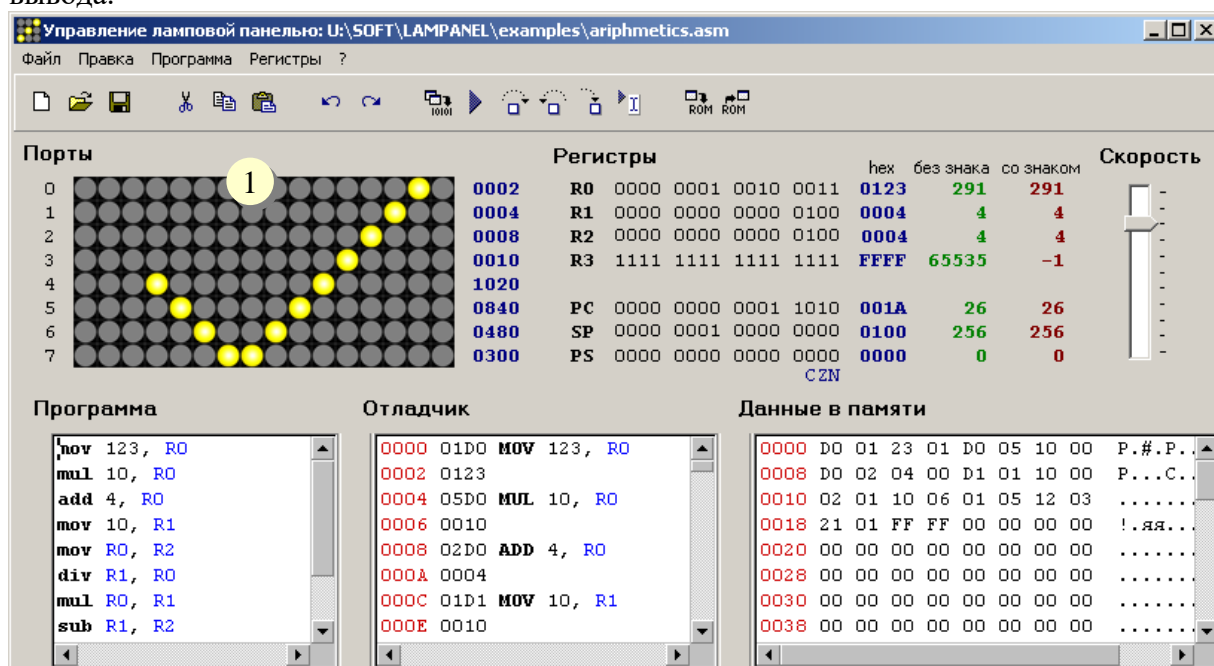
16. Загрузите свой файл в папку напарника, используя общий доступ к ней.

## Практическая работа № 17.

### Процессор и устройства вывода

Возможности программы «ЛамПанель»

Наконец, мы подошли к самой интересной возможности программы «ЛамПанель» - управлению ламповой панелью. Ламповая панель (цифра 1 на рисунке) – это устройство Вывода.



Обмен данными процессора и внешнего устройства происходит через *порты* – регистры контроллера внешнего устройства. У ламповой панели 8 портов, которые называются **P0**, **P1**, **P2**, **P3**, **P4**, **P5**, **P6** и **P7**. Каждый порт «отвечает» за одну строку лампочек, например, для того, чтобы «зажечь» всю верхнюю строку нужно записать в порт **P0** код  $FFFF_{16}$  (все 16 бит – единичные). Для этого можно использовать, например, команды

```
MOV FFFF, R0
OUT R0, P0
```

К сожалению, записать число сразу в порт нельзя – сначала нужно записать его в регистр (в данном примере – в **R0**), а потом – из регистра в порт.

Для того, чтобы изменить второй сверху ряд лампочек, нужно записать новое значение в **P1** и т.д.; последний ряд управляется портом **P7**. Например, для того, чтобы все ряды лампочек горели одинаково, можно записать нужный код сначала в регистр:


```
MOV AAAA, R0
```

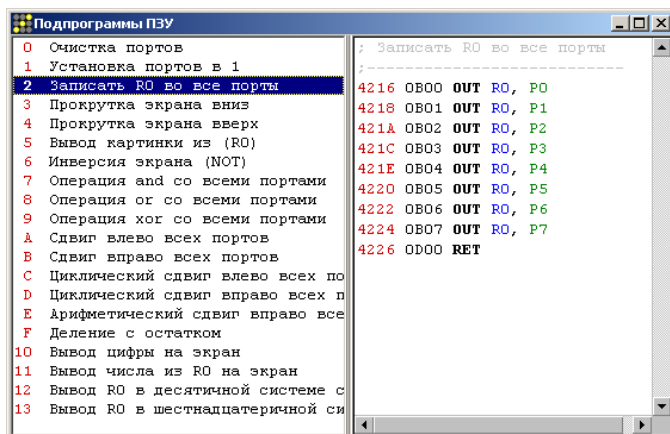
а затем из этого регистра – во все порты:

```
OUT R0, P0
OUT R0, P1
...
OUT R0, P7
```

Здесь многоточие обозначает аналогичные команды записи содержимого регистра **R0** в порты **P2...P6**. Однако вместо последней серии из 8 команд можно использовать всего одну:

```
SYSTEM 2
```

Эта команда вызывает системную процедуру с номером 2, находящуюся в ПЗУ компьютера. Для того, чтобы увидеть все процедуры, которые записаны в ПЗУ, нужно щелкнуть по кнопке  или выбрать пункт верхнего меню *Программа – Просмотр ПЗУ*. После этого появляется окно, в левой части которого перечислены все системные процедуры (с их номерами), а в правой части показывается код выбранной процедуры:



В этом списке есть много полезных процедур, в том числе

0 – очистка экрана (погасить все лампочки);

1 – зажечь все лампочки на панели;

3-4 – прокрутка изображения вниз и вверх;

6-9 – логические операции;

A<sub>16</sub>-E<sub>16</sub> – сдвиги битов;

12<sub>16</sub> – вывод числа, записанного в регистр **R0**, в десятичной системе счисления;

13<sub>16</sub> – вывод числа, записанного в регистр **R0**, в шестнадцатеричной системе счисления.

Обратите внимание, что номер системной процедуры задается в шестнадцатеричной системе счисления.

Рассмотрим еще одну задачу: вывести на экран рисунок, закодированный в виде шестнадцатеричных чисел (бит, равный единице, обозначает горящую лампочку). Для этого нужно сначала записать коды рисунка в память. Поскольку наш компьютер основан на архитектуре фон Неймана, в нем программа и данные находятся в одной области памяти. Поэтому данные можно записать с помощью специальной команды **DATA** после команды **STOP**:

... ; здесь будет программа

**STOP**

**M:** ; метка – начало блока данных

**DATA AAAA** ; код первой строчки

**DATA 5555**

**DATA AAAA**

**DATA 5555**

**DATA AAAA**

**DATA 5555**

**DATA AAAA**

**DATA 5555** ; код последней строчки

Для того, чтобы вывести этот рисунок на экран, нужно записать его адрес в регистр **R0** и вызвать системную процедуру с номером 5:

**MOV @M, R0** ; записать адрес метки **M** в регистр **R0**

**SYSTEM 5** ; вывести на экран рисунок, адрес которого в **R0**

**STOP**

**M:**

**DATA AAAA**; код первой строчки

...

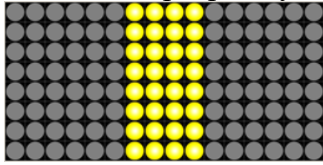
*Задание на практическую работу*

**Уровень А.**

17. Запишите в таблицу минимальное и максимальное числа, которые можно вывести на ламповую панель, если использовать шестнадцатеричную систему:

минимальное число		максимальное число	
десятичное	шестнадцатеричное	десятичное	шестнадцатеричное

18. Составьте программу, после выполнения которой ламповая панель выглядит так:



*Программа:*

19. Как вы думаете, что выведет приведенная выше (в теоретической части) программа, которая вызывает системную процедуру с номером 5? Проверьте ваш ответ с помощью тренажёра.

*Ответ:*

### Уровень В.

20. Закодируйте изображение домика и выведите его на экран.

*Программа:*

### Уровень С.

21. Добавьте в предыдущую программу команды, которые сначала шифруют изображение, используя операцию «исключающее ИЛИ» с маской  $ВСА7_{16}$ , а затем – восстанавливают исходное изображение. При изменении маски программа не должна изменяться. Изучите текст системной процедуры, которую вы используете.

*Программа:*

### Уровень D.

22. Напишите программу, которая делает «бегущую строку» из рисунка-домика.

*Программа:*

23. Напишите программу, которая организует «обратный отсчет» от 100 до 0, а затем выводит рисунок с домиком и останавливается.

*Программа:*

## Глава 6. Программное обеспечение

### Практические работы

#### Практическая работа № 18.

#### Инсталляция программ

##### Для ОС Windows

1. Запустите инсталлятор-тренажёр программы **FineReader**, расположенный по адресу, указанному учителем.
2. Установите программу со следующими настройками:
  - полная версия, код: **ABBY-1234-5678-9012-3456**
  - папка для установки программы – по указанию учителя (должна быть открыта на запись);
  - остальные параметры – в соответствии с персональным заданием на карточке.
3. После предъявления работы учителю полностью удалите программу.

##### Для ОС Linux

1. Выполните установку (удаление) какой-либо программы (по заданию учителя) с помощью менеджера пакетов или из командной строки.

Вариант 1	Вариант 2	Вариант 3
<b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, белорусский, узбекский, Basic <input checked="" type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word	<b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, киргизский, суахили, Java, Pascal, эсперанто <input type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word	<b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, французский, чувашский, Fortran <input checked="" type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word
Вариант 4	Вариант 5	Вариант 6
<b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, немецкий, казахский, суахили, эсперанто <input type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word	<b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, итальянский, албанский, COBOL <input checked="" type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word	<b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, испанский, шведский, эсперанто <input type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word
Вариант 7	Вариант 8	Вариант 9
<b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, английский, шведский, азербайджанский, Pascal <input checked="" type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word	<b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, немецкий, испанский, узбекский, Java, COBOL <input checked="" type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word	<b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, итальянский, киргизский, узбекский, Basic, Fortran <input type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word
Вариант 10	Вариант 11	Вариант 12
<b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, испанский, белорусский, албанский, Pascal, эсперанто <input checked="" type="checkbox"/> Обучение на примерах	<b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, английский, немецкий, белорусский, узбекский, Java <input type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы	<b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, французский, украинский, белорусский, Fortran, эсперанто <input checked="" type="checkbox"/> Обучение на примерах

<input checked="" type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word	<input checked="" type="checkbox"/> Иконка на панели MS Word	<input type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word
---	--	---

<b>Вариант 13</b> <b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, немецкий, чувашский, узбекский, COBOL <input type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word	<b>Вариант 14</b> <b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, испанский, албанский, Pascal, эсперанто <input checked="" type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word	<b>Вариант 15</b> <b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, итальянский, суахили, узбекский, Java <input checked="" type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word
<b>Вариант 16</b> <b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, шведский, испанский, азербайджанский, Pascal <input type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word	<b>Вариант 17</b> <b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, украинский, белорусский, киргизский, узбекский, Java, эсперанто <input checked="" type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input type="checkbox"/> Иконка на панели MS Word	<b>Вариант 18</b> <b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, испанский, итальянский, киргизский, Fortran, Pascal <input type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word
<b>Вариант 19</b> <b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, немецкий, французский, чувашский, COBOL <input checked="" type="checkbox"/> Обучение на примерах <input type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word	<b>Вариант 20</b> <b>Интерфейс:</b> русский, английский <b>Языки распознавания:</b> русский, испанский, узбекский, суахили, Pascal, COBOL <input type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word	<b>Вариант 21</b> <b>Интерфейс:</b> русский <b>Языки распознавания:</b> русский, итальянский, немецкий, испанский, киргизский, Basic <input checked="" type="checkbox"/> Обучение на примерах <input checked="" type="checkbox"/> Демонстрационные файлы <input checked="" type="checkbox"/> Иконка на панели MS Word

**Практическая работа № 19.**  
**Сканирование и распознавание текстов**  
**Уровень L.**

- 1) Подготовьте бумажную страницу с печатным текстом, которую вы собираетесь сканировать.
- 2) Выполните сканирование в черно-белом режиме (оттенки серого) с разрешением 300 ppi и глубиной кодирования цвета 8 бит на пиксель (256 оттенков серого), сохраните файл в формате JPG.
- 3) С помощью программы или онлайн-сервисов (например, <http://en.pdf24.org/onlineConverter.html>) преобразуйте полученную картинку в формат PDF.
- 4) Найдите в Интернете информацию о формате DjVu.
- 5) Используя сервис <http://any2djvu.djvuzone.org/>, преобразуйте картинку в формат DjVu. Сравните качество и размеры полученных PDF-файла и DjVu-файла.

Формат	JPG	PDF	DjVu
Качество (1-5)			
Размер			

- 6) Выполните распознавание текста с помощью программы (например, *CuneiForm*) или онлайн-сервисов, например:
  - <http://www.newocr.com/>
  - <http://www.free-ocr.com/>
  - <http://www.ocronline.com/ru/>
  - <http://www.onlineocr.net/ru/>
  - <http://finereader.abbyyonline.com/>
- 7) Сохраните результат распознавания в виде документа в формате DOC и исправьте ошибки распознавания, если они были. Затем преобразуйте документ в формат PDF.

**Уровень M.**

- 8) Преобразуйте полученный PDF-документ в формат DjVu (с помощью сервиса <http://any2djvu.djvuzone.org/>) и сравните размеры полученных файлов, заполнив таблицу:

Тип документа	Размер
<i>Без распознавания</i>	
Формат JPG	
Формат PDF	
Формат DjVu	
<i>После распознавания</i>	
Формат DOC	
Формат PDF	
Формат DjVu	

- 9) Отсканируйте ту же страницу с разрешениями 200 ppi, 150 ppi и 100 ppi, и проверьте, как изменяется качество распознавания. Сделайте выводы.

**Уровень N.**

- 10) Попробуйте отсканировать и распознать страницу документа, содержащего рисунки, таблицы и формулы. Сделайте выводы.

## Практическая работа № 20. Возможности текстовых процессоров

### Уровень А.

1. Загрузите файл *Дейкстра.doc*.
2. Выделите заголовок стилем *Заголовок 1*.
3. Создайте стиль *Биография* со следующими характеристиками:
  - шрифт *Times New Roman*, 12 пт
  - абзацный отступ (отступ в первой строке) 1 см
  - выравнивание по ширине
  - междустрочный интервал 1,25 строки
  - дополнительные интервалы : до абзаца – 0, после абзаца – 6 птВсе абзацы текста оформите с помощью стиля *Биография*.
4. Включите режим автоматической расстановки переносов.
5. Включите проверку орфографии для русского языка.
6. Поместите информацию о книге *Operating System* в сноску в нижней части страницы.
7. Сделайте гиперссылками
  - имена ученых: *Тони Хоар*, *Никлаус Вирт* (только когда они встречаются в первый раз); гиперссылка должна указывать на статью в Интернете с информацией об этом ученом;
  - выражения «грамматика *Ван Вейнгаардена*» и «*Алгоритм Дейкстры*»; ссылки должна указывать на статьи с описанием упомянутой грамматики и алгоритма;
  - выражение «*премия Тьюринга*»; ссылка должна указывать на статью с информацией об этой премии.
8. С помощью тезауруса найдите синонимы и антонимы к слову «впоследствии».

### Уровень В.

9. Преобразуйте документ в формат PDF. Если на вашем компьютере не установлена программа-конвертер, можно использовать бесплатные онлайн-сервисы, например,  
<http://www.doc2pdf.net/>  
<http://convertonlinefree.com/>  
<http://docupub.com/pdfconvert/>

### Уровень С.

10. Используя доступные вам системы машинного перевода (например, встроенные возможности *Word* или системы онлайн-перевода типа <http://translate.ru> или <http://translate.google.com>) переведите текст на какой-нибудь другой язык (например, на немецкий), а затем выполните обратный перевод. Сравните результат с оригиналом и сделайте выводы.



## Практическая работа № 21.

### Набор математических текстов

#### Уровень А.

- 1) Скопируйте в свою папку файл **Банка.doc**.
- 2) Замените всю «ручную» нумерацию формул на новую, сделанную с помощью полей.
- 3) Замените «ручные» ссылки на формулы в тексте на обновляемые перекрестные ссылки.

#### Уровень В.

- 4) Переставьте раздел *Объем банки* в конец документа и обновите все поля. Проверьте, чтобы все ссылки обновились правильно.

#### Площадь поверхности банки

Полная площадь поверхности банки складывается из площади боковой поверхности  $S_{бок}$  и двух площадей оснований (доннышек)  $S_0$ :

$$S = S_{бок} + 2S_0 \quad (1)$$

Боковая поверхность – это прямоугольник, высота которого равна  $H$ , а ширина – длине окружности радиуса  $R$ , то есть  $2\pi R$ . Поэтому

$$S_{бок} = 2\pi R \cdot H \quad (2)$$

Доннышко банки (основание цилиндра) – это круг радиуса  $R$ , его площадь равна

$$S_0 = \pi \cdot R^2 \quad (3)$$

Подставляя формулы (2) и (3) в (1), получаем полную площадь поверхности:

$$S = 2\pi R \cdot H + 2\pi \cdot R^2 = 2\pi R(R + H) \quad (4)$$

#### Объем банки

Объем цилиндра равен произведению площади его основания на высоту

$$V = S_0 \cdot H \quad (5)$$

где  $S_0$  – площадь основания. Основание банки – это круг радиуса  $R$ , его площадь равна

$$S_0 = \pi \cdot R^2 \quad (6)$$

Подставляя формулу (6) в (5), получаем

$$V = \pi \cdot R^2 \cdot H \quad (7)$$

Формула (7) позволяет вычислить объем банки, зная ее высоту  $H$  и радиус  $R$ .

#### Уровень С.

- 5) Скопируйте в свою папку файл **Кнут.doc**.
- 6) Сделайте в конце этого документа список использованных источников и замените сноски на ссылки (поля). Книги в списке должны быть расставлены в алфавитном порядке по фамилии автора, а книги одного автора – по годам издания (в порядке возрастания).

#### Дональд Кнут



Дональд Кнут – американский учёный, почётный профессор Стэнфордского университета и нескольких других университетов в разных странах, преподаватель и идеолог программирования.

Самый известный труд Д. Кнута – это серия «Искусство программирования» [2], которая переиздавалась в России несколько раз.

Кнут известен также как создатель настольных издательских систем TEX и METAFONT [1], предназначенных для набора и вёрстки книг с большим количеством формул (в первую очередь – физико-математических). Эти системы распространяются бесплатно.

#### Список использованных источников

1. **Кнут, Д.** Все про TEX [Текст] / Д. Кнут. – М.: Вильямс, 2003.
2. **Кнут, Д.** Искусство программирования [Текст] – Т. 1-3. / Д. Кнут. – М.: Вильямс, 2007.

## Практическая работа № 22.

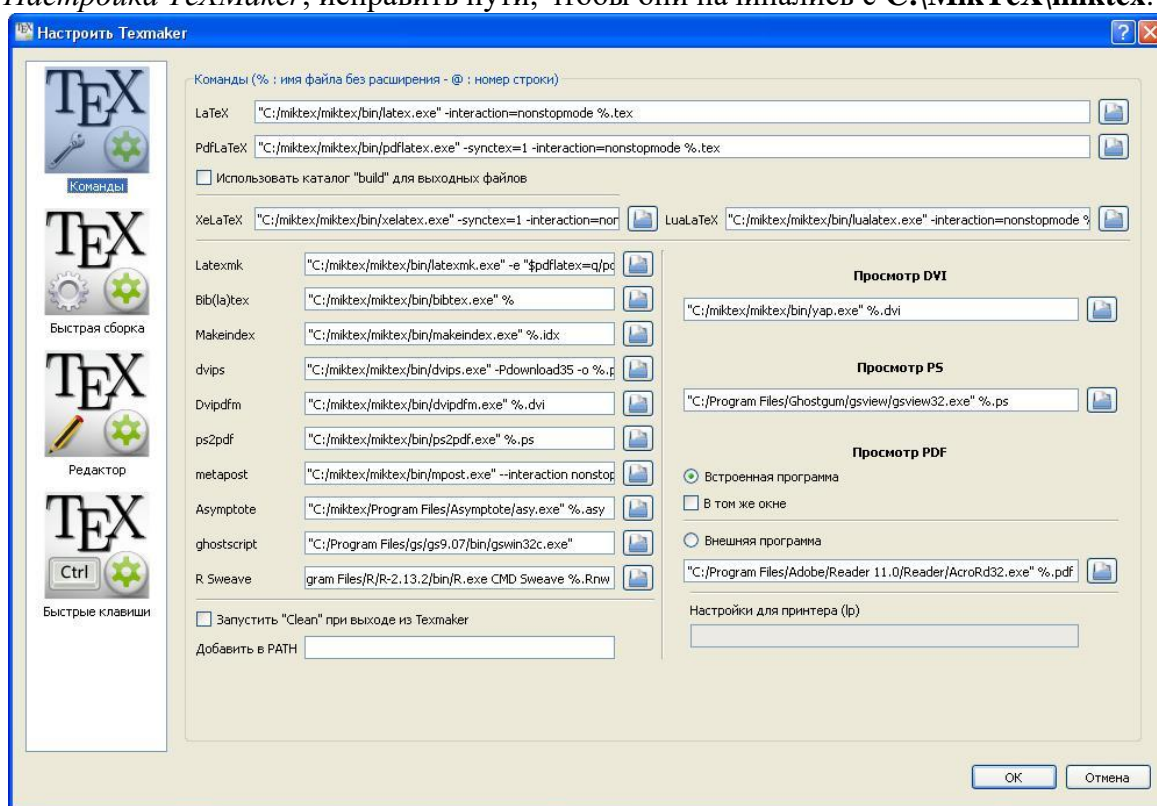
### Набор текстов в LaTeX

#### Установка LaTeX

Установка и настройка пакета среды для подготовки документов в TEX описана, например, в [статье А.С. Пережогина](#).

Готовую сборку MikTeX+TeXmaker для Windows, которая уже русифицирована и не требует установки, можно скачать по ссылке <https://yadi.sk/d/iuVcd0PihCQIFA>. Порядок установки этой сборки:

- 1) распаковать архив в папку **C:\MikTeX**
- 2) дать пользователю полные права на папку **C:\MikTeX\miktex**
- 3) создать на Рабочем столе ярлыки на программы-оболочки:  
**C:\MikTeX\TEXMAKER\texmaker.exe**  
**C:\MikTeX\miktex\bin\texworks.exe**
- 4) для настройки программы TeXMaker запустить её, зайти в меню *Настройка – Настройка TeXMaker*, исправить пути, чтобы они начинались с **C:\MikTeX\miktex:**



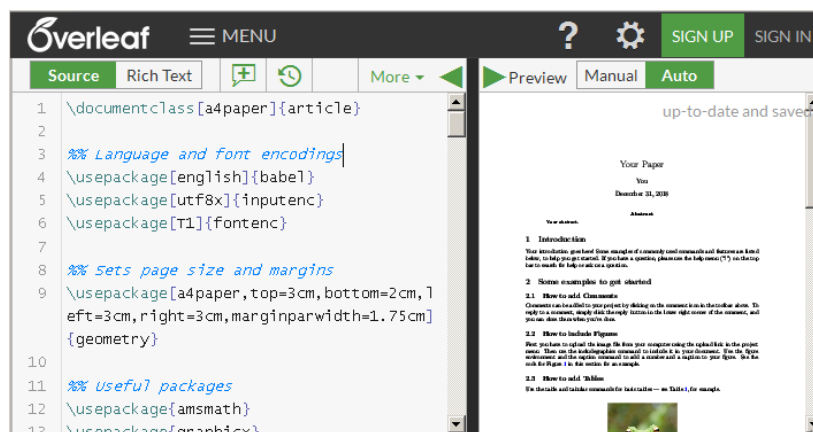
#### Онлайн-сервисы

Для выполнения работы можно использовать онлайн-сервисы, например, <https://v1.overleaf.com/> (в настоящее время требуется регистрация). На главной странице нужно щелкнуть по кнопке

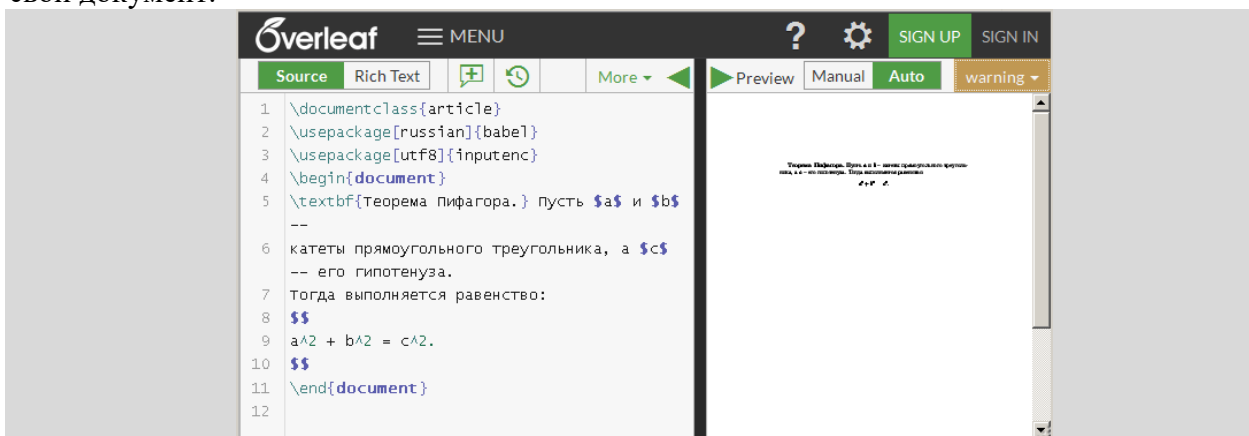
Start writing now! CREATE A NEW PAPER

После этого создаётся новый LaTeX-документ. Нужно переключиться в режим (исходный текст):

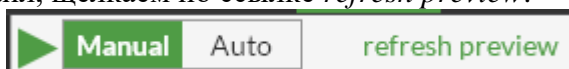
Source



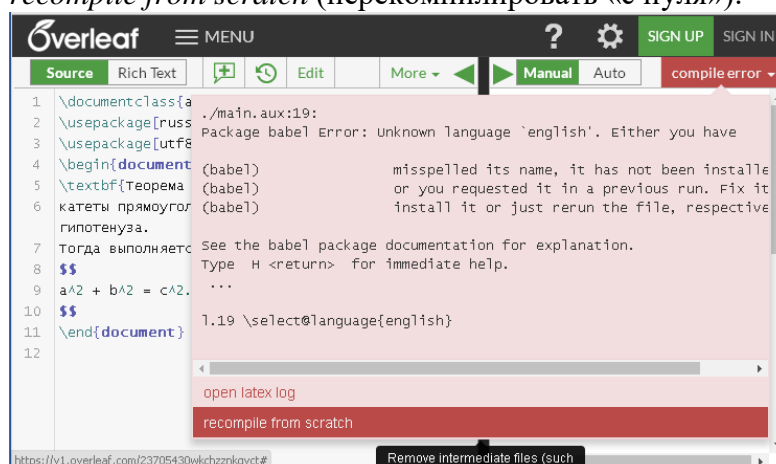
В верхней части окна предпросмотра (справа) переключаем режим на **Manual** (ручной) для того, чтобы при любом изменении текста программа сразу не перестраивала PDF-файл (на это уходит много времени). Затем удаляем содержимое документа в левой части и вводим свой документ.




Чтобы обновить PDF-файл, щёлкаем по ссылке *refresh preview*:



Поскольку сменился язык документа, будет выдано сообщение об ошибке, и нужно выбрать вариант *recompile from scratch* (перекомпилировать «с нуля»):



Если в файле нет ошибок, вы увидите в правой части готовый PDF-документ. Его можно скачать на компьютер, щелкнув по кнопке  в верхней части окна.

Кроме того, можно использовать и другие онлайн-сервисы:

<https://texviewer.herokuapp.com/>

<https://www.papeeria.com> (требуется регистрация)

## Задание на выполнение работы

**Уровень А.** Оформите с помощью LaTeX и сохраните документ в формате PDF:

**Удельная теплоёмкость** ( $c$ ) – это физическая величина, которая показывает, какое количество теплоты необходимо затратить для нагревания 1 кг вещества на 1 градус:

$$c = \frac{Q}{m(t_2 - t_1)}.$$

**Уровень В.** Оформите с помощью LaTeX и сохраните документ в формате PDF:

**Теорема Виета.** Сумма корней приведённого квадратного уравнения равна второму коэффициенту, взятому с противоположным знаком, а произведение корней равно свободному члену.

*Доказательство.* Рассмотрим приведённое квадратное уравнение. Обозначим второй коэффициент буквой  $p$ , а свободный член буквой  $q$ :

$$x^2 + px + q = 0.$$

Дискриминант этого уравнения  $D$  равен  $p^2 - 4q$ .

Пусть  $D > 0$ . Тогда это уравнение имеет два корня:

$$x_1 = \frac{-p - \sqrt{D}}{2}, \quad x_2 = \frac{-p + \sqrt{D}}{2}.$$

Вычислив сумму и произведение этих корней, получаем:

$$x_1 + x_2 = -p, \quad x_1 \cdot x_2 = q.$$

**Уровень С.** Оформите с помощью LaTeX и сохраните документ в формате PDF:

**Пример.** Докажем, что

$$(a^2 + bc)(b^2 + ac)(c^2 + ab) \geq 8a^2b^2c^2 \text{ при } a > 0, b > 0, c > 0.$$

Из соотношения между средним арифметическим и средним геометрическим двух положительных чисел следует, что при указанных значениях переменных

$$\frac{a^2 + bc}{2} \geq \sqrt{a^2bc}, \quad \frac{b^2 + ac}{2} \geq \sqrt{b^2ac}, \quad \frac{c^2 + ab}{2} \geq \sqrt{c^2ab}.$$

Перемножив эти равенства, получим

$$\frac{a^2 + bc}{2} \cdot \frac{b^2 + ac}{2} \cdot \frac{c^2 + ab}{2} \geq \sqrt{a^4b^4c^4},$$

откуда следует

$$(a^2 + bc)(b^2 + ac)(c^2 + ab) \geq 8a^2b^2c^2.$$

## Практическая работа № 23.

### Оформление рефератов

#### Правила оформления реферата

- 1) Поля страницы: верхнее – 2 см; нижнее – 2 см; левое – 3 см; правое – 1 см.
- 2) Шрифт заголовков – *Arial*;  
шрифт основного текста – *Times New Roman*, 14 пт,  
интервал – полуторный; выравнивание по ширине, с  
автоматической расстановкой переносов.
- 3) Реферат должен содержать
  - титульный лист;
  - оглавление;
  - введение (1-2 страницы);
  - основную часть (15-20 страниц);
  - заключение (1-2 страницы);
  - список использованных источников.

Каждая из этих частей (и каждая глава в основной части) начинается с новой страницы.

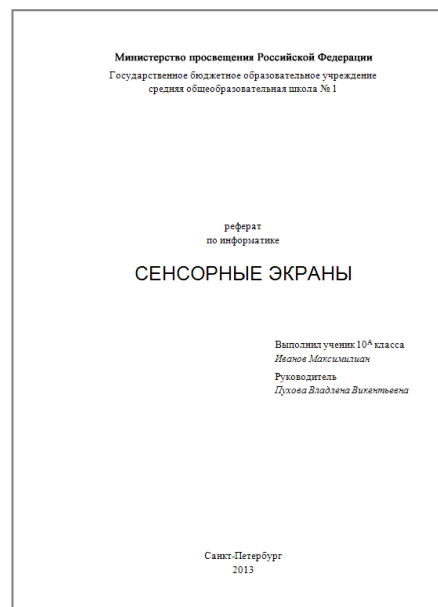
- 4) Страницы реферата нумеруются справа в верхней части страницы; титульный лист входит в общую нумерацию, но сам номер на нём не ставится).
- 5) Титульный лист должен содержать
  - название министерства (Министерство просвещения Российской Федерации)
  - название организации (школы, лицея)
  - слово «реферат», название предмета
  - название реферата (шрифт 28, полужирный)
  - фамилия и имя автора
  - фамилия, имя и отчество руководителя
  - в последних двух строчках – город и год, выравнивание по центру

Пример правильного оформления титульного листа см. на рисунке справа.

- 6) Оглавление строится автоматически на основе включенных в реферат заголовков разных уровней (*Заголовок 1*, *Заголовок 2* и т.д.).
- 7) Во введении дается краткая характеристика изучаемой темы, обосновывается ее актуальность и практическая значимость (где можно использовать). Часто бывает удобно писать введение уже после того, как реферат будет готов.
- 8) Основная часть делится на главы (разделы), которые могут в свою очередь делиться на подразделы. Здесь нужно изложить различные точки зрения на проблему и собственную позицию автора реферата.
- 9) В заключении подводятся итоги исследования, делаются выводы, формулируются новые результаты, полученные в ходе выполненной работы.
- 10) В списке использованных источников перечисляются все материалы, использованные при составлении реферата: книги, статьи, интернет-сайты, электронные ресурсы и др. Работы в списке перечисляются в алфавитном порядке по фамилии автора, работы одного автора – по возрасту года издания. В конце списка перечисляются источники на иностранных языках и интернет-ресурсы.

Примеры правильного оформления элементов списка использованных источников:

Книга:	1. Маслов, Л.А. Нашествия инопланетян на Европу [Текст] / Л.А. Маслов. - СПб.: НЛЮиздат, 2001. - 344 с.
Статья в журнале:	2. Боголюбов, А. Н. О вещественных резонансах в волноводе с неоднородным заполнением [Текст] / А. Н. Боголюбов, А. Л. Делицын, М. Д. Малых // Вестн. Моск. ун-та. Сер. 3, Физика.



	Астрономия. – 2001. – № 5. – С. 23-25.
<i>Электронный документ:</i>	3. Авилова Л. И. Развитие металлопроизводства в эпоху раннего металла (энеолит – поздний бронзовый век) [Электронный ресурс] : состояние проблемы и перспективы исследований // Вестн. РФФИ. 1997. № 2. URL: <a href="http://www.rfbr.ru/pics/22394ref/file.pdf">http://www.rfbr.ru/pics/22394ref/file.pdf</a> (дата обращения: 19.09.2007).
<i>Статья на сайте:</i>	4. Инвестиции останутся сырьевыми // PROGNOSIS.RU: ежедн. интернет-изд. 2006. 25 янв. URL: <a href="http://www.prognosis.ru/print.html?id=6464">http://www.prognosis.ru/print.html?id=6464</a> (дата обращения: 19.03.2007).
<i>Сайт или форум целиком</i>	5. Весь Богородский уезд: форум // Богородск – Ногинск. Богородское краеведение: сайт. Ногинск, 2006. URL: <a href="http://www.bogorodsk-noginsk.ru/forum/">http://www.bogorodsk-noginsk.ru/forum/</a> (дата обращения: 20.02.2007)

#### Задание на практическую работу

Требуется оформить реферат, записанный в файле **Сенсорные\_экраны.doc**, в соответствии с приведенными выше правилами. Для этого:

- 1) установите нужные поля страницы (см. выше)
- 2) сделайте титульный лист реферата (используйте свои имя и фамилию, а также ФИО вашего учителя);
- 3) измените стиль *Обычный* следующим образом:
  - шрифт *Times New Roman*, размер 14 пт
  - выравнивание по ширине
  - абзацный отступ 1 см;
  - интервал после абзаца 3 пт
  - межстрочный интервал 1,5 строки
и оформите этим стилем все абзацы текста;
- 4) включите автоматическую расстановку переносов;
- 5) добавьте нумерацию страниц (в правом верхнем углу, на титульном листе номер не ставится);
- 6) исправьте список использованных источников; например, вместо <http://ru.wikipedia.org> должно быть *Википедия (свободная энциклопедия)*. URL: <http://ru.wikipedia.org> (дата обращения ...)
- 7) сделайте так, чтобы каждый раздел (*Введение, Общие сведения, Типы сенсорных экранов, Заключение, Список использованных источников*) начинался с новой страницы;
- 8) оформите заголовки разделов стилем *Заголовок 1*; удалите ручную нумерацию и добавьте автоматическую нумерацию к разделам основной части (сделайте заголовки разделов элементами списка);
- 9) оформите заголовки подразделов стилем *Заголовок 2*; удалите ручную нумерацию, добавьте такую же автоматическую нумерацию (элементы списка второго уровня);
- 10) добавьте на отдельную страницу (с.2) автособираемое оглавление; слово *Оглавление* должно быть оформлено так же, как и заголовки стиля *Заголовок 1*;
- 11) преобразуйте готовый документ в формат PDF;
- 12) \*разработайте и сохраните шаблон для создания рефератов.

## **Практическая работа №17**

### **Коллективная работа над документами**

#### **Уровень А.**

1. Откройте файл **i-history.doc** (или **i-history.odt**).
2. Найдите в тексте фактические ошибки, создайте для каждой примечание, в примечании укажите правильный вариант и источник информации (адрес веб-страницы).

#### **Уровень В.**

3. Откройте файл **ggl.doc** (или **ggl.odt**).
4. Добавьте к тексту название.
5. Включите режим исправлений и исправьте все ошибки в тексте.

#### **Уровень С.**

6. Разбейтесь на группы по 3-4 человека. Задача группы – подготовить сообщение на какую-то тему в форме небольшого реферата или презентации.
7. Выберите руководителя, который будет организовывать работу группы.
8. Разделите работу между участниками: подбор материала по одному разделу сообщения, подбор иллюстраций, оформление всего документа в едином стиле и др.
9. Зарегистрируйтесь на сайте **accounts.google.com** (создайте для каждого учётные записи, аккаунты). Руководитель создаёт новый документ и предоставляет всем участникам право его редактировать.

## Практическая работа № 17. Знакомство с программой Scribus

Файлы для выполнения этой работы находятся в папке **Scribus**.

1. Запустите программу *Scribus*. Создайте новый односторонний документ, установите поля: левое – 25 мм, правое – 10 мм, верхнее и нижнее – 15 мм.
2. Используя текст из файла *Брусенцов.txt* и фотографию *Брусенцов.jpg*, выполните верстку материала на одной странице:
  - а) определите место для текстовых блоков; центральный блок должен состоять из двух колонок с интервалом 6 мм между ними (**F2 – Текст**);
  - б) используя окно *Свойства* (**F2**), установите шрифт для заголовка и двух блоков, выделенных фоном;
  - в) выберите размер шрифта для каждого блока (**F2 – Текст**);
  - г) создайте новый стиль *СтильТекста* (**F3 – Создать**):
    - шрифт *Arial*, 10 пт
    - дополнительный интервал после абзаца 2 пт;
    - абзацный отступ 10 мм;и присвойте его всем абзацам основного текста (**Ctrl+T** – редактирование текста);
  - д) добавьте в набор цветов плашечные цвета для блоков с цветным фоном; например, можно выбрать цвета A210 и A403 из набора AFNOX; установите цвета фона и отступы для этих блоков (**F2 – Цвета**);
  - е) добавьте фотографию и установите для нее масштабирование по размерам блока (**F2 – Масштабировать до размера блока**);
  - ж) установите для фотографии обтекание текстом по контурной линии; измените контурную линию так, чтобы между текстом и фотографией был некоторый интервал (**F2 – Фигура – По контурной линии – Изменить**);
  - з) включите переносы для центрального блока: в настройках выберите переносы для русского языка (**Файл – Настроить Scribus – Перенос слов**), а затем используйте верхнее меню *Расширения – Вставить переносы*.
3. Экспортируйте документ в формат PDF.

### Дополнительные материалы:

- Уроки по Scribus: <http://rus-linux.net/MyLDP/BOOKS/scribus/Izuchaem-Scribus-01.html>
- Вводный курс по Scribus [http://wiki.scribus.net/canvas/%D0%92%D0%B2%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9%D0%BA%D1%83%D1%80%D1%81%D0%BF%D0%BE\\_Scribus](http://wiki.scribus.net/canvas/%D0%92%D0%B2%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9%D0%BA%D1%83%D1%80%D1%81%D0%BF%D0%BE_Scribus)
- Видеоуроки по Scribus <https://www.youtube.com/watch?v=X8qXUW2AJWI&list=PLAngObgqXBVXdXtNdm9HA2j09G6A-lSse>

### Николай Брусенцов

Николай Петрович Брусенцов впервые в мире создал троичный компьютер «Сетунь», который серийно выпускался в СССР. Интерес к этим идеям сохраняется и в наши дни.

В 50-е годы XX века было написано много статей о тех или иных системах счисления и их использовании в вычислительной технике. Причем для их оценки рассматривались различные критерийные подходы. Один из критериев связан с экономичностью системы счисления. Под этим понимается тот запас чисел, которые можно записать в данной системе с помощью определенного количества знаков.

Математически было доказано, что самой экономичной системой счисления является система с основанием

$$e = 2,71828...$$

(основание натурального логарифма). Ближайшим к этому иррациональному числу является число 3, т. е. троичная система — самая экономичная. Но «такая простота» — как писал в те годы Брусенцов, — троичного представления чисел перед принятием в современных компьютерах двоичного кода, а в том, что с тремя цифрами возможен натуральный код чисел со знаком, а с двумя — невозможность. Недостатком двоичной арифметики и реализующих ее цифровых машин обусловлено именно тем, что двоичным кодом естественно представляемы либо только неотрицательные числа, либо только неоложительно, а для представления всей необходимой для арифметики совокупности — положительных, отрицательных и нуля — приходится пользоваться искусственными приемами типа прямого, обратного или дополнительного кода, системой с отрицательным основанием или цифрами +1, -1 и другими ухищрениями.

Брусенцов начал работу над малой ЭВМ в 1958 году в лаборатории при вычислительном центре МГУ, где работали первые университетские программисты (М.Р. Шур-Бура, К.А. Семенов, Е.А. Жоголев) и заведующий кафедрой вычислительной математики ректора МГУ Сергей Львович Соболев. Было принято решение строить машину не на магнитных элементах.

Размышления о недостатках существующих машин привели Брусенцова к мысли об использовании троичной системы счисления. Вот что он пишет: «Оказалось, что эти элементы не только весьма удобны для построения троичных цифровых устройств. Троичные устройства получаются существенно более быстрыми и структурно более простыми, чем двоичные устройства, реализованные на тех же элементах».

Назвать новую машину решили по имени реки, протекавшей недалеко от университета — «Сетунь». Наверно, такая необычная машина могла родиться только в университетских стенах. Своей простотой и практичностью «Сетунь» обязана представлению чисел и команд в симметричном коде: (-1, 0, 1). По существу, у университетских разработчиков получился первый RIS-компьютер: длина машинного слова — 9 триплов, всего 24 команды, при этом ей удавалось с большой эффективностью реализовать разнообразие алгоритмов. На «Сетунь» решались задачи математического моделирования в физике и химии, оптимизация управления производством, краткосрочных прогнозов погоды, конструкторских расчетов, компьютерного обучения, обработки экспериментальных данных и т.д.

Серийное производство «Сетуни» было поручено Казанскому заводу математических машин. Завод произвел 15–20 машин в год, всего было выпущено 50 машин, 30 из которых работали в вузах страны.

В 1961–1968 годах Брусенцов вместе с коллегами разработали новую машину, впоследствии названную «Сетунь-70». Действующий образец прошел испытания в апреле 1970 года. Однако затем лаборатория Брусенцова была вынуждена по указанию нового начальства прекратить разработку машин.

По материалам сайта <http://chernykh.net>





## Практическая работа № 18.

### Знакомство со средой Scilab

Свободно распространяемая среда *Scilab* ([www.scilab.org](http://www.scilab.org)) предназначена для выполнения научных и инженерных расчётов.

#### Где взять Scilab?

Инсталлятор Scilab для различных операционных систем можно скачать с официального сайта ([www.scilab.org/download/](http://www.scilab.org/download/)). Там же можно скачать и её исходные коды.

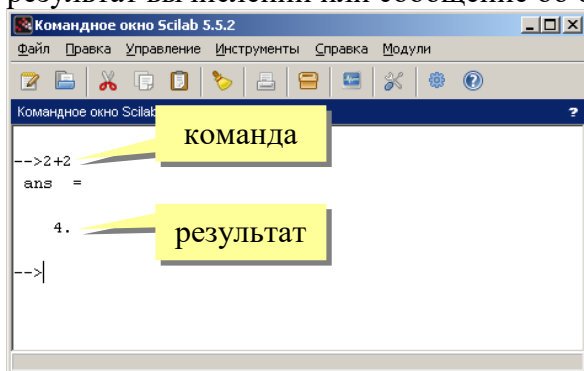
Переносимую (не требующую установки) версию 5.5.2 для Windows можно загрузить по ссылке [https://yadi.sk/d/\\_vBiEzaF-5vImQ](https://yadi.sk/d/_vBiEzaF-5vImQ). После распаковки архива нужно запускать файл **WScilex.exe** из подкаталога **bin**.

Можно использовать также облачную платформу [cloud.scilab.in](http://cloud.scilab.in), в которой работает Scilab.

#### Два режима работы Scilab


Scilab – это интерпретатор команд, который может работать в двух режимах:

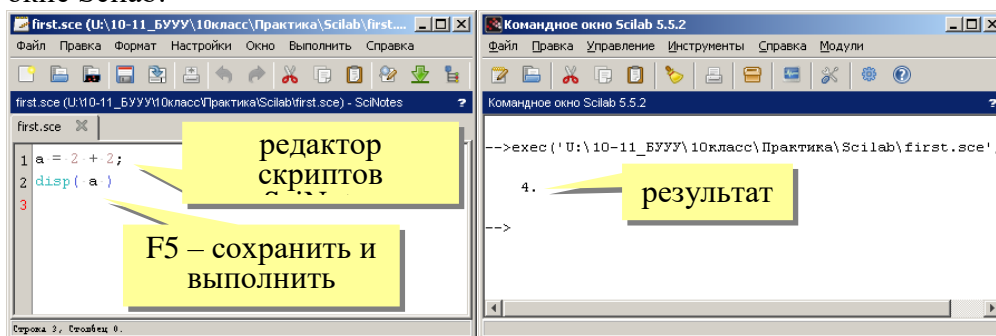
- **режим командной строки:** команды вводятся в специальном окне, которое называется *консолью*, и сразу выполняются; в этом же окне система выводит ответ – результат вычислений или сообщение об ошибке:



В режиме командной строки удобно использовать Scilab как мощный калькулятор.

- **программный режим:** пользователь составляет программу (скрипт), сохраняет её в файле и затем запускает файл на выполнение. Преимущество этого метода состоит в том, что в случае ошибки программу можно легко исправить, не набирая заново все команды. Scilab поддерживает простой язык программирования, который позволяет использовать переменные, циклы, условные вычисления, функции.

Чтобы открыть окно редактора скриптов, нужно щелкнуть по кнопке  в главном окне Scilab:



В редакторе нужно набрать последовательность команд, которые нужно выполнить, сохранить скрипт в виде файла и запустить, нажав клавишу F5. Результаты вычислений в таком режиме не выводятся автоматически в окно консоли; чтобы вывести значение переменной *a* нужно использовать команду

**disp ( a )**

Для выполнения практической работы достаточно использовать командный режим.

#### Уровень А.

9. Используя Scilab как калькулятор в режиме командной строки, вычислите значение

выражения  $\sqrt{1 + \cos(3,53\pi)} \cdot 10 \cdot 310$ . Для вычисления квадратного корня используйте встроенную функцию **sqrt**, число  $\pi$  обозначается как **%pi**. Запишите ответ с точностью 0,001 (оставив 3 знака в дробной части числа).

Ответ:

10. Функция **dec2base** переводит число в систему счисления с заданным основанием. Например, перевести число 100 в систему счисления с основанием 8 можно так:

--> **dec2base(100,8)**

Переведите число 567 в двоичную систему счисления:

Ответ:

Переведите число 259 в систему счисления с основанием 20:

Ответ:

11. Функция **base2dec** переводит число, записанное в системе счисления с заданным основанием, в десятичную систему счисления. Например, перевести число  $123_8$  в десятичную систему счисления можно так:

--> **base2dec("123", 8)**

Обратите внимание, что первый аргумент – символьная строка в кавычках (это связано с тем, что в записи чисел в некоторых системах счисления могут быть латинские буквы).

Переведите число  $1101010000101_2$  в десятичную систему счисления:

Ответ:

Переведите число  $FEDCBA_{16}$  в десятичную систему счисления:

Ответ:

### Уровень В.

12. Для построения графика функции  $y = f(x)$  нужно построить массив значений  $x$ , затем – массив соответствующих значений  $y$ , а затем вызвать встроенную функцию **plot**. Для примера построим график функции  $y = 0,5 \cdot x \cdot \sin(2x)$  на отрезке  $[0; 10]$ . Сначала создаём массив (набор значений) независимой переменной  $x$ : от 0 до 10 с шагом 0,01:

--> **x = 0:0.01:10**

После ввода этой команды вы увидите на экране весь массив значений  $x$ . Чтобы не выводить его, в конце команды нужно поставить точку с запятой.

Затем строим массив значений  $y$ :

--> **y = 0.5\*x .\* sin(2\*x);**

Обратите внимание, что в Scilab можно умножать число на массив (**0.5\*x**), при этом получается новый массив того же размера, в котором все элементы исходного массива умножаются на указанное число. Также можно вызывать функции для массива (**sin(2\*x)**), при этом также получается новый массив, в котором каждый элемент умножается на 2 и для результата вычисляется функция **sin**.

Дальше нужно перемножить элементы двух массивов – **0.5\*x** и **sin(2\*x)**. Эта операция (*поэлементное умножение*) обозначается двумя знаками «.\*». Первый элемент первого массива умножается на первый элемент второго и т.д.

Теперь строим график:

--> **plot(x, y)**

Наконец, добавим на график ось OX как отрезок, соединяющий точки (0, 0) и (10, 0)

--> **plot([0 10], [0 0], 'black')**

Последний аргумент функции – это цвет линии (англ. *black* – чёрный).

Выполните эти команды в Scilab и вставьте в отчёт полученный график функции  $y = 0,5 \cdot x \cdot \sin(2x)$  :

График:

13. Используя средства Scilab, постройте график функции  $f(x) = 2 \sin(4x) + 5 \cos(7x)$  на отрезке  $[0; 3]$  и вставьте его в отчёт.

График:

Определите, сколько решений имеет уравнение  $2 \sin(4x) + 5 \cos(7x) = 0$  на отрезке  $[0,5; 1,8]$ ?

Ответ:

### Уровень С.

14. Среда Scilab умеет выполнять операции с полиномами (многочленами). Сначала создадим базовую переменную, например,  $x$ :

```
--> x = poly(0, "x")
```

Первый аргумент функции **poly** – массив корней нового полинома, второй – имя переменной. Таким образом, приведённый выше вызов создаёт полином с единственным корнем 0 и именем переменной  $x$ . Теперь можно строить полиномы от этой переменной:

```
--> p1 = x - 5
```

```
--> p2 = x^2 - 3*x + 2
```

выполнять с ними операции (складывать, вычитать, умножать):

```
--> p = p1*p2 + 4
```

находить корни полинома (значения  $x$ , при которых он равен 0) с помощью функции **roots**:

```
--> r = roots(p)
```

Выполните приведённые выше команды и запишите полином  $p$ , который у вас получился. Обратите внимание, что Scilab выводит полином в порядке возрастания степеней (от свободного члена к старшей степени  $x$ ), а в математике принят обратный порядок записи.

Запишите коэффициенты полинома вместо многоточий:

$$p(x) = \dots x^3 + \dots x^2 + \dots x + \dots$$

Вычислите корни этого полинома:

15. Найдите корни полинома  $7x^4 - 14x^3 - 7056x^2 - 16114x + 1232777$ :

Ответ:

16. В среде Scilab можно легко решать системы линейных уравнений. Покажем, как решить систему

$$\begin{cases} 2x + 3y = 73 \\ 5x - 4y = -13 \end{cases}$$

Из коэффициентов левых частей этих уравнений можно составить матрицу

(прямоугольную таблицу)  $A = \begin{bmatrix} 2 & 3 \\ 5 & -4 \end{bmatrix}$ , а правые части образуют массив (вектор)

$B = \begin{bmatrix} 73 \\ -13 \end{bmatrix}$ . Тогда исходную систему уравнений можно записать в матричном виде:

$$\begin{bmatrix} 2 & 3 \\ 5 & -4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 73 \\ -13 \end{bmatrix}$$

Например, для того, чтобы получить первое уравнение системы, первую строку матрицы  $A$  нужно поэлементно умножить на вектор (столбец) неизвестных.

В среде Scilab матрицы (а массив  $B$  – тоже матрица, только из одного столбца) записываются по строкам, строки отделяются друг от друга точкой с запятой:

```
--> A = [2 3;5 -4]
```

```
--> B = [73; -13]
```

Теперь столбец с неизвестными переменными  $x$  и  $y$  получается с помощью специального матричного деления:

```
--> A \ B
```

Выполните приведённые выше команды и найдите значения переменных  $x$  и  $y$ .

Ответ:

```
x = ..., y = ...
```

#### Уровень D.

17. Решите систему уравнений

$$\begin{cases} x + 3y + 7z = 118 \\ 2x - 4y + 9z = 76 \\ 5x + 8y - 6z = -47 \end{cases}$$

Ответ:

```
x = ..., y = ..., z = ...
```

18. Среда Scilab позволяет решать задачи оптимизации, то есть находить значения переменной (или нескольких переменных), при которых функция достигает минимума (или максимума). Самая простая функция из этой группы – **fminsearch** – ищет минимум заданной функции без дополнительных ограничений. Функция, минимум которой мы ищем, должна быть определена с помощью специальной команды **deff**:

```
--> deff('y=f(x)', ['y = x^2 + 3*x + 5;'])
```

Эта строчка определяет функцию  $f(x) = x^2 + 3x + 5$ , которая в среде Scilab получает имя **f**.

В скрипте та же самая функция может быть определена иначе:

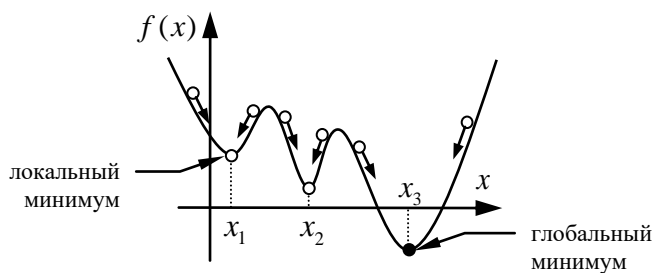
```
function y = f(x)
y = x^2 + 3*x + 5;
endfunction
```

Теперь осталось выполнить оптимизацию, вызвав встроенную функцию **fminsearch**:

```
--> [xmin, ymin] = fminsearch(f, 1)
```

Слева от знака равенства в квадратных скобках записаны имена двух переменных – это означает, что функция вернёт два значения – координаты точки минимума ( $x_{\min}$ ,  $y_{\min}$ ).

Первый аргумент функции **fminsearch** – это имя функции, минимум которой нужно найти. Поиск начинается со значения  $x = 1$  (это второй аргумент). Если минимумов несколько, то будет найдён тот, в который «скатится шарик», установленный на линии в начальной точке (см. рисунок ниже).



Такой минимум называется «локальным». Чтобы найти «самый глубокий» или «глобальный» минимум, нужно выбрать правильное начальное приближение.

Найдите координаты точки минимума функции  $f(x) = x^2 + 3x + 5$ :

Ответ:

$$x_{\min} = \dots, y_{\min} = \dots$$

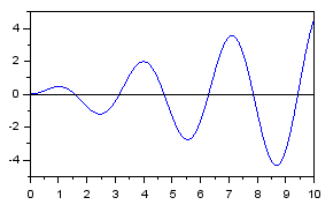
19. Найдите координаты точки глобального минимума функции  $f(x) = 0,5x \sin(2x)$  на отрезке  $[0; 10]$ :

Ответ:

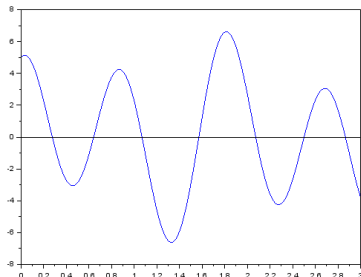
$$x_{\min} = \dots, y_{\min} = \dots$$

**Ответы:**

1. 431,901
2. 1000110111<sub>2</sub>, CJ<sub>20</sub>
3. 6789; 16702650
- 4.



5. 3 решения



6.  $x^3 - 8x^2 + 17x - 6$ ; [4.562; 3; 0.438]
7. [31; -23; -19; 13]
8.  $x = 11, y = 17$
9.  $x = -7, y = 9, z = 14$
10.  $x = -1.5, y = 2.75$
11.  $x = 8.668, y = -4.327$

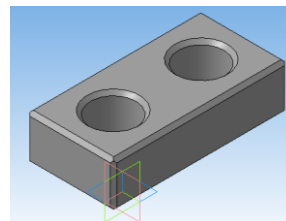
## Практическая работа № 19.


### 3D-моделирование в программе КОМПАС

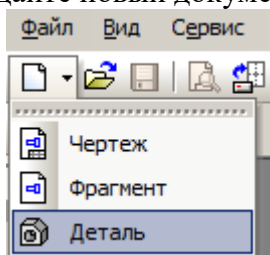
Программа КОМПАС 3D позволяет создавать трёхмерные модели деталей и затем автоматически строить по ним чертежи. В первой из серии работ мы будем строить трёхмерную модель кирпича, а в следующей – построим по ней чертёж и оформим его по современным стандартам.

Облегчённая версия программы КОМПАС 3D 12 LT – бесплатная, её можно использовать в учебных заведениях. Скачать КОМПАС 3D 12 LT можно на странице разработчиков КОМПАС 3D 12 LT <https://kompas.ru/kompas-3d-lt/download/> или по ссылке . Для изучения программы КОМПАС 3D 12 LT полезен видеокурс от разработчиков.

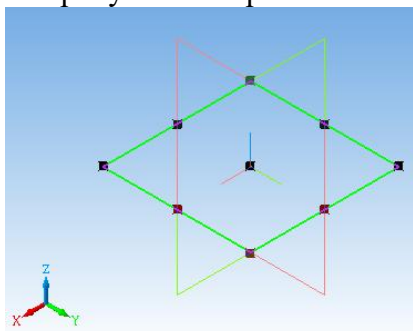
**Уровень А.** В этой работе мы построим трёхмерную модель кирпича с двумя отверстиями (см. рисунок справа).




1. Запустите программу КОМПАС с помощью ярлыка .
2. Создайте новый документ типа «Деталь»:



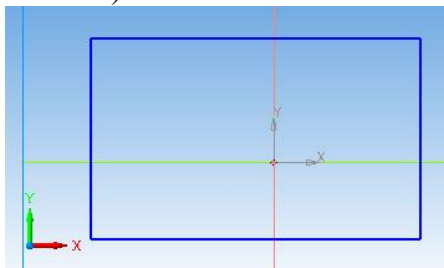
3. Сохраните документ под именем **Кирпич.m3d**.
4. На поле рисунка выберите плоскость XOY (с синим контуром).



5. Найдите на панели кнопку инструментов  и перейдите в режим **Эскиз**.
6. На **Компактной панели** (она по умолчанию располагается вертикально) выберите палитру **Геометрия**, а на ней – инструмент **Прямоугольник**:



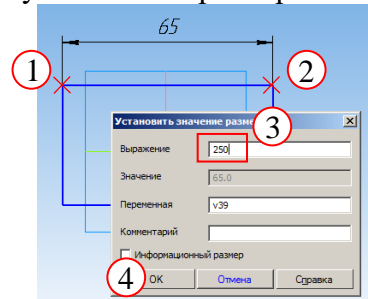
7. Нарисуйте на эскизе прямоугольник любого размера в любом месте. Он должен быть нарисован синей (основной) линией:





8. На **Компактной панели** выберите палитру **Размеры**, а на ней – кнопку **Линейный размер**:




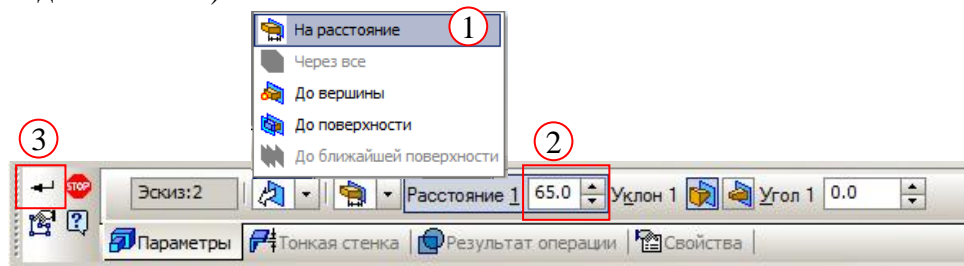
9. Определите ширину прямоугольника 250 мм: для этого сначала щёлкните в левом верхнем углу прямоугольника, затем – в правом верхнем, вытяните вверх размерную линию и щелкните мышью, чтобы её зафиксировать. При этом появляется окно, в котором нужно ввести размер 250.



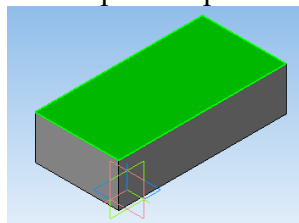
10. Щёлкните по кнопке  для того, чтобы установить масштаб, при котором видны все элементы чертежа.
11. После этого установите второй размер (высоту прямоугольника) 125 мм.
12. Выйдите из режима **Эскиз**, щёлкнув по кнопке . Вращая колесико мыши, отрегулируйте масштаб изображения.
13. Выберите инструмент **Выдавливание** на панели **Редактирование детали**:




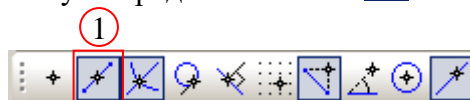
14. На панели свойств объекта (в нижней части экрана), убедитесь, что установлен режим выдавливания **На расстояние**, установите расстояние выдавливания 65 мм. Чтобы завершить операцию, нажмите клавиши **Ctrl+Enter** или щёлкните по кнопке  (**Создать объект**).



15. Выделите щелчком мыши верхнюю грань кирпича и перейдите в режим **Эскиз**.

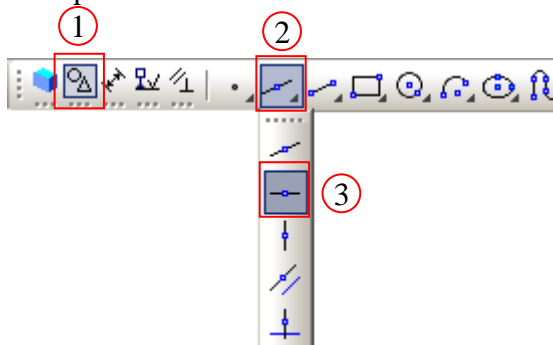


16. Выведите на экран панель инструментов **Глобальные привязки**, нажав правую кнопку мыши на любой панели инструментов и выбрав соответствующий пункт меню. Включите привязку к серединам линий :

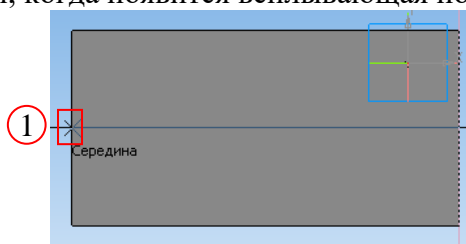


17. На **Компактной панели** выберите панель **Геометрия**, а на ней – кнопку **Вспомогательные линии**, из выпадающего меню выберите горизонтальную

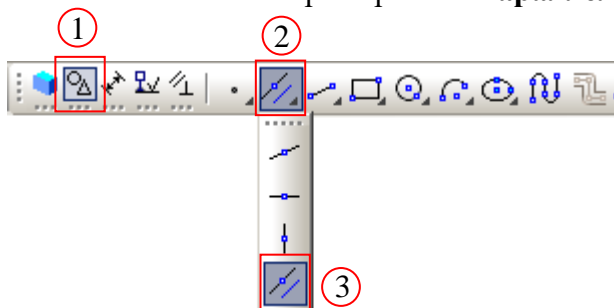
линию. Чтобы появилось выпадающее меню, нужно задержать мышь на кнопке панели инструментов при нажатой левой кнопке.



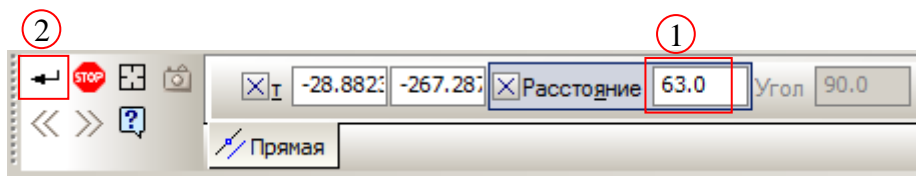
18. Подведите мышку к середине левой (или правой) границы прямоугольника и щёлкните левой кнопкой, когда появится всплывающая подсказка «Середина»:





19. Аналогично нарисуйте вертикальную среднюю линию для кирпича.  
20. В меню вспомогательных линий выберите режим **Параллельная линия**:

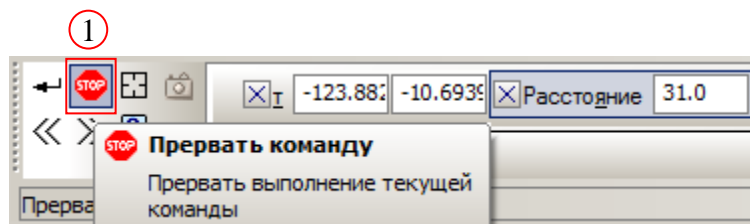


21. Щёлкните на вертикальной средней линии кирпича – так мы выбрали базовую линию. На панели свойств объекта установите расстояние (от базовой линии) 63 мм:



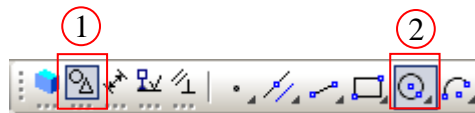
Чтобы завершить построение (построить ДВЕ линии, слева и справа на одинаковом расстоянии), нажмите ДВАЖДЫ клавиши **Ctrl+Enter** или щёлкните по кнопке  (Создать объект).

22. Закончите построение вспомогательных линий щёлкнув по кнопке  **Прервать команду**.

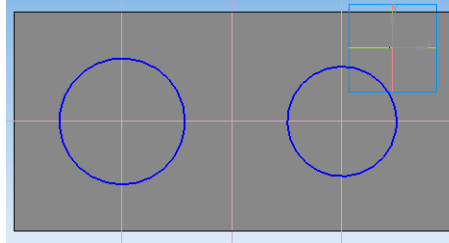


23. Включите инструмент **Окружность** на панели **Геометрия**:

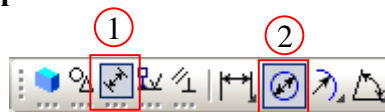




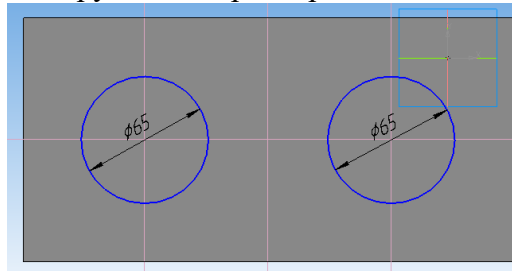
24. Нарисуйте две окружности (любого размера) с центрами в точках, полученных пересечением только что построенных вспомогательных прямых. Сначала нужно щёлкнуть в центре окружности, а затем – растянуть окружность и зафиксировать её вторым щелчком.



25. На **Компактной панели** выберите палитру **Размеры**, а на ней – кнопку **Диаметральный размер**:

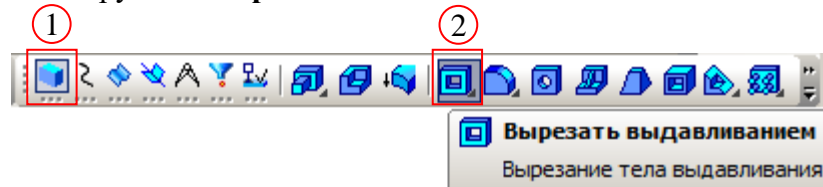


Установите для обеих окружностей размер 65 мм:

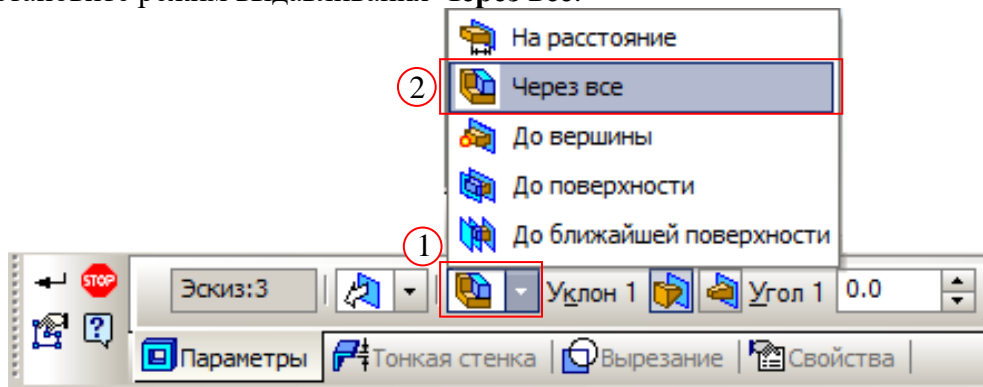



26. Выйдите из режима **Эскиз**, щёлкнув по кнопке . Сохраните модель.

27. Выберите инструмент **Вырезать выдавливанием**

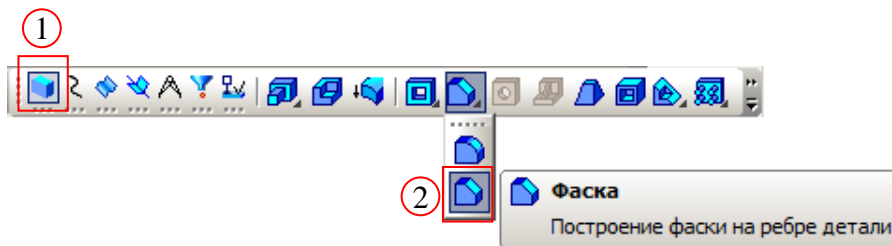


28. Установите режим выдавливания **Через все**:




Чтобы завершить операцию, нажмите клавиши **Ctrl+Enter** или щёлкните по кнопке  (**Создать объект**).

29. Включите инструмент **Фаска**:



30. Включите режим просмотра **Каркас**:

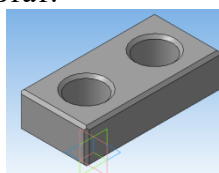


31. Выделите щелчками мыши все рёбра кирпича (включая входы отверстий). Чтобы завершить создание фасок, нажмите клавиши **Ctrl+Enter** или щёлкните по кнопке  (Создать объект).

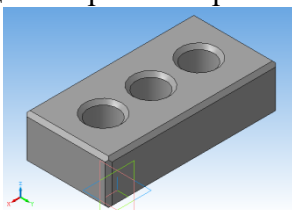
32. Включите режим изображения **Полутоновое**:



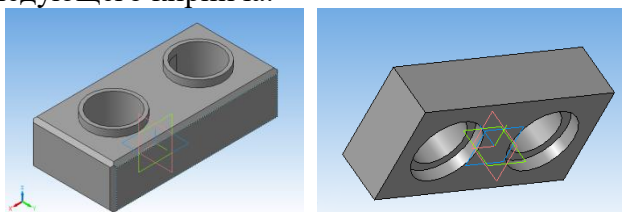
33. Сохраните окончательный результат.



**Уровень В.** Постройте трёхмерную модель кирпича с тремя отверстиями:



**Уровень С.** Постройте трёхмерную модель ЛЕГО-кирпича с двумя отверстиями. Выступы в верхней части кирпича должны точно подходить к выемкам в нижней части следующего кирпича:

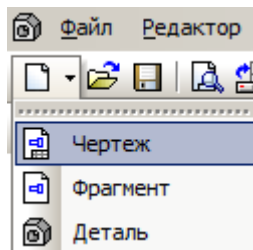


## Практическая работа № 20. Чертежи в программе КОМПАС

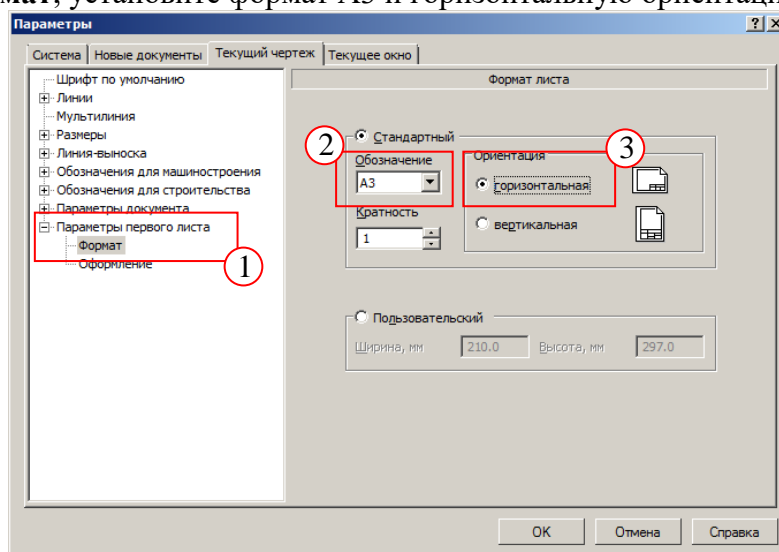
Во второй работе по программе КОМПАС наша задача – построить чертёж детали, для которой мы ранее (в предыдущей работе) создали трёхмерную модель.

### Уровень А. Часть 1. Создание чертежа.


1. Создайте новый файл типа **Чертёж**:

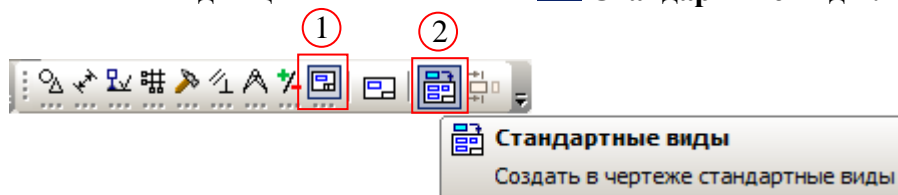


2. С помощью меню **Сервис-Параметры** откройте диалоговое окно и перейдите на вкладку **Текущий чертёж**. Выберите в левой части пункты **Параметры первого листа** – **Формат**, установите формат **A3** и горизонтальную ориентацию:

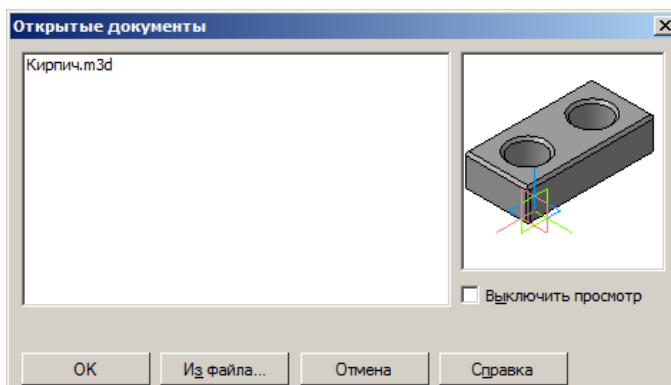


Выберите пункт меню **Окно – Показать закладки**, чтобы видеть открытые документы как вкладки.

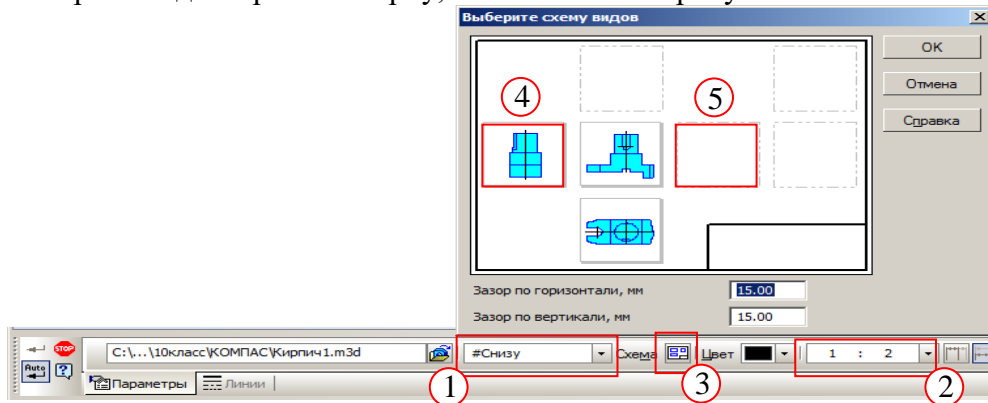
3. На панели **Виды** щёлкните по кнопке  **Стандартные виды**:



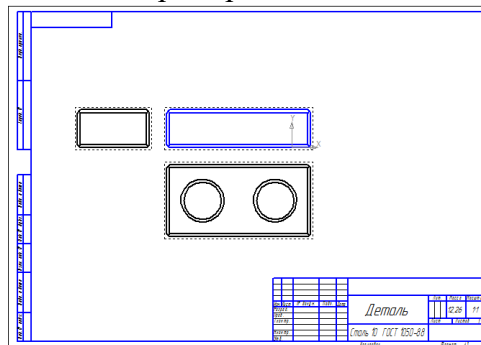
4. Выберите документ с трёхмерной моделью кирпича (он должен быть открыт) и щёлкните по кнопке **ОК** (файл может быть и закрыт, в этом случае нужно выбрать его на диске):



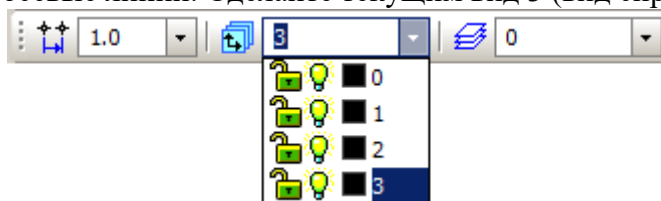
5. На панели свойств установите ориентацию главного вида #Снизу, масштаб 1:2 и выберите виды справа и сверху, как показано на рисунке:



6. Щелчком мыши установите виды примерно так, как показано на рисунке:



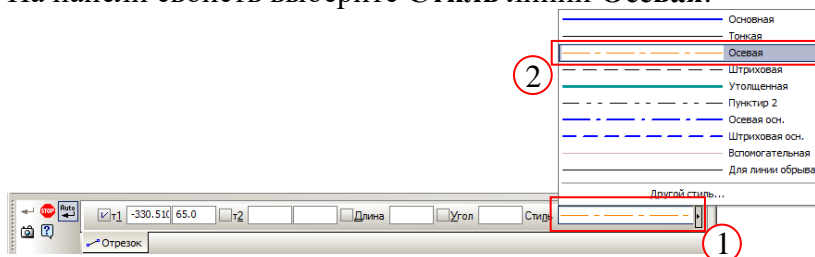
7. Попробуйте перемещать виды, выделяя их щелчком по штриховой рамке и перетаскивая мышью. Перемещение главного вида «тащит» за собой все остальные виды, вид справа перемещается только по горизонтали (по вертикали он связан с главным видом), а вид сверху – только по вертикали.
8. Теперь нарисуем осевые линии. Сделайте текущим вид 3 (вид справа):



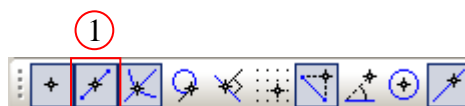
9. На **Компактной панели** выберите панель **Геометрия**, а на ней – инструмент **Отрезок**:



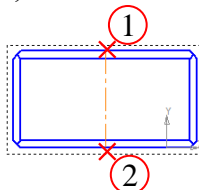
10. На панели свойств выберите **Стиль линии Осевая**:





11. На панели **Глобальные привязки** включите привязку к середине

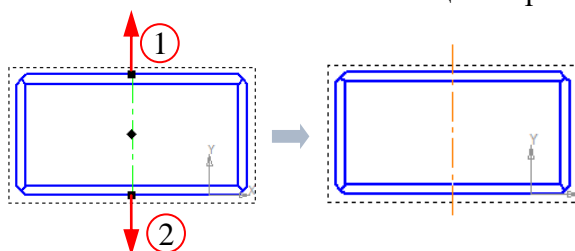


12. Проведите осевую линию на виде справа. Для этого щёлкните мышью сначала в середине верхней границы кирпича, а затем – в середине нижней границы.



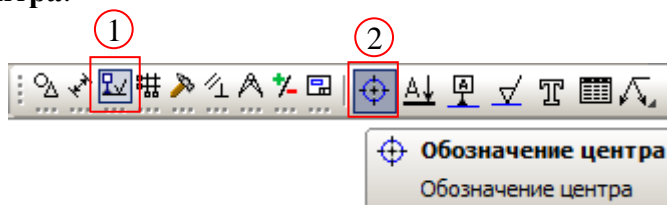
Отключите инструмент **Отрезок**, щелкнув по кнопке  или по кнопке  **Прервать команду**.

13. Выделите осевую щелчком мыши и вытяните её концы вверх и вниз:

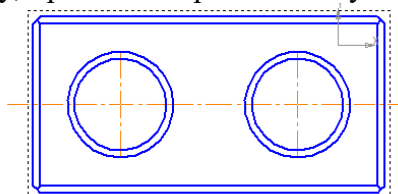


14. Сделайте активным вид 2 (вид сверху).

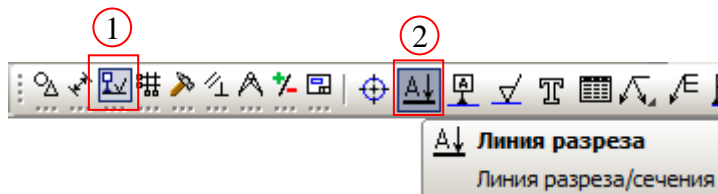
15. На **Компактной панели** включите панель **Обозначения** и выберите инструмент **Обозначение центра**:



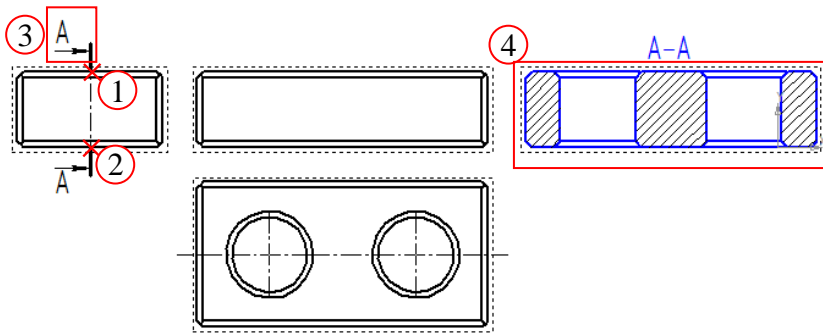
16. Чтобы обозначит центр окружности двумя осевыми линиями, нужно выделить щелчком окружность, а затем вторым щелчком зафиксировать оси. Обозначьте оси окружностей на виде сверху, продлите горизонтальную ось на всю длину кирпича.



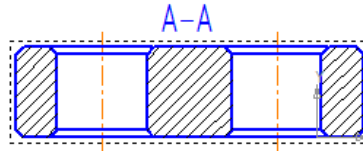
17. Снова сделайте активным вид справа и включите инструмент **Линия разреза** на панели **Обозначения**:



18. Обозначьте плоскость сечения щелчками мыши в середине верхней и нижней границ кирпича на виде справа, затем мышью установите направление взгляда справа налево, третьим щелчком зафиксируйте его. Появится рамка вида-разреза, которую нужно установить справа от главного вида:

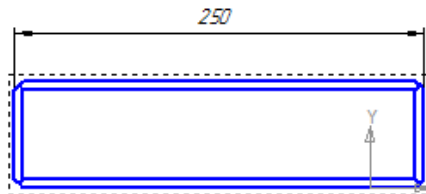


19. Добавьте осевые линии отверстий на разрезе:

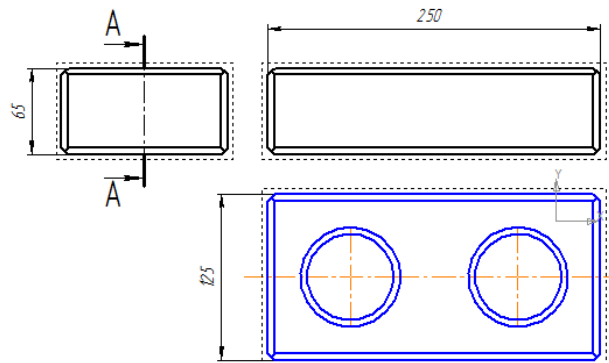


## Часть 2. Простановка размеров

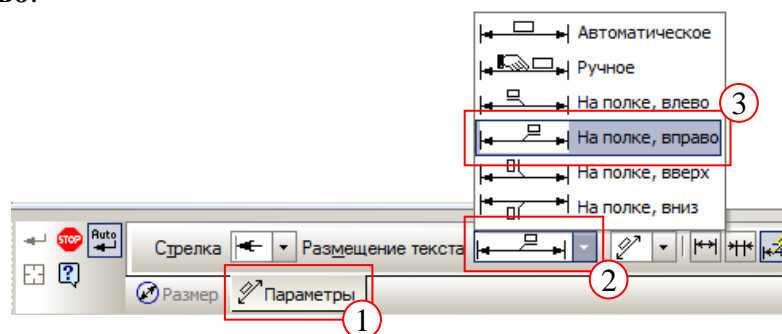
20. Сделайте активным вид 1 (главный вид) и проставьте линейный размер – длину кирпича:



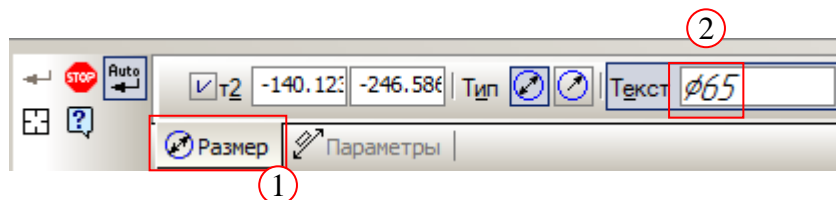
21. На виде справа укажите высоту кирпича, а на виде сверху – его длину:



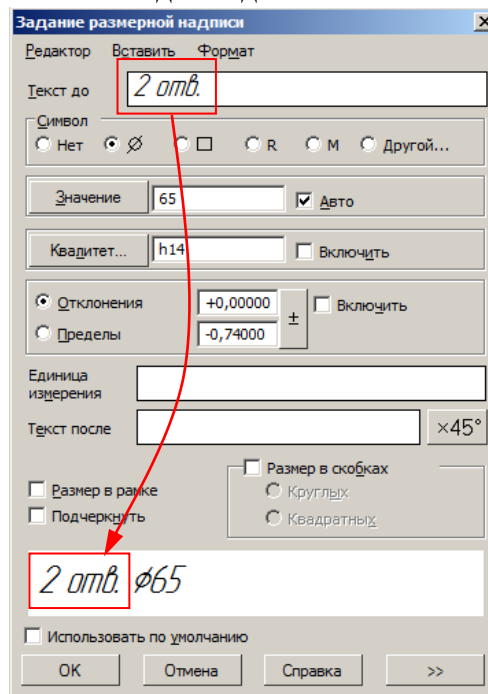
22. Включите инструмент **Диаметральный размер**, на панели свойств объекта перейдите на вкладку **Параметры** и выберите вариант размещения размера **На полке, вправо**:



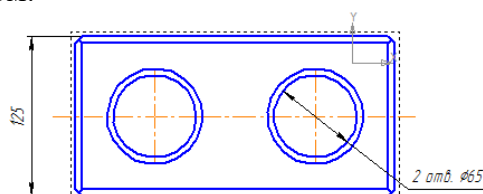
23. Выделите внутреннюю окружность, соответствующую отверстию. На панели свойств перейдите на вкладку **Размер** и щелкните по полю **Текст**.



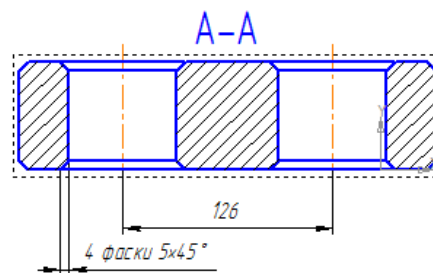
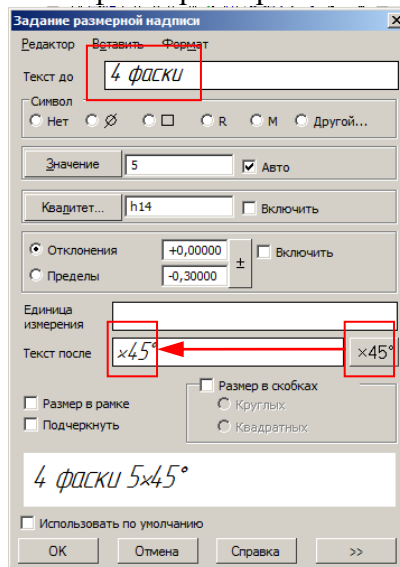
В появившемся окне в поле **Текст до** введите «2 отв. »:



24. Зафиксируйте щелчком мыши нужно положение выносной надписи с диаметральным размером:



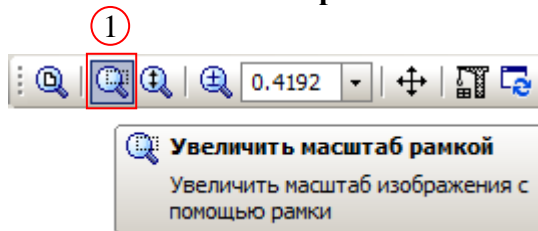
25. Сделайте активным вид-разрез (вид 4) и обозначьте расстояние между центрами отверстий и 4 фаски размером  $5 \times 45^\circ$ :



26. Аналогично обозначьте фаски на остальных видах.

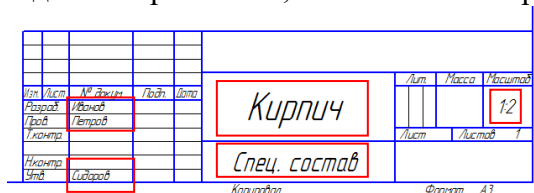
27. Выберите пункт меню **Вставка – Основная надпись**.


28. Включите инструмент **Увеличить масштаб рамкой**:



и выделите рамкой основную надпись, чтобы она занимала основную часть рабочего поля.

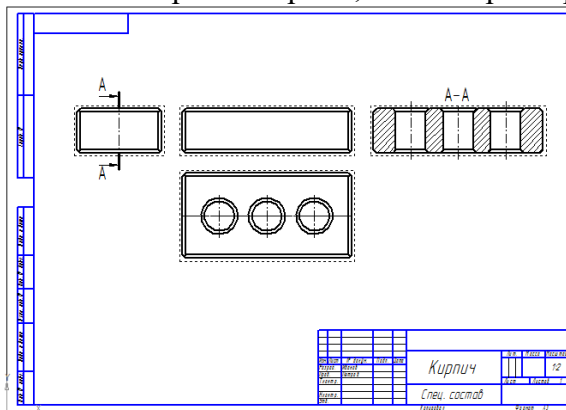
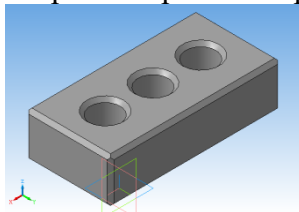
29. Заполните основную надпись чертежа так, как показано на рисунке:



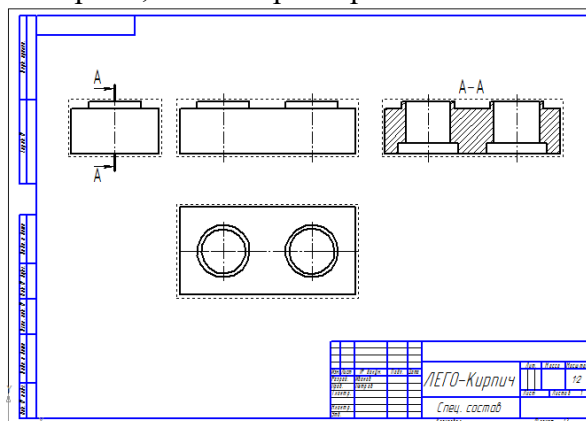
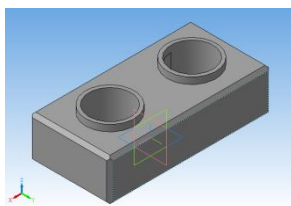
Для того, чтобы зафиксировать изменения, нажмите клавиши **Ctrl+Enter** или щёлкните по кнопке  (**Создать объект**).

30. Сохраните чертёж.

**Уровень В.** Кирпич с тремя отверстиями. Построить чертёж, нанести размеры.



**Уровень С.** Лего-кирпич. Построить чертёж, нанести размеры.





В этих двух работах изучаются пакеты прикладных программ, связанные с выбранной специализацией (по усмотрению учителя).

**Практическая работа № 21.**

**Пакеты прикладных программ по специализации**




**Практическая работа № 22.**

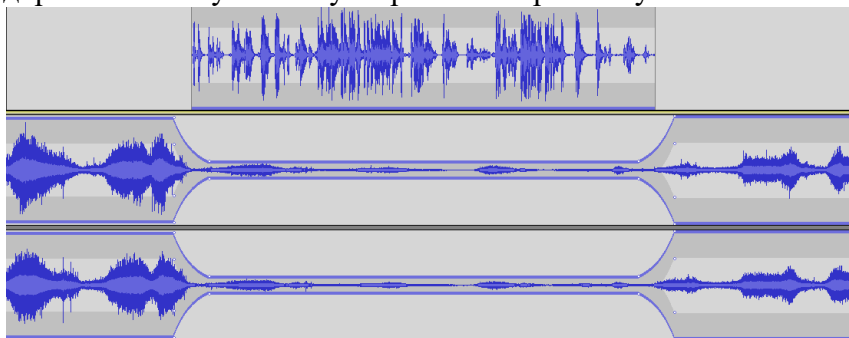
**Пакеты прикладных программ по специализации**

## Практическая работа № 23.

### Знакомство с аудиоредактором

Файлы для выполнения этой работы находятся в папке **Audacity**.

1. Запустите программу *Audacity* и загрузите проект **mouse\_click.aup**.
  - а) увеличивайте масштаб шкалы до тех пор, пока не будут видны отдельные отсчёты (точки);
  - б) включите инструмент *Карандаш*  и удалите щелчок, проводя в обоих каналах ровную линию около нулевой амплитуды;
  - в) экспортируйте полученный файл в формате WAV;
  - г) закройте проект, не сохраняя его.
2. Создайте новый проект и импортируйте файл **tsvetaeva.mp3** (Ольга Остроумова читает стихотворение Марины Цветаевой).
  - а) выделите отрезок записи в начале дорожки, где нет голоса (только шум) и постройте модель шума (*Эффекты – Удаление шума*)
  - б) выделите всю дорожку и удалите шум, используя только что построенную модель с параметрами по умолчанию;
  - в) удалите пустой интервал в начале записи и аплодисменты в конце дорожки;
  - г) преобразуйте дорожку из стерео в моно (*Дорожки – Стерео в моно*);
  - д) выполните нормализацию (*Эффекты – Нормализация* или в другой версии программы *Эффекты – Нормировка сигнала*) с максимальной амплитудой -1 дБ.
3. Импортируйте на вторую дорожку шум морского прибоя из файла **sea-surf.mp3**.
  - а) для того чтобы при прослушивании слышать только эту дорожку, включите для неё режим *Соло*;
  - б) удалите лишние части второй дорожки, оставив примерно на 30 секунд больше, чем на первой дорожке;
  - в) включите инструмент *Перемещение*  и переместите звук на первой дорожке примерно на середину второй дорожки;
  - г) включите инструмент *Изменение огибающей* , установите на огибающей второй дорожки контрольные точки слева и справа от обеих границ первой дорожки и заглушите шум прибоя на время звучания стихотворения:



- д) примените к началу второй дорожки эффект *Плавное нарастание*, а к концу этой дорожки – эффект *Плавное затухание*;
- е) экспортируйте полученный звук в файл **tsvetaeva128.mp3** (формат MP3 с битрейтом 128 Кбит/с); заполните тэги:
  - *Исполнитель*: Ольга Остроумова
  - *Название дорожки*: «Никто ничего не отнял!»
  - *Название альбома*: Стихи Марины Цветаевой
  - *Год*: 2011
- ж) экспортируйте полученный звук в файл **tsvetaeva32.mp3** (формат MP3 с битрейтом 32 Кбит/с); сравните размеры и качество полученных файлов; сделайте выводы;
- з) экспортируйте полученный звук в форматы WAV и Ogg (с параметрами по

умолчанию) и сравните размеры полученных файлов; прослушайте их и сравните качество звучания; сделайте выводы.

**Дополнительные материалы:**

- Учебник по программе *Audacity* <http://audacity.ru/p1aa1.html>
- Как пользоваться *Audacity* <http://itandlife.ru/technology/rabota-so-zvukom/kak-polzovatsya-audacity-gde-skachat-kak-nastroit-i-nachat-ispolzovat/>

## **Практическая работа № 24.**

### **Знакомство с видеоредактором**

*Задания сформулированы в самой общей форме и рассчитаны на видеоредакторы с возможностью нелинейного монтажа. В операционной системе Windows можно использовать онлайн-видеоредактор FileLab (<http://videoeditor.filelab.com/app/>) или бесплатный редактор VideoPad (<http://www.nchsoftware.com/videopad/ru/index.html>).*

#### **Часть 1. Слайд-фильм**

1. *Домашнее задание:* подготовьте 10-15 фотографий и звуковое сопровождение для слайд-фильма длительностью 1 минуту.
2. Используя видеоредактор, постройте слайд-фильм:
  - добавьте титульный и заключительный слайды;
  - выберите переходы между слайдами;
  - добавьте звуковую дорожку;
  - добавьте титры в конец фильма;
  - экспортируйте фильм в разные видеоформаты (WMV, AVI, MPG или др.) и сравните размер файлов.

#### **Часть 2. Клип**

1. *Домашнее задание:* подготовьте несколько видеороликов на выбранную вами тему (можно отснять их самостоятельно, например, с помощью цифрового фотоаппарата или видеокамеры); подберите звуковое сопровождение длительностью около 1 минуты.
2. Используя видеоредактор, создайте клип на выбранную тему:
  - обрежьте фрагменты и расположите их нужным образом;
  - добавьте титульный и заключительный слайды;
  - выберите переходы между фрагментами;
  - добавьте звуковую дорожку;
  - добавьте титры в конец фильма;
  - экспортируйте фильм в разные видеоформаты (WMV, AVI, MPG или др.) и сравните размер файлов.

## **Практическая работа № 25.**

### **Онлайн-сервисы для разработки презентаций**

Работа в группах по 3-4 человека. Используя онлайн-сервисы для разработки презентаций, например,

- презентации Google
- [prezi.com](https://prezi.com)
- [www.canva.com](https://www.canva.com)

разработайте презентацию на выбранную тему. Сделайте краткое сообщение перед классом с показом презентации. Обсудите достоинства и недостатки онлайн-сервисов для создания презентаций.

## Глава 7. Компьютерные сети

### Практические работы

#### Практическая работа № 26.

#### Сравнение поисковых систем

1. Изучите материалы сайта <http://analyzethis.ru/?lang=ru>. Найдите 4-5 параметров, по которым рейтинги поисковых систем *Яндекс* и *Google* отличаются сильнее всего.

Параметр	Яндекс	Google

Сделайте выводы:

--

2. Используя статистику поиска *Google* (<http://www.google.com/insights/search/?hl=ru>), ответьте на следующие вопросы:

- сравните рейтинги запросов «автомобиль» и «велосипед» в разных регионах (добавьте 5 регионов, где эти запросы наиболее популярны):

Регион	автомобиль	велосипед
Мир в целом		

- как вы думаете, почему популярность запроса «велосипед» имеет ярко выраженные колебания?
- найдите три самых популярных запроса со словами «автомобиль» и «велосипед»:

автомобиль	велосипед

3. Используя статистику поисковой системы *Яндекс* (<http://wordstat.yandex.ru/>), ответьте на следующие вопросы:

- найдите три самых популярных запроса со словами «автомобиль» и «велосипед»:

автомобиль	велосипед

- в каком месяце прошедшего года эти запросы были наиболее популярны:

запрос	месяц
автомобиль	
велосипед	

4. Познакомьтесь с поисковой системой *Google Академия* (<https://scholar.google.com>). Используя эту поисковую систему, найдите книги С.М. Окулова по информатике. Какие книги вам удалось найти:

1	
2	
3	

## Практическая работа № 27.

### Тестирование сети

20. Исследуйте свой компьютер и заполните таблицу:

Сетевое имя компьютера	
Рабочая группа (домен)	
IP-адрес	
Маска подсети	
Номер сети	
Номер компьютера в сети	
Шлюз	
Основной DNS-сервер	

21. С какими из этих компьютеров<sup>2</sup> есть связь (отметьте нужный вариант):

192.168.104.254	<input type="checkbox"/> Да	<input type="checkbox"/> Нет
192.168.104.1	<input type="checkbox"/> Да	<input type="checkbox"/> Нет
192.168.102.100	<input type="checkbox"/> Да	<input type="checkbox"/> Нет
192.168.102.12	<input type="checkbox"/> Да	<input type="checkbox"/> Нет
192.168.3.70	<input type="checkbox"/> Да	<input type="checkbox"/> Нет

22. Определите IP-адреса сайтов:

Сайт	IP-адрес
<a href="http://www.lenta.ru">www.lenta.ru</a>	
<a href="http://www.google.ru">www.google.ru</a>	
<a href="http://www.yale.edu">www.yale.edu</a>	
<a href="http://resh.edu.ru">resh.edu.ru</a>	
<a href="http://fcior.edu.ru">fcior.edu.ru</a>	

23. Определите, сколько «прыжков» до следующих сайтов:

Сайт	Число «прыжков»
<a href="http://www.ozon.ru">www.ozon.ru</a>	
<a href="http://www.yandex.ru">www.yandex.ru</a>	
<a href="http://www.utoronto.ca">www.utoronto.ca</a>	

24. Найдите в Интернете сайты Московского и Санкт-Петербургского государственного университетов и определите их IP-адреса

Сайт	IP-адрес

<sup>2</sup> Учитель должен изменить эти адреса локальной сети так, чтобы часть компьютеров конкретной локальной сети были доступны, а часть – нет.

**Практическая работа № 28.**  
**Информационные системы в Интернете**  
**Уровень А.**

- 1) Используя несколько сервисов прогноза погоды, сравните прогноз на сегодняшний день в вашем городе (посёлке):

Прогноз для:

<i>Адрес сервиса</i>	<i>Температура</i>	<i>Влажность</i>	<i>Скорость ветра</i>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

- 2) Варфоломей планирует лететь на самолете из Нижнего Новгорода в Анапу 25 июля, так чтобы 26 июля начать культурный отдых на море. Используя информационные системы в Интернете, предложите ему возможные варианты перелёта:

- а) **прямой рейс** (если есть):

вылет в  время в пути  авиакомпания

стоимость билета  рублей.

- б) **самый быстрый рейс с пересадкой:**

вылет в  пересадка в

время в пути  авиакомпания

стоимость билета  рублей.

- в) **самый дешёвый вариант (с пересадкой или без пересадки):**

вылет в  пересадка в

время в пути  авиакомпания

стоимость билета  рублей.

**Уровень В.**

- 3) Используя сервисы для работы с картами, проложите маршрут из школы домой и определите его длину.

Длина маршрута:

Вставьте в документ снимок экрана, на котором изображён маршрут.

**Уровень С.**

- 4) Узнайте, по какому адресу расположена школа № 146 г. Перми с углублённым изучением математики, физики и информатики:

г. Пермь,

Директор этой школы:

Вход в школу выглядит так (вставьте скриншот с сервиса Яндекс-панорамы):

- 5) Что нарисовано на торце дома под адресу: г. Уфа, ул. Гагарина, д. 14/1? Вставьте скриншот с сервиса Яндекс-панорамы.



## Практическая работа № 29.

### Работа с FTP-сервером

#### Уровень А.

- 1) Подключитесь к FTP-серверу <ftp://ftp.mcsme.ru>, перейдите в каталог **/pub/free-books/uspensky/**. Определите, какие файлы там находятся, какой размер они имеют и когда последний раз изменялись:

Файл	Размер файла	Дата изменения

- 2) Используя поисковую систему [www.filesearch.ru](http://www.filesearch.ru), найдите на FTP-серверах файлы **syntrace.zip** и **minimax.ps**, определите их адреса и размеры.

Файл	FTP-сервер (любой)	Размер файла
<b>syntrace.zip</b>		
<b>minimax.ps</b>		

#### Уровень В.

- 3) Подключитесь к FTP-серверу <ftp://ftp.ctan.org>, перейдите в каталог **/pub/tex/**. Откройте файл **FILES.byname** и найдите с его помощью, в каком каталоге на этом сервере следует искать файл **gfsartemisia.zip**, каковы его размер и дата последнего изменения.

Каталог	Размер файла	Дата изменения

- 4) Подключитесь к FTP-серверу пресс-службы Третьяковской галереи <ftp://press.tretyakov.ru/>. Найдите информацию об одной из ближайших выставок. Добавьте в отчёт краткую информацию об этой выставке и одну иллюстрацию работы, которая там будет выставляться:

Иллюстрация	Краткая информация о выставке

#### Уровень С.

- 5) Подключитесь к FTP-серверу <ftp://mirror.yandex.ru/>, на котором хранятся дистрибутивы (установочные пакеты) различных сборок операционной системы Linux. Определите номера последних версий, размеры и даты обновления образов установочных дисков (они имеют расширение **.iso**) для следующих дистрибутивов:

	Версия	Размер ISO-образа	Дата обновления
<b>AltLinux</b>			
<b>LinuxMint</b>			

## Практическая работа № 30.

### Электронная коммерция

#### Уровень А.

- 1) Найдите в сети адреса онлайн-сервисов Сбербанка и ещё двух любых банков:

Банк	Адрес онлайн-сервиса
Сбербанк	

- 2) Зайдите на сайт [www.ozon.ru](http://www.ozon.ru) и найдите любой товар, который вы хотели бы приобрести. Узнайте его цену, условия оплаты и условия доставки в ваш город или посёлок.

Товар	
Веб-страница	
Цена	
Варианты оплаты	
Условия доставки	

#### Уровень В.

- 3) Используя сервис Яндекс.Маркет, найдите другие интернет-магазины, где можно купить тот же товар, и сравните условия 3 лучших варианта по общей цене:

Интернет-магазин			
Веб-сайт			
Цена товара			
Цена доставки			
Цена с доставкой			

Выберите оптимальный (самый лучший) вариант.

#### Уровень С.

- 4) Используя сервис рекламных объявлений [www.avito.ru](http://www.avito.ru), найдите другие варианты покупки того же товара. Выберите 2-3 лучших варианта

	Состояние товара	Цена	Условия доставки
1			
2			
3			

- 5) Какие риски могут возникнуть при покупке товаров по объявлению?

Ответ:

--

#### Уровень D+.

- 6) Если у вас есть знакомый, который занимается предпринимательской деятельностью, вы можете вместе с ним разместить рекламу на сайте ВКонтакте, используя, например, инструкцию со страницы <http://online-vkontakte.ru/2016/10/kak-razmestit-reklamu-v-gruppe-vkontakte.html>.

## Глава 8. Алгоритмизация и программирование

### Практические работы

*«Каждый из нас лишь выиграет, создавая время от времени "игрушечные программы" с заданными искусственными ограничениями, заставляющими нас до предела напрягать свои способности... Искусство решения мини-задач на пределе своих возможностей оттачивает наше умение для реальных задач».*

**Д. Кнут.**

#### Практическая работа № 31.

##### Знакомство со средой программирования

25. Вывести на экран фразу лесенкой:

```
Вася
  пошёл
    гулять.
```

26. Вывести на экран изображение домика из букв:

```
  А
 АМА
АМММА
АМММММА
МММММ
 ЖЖ ЖЖ
МММММ
```

27. Вывести на экран изображение двух домиков:

```
  А  А
 АМА АМА
АМММА АМММА
АМММММА АМММММА
МММММ МММММ
 ЖЖ ЖЖ ЖЖ ЖЖ
МММММ МММММ
```

## Практическая работа № 32.

### Вычисления

**Уровень А.** Ввести с клавиатуры три целых числа, найти их сумму, произведение и среднее арифметическое. Пример:

Введите три целых числа:

5 7 8

$$5+7+8=20$$

$$5*7*8=280$$

$$(5+7+8)/3=6.667$$

**Уровень В.** Ввести с клавиатуры координаты двух точек (А и В) на плоскости (вещественные числа). Вычислить длину отрезка АВ. Пример:

Введите координаты точки А:

5.5 3.5

Введите координаты точки В:

1.5 2

$$\text{Длина отрезка АВ} = 4.272$$

**Уровень С.** Получить случайное трехзначное число и вывести через запятую его отдельные цифры. Пример:

Получено число 123.

Его цифры 1, 2, 3.

### Практическая работа № 33.

#### Случайные числа

**Уровень А.** Игральный кубик бросается три раза (выпадает три случайных значения).

Сколько очков в среднем выпало? Пример:

Выпало очков:

5 3 1

$$(5+3+1)/3=3$$

**Уровень В.** Игральный кубик бросается три раза (выпадает три случайных значения). Из этих чисел составляется целое число, программа должна найти его квадрат. Пример:

Выпало очков:

1 2 3

Число 123

Его квадрат 15129

**Уровень С.** Получить случайное трёхзначное число и вывести через запятую его отдельные цифры. Пример:

Получено число 123

сотни: 1

десятки: 2

единицы: 3

### **Практическая работа № 34.**

#### **Ветвления**

**Уровень А.** Ввести три целых числа, найти максимальное из них. Пример:

Введите три целых числа:

1 5 4

Максимальное число 5

**Уровень В.** Ввести пять целых чисел, найти максимальное из них. Пример:

Введите пять целых чисел:

1 5 4 3 2

Максимальное число 5

**Уровень С.** Ввести последовательно возраст Антона, Бориса и Виктора. Определить, кто из них старше. Пример:

Возраст Антона: 15

Возраст Бориса: 17

Возраст Виктора: 16

Ответ: Борис старше всех.

Пример:

Возраст Антона: 17

Возраст Бориса: 17

Возраст Виктора: 16

Ответ: Антон и Борис старше Виктора.

## Практическая работа № 35.

### Сложные условия

**Уровень А.** Напишите программу, которая получает три числа и выводит количество одинаковых чисел в этой цепочке.

Пример:

Введите три числа:

5 5 5

Все числа одинаковые.

Пример:

Введите три числа:

5 7 5

Два числа одинаковые.

Пример:

Введите три числа:

5 7 8

Нет одинаковых чисел.

**Уровень В.** Напишите программу, которая получает номер месяца и выводит соответствующее ему время года или сообщение об ошибке.

Пример:

Введите номер месяца:

5

Весна.

Пример:

Введите номер месяца:

15

Неверный номер месяца.

**Уровень С.** Напишите программу, которая получает возраст человека (целое число, не превышающее 120) и выводит этот возраст со словом «год», «года» или «лет». Например, «21 год», «22 года», «25 лет».

Пример:

Введите возраст: 18

Вам 18 лет.

Пример:

Введите возраст: 21

Вам 21 год.

Пример:

Введите возраст: 22

Вам 22 года.

## Практическая работа № 36.

### Циклические алгоритмы-1

**Уровень А.** Напишите программу, которая получает два целых числа А и В ( $0 < A < B$ ) и выводит квадраты всех натуральных чисел в интервале от А до В.

Пример:

Введите два целых числа:

10 12

$10*10=100$

$11*11=121$

$12*12=144$

**Уровень В.** Напишите программу, которая получает два целых числа и находит их произведение, не используя операцию умножения. Учтите, что числа могут быть отрицательными.

Пример:

Введите два числа:

10 -15

$10*(-15)=-150$

**Уровень С.** Ввести натуральное число N и вычислить сумму всех чисел Фибоначчи, меньших N. Предусмотрите защиту от ввода отрицательного числа N.

Пример:

Введите число N:

10000

Сумма 17710



## **Практическая работа № 44а**

### **Циклические алгоритмы – 2**

**Уровень А.** Ввести натуральное число и найти сумму его цифр.

Пример:

Введите натуральное число:

12345

Сумма цифр 15.

**Уровень В.** Ввести натуральное число и определить, верно ли, что в его записи есть две одинаковые цифры, стоящие рядом.

Пример:

Введите натуральное число:

12342

Нет.

Пример:

Введите натуральное число:

12245

Да.

**Уровень С.** Ввести натуральное число и определить, верно ли, что в его записи есть две одинаковые цифры (не обязательно стоящие рядом).

Пример:

Введите натуральное число:

12342

Да.

Пример:

Введите натуральное число:

12345

Нет.

## Практическая работа № 37.

### Циклы по переменной

**Уровень А.** Найдите все пятизначные числа, которые при делении на 133 дают в остатке 125, а при делении на 134 дают в остатке 111.

**Уровень В.** Натуральное число называется *числом Армстронга*, если сумма цифр числа, возведенных в N-ную степень (где N – количество цифр в числе) равна самому числу.

Например,

$153 = 1^3 + 5^3 + 3^3$ . Найдите все трёхзначные Армстронга.

**Уровень С.** Натуральное число называется *автоморфным*, если оно равно последним цифрам своего квадрата. Например,  $25^2 = 625$ . Напишите программу, которая получает натуральное число N и выводит на экран все автоморфные числа, не превосходящие N.

Пример:

Введите N:

1000

1\*1=1

5\*5=25

6\*6=36

25\*25=625

76\*76=5776

## Практическая работа № 45а

### Вложенные циклы

**Уровень А.** В магазине продается мастика в ящиках по 15 кг, 17 кг, 21 кг. Как купить ровно 185 кг мастики, не вскрывая ящики? Сколькими способами можно это сделать?

**Уровень В.** Ввести натуральное число  $N$  и вывести все натуральные числа, не превосходящие  $N$  и делящиеся на каждую из своих цифр.

Пример:

Введите  $N$ :

15

1 2 3 4 5 6 7 8 9 11 12 15

**Уровень С.** Напишите программу, которая получает натуральные числа  $A$  и  $B$  ( $A < B$ ) и выводит все простые числа на отрезке  $[A; B]$ .

Пример:

Введите границы диапазона:

10 20

11 13 17 19

**Уровень D.** Напишите программу, которая получает натуральные числа  $A$  и  $B$  ( $A < B$ ) и выводит все на отрезке  $[A; B]$ , у которых ровно 4 разных делителя.

Пример:

Введите границы диапазона:

10 20

10 14 15

## Практическая работа № 38.

### Процедуры

**Уровень А. 7.** Напишите процедуру, которая принимает параметр – натуральное число  $N$  – и выводит на экран линию из  $N$  символов '-'.  
Пример:

Введите N:  
10  
-----

**Уровень В.** Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с первой.

Пример:  
Введите натуральное число:  
1234  
1  
2  
3  
4

**Уровень С.** Напишите процедуру, которая выводит на экран запись переданного ей числа в римской системе счисления.

Пример:  
Введите натуральное число:  
2013  
ММХІІІ

## Практическая работа № 39.

### Процедуры-2

**Уровень А.** Напишите процедуру, которая переставляет три переданные ей числа в порядке возрастания.

Пример:

Введите три натуральных числа:

10 15 5

5 10 15

**Уровень В.** Напишите процедуру, которая сокращает дробь вида  $M/N$ . Числитель и знаменатель дроби передаются как изменяемые параметры. Процедура возвращает пару чисел (кортеж) – числитель и знаменатель сокращённой дроби.

Пример:

Введите числитель и знаменатель дроби:

25 15

После сокращения:  $5/3$

**Уровень С.** Напишите процедуру, которая возвращает пару чисел (кортеж): наибольший общий делитель и наименьшее общее кратное двух натуральных чисел.

Пример:

Введите два натуральных числа:

10 15

$\text{НОД}(10,15)=5$

$\text{НОК}(10,15)=30$

## Практическая работа № 40.

### Функции

**Уровень А.** Напишите функцию, которая определяет сумму цифр переданного ей числа.

Пример:

Введите натуральное число:

123

Сумма цифр числа 123 равна 6.

**Уровень В.** Напишите функцию, которая находит наибольший общий делитель двух натуральных чисел.

Пример:

Введите два натуральных числа:

7006652 112307574

$\text{НОД}(7006652, 112307574) = 1234.$

**Уровень С.** Напишите функцию, которая «переворачивает» число, то есть возвращает число, в котором цифры стоят в обратном порядке.

Пример:

Введите натуральное число:

1234

После переворота: 4321.

## Практическая работа № 41.

### Логические функции

**Уровень А.** Напишите логическую функцию, которая определяет, является ли переданное ей число совершенным, то есть, равно ли оно сумме своих делителей, меньших его самого.

Пример:

Введите натуральное число:

28

Число 28 совершенное.

Пример:

Введите натуральное число:

29

Число 29 не совершенное.

**Уровень В.** Напишите логическую функцию, которая определяет, являются ли два переданные ей числа взаимно простыми, то есть, не имеющими общих делителей, кроме 1.

Пример:

Введите два натуральных числа:

28 15

Числа 28 и 15 взаимно простые.

Пример:

Введите два натуральных числа:

28 16

Числа 28 и 16 не взаимно простые.

**Уровень С.** Простое число называется гиперпростым, если любое число, получающееся из него откидыванием нескольких цифр, тоже является простым. Например, число 733 – гиперпростое, так как и оно само, и числа 73 и 7 – простые. Напишите логическую функцию, которая определяет, верно ли, что переданное ей число – гиперпростое. Используйте уже готовую функцию **isPrime**, которая приведена в учебнике.

Пример:

Введите натуральное число:

733

Число 733 гиперпростое.

Пример:

Введите натуральное число:

19

Число 19 не гиперпростое.

## Практическая работа № 49а (базовый курс)

### Процедуры и функции

**Уровень А.** Напишите функцию, которая определяет сумму цифр переданного ей числа.

Пример:

Введите натуральное число:

123

Сумма цифр числа 123 равна 6.

**Уровень В.** Напишите функцию, которая находит наибольший общий делитель двух натуральных чисел.

Пример:

Введите два натуральных числа:

7006652 112307574

$\text{НОД}(7006652, 112307574) = 1234$ .

**Уровень С.** Напишите функцию, которая «переворачивает» число, то есть возвращает число, в котором цифры стоят в обратном порядке.

Пример:

Введите натуральное число:

1234

После переворота: 4321.

**Уровень D.** Напишите логическую функцию, которая определяет, являются ли два переданные ей числа взаимно простыми, то есть, не имеющими общих делителей, кроме 1.

Пример:

Введите два натуральных числа:

28 15

Числа 28 и 15 взаимно простые.

Пример:

Введите два натуральных числа:

28 16

Числа 28 и 16 не взаимно простые.



## Практическая работа № 42.

### Рекурсия

**Уровень А.** Напишите рекурсивную функцию, которая вычисляет НОД двух натуральных чисел, используя модифицированный алгоритм Евклида.

Пример:

Введите два натуральных числа:

7006652 112307574

$\text{НОД}(7006652, 112307574) = 1234.$

**Уровень В.** Напишите рекурсивную функцию, которая раскладывает число на простые сомножители.

Пример:

Введите натуральное число:

378

$378 = 2 * 3 * 3 * 3 * 7$

**Уровень С.** Дано натуральное число N. Требуется получить и вывести на экран количество всех возможных *различных* способов представления этого числа в виде суммы натуральных чисел (то есть,  $1 + 2$  и  $2 + 1$  – это один и тот же способ разложения числа 3). Решите задачу с помощью рекурсивной функции.

Пример:

Введите натуральное число:

4

Количество разложений: 4

## Практическая работа (дополнительная)

### Стек

Из учебника вы уже знаете, что при вызове процедур важнейшую роль играет *стек* – область оперативной памяти, в которой хранятся адрес возврата из процедуры и локальные переменные. В этой практической работе мы познакомимся с работой стека на примере учебного компьютера «ЛамПанель», с которым вы уже встречались при изучении глав 4 и 5.

#### Возможности программы «ЛамПанель»

Программа «ЛамПанель» – это модель процессора, который управляет ламповой панелью, то есть, может с помощью специальных команд зажигать и гасить определенные лампочки.

Стек в программе «ЛамПанель» размещается в оперативной памяти, вместе с программой и данными. Оперативная память имеет размер 256 байт, адреса ячеек (байтов) изменяются от 0 до  $255 = FF_{16}$ . Программа начинается с адреса 0 (в начале области памяти), данные обычно расположены сразу за ней. Стек находится в самом конце оперативной памяти и «растет вверх». Это значит, что первое записанное в стек 16-битное слово имеет адрес  $FE_{16}$  (последние два байта памяти), следующее записанное слово – адрес  $FC_{16}$  и т.д. Как же компьютер разбирается, где начинается стек и сколько чисел туда записано?

В процессоре есть специальный регистр **SP** (от англ. *Stack Pointer* – указатель стека), в котором хранится адрес *вершины стека*, то есть последнего записанного в стек 16-битного значения. При запуске программы в регистр **SP** записывается значение  $100_{16}$  (область 1 на рисунке). Этот адрес находится за границами оперативной памяти и говорит о том, что стек пуст.



Для того, чтобы записать значение из регистра общего назначения в стек, используется команда **PUSH** (от англ. *push* – втолкнуть). Например, при выполнении этих команд в стек будет записано значение регистра **R0**:

```
MOV 1234,R0
PUSH R0
```

Обратите внимание на значение регистра **SP** – оно стало равно  $FE_{16}$  и теперь указывает на последнее слово памяти (область 2 на рисунке), в котором записано число  $1234_{16}$  – значение регистра **R0** (сначала младший байт, потом старший).



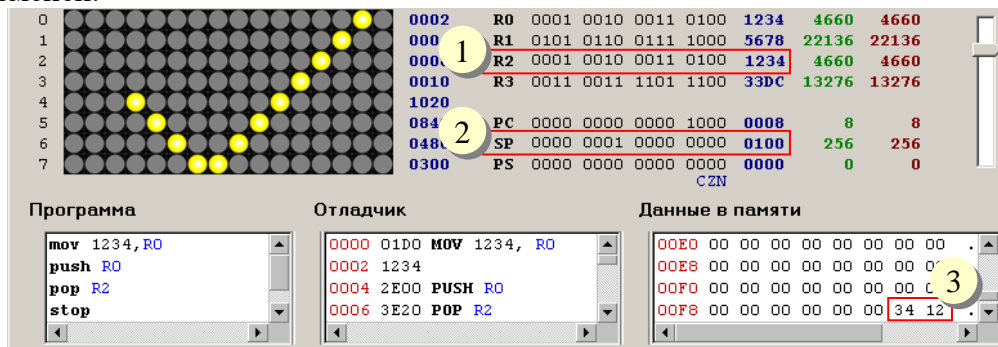
Добавим к программе еще одну команду, которая «снимает» 16-битное значение с вершины стека и записывает его в регистр **R2**:

```
POP R2
```

После этого наблюдаем следующее (см. рисунок ниже):

- в регистре **R2** находится то же значение  $1234_{16}$ , которое было в **R0** (область 1)
- регистр **SP** содержит значение  $100_{16}$ , которое говорит о том, что стек пуст (область 2)

- в последних двух байтах памяти осталось значение  $1234_{16}$ , которое было записано в стек (область 3), но теперь оно уже не является часть стека, поскольку регистр **SP** изменен.



### Вызов подпрограмм

Как вы знаете, подпрограммы – это вспомогательные алгоритмы. Напишем подпрограмму, которая возводит в квадрат значение регистра **R0**. Эта подпрограмма содержит всего одну команду умножения (умножить **R0** на **R0**, записать результат в **R0**):

**MUL R0, R0**

В начале подпрограммы нужно поставить метку (имя подпрограммы), а в конце – команду возврата **RET** (от англ. *return* – возврат), по которой процессор возвращается в точку вызова. Таким образом, вся подпрограмма, которую мы назовём **SQR**, выглядит так:

**SQR:**

**MUL R0, R0**

**RET**

«Паспорт» этой подпрограммы такой:

*Вход:* число в регистре **R0**

*Выход (результат):* квадрат числа в регистре **R0**

Подпрограмма располагается ниже основной программы. Чтобы вызвать подпрограмму, используют команду **CALL** (от англ. *call* – вызвать), после которой записывают имя подпрограммы – метку, с которой она начинается, адрес точки входа. Вот вся программа вместе с подпрограммой:

**MOV 12,R0** ; записать начальное значение в **R0**

**CALL SQR** ; вызвать подпрограмму **SQR**

**STOP** ; конец основной программы

**SQR:** ; точка входа подпрограммы

**MUL R0, R0** ; тело подпрограммы – возведение **R0** в квадрат

**RET** ; возврат из подпрограммы

Остается один вопрос: как процессор определяет адрес возврата из подпрограммы, когда выполняется команда **RET**? Заметим, что в самой команде **RET** адрес не указан. Дело в том, что адрес перехода заранее определить нельзя (нельзя поставить команду безусловного перехода **JMP**), потому что подпрограмма может вызываться из разных мест программы, в том числе из других подпрограмм. Эту проблему оказалось просто решить с помощью стека:

- команда **CALL** записывает в стек адрес возврата из подпрограммы, то есть адрес команды, следующей за командой **CALL**; поскольку регистр **PC** (программный счётчик) всегда содержит адрес следующей команды, процессору достаточно просто скопировать содержимое регистра **PC** в стек;
- после этого в регистр записывается адрес указанной подпрограммы и ей передается управление;
- команда **RET** снимает с вершины стека адрес возврата и записывает его в регистр **PC**, таким образом управление передается следующей команде вызывающей программы.

## Сохранение регистров

Теперь напишем более сложную подпрограмму, которая возводит **R0** в куб. Теперь для вычисления придется задействовать ещё один регистр, например, **R1**:

```
MOV R0,R1 ; скопировать R0 в R1
MUL R0,R0 ; возвести R0 в квадрат
MUL R1,R0 ; умножить R1 на R0
```

Все хорошо, но... мы стерли значение регистра **R1**, которое было до вызова подпрограммы. Чтобы при вызове подпрограммы регистры не стирались, их нужно сохранять при входе в подпрограмму и восстанавливать перед самым выходом. Где сохранять? Самый простой выход – использовать стек, сохранять командой **PUSH**, и восстанавливать командой **POP**. Таким образом полный текст подпрограммы **CUBE** выглядит так:

```
CUBE:
PUSH R1 ; сохраняем R1
MOV R0,R1 ; скопировать R0 в R1
MUL R0,R0 ; возвести R0 в квадрат
MUL R1,R0 ; умножить R1 на R0
POP R1 ; восстанавливаем R1
RET
```

## Передача параметров в подпрограмму

В предыдущих примерах вы уже увидели, что параметры (дополнительные данные) могут передаваться в подпрограмму через регистры общего назначения **R0-R3**. Но этих регистров всего четыре, поэтому таким способом можно передать только четыре 16-битных числа. А что, если нужно передать, например, массив из 100 элементов? В этом случае на помощь приходит стек.

Перед вызовом подпрограммы в стек записываются все передаваемые параметры. Рассмотрим сначала «игрушечную» задачу – написать подпрограмму, которая возводит число в квадрат, причем это число передается через стек. Результат должен быть помещен в регистр **R0**.

Перед вызовом подпрограммы запишем в стек значение **R0**:

```
0000 MOV 12,R0 ; это число нужно возвести в квадрат
0004 PUSH R0 ; запишем его в стек
0006 CALL SQR ; вызов подпрограммы
000A STOP
000C SQR:
... ; здесь будет подпрограмма
```

Если посмотреть на стек (в нижней части оперативной памяти), то после выполнения команды **PUSH R0** он выглядит так:

```
SP-> 00FE 0012
```

Указатель стека **SP** содержит адрес **00FE** и указывает на последнее 16-битное слово памяти. В стеке находится число  $12_{16}$  – значение, передаваемое в подпрограмму.

Когда выполнится команда **CALL**, в стек запишется адрес возврата из подпрограммы, то есть адрес  $000A_{16}$ , по которому расположена команда **STOP**. На этот адрес и будет указывать регистр **SP**:

```
SP-> 00FC 000A
00FE 0012
```

Теперь займемся подпрограммой: как «достать» переданное значение? Сначала нужно скопировать содержимое указателя стека в какой-то регистр общего назначения, например, в **R0**:

```
MOV SP,R0
```

Теперь в **R0** находится адрес вершины стека, но там лежит адрес возврата. Чтобы получить адрес переданного параметра, нужно увеличить **R0** на 2:

## **ADD 2,R0**

Теперь можно взять значение по этому адресу и записать его в тот же регистр **R0**:

## **MOV (R0),R0**

Запись **(R0)** означает «значение, находящееся в памяти по адресу, записанному в **R0**», это так называемый косвенный способ адресации, когда в регистре находится адрес данных, а не значение. Теперь в **R0** уже получено переданное число, и можно возвести его в квадрат. Вот полная подпрограмма:

## **SQR:**

**MOV SP,R0**

**ADD 2,R0**

**MOV (R0),R0**

**MUL R0,R0**

## **RET**

Остается один вопрос – кто же будет освобождать стек, удаляя из него параметры подпрограммы? Тут есть два варианта. Этим может заниматься вызывающая программа – после вызова подпрограммы нужно использовать команду **POP**. Кроме того, это может делать и процедура – для этого нужно применить команду **RET** с параметром, который обозначает количество байт, которые нужно «сбросить» со стека. Например, в нашем случае можно применить команду

## **RET 2**

которая освободит 2 байта (удалит один параметр). Отметим, что параметр команды **RET** – четное число, записанное в шестнадцатеричной системе счисления.

Таким образом, если параметров мало, их удобно передавать через регистры **R0-R3**.

Кроме того, параметры можно передавать через стек. Если подпрограмма обрабатывает большой массив, лучше передать ей адрес этого массива, вместо того, чтобы записывать его в стек.

## **Рекурсия**

Для процессора рекурсивная подпрограмма ничем не отличается от «обычной». Только внутри рекурсивной подпрограммы есть вызов **CALL** по адресу той же самой подпрограммы.

Напишем рекурсивную подпрограмму для вычисления факториала числа, находящегося в регистре **R0**. Результат должен быть помещен в тот же регистр **R0**.

Факториал числа  $N$  вычисляется как произведение всех натуральных чисел от 1 до  $N$ :  $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (N-1) \cdot N$ . Основная идея подпрограммы может быть записана на

псевдокоде так:

**R1:=R0**

**R0:=R0-1**

**вычисляем факториал R0 (вызов процедуры)**

**R0:=R0\*R1**

Перевод на язык ассемблера дает:

## **FACT:**

**MOV R0,R1**

**SUB 1,R0**

**CALL FACT**

**MUL R1,R0**

## **FINISH:**

**RET**

Однако, выполнение этой подпрограммы никогда не закончится, потому что вызовы никогда не остановятся. Нужно добавить условие выхода: если  $R0=1$ , нужно выйти из подпрограммы, то есть перейти на последнюю команду **RET** (перед ней должна быть метка):

## **FACT:**

```

CMP 1,R0
JZ FINISH
MOV R0,R1
SUB 1,R0
CALL FACT
MUL R1,R0
FINISH:
RET

```

Остается еще один недостаток – подпрограмма изменяет значение регистра R1. Нужно при входе в процедуру сохранить его (в стеке), а перед выходом – восстановить. Это вы уже можете сделать самостоятельно.

*Задание на практическую работу*

23. Запустите тренажёр «ЛамПанель». Напишите и отладьте программу, которая меняет местами значение регистров R2 и R3 с помощью стека (не используя других регистров общего назначения).

*Программа:*

*Опишите, как работает стек при выполнении этой программы:*

24. Введите текст программы

```

MOV 12,R0
CALL SQR
STOP
SQR:
MUL R0,R0
RET

```

Заполните таблицу, выполнив программу пошагово с помощью клавиши F7 (пошаговое выполнение со входом в подпрограммы):

Адрес	Команда	Регистры после ее выполнения		
		R0	PC	SP
		?	0000	0100
	<b>MOV 12,R0</b>			
	<b>CALL SQR</b>			
	<b>STOP</b>			
	<b>MUL R0,R0</b>			
	<b>RET</b>			

25. Напишите и отладьте программу с подпрограммой, которая вычисляет куб числа, записанного в регистр R0.

*Программа:*

26. Напишите и отладьте программу с подпрограммой, которая строит RGB-код цвета, 4-битные составляющие которого (R, G и B), записаны соответственно в регистры R0, R1 и R2. Результат должен быть получен в регистре R0.

*Программа:*

27. Выполните предыдущее задание при условии, что параметры передаются через стек, а значения регистров R1 и R2 не должны измениться.

*Программа:*

28. Отладьте программу с рекурсивной подпрограммой, которая вычисляет факториал числа, записанного в регистр R0. При выполнении в пошаговом режиме (клавиша F7)

наблюдайте, как изменяется регистр **SP** и содержимое стека.

*Программа:*

29. Решите задачу предыдущего пункта, используя подпрограмму без рекурсии.

*Программа:*

### Практическая работа № 43.

#### Заполнение массива

**Уровень А.** Введите с клавиатуры числа  $A$  и  $B$  ( $A < B$ ) и заполните массив случайными числами на отрезке  $[A; B]$ .

Пример:

Введите границы диапазона:

7 18

Массив:

13 9 11 17 12

**Уровень В.** Введите с клавиатуры числа  $X$  и  $D$  и заполните массив элементами арифметической прогрессии с начальным элементом  $X$  и разностью  $D$ . Не используйте встроенные функции.

Пример:

Введите  $X$  и  $D$ :

7 2

Массив:

7 9 11 13 15

**Уровень С.** Введите с клавиатуры числа  $X$  и  $D$  и заполните массив элементами арифметической прогрессии с начальным элементом  $X$  и разностью  $D$ , начиная с последнего элемента. Не используйте встроенные функции.

Пример:

Введите  $X$  и  $D$ :

7 2

Массив:

15 13 11 9 7

**Уровень D.** Заполните массив из  $N$  элементов случайными числами в интервале  $[1, N]$  так, чтобы в массив обязательно вошли все числа от 1 до  $N$  (постройте случайную перестановку, не используя для этого встроенные функции).

Пример:

Массив:

3 2 1 4 5



## Практическая работа № 44.

### Перебор элементов массива

**Уровень А.** Заполните массив случайными числами в интервале  $[0,100]$  и найдите среднее арифметическое его значений.

Пример:

Массив:

1 2 3 4 5

Среднее арифметическое 3.000

**Уровень В.** Заполните массив случайными числами в интервале  $[0,100]$  и подсчитайте отдельно среднее значение всех элементов, которые  $<50$ , и среднее значение всех элементов, которые  $\geq 50$ .

Пример:

Массив:

3 2 52 4 60

Среднее арифметическое элементов  $[0,50)$ : 3.000

Среднее арифметическое элементов  $[50,100)$ : 56.000

**Уровень С.** Заполните массив случайными числами в интервале  $[2,100]$  и подсчитайте среднее значение всех элементов, которые представляют собой простые числа.

Пример:

Массив:

10 23 37 46 84

Простые числа:

23 37

Среднее арифметическое: 30.0

**Практическая работа № 45.**

**Линейный поиск в массиве**

**Уровень А.** Заполните массив случайными числами в интервале  $[0,5]$ . Введите число  $X$  и найдите все значения, равные  $X$ .

Пример:

Массив:

1 2 3 1 2

Что ищем:

2

Нашли:  $A[2]=2$ ,  $A[5]=2$

Пример:

Массив:

1 2 3 1 2

Что ищем:

6

Ничего не нашли.

**Уровень В.** Заполните массив случайными числами в интервале  $[0,5]$ . Определить, есть ли в нем элементы с одинаковыми значениями, стоящие рядом.

Пример:

Массив:

1 2 3 3 2 1

Есть: 3

Пример:

Массив:

1 2 3 4 2 1

Нет

**Уровень С.** Заполните массив случайными числами. Определить, есть ли в нем элементы с одинаковыми значениями, не обязательно стоящие рядом.

Пример:

Массив:

3 2 1 3 2 5

Есть: 3, 2

Пример:

Массив:

3 2 1 4 0 5

Нет

## Практическая работа № 46.

### Поиск максимального элемента массива

**Уровень А.** Заполните массив случайными числами и найти минимальный и максимальный элементы массива и их номера.

Пример:

Массив:

1 2 3 4 5

Минимальный элемент:  $A[1]=1$

Максимальный элемент:  $A[5]=5$

**Уровень В.** Заполните массив случайными числами и найти два максимальных элемента массива и их номера.

Пример:

Массив:

5 5 3 4 1

Максимальный элемент:  $A[1]=5$

Второй максимум:  $A[2]=5$

**Уровень С.** Введите массив с клавиатуры и найдите (за один проход) количество элементов, имеющих максимальное значение.

Пример:

Массив:

3 4 5 5 3 4 5

Максимальное значение 5

Количество элементов 3

## **Практическая работа № 47.**

### **Алгоритмы обработки массивов**

**Уровень А.** В. Заполнить массив случайными числами и выполнить циклический сдвиг элементов массива вправо на 1 элемент.

Пример:

Массив:

1 2 3 4 5 6

Результат:

6 1 2 3 4 5

**Уровень В.** Массив имеет четное число элементов. Заполнить массив случайными числами и выполнить реверс отдельно в первой половине и второй половине.

Пример:

Массив:

1 2 3 4 5 6

Результат:

3 2 1 6 5 4

**Уровень С.** Заполнить массив случайными числами в интервале  $[-100,100]$  и переставить элементы так, чтобы все положительные элементы стояли в начала массива, а все отрицательные и нули – в конце. Вычислите количество положительных элементов.

Пример:

Массив:

20 -90 15 -34 10 0

Результат:

20 15 10 -90 -34 0

Количество положительных элементов: 3

## **Практическая работа № 48.**

### **Отбор элементов массива по условию**

**Уровень А.** Заполнить массив случайными числами в интервале  $[-10,10]$  и отобразить в другой массив все чётные отрицательные числа.

Пример:

Массив А:

-5 6 7 -4 -6 8 -8

Массив В:

-4 -6 -8

**Уровень В.** Заполнить массив случайными числами в интервале  $[0,100]$  и отобразить в другой массив все простые числа. Используйте логическую функцию, которая определяет, является ли переданное ей число простым.

Пример:

Массив А:

12 13 85 96 47

Массив В:

13 47

**Уровень С.** Заполнить массив случайными числами и отобразить в другой массив все числа Фибоначчи. Используйте логическую функцию, которая определяет, является ли переданное ей число числом Фибоначчи.

Пример:

Массив А:

12 13 85 34 47

Массив В:

13 34

**Практическая работа № 49.**

**Простые методы сортировки**

**Уровень А.** Напишите программу, в которой сортировка выполняется «методом камня» – самый «тяжёлый» элемент опускается в конец массива.

**Уровень В.** Напишите вариант метода пузырька, который заканчивает работу, если на очередном шаге внешнего цикла не было перестановок.

**Уровень С.** Напишите программу, которая сортирует массив по убыванию суммы цифр числа. Используйте функцию, которая определяет сумму цифр числа.

## **Практическая работа № 50.**

### **Сортировка слиянием**

**Уровень А.** Массив содержит четное количество элементов. Напишите программу, которая сортирует первую половину массива по возрастанию, а вторую – по убыванию. Каждый элемент должен остаться в «своей» половине.

Пример:

Массив:

5 3 4 2 1 6 3 2

После сортировки:

2 3 4 5 6 3 2 1

**Уровень В.** Напишите программу, которая сортирует массив и находит количество различных чисел в нем.

Пример:

Массив:

5 3 4 2 1 6 3 2 4

После сортировки:

1 2 2 3 3 4 4 5 6

Различных чисел: 6

**Уровень С.** Напишите программу, которая сравнивает число перестановок элементов при использовании сортировки «методом пузырька» и слиянием. Проверьте ее на разных массивах, содержащих 1000 случайных элементов, вычислите среднее число перестановок для каждого метода.

## Практическая работа № 51.

### Быстрая сортировка

**Уровень А.** Массив содержит четное количество элементов. Напишите программу, которая сортирует по возрастанию отдельно элементы первой и второй половин массива. Каждый элемент должен остаться в «своей» половине. Используйте алгоритм быстрой сортировки.

Пример:

Массив:

5 3 4 2 1 6 3 2

После сортировки:

2 3 4 5 1 2 3 6

**Уровень В.** Напишите программу, которая сортирует массив и находит количество различных чисел в нем. Используйте алгоритм быстрой сортировки.

Пример:

Массив:

5 3 4 2 1 6 3 2 4

После сортировки:

1 2 2 3 3 4 4 5 6

Различных чисел: 5

**Уровень С.** Напишите программу, которая сравнивает число перестановок элементов при использовании сортировки «пузырьком», методом выбора и алгоритма быстрой сортировки. Проверьте ее на разных массивах, содержащих 1000 случайных элементов, вычислите среднее число перестановок для каждого метода.

**Уровень D.** Попробуйте построить массив из 10 элементов, на котором алгоритм быстрой сортировки показывает худшую эффективность (наибольшее число перестановок). Сравните это количество перестановок с эффективностью метода пузырька (для того же массива).



## Практическая работа № 52.

### Двоичный поиск

**Уровень А.** Заполнить массив случайными числами и отсортировать его. Ввести число X.

Используя двоичный поиск, определить, есть ли в массиве число, равное X.

Подсчитать количество сравнений.

Пример:

Массив:

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 7 9

Введите число X:

2

Число 2 найдено.

Количество сравнений: 2

**Уровень В.** Заполнить массив случайными числами и отсортировать его. Ввести число X.

Используя двоичный поиск, определить, сколько чисел, равных X, находится в массиве.

Пример:

Массив:

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 7 9

Введите число X:

4

Число 4 встречается 2 раз(а).

Пример:

Массив:

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 7 9

Введите число X:

14

Число 14 не встречается.

**Уровень С.** Заполнить массив случайными числами и ввести число и отсортировать его.

Ввести число X. Используя двоичный поиск, определить, есть ли в массиве число, равное X. Если такого числа нет, вывести число, ближайшее к X.

Пример:

Массив:

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 12 19

Введите число X:

12

Число 12 найдено.

Пример:

Массив:

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 12 19

Введите число X:

11

Число 11 не найдено.

Ближайшее число 12.

### **Практическая работа № 53.**

#### **Посимвольная обработка строк**

**Уровень А.** Ввести с клавиатуры символьную строку и заменить в ней все буквы «а» на «б» и все буквы «б» на «а» (заглавные на заглавные, строчные на строчные).

Пример:

Введите строку:

ааббААББссСС

Результат:

ббааББААссСС

**Уровень В.** Ввести с клавиатуры символьную строку и определить, сколько в ней слов.

Словом считается последовательности непробельных символов, отделенная с двух сторон пробелами (или стоящая с краю строки). Слова могут быть разделены несколькими пробелами, в начале и в конце строки тоже могут быть пробелы.

Пример:

Введите строку:

Вася пошел гулять

Найдено слов: 3

**Уровень С.** Ввести с клавиатуры символьную строку и найдите самое длинное слово и его длину. Словом считается последовательности непробельных символов, отделенная с двух сторон пробелами (или стоящая с краю строки). Слова могут быть разделены несколькими пробелами, в начале и в конце строки тоже могут быть пробелы.

Пример:

Введите строку:

Вася пошел гулять

Самое длинное слово: гулять, длина 6

## Практическая работа № 54.

### Функции для работы со строками

**Уровень А.** Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести фамилию и инициалы.

Пример:

Введите фамилию, имя и отчество:

Иванов Петр Семёнович

П.С. Иванов

**Уровень В.** Ввести адрес файла и «разобрать» его на части, разделенные знаком '/'.  
Каждую часть вывести в отдельной строке.

Пример:

Введите адрес файла:

C:/Фото/2013/Поход/vasya.jpg

C:

Фото

2013

Поход

vasya.jpg

**Уровень С.** Напишите программу, которая заменяет во всей строке одну последовательность символов на другую.

Пример:

Введите строку:

(X > 0) and (Y < X) and (Z > Y) and (Z <> 5)

Что меняем: and

Чем заменить: &

Результат

(X > 0) & (Y < X) & (Z > Y) & (Z <> 5)

## Практическая работа № 55.

### Преобразования «строка-число»

**Уровень А.** Напишите программу, которая вычисляет сумму трех чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение:

12+3+45

Ответ: 60

**Уровень В.** Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются только знаки «+» или «-»). Выражение вводится как символьная строка, все числа целые.

Пример:

Введите выражение:

12-3+45

Ответ: 54

**Уровень С.** Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются знаки «+», «-», «\*» и «/»). Выражение вводится как символьная строка, все числа целые. Операция «/» выполняется как целочисленное деление (**div**).

Пример:

Введите выражение:

12\*3+45

Ответ: 81

**Уровень D.** Напишите программу, которая вычисляет выражение, состоящее из чисел, знаков (допускаются знаки «+», «-», «\*» и «/») и круглых скобок. Выражение вводится как символьная строка, все числа целые. Операция «/» выполняется как целочисленное деление (**div**).

Пример:

Введите выражение:

2\*(3+45)+4

Ответ: 100

## **Практическая работа № 56.**

### **Строки в процедурах и функциях**

**Уровень А.** Напишите функцию, которая возвращает первое слово переданной ей символьной строки.

Пример:

Введите строку: Однажды в студёную зимнюю пору...

Первое слово: Однажды

**Уровень В.** Напишите функцию, которая заменяет расширение файла на заданное новое расширение.

Пример:

Введите имя файла: qq

Введите новое расширение: tmp

Результат: qq.tmp

Пример:

Введите имя файла: qq.exe

Введите новое расширение: tmp

Результат: qq.tmp

Пример:

Введите имя файла: qq.work.xml

Введите новое расширение: tmp

Результат: qq.work.tmp

**Уровень С.** Напишите функцию, которая заменяет во всей строке все римские числа на соответствующие десятичные числа.

Пример:

Введите строку:

В ММХIII году в школе СХХIII состоялся очередной выпуск XI классов.

Результат:

В 2013 году в школе 123 состоялся очередной выпуск 11 классов.

## **Практическая работа № 57.**

### **Рекурсивный перебор**

**Уровень А.** В алфавите языке племени «тумба-юмба» четыре буквы: «Ы», «Ш», «Ч» и «О». Нужно вывести на экран все возможные слова, состоящие из **К** букв, в которых вторая буква «Ы». Подсчитайте количество таких слов.

**Уровень В.** В алфавите языке племени «тумба-юмба» четыре буквы: «Ы», «Ш», «Ч» и «О». Нужно вывести на экран все возможные слова, состоящие из **К** букв, в которых есть по крайней мере две одинаковые буквы, стоящие рядом. Подсчитайте количество таких слов.

Программа не должна строить другие слова, не соответствующие условию.

**Уровень С.** В алфавите языке племени «тумба-юмба» четыре буквы: «Ы», «Ш», «Ч» и «О». Нужно вывести на экран все возможные слова, состоящие из **К** букв, в которых есть по крайней мере две одинаковые буквы, не обязательно стоящие рядом.

Программа не должна строить другие слова, не соответствующие условию.

## **Практическая работа № 58.**

### **Сравнение и сортировка строк**

**Уровень А.** Вводится 5 строк, в которых сначала записан порядковый номер строки с точкой, а затем – слово. Вывести слова в алфавитном порядке.

Пример:

Введите 5 строк:

1. тепловоз
2. арбуз
3. бурундук
4. кефир
5. урядник

Список слов в алфавитном порядке:

арбуз, бурундук, кефир, тепловоз, урядник

**Уровень В.** Вводится несколько строк (не более 20), в которых сначала записан порядковый номер строки с точкой, а затем – слово. Ввод заканчивается пустой строкой. Вывести введенные слова в алфавитном порядке.

Пример:

Введите слова:

1. тепловоз
2. арбуз

Список слов в алфавитном порядке:

арбуз, тепловоз

**Уровень С.** Вводится несколько строк (не более 20), в которых сначала записаны инициалы и фамилии работников фирмы. Ввод заканчивается пустой строкой. Отсортировать строки в алфавитном порядке по фамилии.

Пример:

Введите ФИО:

- А.Г. Урядников
- Б.В. Тепловозов
- В.Д. Арбузов

Список в алфавитном порядке:

- В.Д. Арбузов
- Б.В. Тепловозов
- А.Г. Урядников

## Практическая работа (дополнительно)

### Обработка символьных строк: сложные задачи

**Уровень А.** На вход программы подаются данные о результатах районной олимпиады. В первой строке – количество участников  $N$ , а следующие  $N$  строк имеют следующий формат:

**<Фамилия> <Имя> <Баллы>**

Здесь *<Фамилия>* – строка, состоящая не более чем из 20 символов; *<Имя>* – строка, состоящая не более чем из 15 символов; *<Баллы>* – целое число, обозначающее общее количество баллов, набранное участником. Все данные разделены пробелами. Пример входной строки:

**Петечкин Вася 152**

На городскую олимпиаду проходят участники, набравшие в сумме более 100 баллов. Требуется вывести количество участников, прошедших на городскую олимпиаду.

**Уровень В.** На вход программы подаются данные о результатах районной олимпиады. В первой строке – количество участников  $N$ , а следующие  $N$  строк имеют следующий формат:

**<Фамилия> <Имя> <Баллы>**

Здесь *<Фамилия>* – строка, состоящая не более чем из 20 символов; *<Имя>* – строка, состоящая не более чем из 15 символов; *<Баллы>* – целое число, обозначающее общее количество баллов, набранное участником. Все данные разделены пробелами. Пример входной строки:

**Петечкин Вася 152**

На городскую олимпиаду проходят участники, набравшие в сумме более 100 баллов. Требуется вывести список участников, прошедших на городскую олимпиаду (известно, что их не более 25).

**Уровень С.** На вход программы подаются данные о результатах районной олимпиады. В первой строке – количество участников  $N$  (не более 100), а следующие  $N$  строк имеют следующий формат:

**<Фамилия> <Имя> <Баллы>**

Здесь *<Фамилия>* – строка, состоящая не более чем из 20 символов; *<Имя>* – строка, состоящая не более чем из 15 символов; *<Баллы>* – целое число, обозначающее общее количество баллов, набранное участником. Все данные разделены пробелами. Пример входной строки:

**Петечкин Вася 152**

На городскую олимпиаду проходят лучшие 25 участников (набравшие в сумме наибольшее количество баллов). Если другие участники набрали столько же баллов, что и участник, занявший 25-е место, они тоже приглашаются на городскую олимпиаду. Требуется вывести список участников, прошедших на городскую олимпиаду, отсортировав его по фамилии (в алфавитном порядке).



## Практическая работа № 59.

### Матрицы

**Уровень А.** Напишите программу, которая заполняет квадратную матрицу случайными числами в интервале [10,99], и находит максимальный и минимальный элементы в матрице и их индексы.

Пример:

Матрица А:

12 14 67 45

32 87 45 63

69 45 14 11

40 12 35 15

Максимальный элемент  $A[2,2]=87$

Минимальный элемент  $A[3,4]=11$

**Уровень В.** Пиксели рисунка закодированы числами от 0 до 255 (обозначающими яркость пикселей) в виде матрицы, содержащей  $N$  строк и  $M$  столбцов. Нужно преобразовать рисунок в черно-белый по следующему алгоритму:

- вычислить среднюю яркость пикселей по всему рисунку
- все пиксели, яркость которых меньше средней, сделать черными (записать код 0), а остальные – белыми (код 255)

Пример:

Матрица А:

12 14 67 45

32 87 45 63

69 45 14 11

40 12 35 15

Средняя яркость 37.88

Результат:

0 0 255 255

0 255 0 255

255 255 0 0

255 0 0 0

**Уровень С.** Заполните матрицу, содержащую  $N$  строк и  $M$  столбцов, натуральными числами по спирали и змейкой, как на рисунках:

а)

1	2	3	4
10	11	12	5
9	8	7	6

б)

1	3	4	9
2	5	8	10
6	7	14	12

в)

1	6	7	12
2	5	8	11
3	4	9	10

**Практическая работа № 60.**  
**Алгоритмы обработки матриц**

**Уровень А.** Напишите программу, которая заполняет квадратную матрицу случайными числами в интервале [10,99], а затем записывает нули во все элементы выше главной диагонали. Алгоритм не должен изменяться при изменении размеров матрицы.

Пример:

Матрица А:

12 14 67 45

32 87 45 63

69 45 14 30

40 12 35 65

Результат:

12 0 0 0

32 87 0 0

69 45 14 0

40 12 35 65

**Уровень В.** Пиксели рисунка закодированы числами (обозначающими цвет) в виде матрицы, содержащей **N** строк и **M** столбцов. Выполните отражение рисунка сверху вниз:

1	2	3
4	5	6
7	8	9

 → 

7	8	9
4	5	6
1	2	3

**Уровень С.** Пиксели рисунка закодированы числами (обозначающими цвет) в виде матрицы, содержащей **N** строк и **M** столбцов. Выполните поворот рисунка вправо на 90 градусов:

1	2	3
4	5	6
7	8	9

 → 

7	4	1
8	5	2
9	6	3

## **Практическая работа № 61.**

### **Файловый ввод и вывод**

**Уровень А.** Напишите программу, которая находит среднее арифметическое всех чисел, записанных в файле в столбик, и выводит результат в другой файл.

**Уровень В.** Напишите программу, которая находит минимальное и максимальное среди чётных положительных чисел, записанных в файле, и выводит результат в другой файл. Учтите, что таких чисел может вообще не быть.

**Уровень С.** В файле в столбик записаны целые числа, сколько их – неизвестно. Напишите программу, которая определяет длину самой длинной цепочки идущих подряд одинаковых чисел и выводит результат в другой файл.

## **Практическая работа № 62.**

### **Обработка массивов из файла**

**Уровень А.** В файле записано не более 100 чисел. Отсортировать их по возрастанию последней цифры и записать в другой файл.

**Уровень В.** В файле записано не более 100 чисел. Отсортировать их по возрастанию суммы цифр и записать в другой файл. Используйте функцию, которая вычисляет сумму цифр числа.

**Уровень С.** В двух файлах записаны отсортированные по возрастанию массивы неизвестной длины. Объединить их и записать результат в третий файл. Полученный массив также должен быть отсортирован по возрастанию.

### **Практическая работа № 63.**

#### **Обработка смешанных данных из файла**

**Уровень А.** В файле записаны данные о результатах сдачи экзамена. Каждая строка содержит фамилию, имя и количество баллов, разделенные пробелами:

**<Фамилия> <Имя> <Количество баллов>**

Вывести в другой файл фамилии и имена тех учеников, которые получили больше 80 баллов.

**Уровень В.** В предыдущей задаче добавить к полученному списку нумерацию, сократить имя до одной буквы и поставить перед фамилией:

- 1) П. Иванов
- 2) И. Петров
- 3) ...

**Уровень С.** В файле записаны данные о результатах сдачи экзамена. Каждая строка содержит фамилию, имя и количество баллов, разделенные пробелами:

**<Фамилия> <Имя> <Количество баллов>**

Вывести в другой файл данные учеников, которые получили больше 80 баллов.

Список должен быть отсортирован по убыванию балла. Формат выходных данных:

- 3) П. Иванов 98
- 4) И. Петров 96
- 5) ...

## Глава 9. Решение вычислительных задач на компьютере

### Практические работы

#### Практическая работа № 64.

##### Решение уравнений методом перебора

Напишите программу, которая находит все решения заданного вам уравнения на интервале  $[-5;5]$ . Программа должна выполнить следующие действия:

28. Определяет и выводит на экран интервалы, на которых расположены корни уравнения.

29. На каждом интервале, используя метод перебора, ищет решение с точностью 0,001 и выводит полученные решения на экран.

Вычисление функции, стоящей в левой части уравнения, оформите в виде подпрограммы.

**Уровень А.** Интервалы, на которых расположены корни, можно найти с помощью электронных таблиц. Программа запрашивает левую границу очередного интервала и выводит найденный корень уравнения.

Пример:

Введите левую границу интервала:

1.5

Решение: 1.7201

**Уровень В.** Составить две программы, одна из которых выделяет все интервалы, на которых находятся корни, а вторая запрашивает левую границу очередного интервала и выводит найденный корень уравнения.

**Уровень С.** Составить одну программу, которая работает полностью автоматически: достаточно ввести в программу функцию и запустить. Программа находит все интервалы, на которых расположены корни и уточняет решения. После того, как очередной интервал найден, программа выводит его на экран и, применяя метод перебора с нужным шагом, уточняет решение.

Пример:

Интервал  $[-2;-1,5]$

Решение: -1.7201

Интервал  $[1,5;2]$

Решение: 1.7201

**Варианты заданий по теме «Решение уравнений»**

<b>№</b>	<b>Уравнение</b>	<b>Ответы</b>		
<b>1</b>	$x^3 - 8x + 1 = 5 \sin x$	-2.945	0.077	2.854
<b>2</b>	$x^3 - 8x + 1 = -2 \sin x - 10 \cos x$	-2.233	1.088	3.286
<b>3</b>	$x^3 - 8x + 1 = -12 \sin x - 10 \cos x$	-1.332	1.975	3.379
<b>4</b>	$x^3 - 8x + 1 = 5 \sin x + 12 \cos x$	-3.377	-0.683	2.347
<b>5</b>	$x^3 - 8x + 1 = -5 \sin x + 12 \cos x$	-3.468	-1.210	1.798
<b>6</b>	$x^3 + 2x^2 - 8x + 1 = -2 \sin x - 15 \cos x$	-3.123	1.493	2.638
<b>7</b>	$x^3 + 2x^2 - 8x + 1 = -2 \sin x + 5 \cos x$	-4.200	-0.497	1.639
<b>8</b>	$x^3 + 2x^2 - 8x + 1 = -12 \sin x + 8 \cos x$	-4.495	-1.386	0.850
<b>9</b>	$x^3 + 2x^2 - 8x + 1 = 5 \sin x + 12 \cos x$	-4.133	-0.638	1.941
<b>10</b>	$x^3 + 2x^2 - 8x + 1 = -5 \sin x + 12 \cos x$	-4.356	-1.042	1.440
<b>11</b>	$x^3 + 3x^2 - 9x - 5 = -\cos x$	-4.690	-0.406	2.212
<b>12</b>	$x^3 + 3x^2 - 9x - 20 = -8 \sin x + 8 \cos x$	-4.488	-2.100	2.363
<b>13</b>	$x^3 + 3x^2 - 9x - 5 = -5 \sin x + 5 \cos x$	-4.834	-1.066	1.776
<b>14</b>	$x^3 + 3x^2 - 9x - 5 = -10 \sin x + 10 \cos x$	-4.931	-1.441	1.448
<b>15</b>	$x^3 + 3x^2 - 9x - 20 = -18 \sin x + 25 \cos x$	-4.756	-2.101	1.720
<b>16</b>	$x^3 + 2x^2 - 6x - 5 = -\sin x - \cos x$	-3.285	-0.707	2.039
<b>17</b>	$x^3 + 2x^2 - 6x - 5 = -\sin x - 15 \cos x$	-1.986	1.086	2.704
<b>18</b>	$x^3 + 2x^2 - 6x - 5 = -10 \sin x - 20 \cos x$	-1.281	1.973	2.772
<b>19</b>	$x^3 + 2x^2 - 6x - 5 = -10 \sin x + 10 \cos x$	-4.040	-1.866	1.316
<b>20</b>	$x^3 + 2x^2 - 6x - 5 = -18 \sin x + 25 \cos x$	-4.395	-1.968	1.196

## Практическая работа № 65.

### Решение уравнений методом деления отрезка пополам

Напишите программу, которая находит все решения заданного вам уравнения на интервале  $[-5;5]$ . Программа должна выполнить следующие действия:

1. Определяет и выводит на экран интервалы, на которых расположены корни уравнения.
2. Используя метод деления отрезка пополам, на каждом интервале ищет решение с точностью  $0,001$  и выводит полученные решения на экран.

Вычисление функции, стоящей в левой части уравнения, оформите в виде подпрограммы.

**Уровень А.** Интервалы, на которых расположены корни, можно найти с помощью электронных таблиц. Программа запрашивает границы очередного интервала и выводит найденный корень уравнения.

Пример:

Введите границы интервала:

1.5 2

Решение: 1.7201

**Уровень В.** Составить две программы, одна из которых выделяет все интервалы, на которых находятся корни, а вторая запрашивает границы очередного интервала и выводит найденный корень уравнения, а также число шагов, которые потребовались для достижения заданной точности.

Пример:

Введите границы интервала:

1.5 2

Решение: 1.7201

Число шагов: 8

**Уровень С.** Составить одну программу, которая работает полностью автоматически: достаточно ввести в программу функцию и запустить. Программа находит все интервалы, на которых расположены корни и уточняет решения. После того, как очередной интервал найден, программа выводит его на экран и, применяя метод деления отрезка пополам, уточняет решение и определяет необходимое для этого количество шагов.

Пример:

Интервал  $[-2;-1,5]$

Решение: -1.7201

Число шагов: 8

Интервал  $[1,5;2]$

Решение: 1.7201

Число шагов: 8



## Практическая работа № 66.

### Решение уравнений в табличных процессорах

**Уровень А.** С помощью табличного процессора найдите все решения уравнения из примера «Полёт мяча», разобранный в учебнике.

**Уровень В.** Выполните задание уровня А. Затем выделите отдельную ячейку, в которой вычисляется максимальное значение функции  $f(\alpha)$  на выбранном интервале.

Используя эту ячейку и подбор параметра, найдите:

- 1) минимальную начальную скорость, при которой еще можно попасть в мишень;
- 2) максимальное расстояние, при котором можно попасть в мишень при начальной скорости 15 м/с.

**Уровень С.** Выполните задание уровня В. Затем, используя замену  $z = \frac{1}{\cos^2 \alpha}$ , постройте

аналитическое решение уравнения из примера «Полёт мяча». Для практических вычислений используйте электронные таблицы. Сравните точное и численное решения.

## Практическая работа № 67.

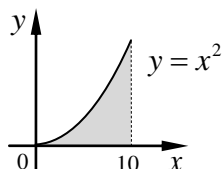
### Дискретизация

#### Уровень А.

1. Найдите длину параболы  $y = x^2$  на отрезке  $[0; 10]$ .

Ответ:

2. Найдите площадь фигуры, ограниченной параболой  $y = x^2$  и осью  $OX$ , на отрезке  $[0; 10]$ .



Ответ:

#### Уровень В.

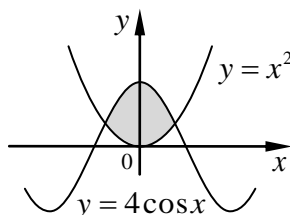
3. Для примера «Полет мяча» вычислите длину траектории движения шарика для углов вылета  $35,5^\circ$  и  $65,8^\circ$ . Сравните полученные результаты.

Ответ:

$$\text{угол } 35,5^\circ \Rightarrow L = \dots$$

$$\text{угол } 65,8^\circ \Rightarrow L = \dots$$

4. Найдите площадь фигуры, ограниченной графиками функций  $y = x^2$  и  $y = 4 \cos x$ .



Ответ:

#### Уровень С.

5. Постройте с помощью табличного процессора эллипс, который описывается уравнением  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$  при  $a = 2$  и  $b = 3$ . Напишите программу, которая вычисляет периметр этого эллипса с помощью дискретизации.

Ответ:

$$L = \dots$$

Сравните ваш результат с тем, что получается по приближенной формуле

$$\text{Рамануджана: } L = \pi \left[ 3(a+b) - \sqrt{(3a+b)(a+3b)} \right].$$

Ответ:

$$\text{По формуле Рамануджана: } L = \dots$$

Экспериментально определите наибольший шаг дискретизации, при котором ошибка численного метода (в сравнении с результатом, полученным по формуле Рамануджана) составляет не более 1%.

Ответ:

$$h_{\max} = \dots$$

6. Найдите площадь фигуры, ограниченной эллипсом

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \text{ при } a = 2 \text{ и } b = 3.$$

*Ответ:*

$S = \dots$

Сравните ваш результат с точным значением  $S = \pi \cdot a \cdot b$ .

*Ответ:*

$S = \dots$

Экспериментально определите наибольший интервал дискретизации, при котором ошибка численного метода составляет не более 1%.

*Ответ:*

$h_{\max} = \dots$

## Практическая работа № 68.

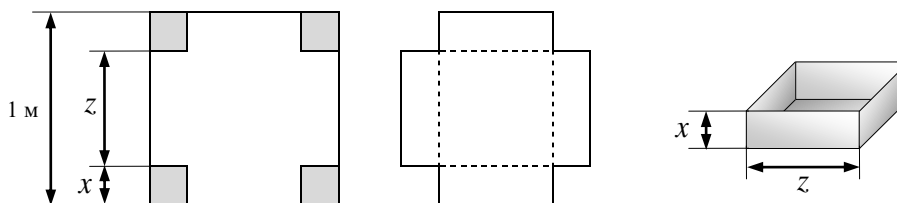
### Оптимизация

#### Уровень А.

1. Напишите программу, которая находит все минимумы и максимумы функции (см. варианты на следующей странице) на отрезке  $[-5;5]$  с точностью 0,001, используя метод «золотого сечения». Границы интервала для поиска каждого минимума и максимума задаются вручную. Определите координаты точек максимума и минимума заданной вам функции на этом отрезке:

Максимум	Минимум
$x = \dots, y = \dots$	$x = \dots, y = \dots$

2. В углах квадратного листа железа, сторона которого равна 1 м, вырезают четыре квадрата со стороной  $x$ . Затем складывают получившуюся развертку (по штриховым линиям на рисунке), сваривают швы и таким образом получается бак:



Определите размеры бака наибольшего возможного объема и значение этого объема.

Ответ:

$x = \dots, z = \dots, V = \dots$
-----------------------------------

#### Уровень В.

3. Напишите программу, которая находит все минимумы и максимумы функции (см. варианты на следующей странице) на отрезке  $[-5;5]$  с точностью 0,001, используя метод «золотого сечения». Границы интервала для поиска каждого минимума и максимума определяются автоматически.

Определите координаты точек максимума и минимума заданной вам функции на этом отрезке:

Максимум	Минимум
$x = \dots, y = \dots$	$x = \dots, y = \dots$

4. Фирма «Рога и копыта» хочет провести рекламную кампанию в газетах. Данные о цене рекламного объявления и тиражах газет внесены в таблицу:

Газета	Тираж	Цена 1 объявл.	Объявлений	Расходы	Охват
<i>Ведомости</i>	10000	1000р.	1	1000 р.	10000
<i>Туризм</i>	3500	570р.	2	1140 р.	7000
<i>Спорт</i>	6000	700р.	3	2100 р.	18000
<i>Правда</i>	20000	1250р.	4	5000 р.	80000
<b>Всего</b>				9240 р.	115000

В каждую газету нужно дать не менее одного и не более 6 объявлений. С помощью надстройки «Поиск решения» табличного процессора определите, сколько объявлений нужно дать в каждую газету, чтобы обеспечить общий охват не менее 200000 человек и при этом израсходовать как можно меньше денег. Исправьте значения в выделенных ячейках таблицы.

#### Уровень С.

5. Напишите программу, которая находит все минимумы и максимумы функции (см. варианты на следующей странице) на отрезке  $[-5;5]$  с точностью 0,001, используя метод «золотого сечения», причём на каждом шаге цикла вычисляется только одно новое значение функции. Границы интервала для поиска каждого минимума и

максимума определяются автоматически. Определите координаты точек максимума и минимума заданной вам функции на этом отрезке:

Максимум	Минимум
$x = \dots, y = \dots$	$x = \dots, y = \dots$

6. Банка имеет форму цилиндра, размеры которого задаются радиусом  $R$  и высотой  $H$ . Площадь полной поверхности банки вычисляется как сумма площади боковой поверхности и площадей двух кругов-оснований. Решите следующие задачи:

- 1) найдите размеры банки, которая имеет максимальный объем при площади полной поверхности  $100 \text{ см}^2$ ;

*Ответ:*

$R = \dots, H = \dots$
------------------------

- 2) найдите размеры банки, которая имеет минимальную площадь полной поверхности при объеме  $500 \text{ см}^3$ .

*Ответ:*

$R = \dots, H = \dots$
------------------------

**Варианты заданий по теме «Оптимизация. Метод дихотомии»**

<b>№</b>	<b>Уравнение</b>	<b>Ответы</b>
<b>1</b>	$f(x) = x^3 - 8x + 1 - 5\sin x$	max:(-1.612,14.703) min:( 1.612,-12.703)
<b>2</b>	$f(x) = x^3 - 8x + 1 + 2\sin x + 10\cos x$	max:(-0.565,12.715) min:( 2.348,-10.426)
<b>3</b>	$f(x) = x^3 - 8x + 1 + 12\sin x + 10\cos x$	max:( 0.368,11.753) min:( 2.760,-4.867)
<b>4</b>	$f(x) = x^3 - 8x + 1 - 5\sin x - 12\cos x$	max:(-2.210,19.058) min:( 0.857,-16.862)
<b>5</b>	$f(x) = x^3 - 8x + 1 + 5\sin x - 12\cos x$	max:(-2.520,12.001) min:( 0.250,-11.374)
<b>6</b>	$f(x) = x^3 + 2x^2 - 8x + 1 + 2\sin x + 15\cos x$	max:(-0.527,17.584) min:( 2.115,-3.568)
<b>7</b>	$f(x) = x^3 + 2x^2 - 8x + 1 + 2\sin x - 5\cos x$	max:(-2.754,21.186) min:( 0.604,-5.861)
<b>8</b>	$f(x) = x^3 + 2x^2 - 8x + 1 + 12\sin x - 8\cos x$	max:(-3.265,23.051) min:(-0.313,-7.637)
<b>9</b>	$f(x) = x^3 + 2x^2 - 8x + 1 - 5\sin x - 12\cos x$	max:(-2.601,30.604) min:( 0.691,-15.677)
<b>10</b>	$f(x) = x^3 + 2x^2 - 8x + 1 + 5\sin x - 12\cos x$	max:(-2.986,27.177) min:( 0.187,-11.281)
<b>11</b>	$f(x) = x^3 + 3x^2 - 9x - 5 + \cos x$	max:(-2.987,21.011) min:( 1.072,-9.490)
<b>12</b>	$f(x) = x^3 + 3x^2 - 9x - 20 + 8\sin x - 8\cos x$	max:(-3.417,15.758) min:( 0.072,-28.036)
<b>13</b>	$f(x) = x^3 + 3x^2 - 9x - 5 + 5\sin x - 5\cos x$	max:(-3.314,27.161) min:( 0.361,-10.723)
<b>14</b>	$f(x) = x^3 + 3x^2 - 9x - 5 + 10\sin x - 10\cos x$	max:(-3.468,33.262) min:(-0.062,-15.031)
<b>15</b>	$f(x) = x^3 + 3x^2 - 9x - 20 + 18\cos x - 25\cos x$	max:(-3.530,35.120) min:(-0.278,-46.268)
<b>16</b>	$f(x) = x^3 + 2x^2 - 6x - 5 + \sin x + \cos x$	max:(-2.208, 5.835) min:( 0.916,-6.647)
<b>17</b>	$f(x) = x^3 + 2x^2 - 6x - 5 + \sin x + 15\cos x$	max:(-0.430,11.088) min:( 2.003,-6.333)
<b>18</b>	$f(x) = x^3 + 2x^2 - 6x - 5 - 10\sin x - 20\cos x$	max:( 0.246,15.493) min:( 2.400,-2.049)
<b>19</b>	$f(x) = x^3 + 2x^2 - 6x - 5 + 10\sin x - 10\cos x$	max:(-3.099,12.605) min:(-0.277,-15.559)
<b>20</b>	$f(x) = x^3 + 2x^2 - 6x - 5 + 18\sin x - 25\cos x$	max:(-3.308,28.171) min:(-0.391,-32.381)

## Практическая работа № 69.

### Статистические расчеты

#### Уровень А.

1. Откройте файл **marks.xls**, который содержит таблицу с отметками учеников по четырём предметам. Часть таблицы показана на рисунке:

	Русский яз.	Математика	Физика	Биология
Аксентьева М.	5	4	4	5
Арбузова К.	3	5	4	5
Бугусов Н.	3	4	3	4

2. С помощью табличного процессора определите:
  - сколько пятерок, четверок и троек получено по каждому предмету;
  - среднюю оценку по каждому предмету;
  - средний балл каждого ученика
  - стандартное отклонение оценок по каждому предмету от среднего значения.
3. Отсортируйте список учеников по убыванию среднего балла.

#### Уровень В.

4. Найдите коэффициенты корреляции (взаимосвязь) между оценками по разным предметам, а также между средним баллом и оценками по каждому из предметов. Оформите данные в виде таблицы:

Коэффициенты корреляции

	Русский	Математика	Физика	Биология	Ср. балл
Русский	1				
Математика	0,155	1			
Физика	0,293	0,640	1		
Биология	0,203	-0,249	0,084	1	
Ср балл	0,686	0,612	0,796	0,405	1

Оставьте у всех значений только три знака в дробной части.

5. Откройте файл **salary.xls**, который содержит таблицу расчёта зарплаты работников фирмы «Рога и Копыта». Требуется добавить нужные формулы в столбцах Премия и Сумма.

	Зарплата	План	Премия	Сумма
Иванов	15 000р.	120%	3 000р.	18 000р.
Петров	15 000р.	100%	- р.	15 000р.
Медведев	15 000р.	90%	- р.	15 000р.
Сурков	16 000р.	110%	1 600р.	17 600р.
Барсуков	16 000р.	120%	3 200р.	19 200р.
...	...	...	...	...
<b>Всего</b>				<b>84 800р.</b>
<b>Максимум</b>				<b>19 200р.</b>
<b>Минимум</b>				<b>15 000р.</b>

Работник получает премию пропорционально перевыполнению плана. Например, если план выполнен на 120% (перевыполнение на 20%), размер премии составит 20% от его зарплаты. Работник, сделавший меньше нормы, не получает премии.

#### Уровень С.

6. Откройте файл **olimp.xls**, который содержит таблицу результатов районной олимпиады по математике. На городской тур проходят те учащиеся, которые набрали на районном туре не менее 10 баллов или решили полностью одну из самых сложных задач 6 или 7. За полное решение задач 1-4 дается 2 балла, задач 5-6 – 3 балла, задачи 7 – 4 балла.

Проходной балл		10								
Фамилия	Пол	Баллы за задачи							Сумма	Прошел
		1	2	3	4	5	6	7		
Айвазян Г.	ж	1	0	2	1	0	1	3	8	0
Бериев Б.	м	2	0	0	0	1	1	4	8	1
Вакин В.	м	2	0	0	1	0	3	0	6	1
Васильева Г.	ж	1	0	2	1	0	1	3	8	0

7. Добавьте необходимые формулы в столбцы *Сумма* и *Прошел*. Нужно сделать так, чтобы проходной балл можно было легко менять в отдельной ячейке (например, в B1).
8. Определите, сколько участников прошли на городскую олимпиаду.
9. Постройте график, который показывает, как изменяется количество прошедших на городскую олимпиаду при изменении проходного балла.

**Уровень D.**

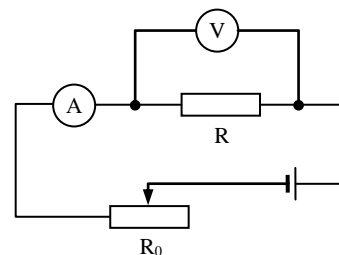
10. Было принято решение предоставить льготы девушкам: сделать для них проходной балл на 1 меньше, чем для юношей. Внесите размер этой льготы в отдельную ячейку и определите число прошедших на городскую олимпиаду с учетом льготы.



## Практическая работа № 70.

### Обработка результатов эксперимента

Для того, чтобы определить неизвестное сопротивление резистора  $R$  собрали электрическую схему, показанную на рисунке. Напряжение на резисторе регулируется с помощью реостата  $R_0$  и измеряется вольтметром  $V$ . Сила тока в цепи измеряется амперметром  $A$ .



#### Уровень А.

В текстовом файле **mnk.txt** результаты измерений записаны в два столбца: в первом столбце напряжение на проводнике, а во втором – соответствующая ему сила тока. Всего в файле 10 строчек с данными.

- 1) Напишите программу, которая читает данные из файла и вычисляет сопротивление резистора  $R$ , используя метод наименьших квадратов для закона Ома:  $U = I \cdot R$ .
- 2) Решите ту же задачу с помощью табличного процессора. Сравните результаты.

**Уровень В.** Выполните задание уровня А при условии, что точное количество измерений (количество строк с данными в файле) неизвестно, но не больше 100.

После этого решите задачу, решив задачу оптимизации в табличном процессоре. Для вычисления величины ошибки используйте функцию **SUMXMY2 (СУММКВРАЗН)**. Сравните полученные результаты.

**Уровень С.** Выполните задание уровня В. Затем с помощью табличного процессора решите задачу двумя другими методами:

**Метод 2.** Используйте в качестве критерия оптимизации не сумму квадратов отклонений, а сумму абсолютных величин (модулей) этих отклонений. Для вычисления модуля примените стандартную функцию **ABS**.

**Метод 3.** Закон Ома можно записать в виде  $I = U \cdot Q$ , где  $Q = \frac{1}{R}$ . Используя метод наименьших квадратов, найдите оптимальное значение  $Q$  с помощью своей программы и с помощью табличного процессора, а затем вычислите соответствующее значение  $R$ .

Сравните результаты, полученные тремя методами. Какой из них вы рекомендуете использовать и почему?

## Практическая работа № 78а (дополнительная) Линии тренда

**Уровень А.** Доходы начинающей фирмы (в тысячах рублей) за первые 5 лет работы приведены в таблице:

Год	1	2	3	4	5
Доход	93	187	270	321	350

С помощью табличного процессора определите возможную зависимость дохода от года работы (выберите лучший из стандартных вариантов, с наибольшим значением  $R^2$ ). С помощью этой зависимости сделайте прогноз развития фирмы на 2 года вперед.

**Уровень В.** По приведенным данным постройте три диаграммы типа «График», отображающую изменение прибыли каждой фирмы за несколько лет.

*Увеличение прибыли от продаж*

	Кей	Альфа	Сигма
2004	93	35	21
2005	187	115	118
2006	270	208	124
2007	321	250	172
2008	378	293	220

Определите заголовки осей и всей диаграммы.

### Задание:

- 1) Для каждого графика постройте линии тренда (нужно выбрать тип зависимости, который лучше всего соответствует данным).
- 2) Выведите на каждую диаграмм формулу для линий тренда и величину достоверности аппроксимации  $R^2$ .
- 3) Используя линии тренда, постройте графики прогнозируемых значений прибыли для всех фирм на следующие 5 лет. Подберите такой вид зависимости, который приводит к достоверным результатам прогноза.
- 4) Используя уравнение линий тренда, вычислите прогнозируемые значения прибыли каждой фирмы на следующие пять лет.

**Уровень С.** В файле **weather.xls** в таблице записаны метеорологические данные: давление в миллиметрах ртутного столба и температура за первую половину месяца.

День	Давление, мм.рт.ст.	Темп., °С
1	730	-10
2	740	-5
3	752	-12
4	760	-15
5	756	-12
6	754	-18
...	...	...

Предполагается, что изменение давления можно описать зависимостью вида

$$p(x) = k_0 + k_1 \sin(b_1 x + b_2)$$

а изменение температуры — более сложной формулой

$$t(x) = k_2 + k_3 \sin(b_3 x + b_4) + k_4 \sin(b_5 x + b_6)$$

где  $x$  — номер дня.

### Задание:

- 1) Определите все неизвестные коэффициенты, при которых указанные формулы дают наиболее точное приближение к исходным данным.
- 2) Оцените точность подбора зависимости, вычислив коэффициент  $R^2$ .
- 3) Постройте графики, отображающие исходные данные и полученные функции (отдельно для давления и температуры). Сделайте прогноз изменения давления и температуры до конца месяца.
- 4) Рассчитайте коэффициент корреляции исходных рядов давления и температуры. Какие выводы он позволяет сделать?

## Практическая работа № 72

### Антивирусная защита

1. Отключите антивирусную защиту, если она была включена.
2. Скопируйте в свой каталог архив **megagame.zip**.
3. Зайдите на сайт **www.virustotal.com** и загрузите на проверку этот архив, щелкнув по кнопке **Choose file**. Затем запустите проверку, щелкнув по кнопке **Scan it!**.

Запишите, какие вирусы были обнаружены:

Сколько антивирусов участвовало в проверке:

Сколько антивирусов не обнаружили ни одного вируса:

4. С помощью сайта **www.virustotal.com** проверьте файл **ROBOT.EXE**.

Сколько антивирусов не обнаружили ни одного вируса:

5. Если на компьютере установлена антивирусная программа, удалите вирусы из всех файлов.

6. Удалите архив **megagame.zip**.

7. Включите антивирусную защиту, если она была включена до выполнения работы.

# 11 класс

## Глава 10. Информация и информационные процессы

### Практические работы

#### Практическая работа № 1.

#### Алгоритм RLE

Файлы для выполнения этой работы находятся в каталоге **RLE**.

Для определения символов по их шестнадцатеричным кодам используйте приведённую ниже таблицу ASCII. В первом столбце записана первая цифра шестнадцатеричного кода символа, а в первой строке – вторая. Например, символ «&» имеет шестнадцатеричный код 26<sub>16</sub>.

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0	NU	SO	ST	ET	EO	EN	AC	BE	BS	TA	LF	VT	F	C	S	SI
.	L	H	X	X	T	Q	K	L		B			F	R	O	
1	DL	DC	DC	DC	DC	NA	SY	ET	CA	EM	SU	ES	F	G	R	US
.	E	1	2	3	4	K	N	B	N		B	C	S	S	S	
2		!	"	#	\$	%	&	'	(	)	*	+	,	—	.	/
.																
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
.																
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
.																
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
.																
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
.																
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DE
.																L

6. Используя алгоритм RLE, закодируйте последовательность символов

**BBBBBVAСССАВВВВВ**

Запишите результат в виде шестнадцатеричных кодов (каждый символ кодируется в виде байта, который представлен двумя шестнадцатеричными цифрами). Проверьте полученный результат с помощью программы RLE.

Ответ:

7. Раскодируйте последовательность, упакованную с помощью алгоритма RLE

(приводятся шестнадцатеричные коды): 01 4D 8E 41 01 4D 8E 41<sub>16</sub>.

Ответ:

8. Определите количество байтов в исходной и распакованной последовательности (см. предыдущее задание) и вычислите коэффициент сжатия:

Сжатая последовательность	Несжатая последовательность	Коэффициент сжатия

9. Проверьте результат, полученный в предыдущем пункте, с помощью программы RLE. Предложите два способа проверки.

Ответ:

10. Постройте последовательности, которые сжимаются алгоритмом RLE ровно в 2 раза, в 4 раза, в 5 раз. Проверьте свои ответы с помощью программы RLE.

Несжатая последовательность	Сжатая последовательность	Коэффициент сжатия
		2
		4
		5

11. Придумайте три последовательности, которые невозможно сжать с помощью алгоритма RLE:

Несжатая последовательность	«Сжатая» последовательность	Коэффициент сжатия

12. Используя программу RLE, примените RLE-сжатие к следующим файлам и найдите для каждого из них коэффициент сжатия:

Файл	Размер без сжатия	Размер после сжатия	Коэффициент сжатия
grad_vert.bmp			
grad_horz.bmp			
grad_diag.jpg			

13. Объясните результаты, полученные в предыдущем пункте:

- почему не удается сжать рисунки в формате JPEG?

Ответ:

- почему для двух рисунков в формате BMP одинакового размера коэффициенты сжатия по алгоритму RLE так сильно отличаются? *Подсказка:* откройте эти рисунки в любой программе просмотра.

Ответ:

14. Оцените максимально достижимый коэффициент сжатия с помощью рассмотренного в учебнике варианта RLE-алгоритма.

Ответ:

В каком случае его удастся достичь?

Ответ:

15. Оцените коэффициент сжатия с помощью RLE-алгоритма в худшем случае. Опишите этот худший случай.

Ответ:

## Практическая работа № 2.

### Сравнение алгоритмов сжатия

Файлы для выполнения этой работы находятся в каталоге **Compress**.

При выполнении этой работы используются программы **RLE** (алгоритм сжатия RLE) и **Huffman** (кодирование Хаффмана и Шеннона-Фано).

6. Запустите программу **Huffman.exe** и закодируйте строку «ЕНОТ НЕ ТОНЕТ», используя методы Шеннона-Фано и Хаффмана. Запишите результаты в таблицу:

	Шеннон и Фано	Хаффман
Длина основного кода		
Длина кодовой таблицы (дерева)		
Коэффициент сжатия (по основным кодам)		
Коэффициент сжатия (с учетом дерева кодов)		

Сделайте выводы.

Ответ:

Как, по вашему мнению, будет изменяться коэффициент сжатия при увеличении длины текста, при условии, что набор символов и частота их встречаемости останутся неизменной? Проверьте ваш вывод с помощью программы (например, можно несколько раз скопировать ту же фразу).

Ответ:

7. Повторите эксперимент с текстом, который записан в файле **enot.txt** (скопируйте этот текст в окно программы через буфер обмена).

	Шеннон и Фано	Хаффман
Длина основного кода		
Длина кодовой таблицы (дерева)		
Коэффициент сжатия (по основным кодам)		
Коэффициент сжатия (с учетом дерева кодов)		

Сделайте выводы.

Ответ:

Нарисуйте в тетради кодовые деревья, которые были построены программой при использовании обоих методов.

8. Используя кнопку **Анализ файла** в программе **Huffman**, определите предельный теоретический коэффициент сжатия для файла **a.txt**<sup>3</sup> при побайтном кодировании.

Ответ:

9. С помощью программ **RLE** и **Huffman** выполните сжатие файла **a.txt** разными способами. Запишите результаты в таблицу:

	RLE	Шеннон и Фано	Хаффман
Размер сжатого файла			
Коэффициент сжатия			

Объясните результат, полученный с помощью алгоритма RLE.

Ответ:

<sup>3</sup> Этот файл имеет объем 1 Мбайт и состоит из одних символов «А».

10. Используя кнопку *Анализ файла* в программе **Huffman**, определите предельный теоретический коэффициент сжатия для файла **a.txt.huf** при побайтном кодировании. Объясните результат.

Ответ:

11. Примените несколько раз повторное сжатие этого файла с помощью алгоритма Хаффмана (новые файлы получают имена **a.txt.huf2**, **a.txt.huf3** и т.д.) и заполните таблицу, каждый раз выполняя анализ полученного файла.

	Размер файла	Предельный коэффициент сжатия
<b>a.txt</b>		
<b>a.txt.huf</b>		
<b>a.txt.huf2</b>		
<b>a.txt.huf3</b>		
<b>a.txt.huf4</b>		
<b>a.txt.huf5</b>		
<b>a.txt.huf6</b>		

Объясните, почему с некоторого момента при сжатии файла его размер увеличивается.

Ответ:

12. Выполните те же действия, используя метод Шеннона-Фано.

	Размер файла	Предельный коэффициент сжатия
<b>a.txt</b>		
<b>a.txt.shf</b>		
<b>a.txt.shf2</b>		
<b>a.txt.shf3</b>		
<b>a.txt.shf4</b>		
<b>a.txt.shf5</b>		
<b>a.txt.shf6</b>		

13. Сравните результаты однократного сжатия файла **a.txt** с помощью алгоритма RLE, методов Шеннона-Фано и Хаффмана, а также результат сжатия этого файла каким-нибудь архиватором.

	Размер файла	Коэффициент сжатия
<b>RLE</b>		
<b>Хаффман</b>		
<b>Шеннон и Фано</b>		
<b>ZIP</b>		
<b>RAR</b>		
<b>7Z</b>		

Объясните результаты и сделайте выводы.

Ответ:

### Практическая работа № 3.

#### Использование архиватора

Файлы для выполнения этой работы находятся в каталоге **Archive**.

1. Изучите возможности архиватора, который установлен на вашем компьютере (*Ark, 7-Zip, WinRAR* или др.).
2. Откройте каталог, указанный учителем. Он должен содержать все файлы, которые используются далее.
3. Распакуйте архив **secret.zip**, который упакован с паролем **secretLatin**. В подкаталогах, получившихся после распаковки, вы должны найти 3 файла, содержащие части высказывания на латинском языке, которое означает «договоры следует выполнять». Запишите полученное высказывание.

Ответ:

4. Создайте новый текстовый файл **latin.txt** и запишите в него это высказывание на латыни. После этого удалите архив **secret.zip**.
5. Выполните сжатие отдельно для каждого из перечисленных в таблице файлов, используя формат архива, указанный учителем. Вычислите коэффициент сжатия (для этого удобно использовать табличный процессор):

*Архиватор 7-Zip со стандартными настройками:*

Имя файла	Описание	Объем до сжатия, Кб	Объем после сжатия, Кб	Коэффициент сжатия
<b>random.dat</b>	случайные данные	391		
<b>morning.zip</b>	сжатый файл	244		
<b>sunset.jpg</b>	рисунок в формате JPEG	730		
<b>prog.exe</b>	программа для <i>Windows</i>	163		
<b>signal.mp3</b>	звук в формате MP3	137		
<b>forest.wav</b>	звук в формате WAV	609		
<b>ladoga.bmp</b>	рисунок в формате BMP	9217		
<b>tolstoy.txt</b>	текст	5379		

Сделайте выводы о том, какие файлы обычно сжимаются лучше, а какие – хуже:

Ответ:

6. Если ваш архиватор позволяет создавать самораспаковывающиеся архивы, сравните размеры обычного архива и SFX-архива для файла **tolstoy.txt**:

Имя архива	Описание	Объем до сжатия, Кб	Объем после сжатия, Кб
<b>tolstoy.7z</b>	обычный архив	5379	
<b>tolstoy.exe</b>	SFX-архив	5379	

Объясните, почему размеры двух архивов получились разные.

Ответ:

7. Удалите все созданные архивы.
8. Переместите рисунки в отдельный каталог **Pictures**, а звуковые файлы – в каталог **Sounds**.
9. Упакуйте рисунки и звуки в архив **Media** с паролем **media123**.
10. Упакуйте все остальные файлы и папки в архив **Data** (без пароля).
11. Удалите все файлы, кроме архивов **Media** и **Data**, и покажите работу учителю.



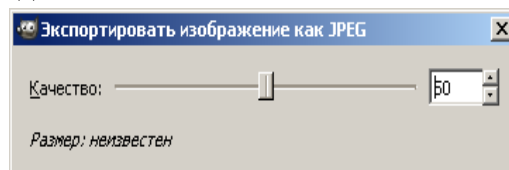
## Практическая работа № 4.

### Сжатие с потерями

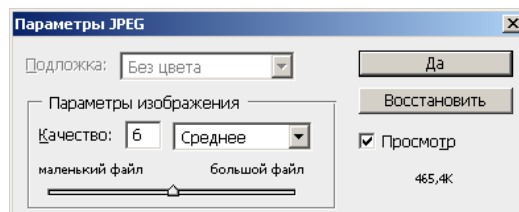
Файлы для выполнения этой работы находятся в каталоге **Lossy**.

1. Скопируйте в свою папку файл **valaam.bmp**.
2. Используя растровый графический редактор (*GIMP, Photoshop*), сохраните несколько копий этого рисунка с разным качеством, от 0% до 100%.

В редакторе *GIMP* нужно выбрать пункт меню *Файл – Экспортировать*, ввести имя файла с расширением **JPG** (например, для файла с качеством 50% можно использовать имя **valaam50.jpg**) и в появившемся окне установить нужное качество:



В редакторе *Photoshop* нужно выбрать пункт меню *Файл – Сохранить как...*, далее в окне сохранения файла выбрать формат **JPEG**, ввести имя файла с расширением **JPG** (например, для файла с качеством 50% можно использовать имя **valaam50.jpg**) и в появившемся окне установить нужное качество (от 0 до 12):



3. В табличном процессоре заполните таблицу

Для *GIMP*:

Качество, %	0	10	20	30	40	50	60	70	80	90	100
Объем файла, Кбайт											

Для *Photoshop*:

Качество, %	0	1	2	3	4	5	6	7	8	9	10	11	12
Объем файла, Кбайт													

С помощью табличного процессора постройте график по этим данным.

График:

Сделайте выводы.

Ответ:

4. Просмотрите файлы, полученные при разных степенях сжатия. Выберите оптимальный на ваш взгляд вариант, когда при небольшом размере файла сохраняется приемлемое качество рисунка.

Ответ:

5. Скопируйте в свою папку звуковой файл **bears.mp3**.
6. Используя звуковой редактор (например, *Audacity*), сохраните несколько копий этого звукового файла с разным качеством. Для формата *Ogg Vorbis* используйте качество от 0 до 10, для формата *MP3* – битрейт от 8 до 128 Кбит/с.
7. В табличном процессоре заполните таблицу

Для формата *Ogg Vorbis*:

Качество	0	1	2	3	4	5	6	7	8	9	10
Объем файла, Кбайт											

Для формата *MP3*:

Битрейт, Кбит/с	8	16	32	48	64	96	128
Объем файла,							

Кбайт							
-------	--	--	--	--	--	--	--

Постройте график по этим данным.

*График:*

--

Объясните, почему получилась именно такая зависимость.

*Ответ:*

--

8. Прослушайте файлы, полученные при разных степенях сжатия. Выберите оптимальный на ваш взгляд вариант, когда при небольшом размере файла сохраняется приемлемое качество звука.

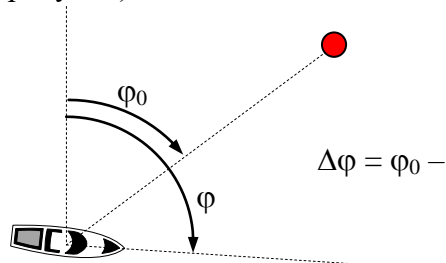
*Ответ:*

--

## Практическая работа № 5.

### Системы управления

Программа **ShipControl.exe**, которая используется в этой работе, позволяет изучить различные законы управления движением судна. Нужно привести судно в район, обозначенный красной точкой; в этом районе находится радиомаяк, так что экипаж в любой момент может определить курс на маяк  $\varphi_0$  и собственный курс судна  $\varphi$  (см. рисунок).



Судно управляется вертикальным рулём, который можно повернуть на угол от  $-35^\circ$  до  $35^\circ$  относительно оси симметрии судна. Положительным будем считать такой угол поворота руля  $\delta$ , который приводит к вращению судна по часовой стрелки (для человека, который смотрит в направлении движения судна, это будет поворот вправо). В ситуации, которая изображена на рисунке, нужно поворачивать влево, поэтому угол поворота руля должен быть отрицательным.

В работе вы попробуете привести судно в заданный район, используя четыре варианта управления судном:

- **ручное управление**, когда вы сами изменяете угол поворота руля;
- авторулевой, использующий **релейный закон управления** – переключение между двумя углами перекладки руля, положительным и отрицательным:

$$\delta = \begin{cases} \delta_R, \varphi < \varphi_0 \\ -\delta_R, \varphi > \varphi_0 \end{cases}$$

Эта запись означает следующее: «если угол  $\varphi$  меньше, чем  $\varphi_0$ , то повернуть руль на угол  $\delta_R$ ; если угол  $\varphi$  больше, чем  $\varphi_0$ , то повернуть руль на угол  $-\delta_R$ ».

- авторулевой, использующий **пропорциональный закон управления (П-регулятор)**, при котором сигнал управления вычисляется как ошибка  $\Delta\varphi = \varphi_0 - \varphi$  (разность между направлением на маяк и направлением движения судна), умноженная на некоторый коэффициент  $k$ :

$$\delta = K(\varphi_0 - \varphi).$$

- авторулевой, использующий **пропорционально-дифференциальный закон управления (ПД-регулятор)**, при котором сигнал управления учитывает не только значение ошибки  $\Delta\varphi$ , но и скорость её изменения  $\omega$ :

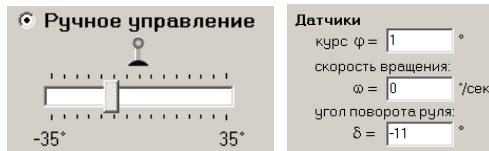
$$\delta = K(\varphi_0 - \varphi) - K_d\omega.$$

Здесь  $K$  и  $K_d$  – коэффициенты регулятора, которые вам предстоит выбрать во время выполнения работы.

Главная проблема состоит в том, что судно – это инерционный объект, поэтому оно не сразу реагирует на поворот руля, а затем, когда оно начнёт поворачиваться, не так просто остановить вращение.

### Уровень А.

1. Запустите программу **ShipControl.exe**. Попробуйте вручную изменять угол поворота руля, перетаскивая мышью рукоятку манипулятора влево и вправо. Убедитесь, что датчик показывает изменение угла поворота руля.



2. Попробуйте управлять положением руля, используя клавиши-стрелки «влево» и «вправо».
3. Щелкните по кнопке  и попытайтесь привести судно в заданный район, управляя им вручную. Если не получилось с первого раза, попробуйте снова. Сделайте не более 5 попыток. Ответьте на вопросы:

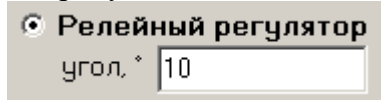
*Удалось ли вам привести судно в заданный район:*

*Добавьте в отчёт скриншот вашей лучшей траектории движения:*

*Чему равна длина пути до заданного района (это число выводится в нижней части окна программы):*

*Как будет двигаться судно, если переложить руль на некоторый угол и больше не менять его положение:*

4. Попробуйте использовать авторулевой с релейным регулятором при  $\delta_R = 10^\circ$ :



*Удалось ли привести судно в заданный район:*

*Сколько времени занял выход в заданную точку:*

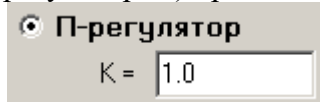
*Добавьте в отчёт скриншот траектории движения:*

5. Уменьшите угол переключки релейного регулятора до  $\delta_R = 4^\circ$  и повторите моделирование.

*Сколько времени занял выход в заданную точку:*

*Что изменилось в поведении судна в сравнении с первым вариантом:*

6. Попробуйте использовать авторулевой с пропорциональным регулятором (П-регулятором) при  $K = 1,0$ :



*Удалось ли привести судно в заданный район:*

*Добавьте в отчёт скриншот траектории движения:*

*Сколько времени занял выход в заданную точку:*

7. Экспериментально определите с точностью до 0,1 минимальный и максимальный коэффициенты усиления П-регулятора, при которых судно достигает заданного района.

	$K_{\min}$	$K$	$K_{\max}$
Значение $K$		1,0	
Число колебаний курса			
Время движения до точки			

Сравните траектории движения при  $K_{\min}$  и  $K_{\max}$  со случаем, когда  $K = 1,0$ . Что происходит при уменьшении и при увеличении  $K$ ?

### Уровень В.

8. Экспериментально определите с точностью до 0,1 коэффициент усиления П-регулятора, при котором судно достигает заданного района за наименьшее время. Запишите в ответе этот коэффициент и время движения.

Добавьте в отчёт скриншот траектории движения:

9. Попробуйте использовать авторулевой с ПД-регулятором при  $K = 1,2$  и  $K_d = 100$ :

П-регулятор  
 $K =$

ПД-регулятор  
 ОС по скорости вращения  
 $K_d =$

Удалось ли привести судно в заданный район:

Добавьте в отчёт скриншот траектории движения:

Сколько времени занял выход в заданную точку:

Чем отличается результат управления при использовании П- и ПД-регуляторов:

Какой регулятор вы бы выбрали? Почему?

### Уровень С.

10. Перейдите на вкладку *Модель* и выберите модель судна-контейнеровоза. Чем отличается эта модель от модели учебного судна?

11. Попробуйте вывести судно в нужную точку на ручном управлении. Получилось ли у вас?

Как вы думаете, почему управлять контейнеровозом сложнее, чем учебным судном:

12. Попробуйте вывести судно в нужную точку с помощью релейного управления, пробуя разные углы перекладки руля. Получилось ли у вас? Если да, вставьте в отчёт скриншот.

13. Попробуйте вывести судно в нужную точку с помощью П-регулятора, пробуя значения коэффициентов.

Получилось ли у вас? При каком значении  $K$ ?

Если да, вставьте в отчёт скриншот.

14. Попробуйте вывести судно в нужную точку с помощью ПД-регулятора с настройками

П-регулятор  
K =

ПД-регулятор  
ОС по скорости вращения  
Kd =

Получилось ли у вас? Если да, вставьте в отчёт скриншот.

Сколько времени потребовалось?

Какой регулятор вы бы выбрали? Почему?

#### Уровень D.

15. Экспериментально найдите значения коэффициентов ПД-регулятора, при которых время выхода в заданную точку наименьшее. Можно сначала изменять один из коэффициентов (например,  $K_d$  с шагом 10), и найти лучший вариант. Затем, зафиксировав  $K_d$ , попробуйте изменять коэффициент  $K$  так, чтобы ещё улучшить результат.

Коэффициенты оптимального регулятора:

K =  
K<sub>d</sub> =

Добавьте в отчёт скриншот траектории движения:

Сколько времени занял выход в заданную точку:

**Глава 11. Моделирование**  
**Практические работы**  
**Практическая работа № 6.**  
**Моделирование работы процессора**  
**Уровень А.**

1. Напишите программу, которая моделирует работу процессора. Процессор имеет 4 регистра, они обозначаются R0, R1, R2 и R3. Все команды состоят из трех десятичных цифр: код операции, номер первого регистра и номер второго регистра (или число от 0 до 9). Коды команд и примеры их использования приведены в таблице:

Код операции	Описание	Пример	Псевдокод
1	запись константы	128	<b>R2 := 8</b>
2	копирование значения	203	<b>R3 := R0</b>
3	сложение	331	<b>R1 := R1 + R3</b>
4	вычитание	431	<b>R1 := R1 - R3</b>

Обратите внимание, что результат записывается во *второй* регистр. Команды вводятся последовательно как символьные строки. После ввода каждой строки программа показывает значения всех регистров.

**Уровень В.**

2. Добавьте в систему команд умножение, деление и логические операции с регистрами – «И», «ИЛИ», «исключающее ИЛИ».
3. Добавьте в систему команд логическую операцию «НЕ». Подумайте, как можно использовать второй регистр.

**Уровень С.**

4. Сделайте так, чтобы в команде с кодом 1 можно было использовать шестнадцатеричные значения констант (0-9, A-F).

**Уровень D+.**

5. Добавьте обработку ошибок типа «неверная команда», «неверный номер регистра», «деление на ноль».
6. \*Добавьте команду «СТОП», которая прекращает работу программы. Введите строковый массив, моделирующий память, и запишите в него программу – последовательность команд. Ваша программа должна последовательно выполнять эти команды, выбирая их из «памяти», пока не встретится команда «СТОП».
7. \*\*Подумайте, как можно было бы организовать условный переход: перейти на N байт вперед (или назад), если результат последней операции – ноль.

## Практическая работа № 7.

### Искусственный интеллект

#### Уровень А.

1. Откройте браузер на странице [quickdraw.withgoogle.com](http://quickdraw.withgoogle.com). Попробуйте выполнить задание – нарисовать несколько предметов, которые нейронная сеть попытается угадать. Сколько из нарисованных вами изображений нейронная сеть определила правильно?

Ответ:

--

2. С помощью сервиса [color.artlebedev.ru](http://color.artlebedev.ru) раскрасьте чёрно-белое изображение (можно использовать файл **boat.jpg** или своё изображение) и добавьте в отчёт оригинал и полученное цветное изображение.

Оригинал	Цветной вариант

3. С помощью сервиса [www.captionbot.ai](http://www.captionbot.ai), использующего нейронные сети, постройте подпись к вашему изображению и переведите её на русский язык.

Оригинал	На русском языке

#### Уровень В.

4. С помощью сервиса [www.how-old.net](http://www.how-old.net) попытайтесь определить возраст человека, который изображен на фотографии (можно использовать свою фотографию или изображение **girl.jpg**).
5. С помощью сервиса [hi.cs.waseda.ac.jp:8082](http://hi.cs.waseda.ac.jp:8082) выполните раскраску чёрно-белого изображения.

Оригинал	Цветной вариант

Сравните результат с тем, который был получен в п. 2.

6. С помощью сервиса [bigjpg.com/ru](http://bigjpg.com/ru) выполните увеличение какого-нибудь изображения в 4 раза (можно использовать файл **flowers.png**). Добавьте в отчёт оригинал и полученное увеличенное изображение.

Оригинал	Увеличенное изображение

#### Уровень С.

7. Установите практикум по искусственному интеллекту с сайта [www.lbai.ru](http://www.lbai.ru) (это нужно сделать заранее).
8. Выполните лабораторные работы 1-4
9. Зарегистрируйтесь на сайте [letsenhance.io/ru/](http://letsenhance.io/ru/). С помощью этого сервиса попробуйте улучшить фотографию и увеличить её в 4 раза.

Оригинал	Увеличенное изображение



## Практическая работа № 8.

### Математическое моделирование

При выполнении этой работы можно использовать шаблон – электронную таблицу в файле **Кредит.xls**.

#### Уровень А.

1. Дядя Ваня решил взять кредит 100 тыс. рублей в банке на год под 13% годовых. Банк предложил ему выплачивать каждый месяц равные суммы (такой платёж называется **аннуитетным**). Сумма ежемесячной выплаты  $p$  вычисляется по формуле:

$$p = S \cdot i \cdot \left( 1 + \frac{1}{(1+i)^n - 1} \right),$$

где  $S$  – сумма кредита,  $i$  – годовая процентная ставка кредита, делённая на 12;  $n$  – число месяцев, в течение которых нужно выплатить кредит.

Используя электронную таблицу или программу на языке программирования, определите, сколько всего денег выплатит дядя Ваня банку по процентам (сверх 100 тыс. рублей).

Ответ:

#### Уровень В.

2. Определите с точность до 0,1% наибольшую величину процентной ставки, при которой дядя Ваня выплатит по процентам не более 10000 руб.

Ответ:

Какова в этом случае будет величина ежемесячной выплаты:

3. Определите с точность до 0,1% наибольшую величину процентной ставки, при которой дядя Ваня будет ежемесячно выплачивать не более 9100 руб.

Ответ:

Какова в этом случае будет общая сумма выплат:

#### Уровень С.

4. Существует ещё один вариант выплаты кредита, при котором выплачиваемая сумма изменяется (уменьшается) каждый месяц. Такая схема называется **дифференцированными платежами**. Выплаты за текущий месяц складываются из двух частей – выплат  $p_1$  за основную часть кредита (100 тыс. руб) и выплат по процентам  $p_2$ .

Первая часть,  $p_1$ , всегда постоянна и определяется как сумму кредита, делённая на количество месяцев. Для примера, который рассмотрен в начале описания этой работы,

$$p_1 = 100000 : 12 = 8333,33 \text{ р.}$$

Вторая часть,  $p_2$ , зависит от номера месяца  $k$ , поэтому будем обозначать её как  $p_{2k}$ . Она вычисляется как оставшаяся сумма основного долга, умноженная на годовую процентную ставку кредита, делённую на 12:

$$p_{2k} = S_k \cdot i$$

Например, в первый месяц выплат  $S_1 = S$  (выплат ещё не было), поэтому

$$p_{21} = 100000 \cdot 0,13 / 12 = 1083,33 \text{ р.}$$

Во второй месяц оставшаяся сумма долга равна

$$S_1 = 100000 - 8333,33 = 91666,67 \text{ р.}$$

При этом получается другая выплата по процентам:

$$p_{22} = 91666,67 \cdot 0,13 / 12 = 993,06 \text{ р.}$$

5. При ставке 13% постройте график выплат дяди Вани при дифференцированных платежах. Сколько всего денег он переплатит по процентам?

*Ответ:*

*Какой способ выплат выгоднее для дяди Вани:*

6. Определите с точность до 0,1% наибольшую величину процентной ставки, при которой дядя Ваня выплатит по процентам не более 10000 руб.

*Ответ:*

*Какова в этом случае будет величина выплаты в первый месяц:*

7. Определите с точность до 0,1% наибольшую величину процентной ставки, при которой дядя Ваня будет ежемесячно выплачивать не более 9100 руб.

*Ответ:*

*Какова в этом случае будет общая сумма выплат:*

## Практическая работа № 9.

### Моделирование движения

При выполнении этой работы можно использовать шаблон – электронную таблицу в файле **Парашютист.xls**.

#### Уровень А.

Парашютист массой 90 кг разгоняется в свободном падении до скорости 10 м/с и на высоте 50 м раскрывает парашют, площадь которого 55 м<sup>2</sup>. Коэффициент сопротивления парашюта  $C = 0,9$ .

1. Постройте графики изменения скорости и высоты полета в течение первых 4 секунд;

Ответ:

2. Определите, через какое время и с какой скоростью приземлится парашютист?

Время:

Скорость:

3. Сравните результаты моделирования с установившимся значением скорости, вычисленным теоретически.

	теоретически	моделирование
Скорость приземления, м/с	5,40	5,40

#### Уровень В.

4. Напишите программу, которая моделирует полет мяча, брошенного вертикально вверх, при

$$r = 33\text{мм}, m = 150\text{г}, v_0 = 20\text{м/с}, \delta = 0,1\text{с}.$$

Остальные необходимые данные есть в тексте § 10. Выполните следующие задания:

- определите время полета, максимальную высоту подъема мяча и скорость в момент приземления;
- вычислите время полета и максимальную высоту подъема мяча, используя модель движения без сопротивления воздуха:

$$t = \frac{2v_0}{g}, \quad h = \frac{v_0^2}{2g}, \quad v = -v_0.$$

- сравните эти результаты с полученными при моделировании с учетом сопротивления;

	без учёта сопротивления	с учётом сопротивления	ошибка (%)
Время полета, с			
Максимальная высота, м			
Скорость приземления, м/с			

- можно ли в этой задаче пренебречь сопротивлением воздуха? почему?

Ответ:

- с помощью табличного процессора постройте траекторию движения мяча, а также графики изменения скорости, ускорения и силы сопротивления (указание: из программы сохраните данные в CSV-файл, разделив данные в каждой строке точками с запятой);

Скорость:

Ускорение:

Сила сопротивления:

- уменьшите шаг сначала до 0,01 с и повторите моделирование; затем выполните моделирование при шаге 0,001; сделайте выводы по поводу выбора шага в данной задаче.

Ответ:

### Уровень С.

5. \*Выполните моделирование движения мяча, брошенного под углом  $45^\circ$  к горизонту (при тех же остальных условиях, что и в задаче уровня В):

- определите время полета, максимальную высоту и дальность полета мяча, скорость в момент приземления;
- сравните результаты со случаем, когда сопротивление воздуха не учитывается

	без учёта сопротивления	с учётом сопротивления	ошибка (%)
Время полета, с			
Максимальная высота, м			
Дальность полета, м			
Скорость приземления, м/с			

- сделайте выводы.

Ответ:

## Практическая работа № 10.

### Моделирование развития популяции

Для выполнения работы откройте файл-заготовку *Популяция.xls*.

#### Уровень А.

1. Постройте графики изменения численности популяции животных для моделей ограниченного и неограниченного роста при  $N_0 = 100$ ,  $K = 0,5$  и  $L = 1000$  в течение первых 15 периодов. Определите, когда модель неограниченного роста перестает быть адекватной (отклонение от модели ограниченного роста составляет более 10%).

Ответ:

2. Используя подбор параметра, определите, при каких коэффициентах  $K$  модель неограниченного роста остается адекватной в течение не менее 10 периодов.

Ответ:

3. Используя модель ограниченного роста из предыдущей задачи, выполните моделирование популяции с учетом отлова ( $R = 100$ ). Предполагается, что животных начали отлавливать через 10 лет после начала наблюдений.

- постройте график изменения численности животных в течение 40 периодов;

- Определите количество животных в состоянии равновесия по результатам моделирования;

Ответ:

- Проверьте, зависит ли это значение от начальной численности?

Ответ:

#### Уровень В.

4. Определите количество животных в состоянии равновесия теоретически, из модели ограниченного роста с отловом; сравните это значение с результатами моделирования

Ответ:

5. Определите, на что влияет начальная численность животных;

Ответ:

6. Определите (по результатам моделирования) максимальный отлов  $R$ , при котором популяция не вымирает.

Ответ:

#### Уровень С.

7. \*Определите максимально допустимый отлов теоретически, из модели ограниченного роста с отловом; сравните это значение с результатами моделирования

Ответ:

Из какого условия вы нашли это значение?

## Практическая работа № 11.

### Моделирование эпидемии

Для выполнения работы откройте файл-заготовку *Эпидемия.xls*.

При эпидемии гриппа число больных  $N$  изменяется по формуле

$$N_{i+1} = N_i + Z_{i+1} - V_{i+1},$$

где  $Z_i$  – количество заболевших в  $i$ -й день, а  $V_i$  – количество выздоровевших в тот же день. Число заболевших рассчитывается согласно модели ограниченного роста:

$$Z_{i+1} = K \cdot \frac{L - N_i - W_i}{L} \cdot N_i,$$

где  $L$  – общая численность жителей,  $K$  – коэффициент роста и  $W_i$  – число переболевших (тех, кто уже переболел и выздоровел, и поэтому больше не заболеет):

$$W_{i+1} = W_i + V_{i+1}.$$

Считается, что в начале эпидемии заболел 1 человек, все заболевшие выздоравливают через 7 дней и больше не болеют.

Выполните моделирование развития эпидемии при  $L = 1000$  и  $K = 0,5$  до того момента, когда количество больных станет равно нулю.

#### Уровень А.

1. Постройте график изменения количества больных.

Ответ:

2. Когда закончится эпидемия?

Ответ:

3. Сколько человек переболеет, а сколько вообще не заболеет гриппом?

Ответ:

4. Каково максимальное число больных в один день?

Ответ:

#### Уровень В.

5. Изменяя коэффициент  $K$  с шагом 0,1, определите, при каких значениях  $K$  модель явно перестает быть адекватной.

Ответ:

#### Уровень С.

6. \*Сравните модель, использованную в этой работе, со следующей моделью:

$$N_{i+1} = \left( 1 + K \cdot \frac{L - N_i - W_i}{L} \right) \cdot N_i, \quad W_{i+1} = W_i + N_{i-7}.$$

Постройте график изменений количества больных этой и предыдущей моделями на одном поле:

Анализируя результаты моделирования, докажите, что эта модель неадекватна.

Ответ:

Какие допущения, на ваш взгляд, были сделаны неверно при разработке этой модели?

Ответ:

Сравните поведение двух моделей при  $K = 0$ ,  $K = 0,3$  и  $K = 1$ . Сделайте выводы.

Ответ:

## Практическая работа № 12.

### Модель «хищник-жертва»

Для выполнения работы откройте файл-заготовку *ХищникЖертва.xls*.

Выполните моделирование биологической системы «щуки-караси»

$$N_{i+1} = \left( 1 + K \cdot \frac{L - N_i}{L} - b_N \cdot Z_i \right) \cdot N_i$$

$$Z_{i+1} = (1 - D + b_Z \cdot N_i) \cdot Z_i$$

где  $N_i$  – численность карасей

$Z_i$  – численность щук

при следующих значениях параметров:

$K = 1$  – коэффициент прироста карасей;

$L = 100$  – предельная численность карасей;

$N_0 = 50$  – начальная численность карасей;

$Z_0 = 10$  – начальная численность щук;

$D = 0,8$  – коэффициент смертности щук без пищи;

$b_N = 0,01$  и  $b_Z = 0,012$  – коэффициенты модели.

Постройте на одном поле графики изменения численности карасей и щук в течение 30 периодов моделирования.

#### Уровень А.

9. Выполните моделирование двух популяций в течение 25 периодов. Постройте графики изменения численности карасей и щук.

Ответ:

10. Сколько карасей и щук живут в водоеме в состоянии равновесия?

Ответ:

11. Что влияет на количество рыб в состоянии равновесия: начальная численность хищников и жертв или значения коэффициентов модели?

Ответ:

#### Уровень В.

12. На что влияет начальная численность хищников и жертв?

Ответ:

13. Подберите значения коэффициентов, при которых модель становится неадекватна.

Ответ:

$b_N =$
$b_Z =$

#### Уровень С.

14. Установите значения коэффициентов  $b_N = 0,008$  и  $b_Z = 0,018$ . Вставьте в отчёт график, который вы увидели:

Как вы можете объяснить это с точки зрения биологии?

Ответ:

15. Установите значения коэффициента  $b_N = 0,01$ . Найдите экспериментально наибольшее значения коэффициента  $b_Z$  (с точностью 0,001), при котором к 25 году щуки вымирают, а численность карасей достигает предельно возможного значения.

*Ответ:*

$bZ =$

Как вы можете объяснить это с точки зрения биологии?

*Ответ:*



## Практическая работа № 11а.

### Модель «две популяции»

Для выполнения работы откройте файл-заготовку *ДвеПопуляции.xls*.

Белки и бурундуки живут в одном лесу и едят примерно одно и то же (конкурируют за пищу). Модель, описывающая изменение численности двух популяций, имеет вид:

$$N_{i+1} = \left( 1 + K_N \cdot \frac{L_N - N_i}{L_N} \right) \cdot N_i - D_N \cdot M_i$$

$$M_{i+1} = \left( 1 + K_M \cdot \frac{L_M - M_i}{L_M} \right) \cdot M_i - D_M \cdot N_i$$

Здесь  $N$  и  $M$  – численность белок и бурундуков;  $L_N$  и  $L_M$  – их максимальные численности;  $K_N$  и  $K_M$  – коэффициенты прироста;  $D_N$  и  $D_M$  – коэффициенты взаимного влияния.

Объясните, на основании каких предположений была построена эта модель.

Ответ:

Выполните моделирование изменения численности двух популяций в течение 30 периодов при  $N_0 = 10$ ,  $M_0 = 20$ ,  $L_N = 70$ ,  $L_M = 50$ ,  $K_N = K_M = 0,7$  и  $D_N = D_M = 0,1$ .

#### Уровень А.

1. Постройте графики изменения численности обеих популяций на одном поле.

2. Является ли эта модель системной? Почему?

Ответ:

3. Какова численность белок и бурундуков в состоянии равновесия?

Ответ:

4. Что влияет на состояние равновесия, коэффициенты модели или начальная численность?

Ответ:

5. На что влияет начальная численность животных?

Ответ:

#### Уровень В.

6. При каком значении коэффициента  $D_M$  через 30 лет останется 30 бурундуков? (используйте подбор параметра).

Ответ:

7. Сколько тогда будет белок?

8. Найдите какие-нибудь значения коэффициента  $D_M$ , при котором модель становится неадекватна. Почему вы так считаете?

Ответ:

#### Уровень С.

9. \*Предложите аналогичную модель взаимного влияния трех видов.

Ответ:

### Практическая работа № 13.

#### Саморегуляция

Для выполнения работы откройте файл-заготовку *Саморегуляция.xls*.

Биологи выяснили, что для каждого вида животных существует некоторая минимальная численность популяции, которая необходима для выживания этой колонии. Это может быть одна пара животных (например, для ондатр) или даже тысячи особей (для американских почтовых голубей). Если количество животных становится меньше этого минимального значения, популяция вымирает. Для этого случая предложена следующая модель изменения численности:

$$N_{i+1} = \left( 1 + K \cdot \frac{L - N_i}{L} \cdot \frac{N_i - M}{N_i + F} \right) N_i, \quad (*)$$

Эта модель отличается от модели ограниченного роста только дополнительным множителем  $\frac{N_i - M}{N_i + F}$ , где  $M$  и  $F$  – некоторые числа (параметры), смысл которых вам предстоит выяснить.

#### Уровень А.

1. Выполните моделирование для 30 периодов при следующих значениях параметров модели:

$$N_0 = 1500, L = 1000, K = 0,5, M = 300, F = 0.$$

Постройте на одном графике кривые, показывающие изменение численности по классической модели ограниченного роста и по модели (\*).

Сделайте выводы и опишите, в чём проявляется саморегуляция для этих моделей.

Ответ:

2. Постройте аналогичные графики для  $F = 500$ :

Сделайте вывод о том, как коэффициент  $F$  на саморегуляцию.

3. Через 10 периодов в результате изменения природных условия число животных уменьшилось до 400 (то есть,  $N_{10} = 400$ ). Используйте значение  $F = 500$ . Выполните моделирование при этих условиях и постройте графики:

Опишите, чем отличается поведение двух сравниваемых моделей.

Как влияет изменение  $F$  на переходные процессы?

#### Уровень В.

4. Повторите моделирование при  $N_{10} = 250$  и постройте графики:

Сделайте выводы:

#### Уровень С.

5. Экспериментируя с моделями, найдите минимальную численность популяции  $N_{\min}$ , при которой она выживает в соответствии с моделью (\*).

Ответ:

6. Сделайте выводы о смысле коэффициента  $M$  в модели (\*).

*Ответ:*

7. Сравните свойства саморегуляции для модели ограниченного роста и модели (\*).

*Ответ:*

## Практическая работа № 14.

### Методы Монте-Карло

#### Уровень А.

1. Используя метод Монте-Карло, приближённо определите значение числа  $\pi$  (см. § 12 учебника). Запишите в таблицу полученные результаты при различных значениях  $N$ :

N	100	1000	10000	100000	1000000
Результат ( $\approx \pi$ )					
Ошибка $\delta$ , %					

Относительная ошибка в процентах вычисляется по формуле

$$\delta = \frac{|\bar{\pi} - \pi|}{\pi} \cdot 100\% ,$$

где  $\pi = 3,1415926\dots$  и  $\bar{\pi}$  – приближённое значение числа  $\pi$ , полученное в результате моделирования.

#### Уровень В.

2. Не используя значение  $\pi$ , с помощью метода Монте-Карло вычислите площади кругов различного диаметра:

$R$	1	2	3	4	5
$S$					

3. С помощью электронных таблиц постройте по этим данным зависимость  $S$  от  $R$ .
4. Используя линию тренда, подберите функцию, которая лучше всего соответствует этой зависимости.

Ответ:

$S = \dots$

5. Сравните эту функцию с известной вам формулой площади круга. Сделайте выводы.

Выводы:

#### Уровень С.

6. Не используя значение  $\pi$ , с помощью метода Монте-Карло вычислите объёмы шаров различного диаметра:

$R$	1	2	3	4	5
$V$					

7. С помощью электронных таблиц постройте по этим данным зависимость  $V$  от  $R$ .
8. Используя линию тренда, подберите функцию, которая лучше всего соответствует этой зависимости.

Ответ:

$V = \dots$

9. Сравните эту функцию с известной вам формулой объёма шара. Сделайте выводы.

Выводы:

## Практическая работа № 15. Системы массового обслуживания

Для моделирования обслуживания клиентов в банке предложена следующая модель:

- за 1 минуту в банк входит случайное число клиентов, от 0 до  $P_{\max}$  (распределение равномерное);
- на обслуживание клиентов требуется от  $T_{\min}$  до  $T_{\max}$  минут; время обслуживания  $T$  определяется для каждой рабочей минуты случайным образом (распределение равномерное);
- моделирование выполняется для интервала времени  $L$ , равного 8-часам (рабочая смена).

- число клиентов, находящихся в помещении банка, вычисляется по формуле

$$N_{i+1} = N_i + P_i - R_i$$

где  $P_i$  – количество клиентов, вошедших за  $i$ -ую минуту, а  $R_i$  – количество клиентов, обслуженных за это время;

- если кассир обслуживает клиента за  $T$  минут, то можно считать, что за 1 минуту он сделает часть работы, равную  $\frac{1}{T}$ ; если предположить, что скорость работы кассиров

одинакова, то  $K$  касс за 1 минуту обслужат  $\frac{K}{T}$  клиентов;

- если считать, что  $N$  клиентов равномерно распределяются по  $K$  кассам, так что средняя длина очереди равна  $Q = \frac{N}{K}$ , а среднее время ожидания в течение этой минуты равно

$$\Delta t = Q \cdot T = \frac{N}{K} \cdot T$$

- достаточным считается число касс, при которых среднее время ожидания  $\Delta t$  превышает установленный предел  $M$  не более, чем 5% рабочего времени в течение дня.

Используя эту вероятностную модель работы банка, напишите программу, с помощью которой определите минимальное необходимое количество касс при следующих исходных данных:

$$P_{\max} = 4, T_{\min} = 1, T_{\max} = 9, M = 15.$$

**Глава 12. Базы данных**  
**Практические работы**  
**Практическая работа № 0.**  
**Работа с готовой таблицей**

8. Откройте базу данных *Учебники.odt*<sup>4</sup> и изучите её структуру. Сколько таблиц она содержит?

Ответ:

9. Откройте таблицу *Учебники*.

10. Найдите любой учебник, выпущенный издательством «Вита-Пресс».

11. Используя «быстрый фильтр» (фильтр по выделенному), найдите все учебники, которые выпустило издательство «Вита-Пресс». Сколько записей было отобрано?

Ответ:

12. Отсортируйте отобранные записи в алфавитном порядке по полю *Авторы*, а учебники одного и того же автора – по номеру класса (в порядке убывания). Покажите работу учителю.

13. Измените фильтр так, чтобы отобрать все учебники по русскому языку издательства «Дрофа». Сколько записей было отобрано?

Ответ:

14. Добавьте в базу данных информацию о новом учебнике: *Минаева С.С., Рослова Л.О., Рыдзе О.А. и. Математика. 1 класс. – М.: «Вентана-Граф», 2013.*

15. Используя фильтр и сортировку<sup>5</sup>, определите, сколько учебников выпустили издательства «Дрофа», «Вита-Пресс» и «Вентана-Граф» для 1-ого класса. Подсказка: программа автоматически считает число выделенных записей.

Ответ:

---

<sup>4</sup> Если на уроках используется программа *Microsoft Access*, нужно открывать файлы с теми же именами и расширением **.mdb**.

<sup>5</sup> В программе *Microsoft Access* для этого достаточно использовать расширенный фильтр.

## Практическая работа № 16.

### Язык SQL (OpenOffice Base)

Язык *SQL* (*Structured Query Language*) – это специальный язык для управления данными. С его помощью можно полностью контролировать базу данных и выполнять все операции, посылая запросы (команды) в текстовом виде. Таким образом, например, можно управлять базой данных с удаленного компьютера.

В этой работе вы познакомитесь с основными командами языка *SQL*:

<b>CREATE TABLE</b>	создать таблицу
<b>SELECT</b>	выбрать данные
<b>UPDATE</b>	изменить данные
<b>DELETE</b>	удалить данные
<b>DROP</b>	удалить таблицу

1. Создайте новую пустую базу данных *SQLbase.odt*.

#### Создание и заполнение таблиц

2. Выберите пункт верхнего меню *Сервис – SQL* и введите следующую команду для создания таблицы:

```
CREATE TABLE "Туры" (  
  "Код" INTEGER NOT NULL PRIMARY KEY,  
  "Страна" VARCHAR(50) NOT NULL,  
  "Транспорт" VARCHAR(20) NOT NULL,  
  "Цена" DECIMAL(20,2) NOT NULL )
```

В этой команде требуется создать таблицу (**CREATE TABLE**) с именем «Туры». В таблице должно быть 4 поля:

*Код* – целое число (**INTEGER**), непустое (**NOT NULL**), первичный ключ таблицы (**PRIMARY KEY**)

*Страна* – строка длиной до 50 символов, непустое

*Транспорт* – строка длиной до 20 символов, непустое

*Цена* – поле для хранения денежной суммы в виде числа с 20 значащими цифрами, из них 2 – в дробной части (**DECIMAL(20,2)**)

#### Названия таблиц и полей заключаются в двойные кавычки!

3. Выполните эту команду (кнопка *Выполнить*). Для того, чтобы увидеть созданную таблицу, нужно закрыть базу данных и открыть её заново. Проверьте, что таблица действительно создана.
4. Зайдите в режим *Конструктора* и проверьте свойства полей таблицы, которые мы задали с помощью команды **CREATE TABLE**.
5. Зайдите в окно *SQL* и выполните команду для добавления в базу новой записи:

```
INSERT INTO "Туры"  
  VALUES(1, 'Финляндия', 'автобус', 1200)
```

Эта команда вставляет (**INSERT**) в таблицу «Туры» одну запись. После ключевого слова **VALUES** в скобках перечислены через запятую значения полей в том порядке, в котором они задавались при создании таблицы.

#### Символьные строки в значениях полей заключаются в апострофы!

6. Выполните ещё одно добавление записи:

```
INSERT INTO "Туры"  
  VALUES (1, 'Норвегия', 'самолёт', 15000)
```

Какая ошибка произошла? В чём её причина?

Ответ:

Исправьте ошибку и добавьте новую запись правильно. В ответе запишите SQL-запрос.

Ответ:

7. Аналогично добавьте в таблицу ещё несколько записей:

Страна	Транспорт	Цена
Швеция	паром	9000 р.
Германия	автобус	15700 р.
Греция	самолёт	23000 р.
Норвегия	автобус	8000 р.
Германия	самолёт	19000 р.

### Выбор данных и сортировка

8. Создайте новый запрос в режиме SQL<sup>6</sup>

```
SELECT * FROM "Туры"
```

и выполните его. Посмотрите на результат.

Этот оператор выберет все поля (\*) всех записей из таблицы «Туры».

9. Вместо \* можно указать через запятую список нужных полей:

```
SELECT "Страна", "Цена" FROM "Туры"
```

Проверьте результат выполнения этого запроса.

10. Чаще всего нужно выбрать только записи, удовлетворяющие некоторому условию отбора. Для этого используется ключевое слово **WHERE**, после которого стоит условие:

```
SELECT * FROM "Туры" WHERE "Страна" = 'Норвегия'
```

Проверьте работу этого оператора.

11. Составьте запрос, который выбирает из таблицы «Туры» значения полей «Страна», «Транспорт» и «Цена» для всех автобусных туров:

Ответ:

12. Составьте запрос, который выбирает из таблицы «Туры» значения всех полей для туров с ценой меньше 10000 руб:

Ответ:

13. Для того, чтобы отсортировать данные по некоторому полю, в запросе после ключевых слов **ORDER BY** (англ. «упорядочить по») указывают название этого поля:

```
SELECT * FROM "Туры" ORDER BY "Цена"
```

Проверьте работу этого запроса.

Если в конце предыдущего запроса добавить слово **DESC** (англ. «descending» – нисходящий), сортировка выполняется в обратном порядке.

14. Составьте запрос, который выбирает из таблицы «Туры» значения всех полей для туров с ценой больше 10000 руб. и сортирует результаты по убыванию цены:

Ответ:

15. В запросах можно использовать стандартные функции. Например, функция **MIN** определяет минимальное значение заданного поля среди всех записей:

```
SELECT MIN("Цена") FROM "Туры"
```

Результат этого запроса – одно число.

16. Составьте запрос, который находит минимальную цену для туров в Норвегию:

Ответ:

17. Результаты запросов можно использовать в других запросах – получается вложенный запрос. Например, запрос

```
SELECT * FROM "Туры" WHERE "Цена" =
```

<sup>6</sup> В LibreOffice для этого можно использовать окно *Сервис – SQL*, отметив флажок «Показать вывод операторов SELECT».



(SELECT MIN("Цена") FROM "Туры")

вернет данные о самом дешевом туре.

18. Составьте запрос, который находит тур минимальной цены на самолёте:

Ответ:

### Изменение и удаление данных

19. Для изменения записей используется оператор **UPDATE**. Запрос, приведенный ниже, увеличивает цены всех туров на 10%:

**UPDATE "Туры" SET "Цена" = "Цена"\*1.1**

Проверьте, что данные в таблице «Туры» действительно изменились.

20. Запишите запрос, который вернёт цены в исходное состояние, и выполните его.

Ответ:

21. Авиакомпании в данный момент представляют скидку на билеты, так что цены всех туров на самолётах составляют 80% от исходных. Составьте и выполните соответствующий запрос.

Ответ:

Какая стоимость получилась у тура в Грецию?

Ответ:

22. Скопируйте таблицу «Туры», назвав копию «Туры2». Удалите все туры в Германию с помощью запроса

**DELETE FROM "Туры2" WHERE "Страна" = 'Германия'**

Проверьте, что данные в таблице «Туры2» действительно изменились.

23. Удалите таблицу «Туры2», которая больше не нужна, с помощью запроса

**DROP TABLE "Туры2"**

Попытайтесь открыть таблицу «Туры2». Загрузить базу данных заново и убедитесь, что таблица «Туры2» действительно удалена.

24. Оператор **SCRIPT** позволяет сохранить базу данных в текстовом формате для её переноса в другую СУБД. Сохраните базу с помощью запроса

**SCRIPT 'имя файла'**

Откройте полученный файл в текстовом редакторе и изучите его.

## Практическая работа № 16-Access.

### Язык SQL (Microsoft Access)

Язык SQL (*Structured Query Language*) – это специальный язык для управления данными. С его помощью можно полностью контролировать базу данных и выполнять все операции, посылая запросы (команды) в текстовом виде. Таким образом, например, можно управлять базой данных с удаленного компьютера.

В этой работе вы познакомитесь с основными командами языка SQL:

<b>CREATE TABLE</b>	создать таблицу
<b>SELECT</b>	выбрать данные
<b>UPDATE</b>	изменить данные
<b>DELETE</b>	удалить данные
<b>DROP</b>	удалить таблицу

1. Создайте новую пустую базу данных *SQLbase.accdb*.

#### Создание и заполнение таблиц

2. Создайте запрос в режиме конструктора и перейдите в режим SQL. Введите следующую команду для создания таблицы:

```
CREATE TABLE [Туры] (  
[Код] INTEGER NOT NULL PRIMARY KEY,  
[Страна] VARCHAR(50) NOT NULL,  
[Транспорт] VARCHAR(20) NOT NULL,  
[Цена] MONEY NOT NULL )
```

В этой команде требуется создать таблицу (**CREATE TABLE**) с именем «Туры». В таблице должно быть 4 поля:

*Код* – целое число (**INTEGER**), непустое (**NOT NULL**), первичный ключ таблицы (**PRIMARY KEY**)

*Страна* – строка длиной до 50 символов, непустое

*Транспорт* – строка длиной до 20 символов, непустое

*Цена* – поле для хранения денежной суммы (**MONEY**)

**Названия таблиц и полей заключаются в квадратные скобки! Если эти названия состоят из одного слова, скобки можно не ставить:**

```
CREATE TABLE Туры (  
Код INTEGER NOT NULL PRIMARY KEY,  
Страна VARCHAR(50) NOT NULL,  
Транспорт VARCHAR(20) NOT NULL,  
Цена MONEY NOT NULL)
```

3. Выполните эту команду (вкладка *Конструктор* – *Выполнить*). Проверьте, что таблица действительно создана.
4. Выполните команду для добавления в базу новой записи:

```
INSERT INTO Туры  
VALUES (1, 'Финляндия', 'автобус', 1200)
```

Эта команда вставляет (**INSERT**) в таблицу «Туры» одну запись. После ключевого слова **VALUES** в скобках перечислены через запятую значения полей в том порядке, в котором они задавались при создании таблицы.

**Символьные строки в значениях полей заключаются в апострофы или двойные кавычки!**

5. Выполните ещё одно добавление записи:

```
INSERT INTO Туры  
VALUES (1, 'Норвегия', 'самолёт', 15000)
```

Какая ошибка произошла? В чём её причина?

Ответ:

Исправьте ошибку и добавьте новую запись правильно. В ответе запишите SQL-запрос.

Ответ:

6. Аналогично добавьте в таблицу ещё несколько записей:

Страна	Транспорт	Цена
Швеция	паром	9000 р.
Германия	автобус	15700 р.
Греция	самолёт	23000 р.
Норвегия	автобус	8000 р.
Германия	самолёт	19000 р.

### Выбор и сортировка данных

7. Выполните запрос на выборку данных:

**SELECT \* FROM Туры**

Посмотрите на результат. Этот оператор выберет все поля (\*) всех записей из таблицы «Туры».

8. Вместо \* можно указать через запятую список нужных полей:

**SELECT Страна, Цена FROM Туры**

Проверьте результат выполнения этого запроса.

9. Чаще всего нужно выбрать только записи, удовлетворяющие некоторому условию отбора. Для этого используется ключевое слово **WHERE**, после которого стоит условие:

**SELECT \* FROM Туры WHERE Страна = 'Норвегия'**

Проверьте работу этого оператора.

10. Составьте запрос, который выбирает из таблицы «Туры» значения полей «Страна», «Транспорт» и «Цена» для всех автобусных туров:

Ответ:

11. Составьте запрос, который выбирает из таблицы «Туры» значения всех полей для туров с ценой меньше 10000 руб:

Ответ:

12. Для того, чтобы отсортировать данные по некоторому полю, в запросе после ключевых слов **ORDER BY** (англ. «упорядочить по») указывают название этого поля:

**SELECT \* FROM Туры ORDER BY Цена**

Проверьте работу этого запроса.

Если в конце предыдущего запроса добавить слово **DESC** (англ. «descending» – нисходящий), сортировка выполняется в обратном порядке.

13. Составьте запрос, который выбирает из таблицы «Туры» значения всех полей для туров с ценой больше 10000 руб. и сортирует результаты по убыванию цены:

Ответ:

14. В запросах можно использовать стандартные функции. Например, функция **MIN** определяет минимальное значение заданного поля среди всех записей:

**SELECT MIN(Цена) FROM Туры**

Результат этого запроса – одно число.

15. Составьте запрос, который находит минимальную цену для туров в Норвегию:

Ответ:

16. Результаты запросов можно использовать в других запросах – получается вложенный запрос. Например, запрос

```
SELECT * FROM Туры WHERE Цена =  
(SELECT MIN(Цена) FROM Туры WHERE Страна = 'Норвегия')
```

вернет данные о самом дешевом туре.

17. Составьте запрос, который находит тур минимальной цены на самолёте:

*Ответ:*

#### Изменение и удаление данных

18. Для изменения записей используется оператор **UPDATE**. Запрос, приведенный ниже, увеличивает цены всех туров на 10%:

```
UPDATE Туры SET Цена = Цена*1.1
```

Проверьте, что данные в таблице «Туры» действительно изменились.

19. Запишите запрос, который вернёт цены в исходное состояние, и выполните его.

*Ответ:*

20. Авиакомпании в данный момент представляют скидку на билеты, так что цены всех туров на самолётах составляют 80% от исходных. Составьте и выполните соответствующий запрос.

*Ответ:*

Какая стоимость получилась у тура в Грецию?

*Ответ:*

21. Скопируйте таблицу «Туры», назвав копию «Туры2». Удалите все туры в Германию с помощью запроса

```
DELETE FROM Туры2 WHERE Страна = 'Германия'
```

Проверьте, что данные в таблице «Туры2» действительно изменились.

22. Удалите таблицу «Туры2», которая больше не нужна, с помощью запроса

```
DROP TABLE Туры2
```

Проверьте, что таблица «Туры2» удалена из списка таблиц.

## Практическая работа № 16-SQLite.

### Язык SQL (SQLite)

Язык SQL (*Structured Query Language*) – это специальный язык для управления данными. С его помощью можно полностью контролировать базу данных и выполнять все операции, посылая запросы (команды) в текстовом виде. Таким образом, например, можно управлять базой данных с удаленного компьютера.

В этой работе вы познакомитесь с основными командами языка SQL:

<b>CREATE TABLE</b>	создать таблицу
<b>INSERT</b>	добавить данные
<b>SELECT</b>	выбрать данные
<b>UPDATE</b>	изменить данные
<b>DELETE</b>	удалить данные
<b>DROP</b>	удалить таблицу

1. Введите имя новой базы данных *SQLbase.db*.
2. Просмотрите структуру базы данных (она должна быть пустая), щёлкнув по кнопке



или выполнив SQL-запрос

**SELECT \* FROM SQLITE\_MASTER;**

Здесь **SQLITE\_MASTER** – это название специальной таблицы, в которой хранится информация обо всех объектах базы данных SQLite (таблицах, индексах и т.п.).

Если базы нет, движок SQLite создаёт ее в момент первого обращения.

### Создание и заполнение таблиц

3. Введите следующую команду для создания таблицы:

```
CREATE TABLE [Туры] (  
[Код] INTEGER NOT NULL PRIMARY KEY,  
[Страна] VARCHAR(50) NOT NULL,  
[Транспорт] VARCHAR(20) NOT NULL,  
[Цена] MONEY NOT NULL )
```

В этой команде требуется создать таблицу (**CREATE TABLE**) с именем «Туры». В таблице должно быть 4 поля:

*Код* – целое число (**INTEGER**), непустое (**NOT NULL**), первичный ключ таблицы (**PRIMARY KEY**)

*Страна* – строка длиной до 50 символов, непустое

*Транспорт* – строка длиной до 20 символов, непустое

*Цена* – поле для хранения денежной суммы (**MONEY**)

**Названия таблиц и полей заключаются в квадратные скобки! Если эти названия состоят из одного слова, скобки можно не ставить:**

```
CREATE TABLE Туры (  
Код INTEGER NOT NULL PRIMARY KEY,  
Страна VARCHAR(50) NOT NULL,  
Транспорт VARCHAR(20) NOT NULL,  
Цена MONEY NOT NULL)
```

4. Выполните эту команду (клавиша F9). Проверьте, что таблица действительно создана.
5. Выполните команду выборки данных из этой таблицы.

**SELECT \* FROM Туры**

Этот оператор выберет все поля (\*) всех записей из таблицы «Туры». Таблица должна быть пустой.

6. Выполните команду для добавления в базу новой записи:

**INSERT INTO Туры**

## VALUES (1, 'Финляндия', 'автобус', 1200)

Эта команда вставляет (**INSERT**) в таблицу «*Туры*» одну запись. После ключевого слова **VALUES** в скобках перечислены через запятую значения полей в том порядке, в котором они задавались при создании таблицы.

**Символьные строки в значениях полей заключаются в апострофы или двойные кавычки!**

7. Выполните ещё одно добавление записи:

```
INSERT INTO Туры  
VALUES (1, 'Норвегия', 'самолёт', 15000)
```

Какая ошибка произошла? В чём её причина?

Ответ:

Исправьте ошибку и добавьте новую запись правильно. В ответе запишите SQL-запрос.

Ответ:

8. Аналогично добавьте в таблицу ещё несколько записей:

Страна	Транспорт	Цена
Швеция	паром	9000 р.
Германия	автобус	15700 р.
Греция	самолёт	23000 р.
Норвегия	автобус	8000 р.
Германия	самолёт	19000 р.

### Выбор и сортировка данных

9. Выполните запрос на выборку всех данных из таблицы *Туры*:

```
SELECT * FROM Туры
```

Проверьте, чтобы в таблицу были добавлены все нужные записи.

10. Вместо \* можно указать через запятую список нужных полей:

```
SELECT Страна, Цена FROM Туры
```

Проверьте результат выполнения этого запроса.

11. Чаще всего нужно выбрать только записи, удовлетворяющие некоторому условию отбора. Для этого используется ключевое слово **WHERE**, после которого стоит условие:

```
SELECT * FROM Туры WHERE Страна = 'Норвегия'
```

Проверьте работу этого оператора.

12. Составьте запрос, который выбирает из таблицы «*Туры*» значения полей «*Страна*», «*Транспорт*» и «*Цена*» для всех автобусных туров:

Ответ:

13. Составьте запрос, который выбирает из таблицы «*Туры*» значения всех полей для туров с ценой меньше 10000 руб:

Ответ:

14. Для того, чтобы отсортировать данные по некоторому полю, в запросе после ключевых слов **ORDER BY** (англ. «упорядочить по») указывают название этого поля:

```
SELECT * FROM Туры ORDER BY Цена
```

Проверьте работу этого запроса.

Если в конце предыдущего запроса добавить слово **DESC** (англ. «descending» – нисходящий), сортировка выполняется в обратном порядке.

15. Составьте запрос, который выбирает из таблицы «Туры» значения всех полей для туров с ценой больше 10000 руб. и сортирует результаты по убыванию цены:

Ответ:

16. В запросах можно использовать стандартные функции. Например, функция MIN определяет минимальное значение заданного поля среди всех записей:

**SELECT MIN(Цена) FROM Туры**

Результат этого запроса – одно число.

17. Составьте запрос, который находит минимальную цену для туров в Норвегию:

Ответ:

18. Результаты запросов можно использовать в других запросах – получается вложенный запрос. Например, запрос

**SELECT \* FROM Туры WHERE Цена =  
(SELECT MIN(Цена) FROM Туры WHERE Страна = 'Норвегия')**

вернет данные о самом дешевом туре.

19. Составьте запрос, который находит тур минимальной цены на самолёте:

Ответ:

#### Изменение и удаление данных

20. Для изменения записей используется оператор UPDATE. Запрос, приведенный ниже, увеличивает цены всех туров на 10%:

**UPDATE Туры SET Цена = Цена\*1.1**

Проверьте, что данные в таблице «Туры» действительно изменились.

21. Запишите запрос, который вернёт цены в исходное состояние, и выполните его.

Ответ:

22. Авиакомпании в данный момент представляют скидку на билеты, так что цены всех туров на самолётах составляют 80% от исходных. Составьте и выполните соответствующий запрос.

Ответ:

Какая стоимость получилась у тура в Грецию?

Ответ:

23. Создайте новую таблицу «Туры 2», имеющую точно такую же структуру, как и таблица «Туры». Запишите соответствующий запрос:

Ответ:

24. Просмотрите содержание таблицы «Туры 2». Какой запрос нужно ввести для этого?

Ответ:

Сколько записей вы обнаружили?

Ответ:

25. Скопируйте все данные из таблицы «Туры» в новую таблицу «Туры 2»

**INSERT INTO Туры2 SELECT \* FROM Туры;**

Проверьте, что данные в таблице «Туры2» действительно появились.

26. Удалите из таблицы «Туры2» все туры в Германию с помощью запроса

**DELETE FROM Туры2 WHERE Страна = 'Германия'**

Проверьте, что данные в таблице «Туры2» действительно изменились.

27. Удалите таблицу «Туры2», которая больше не нужна, с помощью запроса

**DROP TABLE Туры2**

Проверьте, что таблица «Туры2» удалена из списка таблиц.



## Практическая работа № 17.

### Создание базы данных

#### Уровень А.

1. Создайте новую базу данных *Футбол*.
2. Создайте таблицу *Футбол*. В конструкторе добавьте следующие поля:
  - *Команда* (ключевое поле, размер поля – 20 символов)
  - *Выигрыши*
  - *Ничьи*
  - *Проигрыши*
  - *Зарплата*

Определите правильный тип данных для каждого поля.

Ответ:

3. Введите следующие данные:

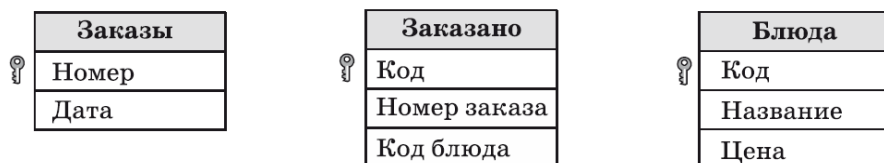
Команда	Выигрыши	Ничьи	Проигрыши	Зарплата
Аметист	10	7	3	13 290 р.
Бирюза	5	8	7	12 500 р.
Восход	13	5	2	22 000 р.
Закат	7	8	5	18 780 р.
Коллектор	11	6	3	20 200 р.
Кубань	6	12	2	14 000 р.
Малахит	12	3	5	17 340 р.
Ротор	8	12	0	15 820 р.
Статор	9	10	1	19 300 р.
Финиш	12	0	8	12 950 р.

4. Отсортируйте записи по убыванию количества выигрышей.
5. Примените фильтр, который отбирает только команды, имеющие более 10 побед и меньше 5 проигрышей. Сколько команд отобрано с помощью этого фильтра?

Ответ:

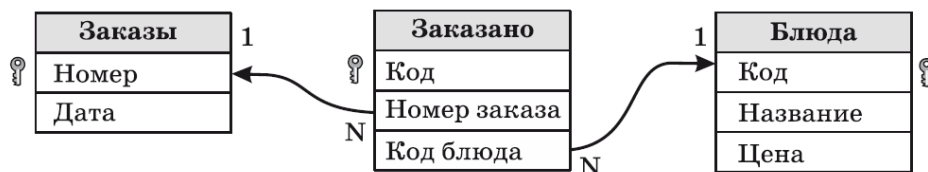
#### Уровень В.

1. Создайте новую базу данных *Кафе*.



Все таблицы постройте с помощью конструктора.

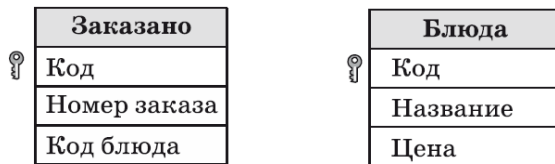
1. Установите связи между таблицами:



2. Заполните таблицы данными из §14 (рис. 3.16). Добавьте свои данные ещё о трёх заказах, сделанных в другие дни.

#### Уровень С.

1. Создайте новую базу данных *Кафе*. Таблицы *Блюда* и *Заказано* постройте с помощью конструктора.



2. Запишите *SQL*-запрос для создания таблицы *Заказы*:

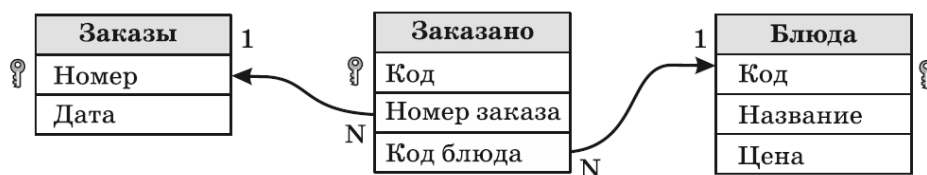


Ответ:

*Примечание.* Тип «дата» на языке *SQL* записывается как **DATE**.

Выполните этот запрос. В списке таблиц должна появиться новая таблица *Заказы*.

3. Установите связи между таблицами:



4. Заполните таблицу *Блюда* данными из §14 (рис. 3.16).  
 5. Таблицу *Заказы* заполните с помощью *SQL*-запросов. Запишите эти запросы в следующем поле:

Ответ:

*Примечание.* Дата 03.11.2014 вводится на языке *SQL* следующим образом:

– в *OpenOffice Base* как '2014-11-03';

– в русской версии *Microsoft Access* как '03.11.2014'.


6. Заполните таблицу *Заказано* вручную в режиме таблицы.  
 7. Добавьте в базу (любым способом) новые данные ещё о трёх заказах, сделанных в другие дни.

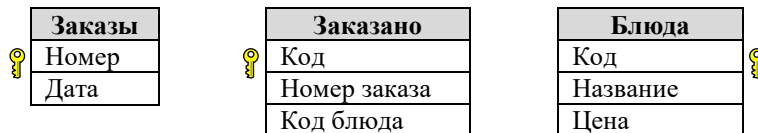
## Практическая работа (дополнительная).

### Язык SQL (многотабличная база данных, OpenOffice Base)

В этой работе вы познакомитесь с новой командой **ALTER TABLE** (изменить таблицу) языка *SQL* и научитесь составлять запросы к многотабличной реляционной базе данных.

#### Создание и заполнение таблиц

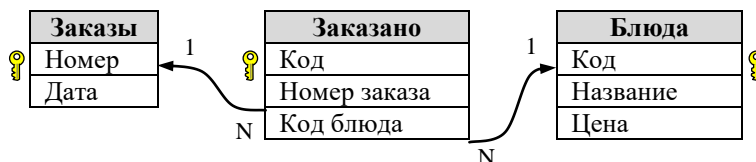
1. Используя только SQL-запросы, постройте три таблицы для реляционной базы данных кафе (значок  обозначает первичный ключ таблицы):



Запишите соответствующие SQL-запросы:

Ответ:

2. Теперь нужно построить связи между таблицами:



Для этой цели используется команда **ALTER TABLE** (англ. изменить таблицу). Построить связь между таблицами – это значит задать ограничение (**CONSTRAINT**), которое связывает первичный ключ одной таблицы с полем соответствующего типа другой. Если связываемое поле второй таблицы – неключевое, то оно называется *внешним ключом* (**FOREIGN KEY**). Например, команда для создания связи 1:N между ключевым полем *Номер* таблицы *Заказы* и неключевым полем *Номер заказа* таблицы *Заказано* выглядит так:

```
ALTER TABLE "Заказано"  
ADD CONSTRAINT ORDER_NO  
FOREIGN KEY("Номер заказа")  
REFERENCES "Заказы"("Номер")
```

Такая запись дословно означает:

**Изменить таблицу "Заказано"**  
**добавить связь ORDER\_NO**  
**внешний ключ "Номер заказа"**  
**ссылается на поле "Номер" таблицы "Заказы"**

Здесь **ORDER\_NO** – это просто имя, которое мы выбрали для этой связи (можно было выбрать и другое).

3. Введите и выполните показанный выше SQL-запрос на добавление связи. Зайдите в меню *Сервис – Связи* и убедитесь, что связь действительно создана<sup>7</sup>.
4. Составьте и выполните SQL-запрос на добавление второй связи. Запишите этот запрос в поле для ответа.

Ответ:

5. С помощью SQL-запросов заполните базу следующими данными

<sup>7</sup> В *OpenOffice Base* иногда связи появляются в окне не сразу, а только после перезагрузки базы.

Заказы		Заказано			Блюда		
Номер	Дата	Код	Номер заказа	Код блюда	Код	Название	Цена
1	11.04.13	1	1	1	1	борщ	80 р.
2	12.04.13	2	1	3	2	бифштекс	110 р.
		3	1	4	3	гуляш	70 р.
		4	2	1	4	чай	10 р.
		5	2	2	5	кофе	50 р.
		6	2	2			
		7	2	5			

### Выбор данных и сортировка

6. Построим в режиме SQL запрос *СоставЗаказа*, который выводит номер заказа и название заказанных блюд. Эти данные находятся в двух таблицах – *Заказано* и *Блюда*, поэтому их нужно как-то объединить. Для этого используется связь 1:N между таблицами, которую мы недавно установили. Действительно, для каждой записи в таблице *Заказано* нужно выбрать название блюда из таблицы *Блюда*, код которого совпадает с полем *Заказано.Код блюда*.

Это запрос на выборку данных, поэтому используем оператор **SELECT**:

```
SELECT "Заказано"."Номер заказа", "Блюда"."Название"
FROM "Заказано", "Блюда"
WHERE "Заказано"."Код блюда" = "Блюда"."Код"
```

Здесь из таблиц *Заказано* и *Блюда* выбираются поля *Номер заказа* и *Название*; условие в последней строке связывает таблицы.

Поскольку названия полей в таблицах, из которых идет выбор, не совпадают, можно было записать запрос в сокращенной форме, указав после оператора **SELECT** только названия нужных полей:

```
SELECT "Номер заказа", "Название"
FROM "Заказано", "Блюда"
WHERE "Заказано"."Код блюда" = "Блюда"."Код"
```

7. Теперь добавим в запрос дату заказа. Она находится в таблице *Заказы*, которая пока в запросе не участвует. Таким образом, нам нужно объединить три таблицы. Условие отбора получается сложным, два условия (связи по коду блюда между таблицами *Заказано* и *Блюда* и по номеру заказа между таблицами *Заказы* и *Заказано*) объединяются с помощью логической операции AND (И):

```
SELECT "Номер заказа", "Дата", "Название"
FROM "Заказано", "Блюда", "Заказы"
WHERE "Заказано"."Код блюда" = "Блюда"."Код"
AND "Заказано"."Номер заказа" = "Заказы"."Номер"
```

Проверьте результат выполнения этого запроса.

8. Измените запрос так, чтобы он выбирал только блюда из состава заказа № 1. Запишите этот запрос:

Ответ:

9. Построим еще один запрос *Итоги*, в котором для каждого заказа выводится его номер, дата и общая сумма (с помощью функции **SUM**).

```
SELECT "Номер заказа", "Дата", SUM("Цена")
FROM "Заказано", "Блюда", "Заказы"
WHERE "Заказано"."Код блюда" = "Блюда"."Код"
AND "Заказано"."Номер заказа" = "Заказы"."Номер"
GROUP BY "Номер заказа", "Дата"
```

В последней строке указано, что по полям *Номер заказа* и *Дата* выполняется группировка, то есть сумма цен считается для каждой уникальной пары «*Номер заказа – Дата*».

В таблице с результатами запроса заголовок столбца с суммой выглядит не совсем понятно для пользователя:

```
«SUM("Блюда"."Цена")»
```

Для того, чтобы сделать у этого столбца заголовок «*Сумма*», нужно добавить в первую строку запроса после SUM("Цена") так называемый псевдоним (подпись) с ключевым словом **AS**:

```
SELECT "Номер заказа", "Дата", SUM("Цена") AS "Сумма"
```

...

Проверьте результат выполнения этого запроса. Псевдонимы можно задавать для всех значений, которые выводятся в запросе.

10. Измените запрос так, чтобы заказы были отсортированы в порядке убывания суммы (используйте ключевые слова **ORDER BY**).

Ответ:

### Вложенные запросы

11. Построим запрос *МинСумма*, который выводит минимальную сумму заказа. Для этого будем использовать уже готовый запрос *Итоги*. Таким образом, источником данных для запроса *МинСумма* будет не таблица, а другой запрос. Отметим, что предварительно в запросе *Итоги* нужно отменить сортировку. Запрос получается очень простым

```
SELECT MIN("Сумма") AS "Сумма" FROM "Итоги"
```

12. Наконец, можно вывести информацию о заказе с минимальной суммой:

```
SELECT "Номер заказа", "Дата", "Сумма"  
FROM "Итоги", "МинСумма"  
WHERE "Итоги"."Сумма" = "МинСумма"."Сумма"
```

Обратите внимание, что этот запрос использует результаты выполнения двух ранее построенных запросов – *Итоги* и *МинСумма*. Запрос *Итоги* можно было и не составлять, а вместо этого использовать вложенный запрос (запрос в запросе):

```
SELECT "Номер заказа", "Дата", "Сумма"  
FROM "Итоги"  
WHERE "Сумма" =  
(SELECT MIN("Сумма") FROM "Итоги")
```

Заметим, что если в базе данных есть информация о нескольких заказах с такой же (минимальной) суммой, будет показана информация обо всех этих заказах.

13. Измените запрос так, чтобы получить список всех заказов, сумма которых больше средней.


Ответ:

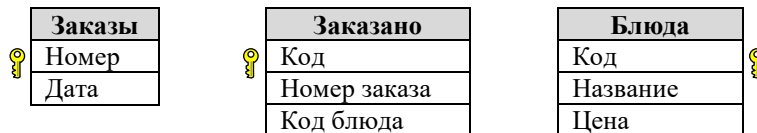
## Практическая работа (дополнительная).

### Язык SQL (многотабличная база данных, MS Access)

В этой работе вы познакомитесь с новой командой **ALTER TABLE** (изменить таблицу) языка *SQL* и научитесь составлять запросы к многотабличной реляционной базе данных.

#### Создание и заполнение таблиц

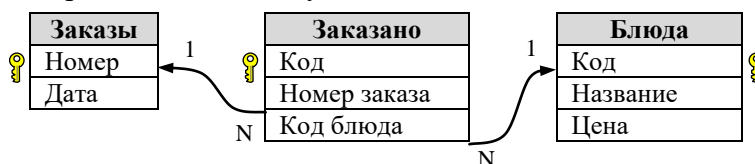
1. Используя только SQL-запросы, постройте три таблицы для реляционной базы данных кафе (значок  обозначает первичный ключ таблицы):



Запишите соответствующие SQL-запросы:

Ответ:

2. Теперь нужно построить связи между таблицами:



Для этой цели используется команда **ALTER TABLE** (англ. изменить таблицу). Построить связь между таблицами – это значит задать ограничение (**CONSTRAINT**), которое связывает первичный ключ одной таблицы с полем соответствующего типа другой. Если связываемое поле второй таблицы – неключевое, то оно называется *внешним ключом* (**FOREIGN KEY**). Например, команда для создания связи 1:N между ключевым полем *Номер* таблицы *Заказы* и неключевым полем *Номер заказа* таблицы *Заказано* выглядит так:

```
ALTER TABLE Заказано
ADD CONSTRAINT ORDER_NO
FOREIGN KEY([Номер заказа])
REFERENCES Заказы(Номер)
```

Такая запись дословно означает:

**Изменить таблицу Заказано**  
добавить связь **ORDER\_NO**  
внешний ключ **[Номер заказа]**  
ссылается на поле **Номер** таблицы **Заказы**

Здесь **ORDER\_NO** – это просто имя, которое мы выбрали для этой связи (можно было выбрать и другое).

3. Введите и выполните показанный выше SQL-запрос на добавление связи. Зайдите в меню *Сервис – Связи* и убедитесь, что связь действительно создана<sup>8</sup>.
4. Составьте и выполните SQL-запрос на добавление второй связи. Запишите этот запрос в поле для ответа.

Ответ:

5. С помощью SQL-запросов заполните базу следующими данными

<sup>8</sup> В *OpenOffice Base* иногда связи появляются в окне не сразу, а только после перезагрузки базы.

Заказы		Заказано			Блюда		
Номер	Дата	Код	Номер заказа	Код блюда	Код	Название	Цена
1	11.04.13	1	1	1	1	борщ	80 р.
2	12.04.13	2	1	3	2	бифштекс	110 р.
		3	1	4	3	гуляш	70 р.
		4	2	1	4	чай	10 р.
		5	2	2	5	кофе	50 р.
		6	2	2			
		7	2	5			

### Выбор данных и сортировка

6. Построим в режиме SQL запрос *СоставЗаказа*, который выводит номер заказа и название заказанных блюд. Эти данные находятся в двух таблицах – *Заказано* и *Блюда*, поэтому их нужно как-то объединить. Для этого используется связь 1:N между таблицами, которую мы недавно установили. Действительно, для каждой записи в таблице *Заказано* нужно выбрать название блюда из таблицы *Блюда*, код которого совпадает с полем *Заказано.Код блюда*.

Это запрос на выборку данных, поэтому используем оператор **SELECT**:

```
SELECT Заказано.[Номер заказа], Блюда.Название
```

```
FROM Заказано, Блюда
```

```
WHERE Заказано.[Код блюда] = Блюда.Код
```

Здесь из таблиц *Заказано* и *Блюда* выбираются поля *Номер заказа* и *Название*; условие в последней строке связывает таблицы.

Поскольку названия полей в таблицах, из которых идет выбор, не совпадают, можно было записать запрос в сокращенной форме, указав после оператора **SELECT** только названия нужных полей:

```
SELECT [Номер заказа], Название
```

```
FROM Заказано, Блюда
```

```
WHERE Заказано.[Код блюда] = Блюда.Код
```

7. Теперь добавим в запрос дату заказа. Она находится в таблице *Заказы*, которая пока в запросе не участвует. Таким образом, нам нужно объединить три таблицы. Условие отбора получается сложным, два условия (связи по коду блюда между таблицами *Заказано* и *Блюда* и по номеру заказа между таблицами *Заказы* и *Заказано*) объединяются с помощью логической операции AND (И):

```
SELECT [Номер заказа], Дата, Название
```

```
FROM Заказано, Блюда, Заказы
```

```
WHERE Заказано.[Код блюда] = Блюда.Код
```

```
AND Заказано.[Номер заказа] = Заказы.Номер
```

Проверьте результат выполнения этого запроса.

8. Измените запрос так, чтобы он выбирал только блюда из состава заказа № 1. Запишите этот запрос:

Ответ:

9. Построим еще один запрос *Итоги*, в котором для каждого заказа выводится его номер, дата и общая сумма (с помощью функции **SUM**).

```
SELECT [Номер заказа], Дата, SUM(Цена)
```

```
FROM Заказано, Блюда, Заказы
```

```
WHERE Заказано.[Код блюда] = Блюда.Код
```

```
AND Заказано.[Номер заказа] = Заказы.Номер
```

```
GROUP BY [Номер заказа], Дата
```

В последней строке указано, что по полям *Номер заказа* и *Дата* выполняется группировка, то есть сумма цен считается для каждой уникальной пары «*Номер заказа – Дата*».

В таблице с результатами запроса заголовок столбца с суммой выглядит не совсем понятно для пользователя, например:

«**Expr1002**»

Для того, чтобы сделать у этого столбца заголовок «*Сумма*», нужно добавить в первую строку запроса после SUM("Цена") так называемый псевдоним (подпись) с ключевым словом **AS**:

```
SELECT [Номер заказа], Дата, SUM(Цена) AS Сумма
```

...

Проверьте результат выполнения этого запроса. Псевдонимы можно задавать для всех значений, которые выводятся в запросе.

10. Измените запрос так, чтобы заказы были отсортированы в порядке убывания суммы (используйте ключевые слова **ORDER BY**).

Ответ:

### Вложенные запросы

11. Построим запрос *МинСумма*, который выводит минимальную сумму заказа. Для этого будем использовать уже готовый запрос *Итоги*. Таким образом, источником данных для запроса *МинСумма* будет не таблица, а другой запрос. Отметим, что предварительно в запросе *Итоги* нужно отменить сортировку. Запрос получается очень простым

```
SELECT MIN(Сумма) AS Сумма FROM Итоги
```

12. Наконец, можно вывести информацию о заказе с минимальной суммой:

```
SELECT [Номер заказа], Дата, Итоги.Сумма  
FROM Итоги, МинСумма  
WHERE Итоги.Сумма = МинСумма.Сумма
```

Обратите внимание, что этот запрос использует результаты выполнения двух ранее построенных запросов – *Итоги* и *МинСумма*. Запрос *МинСумма* можно было и не составлять, а вместо этого использовать вложенный запрос (запрос в запросе):

```
SELECT [Номер заказа], Дата, Сумма  
FROM Итоги  
WHERE Сумма =  
(SELECT MIN(Сумма) AS Сумма FROM Итоги)
```

Заметим, что если в базе данных есть информация о нескольких заказах с такой же (минимальной) суммой, будет показана информация обо всех этих заказах.

13. Измените запрос так, чтобы получить список всех заказов, сумма которых больше средней.

Ответ:



## Практическая работа № 18.

### Создание запросов

#### Уровень А.

При выполнении этой работы используется база данных *Футбол*, с которой вы работали на предыдущем уроке.

1. Постройте запрос с именем *ЗапросЛучшие*, который выводит все поля таблицы *Футбол* для всех команд, имеющих более 8 побед и меньше 5 проигрышей. Команды должны быть расставлены по убыванию числа побед. Сколько команд осталось в результатах запроса?

Ответ:

Как выглядит этот запрос на языке SQL?

Ответ:

2. Постройте запрос с именем *ЗапросОчки*, который отбирает все поля из таблицы, кроме поля Зарплата. Как выглядит этот запрос на языке SQL?

Ответ:

3. Добавьте в этот запрос вычисляемое поле *Очки* (за каждую победу команда получает 3 очка, за ничью – 1 очко).

Как выглядит этот запрос на языке SQL?

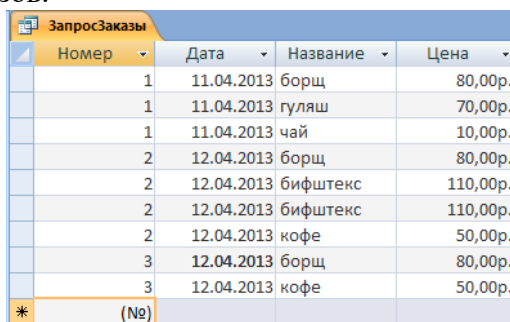
Ответ:

4. Отсортируйте записи (с помощью запроса) так, чтобы на 1-ом месте стоял победитель (команда, набравшая наибольшее количество очков). Какая команда оказалась на первом месте и сколько очков она набрала?

Ответ:

#### Уровень В.

1. При выполнении этой работы используется база данных *Кафе*, с которой вы работали на предыдущем уроке.
2. Постройте простой запрос *ЗапросЗаказы*, который собирает всю информацию о составе сделанных заказов.



Номер	Дата	Название	Цена
1	11.04.2013	борщ	80,00р.
1	11.04.2013	гуляш	70,00р.
1	11.04.2013	чай	10,00р.
2	12.04.2013	борщ	80,00р.
2	12.04.2013	биштекс	110,00р.
2	12.04.2013	биштекс	110,00р.
2	12.04.2013	кофе	50,00р.
3	12.04.2013	борщ	80,00р.
3	12.04.2013	кофе	50,00р.

Как выглядит этот запрос на языке SQL?

Ответ:

3. Объясните, как на языке SQL задаются связи между таблицами?

Ответ:

4. Постройте итоговый запрос *ЗапросКоплате*, который подсчитывает общую сумму оплаты по каждому из сделанных заказов.

ЗапросОплате		
Номер заказа	Дата	Сумма
1	11.04.2013	160,00р.
2	12.04.2013	350,00р.
3	12.04.2013	130,00р.

Как выглядит этот запрос на языке SQL?

Ответ:

5. Объясните, как выполняется на языке SQL суммирование цен отдельных блюд?

Ответ:

6. Объясните, что означают ключевые слова **GROUP BY** на языке SQL?

Ответ:

### Уровень С.

1. Постройте заказ, который вычисляет минимальную сумму заказа.

МинСумма	
Сумма	
130,00р.	

Как выглядит этот запрос на языке SQL?

Ответ:

2. С помощью языка SQL постройте и выполните запрос, который возвращает сведения о заказе с минимальной суммой.

ЗаказМинСумма		
Номер заказа	Дата	Сумма
3	12.04.2013	130,00р.

Как выглядит этот запрос на языке SQL?

Ответ:

## Практическая работа № 19.

### Формы для ввода данных


#### Уровень А.

При выполнении этой работы используется база данных *Футбол*, с которой вы работали на предыдущих уроках.

14. Постройте форму на основе таблицы *Футбол* с помощью мастера.

15. Добавьте на форму изображение футбольного мяча (файл **ball.jpg**).

16. Удалите надпись «Команда» и выделите название команды жирным шрифтом увеличенного размера. Внешний вид формы должен получиться примерно такой, как на рисунке справа.

	<b>Аметист</b>
Победы	<input type="text" value="10"/>
Ничьи	<input type="text" value="7"/>
Поражения	<input type="text" value="3"/>
Зарплата	<input type="text" value="13 200 руб."/>

17. С помощью этой формы добавьте в таблицу еще три записи:

Команда	Выигрыши	Ничьи	Проигрыши	Зарплата
Зубр	8	5	7	15 900 р.
Тур	10	5	6	12 300 р.
Бизон	15	3	3	18 700 р.

18. Проверьте, появились ли эти записи в таблице *Футбол*.

19. Выполните запрос *ЗапросОчки* и посмотрите, как изменилась ситуация в турнирной таблице. Какая команда теперь на первом месте и сколько у неё очков?

Ответ:

#### Уровень В-С.

1. Используя материал учебника, постройте форму, показанную на рисунке.

Заказ	<input type="text" value="1"/>								
Дата	<input type="text" value="11.12.12"/>								
	<table border="1"><thead><tr><th>Блюдо</th><th>Цена</th></tr></thead><tbody><tr><td>▶ борщ</td><td>80 руб.</td></tr><tr><td>гуляш</td><td>70 руб.</td></tr><tr><td>чай</td><td>10 руб.</td></tr></tbody></table>	Блюдо	Цена	▶ борщ	80 руб.	гуляш	70 руб.	чай	10 руб.
Блюдо	Цена								
▶ борщ	80 руб.								
гуляш	70 руб.								
чай	10 руб.								
Запись	<input type="text" value="1"/> из <input type="text" value="3"/>								
К оплате	<input type="text" value="160 руб."/>								

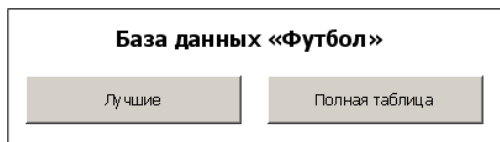
## Практическая работа № 20.

### Кнопочные формы

#### Уровень А.

При выполнении этой работы используется база данных *Футбол*, с которой вы работали на предыдущих уроках.

1. Постройте формы для ввода данных на основе запросов *ЗапросЛучшие* и *ЗапросОчки*.
2. Постройте кнопочную форму, в которой каждая кнопка служит для открытия соответствующей формы с данными:

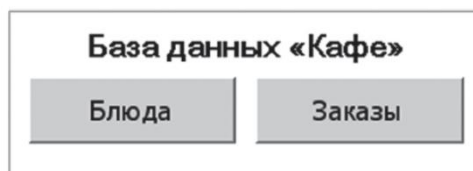


3. Создайте макрос, который автоматически открывает кнопочную форму при открытии базы.

#### Уровень В-С.

При выполнении этой работы используется база данных *Кафе*, с которой вы работали на предыдущих уроках.

1. Создайте табличные формы *Блюда* и *Заказы* для редактирования данных.
2. Используя материал учебника, построьте кнопочную форму, показанную на рисунке:



При щелчке по одной из кнопок должна появляться соответствующая форма.

3. Создайте макрос, который автоматически открывает кнопочную форму при открытии базы (он должен обязательно иметь имя **AutoExec**).

## Практическая работа № 21.

### Отчёты

#### Уровень А.

При выполнении этой работы используется база данных *Футбол*, с которой вы работали на предыдущих уроках.

4. Постройте отчет *Результаты* на основе запроса *ЗапросОчки* с помощью мастера. Расположите команды по убыванию количества набранных очков.
5. Измените внешний вид отчета так, как показано на рисунке. При редактировании отчёта в *OpenOffice Writer* нужно добавить новый столбец *Место* и установить для ячейки стиль нумерованного списка.

#### Турнирная таблица

Автор: *Никодим Озерный*  
Дата: 20.04.13

<i>Команда</i>	<i>Победы</i>	<i>Ничьи</i>	<i>Поражения</i>	<i>Очки</i>	<i>Место</i>
Коллектор	11	6	3	39	1
Статор	9	10	1	37	2
Аметист	10	7	3	37	3
Финиш	12	0	8	36	4
Ротор	8	12	0	36	5
Тур	10	5	6	35	6
Кубань	6	12	2	30	7

Страница 1/1

#### Уровень В-С.

При выполнении этой работы используется база данных *Кафе*, с которой вы работали на предыдущих уроках.

1. Используя материал учебника, постройте отчёт с группировкой, показанный на рисунке.

<b>Заказы</b>	
Исполнитель: Василий Петров	
Дата: 15.12.12	
<b>Дата 11.12.12</b>	
<b>Заказ 1</b>	
<i>Блюдо</i>	<i>Цена</i>
гуляш	40 руб.
борщ	30 руб.
рассольник	20 руб.
<b>Дата 12.12.12</b>	
<b>Заказ 2</b>	
<i>Блюдо</i>	<i>Цена</i>
компот	10 руб.

## Практическая работа № 22.

### Нереляционные базы данных

В этой работе мы познакомимся с документо-ориентированной СУБД *MongoDB* (<http://www.mongodb.org>). Эта СУБД кроссплатформенная и относится к категории свободного программного обеспечения (*Open Source*). Для управления данными используется язык *JavaScript*.

Данные хранятся не в виде таблиц, а в виде *коллекций документов*. *Документ* – это объект, имеющий свойства, то есть пары «имя-значение». Главное свойство документа – это его *идентификатор* (код), который всегда играет роль первичного ключа.

В одной коллекции могут быть совершенно разные документы с разным набором свойств, это и отличает документо-ориентированную БД от реляционной.

Информация об объекте записывается в фигурных скобках, например:

```
{ name: "Вася", age: 16 }
```

Этот объект имеет два свойства (поля) – свойство **name** со значением «Вася» и свойство **age** со значением 16. Такой текстовый формат записи называется **JSON** (англ. *JavaScript Object Notation* = запись объектов с помощью *JavaScript*).

Свойства объекта могут быть списками значений (массивами), они перечисляются в квадратных скобках через запятую:

```
{ name: "Вася", age: 16, lang: ["C", "Pascal", "JavaScript"] }
```

Здесь свойство **lang** – массив, в котором записаны названия языков программирования.

Свойства могут сами быть объектами со своими свойствами, например:

```
{  
  name: "Вася",  
  family: { mother: "Вера", father: "Петя" }  
}
```

Свойство **family** содержит два внутренних свойства (поля): **mother** и **father**.

### Начало работы

1. Запустите программу **mongo** в консольном режиме. Определите рабочую базу данных, выполнив команду **db** (от англ. *database* = база данных)

**db**

Ответ:

2. Мы построим простую базу данных для блога в Интернете. Переключитесь на базу данных **blog** с помощью команды

**use blog**

Если этой базы данных раньше не было, она будет создана. Проверьте, что она действительно стала активной.

3. Для добавления объекта в коллекцию используется оператор **insert** (*вставить*):

**db.posts.insert ( документ )**

Первая часть записи, «**db**», означает обращение к рабочей базе данных; вторая, «**posts**» - название коллекции (если такой коллекции нет, она будет создана), в скобках записываются свойства объекта-документа. В нашем примере в базе **blog** будет одна коллекция **posts** (сообщения, записи, «посты»). Для каждого поста нужно задать дату и текст. Добавьте одну запись следующим образом<sup>9</sup>:

**db.posts.insert (**  
**{date: new Date("04/23/2020"), text: "Привет!"} )**

Дата (свойство **date**) строится с помощью функции **Date**; ей передается символьная запись даты в формате, принятом в США (месяц/день/год).

---

<sup>9</sup> Если вы работаете в операционной системе *Windows*, не используйте русские буквы в значениях свойств. Это связано с тем, что в *MongoDB* для хранения символьных строк используется кодировка UTF-8, которая не очень хорошо поддерживается в консоли *Windows*.

4. Проверьте, создана ли коллекция **posts** с помощью команды, которая показывает все коллекции текущей базы:

**show collections**

5. Проверьте, добавлена ли запись, с помощью команды

**db.posts.find()**

которая ищет и показывает все документы, входящие в коллекцию **posts**. Для вывода в красивом формате можно дополнительно вызвать функцию **pretty** (англ. приятный)

**db.posts.find().pretty()**

В этом случае, если список свойств документа не помещается в одну строку, он выводится в столбик:

```
{
  "_id": ObjectId("5176abbc06a6380da34966a2"),
  "date": ISODate("2020-04-22T20:00:00Z"),
  "text": "Привет!"
}
```

СУБД автоматически добавила поле «**\_id**» (идентификатор, код), которое представляет собой суррогатный первичный ключ и строится случайным образом с помощью функции **ObjectId**.

Дата преобразована в формат Международной организации по стандартизации (ISO) и, кроме собственно даты, включает еще и время. Мы задали дату 23.04.2020, время не задавали, поэтому принимается время 00:00. По Гринвичу в этот момент было 20:00, а дата – 22.04.2020.

6. Если нужно, значение ключа «**\_id**» можно задать самостоятельно, какое мы хотим. Удалим все документы из коллекции командой **remove**:

**db.posts.remove()**

и добавим четыре новых поста в базу, указав явно идентификаторы<sup>10</sup>:

```
db.posts.insert ( { _id: 1, date: new Date("04/23/2020"), text: "Привет!" } )
```

```
db.posts.insert ( { _id: 2, date: new Date("04/24/2020"), text: "Это второй пост." } )
```

```
db.posts.insert ( { _id: 3, date: new Date("04/25/2020"), text: "Это третий пост." } )
```

```
db.posts.insert ( { _id: 4, date: new Date("04/26/2020"), text: "Это четвёртый пост." } )
```

Убедитесь, что документы действительно добавлены в коллекцию.

## Поиск

7. Для поиска нужных документов используется уже знакомая нам функция **find**. В скобках можно задать критерий поиска – это объект (записанный в фигурных скобках), который содержит название поля и значение этого поля, которое мы ищем. Например, найдите документ с кодом (идентификатором, **\_id**), равным 2:

```
db.posts.find( { _id: 2 } )
```

8. В условиях можно использовать не только строгие равенства, как в предыдущее примере, но и неравенства. Неравенство – это тоже объект, у которого специальное название свойства, начинающееся знаком \$:

\$ne – не равно,

\$lt – меньше,     \$lte – меньше или равно,

\$gt – больше,     \$gte – больше или равно

Например, найдем записи с идентификатором больше 2:

```
db.posts.find( { _id: { $gt: 2 } } )
```

Сколько документов найдено?

*Ответ:*

---

<sup>10</sup> Можно создать в рабочем (текущем) каталоге текстовый файл с расширением **.js** (*JavaScript*), включающий эти команды, и загрузить его с помощью команды **load**, например:

```
load('data.js')
```

9. Найдем посты, написанные 24.04.2020 или позднее:

```
db.posts.find({date: {$gte: new Date("04/24/2020")}})
```

10. Для поиска можно использовать сразу несколько условий. Если все условия нужно выполнить одновременно, их записывают как один объект с несколькими свойствами. Например, следующий запрос находит все документы, у которых свойство «`_id`» больше 2, а дата создания – не раньше 26.04.2020:

```
db.posts.find( { _id: {$gt: 2},  
               date: {$gte: new Date("04/26/2020")} })
```

11. Самое мощное средство поиска в *MongoDB* – это оператор `$where`, которому можно передать строку в кавычках, задающую условие поиска на *JavaScript*, например,

```
db.posts.find( { $where: "this._id > 2" } )
```

Здесь `this` – это объект (документ), который требуется проверить; через точку записывается название нужного поля. Оператору `$where` можно передать любую функцию на *JavaScript*, которая возвращает логическое значение (истинное, если условие отбора выполняется). Например, запрос, показанный в п. 10, можно было записать так:

```
db.posts.find( { $where: function()  
               { return this._id > 2 &&  
                 this.date >= new Date("04/26/2020")  
               }  
             } )
```

Пара символов `&&` в языке *JavaScript* обозначает логическую операцию «И», а символы `||` – логическую операцию «ИЛИ». Проверка на равенство записывается как `==`, а условие «не равно» – как `!=`.

12. Составьте запрос для поиска всех документов, у которых свойство `_id` равно 1 или дата создания равна 26.04.2020.

Учтите, что в *MongoDB* дата хранится вместе со временем, то есть `new Date()` создает объект время+дата, соответствующий моменту 00:00 заданного дня (по местному времени). Поэтому отобрать документы с конкретной датой можно с помощью двойного неравенства (прямое сравнение не работает).

Ответ:

### Сортировка

13. Для сортировки используется функция `sort`. Она сортирует те документы, которые предварительно найдены с помощью `find`. При вызове функции `sort` в скобках указывается порядок сортировки – объект (в фигурных скобках), содержащий название поля для сортировки; значение этого поля может быть 1 (сортировка по возрастанию) или «-1» (по убыванию).

Отсортируйте документы по возрастанию даты

```
db.posts.find().sort( {date: 1} )
```

а теперь по убыванию:

```
db.posts.find().sort( {date: -1} )
```

### Изменение

14. Для изменения документов используют команду `update`. Допустим, мы хотим добавить к документу с идентификатором 1 новое логическое свойство `visible` (англ. *видимый*) и присвоить ему значение `false` (ложь), которое означает, что этот пост пока скрыт и выводить его на веб-страницу не нужно.

Функции `update` передаются два объекта в фигурных скобках: условие, позволяющее найти нужный документ, и свойства, которые у него нужно изменить. Попробуйте выполнить команду

```
db.posts.update( { _id: 1 }, { visible: false } )
```



и посмотрите, что получилось в результате.

Ответ:

15. Восстановите исходный пост с кодом 1. Какую команду нужно ввести?

Ответ:

16. Для того, чтобы не заменять документ полностью, а изменить (или добавить) значение какого-то поля, нужно использовать специальный объект со свойством **\$set** (установить):

```
db.posts.update( { _id: 1 }, { $set: { visible: false } } )
```

Примените эту команду и проверьте результат её выполнения.

Обратите внимание, что сейчас в коллекции **posts** находятся документы с разной структурой: один из них имеет свойство **visible**, а остальные – нет. При этом никаких изменений в структуру базы данных вносить не пришлось.

17. Найдите все документы, у которых свойство **visible** равно **false**. Какой запрос нужно для этого выполнить?

Ответ:

18. Теперь выполним множественное обновление: установим свойство **visible** равным **true** (истина) для всех документов, для которых это свойство не установлено (или, что то же самое, равно специальному нулевому значению **null**):

```
db.posts.update( { visible: null },  
  { $set: { visible: true } }, { multi: true } )
```

Третий параметр – объект со свойством **multi** (англ. *множественный*), равным **true** (истина) разрешает изменение нескольких документов сразу (если его не указать, будет изменён только один документ – тот, который найден первым).

19. Составьте запрос для поиска всех документов, у которых свойство **visible** равно **true**, а дата создания – не позднее 25.04.2020:

Ответ:

Проверьте результат его работы.

20. Теперь добавим к комментариям к одному из постов (с идентификатором 2). Заметьте, что благодаря документо-ориентированной СУБД заранее планировать наличие комментариев не нужно – мы можем добавлять новые свойства к любому документу «на ходу».

Комментариев может быть много, поэтому новое свойство **comments** будет массивом. Для добавления нового элемента в массив используется специальный объект со свойством **\$push** (англ. *втолкнуть*):

```
db.posts.update( { _id: 2 },  
  { $push: { comments: "Комментарий 1" } } )  
db.posts.update( { _id: 2 },  
  { $push: { comments: "Комментарий 2" } } )  
db.posts.update( { _id: 2 },  
  { $push: { comments: "Комментарий 3" } } )
```

Когда выполняется первая из этих команд, у документа с идентификатором 2 еще нет свойства **comments** – оно будет создано автоматически.

21. Все документы с комментариями можно найти с помощью объекта со свойством **\$exists** (существует):

```
var c = db.posts.find( { comments: { $exists: true } } )
```

Результат этого запроса не выводится на экран, а записывается в переменную **c**. Затем можно определить его длину (число найденных документов) с помощью

функции **length** (англ. *длина*) и вывести на экран в цикле в формате *JSON* с помощью функции **printjson**:

```
for(i = 0; i < c.length(); i++) printjson( c[i] )
```

Можно работать и с отдельными свойствами, например, вывести на экран даты всех найденных постов с помощью функции **print**:

```
for(i = 0; i < c.length(); i++) print( c[i].date )
```

### Удаление

22. Как вы уже знаете, для удаления всех документов из коллекции используется команда **remove**. Запишите команду, которая удаляет все документы из коллекции **posts**, но не выполняйте её.

Ответ:

23. С помощью команды **remove** можно удалять отдельные записи – условие для поиска нужных записей задается как параметр функции. Например:

```
db.posts.remove( { _id: 4 } )
```

Выполните эту команду и проверьте её выполнение.

24. Запишите команду для удаления всех документов, у которых свойство **visible** установлено в **false** выполните её.

Ответ:

25. С помощью команды **drop** вся коллекция удаляется из базы. Например,

```
db.posts.drop()
```

26. Объясните, в чём отличие между командами **remove()** и **drop()**.

Ответ:

27. Проверьте, что коллекция **posts** действительно была удалена. Какую команду нужно для этого использовать?

Ответ:

### Дополнительная информация:

1. <http://www.mongodb.org> – официальный сайт разработки *MongoDB*.
2. <http://docs.mongodb.org/manual/installation/> – инструкция по установке *MongoDB* для разных операционных систем.
3. <http://jsman.ru/mongo-book/> – русский перевод книги К. Seguin «*The Little MongoDB Book*».

## Установка MongoDB

### Windows

1. Скачайте архив с программой со страницы <http://www.mongodb.org/downloads>
2. Распакуйте архив в отдельный каталог в любом месте диска, например, в каталог **C:\MongoDB**. Внутри этого каталога должен появиться каталог **bin**, в котором находятся все файлы СУБД.
3. Создайте новый каталог, где будут храниться данные. Удобно создать это каталог прямо внутри каталога **C:\MongoDB**, например, **C:\MongoDB\data**.
4. Создайте в каталоге **C:\MongoDB\bin** командный файл **start.bat** с командой запуска серверной части, в параметре **dbpath** указывается путь к каталогу с базами данных:

```
C:\MongoDB\bin\mongod.exe --dbpath C:\MongoDB\data
```

5. Создайте (например, на рабочем столе) ярлык на файл **C:\MongoDB\bin\start.bat** для запуска серверной части.

6. Создайте (например, на рабочем столе) ярлык на файл **C:\MongoDB\bin\mongo.exe** для запуска консоли (оболочки).
7. Запустите серверную часть, затем запустите консоль.

### Ubuntu

#### Установка пакета

1. В Терминале введите следующую команду для импорта публичного GPG-ключа (<http://docs.mongodb.org/10gen-gpg-key.asc>):  
**sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10**
2. Создайте файл **/etc/apt/sources.list.d/10gen.list** и добавьте в него строку для обращения к репозиторию компании 10gen:  
**deb http://downloads-distrow.mongodb.org/repo/ubuntu-upstart dist 10gen**
3. Перегрузите список репозитория командой  
**sudo apt-get update**
4. Установите последнюю версию пакета командой  
**sudo apt-get install mongodb-10gen**

#### Запуск программы

5. Запустите серверную часть **mongod** как службу командой  
**sudo service mongodb start**
6. Запустите командную оболочку (консоль) командой  
**mongo**

Установка на других ОС: <http://docs.mongodb.org/manual/installation/>

## **Практическая работа № 23.**

### **Простая экспертная система**

1. Выберите одну из предложенных тем (или любую другую, которая вам близка) для своей экспертной системы:
  - «Анализ неисправностей компьютера (автомобиля, велосипеда и т.п.)»
  - «Подбор комплектующих для нового компьютера»
  - «Выбор места отдыха во время отпуска»
  - «Выбор места рыбалки в выходные»
  - «Выбор материалов для строительства»
  - «Определение типа корабля (самолёта, автомобиля)»
  - «Автомат, определяющий задуманного артиста (политика, героя мультфильма и т.п.)»
  - ...
2. Постройте базу знаний в виде дерева решений.
3. Разработайте программу на любом языке программирования (или постройте эту экспертную систему с помощью любого готового программного средства).

## Глава 13. Создание веб-сайтов

### Практические работы

#### Практическая работа № 24.

##### Текстовая веб-страница

Перейдите в каталог **RUSKEALA**. Загрузите документ **ruskeala.html** – заготовку для новой веб-страницы. Оформите страницу так, как на образце справа.

##### Уровень А.

- 1) Добавьте тэги, которые должны быть в правильном HTML-документе (`<!DOCTYPE>`, `<html>`, `<head>`, `<body>`).
- 2) В заголовке страницы (`<title>`) должны быть указаны фамилия и имя автора: «Работа Василия Иванова».
- 3) Выделите заголовок документа с помощью соответствующего тэга.
- 4) Выделите абзацы с помощью нужного тэга, установите выравнивание по ширине.
- 5) Замените везде знаки «минус» на тире, сделайте так, чтобы тире не отрывались от предыдущих слов.
- 6) Оформите перечисления как маркированный и нумерованный списки.

##### Уровень В.

- 7) Сделайте названия соборов, музеев и дворцов Санкт-Петербурга ссылками на соответствующие страницы в Интернете.

##### Уровень С.

- 8) Сделайте обозначение трассы E105 гиперссылкой (куда – решите сами).
- 9) Сделайте названия населённых пунктов ссылками на карты сервиса Яндекс-Карты.
- 10) Сделайте адреса веб-сайтов в нижней части страницы гиперссылками на эти сайты.

Работа Василия Иванова

### Мраморный карьер Рускеала

Рускеальские каменоломни — это памятник природы и горного дела. Они находятся в Соргавальском районе, в 24 километрах к северу от города Соргавала, рядом с поселком Рускеала. Мрамор тут добывали еще с конца XVII века. Рускеальский мрамор использовался при украшении очень многих архитектурных шедевров Петербурга, Царского Села, Гатчины. Только с 1769 по 1830 год здесь наломали 200 000 тонн мрамора.

Самым интересным считается Мраморный каньон. Это заполненный естественными водами самый старый карьер открытой добычи камня. Отсюда были получены блоки для облицовки многих архитектурных творений Санкт-Петербурга:

- Исаакиевского собора
- Казанского собора
- Эрмитажа
- Мраморного дворца
- Михайловского замка
- станций метро «Приморская» и «Ладожская»

Добраться до карьера Рускеала от Петрозаводска не так просто:

1. вначале нужно ехать по трассе E105 (M18) до [Прижи](#);
2. свернуть направо и двигаться по трассе P21 в сторону [Соргавала](#);
3. не доезжая до Соргавала, свернуть на [Вартсила](#);
4. проехать поселок [Рускеала](#);
5. затем будет мост через реку Тохмаййоки, сразу за которым будет указатель налево к Мраморному каньону.

Текст с сайтов [www.all-karelia.ru](http://www.all-karelia.ru), [tourskazka.ru](http://tourskazka.ru), [ruskeala.info](http://ruskeala.info).

## Практическая работа № 25.

### Оформление страницы

1. Скопируйте на свой компьютер файл **text.htm**. Посмотрите, как выглядит страница в браузере.
2. Добавьте заголовок страницы (**TITLE**), укажите там название документа и фамилию автора, например:  
*Примеры оформления текста: работа Василия Пупкина*
3. Найдите и выделите соответствующими тэгами заголовков документа и заголовки разделов.
4. Разбейте текст на абзацы.
5. Замените верхние кавычки на «ёлочки».
6. Замените везде знаки «минус» на тире, сделайте так, чтобы тире не отрывались от предыдущих слов.
7. Выделите слова «сервер» и «клиент» в первом абзаце с помощью тэга **EM**.
8. В следующих двух абзацах выделите определяемые слова («сервер» и «клиент») с помощью тэга **DFN**, а английские слова – с помощью тэга **EM**.
9. Оформите математические и химические формулы, используя верхние и нижние индексы. Формула должна занимать отдельную строку и быть выровнена по центру. Точку или запятую, которая следует за формулой, нельзя отрывать от формулы. Исправьте фактические ошибки в формулах, допущенные при наборе.
10. Выделите имена переменных в тексте с помощью тэга **EM**.
11. Оформите программу на языке Паскаль так, чтобы сохранилось все форматирование.

## Практическая работа № 26.

### Оформление страницы с помощью CSS

1. Откройте **text.htm**, полученный в результате предыдущей работы.
2. Создайте в той же папке стилевой файл **mystyle.css**, подключите его к веб-странице и определите стили оформления, перечисленные в следующих пунктах. Для получения дополнительной справочной информации по CSS используйте Интернет, например, сайт [css.manual.ru](http://css.manual.ru).
3. Задайте свои цвета фона и текста. Текст должен хорошо читаться на выбранном фоне.
4. Определите цвет заголовков (**H1**, **H2**), отличающийся от цвета основного текста. Заголовки **H2** должны также выделяться фоном, причем фон должен быть темный, а буквы — светлые. Можно добавить внутренний отступ (**padding**) в 5 пикселей.
5. Определите класс **formula**, применимый как к абзацам (**P**), так и к отдельным словам (**SPAN**), и задайте для него *выравнивание по центру; жирный шрифт; курсив*. Используя этот класс, выделите абзацы с формулами и имена переменных в тексте.
6. Определите класс **definition**, применимый только к абзацам (**P**), и задайте для него: *рамку толщиной в 1 пиксель; внутренний отступ в 10 пикселей*. Используя этот класс, выделите два абзаца с определениями.

Примеры оформления текста: работа Василия Пупкина

#### Примеры оформления текста

##### Сети типа «клиент-сервер»

Часто при организации связи между двумя компьютерами за одним компьютером закрепляется роль поставщика ресурсов (программ, данных и т.д.), а за другим — роль пользователя этих ресурсов. В этом случае первый компьютер называется *сервером*, а второй — *клиентом* или рабочей станцией.

*Сервер* (англ. *serve* — обслуживать) — это высокопроизводительный компьютер с большим объемом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования).

*Клиент* (англ. *client*) — любой компьютер, имеющий доступ к услугам сервера. Клиентом также называют прикладную программу, которая от имени пользователя получает услуги сервера. Соответственно, программное обеспечение, которое позволяет компьютеру предоставлять услуги другому компьютеру, называют сервером — так же, как и сам компьютер.

##### Математические формулы

Пифагор доказал, что

$$a^2 + b^2 = c^2,$$

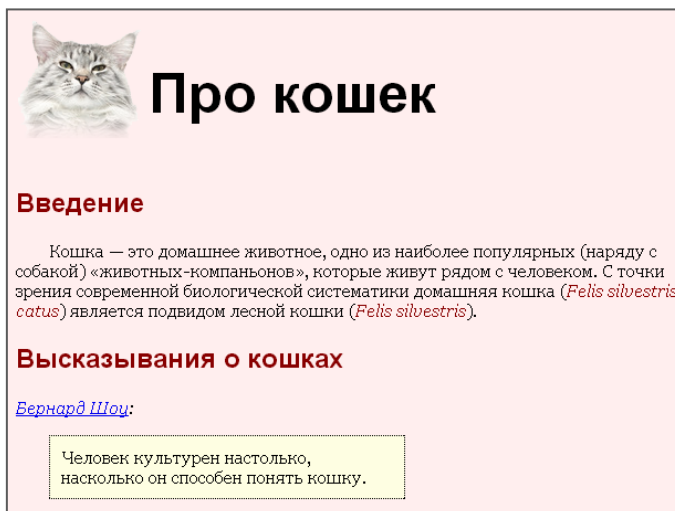
где  $a$  и  $b$  — длины двух катетов, а  $c$  — длина катета прямоугольного треугольника.

Разность кубов чисел может быть представлена в виде

$$a^3 - b^3 = (a-b)(a^2 + ab + b^2).$$

## Практическая работа № 26-2 (дополнительная) Оформление страницы с помощью CSS. Часть 2

1. Скопируйте на свой компьютер каталог CSS. Откройте файл **cats.htm**. Все последующие задания нужно выполнять с помощью CSS.
2. Установите для тела страницы бледно-розовый фон (цвет #FFEEEE) и шрифт *Georgia*, а если его нет – любой шрифт с засечками (*serif*).
3. Для заголовков **H1** и **H2** определите шрифт *Arial*, если его нет – *Helvetica*, а если нет их обоих, то любой рубленый шрифт (*sans-serif*).
4. Для заголовка **H1** задайте фоновый рисунок **cat.png** из подкаталога **images** (без повторения).
5. Для заголовков **H2** определите темно-красный цвет (#800000).
6. Создайте класс оформления **latin** для выделения слов и выражений на латинском языке: курсив, тёмно-красный цвет. Выделите в тексте все латинские слова с помощью этого стиля.
7. Установите для всех абзацев отступ (красную строку) 30 пикселей.
8. Создайте новый стиль оформления абзацев – **author**: курсивный шрифт, без абзацного отступа. Выделите этим стилем имена и фамилии авторов цитат.
9. Создайте стиль оформления абзацев с цитатами с именем **quote**:
  - а) фон – светло-жёлтый (#FEFEE2);
  - б) абзацного отступа нет;
  - в) ширина 50% от ширины окна браузера;
  - г) внешние отступы: сверху – 0, справа и слева по 30 пикселей, снизу – 10 пикселей;
  - д) рамка шириной 1 пиксель, точечная (*dotted*), черного цвета;
  - е) внутренние отступы 10 пикселей.
10. Найдите в Интернете информацию про авторов цитат и сделайте их имена ссылками на соответствующие страницы.
11. Установите для ссылок, которые находятся внутри абзаца стиля **author**, красный цвет при наведении мыши.





## Практическая работа № 27.

### Вставка рисунков в документ

1. Скопируйте на свой компьютер каталог **IMG**. Откройте файл **img.htm** в редакторе. Ваша задача – оформить веб-страницу про некоторые направления в музыке XX века. Для оформления нужно использовать рисунки из подкаталога **images**.
2. Добавьте заголовок страницы (**TITLE**), укажите там название документа и фамилию автора, например:  
*Веб-страница с рисунками: работа Василия Пупкина*
3. Найдите и выделите соответствующими тэгами заголовков документа и заголовки разделов.
4. Разбейте текст на абзацы.
5. Замените верхние кавычки на «ёлочки».
6. Замените везде знаки «минус» на тире, сделайте так, чтобы тире не отрывались от предыдущих слов.
7. Создайте стилевой файл **mystyle.css** и подключите его к веб-странице. С помощью стилевого файла:
  - а) Установите в качестве фона страницы рисунок **back.jpg**;
  - б) Выберите соответствующий цвет текста.
  - в) Создайте класс **name**, определите для него жирный шрифт, курсив и свой цвет, и выделите имена и фамилии, названия ансамблей, фильмов, песен (тэг **SPAN**).
8. Добавьте на страницу рисунки, расположенные в подкаталоге **images**.
9. Установите для рисунков выравнивание (с обтеканием текстом) и всплывающие подсказки. Рисунки лучше ставить в шахматном порядке (один налево, следующий – направо и т.д.)
10. С помощью стилевого файла установите для рисунков внешний отступ (**margin**) 10 пикселей.

## Практическая работа № 28.

### Вставка звука и видео в документ

25. Скопируйте на свой компьютер каталог **MEDIA**. Откройте файл **media.htm** в редакторе. Ваша задача – научиться добавлять на веб-страницу звуковые файлы, флэш-анимацию и видеоролики. Для этого нужно использовать мультимедийные данные из подкаталогов **audio**, **flash** и **video**. Места вставки медиафайлов вы легко определите из текста файла **media.htm**.
26. Добавьте на веб-страницу звуковой файл **hcopter.wav** из каталога **audio** (шум вертолёта). Используйте тэг **embed**, отключите автозапуск проигрывания, установите ширину проигрывателя 300 пикселей и высоту 20 пикселей.
27. Добавьте на веб-страницу звуковой файл **nature.mp3** из каталога **audio** (звуки природы). Используйте тэг **embed**, укажите те же атрибуты, что и в предыдущем случае.
28. Добавьте на веб-страницу звуковой файл **nature.ogg** из каталога **audio** (звуки природы). Используйте тэг **audio**, укажите его атрибуты:  
**src="адрес файла"** ссылка на файл-источник  
**controls="controls"** показывать элементы управления  
**width="300"** ширина проигрывателя в пикселях  
**height="20"** высота проигрывателя в пикселях

Откройте веб-страницу в браузере и выясните, поддерживает ли ваш браузер тэг **audio**.

Ответ:

- 
29. С помощью тэга **embed** добавьте на страницу флэш-ролик **cube.swf** (вращающийся кубик) из каталога **flash**. Установите размеры 257 на 200 пикселей.
  30. С помощью тэга **embed** добавьте на страницу флэш-ролик **lightbot.swf** (игра *Lightbot*) из каталога **flash**. Установите размеры 640 на 480 пикселей. Откройте веб-страницу в браузере и посмотрите, как работает флэш-анимация.
  31. Зайдите на сервис **youtube.com**, с помощью поиска по сайту найдите ролик по теме «HTML5» и вставьте на веб-страницу. Для этого нужно щелкнуть по кнопке *Поделиться* под роликом, затем – по строчке *HTML-код*. После этого остается просто скопировать код, появившийся в текстовом окне, на веб-страницу.
  32. С помощью тэга **video** добавьте на страницу видеофайл **finish.mp4** из каталога **video** следующим образом:

```
<video width="ширина" height="высота" controls="controls">  
<source src="адрес файла"  
      type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>  
</video>
```

Установите размеры окна 320 на 240 пикселей.

33. С помощью тэга **video** добавьте видеофайл **finish.webm** из каталога **video** следующим образом:

```
<video width="ширина" height="высота" controls="controls">  
<source src="адрес файла"  
      type='video/webm; codecs="vp8, vorbis"'>  
</video>
```

Установите размеры окна 320 на 240 пикселей.

34. Обновите страницу в браузере и выясните, поддерживает ли ваш браузер тэг **video**, и с какими форматами он умеет работать.

Ответ:

## Практическая работа № 29.

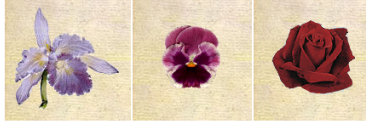
### Таблицы

1. Скопируйте на свой компьютер каталог **TABLE**. Откройте файл **table.htm** в редакторе. Ваша задача – оформить веб-страницу так, как показано на образце (см. ниже на этой странице). Для этого нужно использовать рисунки из подкаталога **images** и стилевой файл.
2. Добавьте заголовок страницы (**title**), укажите название документа и фамилию автора, например:  
*Таблицы: работа Василия Пупкина.*
3. С помощью стилевого файла установите для всех таблиц вертикальный отступ (**margin**) 10 пикселей.
4. Присвойте первой таблице класс **sport** для того, чтобы можно было управлять оформлением её ячеек независимо от других таблиц.
5. Введите класс **left** для ячеек первой таблицы (селектор **table.sport td.left**), установите для них выравнивание по левой границе. Для остальных ячеек таблицы (селектор **table.sport td**) установите (с помощью стилевого файла) выравнивание по центру. Ячейки, содержащие заголовки и места, занятые командами, оформите с помощью тэга **th**.
6. Для всех ячеек первой таблицы установите внутренний отступ 2 пикселя по вертикали и 5 пикселей по горизонтали.
7. Для второй таблицы используйте три класса ячеек:
  - header** фоновый рисунок **back.jpg** из каталога **images**;; белый цвет символов, шрифт без засечек (**sans-serif**), жирный, выравнивание по центру
  - picture** выравнивание по центру, фоновый рисунок **paper.jpg** из каталога **images**; внутренний отступ (**padding**) 10 пикселей, ширина ячейки (**width**) 120 пикселей
  - note** шрифт Arial, Helvetica или любой шрифт без засечек (**sans-serif**), цвет символов #800000, выравнивание по центру, внутренний отступ (**padding**) 5 пикселей сверху и снизу, и 10 пикселей по бокам, шрифт без засечек
8. Замените верхние кавычки на «ёлочки».
9. Выделите латинские слова и их перевод с помощью тэга **em**.
10. Рекламный блок в третьей таблице расположен в отдельной ячейке, которой присвоен класс **ad**: рамка цвета #CCCCCC толщиной 1 пиксель, фоновый цвет **rgb(230,230,255)**, выравнивание по центру. Текст должен быть ссылкой на веб-страницу в Интернете с рецептом салата «Цезарь».

## Данные в таблицах

Команда	Баллы				Место
	I тур	II тур	III тур	Итого	
«Ротор»	10	15	8	33	I
«Статор»	12	15	3	30	II
«Мотор»	15	10	3	28	III

## Гербарий Вассуалия Лоханкина



Орхидея

Анютины глазки

Роза

## Табличная вёрстка

## Юлий Цезарь



В 44 г. до н. э. Юлий Цезарь, достигнув высшей власти и осыпанный всеми титулами и почестями, которые могла дать республика, провозглашает себя пожизненным диктатором. Сторонники республики убивают Цезаря. Но римляне не поддержали убийц Цезаря, несмотря на все их благородные намерения. Вспыхнула гражданская война, в результате которой Римская республика пала, и Рим стал империей. Цезарь был кумиром для многих политиков будущего, и его слова «*Veni. Vidi. Vici*» (лат. «*Пришел. Увидел. Победил*») стали девизом для многих.



Рецепт салата  
«Цезарь»

## Практическая работа № 30.

### Блоки

В этой работе вы освоите трёхколоночную верстку без использования таблиц.

1. Скопируйте на свой компьютер каталог **DIV**. Откройте файл **baikal.htm** в редакторе. Ваша задача – оформить веб-страницу так, как показано на образце (см. на следующей странице). Для этого нужно использовать рисунки из подкаталога **images** и стилевой файл.
2. Добавьте заголовок страницы (**TITLE**), укажите название документа и фамилию автора, например:  
*Блочная вёрстка: работа Василия Пупкина.*
3. Создайте стилевой файл **baikal.css** и подключите его к веб-странице. Установите для всех элементов страницы (они обозначаются знаком **\***) нулевые внутренние и внешние отступы:

```
* {  
    margin:0px;  
    padding:0px;  
}
```
4. Добавьте в блок с идентификатором **header** заголовок документа «Озеро Байкал». В стилевом файле установите для этого блока высоту 80 пикселей и фоновый рисунок **header.jpg** из каталога **images** (без повторения). Добавьте для заголовка отступы слева и сверху (как на образце).
5. Добавьте в блок **pogoda** два скрипта, которые записаны в файле **informer.txt**. Они выводят на страницу информацию о погоде в двух посёлках на берегу Байкала – в Листвянке и Хужире.
6. Добавьте в блок **photo** четыре фотографии с именами **baikal1.jpg**, **baikal2.jpg**, **baikal3.jpg**, **baikal4.jpg** из подкаталога **images**.
7. Добавьте в блок **content** текст, записанный в файле **text.txt**. Оформите абзацы с помощью тэга **p**. В стилевом файле для тэгов **p** установите отступы со всех сторон 5 пикселей и абзацный отступ (красную строку) 20 пикселей.
8. В тексте замените, где нужно, знаки «минус» на тире, перед тире поставьте неразрывные пробелы. Поставьте неразрывные пробелы между числами и единицами измерения.  
Добавьте в конец текста абзац со словами  
*По материалам [Википедии](#).*  
Выровняйте абзац по правой границе и выделите этот текст с помощью тэга **em**. Слово «Википедии» должно быть ссылкой на страницу Википедии, посвящённую Байкалу.
9. Добавьте в блок **footer** текст, содержащий вашу фамилию, имя и класс, например,  
*Работу выполнил Василий Пупкин, 11<sup>А</sup> класс*
10. Посмотрите, как выглядит страница в браузере. Теперь остается с помощью стилей оформить блоки так, чтобы они заняли правильное место на странице.
11. В стилевом файле для блока **pogoda** установите ширину 140 пикселей, отступ слева 10 пикселей и обтекание слева (**float:left;**). Посмотрите, как теперь выглядит страница.
12. В стилевом файле для блока **photo** установите ширину 220 пикселей, отступ слева 10 пикселей и обтекание справа (**float:right;**). Посмотрите, как теперь выглядит страница.
13. Теперь сделаем так, чтобы центральная колонка (блок **content**) не залезала на левую и правую. В стилевом файле для блока **content** установите отступ слева 140 пикселей и отступ справа 220 пикселей. Посмотрите, как теперь выглядит страница.
14. Остается оформить нижний блок («подвал»). В стилевом файле установите для блока **footer** цвет фона **#CCCCCC**, внутренние отступы – сверху 5 пикселей, снизу –

10 пикселей, выравнивание текста по центру. Для того, чтобы этот блок не обтекался другими ни слева, ни справа, добавьте свойство **clear:both;**

15. Посмотрите на окончательный результат в браузере:

## Озеро Байкал

**погода@mail.ru**  
Листвянка  
+6°  
утром: +13°  
завтра: +10°  
Давление 719 mmHg  
Влажность 72%  
Ветер 2 м/с В  
небольшой дождь  
подробный прогноз

**погода@mail.ru**  
Хужир  
+4°  
утром: +4°  
завтра: +8°  
Давление 718 mmHg  
Влажность 78%  
Ветер 2 м/с СВ  
небольшой дождь  
подробный прогноз

Байкал — озеро в южной части Восточной Сибири, самое глубокое озеро на планете, крупнейшее в мире хранилище запаса пресной воды.

Озеро находится в своеобразной котловине, со всех сторон окружённой горными хребтами и сопками. Западное побережье — скалистое и обрывистое, рельеф восточного побережья более пологий, местами горы отступают от берега на десятки километров.

В байкальской воде очень мало растворённых и взвешенных минеральных веществ, ничтожно мало органических примесей, много кислорода.

Вода в озере настолько прозрачна, что отдельные камни и различные предметы бывают видны на глубине 40 м. В это время байкальская вода бывает синего цвета. Летом же и осенью, когда в прогретой солнцем воде развивается масса растительных и животных организмов, прозрачность её снижается до 8—10 м, и цвет становится синезелёным и зелёным.

Вода в Байкале холодная. Температура поверхностных слоёв даже летом не превышает +8...+9°C, в отдельных заливах — до +15 °С. Температура глубинных слоёв — около +4°C.


Как правило, желающие посетить Байкал сначала отправляются в один из ближайших крупных городов: Иркутск, Улан-Удэ или Северобайкальск. Проезжая по Транссибирской магистрали между Иркутском и Улан-Удэ, можно часами любоваться видами озера, простирающегося прямо за окном поезда.

В 70 км от Иркутска, на берегу Байкала возле истока Ангары расположен посёлок Листвянка — одно из самых популярных мест туризма на Байкале. Добраться сюда из областного центра можно на автобусе или теплоходе всего за час с небольшим. Самые популярные маршруты проходят из посёлка Листвянка до Больших Котов, на полуострове Святой Нос.

Одним из красивейших уголков на Байкале считается Чивыркуйский залив, который богат живописными бухтами, загадочными островами, целебными источниками. Хороший вид на залив открывается с вершин Святого Носа, на который можно попасть из посёлка Усть-Баргузин.

На различных участках вокруг озера проходит Большая Байкальская Тропа — система экологических троп и один из прекраснейших способов для туристов увидеть уникальную природу и насладиться захватывающими видами и панорамами Байкала.

*По материалам [Википедии](#).*



Работу выполнил Василий Пупкин, 11<sup>А</sup> класс

## Практическая работа № 31.

### Использование XML

1. Скопируйте на свой компьютер в отдельные папки файлы **min.docx** (документ *Microsoft Office 2007*) и **min.odt** (документ *OpenOffice.org Writer*). Переименуйте файла так, чтобы они имели расширение **\*.zip**, и распакуйте любым архиватором, который поддерживает этот метод сжатия.

Файлы с какими расширениями вы увидели после распаковки архивов? Просмотрите эти файлы в любом текстовом редакторе, изучите их структуру.

Ответ:

2. Скопируйте на свой компьютер файл **rect.svg** (векторный рисунок в формате SVG) и откройте его в браузере *Google Chrome*, *Mozilla Firefox* или *Opera*. Затем откройте этот файл в любом текстовом редакторе и изучите его структуру.
3. Скопируйте на свой компьютер файл **math.htm** (веб-страница с формулой в формате MathML) и откройте его в браузере *Mozilla Firefox* или *Opera*. Затем откройте этот файл в любом текстовом редакторе и изучите его структуру.
4. Найдите на компьютере файлы конфигурации с расширением **.xml**. В операционной системе *Microsoft Windows* используйте для поиска каталог *Program Files*, а в *Linux* – каталог */usr*. Откройте один из найденных файлов в браузере или в любом текстовом редакторе и изучите его структуру.

#### База данных в формате XML

1. Скопируйте на свой компьютер файлы **europe.xml**, **europe2.xml** и **europe.xsl**. Первые два файла содержат базу данных по странам Европы (в разных форматах), а с помощью третьего мы будем задавать оформление данных на веб-странице.

Работа может выполняться с помощью браузеров *Mozilla Firefox*, *Opera*, *Internet Explorer* или *Google Chrome*<sup>11</sup>.

2. Откройте файл **europe.xml** в браузере, используя команду меню *Файл – Открыть* и изучите его структуру. Кроме того, файл **europe.xml** можно открыть в любом текстовом редакторе. В первой строке

```
<?xml version="1.0" encoding="UTF-8"?>
```

записана используемая версия языка XML и кодировка (здесь – UTF-8). Далее расположены сведения о странах Европы в таком формате:

```
<СтраныЕвропы>
  <Страна>
    <Название>Греция</Название>
    <Столица>Афины</Столица>
    <Население>11</Население>
    <Площадь>132</Площадь>
  </Страна>
```

```
...
</СтраныЕвропы>
```

Корневой элемент в этом XML-файле называется *СтраныЕвропы*, в него вложено несколько элементов *Страна*, каждый из которых имеет вложенные элементы: *Название*, *Столица*, *Население* и *Площадь*.

#### Вывод данных на веб-страницу

3. В текстовом редакторе добавьте в файл вторую строку, которая подключает стилевой файл **europe.xsl**:

```
<?xml-stylesheet type='text/xsl' href='europe.xsl'?>
```

<sup>11</sup> Для того, чтобы выполнить работу в *Google Chrome*, нужно запустить его с ключом `--allow-file-access-from-files`.

Файлы с расширением XSL (от англ. *eXtensible Stylesheet Language*, расширяемый язык стилей) задают правила преобразования XML-документа в другой формат, например, в HTML. Браузер, открывая XML-файл с подключенной таблицей стилей, строит HTML-документ. Посмотрите, как выглядит теперь файл **europe.xml**, если открыть его в браузере.

4. Откройте файл **europe.xsl** в текстовом редакторе и изучите его. Найдите две строчки, которые выбирают информацию из базы данных:

```
<xsl:value-of select="//Страна/Название"/>
<xsl:value-of select="//Страна/Столица"/>
```

Обратите внимание, что это пустые элементы (без содержания), они заканчиваются знаками «`</>`», то есть сразу закрываются.

Здесь *Название* и *Столица* – это названия полей (тэгов) в XML-файле. Оба эти поля находятся внутри контейнера *Страна*. Запись `//Страна/Название` означает «значение поля *Название* в контейнере *Страна*». В XML-файле хранятся данные по многим странам, проверьте, данные какой страны появятся на экране:

Ответ:

Почему появились данные именно этой страны?

Ответ:

5. В XSL-файле можно использовать тэги языка HTML. Например, можно заключить данные в таблицу:

```
<table border="1">
<tr>
<td><xsl:value-of select="//Страна/Название"/></td>
<td><xsl:value-of select="//Страна/Столица"/></td>
</tr>
</table>
```

Самостоятельно добавьте в таблицу оставшиеся данные (население в миллионах человек и площадь страны в тысячах квадратных километров), а также заголовки столбцов (используйте тэг «**th**»).

6. Откройте файл **europe.xml** в браузере и убедитесь, что вы видите таблицу, где указаны все данные одной страны.

Для того чтобы вывести данные по всем странам, используем цикл. Для этого в стилевой файл нужно добавить строчки, обозначающие начало и конец цикла:

```
<table border="1">
<tr>
<th>Страна</th>
<th>Столица</th>
...
</tr>
<xsl:for-each select="//Страна">
<tr>
<td><xsl:value-of select="Название"/></td>
<td><xsl:value-of select="Столица"/></td>
...
</tr>
</xsl:for-each>
</table>
```

В данном случае цикл выполняется по всем элементам *Страна*, то есть, по всем странам. Здесь двойной слэш «`//`» перед словом «Страна» показывает, что цикл применяется ко всем элементам с таким названием, независимо от уровня



вложенности. Вместо этого можно было указать точный адрес элементов Страна так: «СтраныЕвропы/Страна» – это означает «все элементы Страна внутри элемента СтраныЕвропы».

**Обратите внимание, что теперь в тэгах «xsl:value-of» значение атрибута «select» – это просто название нужного поля, а символы «//Страна/» нужно убрать. Это означает, что нас интересует поле именно в том контейнере Страна, который выбран в цикле (а не в первом попавшемся).**

Добавьте цикл в нужное место в XSL-файл и обновите файл в браузере. Вы должны увидеть в таблице информацию по всем странам.

### Сортировка

7. Таблицу можно отсортировать по любому столбцу. Для этого в стилевой файл сразу после начала цикла добавим элемент

```
<xsl:sort order="ascending" select="Название"/>
```

Сортировка выполняется по полю *Название* в порядке возрастания (англ. *ascending*). Для сортировки по убыванию значение параметра **order** (англ. *порядок*) должно быть равно **descending** (англ. *убывающий*). Посмотрите, как теперь выглядит файл в браузере.

8. Самостоятельно измените порядок сортировки: сделайте сортировку по убыванию количества жителей страны. Подумайте, почему результат отличается от ожидаемого.

Ответ:

9. Для того чтобы правильно выполнить сортировку по числовым данным, в команду сортировки нужно добавить параметр **data-type** (тип данных), который должен быть равен **number** (англ. *число*):

```
<xsl:sort ... data-type="number"/>
```

Внесите это изменение и проверьте правильности сортировки.

### Выборка

10. С помощью стилевого файла можно задать условие отбора данных. Для этого применяется элемент **xsl:if**. Этот парный элемент (контейнер) нужно вставлять внутрь цикла после элемента сортировки:

```
<xsl:for-each ...>
  <xsl:sort .../>
  <xsl:if test="Население > 20">
    ...
  </xsl:if>
</xsl:for-each>
```

В этом примере отбор идет по значению поля *Население*. Выбираются только те страны, у которых население более 20 млн человек. Проверьте работу этого элемента.

Можно также использовать другие операторы сравнения:

> или <b>&amp;gt;</b>	больше или равно (англ. <i>greater than</i> )
>= или <b>&amp;gt;=</b>	больше или равно (англ. <i>greater than</i> )
<b>&amp;lt;</b>	меньше (англ. <i>less than</i> )
<b>&amp;lt;=</b>	меньше или равно

Знаки > и < в коде веб-страниц заменяют соответственно на **&gt;** и **&lt;**; для того, чтобы браузер не путал их с угловыми скобками, ограничивающими тэги. Для знака < такая замена в стилевом XSL-файле обязательна.

11. Самостоятельно сделайте выборку по тем странам, площадь которых меньше, чем 200 тысяч км<sup>2</sup>, отсортируйте строки таблицы по возрастанию площади стран.
12. Вложенные элементы **xsl:if** дают эффект логической операции «И» (требуют выполнения двух условий одновременно). Добавьте в условие отбора еще одно

ограничение: население больше, чем 10 млн. человек. Проверьте результат в браузере.

13. \*Подумайте, как можно добиться эффекта логической операции «ИЛИ».

#### Условное форматирование

14. С помощью элемента **xsl:if** можно сделать условное форматирование, то есть изменять свойства HTML-элементов. Например, чтобы изменить цвет для всех чётных строк таблицы, после открывающего тэга **tr** внутри цикла нужно добавить строки:

```
<xsl:if test="position() mod 2 = 0">
  <xsl:attribute name="bgcolor">#E6E6FF</xsl:attribute>
</xsl:if>
```

Если позиция элемента (она вычисляется функцией **position**) – чётное число (остаток от деления на 2 равен 0), то для всей строки (предыдущего элемента **tr**)

устанавливается свойство **bgcolor** (фоновый цвет), равный **#E6E6FF**.

Проверьте, как выглядит страница в браузере после этих изменений.

Вы заметите, что строки раскрашены неправильно (фон не чередуется). Это произошло потому, что функция **position** использует счётчик цикла, который перебирает все страны в базе, хотя фактически выводятся данные только по тем странам, который соответствуют двум вложенным условиям отбора.

15. Чтобы исправить ситуацию, сделаем так, чтобы цикл работал только для тех стран, которые нам нужны. Для этого уберем оба контейнера-условия **xsl:if** и изменим заголовок цикла, включив в него оба условия, связанные логической операцией «**and**» (И):

```
<xsl:for-each
  select="//Страна[Население > 10 and Площадь < &lt;200]">
```

Проверьте, что после этого и отбор данных, и раскраска строк работают правильно.

В сложных условиях можно использовать также логическую операцию «**or**» (ИЛИ).

16. Измените условие так, чтобы отобразить страны, население которых меньше 20 млн. человек, а также все страны, площадь которых больше 100 тыс. км<sup>2</sup>.

#### Подключение CSS-файлов

17. Создайте новый CSS-файл **europe.css** и подключите его к веб-странице. Для этого нужно ставить ссылку на него внутрь контейнера **head** в XSL-файле. Установите с помощью CSS серый цвет для ячеек таблицы, заданных тэгом **TH**. Проверьте результат в браузере.

18. Установите в CSS-файле для ячеек таблицы внутренний отступ по вертикали 2 пикселя и по горизонтали 5 пикселей.

19. Присвойте ячейкам, в которых выводится население и площадь стран, класс **number**, и для него установите в CSS-файле выравнивание по правой границе. Окончательный результат должен выглядеть примерно так:

Страна	Столица	Население	Площадь
Нидерланды	Амстердам	15	42
Португалия	Лиссабон	11	92
Греция	Афины	11	132

#### Работа с атрибутами

20. Откройте файл **europe2.xml** в браузере, используя команду меню *Файл – Открыть* и изучите его структуру. Как видите, эта база строится иначе – свойства стран заданы не как вложенные элементы, а как атрибуты, например:

```
<Страна Название="Греция" Столица="Афины"
  Население="11" Площадь="132" />
```

Как ни странно, переход к такому формату базы данных потребует минимальных изменений в стилевом файле. Для обращения к атрибутам нужно перед названием атрибута просто поставить знак @.

21. Скопируйте XSL-файл **europe.xml**, назвав копию **europe2.xml**. Перед всеми названиями атрибутов (которые раньше были названиями вложенных элементов) добавьте знак @ (так вместо «**Название**» нужно написать «**@Название**»).
22. Подключите к файлу **europe2.xml** таблицу стилей **europe2.xsl** и проверьте, что получается тот же самый результат, что и раньше.

На этом выполнение основной части работы закончено.

### Рисунки и ссылки<sup>12</sup>

В XML-файле можно хранить пути к файлам изображений и ссылки на ресурсы Интернета. Как известно, адрес изображения на веб-странице указывается как значение атрибута **src** тэга **<img>**, а адрес перехода по гиперссылке – как значение атрибута **href** тэга **<a>**. Проблема состоит в том, что значение, взятое из XML-файла, нужно «встроить» внутрь тэга веб-страницы, передать как значение атрибута. Для этого используют специальный элемент языка XSL, который называется **xsl:attribute** (мы уже встречались с ним в разделе *Условное форматирование*).

Пусть в XML-файле хранятся данные об автомобилях в виде записей следующего формата:

```
<Авто Название="Ford Fusion" Фото="images/fordfsn.jpg"
Ссылка="http://www.ford.com/cars/fusion"/>
```

Для того, чтобы вставить рисунок, адрес которого записан как атрибут **Фото**, в стилевом XSL-файле используем следующий код:

```
<img>
  <xsl:attribute name="src">
    <xsl:value-of select="@Фото"/>
  </xsl:attribute>
</img>
```

Элемент **xsl:attribute** изменяет значение атрибута **src** тэга **<img>**, внутри которого он находится. Новое значение атрибута будет равно значению атрибута **Фото** записи из базы данных.

Аналогично меняется атрибут **href** у гиперссылки:

```
<a>
  <xsl:attribute name="href">
    <xsl:value-of select="@Ссылка"/>
  </xsl:attribute>
  <xsl:value-of select="@Название"/>
</a>
```

В данном примере из записи базы данных берем два атрибута: **Название** (оно становится текстом ссылки) и **Ссылка** (адрес перехода).

**Задание для самостоятельной работы.** Постройте базу данных в формате XML, содержащую рисунки и адреса перехода по гиперссылкам, и напишите стилевой файл для вывода этих данных на веб-страницу.

### Использование Javascript<sup>13</sup>

Покажем, как можно использовать язык программирования *JavaScript* для управления выводом данных из XML-файла на веб-страницу. Наша задача – сделать заголовки столбцов гиперссылками, щелчок по которым сортирует данные по выбранному столбцу. Как вы знаете, сортировка данных задается в таблице стилей. Поэтому задача сводится к тому, чтобы после щелчка по гиперссылке динамически *заменить таблицу стилей* текущей веб-страницы. При этом нельзя, как мы делали раньше, жёстко связывать XML-

<sup>12</sup> Дополнительное задание для любознательных.

<sup>13</sup> Дополнительное задание для любознательных.

файл со стилевым файлом, поэтому ссылку на стилевой файл в начале XML-документа нужно убрать.

Основным файлом будет HTML-страница, которую мы назовем **europe.htm**. После загрузки этой страницы на неё будут динамически подгружаться данные из XML-файла и некоторая таблица стилей (начальный вариант, без сортировки). После щелчка по гиперссылке таблица стилей должна заменяться. Это означает, что придётся создать несколько стиливых таблиц с расширением **.xsl**, которые будут отличаться только порядком сортировки данных.

1. Скопируйте файлы **europe.css**, **europe2.xml** и **europe2.xsl** в отдельный каталог (например, с именем **JS**).
2. Создайте веб-страницу **europe.htm** следующего содержания:

```
<html>
<head>
<link rel="stylesheet" href="europe.css" type="text/css" />
</head>
<body>
  <div id="data">
  </div>
</body>
</html>
```

На этой странице подключается стилевой файл **europe.css** и в теле документа размещен блок с идентификатором **data**, в который мы будем загружать данные из XML-файла.

3. Скопируйте в каталог **JS** файл **loadxml.js**, в котором записаны функции для динамической загрузки XML-документа и стилевой XSL-таблицы<sup>14</sup>. Подключите этот файл к веб-странице с помощью тэга **script**.
4. Удалите из файла **europe2.xml** строку, в которой подключается стилевой файл **europe2.xsl**.
5. Удалите из файла **europe2.xsl** строки, содержащие HTML-тэги (теперь они находятся в теле HTML-страницы), а также команды для сортировки и поиска.
6. Добавьте в тело веб-страницы скрипт с функцией **init**, которая загружает XML-документ **europe2.xml** и таблицу стилей **europe2.xsl** в блок с идентификатором **data**:

```
<script type="text/javascript">
function init() {
  source = loadXMLDoc("europe2.xml");
  loadStyle("europe2.xsl", "data");
  return true;
}
</script>
```

и вызовите функцию **init** в обработчике события **onLoad** (он срабатывает сразу после загрузки страницы):

```
<body onLoad="init();">
```

Откройте веб-страницу **europe.htm** в браузере и убедитесь, что выводятся данные по всем странам.

7. Теперь осталось организовать сортировку. Заголовки столбцов таблицы находятся в стилевом файле **europe2.xsl**, их нужно сделать гиперссылками, так чтобы при щелчке по ним вызывались функции, выполняющие сортировку (подгружающие другую таблицу стилей). Например, вместо строки

```
<th>Страна</th>
```

нужно записать

---

<sup>14</sup> Изучая этот файл, вы заметите, что для браузера *Internet Explorer* предусмотрен особый вариант действий, потому что он не поддерживает команды, работающие в остальных браузерах.

```
<th>  
<a href="#" onClick="loadStyle('europe2n.xsl','data');" >Страна</a>  
</th>
```

Вызов функции **loadStyle** в обработчике события **onClick** (при щелчке мышью) загружает таблицу стилей **europe2n.xsl**, которая должна выполнить сортировку по названию страны.

Тем же способом превратите все заголовки столбцов таблицы в гиперссылки.

Конечно, имена таблиц стилей должны быть разными. Например, последняя буква названия XSL-файла может обозначать порядок сортировки: **n** (от англ. *name*) – по названию страны; **c** (от англ. *capital*) – по названию столицы; **p** (от англ. *people*) – по числу жителей; **a** (от англ. *area*) – по площади.

8. Скопируйте основную таблицу стилей **europe2.xsl** четыре раза, получив файлы с теми именами, которые указаны в гиперссылках. В каждый из них добавьте нужный вариант сортировки.
9. Откройте веб-страницу **europe.htm** в браузере и убедитесь, что все варианты сортировки работают правильно. Особое внимание обратите на сортировку по числовым данным (количеству жителей и площади).

## Практическая работа № 32.

### Динамический HTML

1. Скопируйте на свой компьютер каталог **SCRIPT**. Откройте в редакторе файл **valaam.htm** и посмотрите, как он выглядит в браузере.  
Ваша задача – оформить документ так, как показано на следующей странице. В нем будет один скрытый блок. Для оформления используйте рисунки из каталога **images**.
2. Добавьте тэги, необходимые для правильного HTML-документа. Добавьте заголовок страницы (**TITLE**), укажите название документа и фамилию автора, например:  
*Валаам: работа Василия Пупкина.*
3. Выделите заголовок документа стилем **H1**. Справа от заголовка добавьте рисунок **valaam\_gray.gif**, который будет ссылкой на сайт **valaam.ru**. При наведении мыши рисунок должен меняться на **valaam.gif**.
4. Выделите абзацы в тексте с помощью тэга **P**.
5. Создайте стилевой файл **valaam.css** и файл для скриптов **valaam.js**, подключите эти файлы к документу. Все оформление должно быть сделано с помощью CSS, все скрипты «убраны» в файл **valaam.js**.
6. Замените верхние кавычки на «ёлочки».
7. Замените везде, где нужно, обычные пробелы на неразрывные, и знаки «минус» – на тире; сделайте так, чтобы тире не отрывались от предыдущих слов.
8. Сделайте плавающий блок, содержащий фотографию Валаама и подпись «Фото А. Колыбалова ([www.rg.ru](http://www.rg.ru)) ». Адрес сайта должен быть ссылкой на этот сайт.  
Подпись должна быть набрана шрифтом без засечек (**sans-serif**), курсивом, размер 80% от размера шрифта основного текста, внешних отступов (**margin**) нет.
9. Оформите скрытый блок, включающий ту часть текста, которой нет на экране в краткой версии (см. оборот). Присвойте этому блоку имя (**id**), установите следующее оформление: цвет фона #EEEEFF; рамка синяя, сплошная, толщиной 1 пиксель; отступы по бокам 10 пикселей.
10. В нижнюю часть скрытого блока добавьте внутренний плавающий блок, содержащий фотографию резьбы по дереву с подписью ‘*К. Гоголев. «На пристани» (резьба по дереву)*’. Оформление подписи должно быть такое же, как и для первого плавающего блока.
11. Сделайте так, чтобы скрытый блок показывался при щелчке по словам «гениев творчества и науки».
12. В конце скрытого блока должна быть ссылка с текстом «Свернуть», при щелчке по которой блок скрывается.
13. В конце документа добавьте форму с вопросом «Назовите ближайший город, из которого можно приехать на о. Валаам» (правильный ответ – Сортавала). Выделите вопрос с помощью тэга **STRONG**. Сделайте так, чтобы после щелчка по кнопке *Ответить* на экран выдавалось сообщение «*Правильно*» или «*Неправильно*».

## Остров Валаам [✿ valaam.ru](http://valaam.ru)

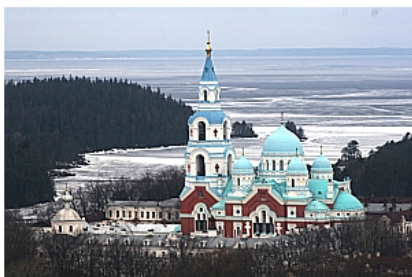


Фото А. Колыбалова ([www.rq.ru](http://www.rq.ru))

императорской фамилии. Также приезжал на Валаам святитель Игнатий (Брянчанинов). Природа Валаама вдохновляла известнейших [гениев творчества и науки](#).

Валаам — остров в северной части Ладожского озера, самый большой в составе Валаамского архипелага. На острове расположен посёлок Валаам, входящий в Сортавальское городское поселение, и Валаамский ставропигиальный мужской монастырь, являющийся памятником русского зодчества. Название острова, возможно, происходит от финно-угорского слова «валамо» — высокая (горная) земля.

Остров неоднократно посещали императоры Александр I и Александр II, другие члены императорской фамилии. Также приезжал на Валаам святитель Игнатий (Брянчанинов). Природа Валаама вдохновляла известнейших [гениев творчества и науки](#).

Назовите ближайший город, из которого можно приехать на о. Валаам:

Ответить

Полная версия (развернутый вид):

## Остров Валаам [✿ valaam.ru](http://valaam.ru)

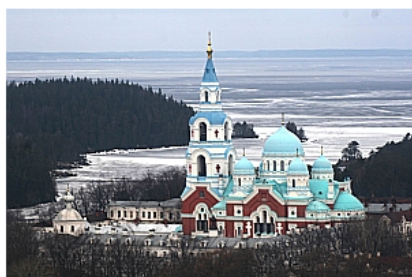


Фото А. Колыбалова ([www.rq.ru](http://www.rq.ru))

императорской фамилии. Также приезжал на Валаам святитель Игнатий (Брянчанинов). Природа Валаама вдохновляла известнейших [гениев творчества и науки](#).

Валаам — остров в северной части Ладожского озера, самый большой в составе Валаамского архипелага. На острове расположен посёлок Валаам, входящий в Сортавальское городское поселение, и Валаамский ставропигиальный мужской монастырь, являющийся памятником русского зодчества. Название острова, возможно, происходит от финно-угорского слова «валамо» — высокая (горная) земля.

Остров неоднократно посещали императоры Александр I и Александр II, другие члены императорской фамилии. Также приезжал на Валаам святитель Игнатий (Брянчанинов). Природа Валаама вдохновляла известнейших [гениев творчества и науки](#).

Здесь побывали художники И.И. Шишкин, Ф.А. Васильев, А.И. Куинджи, писатели и поэты Н.С. Лесков, Ф.И. Тютчев, А.Н. Апухтин, И.С. Шмелёв, Б.К. Зайцев, композиторы П.И. Чайковский, А.К. Глазунов, учёные М.Н. Миклухо-Маклай, Д.И. Менделеев и множество других. В XIX веке во время своего путешествия по России остров посетил Александр Дюма-отец.

Хорошо известны валаамские пейзажи, написанные И.И. Шишкиным («Вид Валаама», 1860), А.И. Куинджи («На острове Валаам», 1873) и Николаем Рерихом («Святой остров», 1917). Ряд современных художников, в частности, известный петрозаводский график А.И. Авдышев, в 1970-х годах создал серию черно-белых линогравюр. Приезжал на Валаам и посвящал ему свои работы мастер объёмной резьбы по дереву, сортавальский художник Кронид Гоголев.



К. Гоголев. «На пристани»  
(резьба по дереву)

[Свернуть](#)

Назовите ближайший город, из которого можно приехать на о. Валаам:

Ответить

## Практическая работа № 33.

### Услуги хостинга

1. Сравните 3-4 бесплатных хостинга сайтов по следующим критериям:

- место, выделяемое под сайт
- ограничение трафика
- операционная система
- наличие систем управления содержанием (CMS)
- возможность использования PHP
- возможность использования СУБД MySQL
- доступ по FTP
- количество почтовых ящиков
- наличие рекламы

Результаты оформите в виде таблицы.

Для сравнения можно использовать сайты из Яндекс-каталога

[http://yaca.yandex.ru/yaca/cat/Computers/Internet/Hosting/Free\\_hostings/](http://yaca.yandex.ru/yaca/cat/Computers/Internet/Hosting/Free_hostings/)

или приведенного ниже списка

<http://ucoz.ru>

<http://webservis.ru/>

<http://www.hostinger.ru/>

<http://www.holm.ru/>

<http://www.hut.ru/>

<http://wallst.ru/>

<http://www.fatal.ru/>

<http://www.radyx.ru/>

<http://freehost.int.ru/>

Если вы владеете английским языком, добавьте к сравнению англоязычные бесплатные хостинги, например:

<http://www.000webhost.com/>

<http://0fees.net/>

<http://byethost.com/>

<http://vlexofree.com/>

Сделайте выводы:

Ответ:

2. Сравните по тем же критериям 3-4 минимальных тарифа на платных хостингах сайтов. Для сравнения можно использовать сайты из Яндекс-каталога

[http://yaca.yandex.ru/yaca/cat/Computers/Internet/Hosting/Paid\\_hostings/](http://yaca.yandex.ru/yaca/cat/Computers/Internet/Hosting/Paid_hostings/)

или приведенного ниже списка

<https://www.nic.ru/>

<http://www.hc.ru/ru/>

<http://www.mchost.ru/>

<http://masterhost.ru/>

<http://sprinthost.ru/>

<http://dehost.biz/>

3. Сделайте выводы:

Ответ:



## Глава 14. Элементы теории алгоритмов

### Практические работы

#### Практическая работа № 34.

#### Машина Тьюринга

1. Наберите программу из учебника (или из презентации), которая увеличивает двоичное число на 1 и проверьте её работу. Будет ли правильно работать эта программа, если вначале каретка расположена справа от числа? Почему?

Ответ:

2. Измените программу для увеличения двоичного числа на 1 так, чтобы она работала правильно, если вначале каретка расположена справа от числа.

	$q_1$	$q_2$
0	→	$1 \cdot q_0$
1	→	$0 \leftarrow$
□	$\leftarrow q_2$	$1 \cdot q_0$

3. Опишите алгоритм работы программы для машины Тьюринга:

	$q_1$	$q_2$
a	$q_2$	□ ←
$\bar{b}$	$q_2$	□ ←
□	←	$q_0$

Ответ:

При каком начальном состоянии ленты и положении каретки эта программа заклинивается?

Ответ:

4. Составьте программу для машины Тьюринга, которая заменяет в двоичном числе все 0 на 1 и все 1 на 0 (из числа 10101100 получается 01010011). Каретка находится слева от числа.

	$q_1$	$q_2$
0		
1		
□		

Описание состояний:

$q_1 -$   
 $q_2 -$

5. Составьте программу для машины Тьюринга, которая умножает двоичное число на 2. Каретка находится над числом.

	$q_1$
0	
1	
□	

Описание состояний:

$q_1 -$

6. Составьте программу для машины Тьюринга, которая увеличивает троичное число на 1. Каретка находится справа от числа.

	$q_1$	$q_2$
0		
1		
2		

<input type="checkbox"/>		
--------------------------	--	--

Описание состояний:

$q_1$  –  
 $q_2$  –

При каком начальном положении каретки эта программа зацикливается?

Ответ:

7. Составьте программу для машины Тьюринга, которая уменьшает двоичное число на 1. Каретка находится над числом.

	$q_1$	$q_2$	$q_3$	$q_4$
0				
1				
<input type="checkbox"/>				

Описание состояний:

$q_1$  –  
 $q_2$  –  
 $q_3$  –  
 $q_4$  –

При каком начальном положении каретки эта программа зацикливается?

Ответ:

8. Составьте программу для машины Тьюринга, которая умножает троичное число на 2. Каретка находится над числом.

	$q_1$	$q_2$	$q_3$
0			
1			
2			
<input type="checkbox"/>			

Описание состояний:

$q_1$  –  
 $q_2$  –  
 $q_3$  –

9. Дана строка, состоящая только из символов «а» и «б». Составьте программу для машины Тьюринга, которая переставляет последний символ в начало строки. Каретка находится над первым символом строки.

	$q_1$	$q_2$	$q_3$	$q_4$
а				
б				
<input type="checkbox"/>				

Описание состояний:

$q_1$  –  
 $q_2$  –  
 $q_3$  –  
 $q_4$  –

10. \*Дана строка, состоящая только из символов «а» и «б». Составьте программу для машины Тьюринга, которая сортирует символы, то есть переставляет все буквы «а» в начало строки. Каретка находится над первым символом строки. Используйте состояния, которые перечислены в таблице.

	$q_1$	$q_2$	$q_3$	$q_4$
а				

б				
□				

*Описание состояний:*

$q_1$  – каретка идёт вправо по цепочке букв «а»  
 $q_2$  – каретка идёт вправо по цепочке букв «б»  
 $q_3$  – каретка идёт влево и ищет конец цепочки букв «б»  
 $q_4$  – замена буквы «б» на букву «а»

11. \*Составьте программу для машины Тьюринга, которая складывает два числа в двоичной системе, разделенные на ленте знаком «+».
12. \*Составьте программы для машины Тьюринга, которые увеличивают и уменьшают на единицу число, записанное в десятичной системе счисления.
13. Составьте программы для машины Тьюринга, которые выполняют сложение и вычитание двух чисел в десятичной системе счисления.

### Практическая работа № 34.

#### Машина Поста

6. Что делает следующая программа для машины Поста?

1	←
2	? 3,4
3	1 1
4	стоп

Ответ:

Как она будет работать при различных начальных состояниях ленты?

Ответ:

7. Напишите программу для машины Поста, которая неприменима (то есть зацикливается) при любом начальном состоянии ленты.

1	
2	
3	

8. Напишите программу для машины Поста, которая увеличивает на 1 число, записанной в унарной системе счисления. Каретка стоит над первой (самой левой) отметкой.

1	
2	
3	
4	

9. Решите задачу 2 при условии, что в начале работы каретка расположена где-то справа от записи числа.

1	
2	
3	
4	
5	

10. Напишите программу для машины Поста, которая уменьшает на 1 число, записанной в унарной системе счисления. Каретка стоит над первой (самой левой) отметкой.

1	
2	
3	
4	
5	

11. На ленте расставлены метки, между которыми могут быть пропуски длиной в одну ячейку. Заполнить все пропуски метками. Каретка стоит над самой левой меткой.

12. \*Напишите программу для машины Поста, которая удваивает число, записанное в единичной системе счисления. Каретка расположена над первой отметкой числа.

13. \*Напишите программу для машины Поста, которая складывает два числа в единичной системе счисления. Каретка расположена над пробелом, разделяющим эти числа на ленте.

14. \*Напишите программу для машины Поста, которая складывает два числа, записанных в унарной системе. Числа расположены на неизвестном расстоянии друг от друга. Каретка находится над левой границей первого (левого) числа.

15. \*\*Напишите программу для машины Поста, которая складывает несколько натуральных чисел. Каждое число кодируется как последовательность

расположенных рядом отметок (в унарной системе счисления). Числа отделены друг от друга пробелами. Каретка находится справа от первого числа.

16. \*\*Написать программу для машины Поста, которая находит единственную метку на ленте, которая расположена неизвестно где. Каретка должна остановиться на метке, все другие (временные) метки должны быть стерты. В начале работы каретка расположена над пустой ячейкой.

## Практическая работа № 35.

### Нормальные алгоритмы Маркова (НАМ)

1. Что делает следующий НАМ, если применить его к символьной цепочке, состоящей из нулей и единиц:

$*0 \rightarrow 0*$

$*1 \rightarrow 1*$

$* \rightarrow =.$

$\rightarrow *$

Ответ:

Как будет работать этот алгоритм при различных начальных состояниях ленты?

Ответ:

14. Напишите НАМ, который «сортирует» цифры двоичного числа так, чтобы сначала стояли все нули, а потом – все единицы.

Ответ:

15. Напишите НАМ, который удаляет последний символ строки, состоящей из цифр 0 и 1. Какую операцию он выполняет, если рассматривать строку как двоичную запись числа?

Ответ:

16. Напишите НАМ, который умножает двоичное число на 2, добавляя 0 в конец записи числа.

Ответ:

17. Напишите НАМ, который переводит число из двоичной системы счисления в единичную (унарную).

Ответ:

18. Дано слово, состоящее из букв «а», «б» и пробелов. Постройте нормальный алгоритм Маркова, который символы «а» переносит влево, символы «б» – вправо, а пробелы оставляет посередине.

Ответ:

19. Дано число в унарной системе счисления (от 1 до 15), сразу после числа стоит точка. Напишите НАМ, который представляет это число в виде суммы степеней двойки (например, для числа 15 нужно получить «8+4+2+1»), и удаляет точку в конце.

Ответ:

20. \*Напишите НАМ, который переводит число из четверичной системы счисления в двоичную. Используйте специальный знак (например, \*), который отделяет обработанную часть числа от необработанной.

Ответ:

21. \*\*Дана последовательность скобок. Напишите НАМ, который проверяет правильность скобочной структуры (парность и вложенность скобок). Например, выражение  $00(0)(00)(0)0$  – правильное, а выражения  $00(0)(00)(0)$  и  $00(0)(00)(0)$  – неправильные.

Если все правильно, лента должна быть пустой, если выражение неверное, на ленте должны остаться «неправильные» скобки.

22. \*\*Напишите НАМ, который увеличивает на единицу число, записанное в троичной системе счисления. Используйте специальные символы (например, # и \*) для обозначения начала числа и того разряда, который нужно увеличивать первым.
23. \*\*Напишите НАМ, который уменьшает на единицу число, записанное в троичной системе счисления. Используйте специальные символы (например, # и \*) для обозначения начала числа и того разряда, который нужно уменьшать первым.
24. \*\*Напишите НАМ, который складывает два числа в двоичной системе счисления. Числа разделены знаком «+», например, из строки «1011+1101» нужно получить «11000». Используйте алгоритм, в котором последовательно первое число увеличивается на 1, а второе – уменьшается на 1, пока второе не станет равно 0.

**Практическая работа № 37.**

**Нормальные алгоритмы Маркова (НАМ)**

12. Целое число  $n$  записано на ленте в унарной системе счисления (как последовательность из  $n$  меток). Напишите программу для машины Поста, которая вычисляет функцию

$$f(n) = \begin{cases} 1, & \text{если } n - \text{чётное} \\ 0, & \text{если } n - \text{нечётное} \end{cases}$$

и таким образом доказывает разрешимость этой задачи. В начальный момент каретка стоит над первой меткой числа.

*Ответ:*

1	
2	
3	
4	
5	
6	
7	
8	

13. Используя любого универсального исполнителя (машину Тьюринга, машину Поста или нормальный алгоритм Маркова), докажите вычислимость функции

$$f(n) = \begin{cases} 1, & \text{если } n - \text{делится на } 3 \\ 0, & \text{если } n - \text{не делится на } 3 \end{cases}$$

Целое число  $n$  записано в унарной системе счисления.

*Ответ:*

--



## Практическая работа № 38.

### Инвариант цикла

1. Определите инвариант цикла для следующего алгоритма двоичного поиска (предполагается, что элементы массива  $A$  отсортированы по неубыванию):

```
L:= 1; R:= n + 1
нц пока L < R - 1
  c:= div(L + R, 2)
  если X < A[c] то
    R:= c
  иначе
    L:= c
все
кц
```

Ответ:

Используя найденный инвариант, определите, какой именно элемент массива будет найден, если в массиве есть несколько элементов, равных  $X$ .

Ответ:

Как нужно изменить инвариант (и цикл!), чтобы найти *первый* элемент, равный  $X$ ?

Ответ:

2. Определите инвариант для следующего цикла.

```
k := 0; b := 1
нц пока k < n
  k := k + 1
  b := b * a
кц
```

Ответ:

Что будет вычислено в переменной  $b$ ?

Ответ:

3. Определите инвариант для следующего цикла.

```
k := n; b := 1
нц пока k > 0
  k := k - 1
  b := b * a
кц
```

Ответ:

Что будет вычислено в переменной  $b$ ?

Ответ:

4. Запишите предусловие и постусловие для алгоритма вычисления суммы всех делителей числа.

Ответ:

5. Запишите предусловие и постусловие для алгоритма проверки числа на простоту.

Ответ:

6. Запишите предусловие и постусловие для алгоритма определения количества слов в символьной строке.

*Ответ:*

7. Запишите предусловие и постусловие для алгоритма двоичного поиска в отсортированном массиве.

*Ответ:*

8. Запишите предусловие и постусловие для алгоритма перестановки элементов массива в обратном порядке.

*Ответ:*

9. Запишите предусловие и постусловие для алгоритма преобразования числа из символьной записи в значение целого типа.

*Ответ:*

10. Предложите другие начальные значения переменных  $b$ ,  $k$  и  $p$  в алгоритме быстрого возведения в степень (см. Пример 4 в §37 учебника). Инвариант цикла должен сохраниться.

*Ответ:*

11. Оцените сложность алгоритма быстрого возведения в степень при  $n = 2^m$ , где  $m$  – натуральное число.

*Ответ:*

## Практическая работа № 38а (дополнительная).

### Сложность вычислений

4. Запишите алгоритм, который находит все делители натурального числа.

Ответ:

Какие элементарные операции в нём можно выделить?

Ответ:

Оцените количество этих операций.

Ответ:

Оцените асимптотическую сложность алгоритма.

Ответ:

5. Запишите алгоритм, который определяет, является ли заданное натуральное число простым.

Ответ:

Какие элементарные операции в нём можно выделить?

Ответ:

Оцените количество этих операций.

Ответ:

Оцените асимптотическую сложность алгоритма.

Ответ:

6. Запишите алгоритм, который находит количество положительных элементов массива.

Ответ:

Какие элементарные операции в нём можно выделить?

Ответ:

Оцените количество этих операций.

Ответ:

Оцените асимптотическую сложность алгоритма.

Ответ:

7. Запишите алгоритм, который находит три минимальных элемента массива.

Ответ:

Какие элементарные операции в нём можно выделить?

Ответ:

Оцените количество этих операций.

Ответ:

Оцените асимптотическую сложность алгоритма.

Ответ:

8. Запишите алгоритм, который находит количество элементов массива, равных минимальному элементу.

Ответ:

Какие элементарные операции в нём можно выделить?

Ответ:

Оцените количество этих операций.

Ответ:

Определите асимптотическую сложность алгоритма.

Ответ:

9. Запишите алгоритм, который находит (и выводит на экран) символы, которые встречаются в символьной строке более одного раза.

Ответ:

Какие элементарные операции в нём можно выделить?

Ответ:

Оцените количество этих операций.

Ответ:

Оцените асимптотическую сложность алгоритма.

Ответ:

10. Алфавит языка племени «тумба-юмба» содержит  $k$  символов. Запишите алгоритм построения всех возможных слов этого языка, имеющих длину  $n$  символов.

Ответ:

Какие элементарные операции в нём можно выделить?

Ответ:

Оцените количество этих операций.

Ответ:

Оцените асимптотическую сложность алгоритма.

Ответ:

## Глава 15. Алгоритмизация и программирование

### Практические работы

#### Практическая работа № 35.

#### Решето Эратосфена

##### Уровень А.

16. Напишите две программы, которые находят все простые числа от 1 до  $N$  двумя разными способами:

- проверкой каждого числа из этого интервала диапазона на простоту;
- используя решето Эратосфена.

##### Уровень В.

17. Сравните число шагов цикла (или время работы) этих программ для разных значений  $N$ .

N	Количество шагов	
	простой перебор	решето Эратосфена
100		
1000		
2000		
3000		
4000		
5000		

Постройте (например, используя электронные таблицы) для каждого варианта зависимость количества шагов от  $N$ .

Ответ:

##### Уровень С.

18. Сделайте выводы о сложности алгоритмов.

Ответ:

## Практическая работа № 36.

### Длинные числа

#### Уровень А.

17. Соберите и выполните программу для вычисления  $100!$ , приведённую в учебнике. Переделайте её, чтобы она вычисляла  $200!$ .  
Дополните программу так, чтобы она определяла, сколько цифр входит в это число. Запишите результат (длину десятичной записи числа  $100!$ ) в поле для ответа.

Ответ:

18. Оформите вывод длинного числа на экран в виде отдельной процедуры. Учтите, что число может быть нулевым. Запишите эту процедуру в поле для ввода ответа.

Ответ:

#### Уровень В.

19. Составьте процедуру для ввода длинных чисел из файла в массив. Запишите её в поле для ввода ответа.

Ответ:

20. Напишите программы для сложения и вычитания длинных чисел. Данные вводятся из текстового файла.

Для справки: <http://comp-science.narod.ru/DL-AR/okulov.htm>

#### Уровень С.

21. Напишите программы для умножения и деления длинных чисел. Данные вводятся из текстового файла. В программе деления нужно получить целое частное и остаток от деления.

Для справки: <http://comp-science.narod.ru/DL-AR/okulov.htm>

22. Выведите на экран точное значение числа  $2^{-200}$  в десятичной системе счисления. Сосчитайте, сколько в нём различных ненулевых цифр:

1	2	3	4	5	6	7	8	9

#### Уровень D.

23. Напишите программу для извлечения квадратного корня из длинного числа. Данные вводятся из текстового файла.

Для справки: <http://spacemath.xyz/algorithm-izvlecheniya-kvadratnogo-kornya/>

## Практическая работа № 37.

### Структуры

#### Уровень А.

1. В файле **marks.csv** записаны сведения о результатах экзаменов в 9-х классах N-ского района. Всего в списке 1000 записей, каждая из которых содержит

- фамилию
- имя
- отметки по четырем предметам (алгебре, русскому языку, физике и истории).

Все элементы в каждой строке разделены запятыми.

Напишите программу, которая читает данные из файла в массив структур (записей) и выводит на экран (или в файл):

- а) средний балл в районе по каждому предмету;
- б) максимальную сумму баллов, полученных учащимися;

#### Уровень В.

в) список учащихся (фамилии и имена), набравших эту максимальную сумму, в алфавитном порядке;

г) количество учащихся, получивших хотя бы одну отметку «2».

#### Уровень С.

2. В файле **files.csv** записаны сведения о файлах. Всего в списке 280 записей, каждая из которых содержит

- имя файла;
- размер файла в Кбайтах;
- тип файла (аудио, видео, изображение, презентация, текстовый, электронная таблица);
- дату создания файла;
- дату последнего изменения файла;
- и уровень доступа.

Все элементы в каждой строке разделены запятыми.

Напишите программу, которая читает данные из файла в массив структур (записей) и выводит на экран (или в файл):

- а) количество файлов каждого типа;
- б) список 10 самых больших файлов, отсортированный по именам файлов (для каждого вывести имя файла и размер);
- в) список презентаций ограниченного доступа, которые изменялись в 2012 году; для каждого файла вывести его имя, тип, доступ и дату изменения; список нужно отсортировать в алфавитном порядке по именам файлов;
- г) список видео размером больше 100 Мбайт, созданных во второй половине 2011 года; для каждого файла вывести его имя, тип, размер и дату создания; список нужно отсортировать по убыванию размеров файлов.

## **Практическая работа № 41а (дополнительная).**

### **Система управления базой данных**

Постройте программу, которая работает с базой данных в виде типизированного файла.

Содержание базы данных можно выбрать на свой вкус. Например, в ней могут храниться

- сведения об аудио- или видеозаписях;
- каталог библиотеки;
- каталог художественных фильмов;
- данные об автомобилях;
- данные о членах спортклуба;
- данные о животных в зоопарке и т.п.

Набор полей структуры определите самостоятельно в соответствии с задачей.

За основу можно принять файл-заготовку **database.py**.

#### **Уровень А.**

Ваша СУБД (система управления базой данных) должна уметь выполнять такие действия:

- а) чтение базы в массив структур при запуске программы;
- б) просмотр записей;
- в) добавление записей;
- г) удаление записей.

#### **Уровень В.**

Предусмотрите обработку ошибочных ситуаций:

- д) для новой записи нет места в массиве (добавляется новая запись в массив, который уже содержит 100 записей; для проверки установите меньшее ограничение, например, 5);
- е) введен неправильный номер удаляемой записи.

#### **Уровень С.**

ж) Добавьте возможность сортировки записей по двум полям (на выбор).

#### **Уровень D (Дополнительные задания):**

- з) Предусмотрите какой-нибудь вариант шифрования данных, так чтобы их нельзя было просмотреть в текстовом редакторе.
- и) Добавьте к возможностям вашей СУБД экспорт данных в тестовый файл в формате XML.



## Практическая работа № 38.

### Словари

#### Уровень А.

35. В каждой строчке файла **dictionary.txt** записана пара слов: слово на русском языке и его перевод на английский. Нужно написать программу, которая загружает в память словарь и позволяет пользователю переводить слова с русского на английский.
36. Если слова нет в словаре, программа должна выводить сообщение «Не знаю».
37. Программа должна заканчивать работу, если введена пустая строка.

#### Уровень В.

38. Добавить в программу для работы со словарём (см. Уровень А) следующую возможность: если пользователь вводит слово, которого нет в словаре, программа запрашивает его перевод и добавляет новую пару в словарь (обновляя файл на диске).
39. На вход программы подаются произвольные алфавитно-цифровые символы. Ввод этих символов заканчивается точкой. Требуется написать программу, которая будет печатать последовательность строчных английских букв ('a' 'b'... 'z') из входной последовательности и частот их повторения. Печать должна происходить в алфавитном порядке. Например, пусть на вход подаются следующие символы:

**fhb5kbfыshfm.**

В этом случае программа должна вывести

**b2**

**f3**

**h2**

**kl**

**ml**

**s1**

#### Уровень С.

40. В файле **olymp.txt** записаны сведения об участниках олимпиады. В первой строке записано количество учащихся N, каждая из следующих N строк имеет формат:

**<Фамилия> <Инициалы> <номер школы>**

где **<Фамилия>** – строка, состоящая не более чем из 20 символов, **<Инициалы>** – строка, состоящая из 4-х символов (буква, точка, буква, точка), **<номер школы>** – не более чем двузначный номер. **<Фамилия>** и **<Инициалы>**, а также **<Инициалы>** и **<номер школы>** разделены пробелам. Пример входной строки:

**Иванов П.С. 57**

Требуется написать программу, которая будет выводить на экран в порядке убывания номера школ, которые прислали на олимпиаду меньше всего участников (таких школ может быть несколько).

41. В файле **towns.txt** записаны данные о странах и городах. В первой строке записаны натуральные числа S и G. Следующие S строк имеют такой формат:

**<страна> <город1> <город2> <город3>...**

Количество городов для каждой страны не превышает 1000. В следующих G строках записаны названия городов, по одному в каждой строке. Для каждого из них программа должна вывести название страны, в которой находится этот город, или знак ?, если страна неизвестна. Например, для файла, содержащего такие данные:

**2 3**

**Россия Москва Кемерово Якутск**

**Финляндия Хельсинки Турку**

**Хельсинки**

**Магнитогорск**

**Кемерово**

программа должна вывести

**Хельсинки → Финляндия**  
**Магнитогорск -> ?**  
**Кемерово -> Россия**

**Практическая работа № 39.**  
**Алфавитно-частотный словарь**

**Уровень А.**

16. Постройте программу, которая составляет алфавитно-частотный словарь для заданного файла со списком слов. Программа должна выводить результат работы в другой файл.

**Уровень В.**

17. Выведите все найденные слова в файл в порядке убывания частоты, то есть в начале списка должны стоять слова, которые встречаются в файле чаще всех. Слова, встречающиеся одинаковое количество раз, должны быть расставлены по алфавиту.

**Уровень С.**

18. Постройте программу, которая составляет алфавитно-частотный словарь для заданного файла с текстом. Не забудьте, что строки текста нужно разделить на слова и удалить все знаки препинания.

**Уровень D (проект).**

19. Постройте алфавитно-частотные словари для текстов разных авторов. Сравните частоты использования предлогов и местоимений.

## **Практическая работа № 40.**

### **Использование стека**

#### **Уровень А.**

2. Напишите программу, которая вычисляет с помощью стека значение арифметического выражения, записанного в постфиксной форме. Выражение вводится с клавиатуры в виде символьной строки.

#### **Уровень В.**

3. Напишите программу, которая вычисляет значение арифметического выражения, записанного в префиксной форме. Выражение вводится с клавиатуры в виде символьной строки.

#### **Уровень С.**

4. Сделайте в программе для задачи А (вычисление выражения в постфиксной форме) обработку ошибок: при неправильном выражении программа должна выдавать сообщение об ошибке. Поработайте в парах: протестируйте программу напарника и попробуйте ввести такое выражение, на котором она аварийно завершится.

#### **Уровень D.**

5. Найдите в литературе или в Интернете алгоритм перевода арифметического выражения из инфиксной формы в постфиксную, и напишите программу, которая решает эту задачу.

## **Практическая работа № 41.**

### **Скобочные выражения**

#### **Уровень А.**

4. Напишите программу, которая проверяет правильность скобочного выражения с четырьмя видами скобок: (), [], {} и <>.

#### **Уровень В.**

5. Доработайте программу так, чтобы в случае ошибки она выводила номер первого ошибочного символа

#### **Уровень С.**

6. Разработайте программу, которая строит все правильные скобочные выражения заданной длины с тремя типами скобок.

## Практическая работа № 42.

### Очереди

#### Уровень А.

1. Каждую секунду на вход программы поступает результат измерения – целое число. Данных настолько много, что хранить их все в памяти невозможно. Нужно написать программу, которая находит наибольшую сумму двух результатов измерений, между которыми прошло **ровно 5** секунд.

**Исходные данные** записаны в файл **queue.dat**: в первой строке – количество измерений  $N > 5$ , в каждой из следующих  $N$  строк – одно целое число, не превышающее по модулю 1000.

#### Уровень В.

2. Каждую секунду на вход программы поступает результат измерения – целое число. Данных настолько много, что хранить их все в памяти невозможно. Нужно написать программу, которая находит наибольшую сумму двух результатов измерений, между которыми прошло **не более 5** секунд.

**Исходные данные** записаны в файл **queue.dat**: в первой строке – количество измерений  $N > 5$ , в каждой из следующих  $N$  строк – одно целое число, не превышающее по модулю 1000.

#### Уровень С.

3. Каждую секунду на вход программы поступает результат измерения – целое число. Данных настолько много, что хранить их все в памяти невозможно. Нужно написать программу, которая находит наибольшую сумму двух результатов измерений, между которыми прошло **более 5** секунд.

**Исходные данные** записаны в файл **queue.dat**: в первой строке – количество измерений  $N > 5$ , в каждой из следующих  $N$  строк – одно целое число, не превышающее по модулю 1000.

## **Практическая работа № 43.**

### **Заливка области**

#### **Уровень А.**

28. Напишите программу, которая выполняет заливку одноцветной области заданным цветом. Матрица, содержащая цвета пикселей, вводится из файла. Затем с клавиатуры вводятся координаты точки заливки и цвет заливки. На экран нужно вывести матрицу, которая получилась после заливки. Все операции с очередью вынесите в отдельный модуль.

#### **Уровень В.**

29. Перепишите предыдущую программу – используйте статический массив для организации очереди. Считайте, что в очереди может быть не более 100 элементов. Предусмотрите обработку ошибки «очередь переполнена».

#### **Уровень С.**

30. Напишите оптимизированное решение задачи о заливке области, в котором

- а) в очередь добавляются только те точки, которые имеют тот же цвет, что и начальная точка;
- б) точки, уже добавленные в очередь, не добавляются повторно;

В чём преимущества и недостатки такого алгоритма?

#### **Уровень D. Проект.**

31. Найдите в литературе или в Интернете описание формата BMP. Напишите программу, которая выполняет заливку одноцветной области рисунка в формате заданным цветом и сохраняет получившийся файл на диске. Координаты точки заливки и цвет заливки вводятся с клавиатуры. Все операции с очередью и рисунками в формате BMP вынесите в отдельные модули.

## **Практическая работа № 44.**

### **Обход дерева**

#### **Уровень А.**

1. Напишите программу, которая выполняет обходы дерева «корень – левое – правое», «левое – правое – корень», «левое – корень – правое» с помощью рекурсии.

#### **Уровень В.**

2. Напишите программу, которая выполняет обходы дерева в глубину и в ширину **без использования рекурсии** (используйте стек и очередь).

#### **Уровень С.**

3. Напишите программу, которая вводит дерево из файла (придумайте формат хранения данных) и выполняет обход дерева в ширину.

#### **Уровень D.**

4. Напишите программу, которая вводит дерево из файла (придумайте формат хранения данных) и выполняет обход дерева в ширину, **не используя очередь**.



## **Практическая работа № 45.**

### **Вычисление арифметических выражений**

#### **Уровень А.**

1. Напишите программу, которая вводит и вычисляет арифметическое выражение без скобок. Все операции с деревом вынесите в отдельный модуль.

#### **Уровень В.**

2. Добавьте в предыдущую программу процедуры обхода построенного дерева так, чтобы получить префиксную и постфиксную запись введенного выражения.
3. Добавьте в предыдущую программу процедуру обхода дерева в ширину.

#### **Уровень С.**

4. Усовершенствуйте программу (см. задачу 1), чтобы она могла вычислять выражения со скобками.

#### **Уровень D.**

5. Включите в вашу программу обработку некоторых ошибок (например, два знака операций подряд). Поработайте в парах: обменяйтесь программами с соседом и попробуйте найти выражение, при котором его программа завершится аварийно (не выдаст сообщение об ошибке).

## **Практическая работа № 46.**

### **Хранение двоичного дерева в массиве**

1. Измените программу для вычисления арифметического выражения (см. предыдущую практическую работу), так чтобы дерево хранилось в виде массива. Все операции с деревом вынесите в отдельный модуль.

**Практическая работа № 47.**

**Алгоритм Прима-Крускала**

**Уровень А.**

1. Напишите программу, в которой задаётся весовая матрицу графа размером  $6 \times 6$ , а затем строится минимальное остовное дерево для этого графа.

**Уровень В.**

2. Напишите программу, которая вводит из файла весовую матрицу графа и строит для него минимальное остовное дерево.

**Уровень С.**

3. Оцените асимптотическую сложность алгоритма Крускала.

*Ответ:*

--

## Практическая работа № 48.

### Алгоритм Дейкстры

#### Уровень А.

1. Напишите программу, в которой задаётся весовая матрица графа размером  $6 \times 6$ , затем с клавиатуры вводятся номера начальной и конечной вершин и определяется длина оптимального маршрута между ними.

#### Уровень В.

2. Напишите программу, которая вводит из файла **dijkstra.dat** весовую матрицу графа, затем вводит с клавиатуры номера начальной и конечной вершин и определяет длину оптимального маршрута между этими вершинами.

#### Уровень С.

3. Напишите программу, которая вводит из файла **dijkstra.dat** весовую матрицу графа, затем вводит с клавиатуры номера начальной и конечной вершин и определяет оптимальный маршрут между этими вершинами (длину и последовательность переходов между вершинами).
4. Оцените асимптотическую сложность алгоритма Дейкстры.

Ответ:

--

**Практическая работа № 49.**  
**Алгоритм Флойда-Уоршелла**

**Уровень А.**

1. Напишите программу, которая вводит из файла **floyd.dat** весовую матрицу графа и определяет длины всех оптимальных маршрутов с помощью алгоритма Флойда-Уоршелла. Полученную матрицу длин оптимальных путей выведите на экран.

**Уровень В.**

2. Напишите программу, которая вводит из файла **floyd.dat** весовую матрицу графа и определяет длины всех оптимальных маршрутов с помощью алгоритма Флойда-Уоршелла. Полученную матрицу длин оптимальных путей выведите на экран. После этого программа должна запросить номера начальной и конечной вершин графа и вывести на экран оптимальный маршрут (последовательность вершин) и его длину.

**Уровень С.**

3. Напишите программу, которая решает следующую задачу о размещении магазина. Имеется  $N$  домов, в каждом из которых живет  $p_i$  жителей ( $i = 1, \dots, N$ ). В одном из домов надо разместить магазин, так чтобы общее расстояние, проходимое всеми жителями по дороге в магазин, было минимальным. В каком доме нужно разместить магазин?

Для определения кратчайших путей используйте алгоритм Флойда-Уоршелла. Весовую матрицу графа и число жителей вводите из файла **shop.dat**.

[https://e-maxx.ru/algo/floyd\\_warshall\\_algorithm](https://e-maxx.ru/algo/floyd_warshall_algorithm)

## Практическая работа № 50.

### Числа Фибоначчи

#### Уровень А.

1. Напишите программу для вычисления N-ого числа Фибоначчи, использующую рекурсию. Проверьте ее работу при различных значениях N и постройте (например, с помощью электронных таблиц) график зависимости времени счёта от N.

#### Уровень В.

Оцените асимптотическую сложность этого алгоритма.

Ответ:

#### Уровень С.

2. Напишите программу для вычисления N-ого числа Фибоначчи, использующую динамическое программирование. Проверьте ее работу при различных значениях N и постройте (например, с помощью электронных таблиц) график зависимости времени счёта от N. Сравните эти данные с рекурсивным вариантом программы:

N	Время счёта	
	рекурсия	динамическое программирование
1		
10		
20		
25		
28		
30		

Оцените асимптотическую сложность алгоритма, использующего динамическое программирование.

Ответ:

## **Практическая работа № 51.**

### **Задача о куче**

#### **Уровень А.**

1. Напишите программу, которая решает задачу о куче (см. учебник). Данные вводятся из текстового файла в следующем формате:

- в первой строке записано количество камней  $N$  и требуемый вес кучи  $W$ ;
- во второй строке перечислены веса камней  $p_i$  ( $i = 1, \dots, N$ ).

Считайте, что  $N \leq 100$  и  $W \leq 100$ .

Результаты работы программы выводятся в текстовый файл в следующем формате:

- в первой строке выводится количество используемых камней и вес полученной кучи;
- во второй строке выводятся веса выбранных камней.

#### **Уровень В.**

2. Измените программу так, чтобы память под используемые массивы выделялась динамически. Допущение о том, что  $N \leq 100$  и  $W \leq 100$  не используйте.

#### **Уровень С.**

3. Напишите программу, которая решает задачу о ранце (см. задачу 3 после § 45).

## Практическая работа № 52.

### Количество программ

#### Уровень А.

1. У исполнителя *Калькулятор* есть три команды, которым присвоены номера:

1. прибавь 1
2. умножь на 2
3. умножь на 3

Напишите программу, которая определяет количество программ для преобразования числа  $A$  в число  $B$  с помощью этого исполнителя. Числа  $A$  и  $B$  вводятся с клавиатуры.

**Пример:**  $A = 3, B = 135$ .

Ответ: 20858.

#### Уровень В.

2. У исполнителя *Калькулятор* есть две команды, которым присвоены номера:

1. прибавь 1
2. увеличь вторую с конца цифру на 1

Первая из них увеличивает число на экране на 1, вторая – увеличивает на 1 число десятков. Если перед выполнением команды 2 вторая с конца цифра равна 9, она не изменяется.

Напишите программу, которая определяет количество программ для преобразования числа  $A$  в число  $B$  с помощью этого исполнителя. Числа  $A$  и  $B$  вводятся с клавиатуры.

**Пример:**  $A = 3, B = 65$ .

Ответ: 28711.

#### Уровень С.

3. У исполнителя *Калькулятор* есть три команды, которым присвоены номера:

1. прибавь 1
2. умножь на 2
3. умножь на 3

Напишите программу, определяющую количество программ, которые преобразуют исходное число  $A$  в число  $B$ , и при этом траектория вычислений содержит число  $M$  и не содержит чисел  $X$  и  $Y$ ? Все неизвестные значения вводятся с клавиатуры.

**Пример:**  $A = 3, B = 135, C = 25, X = 13, Y = 39$ .

Ответ: 4130.



## Практическая работа № 53.

### Размен монет

#### Уровень А.

1. Напишите программу, которая решает задачу о размене монет (см. учебник). Данные вводятся из текстового файла в следующем формате:

- в первой строке записано количество типов монет  $N$  и требуемая сумма  $W$ ;
- во второй строке перечислены достоинства монет  $p_i$  ( $i = 1, \dots, N$ ).

Считайте, что  $N \leq 10$  и  $W \leq 100$  и в наборе монет есть монета достоинством  $p_i = 1$ . Программа должна вывести на экран количество вариантов сдачи.

#### Уровень В.

2. Измените программу так, чтобы она работала и в том случае, если в наборе монет нет монеты достоинством  $p_i = 1$ . Допущение о том, что  $N \leq 10$  и  $W \leq 100$  не используйте.

#### Уровень С.

3. Добавьте в программу второй способ вычисления количества вариантов – рекурсивный перебор. Добейтесь совпадения результатов, полученных двумя методами.

## Глава 16. Объектно-ориентированное программирование

### Практические работы и проекты

#### Практическая работа № 54.

##### Движение на дороге (проект)

19. Изучите материал § 43 учебника (*Создание объектов в программе*).
20. Постройте программу, которая описана в тексте параграфа. Используя выбранную среду программирования, изобразите движение машин по дороге в текстовом или графическом режиме. Для этого дополните класс **TCar**.
21. \*Добавьте в модель светофор, который переключается автоматически по программе (например, 5 с горит красный свет, затем 1 с – жёлтый, потом 5 с – зеленый и т.д.). Измените классы так, чтобы машина запрашивала у объекта *Дорога* местоположение ближайшего светофора, а затем обращалась к светофору для того, чтобы узнать, какой сигнал горит. Машины должны останавливаться у светофора с запрещающим сигналом.

## **Практическая работа № 55.**

### **Скрытие внутреннего устройства объектов**

1. Измените построенную ранее программу моделирования движения автомашин по дороге так, чтобы все поля у объектов были закрытыми. Используйте свойства для доступа к данным.

## Практическая работа № 56.

### Классы логических элементов. Проект

1. Используя материалы § 45 учебника, постройте модуль, описывающий иерархию классов логических элементов. Для проверки его работы напишите программу, которая строит таблицу истинности последовательного соединения логических элементов «И» и «НЕ».
2. Добавьте в иерархию классов элементы «исключающее ИЛИ», «И-НЕ» и «ИЛИ-НЕ». Вынесите всю построенную систему классов логических элементов в отдельный модуль **LogElement**.
3. «Соберите» в программе RS-триггер из двух логических элементов «ИЛИ-НЕ», постройте его таблицу истинности (обратите внимание на вариант, когда оба входа нулевые).
4. \*Постройте новый класс **TNOrTrigger**, моделирующий RS-триггер на двух логических элементах «ИЛИ-НЕ». От какого класса лучше наследовать этот класс? Почему?

Ответ:

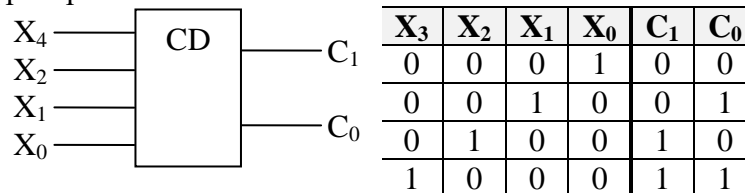
Как лучше включить в этот класс два элемента «ИЛИ-НЕ»? Будете ли вы использовать для этого наследование?

Ответ:

5. \*Выполните задания 3-4 для RS-триггера на двух логических элементах «И-НЕ». Определите, при каких входах состояние сохраняется, а какие входные сигналы запрещены.

Ответ:

6. \*Постройте логический элемент «шифратор», который выдает на выходы двоичное представление номера входа, на который подали сигнал. Обозначение и таблица истинности шифратора «4 в 2» показаны ниже:

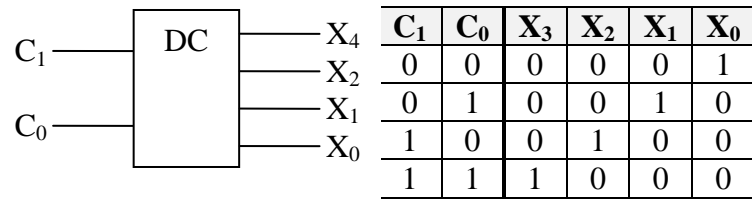


Все комбинации входов, которые не перечислены в таблице истинности, считаются запрещенными. Выполните следующие задания:

- а) по таблице истинности составьте логическую схему шифратора; постарайтесь использовать минимальное число логических элементов;
- б) добавьте в программу новый класс – модель шифратора (англ. *coder*), в которой будут использованы уже имеющиеся классы логических элементов; Будете ли вы использовать наследование? Почему?

Ответ:

- в) напишите программу, которая строит таблицу истинности шифратора, используя построенную модель.
7. \*Постройте модель логического элемента «дешифратор» (англ. *decoder*), который выполняет обратную операцию. Напишите программу, которая строит его таблицу истинности.



$C_1$	$C_0$	$X_3$	$X_2$	$X_1$	$X_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

## Практическая работа № 57.

### Работа с формой

24. Постройте программу с графической формой, описание которой приведено в § 47 учебника.
25. Определите обработчики событий
  - OnClick** (щелчок мыши по форме) – уменьшить ширину и высоту формы на 10 пикселей, если соответствующий размер больше, чем 100 пикселей;
  - OnDbClick** (двойной щелчок на форме) – установить размеры формы 400 на 100 пикселей;
  - OnResize** (изменение размеров формы) – вывести размеры формы в заголовке окна (изменяйте свойство **title**).
26. \*Выясните (см. текст учебника), как ограничить пределы изменения размеров формы (то есть установить минимальную и максимальную ширину и высоту). Сделайте так, чтобы размеры формы «замораживались» при нажатии на клавишу «F» и снова «размораживались» при нажатии на клавишу «U».

## **Практическая работа № 58.**

### **Использование компонентов**

2. Постройте программу для просмотра рисунков, описание которой приведено в § 48 учебника.
3. \*Дополните программу так, чтобы рисунок, размер которого меньше, чем отведенное для его поле, центрировался в рамках этого поля.

*Подсказка:* нужно «подложить» под компонент *TImage* панель (компонент *TPanel*), у компонента *TImage* отключить принудительное выравнивание, и менять положение и размеры этого компонента при изменении размеров формы (можно использовать обработчик события *OnResize* панели, на которой лежит *TImage*).

## Практическая работа № 59.

### Ввод данных

14. Постройте программу «RGB-кодирование», описание которой приведено в § 48 учебника. Добавьте в эту программу защиту от ввода слишком больших чисел (больших, чем 255).
15. \*Добавьте в программу возможность обратного перевода цифровых кодов: введенное значение шестнадцатеричного кода должно преобразовываться в отдельные значения RGB, например, из «#FF00CC» нужно получить  $R = 255$ ,  $G = 0$  и  $B = 204$ .

Возможно, вам будет полезно логическое свойство **Focused**, которое есть у компонентов, принимающих ввод с клавиатуры. Оно равно **True**, если компонент имеет «фокус ввода», то есть пользователь вводит данные в это поле. Например, если **Edit1** – это компонент класса **TEdit**, можно написать

**if Edit1.Focused then**

**{ компонент принимает ввод с клавиатуры }**

**else**

**{ компонент НЕ принимает ввод с клавиатуры }**

16. Разработайте программу для перевода морских миль в километры (1 миля = 1852 м).
17. Разработайте программу для решения системы двух линейных уравнений. Обратите внимание на обработку ошибок при вычислениях.
18. Разработайте программу для перевода суммы в рублях в другие валюты.
19. Разработайте программу для перевода чисел и десятичной системы в двоичную, восьмеричную и шестнадцатеричную.
20. Разработайте программу для вычисления информационного объема рисунка по его размерам и количеству цветов в палитре.
21. Разработайте программу для вычисления информационного объема звукового файла при известных длительности звука, частоте дискретизации и глубине кодирования (числу бит на отсчёт).



## **Практическая работа № 60.**

### **Совершенствование компонентов**

11. Постройте программу с компонентом **TIntEdit**, описание которой приведено в § 49 учебника.
12. \*Найдите в Интернете информацию по установке нового компонента в палитру компонентов среды *Lazarus* (или *Delphi*) и выполните такую установку. Измените программу так, чтобы она использовала компонент **TIntEdit** из палитры компонентов. Учтите, что при установке компонента в *Lazarus* потребуется пересборка всей среды (в *Delphi* – нет).

## Практическая работа № 61.

### Калькулятор. Проект

1. Постройте программу с компонентами, описание которой приведено в § 50 учебника.
2. Измените программу так, чтобы она вычисляла выражения с вещественными числами (для перевода вещественных чисел из символьного вида в числовой используйте функцию **StrToFloat**).
3. Добавьте в программу обработку ошибок. Подумайте, какие ошибки может сделать пользователь. Какие ошибки могут возникнуть при вычислениях? Как их обработать?

\*Сделайте обработку ошибок, используя исключения (*Exception*).

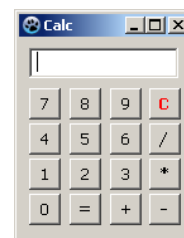
4. \*Измените программу так, чтобы она вычисляла выражения со скобками. *Подсказка:* нужно искать последнюю операцию с самым низким приоритетом, стоящую *вне* скобок.

Предусмотрите сообщение об ошибке «Непарные скобки в выражении». Для обработки этой ошибки используйте исключение – наследник класса **Exception**:

```
class EBracketException(Exception):
```

```
pass
```

5. Постройте программу «Калькулятор» для выполнения вычислений с целыми числами (см. рисунок).



## Глава 17. Компьютерная графика и анимация

### Практические работы

#### Практическая работа № 62.

#### Коррекция изображений

В этой работе мы научимся менять размеры изображения и разрешение, которое используется при печати, а также поворачивать и кадрировать (обрезать) отсканированное изображение.

22. Откройте файл **flower.jpg**.

23. Измените размеры рисунка (*Изображение – Размеры изображения*) так, чтобы его ширина стала равна 200 пикселей. Используйте кубическую интерполяцию. Сохраните результат под именем **flower-cubic.jpg**.

24. Определите разрешение при печати, а также размеры рисунка в пикселях и в миллиметрах (*Изображение – Размеры изображения*), запишите их в тетрадь.

25. Установите разрешение при печати 300 пикселей на дюйм (*Изображение – Разрешение при печати*). Снова определите размеры рисунка в пикселях и миллиметрах, сравните с данными, которые вы раньше записали в тетрадь.

Объясните результаты:

26. Снова откройте исходный файл **flower.jpg**.


27. Измените размеры рисунка так, чтобы его ширина стала равна 200 пикселей, но теперь в списке *Интерполяция* установите вариант *Никакая*. Сохраните результат под именем **flower-none.jpg**.


28. Сравните полученные два файла (используйте увеличение) и определите, чем они отличаются. Почему?

Ответ:

29. Откройте файл **scan-photo.jpg** из каталога.

30. Поверните отсканированное изображение на 90 градусов (*Изображение – Преобразования*).

31. С помощью инструмента  *Вращение* поверните рисунок так, чтобы стороны отсканированной фотографии стали параллельны сторонам изображения.


32. Включите инструмент  *Кадрирование* и выделите прямоугольную область, оставив только нужную часть рисунка. Обрежьте поля, нажав на клавишу *Enter*.

33. Сохраните полученную фотографию под именем **photo.jpg**.

#### Коррекция фотографий

#### Исправление перспективы

*Искажение перспективы при съемке фотоаппаратами проявляется, например, в том, что вертикальные линии (например, стены домов) на фотографии оказываются наклонными. Эти искажения легко исправить в графическом редакторе.*

1. Откройте файл **column.jpg**. Используя инструмент  *Перспектива*, исправьте искажения на фотографии – сделайте стены домов вертикальными.

#### Коррекция уровней

*Если фотография слишком светлая, слишком темная или малоконтрастная, часто ее качество можно улучшить с помощью коррекции уровней. Однако этот прием не всегда хорошо работает при редактировании пейзажей.*

2. Откройте файл **morning.jpg**. Используя меню *Окна – Прикрепляющиеся диалоги – Гистограмма*, посмотрите, как можно улучшить изображение.

Ответ:

3. Используя окно *Цвет – Уровни*, исправьте изображение и сохраните его.

4. Сделайте то же самое с фотографиями **evening.jpg** и **sun.jpg**.

5. Откройте файл **lake.jpg** и попробуйте откорректировать его тем же способом. Получилось ли у вас? Сделайте выводы.

Ответ:

### **Коррекция цвета**

*Некоторые фотоаппараты достаточно сильно искажают естественные цвета, особенно при слабом освещении. Часто этот недостаток фотоснимков можно исправить с помощью коррекции цвета.*



6. Откройте файл **color.jpg**. Используя меню *Цвет – Цветовой баланс*, выполните коррекцию изображения. Сохраните его на диске.

### **Зачетная работа**

7. Откройте файл **mountain.jpg**. Используя все известные вам приемы, выполните коррекцию изображения. Сохраните его на диске.
8. С помощью меню *Цвет – Обесцвечивание*, получите черно-белый вариант фотографии и сохраните его с именем **mountain-gray.jpg**.

### **Ретушь**

*Ретушь часто приходится применять для восстановления отсканированных старых фотографий, которые имеют дефекты.*


9. Откройте файл **boy.jpg**. Примените коррекцию уровней.
10. Устраните дефекты изображения, используя инструменты ретуши:  *Штамп*,  *Лечащая кисть*. Сохраните результат.

## Практическая работа № 63.


### Работа с областями

#### Сглаживание и растушевка

*Сначала нужно разобраться в том, что такое сглаживание и растушевка при выделении областей.*

4. Создайте новый документ размером 300 на 100 пикселей с белым фоном.
5. Установите черный цвет фона (это можно сделать, нажав клавишу X – поменять основной и фоновый цвета).
6. Включите инструмент  *Эллипс*, в свойствах инструмента выключите режимы *Сглаживание* и *растушевать края*.
7. В левой части выделите круг размером примерно 80 на 80 пикселей и залейте его фоновым цветом, нажав клавишу *Delete*.
8. Включите режим *Сглаживание* и постройте справа еще один круг примерно такого же размера. Залейте его фоновым цветом. Увеличьте рисунок и сравните границы первого и второго кругов.
9. Включите режим *Растушевать края* и установите радиус растушевки около 20 пикселей. Постройте справа еще один круг и залейте его фоновым цветом. Сравните границы всех трёх полученных кругов.


*Теперь используем выделение с размытой границей для создания открытки.*

10. Откройте файл **baikal.jpg**.
11. Выберите инструмент  *Эллипс*, включите режим растушевки краев с радиусом 100 пикселей. Выделите центральную часть фотографии.
12. Сделайте инверсию выделения, нажав клавиши *Ctrl+I* (при этом выделенные области становятся невыделенными и наоборот).
13. Установите белый фоновый цвет и залейте им выделенную область. Сохраните получившийся рисунок.



#### Кадрирование нескольких изображений

*Часто несколько небольших фотографий для ускорения сканируют сразу, так что они оказываются на одном изображении (в одном файле). Мы научимся вырезать отдельные рисунки и сохранять их в файлах.*

14. Откройте файл **scan-multi.jpg**.
15. Включите инструмент  *Лассо*. Выделите одну из фотографий, щелкая в её углах и затем в первой точке, чтобы замкнуть контур выделения.
16. Скопируйте выделенную область в буфер обмена (*ПКМ – Правка – Копировать*).
17. Создайте новое изображение из рисунка, который содержится в буфере обмена (*ПКМ – Правка – Вставить как – Новое изображение* или *Shift+Ctrl+V*).
18. Поверните рисунок, если нужно, и обрежьте поля. Сохраните его в виде отдельного файла с именем **photo-1.jpg**.
19. Сохраните таким же способом остальные фотографии.


## Практическая работа (дополнительная)

### Быстрая маска. Фильтры

#### Быстрая маска

Сделаем так, чтобы одна область рисунка оказалась подсвеченной, а остальные были затемнены.



1. Откройте файл **gimp-tools.png**.
2. Переключитесь в режим быстрой маски (меню *Выделение – Переключить быструю маску*).
3. Установите серый основной цвет и залейте весь рисунок (*Правка – Залить цветом переднего плана*). Таким образом, все пиксели оказываются наполовину выделенными.
4. Выберите инструмент  *Прямоугольник* и установите радиус растушевки около 15 пикселей. Выделите любую кнопку. Установите черный основной цвет и залейте им выделенную область. При этом мы отменяем выделение этой кнопки, размывая границу.
5. Отключите режим быстрой маски (меню *Выделение – Переключить быструю маску*).
6. Установите черный основной цвет и залейте им выделенную область.

#### Умные ножницы + быстрая маска

Нужно отделить объект (бабочку) от фона. Это может понадобиться, например, для того, чтобы заменить фон фотографии.

7. Откройте файл **butterfly.jpg**. В этом задании мы отделим бабочку от фона, используя инструмент  *Умные ножницы* и режим быстрой маски.
8. Включите инструмент  *Умные ножницы* и выделите бабочку, щелкая в опорных точках вдоль контура. Замкните контур, щелкнув на первой опорной точке, и превратите его в выделение, щелкнув внутри контура.
9. Переключитесь в режим быстрой маски (меню *Выделение – Переключить быструю маску*).
10. Включите инструмент  *Кисть* и выберите кисть с размытыми границами, например,  *Circle Fuzzy* (19). Отрегулируйте размер кисти с помощью движка *Масштаб* в свойствах инструмента.
11. Подправьте маску, используя черный и белый цвета. Рисование белым цветом выделяет область, черным – отменяет выделение. Для переключения между черным и белым цветами можно использовать клавишу *X*.
12. Аккуратно выделите усики бабочки. Для этого нужно уменьшить размер кисти и увеличить масштаб рисунка.
13. Отключите режим быстрой маски (меню *Выделение – Переключить быструю маску*).
14. Сделайте инверсию выделения, чтобы выделить все, кроме бабочки (клавиши *Ctrl+I*).
15. Выберите серый основной цвет и черный цвет фона, затем залейте выделенную область с помощью инструмента  *Градиент*, который дает плавный переход от одного цвета к другому.
16. Обрежьте лишние части изображения и сохраните окончательный результат под именем **butterfly-ready.jpg**.



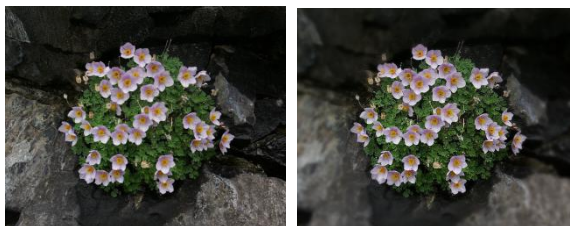
#### Убираем «красные глаза»

Уберем «эффект красных глаз», который возникает при фотосъемке со вспышкой.

17. Откройте файл **red-eyes.jpg**.
18. Выделите область глаз и, используя окно *Цвет – Тонирование*, исправьте «красные глаза» на фотографии.

## Повышение резкости и размывание


Чтобы фотография стала более выразительной, привлечем внимание к цветам и сделаем немного размытым задний план.



было

стало






Для этого применим два фильтра: *Повышение резкости* и *Гауссово размывание*.

19. Откройте файл **flowers.jpg**.
20. Выберите инструмент  *Эллипс*, включите режим растушевки краев с радиусом 100 пикселей. Выделите цветы на фотографии.
21. Примените фильтр *Улучшение – Повышение резкости*, его настройки подберите экспериментально.
22. Сделайте инверсию выделения, чтобы выделить все, кроме цветов (клавиши *Ctrl+I*).
23. Примените фильтр *Размывание – Гауссово размывание* с радиусом 8-10 пикселей. Это позволит размыть задний план, чтобы не отвлекать внимание от цветов.
24. Сохраните результат с именем **flowers-ready.jpg**.

## Практическая работа № 64. Многослойные изображения Рисунок со слоями

Построим многослойный рисунок, в котором человек стоит на фоне озера Байкал. Фоновый рисунок, фигура человека, его футболка, кепка и шорты – это отдельные слои. Мы научимся добавлять надписи (текстовые слои) и в конце работы поместим человека за скалу. При этом никаких необратимых изменений (типа удаления частей рисунка) мы делать не будем.



1. Откройте файл **layers.xcf**.
2. Измените порядок слоев так, чтобы слой *Тело* оказался сверху.
3. С помощью инструмента  *Перемещение* установите человека там, где вам нравится.
4. Используя команду *Файл – Открыть как слои*, откройте файлы **shorts.png** и **t-shirt.png**. Переместите слои так, чтобы человек оказался одетым. Сделайте так, чтобы новые слои назывались *Шорты* и *Футболка*.
5. Откройте файл **cap.png**. Из окна *Слои* перетащите слой *Фон* в окно файла **layers.xcf** назовите добавленный слой *Кепка*.
6. Выделите слой *Футболка* и измените её цвет на красный, применив тонирование (*Цвет – Тонирование*).
7. Включите инструмент  *Текст*, на панели свойств инструмента выберите на свой вкус название и размер шрифта, установите белый цвет. Добавьте новый текстовый слой так, чтобы на футболке появился номер вашей школы.
8. Свяжите все слои, из которых состоит фигура человека. Для этого нужно щелкнуть мышкой справа от значка  в строке каждого из этих слоев в окне *Слои*. Попробуйте перемещать человека.
9. Теперь мы сделаем так, что человек будет выглядывать из-за скалы. Отключите все слои, из которых состоит человек, щелкнув по значкам .
10. Используя  *Умные ножницы* и режим быстрой маски, выделите скалу в нижней части рисунка. Скопируйте ее в буфер обмена (клавиши *Ctrl+C*) и вставьте выделение как новый слой (меню *Правка – Вставить как – Новый слой*). Переместите вставленное на свое место. Назовите новый слой *Скала* и сделайте его самым верхним.
11. Включите все слои, из которых состоит человек. Переместите человека так, как показано на рисунке. Сохраните работу под именем **layers-ready.xcf**.

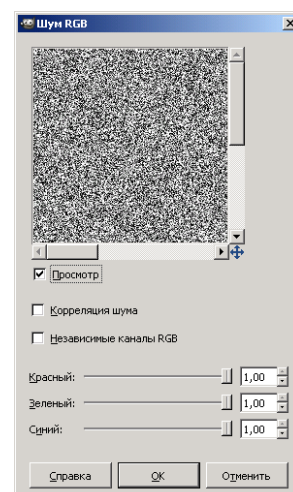


### Рельефный фон


Построим рельефный фон, который можно использовать, например, для оформления веб-страниц. Применим фильтр из группы «Шум» и слои, накладывая их друг на друга различными способами.



12. Создайте новый рисунок произвольного размера и залейте его серым цветом.
13. Создайте копию слоя и примените к нему фильтр *Шум – Шум RGB*. Чтобы шум не был цветным, в окне настройки снимите флажок *Независимые каналы RGB*. Уровень шума настройте помощью движков.
14. Назовите новый слой *Шум* и установите для него режим наложения *Экран*.
15. Создайте копию слоя *Шум*, назовите ее *Шум-2*. Выполните инверсию цвета (черный меняется на белый и наоборот) помощью команды меню *Цвет – Инvertировать*. Установите режим наложения слоя *Умножение*.





16. Для того, чтобы создать эффект рельефа, с помощью инструмента  *Перемещение* сместите слой *Шум-2* на 1 пиксель вправо и на 1 пиксель вниз.
17. Установите непрозрачность слоев *Шум* и *Шум-2* около 20%.
18. Создайте новый слой выше всех существующих слоев и залейте его каким-нибудь цветом или градиентом. Установите для этого слоя режим наложения *Рассеянный цвет*. Сохраните результат.

## Практическая работа (дополнительная)

### Маска слоя

### Маска слоя

Выполним фотомонтаж из двух снимков, один из которых сделан в ясную погоду, а второй – в ненастную. Для этого будем использовать маску слоя, чтобы не портить исходные изображения.




Этот прием можно применять, например, для построения панорамы из нескольких кадров.

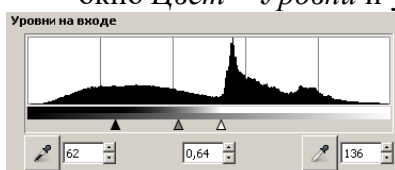
19. Откройте файл **light-lake.jpg**.
20. С помощью меню *Изображение – Размер холста* увеличьте ширину изображения до 1500 пикселей, сохранив неизменной высоту. Сохраните файл под именем **lake.xcf** (обязательно в формате **XCF**, чтобы сохранить слои!)
21. Создайте новый слой из файла **dark-lake.jpg** (меню *Файл – Открыть как слои*) и сдвиньте этот рисунок на правую границу изображения.
22. Добавьте к новому слою маску, выбрав пункт *Добавить маску слоя* из контекстного меню (оно появляется при нажатии правой кнопки мыши в списке слоев). Заполните маску белым цветом (полностью непрозрачная маска).
23. Установите непрозрачность верхнего слоя около 50%, чтобы видеть границы обоих слоев.
24. Выделите значок маски слоя (он должен быть справа от уменьшенного изображения слоя). Заполните ее горизонтальным градиентом от черного до белого цвета так, чтобы черный цвет был там, где начинается верхний слой, а белый – там, где заканчивается нижний.
25. Увеличьте непрозрачность верхнего слоя до 100% и сохраните результат.

### Замена фона для сложного объекта

Заменяем фон, на котором изображен объект. Способ, который мы применим, подходит для выделения сложного объекта: дерева, волос, шерсти животного и т.д. на контрастном фоне. Важно, что при этом мы не испортим исходное изображение – все операции выполняются с помощью маски слоя.





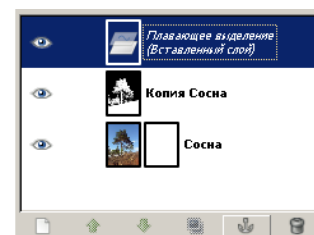
26. Откройте файл **pine-tree.jpg**. Переименуйте единственный слой, дав ему имя *Сосна*.
27. Создайте копию слоя *Сосна*, перетащив его на значок  в нижней части окна *Слои*. Из этой копии мы создадим маску слоя.
28. Выделите слой-копию и преобразуйте изображение в черно-белый вариант (*Цвет – Обесцвечивание*).
29. Теперь превратим черно-белую копию в маску. Для этого сузим диапазон яркостей: сделаем так, чтобы большинство пикселей были либо черными, либо белыми. Откройте окно *Цвет – Уровни* и установите движки примерно так, как на рисунке:



Обратите внимание, что белый движок расположен чуть-чуть левее максимума гистограммы. Это сделано для того, чтобы все светлые пиксели (небо) стали белыми. Результат показан на рис. 1.



30. Нам нужно заменить только небо (белые пиксели), а на нашей маске белыми оказались и другие пиксели, расположенные на скале и на стволе дерева. Включите инструмент  *Кисть* и черным цветом аккуратно закрасьте все лишние светлые места, как на рис. 2.
31. Белый цвет в маске обозначает полную непрозрачность (изображение остается), а чёрный – полную прозрачность. Нам нужно сделать прозрачной как раз область неба, поэтому необходимо выполнить инверсию изображения (заменить чёрный цвет на белый и наоборот, как на рис. 3; меню *Цвет – Инвертирование*).
32. Теперь превратим черно-белый слой в маску слоя *Сосна*. Прежде всего, добавьте маску к этому слою (*ПКМ – Добавить маску слоя*), залив ее белым цветом.
33. Перейдите на слой-копию, выделите все пиксели (*Ctrl+A*) и скопируйте весь слой в буфер обмена (*Правка – Копировать*).
34. Выделите маску слоя *Сосна* (щелкните именно по уменьшенному изображению маски).
35. Вставьте изображение из буфера обмена (*Ctrl+V*), при этом в списке слоев появляется плавающее выделение (см. рисунок справа). Щелкните по кнопке , чтобы прикрепить вставленное изображение к маске слоя.
36. Отключите видимость слоя-копии. Теперь вы должны увидеть вместо неба клетчатый фон.
37. Теперь загрузим новый фон и расположим его под слоем *Сосна*. Откройте файл **rainbow.jpg** с помощью команды *Файл – Открыть как слой*. Перетащите новый слой под слой *Сосна*. Сохраните результат в файле **pine-tree.xcf**.




## Практическая работа № 65.

### Каналы

#### Модель RGB

В этой работе мы научимся разбирать рисунок на отдельные каналы, редактировать каналы и собирать изображение из каналов. Заодно освоим новые способы построения черно-белого изображения из цветного.

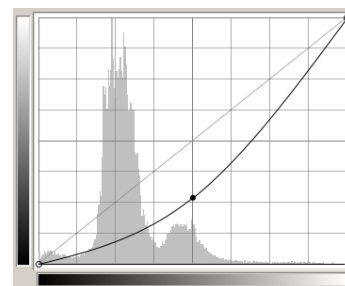
1. Создайте новый рисунок размером 100 на 270 пикселей, заполните его черным цветом.
2. Выберите инструмент  *Эллипс*, включите сглаживание и отключите растушевку краев. Нарисуйте три цветных круга (красный, желтый и зеленый) так, чтобы получился светофор, у которого все лампочки горят.
3. Перейдите в окно *Каналы* и отключите красный канал. Объясните то, что вы увидели.
4. Теперь включите красный канал и отключите зелёный. Затем включите зеленый и отключите синий канал. Объясните результаты.
5. Откройте файл **flower.jpg**. Посмотрите, как меняется изображение, если отключать поочередно красный, зеленый и синий каналы. Какой канал почти не влияет на результат? Какой оказывает наиболее сильное влияние?
6. Постройте новый файл, в котором каждый из каналов выделен в отдельный слой с помощью меню *Цвет – Составляющие – Разобрать* (модель RGB). В полученном изображении каждый канал представляет собой отдельный черно-белый слой.
7. Удалите слои «красный» и «синий». Создайте две копии оставшегося слоя, назовите их «красный» и «синий». Расположите слой «красный» над «зеленым», а «синий» – в самом низу.
8. Постройте новое изображение из этих цветовых каналов (*Цвет – Составляющие – Собрать*). Объясните, почему оно стало чёрно-белым?
9. Сохраните результат в файле **flower-gray.jpg** и закройте файл.
10. Вернитесь к исходной фотографии **flower.jpg**, откройте окно *Цвет – Составляющие – микшер каналов*, отметьте флажок *Монохромный*. Передвигая движки, которые задают влияние каждого из каналов, постройте черно-белое изображение, которой вам больше нравится. Сохраните результат под именем **flower-gray2.jpg** и сравните его с файлом **flower-gray.jpg**.

#### Модель CMYK

Теперь познакомимся с цветовыми каналами модели CMYK (*Cyan – голубой, Magenta – пурпурный, Yellow – жёлтый, black или Key color – ключевой цвет = чёрный*). Эта модель применяется при печати цветных изображений. Поэтому многие профессионалы корректируют фотографии, предназначенные для печати, только в CMYK.

К сожалению, пока GIMP не может напрямую работать с изображениями, закодированными с помощью модели CMYK. Однако можно «разобрать» рисунок на отдельные каналы, затем скорректировать нужные каналы и снова собрать изображение.

11. Откройте файл **sunrise.jpg**.  
*Видно, что в этой фотографии очень много пурпурного (фиолетового) цвета. Уменьшить его влияние удобнее всего с помощью M-канала цветовой модели CMYK.*
12. Постройте новый файл, разделив изображение на каналы в цветовой модели CMYK (*Цвет – Составляющие – Разобрать*, модель CMYK).
13. В окне *Каналы* отключите видимость верхнего слоя «голубой\_k», чтобы видеть слой «пурпурный\_k» и изменения в нем.
14. Выделите слой «пурпурный\_k». Откройте окно с кривыми (*Цвет – Кривые*) и измените кривую примерно так, как показано на рисунке. Для этого нужно «схватить» мышкой середину кривой и сместить ее



немного вниз. Таким образом, мы уменьшили влияние пурпурного цвета во всем тоновом диапазоне, но больше всего – в средних тонах.

15. Закройте окно *Кривые*. Теперь соберем изображение обратно, применив команду меню *Цвет – Составляющие – Воссоединить*. После этого исходное изображение будет «пересобрано» (в том окне, где открыт файл **sunrise.jpg**) из отдельных каналов модели СМΥК.
16. Если пурпурного цвета все равно много, примените такую же процедуру еще раз. Сохраните результат с именем **sunrise-ready.jpg**.

### Модель Lab


*В этой работе мы узнаем, как можно повысить резкость изображения, используя цветовую модель Lab. Она задает физические характеристики цвета и не зависит от свойств устройства вывода.*

*Резкость увеличивается только в канале L (Lightness, светлота), тогда как остальные каналы (определяющие цветовой тон) не изменяются. Это позволяет избежать цветовых искажений на границах объектов, которые нередко появляются при повышении резкости сразу во всех каналах.*


17. Откройте файл **flower.jpg** и сохраните его копию под именем **flower-2.jpg**.
18. Постройте новый файл, разделив изображение на каналы в цветовой модели *Lab* (*Цвет – Составляющие – Разобрать*, модель *Lab*).
19. Выделите в списке слоев слой L (это L-канал модели *Lab*) и примените к нему фильтр *Улучшение – Повышение резкости*.
20. Выберите пункт меню *Цвет – Составляющие – Воссоединить*, чтобы собрать изображение с измененным L-каналом в том же файле.

### Альфа-канал

*Альфа-канал задает прозрачность пикселей слоя. В этой работе вы узнаете, как сделать изображение с прозрачными и полупрозрачными областями. Это может потребоваться, например, при создании картинки, которая может размещаться на фонах разного цвета (фон должен просвечивать через прозрачные области).*

21. Откройте файл **gimp-logo.jpg**.
22. Попробуем удалить белый фон. Включите инструмент  *Волшебная палочка* и щелкните мышью на белом фоне для того, чтобы выделить его.
23. Теперь нажмите клавишу *Delete* (удалить), и убедитесь, что выделенная область была залита фоновым цветом (белым, если вы его не меняли).

*Дело в том, что нижний слой считается фоновым, то есть полностью непрозрачным. Для того, чтобы работать с прозрачными областями, нужно добавить к изображению альфа-канал, который «отвечает» за прозрачность.*

24. Перейдите к списку каналов и убедитесь, что сейчас изображение строится из трех каналов: красного, зелёного и синего.
25. В окне *Слои* нажмите правую кнопку мыши на единственном слое и выберите команду *Добавить альфа-канал*. Посмотрите список каналов – в нем должен появиться канал *Альфа*, который залит белым цветом (это значит, что все пиксели полностью непрозрачные).
26. Теперь еще раз удалите выделенную область, нажав клавишу *Delete*. Теперь вокруг логотипа должен появиться клетчатый фон, обозначающий прозрачность. В альфа-канале эти области закрашены черным цветом.
27. Чтобы избавиться от белого контура по краям рисунка, увеличьте выделенную область на 1 пиксель (*Выделение – Увеличить*) и еще раз удалите ее.
28. Переименуйте существующий слой в *Логотип*. Создайте новый прозрачный слой, дайте ему имя *Тень* и разместите его под слоем *Логотип*.
29. Включите инструмент  *Эллипс* и установите радиус растушевки 40-50 пикселей. На слое *Тень* выделите область тени и залейте ее черным цветом. Посмотрите, как изменился альфа-канал.



30. Установите для слоя *Тень* непрозрачность около 30% и снова посмотрите на изменения в альфа-канале.
31. Самостоятельно нарисуйте тень от кисточки.
32. Для сохранения изображений с прозрачными и полупрозрачными областями удобно использовать формат PNG, который умеют показывать все современные браузеры. Сохраните результат под именем **gimp-logo.png** (настройки можно не менять).

## Практическая работа № 66. Иллюстрации для веб-сайтов

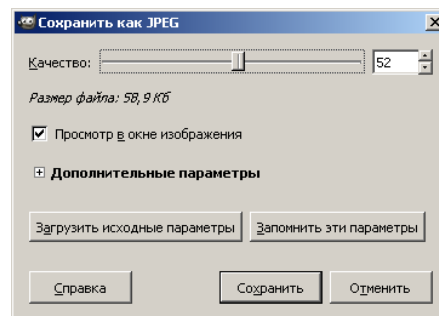
В этой работе мы научимся сохранять изображения, предназначенные для размещения на веб-сайтах. Главная задача – добиться минимального размера и сохранить приемлемое качество.

20. Откройте файл **travel.xcf**.

21. Установите масштаб 200% (*Вид – Масштаб – 2:1 (200%)*). Передвиньте рисунок так, чтобы видеть границу между горами и небом, в этом месте будут особенно ясно видны искажения при сжатии в формате JPEG.

22. Выберите пункт меню *Файл – Сохранить копию*, в появившемся окне введите имя файла **travel.jpg**.

23. В окне *Сохранить как JPEG* отметьте флажок *Просмотр в окне изображения*. Сдвигая движок *Качество*, наблюдайте (в этом же окне), как изменяется размер файла, а в основном окне с фотографией – внешний вид получившегося изображения. Смотрите, как при снижении качества (т.е. при увеличении степени сжатия) появляются искажения: квадраты 8 на 8 пикселей и лишние темные точки на границе светлых и темных областей.



24. Выберите минимальное качество, которое вы считаете допустимым, и сохраните файл **travel.jpg**.

Какой уровень качества вы считаете допустимым?

Ответ:

25. Попробуем сохранить фотографию в формате GIF (с палитрой). Выберите пункт меню *Файл – Сохранить копию*, в появившемся окне введите имя файла **travel.gif**.

Сохраните файл, не меняя никаких настроек.

26. Откройте файл **travel.gif**, увеличьте изображение и посмотрите, устраивает ли вас его качество. Сделайте выводы. Закройте файл.

Ответ:

27. Сохраните копию файла в формате PNG (для этого нужно ввести имя **travel.png**). Он использует сжатие без потерь, поэтому качество не ухудшится. Сравните размеры файлов **travel.jpg** и **travel.png**.

Ответ:

28. По результатам этого эксперимента сделайте выводы о том, в каком формате лучше всего готовить фотографии для веб-сайта.

Ответ:

## Практическая работа № 67.

### Анимация

#### Простая анимация (замена)

В этой работе мы научимся строить простые анимированные файлы.

Каждый кадр анимации – это отдельный слой.



1. Выберите команду *Файл – Открыть как слои* из верхнего меню и выберите файлы **pause.png**, **letter-m.png**, **letter-i.png**, и **letter-r.png**.
2. Переставьте слои так, чтобы слой с буквой М стояли в самом низу, далее – слой с буквой И, затем – слой с буквой Р, и в самом верху – слой «пауза». Сохраните получившийся файл с 4 слоями под именем **message.xcf** (*Файл – Сохранить как*).
3. Откройте окно фильтра *Анимация – Воспроизведение* и запустите анимацию, щелкнув по кнопке *Воспроизвести*. Вы увидите, что анимация выполняется очень быстро (кадры по умолчанию сменяются через 0,1 с).
4. Чтобы увеличить интервал до 1 с, измените названия всех слоев, добавив в конец названий текст «(1000ms)» (без пробела, латинские буквы «m» и «s»). Проверьте, как это влияет на скорость анимации.
5. Используя копирование слоев, добейтесь того, чтобы сигнальщик передавал сообщение:

**МИР <пауза> РИМ <пауза>**

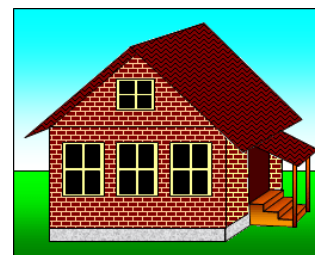
причем каждая пауза должно длиться 2 с.

6. Сохраните результат под именем **message.gif**, выбрав вариант *Сохранить как анимацию*. В окне параметров выберите в списке *Расположение кадров* режим *Один кадр на слой (замена)*.

#### Анимация с объединением слоев

Второй вид анимации использует наложение слоев друг на друга (*объединение*).

7. Откройте файл **house.psd** (это файл со слоями, сделанный в редакторе *Adobe Photoshop*).
8. Расставьте слои в таком порядке, в котором обычно происходит строительство дома (снизу вверх).
9. Постройте анимацию, в которой смена кадров происходит через 1 с.
10. Сохраните анимацию в формате GIF, указав, что используется режим *Наложение слоев*.





## Практическая работа № 68.

### Векторная графика

#### Уровень А.

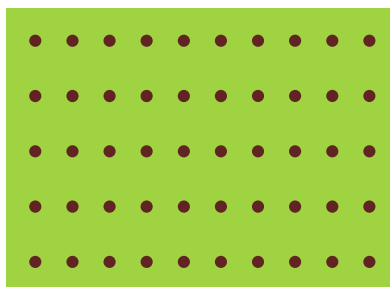
1. Нарисуйте домик (вы можете выбрать уровень сложности):



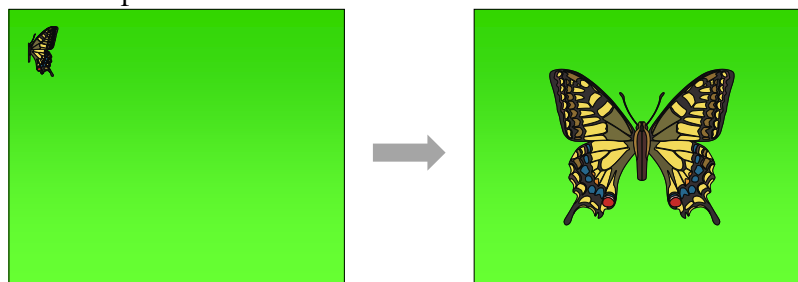
2. Сгруппируйте все элементы рисунка и постройте его копию, уменьшенную в два раза.

#### Уровень В.

3. Иллариону нужно составить план посадки картофеля на своём участке. Он хочет посадить 5 рядов по 10 кустов картошки в каждом. Помогите ему составить план поля:

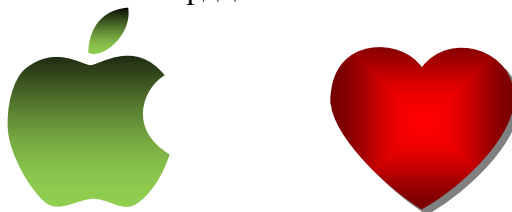


4. Откройте файл **butterfly.doc** (или **butterfly.rtf**, или **butterfly.odg**). Половина бабочки нарисована в векторном редакторе как набор мелких элементов. Нужно построить полное изображение бабочки большого размера и разместить его по центру прямоугольника с фоном.



#### Уровень С.

5. Нарисуйте надкушенное яблоко и сердце:




Постарайтесь использовать как можно меньше узлов.

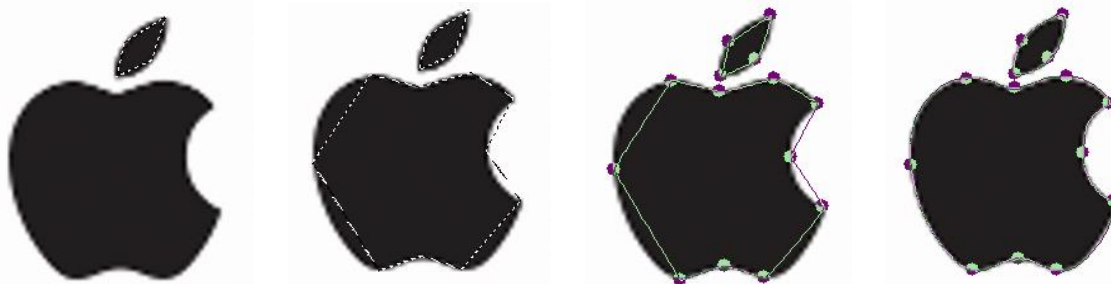
## Практическая работа № 69.



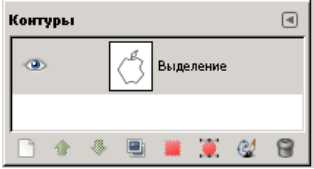


### Контур в GIMP

#### Знакомство с контурами

В этой работе мы построим стилизованный логотип фирмы Apple. Для этого применим контуры – векторные объекты, которые позволяют выделить область сложной формы.

1. Откройте файл **apple.jpg** из папки 71. Это логотип Apple, который мы будем использовать как шаблон. К сожалению, это изображение имеет слишком низкое разрешение. Мы построим новый, более качественный рисунок.
2. Увеличьте размер изображения так, чтобы его ширина была равна 400. Сохраните новый файл под именем **apple.xcf**.
3. Включите инструмент  Лассо и выделите (неточно) листик яблока четырьмя щелчками мыши (рис. 1).




4. Чтобы добавить новое выделение к существующему, при нажатой клавише *Shift* выделите яблоко по опорным точкам (рис. 2).
  5. Откройте окно *Контур* (меню *Окна – Прикрепляющиеся диалоги – Контур*). Щелкнув по кнопке  в нижней части этого окна, превратите выделение в контур. Затем отмените выделение (*Выделение – Снять* или *Shift+Ctrl+A*) и сделайте контур видимым, щелкнув в левой части напротив имени контура (там появится значок ).
- 
6. Включите инструмент  *Контур* и выделите контур, щелкнув по нему мышкой в основном окне. Узлы контура должны быть обозначены кружками.
  7. Удалите лишние узлы, оставив только те, которые показаны на рис. 3. Для этого нужно щелкать по ним, удерживая клавиши *Shift* и *Ctrl*.
  8. Перемещая мышкой узлы контура и сегменты (отрезки между узлами), подгоняйте форму контура к форме яблока. Окончательно подправьте контур с помощью «рукояток», задающих кривизну линии у каждого узла (рис. 4).
  9. Отключите видимость контура. Перейдите в окно *Слой*, выделите всё (*Ctrl+A*) и удалите содержимое слоя.
  10. В окне *Контур* выделите сохраненный контур и превратите его в выделение, щелкнув по кнопке  в нижней части окна.
  11. Включите инструмент *Градиент*, в области настроек инструмента выберите из готовых шаблонов любой градиент, который вам понравится, и залейте выделенную область.
  12. Установите основным цветом цвет, которым вы хотите обвести контур, и выполните обводку (*Правка – Обвести выделенное*).
  13. Примените к выделенной области фильтр *Свет и тень – Отбрасываемая тень*. Сохраните результат.

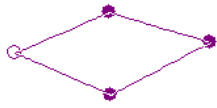
#### Контур помогает рисовать куб

В этой работе мы построим изображение кубика. Для того, чтобы все грани идеально стыковались между собой, мы построим три контура, которые будут получены из одного.

14. Создайте новый рисунок размером 300 на 300 пикселей.



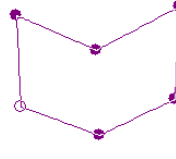
15. Включите инструмент  *Контур* и включите на панели свойств инструмента флажок Многоугольник. Постройте новый контур, ограничивающий верхнюю грань (рис. 1) и назовите его *Верх*.



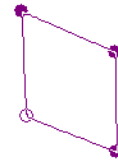
1




2



3



4

16. Скопируйте этот контур (кнопка  в нижней части окна *Контур*) и назовите контур-копию *Левая грань*.
17. Переместите верхний узел вниз, в точку, где будет нижняя вершина куба. Добавьте на боковые стороны по одному новому узлу (рис. 2) – для этого щелкните на них при нажатой клавише *Ctrl*.
18. Переместите новые узлы в точки, где будут располагаться нижние боковые вершины (рис. 3).
19. Скопируйте контур *Левая грань* и назовите копию *Правая грань*.
20. Выделите в окне *Контур* контур *Левая грань* и удалите два правых узла, щелкнув на них при нажатых клавишах *Shift+Ctrl* (рис. 4).
21. Выделите в окне *Контур* контур *Правая грань* и таким же способом удалите два левых узла.
22. Залейте фоновый слой градиентом *Deep Sea*.
23. Создайте новый слой *Кубик*. Используя контуры для выделения областей (как в предыдущей работе), залейте грани куба градиентами примерно так, как показано на рисунке.
24. Сохраните результат.



## Глава 18. Трёхмерная графика

### Версия с русскоязычным интерфейсом

### Настройка Blender

Новейшая русификация:

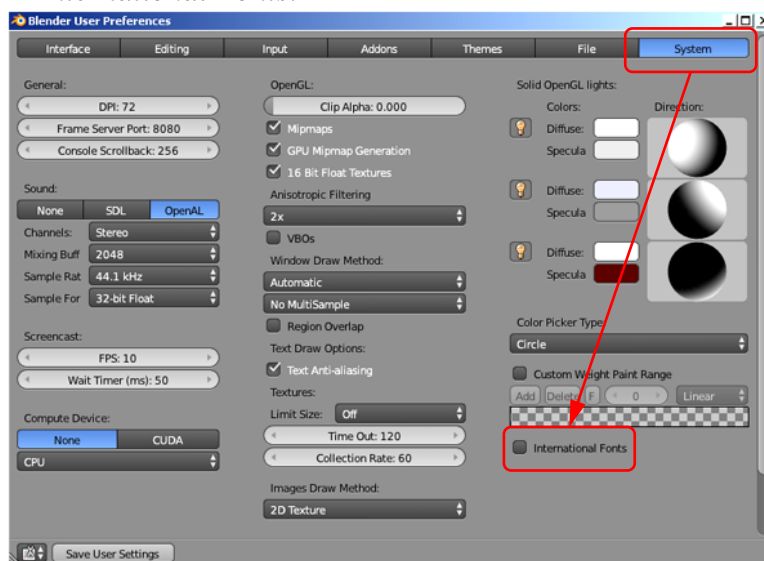
На сайте <https://translateblender.ru> скачать файл blender.mo по ссылке <https://translateblender.ru/downloads/blender.mo>

и сохранить его в папку

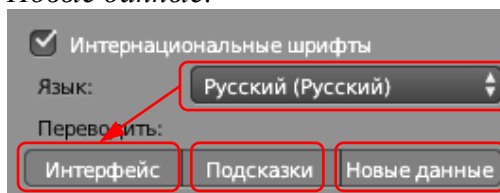
C:/Program Files/blender/2.68/datafiles/locale/ru/LC\_MESSAGES/blender.mo

Включение русскоязычного интерфейса:

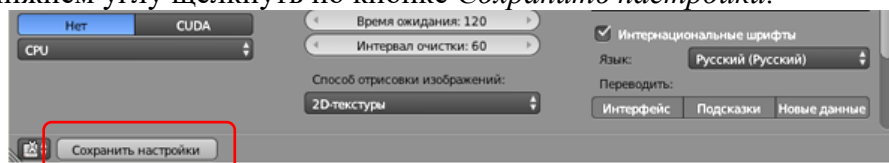
Запустить *Blender*, зайти в меню *File – User Preferences*, перейти на вкладку *System* и включить флажок *International Fonts*:



Затем здесь же выбрать язык *Русский* и левой кнопкой мыши щёлкнуть по кнопкам *Интерфейс*, *Подсказки* и *Новые данные*:

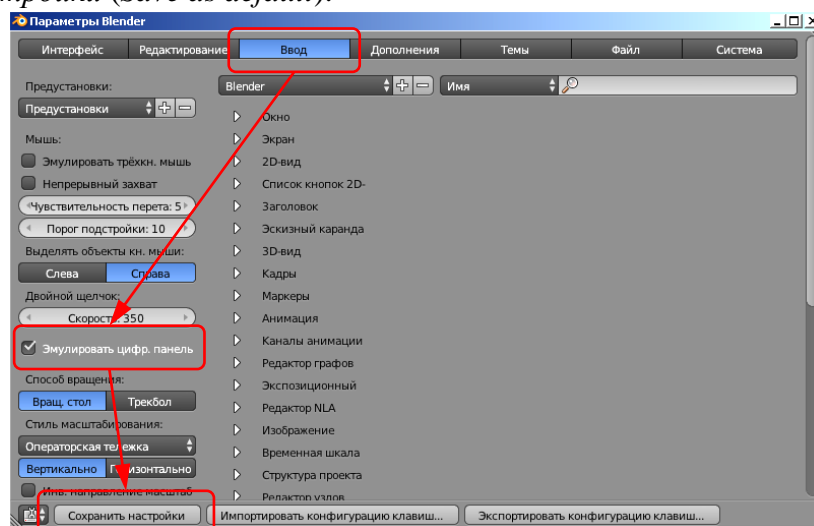


В левом нижнем углу щёлкнуть по кнопке *Сохранить настройки*:



Настройка для ноутбуков:

Программа *Blender* использует дополнительную цифровую клавиатуру (англ. *Numpad*) для переключения между различными проекциями, вращения и перемещения рабочей проекции. Для ноутбуков, у которых этой клавиатуры чаще всего нет, можно «переключить» её на стандартные цифровые клавиши. Для этого нужно войти в окно настроек *Файл – Параметры* (*File – User Preferences* или клавиши *Ctrl+Alt+U*), отметить флажок *Эмулировать цифр. панель* (*Emulate Numpad*) и щёлкнуть по кнопке *Сохранить настройки* (*Save as default*).





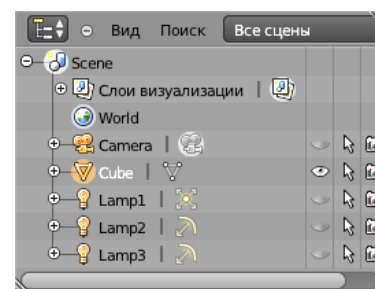
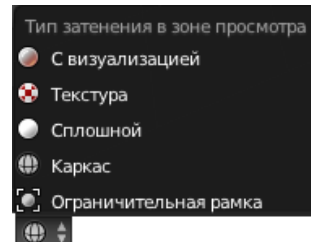
## Практическая работа № 70.

### Введение в 3D-моделирование. Управление сценой

- 1) Запустите *Blender*. Откройте в *Blender* модель **container.blend** (меню *Файл - Открыть, File – Open*).

**Важно!** При всех операциях с окном трехмерной проекции (3D View) курсор должен находиться в рабочей области этого окна.

- 2) Вращая колесико мыши, измените масштаб изображения так, чтобы модель занимала основную часть рабочей области.
- 3) С помощью элемента управления  в нижней части окна установите режим просмотра *Сплошной (Solid)*. Что изменилось?
- 4) Нажмите клавишу *Z*, чтобы вернуться обратно в режим *Каркас (Wireframe)*.
- 5) Для того, чтобы перейти к ортогональной проекции, нажмите клавишу *Num5* («5» на цифровой клавиатуре). Что изменилось? Нажмите еще раз на клавишу *Num5*, чтобы перейти обратно к перспективной проекции.
- 6) Установите режим просмотра *Текстура (Textured)*. Что изменилось?
- 7) Нажмите на колёсико (среднюю кнопку мыши) и поворачивайте контейнер, начиная вращение с разных точек поля.
- 8) Нажмите клавишу *Shift* и попробуйте перемещать видимую область, нажав на колёсико мыши.
- 9) Используя перемещение и вращение, посмотрите на контейнер со всех сторон. Запишите в тетрадь то, что написано на каждой из шести граней.
- 10) Попробуйте установить стандартные проекции: вид сверху (клавиша *Num7*), вид снизу (*Ctrl+Num7*), вид спереди (*Num1*), вид снизу (*Ctrl+Num1*), вид справа (*Num3*), вид слева (*Ctrl+Num3*). Курсор мыши при этом должен находиться над рабочим полем.
- 11) Попробуйте выбирать те же самые команды из меню *Вид (View)* в нижней части окна.
- 12) Попробуйте вращать изображение с помощью клавиш *Num2, Num4, Num6, Num8*.
- 13) Попробуйте сдвигать изображение с помощью клавиш *Ctrl+Num2, Ctrl+Num4, Ctrl+Num6, Ctrl+Num8*.
- 14) Перейдите в режим четырех проекций (*Quad View*, клавиши *Ctrl+Alt+Q*). Вернитесь обратно с помощью той же комбинации клавиш.
- 15) Постройте готовую картинку (выполните *рендеринг*), нажав клавишу *F12*.
- 16) Сохраните рисунок в виде файла в именем **container1.png** в своей папке (клавиша *F3*). Курсор мыши при этом должен находиться над окном с рисунком.
- 17) Нажмите клавишу *Esc*, чтобы вернуться к окну 3D-проекции (3D View).
- 18) В окне *Структура проекта (Outliner)* посмотрите, какие объекты есть на сцене.
- 19) Щелкнув на значках , включите изображение на сцене для ламп (объекты *Lamp1, Lamp2* и *Lamp3*) и камеры (объект *Camera*).
- 20) Выделите щелчком мыши название камеры в окне *Структура проекта*. Используя колёсико мыши, измените масштаб так, чтобы видеть камеру и лампы на рабочем поле.
- 21) Нажав клавишу *Num0*, переключитесь на вид с камеры.
- 22) Нажмите клавиши *Shift+F*, чтобы перевести камеру в режим «полёта» (англ. *fly mode*). Теперь, перемещая мышку, вы можете настроить вид в камере так, чтобы






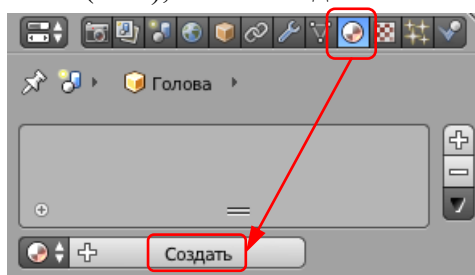
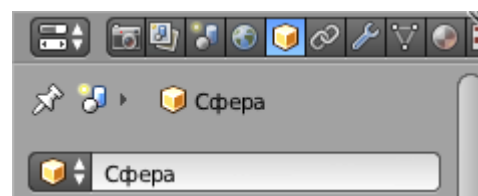
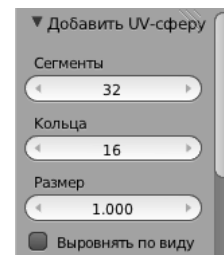
хорошо видеть весь контейнер. Для изменения масштаба используйте колёсико мыши. Когда найден нужный вид, нажмите левую кнопку мыши (ЛКМ).

23) Выполните рендеринг (F12) и сохраните рисунок под именем **container2.png**.

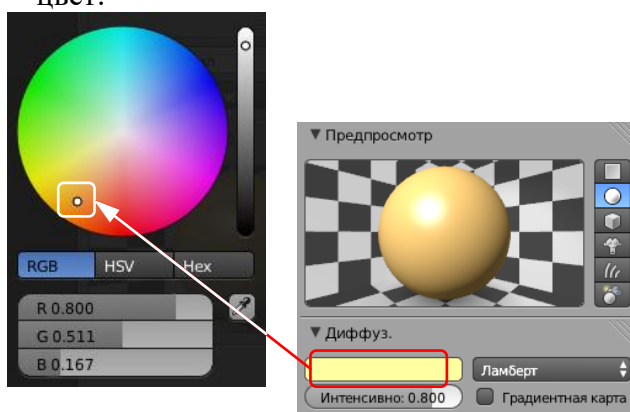
## Практическая работа № 71.

### Работа с объектами

27. Откройте *Blender*, при этом загружается сцена, установленная по умолчанию: куб в начале координат, камера и лампа.
28. Куб сейчас выделен. Удалите куб, нажав клавишу *Delete*. В начале координат остался курсор . Он показывает место, где будет создан новый объект.
29. С помощью верхнего меню *Добавить – Полисетка – UV-сфера* (*Add – Mesh – UV-sphere*) создайте новую сферу. Увеличьте масштаб изображения (колёсиком мыши) и переключитесь в режим каркаса (клавиша *Z*).
30. На панели инструментов (слева внизу) уменьшите число сегментов (*Segments*) до 16, увеличьте число колец (*Rings*) до 20 и размер (*Scale*) до двух (см. рисунок). Наблюдайте за изменением каркаса модели.
31. С помощью элемента управления  в нижней части окна установите режим просмотра *Сплошной* (*Solid*).
32. Найдите панель свойств в правой части окна программы, перейдите на страницу *Объект* (*Object*). Измените название объекта, например, на *Голова* (*Head*).
33. Перейдите на страницу свойств  *Материал* (*Material*) и щелкните по кнопке *Создать* (*New*), чтобы создать новый материал:



34. Щелкните по цветовому полю под словом *Диффузный* (*Diffuse*) – этот так называемый диффузный цвет и есть цвет предмета в нашем понимании, и выберите цвет:



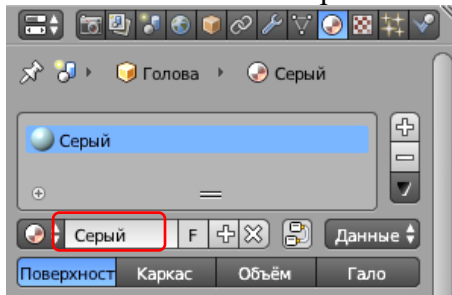
Попробуйте менять составляющие цвета в модели RGB: щелчок мыши переводит поле в режим редактирования. Кроме того, можно «схватить» поле (нажать ЛКМ и не отпускать её) и перетащить значение влево или вправо (соответственно уменьшая или увеличивая его).

35. Попробуйте менять цвет в цветовой модели HSV. Посмотрите, как смещается указатель на поле при изменении параметров *H* (англ. *hue* – тон), *S* (англ. *saturation* – насыщенность) и *V* (англ. *value* – величина или яркость).
36. Перейдите в режим ввода шестнадцатеричного кода цвета (кнопка *Hex* под цветовым

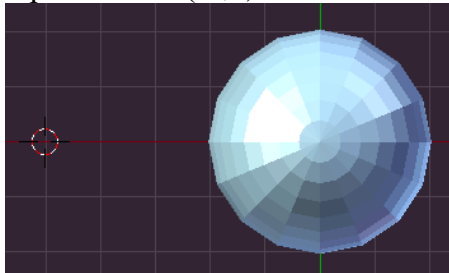


кругом) и задайте цвет A5C8CE.

37. Измените название материала на *Серый* (серый):



38. Переключитесь на вид сверху (Num7) и щелчком ЛКМ поставьте курсор в точку с координатами (-5,0):

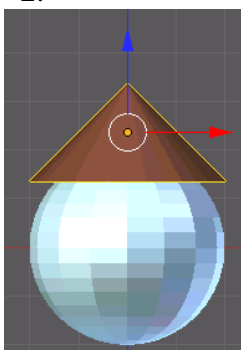
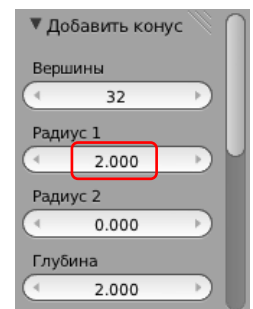


При создании нового объекта его начало координат будет находиться в этой точке.

39. Сохраните сцену под именем **objects.blend**.

40. Добавьте на сцену новый конус. Для этого нажмите клавиши *Shift+A* и выберите из всплывающего меню пункт *Полисетка – Конус (Mesh – Cone)*.

41. В нижней части панели инструментов увеличьте радиус конуса до 2.



42. Переключитесь на вид спереди (*Num1*), схватите мышью за центр конуса (внутри белой окружности) и поставьте конус на сферу. Задайте для конуса имя *Шляпа (Hat)* и материал коричневого цвета. Дайте материалу название *Коричневый (Brown)*.

43. Попробуйте выделять (ПКМ = правой кнопкой мыши) разные объекты и двигать их.

44. Постройте еще один конус и придайте ему форму морковки (это будет нос снеговика):

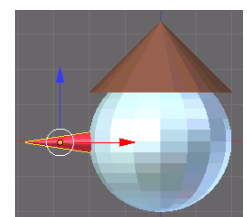
45. Нажмите клавишу *N*, чтобы появилось окно свойств объекта.

46. Нажмите клавишу *R* (*Rotate*), чтобы перевести конус в режим вращения. Мышкой поверните конус в нужное положение (положите его на бок) и наблюдайте, как меняются значения на панели свойств. Щелчок ЛКМ завершает перемещение. Отмените поворот, нажав клавиши *Ctrl+Z*.

47. Нажмите клавишу *R* (поворот), затем – клавишу *Y* (поворот относительно оси *Y*). После этого наберите на клавиатуре «-90» (поворот на -90 градусов). Отмените поворот, нажав клавиши *Ctrl+Z*.


48. Щелкните мышкой на панели свойств в поле *Вращение (Rotation)* – *Y* и измените значение на -90. Все три испробованных способа должны приводить к одному и тому же результату.

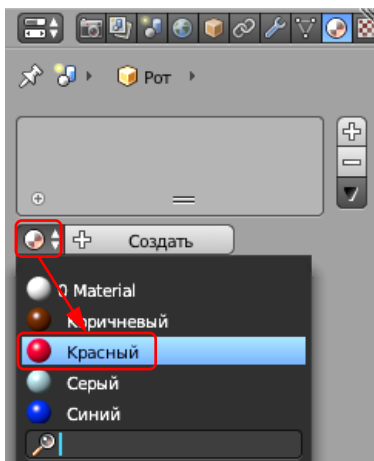
49. Поставьте нос на место, передвинув его на виде спереди. Задайте для этого конуса имя *Нос (Nose)* и материал красного цвета. Дайте материалу название *Красный (Red)*.




50. Создайте новую сферу типа *Полисетка – Икосаэдр (Icosphere)*. Перейдите в режим

*Каркас (Wireframe, клавиша Z)* и сравните, чем отличаются каркасы двух сфер. Вернитесь обратно в режим просмотра *Сплошной (Solid)*.

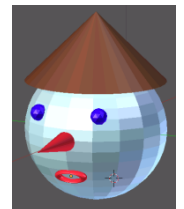
51. Нажмите клавишу S (scale, изменение размеров) и, перемещая мышью, уменьшите новую сферу так, чтобы она могла быть добавлена к голове снеговика как глаз (щелчок ЛКМ завершает изменение размеров). Наблюдайте за изменением размеров на панели свойств.
52. Задайте ей имя *Глаз-1 (Eye-1)* и синий цвет материала. Назовите материал *Синий (Blue)*.
53. Создайте копию полученного глаза, нажав клавиши *Shift+D*, затем клавишу X (перемещать копию только вдоль оси X) и передвинув копию мышкой влево или вправо (щелчок ЛКМ завершает перемещение). Дайте ей имя *Глаз-2 (Eye-2)*. Проверьте, какой материал был выбран для копии.
54. Выделите шляпу и временно отключите её изображение на экране, щелкнув по значку  в окне *Структура проекта (Outliner, в правой верхней части окна)*.
55. Выделите оба глаза (ПКМ при нажатой клавише *Shift*) и передвиньте их вверх (по оси Y) на уровень, где должны быть глаза.
56. Перейдите к виду сверху (клавиша *Num7*) и поставьте глаза на место симметрично относительно носа. Проверьте правильность установки, покрутив голову в разные стороны, потом верните шляпу на место.



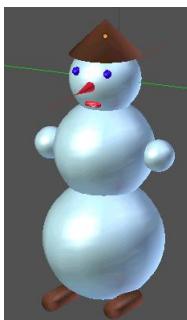
57. Добавьте новый объект – тор («бублик»), меню *Добавить – Полисетка – Тор (Add – Mesh – Torus)*. Измените его параметры так, чтобы получилось изображение рта. Задайте для объекта имя *Рот (Mouth)*.

58. Перейдите на страницу свойств  *Материал (Material)* и щелкните на кнопке слева от кнопки *Создать (New)* для того, чтобы назначить тору существующий материал. Выберите из списка материал *Красный*.

59. Используя перемещение и вращение, установите рот на место.



60. Выделите сферу-голову и попробуйте переместить ее. Передвинулись ли остальные элементы?
61. Выделите одновременно глаза, нос, рот и шляпу, а затем (в самом конце!) – сферу-голову. Теперь нажмите на клавиши *Ctrl+P*, чтобы сделать последний объект родительским для всех остальных. Попробуйте теперь перемещать голову. Что наблюдаете? Попробуйте перемещать шляпу. Переместились ли остальные объекты?
62. В окне *Структура проекта (Outliner)* посмотрите, как изменилось расположение объектов в структуре сцены.
63. Используя сферы и два цилиндра для ног, объекты *Цилиндр (Cylinder)*, достройте полную фигуру снеговика, как на рисунке. Для ног установите тот же материал, что и для шляпы.
64. Установите правильно камеру и выполните рендеринг. Сохраните



полученную

**Задание для самостоятельной работы:**  
Постройте трехмерную фигуру в файле



картинку в файле **snowman.png**.

**самостоятельной работы:**

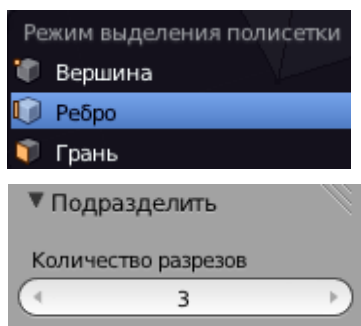
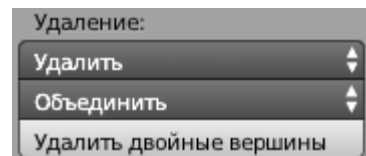
колонну из объектов-примитивов. Сохраните модель в файле **column.blend**, а результат рендеринга – в **column.png**.

## Практическая работа № 72.

### Сеточные модели

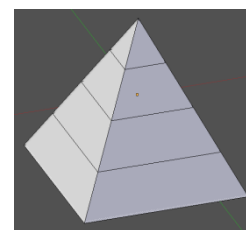
1. Откройте *Blender*, при этом загружается сцена, установленная по умолчанию: куб в начале координат, камера и лампа.
2. Перейдите в режим редактирования (*Edit Mode*), нажав клавишу *Tab*. Отмените выделение вершин, нажав на клавишу *A* (англ. *all* – все).
3. Нажмите клавишу *B*, чтобы перейти в режим выделения прямоугольником. Обведите левой кнопкой мыши все 4 вершины верхней грани. Чтобы свести их вместе и построить пирамиду, применим масштабирование с коэффициентом 0. Для этого нужно нажать клавишу *S*, набрать на клавиатуре 0 и нажать *Enter*. Теперь 4 вершины находятся в одной точке.

4. Чтобы оставить только одну вершину из этих четырёх, щёлкните по кнопке *Удалить двойные вершины* (*Remove Doubles*) на панели инструментов (в левой части окна). Убедитесь, что справа в заголовке окна появилось сообщение «Удалено 3 вершины» (*Removed 3 vertices*).




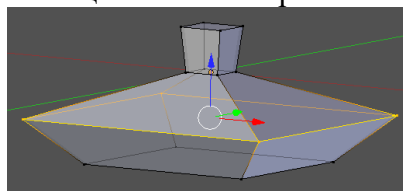
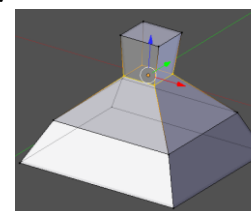
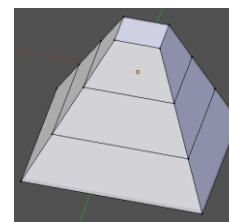
5. Нажмите на клавиши *Ctrl+Tab* и выберите в появившемся всплывающем меню пункт *Рёбра* (*Edges*) чтобы включить режим выделения рёбер.


6. Перейдите к виду сверху (*Num7*) и выделите все 4 наклонных ребра (ПКМ при нажатой клавише *Shift*). Щёлкните по кнопке *Подразделить* (*Subdivide*) на панели инструментов слева и установите *Количество разрезов*

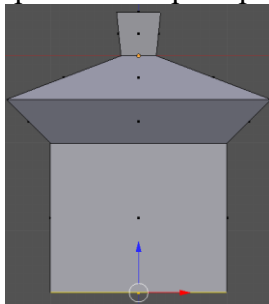


(*Number of Cuts*), равное 3 (чтобы разбить каждое ребро на 4 равных части). Пирамида будет разделена на 4 слоя.

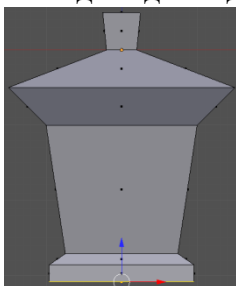
7. С помощью элемента управления  в нижней части окна включите режим работы с вершинами (самая левая кнопка).
8. Чтобы построить усеченную пирамиду, выделите вершину и нажмите клавишу *Delete*. Появится всплывающее меню, в котором нужно выбрать удаляемые элементы. Выберите *Вершины* (*Vertices*). Посмотрите на пирамиду сверху – появилось отверстие в торце.
9. Нажмите клавишу *C* (англ. *circle selection*, круговое выделение), колёсиком мыши отрегулируйте область выделения, которая обозначается белой точечной окружностью. Выделите с помощью ЛКМ все 4 верхние вершины и нажмите клавишу *Esc*, чтобы закончить выделение.
10. Чтобы создать закрывающую грань сверху, нажмите клавишу *F*.
11. Выделите первый промежуточный ряд вершин, нажав клавишу *Alt* и щёлкнув по одному из рёбер, образующих контур. Применив масштабирование (клавиша *S*), уменьшите размеры контура.
12. Выделите следующий контур и увеличьте его размер с помощью масштабирования:



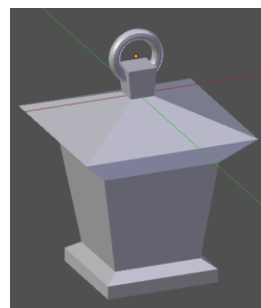
13. С помощью элемента управления  включите режим работы с гранями (последняя кнопка), поверните объект так, чтобы видеть нижнюю грань, и выделите ее. Нажмите кнопку E (англ. *extrude* – выдавливание) и переместите выделенную грань вниз примерно так, как на следующем рисунке.



14. Используя масштабирование, уменьшите размеры нижней грани. Затем, используя дважды выдавливание и масштабирование, завершите построение нижней части:




15. Добавьте колечко, за которое такой фонарь можно подвесить – это объект *Top (Torus)*:  
16. Сохраните модель под именем **light.blend**.

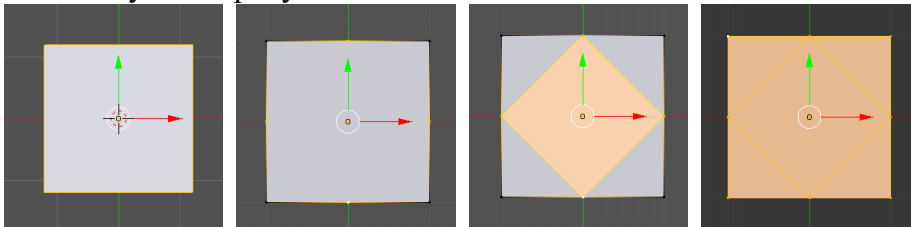




## Практическая работа № 73.

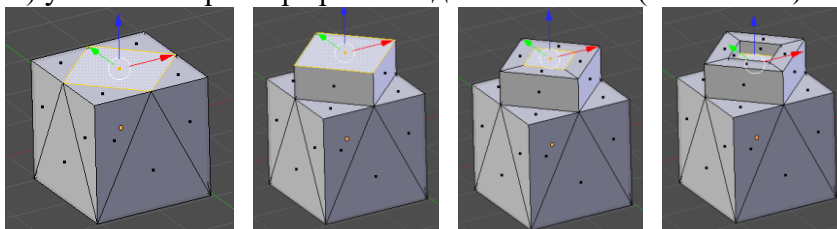
### Модификаторы

#### Тренировка

1. Откройте *Blender* со сценой по умолчанию. Перейдите к виду сверху (клавиша *Num7*) и включите режим работы с гранями (клавиши *Ctrl+Tab* или ). Сейчас мы построим нестандартную разбивку верхней грани куба так, как показано на следующих рисунках:




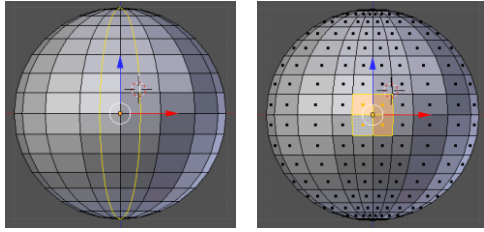
2. Перейдите в режим редактирования (клавиша *Tab*) и отмените выделение всех граней (клавиша *A*). Проверьте, чтобы кнопка  в нижней части рабочей области, которая ограничивает выделение только видимыми гранями (запрещает выделение на обратной стороне фигуры) была включена.
3. Перейдите к ортографической проекции (*Num5*). Выделите и удалите верхнюю грань: нажмите клавишу *Delete*, во всплывающем меню выберите объекты для удаления *Грани (Faces)*.
4. Переключитесь в режим работы с ребрами (*Ctrl+Tab* или ). Выделите 4 верхних ребра и разбейте их пополам с помощью инструмента *Подразделить (Subdivide)*.
5. Выделите только что добавленные вершины в серединах рёбер и соедините их новой гранью (клавиша *F*). Затем выделите все верхние угловые вершины и снова нажмите клавишу *F*, чтобы достроить 4 угловых грани на верхней стороне куба.
6. Перейдите в режим работы с гранями и выделите центральную грань. Нажмите клавишу *E* и выдавите эту грань вверх. Снова нажмите клавишу *E* и сразу *Enter*, чтобы продублировать угловые вершины. С помощью масштабирования (клавиша *S*) уменьшите размер грани и вдавите ее вниз (клавиша *E*).



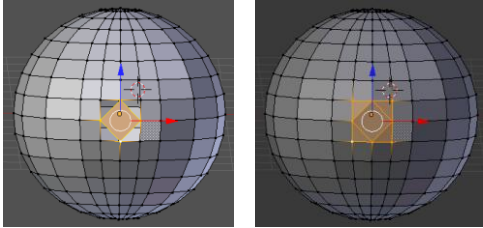
7. Сохраните модель под именем **mesh.blend**.

#### Модель головы слоника

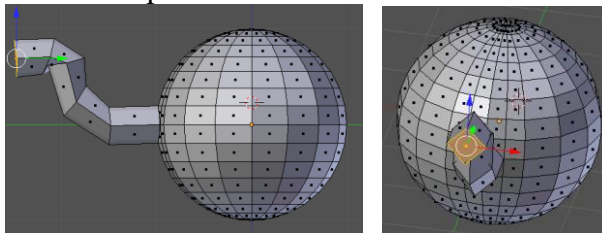
1. Создайте новую сцену. Выделите куб и удалите его (клавиша *Delete*).
2. Добавьте на сцену новый объект – сферу (*UV-sphere*) и уменьшите число её сегментов до 16.
3. Включите вид спереди (клавиша *Num1*), ортографическую проекцию (*Num5*) и перейдите в режим редактирования (*Edit Mode*) с помощью клавиши *Tab*.
4. Отмените выделение (клавиша *A*).
5. Включите режим работы с гранями (*Ctrl+Tab* или ). С помощью инструмента *Разрезать петли со сдвигом (Loop Cut and Slide)* сделайте два сечения по «меридианам», как показано на рисунке.






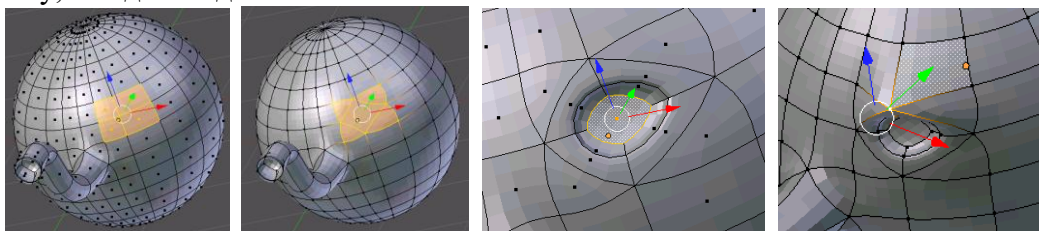
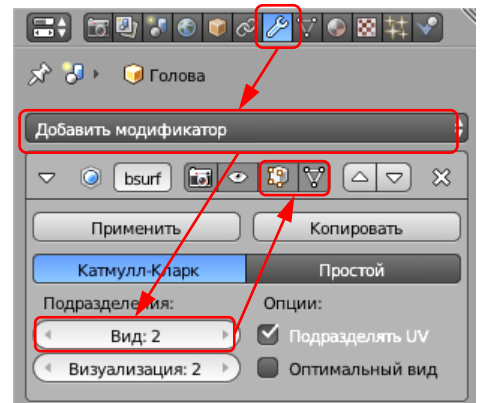
6. Выделите четыре центральных грани и удалите их (клавиша *Delete*, во всплывающем меню выбрать *Грани, Faces*).
7. Перейдите в режим работы с вершинами, выделите четыре вершины, как показано на рисунке, и постройте грань между ними (клавиша *F*). Достройте оставшиеся треугольные грани.



8. Переключитесь в режим редактирования граней, выделите центральную грань и перейдите к виду справа (*Num3*). Нажмите клавишу *Ctrl* и, удерживая её, щелчками мыши постройте хобот:

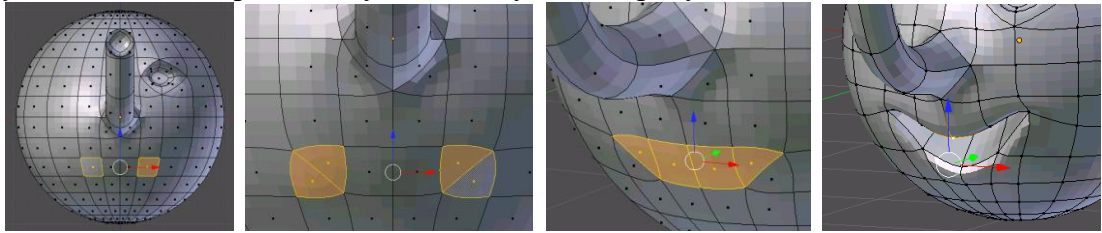


9. Выделите грань в торце хобота, нажмите клавише *E* (выдавливание) и сразу же клавишу *Enter* (будет создана копия всего контура грани). Затем снова нажмите клавишу *E* и вдавите торец внутрь хобота.
10. Примените модификатор *Подразделение поверхности (Subdivision surface)*. Для этого нужно перейти на страницу свойств  *Модификаторы (Modifiers)*, щелкнуть по кнопке *Добавить модификатор (Add Modifier)* и выбрать модификатор *Подразделение поверхности (Subdivision surface)*.
11. В свойствах модификатора в поле *Вид (View)* увеличьте число делений грани при просмотре до 2. Для того, чтобы модификатор применялся в режиме редактирования сеточной модели, включите кнопки-выключатели  и .
12. Выделите 4 грани из которых будем строить глаз. Удалите их и постройте такую же сетку, как для выдавливания хобота:

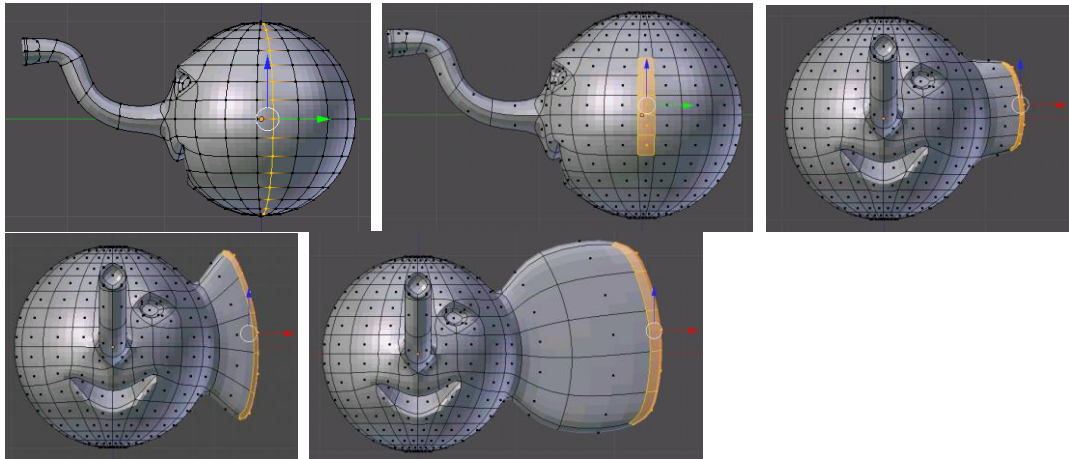



13. Выделите центральную грань отмеченной части, нажмите клавише *E* (выдавливание) и сразу же клавишу *Enter* (будет создана копия всего контура

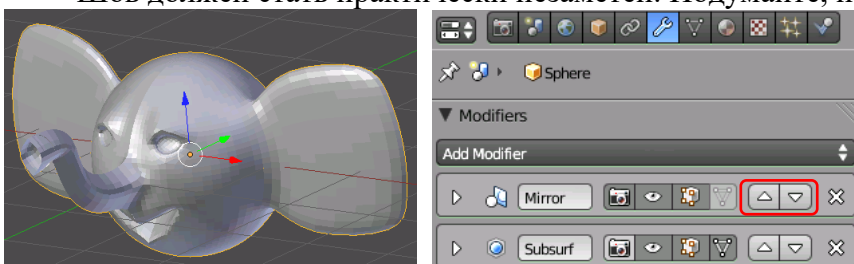
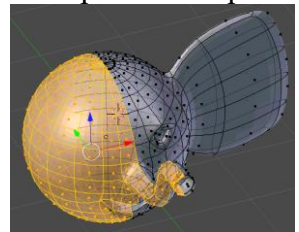
- грани). Затем постройте внутренний контур, используя масштабирование (клавиша E). Затем снова нажмите клавишу E и вдавите центральную грань немного внутрь.
14. Перейдите к виду сверху и передвиньте вершину над глазом немного вперед.
  15. Перейдите к виду спереди. Выделите две грани, в которых будут уголки рта, и удалите их. Постройте такую же сетку, как на рисунке.



16. Выделите 4 грани (см. рисунок) и вдавите их внутрь. Перейдите к виду спереди и измените положение вершин так, чтобы рот стал улыбающимся.
17. Перейдите к виду справа и примените инструмент *Разрезать петлей со сдвигом* (*Loop Cut and Slide*) для одной из граней. Затем выделите 5 граней и примените выдавливание (клавиша E). После этого используйте масштабирование (клавиша S) и затем снова выдавливание.

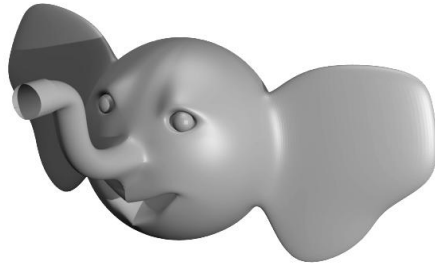


18. Перейдите к виду спереди. Проверьте, чтобы кнопка , которая ограничивает выделение только видимыми гранями (запрещает выделение на обратной стороне фигуры) была отключена. Включите режим работы с гранями и нажмите клавишу В (выделение прямоугольником) и выделите все грани левой части фигуры. Повернув модель, проверьте, чтобы были выделены все грани в левой части головы. Если какие-то грани остались невыделенными, добавьте их к выделению щелчком ПКМ при нажатой клавише Shift. Удалите выделенные грани.
19. Примените к оставшейся половине модификатор *Отражение* (*Mirror*).
20. Переключитесь в режим объектов (*Object Mode*), нажав клавишу Tab и найдите шов на границе между половинками. С помощью кнопок со стрелками на странице свойств *Модификаторы* (*Modifiers*) поменяйте порядок применения модификаторов. Шов должен стать практически незаметен. Подумайте, почему так произошло.



21. Добавьте две небольшие сферы и установите их внутри глазных впадин.



22. Перемещая вершины, исправьте форму ушей так, как показано на рисунке:

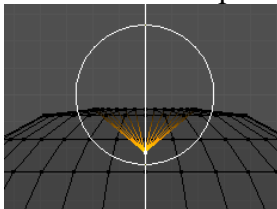
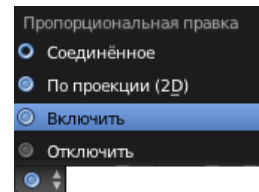




23. Сохраните модель под именем **slon.blend**.

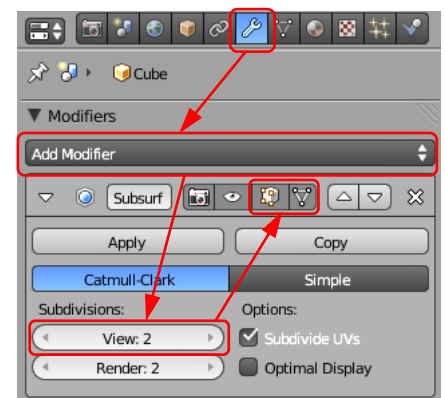
24. Выберите положение камеры и источника света. Выполните рендеринг и сохраните картинку под именем **slon.png**.


### Сеточные модели и модификаторы (яблоко)

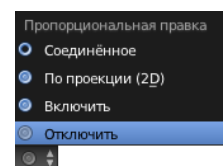
1. Запустите *Blender* и удалите (клавишей *Delete*) куб из автоматически созданной сцены.
2. Нажмите клавиши *Shift+A* и добавьте сферу (*Полисетка – UV-сфера*, *Mesh – UV Sphere*).
3. Перейдите в режим правки (редактирование сеточной модели, *Edit Mode*, клавиша *Tab*) и отмените выделение всех вершин (клавиша *A*).
4. С помощью элемента управления  в нижней части окна включите режим *Пропорциональная правка*  (*Proportional Editing*).
5. Перейдите к виду спереди (клавиша *Num1*) и включите ортографическую проекцию (*Num5*).
6. Включите режим просмотра *Каркас* (*Wireframe*), нажав на клавишу *Z*, и выделите самую верхнюю вершину сферы (ПКМ).
7. Нажмите клавишу *G* (англ. *grab*, режим перемещения), колесиком мыши отрегулируйте область, которую затрагивает пропорциональное редактирование (белая окружность), и сместите вершину вниз. Затем сделайте то же самое с самой нижней вершиной (сместите ее немного вверх).



8. Включите режим просмотра поверхностей *Сплошной* (*Solid*, клавиша *Z*) и посмотрите, что получилось.
9. Перейдите к режиму работы с объектами (*Object Mode*, клавиша *Tab*) и примените модификатор *Подразделение поверхности* (*Subdivision surface*). Увеличьте число деления граней при просмотре до 2 (поле *View*). Для того, чтобы модификатор применялся в режиме редактирования сеточной модели, включите кнопки-выключатели  и .



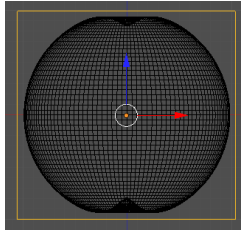
10. Отключите режим пропорционального редактирования, выбрав вариант *Отключить* (*Disable*) в списке элемента управления .



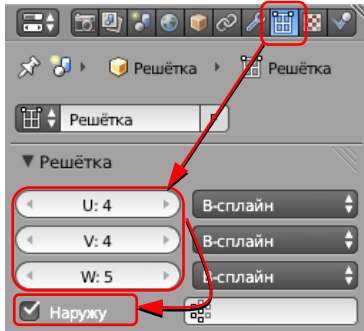
11. Переключитесь в режим выделения объектов (*Object Mode*, клавиша *Tab*) и включите просмотр каркаса (*Wireframe*, клавиша *Z*).

12. Нажмите клавиши *Shift-A* и создайте новую решётку (*Добавить – Решётка*, *Add – Lattice*). Разместите ее так, чтобы она охватывала все яблоко.

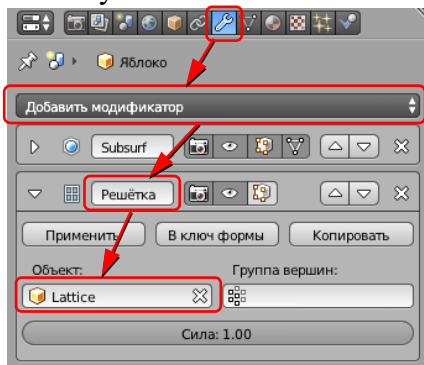




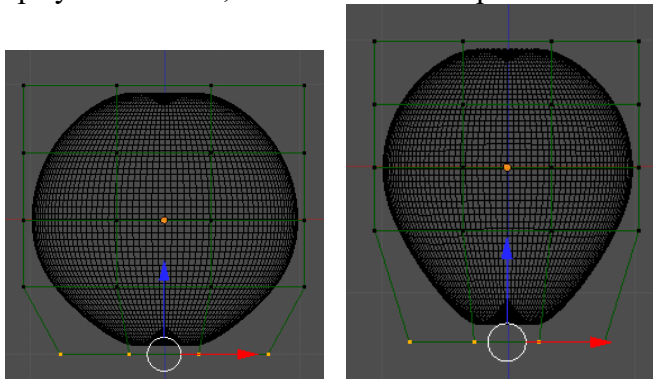
13. На странице свойств *Lattice* увеличьте количество разбиений решетки по осям. Отметьте флажок *Наружа* (*Outside*), чтобы все внутренние вершины и ребра решётки были удалены (остаются только вершины на наружных стенках).



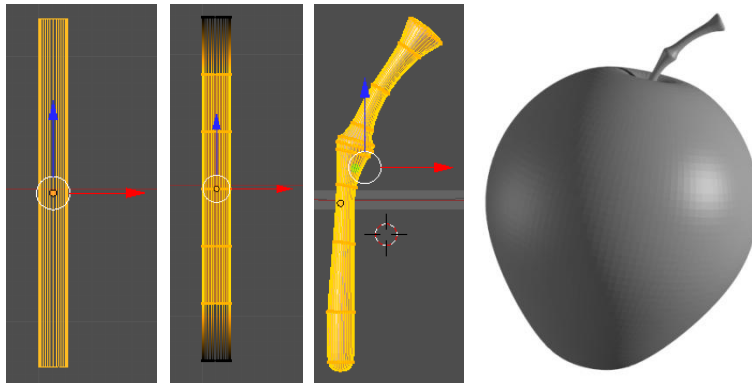
14. Выделите яблоко и примените к нему модификатор *Решётка*. В поле *Object* (объект) на странице свойств нужно выбрать созданную решётку (она по умолчанию имеет имя *Lattice*).



15. Выделите решётку и включите режим редактирования сетки (*Edit Mode*). Перейдите к виду спереди, нажмите клавишу *B* (выделение прямоугольником) и выделите все вершины нижнего ряда, обводя их левой кнопкой мыши.
16. Используя масштабирование (клавиша *S*), уменьшите ширину сетки в этом месте. Затем сместите выделенные узлы немного вниз. Используя эти приемы, измените форму яблока так, как вам больше нравится.



17. Постройте цилиндр и отрегулируйте его размеры примерно по размерам черенка яблока.

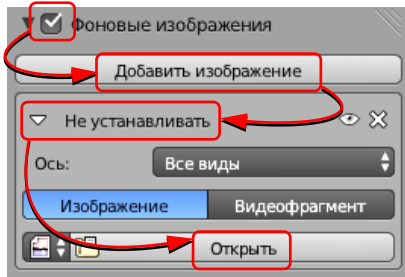



18. Перейдите в режим редактирования и с помощью инструмента *Разрезать петлей со сдвигом (Loop Cut and Slide)* добавьте еще 8-9 горизонтальных ребер по контуру. Используя перемещение, поворот и масштабирование сечений, постройте черенок яблока, как на рисунке.
19. Перейдите в режим работы с объектами и примените к черенку модификатор *Подразделение поверхности (Subdivision surface)*. Установите в поле *View* число делений для вывода на экран – 2.
20. Измените масштаб черенка и установите его на место.
21. Сохраните полученную модель под именем **apple.blend**.
22. Выполните рендеринг полученного яблока и сохраните его в файле **apple.png**.

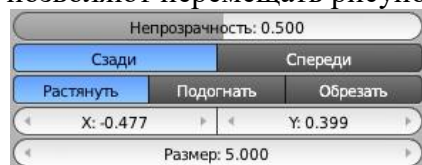
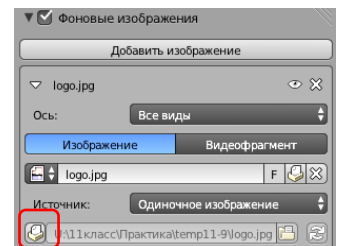
## Практическая работа № 74.

### Кривые

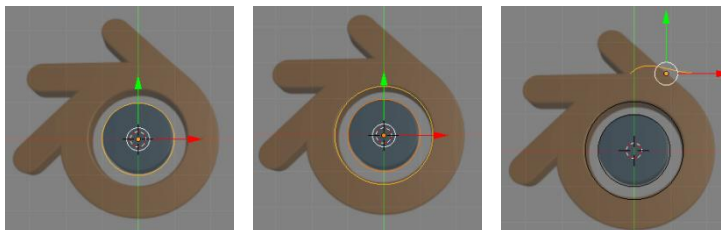
1. Запустите *Blender* и удалите (клавишей *Delete*) куб из автоматически созданной сцены.
2. Нажмите клавишу *N*, чтобы показать панель свойств. Найдите на ней флажок-переключатель *Фоновые изображения* (*Background Images*) и отметьте его стрелкой слева от флажка раскройте группу элементов управления.
3. Щелкните по кнопке *Добавить изображение* (*Add Image*), затем – по белой стрелке слева от надписи *Не устанавливать* (*Not Set*), если эта панель закрыта. Затем щелкните по кнопке *Open* (открыть) и выберите на диске файл **logo.jpg**.



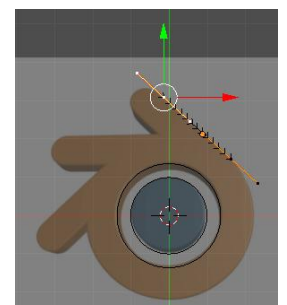
4. Щёлкните по кнопке  слева от имени файла, чтобы рисунок был «упакован» в *blend*-файл. Перейдите к виду сверху (*Num7*) и включите ортографическую проекцию (*Num5*), при этом рисунок должен появиться в рабочей области. Изменяя параметр *Непрозрачность* (*Transparency*), отрегулируйте прозрачность рисунка. Размер изменяется с помощью параметра *Размер* (*Size*), а величины *X* и *Y* позволяют перемещать рисунок в плоскости проекции.



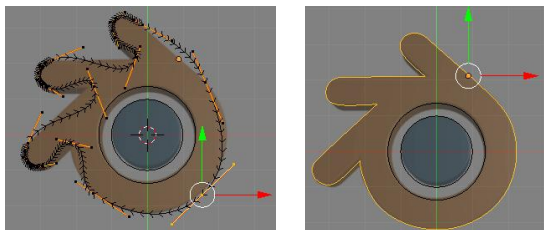
5. Установите курсор в центр сцены (в начало координат), нажав клавиши *Shift+S* и выбрав пункт всплывающего меню *Курсор в центр* (*Cursor to Center*). Нажмите клавиши *Shift+A* и добавьте на сцену окружность (*Добавить – Кривая – Окружность*, *Add – Curve – Circle*).
6. Используя элементы управления фоновым изображением (см. п. 4), переместите рисунок так, чтобы центр внутреннего круга оказался в начале координат. Измените размер окружности так, чтобы она совпала с границей внутреннего круга. Добавьте вторую окружность и измените ее размер так, чтобы она совпала с вырезом «желтой» части логотипа.



7. Добавьте кривую Безье (*Добавить – Кривая – Безье*, *Add – Curve – Bezier*) и переместите ее вправо и вверх. Перейдите в режим редактирования (*Edit Mode*, клавиша *Tab*), выделите оба узла, нажмите клавишу *V* и выберите из меню тип рукоятки *Автоматический* (*Automatic*), чтобы перевести направляющие в автоматический режим. Разместите их так, как показано на рисунке.



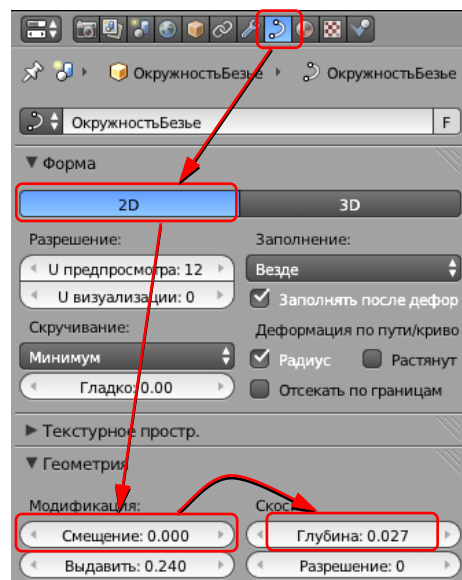
8. Выделите верхнюю вершину. Нажав и не отпуская клавишу *Ctrl*, щелчками ЛКМ создайте новые вершины в ключевых точках контура. Замкните контур, нажав клавиши *Alt+C*.



9. Отрегулируйте положения узлов и направляющих. Для того, чтобы преобразовать гладкий узел в угловой, нужно нажать клавишу *V* и выбрать из меню нужный тип узла.
10. Перейдите в режим объекта (клавиша *Tab*). Для того, чтобы объединить две окружности и внешний контур в один объект, выделите их и нажмите клавиши *Ctrl+J* (англ. *join* – соединить).
11. На странице свойств контуров включите режим *2D* (плоский контур) и увеличьте значения параметров *Выдавить* (*Extrude*, этот параметр регулирует толщину пластины) и *Скос – глубина* (*Bevel – Depth*, фаска).



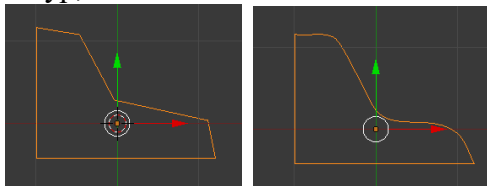
12. Сохраните модель под именем **logo.blend**.
13. Выберите вид с камеры и выполните рендеринг. Сохраните картинку под именем **logo.png**.



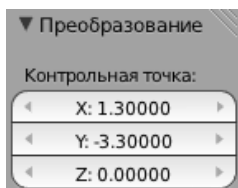
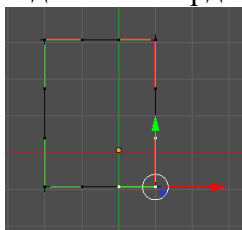
## Практическая работа № 78а (дополнительная).

### Профиль

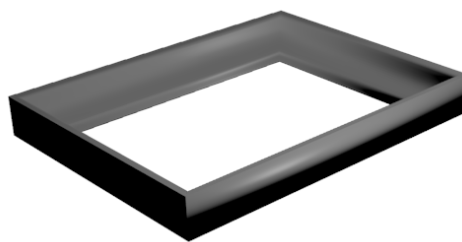
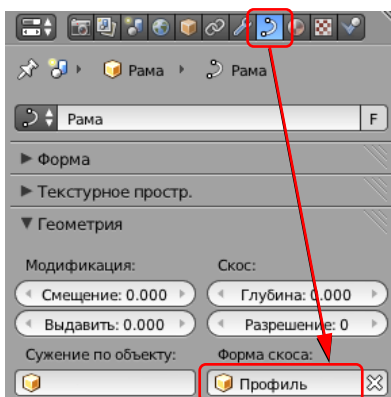
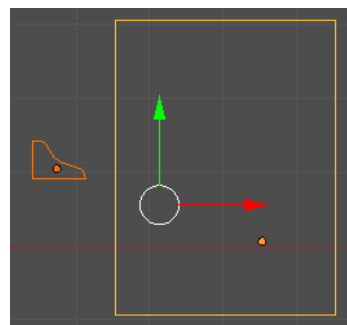
1. Запустите *Blender* и удалите (клавишей *Delete*) куб из автоматически созданной сцены.
2. Перейдите к виду сверху (*Num7*) и добавьте на сцену кривую Безье (*Добавить – Кривая – Безье, Add – Curve – Bezier*).
3. Включите режим редактирования (*Edit Mode*, клавиша *Tab*), выделите оба узла и преобразуйте их в векторные (клавиша *V*).
4. Выделите одну вершину и щелчками левой кнопки мыши при нажатой клавише *Ctrl* добавьте новые узлы так, чтобы получилось сечение рамки для картины. Замкните контур, нажав *Alt+C*.



5. Сделайте три внутренних узла гладкими, как показано на рисунке.
6. Перейдите обратно в режим работы с объектами (*Object Mode*) и добавьте еще одну кривую Безье (*Добавить – Кривая – Безье, Add – Curve – Bezier*).
7. Преобразуйте все узлы в векторные. Добавьте еще два узла и замкните контур так, чтобы получился прямоугольник. Для того, чтобы точно выровнять узлы, можно вводить их координаты на панели преобразований (клавиша *N*).



8. Приведите в соответствие размеры сечения и контура-рамки.
9. Выделите прямоугольный путь, перейдите на страницу свойств кривой (*Object Data*) и выберите в списке *Форма скоса (Bevel Object)* название кривой-профиля.

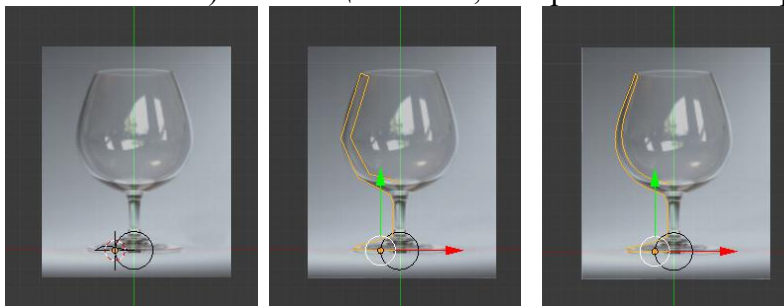


10. Если нужно отрегулируйте размер профиля (возможно, его нужно будет уменьшить).
11. Сохраните модель под именем **frame.blend**.
12. Выполните рендеринг и сохраните картинку под именем **frame.png**.

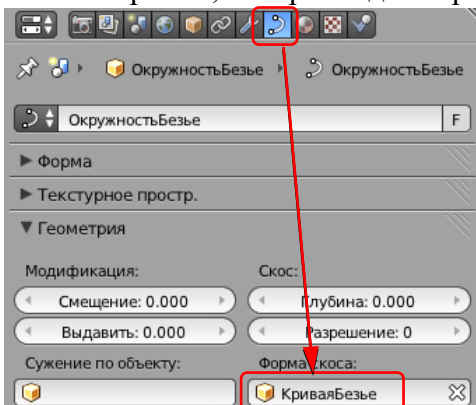
## Практическая работа № 786 (дополнительная).

### Тела вращения

42. Запустите *Blender* и удалите (клавишей *Delete*) куб из автоматически созданной сцены.
43. Переключитесь на вид сверху (*Num7*), в ортографическую проекцию (*Num5*).
44. Добавьте окружность (*Добавить – Кривая – Окружность*, *Add – Curve – Circle*), ее радиус будет по умолчанию равен 1.
45. Поставьте курсор на крайнюю левую точку окружности и добавьте кривую Безье (*Добавить – Кривая – Безье*, *Add – Curve – Bezier*).
46. Добавьте рисунок **wineglass.jpg** на сцену (см. одну из предыдущих работ). Расположите рисунок так, чтобы центральная точка основания ножки была в начале координат.
47. Выделите кривую Безье, которая будет задавать профиль. Перейдите в режим редактирования (*Edit Mode*, клавиша *Tab*). Выделите все узлы и нажмите клавишу *V*, чтобы сделать их векторными, и постройте профиль бокала в виде ломаной линии. Ее начало и конец должны быть на оси *Y* (вертикальной оси, где будет ось вращения).
48. Затем отрегулируйте профиль, преобразуя некоторые узлы из угловых в гладкие (автоматические) с помощью меню, которое появляется при нажатии клавиши *V*.





49. Перейдите в режим работа с объектами (клавиша *Tab*), выделите окружность. На странице свойств объекта (*Object Data*) в поле *Форма скоса (Bevel Object)* выберите название кривой, которая задает профиль.

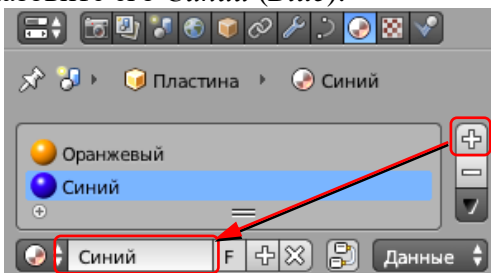
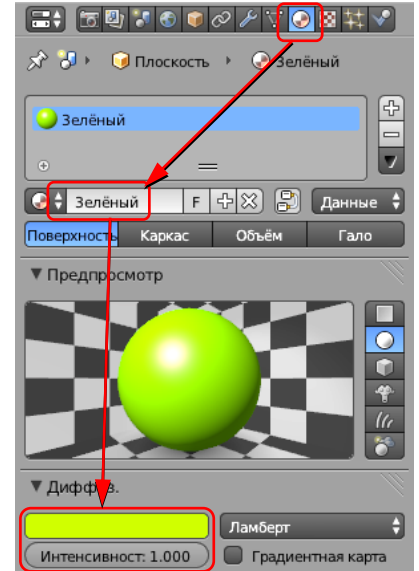


50. Сохраните модель под именем **glass.blend**.
51. Установите свет и камеру так, как вы считаете нужным. Выполните рендеринг и сохраните картинку в файле **glass.png**.

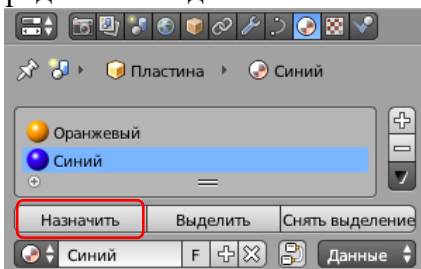
## Практическая работа № 75. Материалы и текстуры

### Материалы

2. Запустите *Blender* и откройте файл **blender.blend**, в котором построен логотип *Blender* и плоскость.
3. С помощью элемента управления  в нижней части окна установите режим просмотра *Текстура (Textured)*.
4. Выделите плоскость, перейдите на страницу свойств  *Материал (Material)* и щелкните по кнопке *Создать (New)*. В поле *Диффуз. (Diffuse, диффузный цвет)* установите зеленый цвет. Дайте материалу имя *Зелёный (Green)*. Увеличьте интенсивность (*Intensity*) до единицы.
5. Выберите объект-логотип и установите для него оранжевый цвет.
6. Сделаем центральную часть логотипа синей. Для этого нужно создать многокомпонентный материал. Щелкните по кнопке со знаком «плюс» справа от списка материалов, а затем по кнопке *Создать (New)*, установите для нового материала синий цвет и назовите его *Синий (Blue)*:






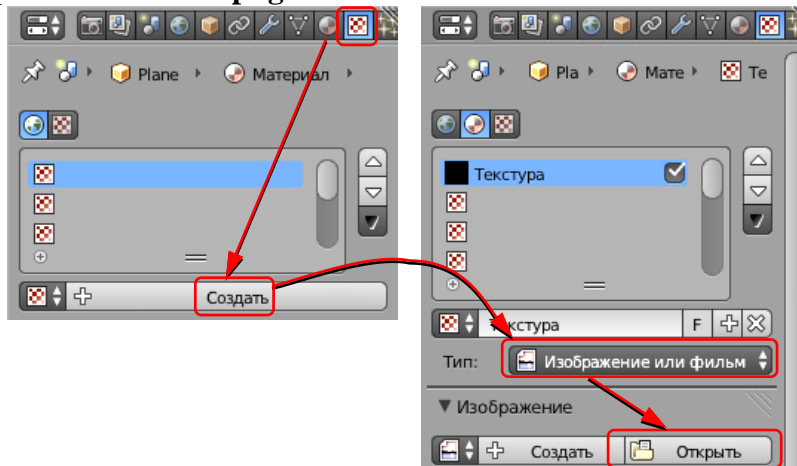
7. Перейдите в режим редактирования (*Edit Mode*, клавиша *Tab*). Выделите все вершины внутренней окружности и щелкните по кнопке *Назначить (Assign)*. Средняя часть должна стать синей.





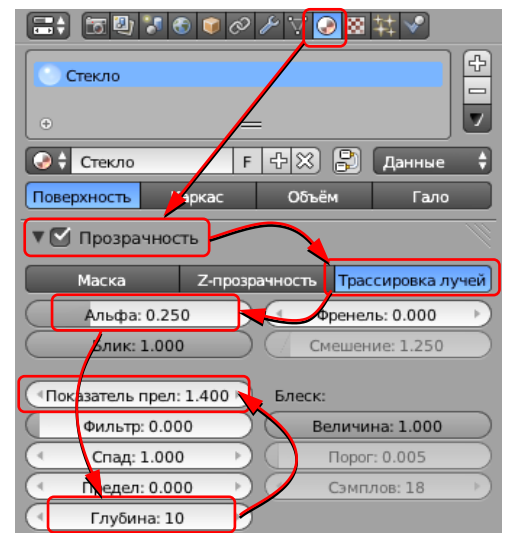
8. Сохраните модель под именем **blender1.blend** (меню *File – Save as...*). Выполните рендеринг и сохраните картинку в файле **blender1.png**.

## Текстуры

29. Запустите *Blender* и откройте файл **wineglass.blend**, в котором на сцену помещены бокал и две плоскости.
30. С помощью элемента управления  в нижней части окна установите режим просмотра *Текстура (Textured)*.
31. Выделите бокал и перейдите в режим редактирования (*Edit Mode*). Включите режим работы с гранями (*Ctrl+Tab* или ) и выделите все грани (клавиша *A*). Затем нажмите клавиши *Ctrl+N*, при этом программа заново пересчитывает нормали к граням так, чтобы они были направлены наружу.
32. Выделите нижнюю плоскость, создайте новый материал (см. предыдущую работу) и перейдите на страницу свойств  *Текстура (Texture)*. Создайте новую текстуру (кнопка *Создать, New*). В списке *Type* (тип текстуры) выберите вариант *Image or Movie* (рисунок или видео). Затем щелкните по кнопке *Open* (открыть) и выберите файл **black-white.png**.






33. Щёлкните по кнопке  слева от имени файла, чтобы рисунок был «упакован» в *blend*-файл.
34. Аналогично назначьте вертикальной плоскости новый материал с текстурой из файла **red-white.png**.
35. Выделите бокал, перейдите на страницу свойств  *Материал (Material)* и щелкните по кнопке *Создать (New)*. Назовите его *Стекло (Glass)*.
36. Включите флажок *Прозрачность (Transparency)*, режим *Трассировка лучей (Raytrace)* и установите параметр *Альфа (Alpha, непрозрачность)* равным 0,25. Выполните рендеринг (это может занять около минуты).
37. Увеличьте значение параметра *Глубина (Depth)* до 10 и выполните рендеринг. Что изменилось?
38. Увеличьте значение параметра *Показатель преломления (IOR, Index of Refraction)* до 1.4 и выполните рендеринг. Что изменилось?
39. Сохраните модель под именем **wineglass1.blend**.
40. Сохраните картинку под именем **wineglass.png**.

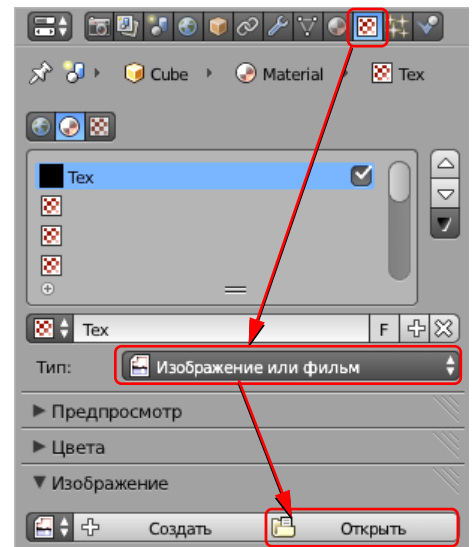






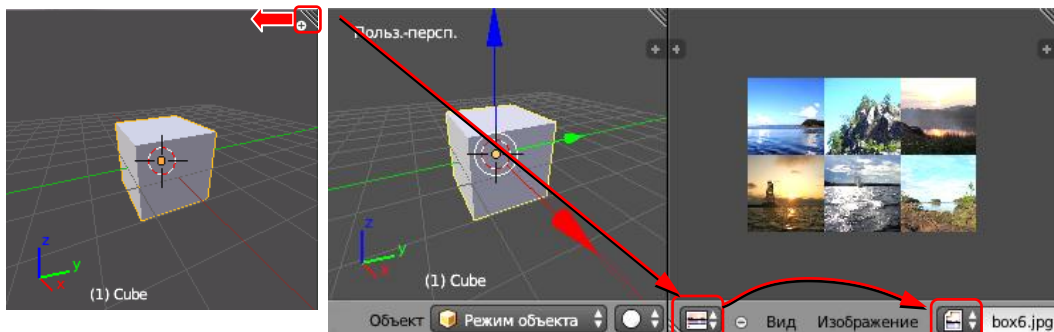
## Практическая работа (дополнительная)




### UV-развёртка (куб)

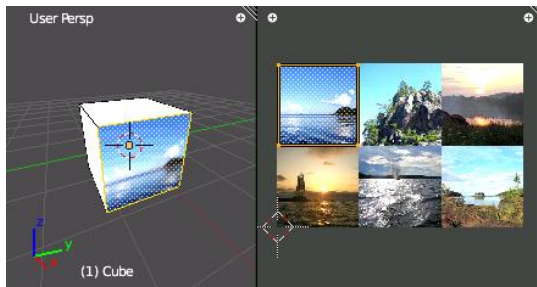
1. Запустите *Blender* и выделите куб, созданный автоматически.
2. Перейдите на страницу свойств  *Материал (Material)* и убедитесь, что для него автоматически создан новый материал.
3. Перейдите на страницу свойств  *Текстура (Texture)*. В поле *Тип (Type)* выберите вариант *Изображение или видео (Image or Movie)*. Затем щёлкните по кнопке *Открыть (Open)* и выберите файл **box6.jpg**. Щёлкните по кнопке  слева от имени файла, чтобы рисунок был «упакован» в blend-файл.
4. На панели *Mapping* (отображение) в поле *Координаты (Coordinates)* выберите вариант *UV (UV-развертка)*:



5. Наведите курсор мыши на правый верхний угол рабочего поля, курсор должен превратиться в белый крестик. Нажмите ЛКМ и «вытащите» второе окно. В этом окне щёлкните по кнопке  и выберите тип окна *Редактор UV/изображений (UV/Image Editor)*. Затем щёлкните по кнопке  и выберите рисунок **box6.jpg**. Вращая колёсико мыши, уменьшите его размер так, чтобы весь рисунок был виден в окне.



6. Переведите курсор в окно трехмерной проекции и перейдите в режим редактирования (*Edit Mode*, клавиша *Tab*). Включите режим работы с гранями (*Ctrl+Tab* или ), отмените выделение всех граней (клавиша *A*) и выделите одну грань (ПКМ).
7. Из меню, расположенного под рабочей областью окна трехмерной проекции, выберите пункт *Полисетка – UV-развертка – Развернуть (Mesh – UV Unwrap – Unwrap)*.
8. С помощью элемента управления  в нижней части окна установите режим просмотра  *Текстура (Textured)*.
9. Перейдите к окну *Редактор UV/изображений (UV/Image Editor)*. Используя те же методы, которые вы применяли для работы с сеточными моделями, настройте рамку так, чтобы на выбранной грани был один рисунок из шести. Например, можно выделять вершины с помощью ПКМ, масштабировать их (клавиша *S*), перемещать (клавиша *G*), вращать (клавиша *R*).



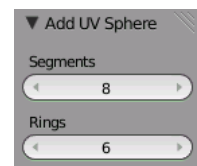
10. Аналогично разместите на остальных гранях оставшиеся 5 картинок.
11. Сохраните модель под именем **box6.blend**.
12. Установите источник света так, чтобы он освещал три грани, видимые с камеры. Выполните рендеринг и сохраните картинку под именем **box6-1.png**.
13. Переставьте источник света и камеру так, чтобы видеть оставшиеся три грани. Выполните рендеринг и сохраните картинку под именем **box6-2.png**.




## Практическая работа (дополнительная)

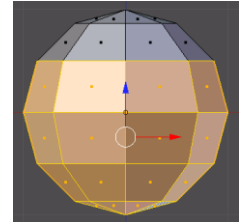
### UV-развёртка (зонтик)

1. Запустите *Blender* и удалите куб, созданный автоматически.
2. Создайте сферу *Добавить – Полисетка – UV-сфера (Add – Mesh – UV Sphere)* и измените (внизу слева от рабочего поля) число сегментов и колец: сделайте 8 сегментов (*Segments*) и 6 колец (*Rings*).

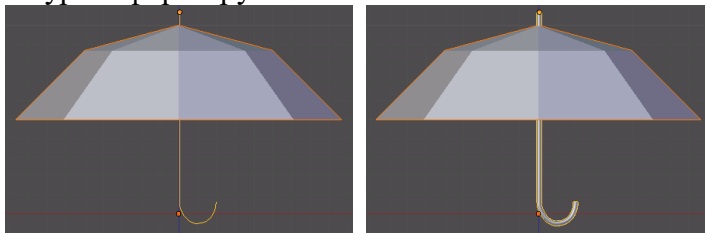


3. Перейдите в режим редактирования (*Edit Mode*, клавиша *Tab*). Включите режим работы с гранями (*Ctrl+Tab* или ) , отмените выделение всех граней (клавиша *A*).

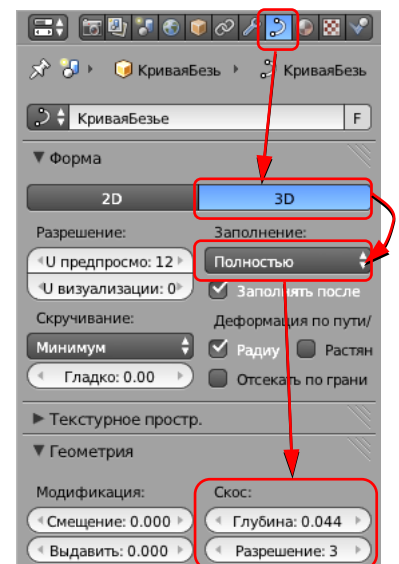
4. Проверьте, чтобы кнопка , которая ограничивает выделение только видимыми гранями (запрещает выделение на обратной стороне фигуры) была отключена. Перейдите к виду спереди (*Num1*). Нажмите клавишу *B* (выделение прямоугольником) и выделите 4 нижних кольца граней (см. рисунок). Удалите их (*Delete – Грань*).






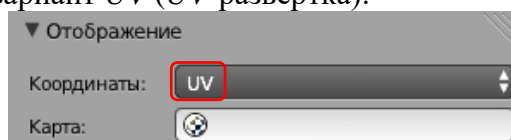
5. Перейдите в режим работы с объектами (*Object Mode*, клавиша *Tab*). Добавьте кривую Безье (*Добавить – Кривая – Безье, Add – Curve – Bezier*). Поверните её на 90 градусов вокруг оси X (клавиша *R*, затем *X*, затем набрать 90 на клавиатуре, затем *Enter*).
6. Перейдите в режим редактирования (*Tab*) и постройте контур по форме ручки зонтика:






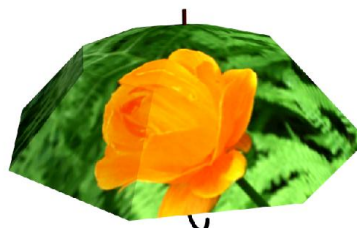
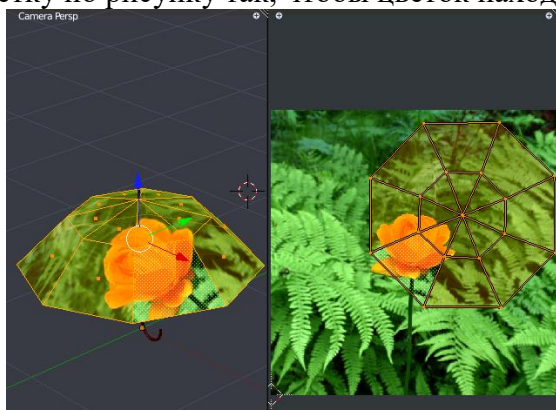
7. Перейдите на страницу свойств кривой (*Object Data*) включите свойство *3D* (трехмерная кривая), выберите в списке *Заполнение (Fill)* вариант *Полностью (Full)* флажки *Front* (передняя сторона) и *Back* (задняя сторона). Затем измените параметры блока *Bevel* (фаска): увеличьте глубину (*Depth*), она определяют толщину ручки, и разрешение (*Resolution*) до 3 (гладкость поверхности).



8. Перейдите на страницу свойств  *Материал (Material)* и установите для ручки материал коричневого цвета.
9. Выделите купол зонтика, создайте для него новый материал (см. предыдущую работу) и уменьшите интенсивность (*Intensity*) зеркального отражения до нуля (панель *Specular*).
10. Перейдите на страницу свойств  *Текстура (Texture)*. Создайте новую текстуру (кнопка *Создать, New*). В списке *Тип (Type)* выберите вариант *Изображение или видео (Image or Movie)*. Затем щелкните по кнопке *Открыть (Open)* и выберите файл **flower.jpg**. Щёлкните по кнопке  слева от имени файла, чтобы рисунок был «упакован» в blend-файл.
11. На панели *Отображение (Mapping)* в поле *координаты (Coordinates)* выберите вариант *UV (UV-развертка)*:




12. Так же, как и в предыдущей работе, «вытащите» второе окно и выберите тип окна *Редактор UV/изображений (UV/Image Editor)*.
14. Перейдите в режим редактирования (*Edit Mode*). Выделите все грани. Из нижнего меню *Полисетка (Mesh)* выберите пункт *UV-развертка – Развернуть (UV Unwrap – Unwrap)*.
15. С помощью элемента управления  в нижней части окна установите режим просмотра  *Текстура (Textured)*.
16. Нажмите клавишу *Num0*, чтобы перейти к виду с камеры.
17. Перейдите к окну *Редактор UV/изображений (UV/Image Editor)*. Щёлкните по кнопке  и выберите рисунок **flower.jpg**. Уменьшите сетку, нажав клавишу *S* (scale, масштабировать). Затем нажмите клавишу *G* (*grab*, схватить) и переместить сетку по рисунку так, чтобы цветок находился сбоку.

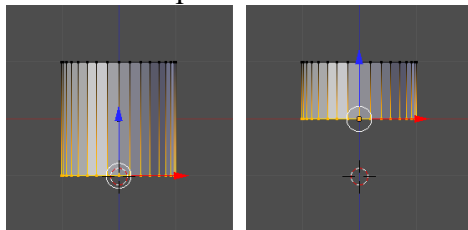






18. Сохраните модель под именем **umbrella.blend**.
19. Выполните рендеринг (F12). Сохраните картинку под именем **umbrella.png**.

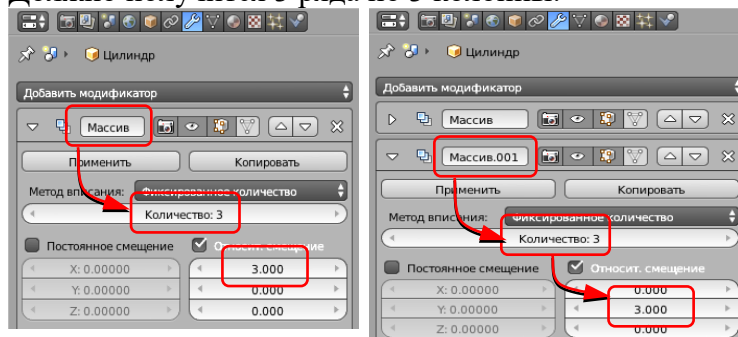
## Практическая работа № 76.

### Рендеринг

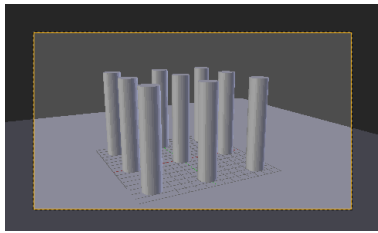
1. Запустите *Blender* и удалите куб, созданный автоматически.
2. Добавьте плоскость *Добавить – Полисетка – Плоскость* (*Add – Mesh – Plane*) и увеличьте ее размеры в 10 раз (нажмите клавишу *S*, наберите на клавиатуре «10» и нажмите *Enter*).
3. Временно скройте плоскость, нажав на значок с изображением глаза на панели *Структура проекта* (*Outliner*).
4. Добавьте на сцену цилиндр (*Добавить – Полисетка – Цилиндр*, *Add – Mesh – Cylinder*). Включите режим редактирования (*Edit Mode*, клавиша *Tab*) и перейдите к виду спереди (*Num1*) и включите ортографическую проекцию (*Num5*).
5. Проверьте, чтобы кнопка , которая ограничивает выделение только видимыми гранями (запрещает выделение на обратной стороне фигуры) была отключена.
6. Нажмите клавишу *B* (выделение прямоугольником) и обведите левой кнопкой мыши все нижние вершины.



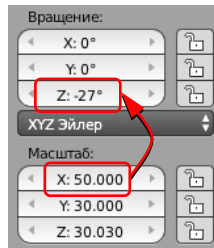
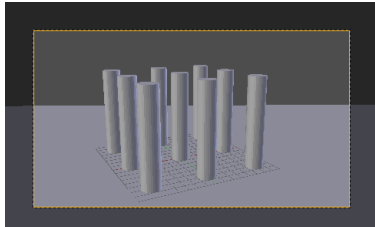
7. Включите кнопку  (привязка) под окном трёхмерной проекции и выберите справа от нее вариант  (привязка к координатной сетке). Переместите выделенные вершины вверх так, чтобы они оказались на уровне  $Z=0$ . Затем отключите привязку, щелкнув по кнопке .
8. Вернитесь в режим работы с объектами (*Object Mode*, клавиша *Tab*) и увеличьте высоту цилиндра: нажмите клавишу *S* (масштабирование), затем клавишу *Z* (ограничить действие осью *Z*), наберите на клавиатуре «10» и нажмите *Enter*.
9. Перейдите на страницу свойств  *Модификаторы* (*Modifiers*), щелкните по кнопке *Добавить модификатор* (*Add Modifier*) и выберите модификатор *Массив* (*Array*). Установите в параметрах модификатора *Количество* (*Count*) равное 3, и смещение 3 по оси *X*. Должно получиться 3 колонны.
10. Добавьте еще один модификатор *Массив* (*Array*), теперь со смещением 3 по оси *Y*. Должно получиться 3 ряда по 3 колонны.



11. Перейдите к виду сверху (*Num7*). Включите показ плоскости и поставьте колонны в центр плоскости.
12. Выделите плоскость и увеличьте ее в 3 раза.
13. Установите камеру так, чтобы вид с нее был примерно таким, как на рисунке:



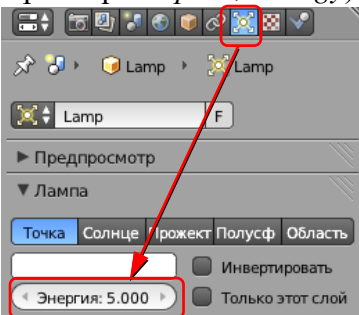
14. Выведите на экран панель преобразований (клавиша *N*), увеличьте *Масштаб* (*Scale*) по оси *X* до 50 и поверните плоскость вокруг оси *Z* так, чтобы граница плоскости стала параллельной горизонту:



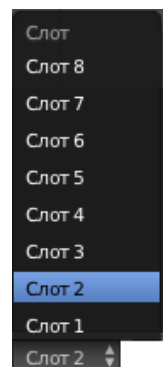
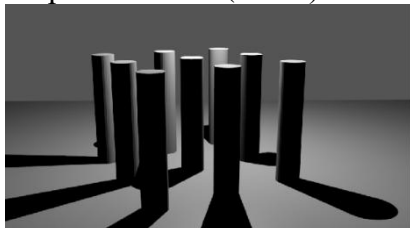
15. Как и в предыдущих работах, «вытащите» второе окно и выберите тип окна *Редактор UV/изображений* (*UV/Image Editor*). В этом окне будет показываться результат рендеринга.  
 16. Выполните рендеринг (F12).



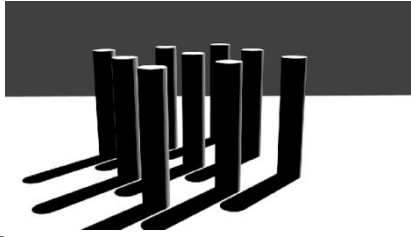
17. Переместите источник света вверх так, чтобы он располагался над колоннами. Перейдите на страницу свойств лампы (*Object Data*) и увеличьте мощность лампы (параметр *Энергия*, *Energy*) до 5:



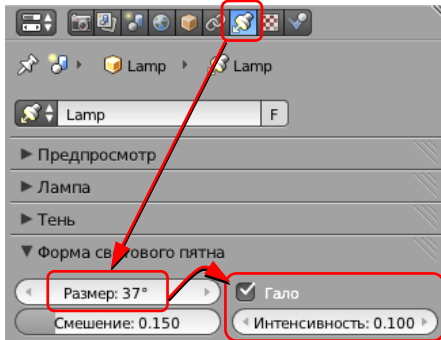
18. В нижней части окна *Редактор UV/изображений* (*UV/Image Editor*), выберите *Слот 2* (*Slot 2*). Выполните рендеринг:



19. Измените тип лампы на *Солнце* (*Sun*), перейдите к слоту 3 и выполните рендеринг. Обратите внимание на расположение теней.



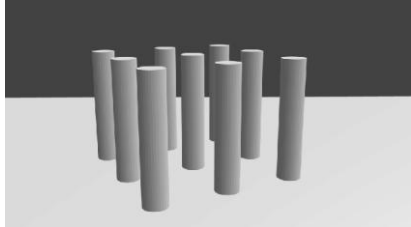
20. Измените положение источника и направление света. Выполните рендеринг.
21. Измените тип лампы на *Прожектор (Spot)*. Установите угол конуса 37 градусов (*Размер, Size*), включите эффект «Гало» (*Halo*) и установите его *Интенсивность (Intensity)* равной 0,1.



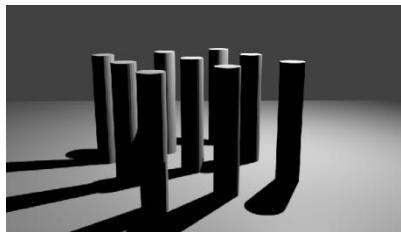
22. Используя перемещение и вращение на видах спереди и сбоку, настройте лампу так, чтобы она светила сверху на колонны. Перейдите к слоту 4 и выполните рендеринг:



23. Измените тип лампы на *Полусфера (Hemi)*, уменьшите ее энергию (*Energy*) до 0,5. Перейдите к слоту 5 и выполните рендеринг.




24. Измените тип лампы на *Область (Area)*, перейдите к слоту 6 и выполните рендеринг.

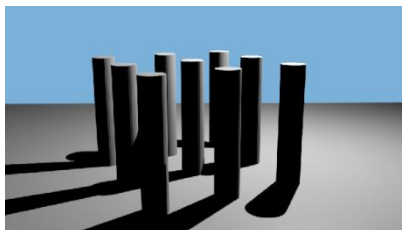
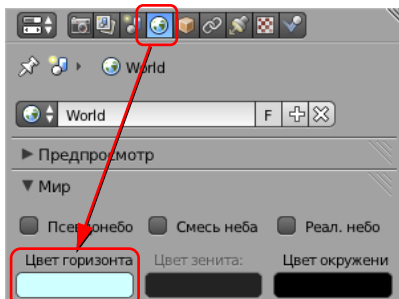


6.

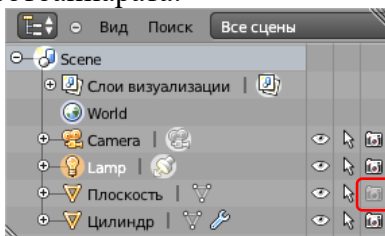
## Практическая работа (дополнительная)

### Рендеринг

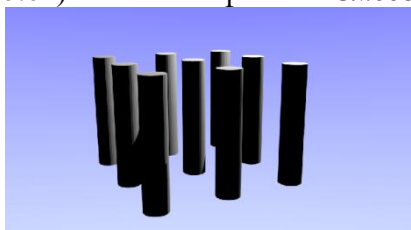
1. Загрузите в *Blender* сцену, построенную в предыдущей работе.
2. Как и в предыдущей работе, сохраняйте все изображения в разных слотах.
3. Перейдите на страницу свойств  *Мир (World)*. Установите голубой цвет в поле *Цвет горизонта (Horizon Color)*. Выполните рендеринг, посмотрите, что изменилось.



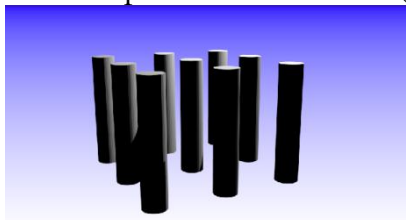
4. Сделайте плоскость невидимой при рендеринге, щелкнув на значке с изображением фотоаппарата:



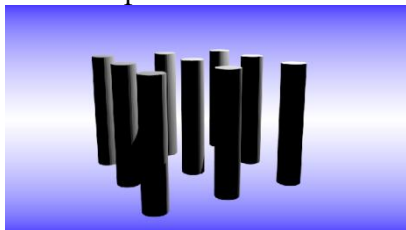
5. Установите белый *Цвет горизонта (Horizon Color)* и синий *Цвет зенита (Zenith Color)*. Отметьте флажок *Смесь неба (Blend Sky)* и выполните рендеринг.



6. Отметьте флажок *Псевдонебо (Paper Sky)* и выполните рендеринг.

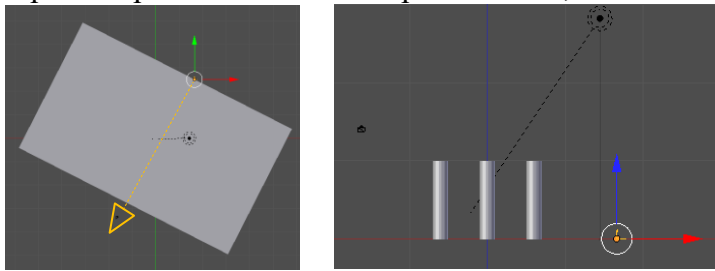


7. Отметьте флажок *Реальное небо (Real Sky)* и выполните рендеринг.

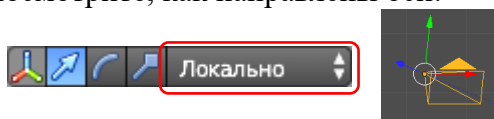
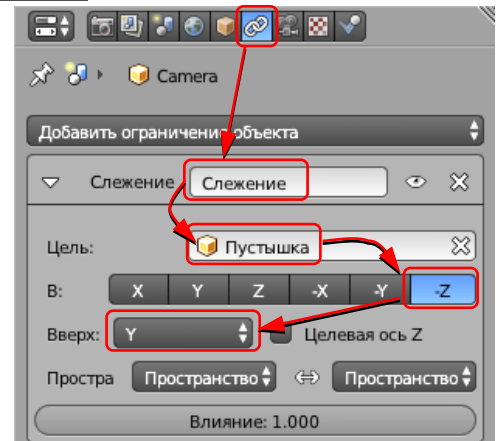




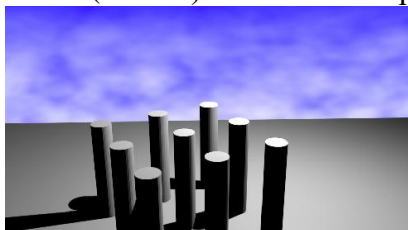
8. Сделайте плоскость видимой при рендеринге, щёлкнув по кнопке с изображением фотоаппарата в окне *Структура проекта (Outliner)*.
9. Перейдите к виду сверху (*Num7*) и добавьте пустой объект (*Добавить – Пустышка – Куб, Add – Empty – Cube*) напротив камеры на самый край плоскости. На виде спереди переместите его по вертикали так, чтобы его координата *Z* была равна нулю.




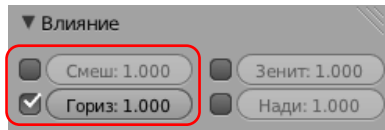
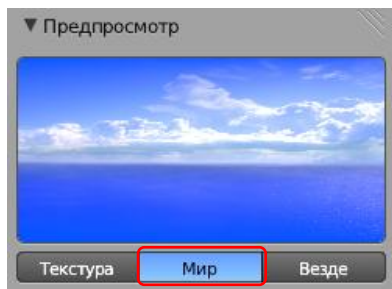
10. Установите для камеры ограничение: она должна при любых перемещениях быть направлена на только что созданный пустой объект. Выделите камеру, перейдите на страницу свойств *Ограничения (Constraints)*. Щёлкните по кнопке *Добавить ограничение объекта (Add Constraint)*, выберите ограничение *Слежение (Track To)* и выберите в поле *Цель (Target)* пустой объект *Пустышка*.
11. В окне трёхмерной проекции переключитесь на локальную систему координат (*Local*) и посмотрите, как направлены оси:




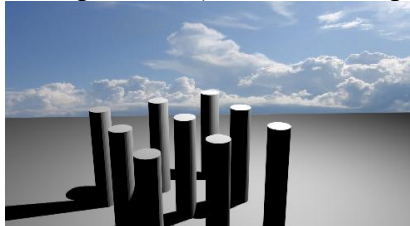
12. На панели *Constraints* выберите направление «-Z», а в поле *Вверх (Up)* – «Y». Это значит, что локальная ось *Z* камеры будет направлена в сторону, противоположную объекту, а ее ось *Y* – вверх.
13. Вернитесь к глобальным осям координат (*Global*) координат и переместите камеру немного вверх. Выполните рендеринг.
14. В окне трёхмерной проекции отмените выделение всех объектов (клавиша *A*). Перейдите на страницу свойств *Мир (World)*, а затем – на страницу *Текстура (Texture)*. Добавьте новую текстуру (кнопка *Создать, New*), оставьте вариант *Облака (Clouds)* и выполните рендеринг.




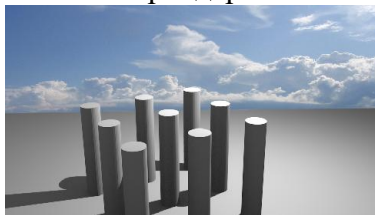
15. В списке *Тип (Type)* выберите вариант *Изображение или видео (Image or Movie)*. Затем щёлкните по кнопке *Открыть (Open)* и выберите файл **sky.jpg**. Щёлкните по кнопке  слева от имени файла, чтобы рисунок был «упакован» в blend-файл.
16. На панели *Предпросмотр (Preview)* включите режим *Мир (World)*, просмотр фона при рендеринге). Затем на панели *Влияние (Influence)* отключите флажок *Смешение (Blend)* и включите *Горизонт (Horizon)*.



17. Перейдите на страницу свойств  *Мир (World)*. Чтобы отключить влияние цветов, установленных на этой странице, отключите флажок *Смесь неба (Blend Sky)*. Также отключите флажок *Реальное небо (Real Sky)*, чтобы фоновый рисунок не искажался (не искривлялся). Выполните рендеринг.



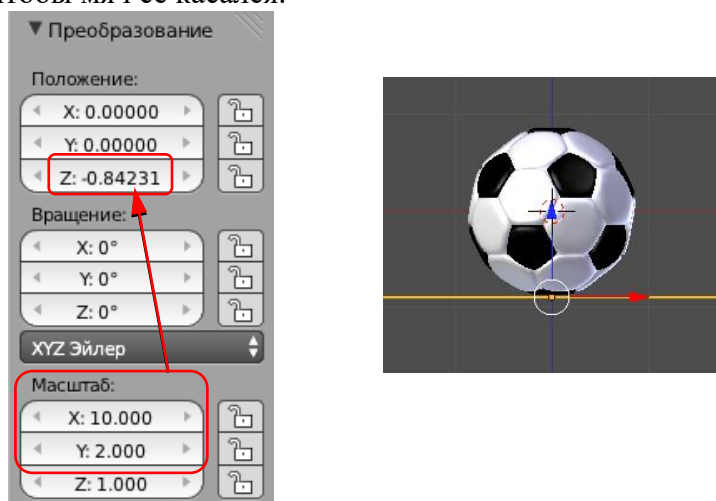
18. На странице свойств  *Мир (World)* включите флажок *Ambient Occlusion* (подсветка теней) и установите *Множитель (Factor, мощность подсветки)* равный 0,1. Выполните рендеринг.



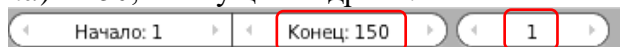
## Практическая работа № 77.

### Анимация

7. Загрузите в *Blender* файл **ball.blend** (меню *Файл – Открыть, File – Open*) и сохраните его под именем **ball-ready.blend** (меню *Файл – Сохранить как, File – Save As*).
8. Перейдите к виду сверху и добавьте на сцену плоскость (*Добавить – Полисетка - Плоскость, Add – Mesh – Plane*). Нажмите клавишу *N*, чтобы вывести на экран панель преобразований. Увеличьте размеры площадки в 10 раз по оси *X* и в 2 раза по оси *Y* (в группе *Масштаб, Scale*). Затем перейдите к виду спереди (*Num1*), включите ортографическую проекцию (*Num5*) и переместите плоскость немного вниз так, чтобы мяч её касался.

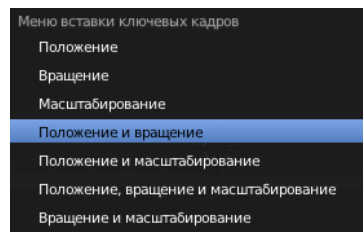
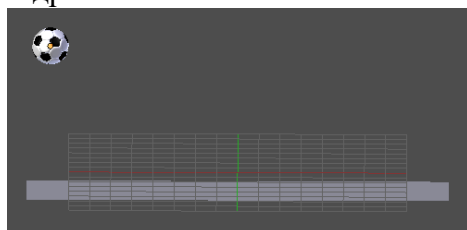


9. В нижней части окна под временной шкалой установите последний кадр (*Конец, End*) – 150, и текущий кадр – 1.

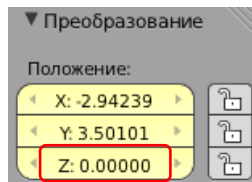


10. Немного наклоните вид «на себя», чтобы видеть плоскость, и установите мяч на некоторой высоте на левой границе плоскости (см. рисунок). Затем нажмите клавишу *I* (вставить ключевой кадр) и выберите из списка вариант *Положение и вращение (LocRot, Location – Rotation)*.

кадр 1



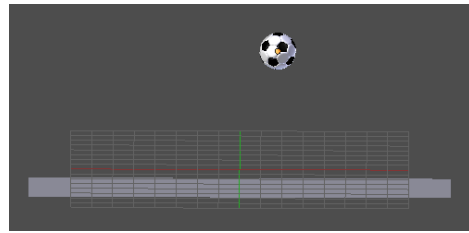
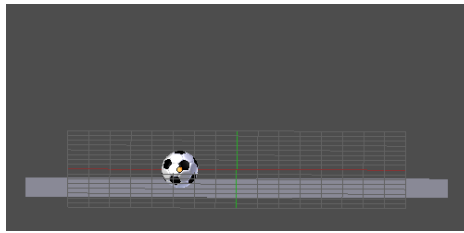
11. Сделайте текущим кадр 50 и установите мяч на плоскость, сдвинув его вправо. Поскольку мяч касается плоскости при *Z*-координате, равной нулю, удобно выставить это значение вручную на панели преобразований (клавиша *N*).



12. Вставьте новый ключевой кадр типа *Положение и вращение (LocRot)*. Обратите внимание, что на панели преобразований координаты (*Location*) и углы поворота (*Rotation*) выделены желтым фоном, это значит, что для них установлена анимация.

кадр 50

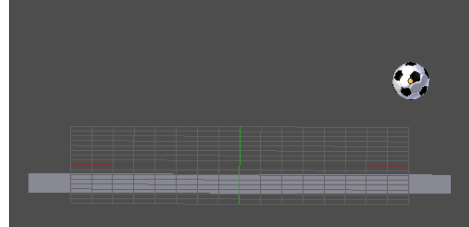
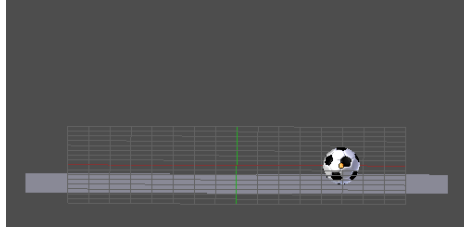
кадр 90



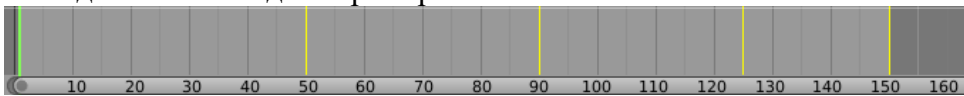
Аналогично добавьте ключевые кадры 90, 125 и 150.

кадр 125

кадр 150



После этого временная шкала (*Timeline*) с расставленными ключевыми кадрами должна выглядеть примерно так:

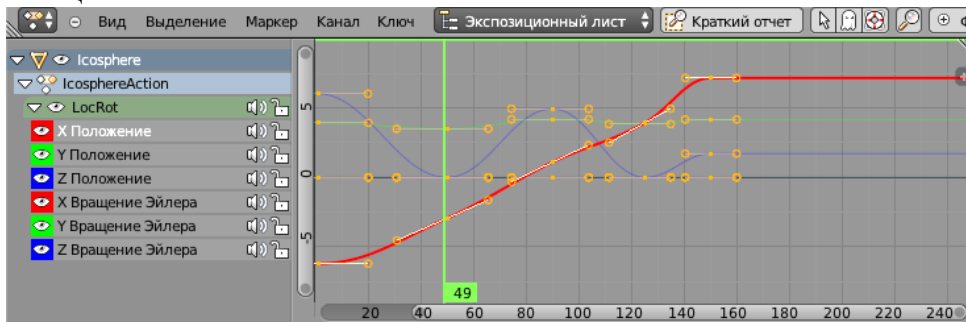
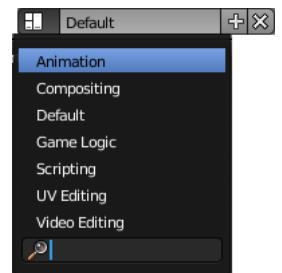


13. Запустите анимацию, нажав клавиши *Alt+A*, и посмотрите, что получилось.

Остановить анимацию можно с помощью клавиши *Esc*. Что не так?

14. В верхней части экрана найдите список, позволяющий выбрать стандартное расположение окон на экране, и выберите в нём вариант *Animation* (анимация).

15. Переведите курсор в окно Редактора F-кривых (*F-Curve Editor*). Щелчком мыши по треугольнику слева от типа ключевого кадра (*LocRot*) раскройте список всех кривых анимации.

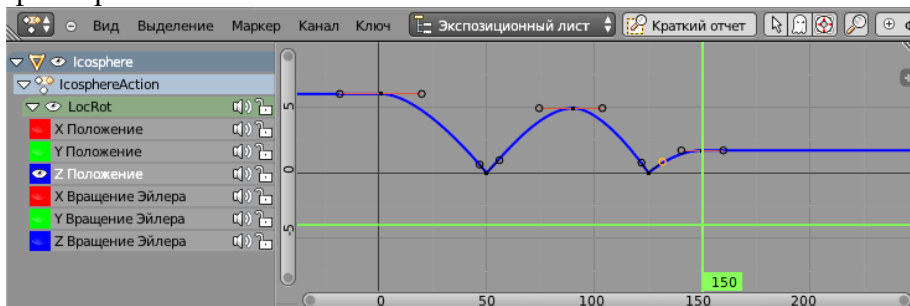


16. Раскройте окно редактора кривых на полный экран, нажав *Ctrl* и клавишу «стрелка вверх».

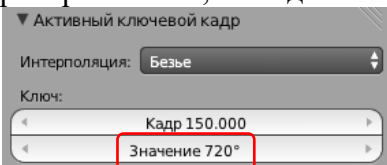
17. Щёлкая по значкам с изображением глаза, отключите все кривые, кроме *X Положение* (*X Location*). Оставьте в ней только два узла (в кадрах 1 и 150). Для удаления узла нужно выделить его и нажать клавишу *Delete*. Сделайте эти узлы векторными (выделить, нажать клавишу *V*, выбрать в меню *Векторный*, *Vector*).



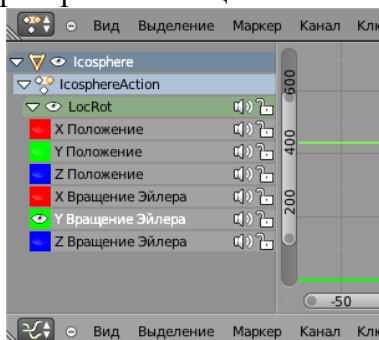
18. Отключите все кривые, кроме *Z Location* (*Z*-координата). Сделайте узлы, в которых мяч касается плоскости, свободными (клавиша *V*, выбрать из списка *Свободный*, *Free*) и измените направляющие так, чтобы кривая состояла из кусков парабол. Для перемещения выделенного узла или рукоятки нужно нажать клавишу *G* (*grab*, схватить) мышкой передвинуть узел в нужное положение и нажать ЛКМ, чтобы зафиксировать его.



19. Вернитесь к нормальному (не полноэкранному) режиму, нажав на клавиши *Ctrl*+«стрелка вверх». Проверьте анимацию, нажав клавиши *Alt*+*A*.
20. Теперь осталось сделать так, чтобы мяч вращался. Зададим вращение только вокруг оси *Y* (вращение вокруг других осей вы сможете сделать самостоятельно). Оставьте в окне редактора кривых только кривую *Y Вращение Эйлера* (*Y Euler Rotation*) и перейдите в полноэкранный режим.
21. Оставьте на кривой только два узла в кадрах 1 и 150. Пусть за время анимации мяч поворачивается на 720 градусов, то есть делает два оборота вокруг своей оси. Выделите узел в кадре 150, нажмите клавишу *N*, чтобы вызвать панель преобразований, и введите *Y*-координату узла 720°.

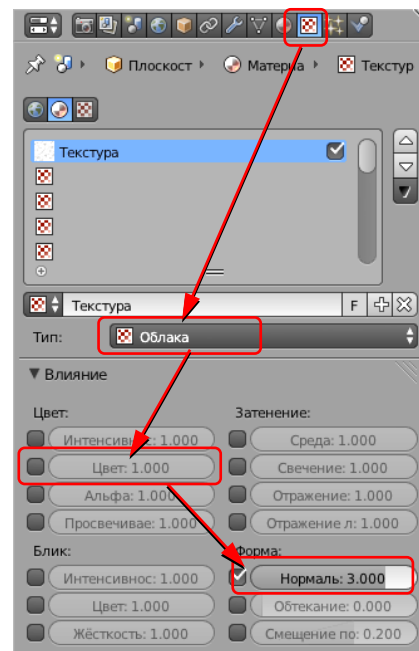
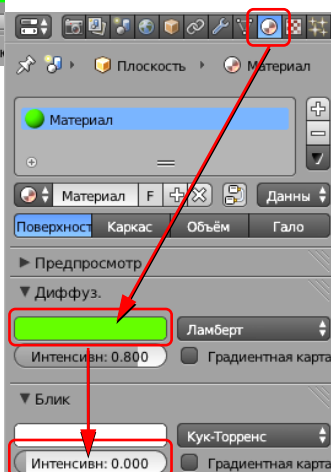



22. Нажмите клавишу *Home*, чтобы автоматически подобрать масштаб графика (его можно также регулировать колесиком мыши). Сделайте оба узла векторными и проверьте анимацию.





23. Перейдите к виду с камеры (*Num0*) и настройте его так, чтобы видеть мяч во время всей анимации (используйте «режим полёта», клавиши *Shift*+*F*).

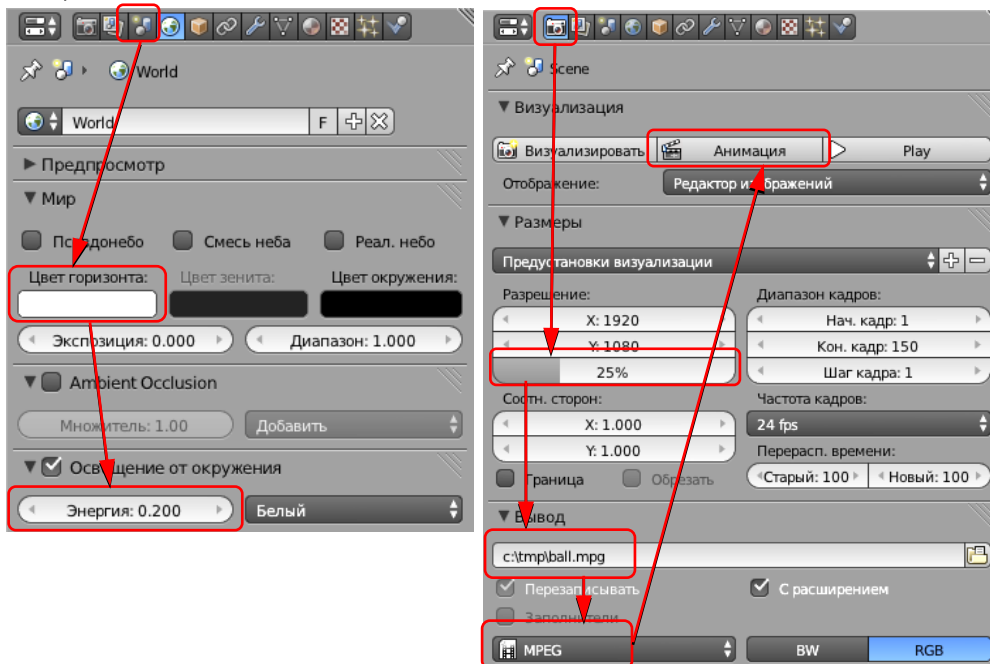
24. Теперь «покрасим» плоскость в зелёный цвет и сделаем ее шероховатой. Перейдите на страницу




свойств  *Материал (Material)* и щелкните по кнопке *Создать (New)*. В поле *Диффузный (Diffuse)* установите зеленый цвет. Уменьшите интенсивность бликов до нуля: поле *Интенсивность, Intensity* на панели *Блик (Specular)*.

25. Перейдите на страницу свойств  *Текстура (Texture)*. Создайте новую текстуру (кнопка *Создать, New*). В списке *Тип (Type)* оставьте вариант по умолчанию *Clouds* (облака). На панели *Influence* (влияние) снимите флажок *Цвет (Color)* и включите флажок *Нормаль (Normal)*, увеличив значение в этом поле до 3. Выполните рендеринг и посмотрите результат.

26. Перейдите на страницу свойств  *World* (окружение). Установите белый *Цвет горизонта (Horizon Color)*, включите внешнее освещение (флажок *Освещение от окружения, Environment Lighting*) и уменьшите его интенсивность (*Энергия, Energy*) до 0,2.





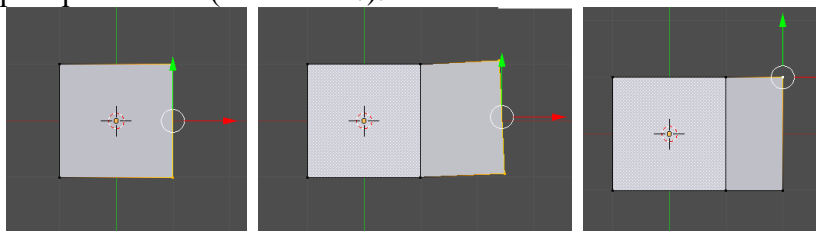
27. Перейдите на страницу свойств  *Рендеринг (Render)*. Уменьшите размер до 25% (чтобы ускорить рендеринг). Установите формат выходного файла MPEG, задайте имя файла (с расширением **\*.mpg**). Затем щелкните по кнопке *Animation* (анимация) и ждите, когда ролик будет готов. Это может занять несколько минут даже на достаточно мощном компьютере.

28. Найдите получившийся видеофайл на диске и просмотрите его в проигрывателе.

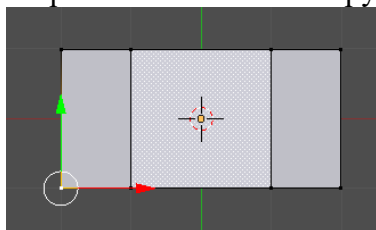
## Практическая работа (дополнительная)

### Анимация. Ключевые формы



32. Запустите *Blender*, выделите куб, созданный по умолчанию. Перейдите в режим редактирования (*Edit Mode*, клавиша *Tab*). Включите режим работы с гранями (*Ctrl+Tab* или ) , выделите верхнюю грань и удалите её.
33. Перейдите к виду сверху (*Num7*) и включите ортогографическую проекцию (*Num5*). Переключитесь в режим работы с ребрами (*Ctrl+Tab* или ) . Выделите одно из верхних ребер, и постройте новое параллельное ребро щелчком ЛКМ при нажатой клавише *Ctrl*. Переключитесь в режим работы с вершинами и установите вершины точно в узлы координатной сетки. Для этого можно использовать панель преобразований (клавиша *N*).

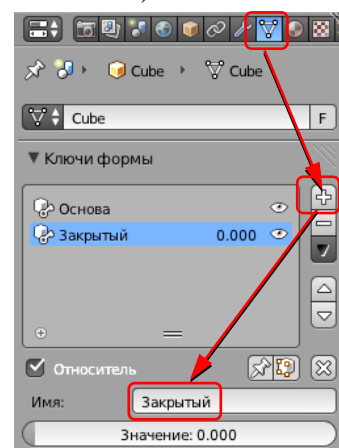



34. Постройте аналогично вторую створку. Это будет базовая форма (*Basis*).

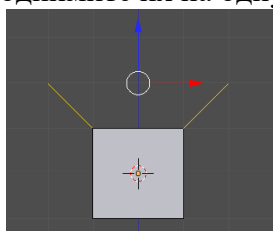


35. Вернитесь к режиму работы с объектами (*Object Mode*, клавиша *Tab*).

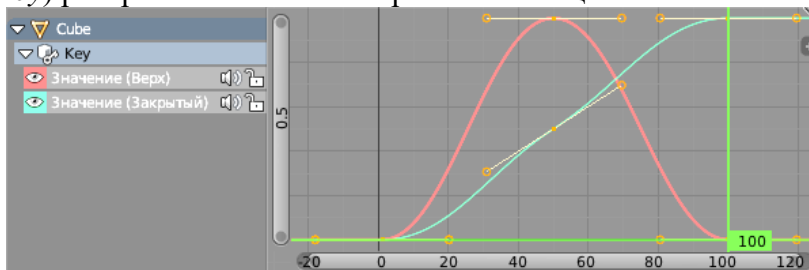
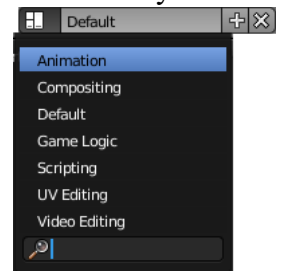
36. Перейдите на страницу свойств кривой  (*Object Data*), найдите панель *Ключи формы* (*Shape Keys*) и создайте с помощью кнопки  два ключа. Первый будет называться *Основа* (*Basis*, базовая форма, открытая коробка), а для второго измените имя на *Закрытый* (*Closed*). Выделите в этом списке ключевую форму *Закрытый*.



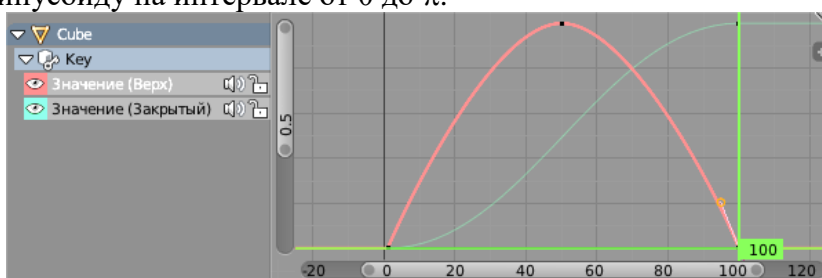
37. Перейдите в режим редактирования сеточной модели (*Edit Mode*, клавиша *Tab*). Переключитесь в режим работы с ребрами (*Ctrl+Tab* или ) . Выделите два крайних ребра створок и соедините их, уменьшив масштаб по оси *X* до нуля. Для этого нужно нажать клавишу *S* (*scale*, изменить масштаб), затем клавишу *X* (только по оси *X*), затем *0* (уменьшить размер до нуля) и *Enter* (завершить изменения).
38. Вернитесь к режиму работы с объектами (*Object Mode*, клавиша *Tab*). Попробуйте изменять *Значение* (*Value*) на панели *Shape Keys*. Что при этом происходит? Правильно ли закрывается коробка? Почему?
39. Уменьшите значение *Value* до нуля и создайте еще одну ключевую форму с именем *Верх* (*Up*). Выделите два боковых ребра створки, перейдите к виду спереди (*Num1*) и поднимите их на одну единицу:





40. Вернитесь к режиму работы с объектами (*Object Mode*, клавиша *Tab*). В нижней части экрана установите конечный кадр анимации (*End*) – 100. Выделите на временной шкале первый кадр, убедитесь, что для ключевых форм *Закрытый* (*Closed*) и *Верх* (*Up*) установлены нулевые значения (*Value*).
41. Выделите в списке форму *Закрытый* (*Closed*), нажмите ПКМ на поле *Значение* (*Value*) и выберите из всплывающего меню команду *Вставить ключевой кадр* (*Insert Keyframe*). Аналогично вставьте ключевой кадр для формы *Верх* (*Up*).
42. Перейдите к кадру 50. Измените параметр *Значение* (*Value*) для формы *Закрытый* (*Closed*) на 0,5, а для формы *Верх* (*Up*) – на 1. Вставьте ключевые кадры для обеих форм.
43. Перейдите к кадру 100. Измените параметр *Значение* (*Value*) для формы *Закрытый* (*Closed*) на 1, а для формы *Верх* (*Up*) – на 0. Вставьте ключевые кадры для обеих форм.
44. Проверьте, как работает анимация (*Alt+A*) на виде спереди. Что плохо? Почему?
45. В верхней части экрана найдите список, позволяющий выбрать стандартное расположение окон на экране, и выберите в нём вариант *Анимация* (*Animation*).
46. Переведите курсор в окно редактора F-кривых (*F-Curve Editor*). Щелчком мыши по треугольнику слева от типа ключевого кадра (*Key*) раскройте список всех кривых анимации.



47. Раскройте окно редактора кривых на полный экран, нажав *Ctrl* и клавишу «стрелка вверх». Удалите узел в кадре 50 для кривой анимации ключа *Закрытый* (*Closed*) (она показана голубым цветом). Первый и последний узлы анимации ключа *Верх* (*Up*) преобразуйте в свободные (клавиша *V*, выбрать из меню вариант *Свободный*, *Free*), и отрегулируйте направляющие рукоятки так, чтобы кривая стала похожа на синусоиду на интервале от 0 до  $\pi$ .



48. Проверьте, как работает анимация? Заметны ли улучшения?
49. Выберите для коробки цвет материала, похожий на картон.
50. Перейдите на страницу свойств  *Мир* (*World*). Установите белый *Цвет горизонта* (*Horizon Color*), включите *Освещение от окружения* (флажок *Environment Lighting*) и уменьшите его интенсивность (*Энергия*, *Energy*) до 0,2.
51. Перейдите на страницу свойств  *Рендеринг* (*Render*). Установите размер кадра 400 на 400 пикселей, качество 100%. Установите формат выходного файла MPEG, задайте имя файла (с расширением *\*.mpg*).
52. Нажмите клавишу *Num0*, чтобы перейти к виду с камеры. Настройте этот вид так, чтобы видеть всю анимацию (*Shift+F*, «режим полета»).
53. Щёлкните по кнопке *Анимация* (*Animation*) и ждите, когда ролик будет готов. Это может занять несколько минут даже на достаточно мощном компьютере.



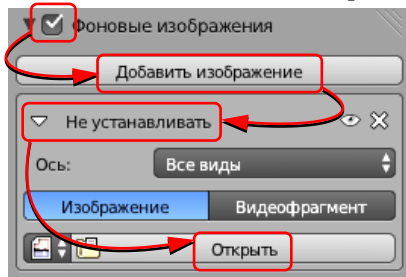
*Дополнительные задания:*


54. Добавьте вторую пару створок, которые открываются в другом направлении.
55. Сделайте так, чтобы коробка не закрывалась, а открывалась.

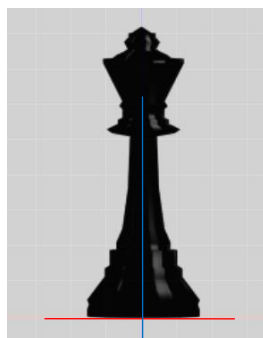
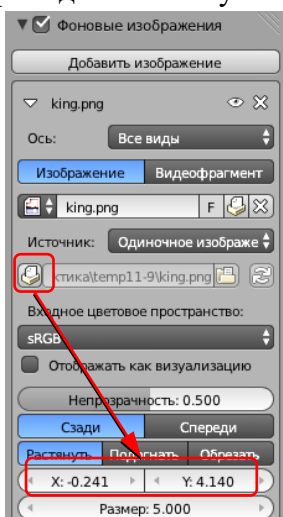
## Практическая работа (дополнительная)

### Анимация. Арматура

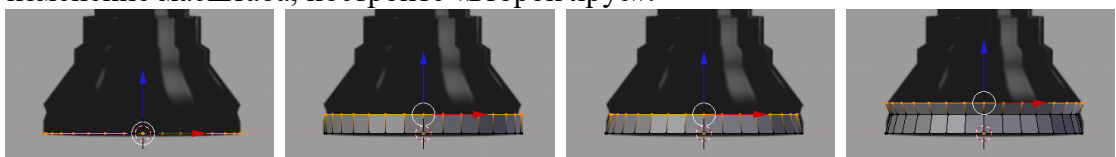
5. Запустите *Blender* и удалите куб, созданный по умолчанию.
14. Нажмите клавишу *N*, чтобы показать панель свойств. Найдите на ней флажок-переключатель *Фоновые изображения* (*Background Images*) и отметьте его стрелкой слева от флажка раскройте группу элементов управления.
15. Щелкните по кнопке *Добавить изображение* (*Add Image*), затем – по белой стрелке слева от надписи *Не устанавливать* (*Not Set*), если эта панель закрыта. Затем щелкните по кнопке *Open* (*Открыть*) и выберите на диске файл **king.png**.



6. Щелкните по кнопке  слева от имени файла, чтобы рисунок был «упакован» в *blend*-файл. Перейдите к виду спереди (*Num1*) и включите ортографическую проекцию (*Num5*), при этом рисунок должен появиться в рабочей области. Изменяя координаты *X* и *Y*, добейтесь, чтобы ось *Z* совпала с осью симметрии фигуры, а ось *X* проходила по низу основания.

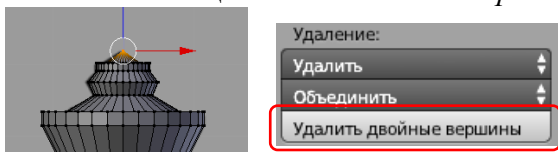


7. Установите курсор в центр сцены (в начало координат), нажав клавиши *Shift+S* и выбрав пункт всплывающего меню *Курсор в центр* (*Cursor to Center*). Нажмите клавиши *Shift+A* и добавьте на сцену окружность Безье (*Добавить – Полисетка – Окружность*, *Add – Mesh – Circle*).
8. Перейдите в режим редактирования сеточной модели (*Edit Mode*, клавиша *Tab*). Выделите все вершины, нажмите клавишу *S* (масштабирование) и измените размер окружности так, чтобы ее диаметр совпал с диаметром основания. Нажмите клавишу *E* (выдавливание), затем клавишу *Z* (только вдоль оси *Z*) и переместите новые вершины вверх до первого изгиба контура. Теперь нажмите клавишу *S* и уменьшите размер нового контура в соответствии с профилем. Снова примените выдавливание и изменение масштаба, постройте «второй ярус».

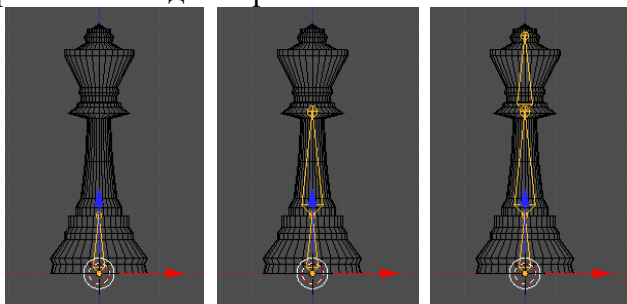


17. Аналогично постройте весь профиль. В самой верхней точке уменьшите масштаб до

нуля (клавиша *S*, затем «0», затем *Enter*). После этого удалите совпадающие вершины, щелкнув по кнопке *Удалить двойные вершины (Remove Doubles)* на панели инструментов (в левой части окна). Убедитесь, что справа в заголовке окна появилось сообщение «Удалено 31 вершины» (*Removed 31 vertices*).

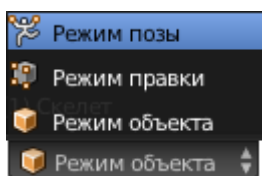


9. Отключите фоновый рисунок – снимите флажок *Фоновые изображения (Background Images)*.
10. Выберите для короля чёрный цвет материала.
11. Сохраните файл под именем **king.blend**.
12. Установите курсор в центр сцены (в начало координат), нажав клавиши *Shift+S* и выбрав пункт всплывающего меню *Курсор в центр (Cursor to Center)*. Нажмите клавиши *Shift+A* и добавьте на сцену арматуру – одиночную кость (*Добавить – Скелет – Одна кость, Add – Armature – Single Bone*).
13. Переключитесь в режим просмотра *Каркас (Wireframe, клавиша Z)*. Увеличьте размер кости так, чтобы она охватила всю нижнюю часть. Затем перейдите в режим редактирования (*Edit Mode, клавиша Tab*) и выделите верхний узел кости. Нажмите клавишу *E* (выдавливание), затем клавишу *Z* (вдоль оси *Z*) и «вытащите» вторую кость так, чтобы она перекрыла среднюю часть фигуры. Аналогично вытащите третью кость для верхней части.



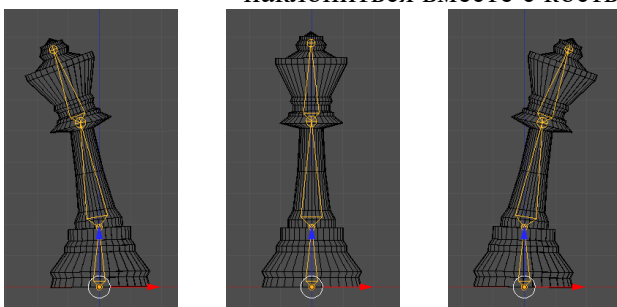
14. Вернитесь в режим работы с объектами. Выделите сначала фигуру короля, а затем (при нажатой клавише *Shift*) – арматуру. Нажмите клавиши *Ctrl+P*, чтобы сделать арматуру родительским объектом для короля. Из всплывающего меню выберите вариант *Скелетная деформация с автоматическими весами (Armature Deform With Automatic Weights)*. Это значит, что связь вершин с конкретными

Установить родителя на	
Объект	Ctrl P
Объект (с сохранением преобразований)	Ctrl P
Скелетная деформация	Ctrl P
С пустыми группами	Ctrl P
С весами обёртки	Ctrl P
С автоматическими весами	Ctrl P
Кость	Ctrl P
Кость (относительно)	Ctrl P







костями будет установлена автоматически.

15. С помощью списка режимов перейдите в режим определения положений (*Pose Mode*). Выделите какую-нибудь кость и попробуйте вращать ее (клавиша *R*). Фигура короля должна наклониться вместе с костью. Отмените вращение кости.



16. Установите длину анимации 100 кадров (параметр *End*) и сделайте текущим первый

кадр. Включите режим автоматической записи ключевых кадров, щелкнув по кнопке  под временной шкалой. Выделите среднюю кость, немного наклоните ее в сторону и верните обратно (чтобы создать ключевой кадр, но оставить исходное положение короля). То же самое сделайте с верхней костью.

17. Перейдите в кадр 25. Наклоните верхнюю и среднюю кости влево на 10 градусов (клавиша *R*, набрать «-10», нажать *Enter*).
18. Перейдите в кадр 50. Восстановите вертикальное положение короля. Для этого наклоните верхнюю и среднюю кости вправо на 10 градусов (клавиша *R*, набрать «10», нажать *Enter*).
19. Перейдите в кадр 75. Наклоните верхнюю и среднюю кости вправо на 10 градусов.
20. Перейдите в кадр 100. Восстановите вертикальное положение короля.
21. Остановите автоматическую запись ключевых кадров, щелкнув по кнопке . Нажмите клавиши *Alt+A*, чтобы посмотреть анимацию.
22. Перейдите на страницу свойств  *Mup (World)*. Установите белый *Цвет горизонта (Horizon Color)*, включите *Освещение от окружения (Environment Lighting)* и уменьшите его интенсивность (*Энергия, Energy*) до 0,2.
23. Перейдите на страницу свойств  *Рендеринг (Render)*. Установите размер кадра 400 на 400 пикселей, качество 100%. Установите формат выходного файла MPEG, задайте имя файла (с расширением **\*.mpg**).
24. Нажмите клавишу *Num0*, чтобы перейти к виду с камеры. Настройте этот вид так, чтобы видеть всю анимацию (*Shift+F*, «режим полета»).
25. Щелкните по кнопке *Animation* (анимация) и ждите, когда ролик будет готов. Это может занять несколько минут даже на достаточно мощном компьютере.

## Практическая работа № 78.

### Язык VRML

2. Постройте модель комнаты, описанную в § 67 учебника.
3. Сделайте комнату замкнутой: опишите все четыре стены, пол и потолок. Поскольку по умолчанию точка наблюдения будет вне комнаты, подумайте, как попасть внутрь (используйте кнопку *Collisions*).
4. Подготовьте и наложите на каждую стену отдельную текстуру.
5. Напишите программу, которая создает VRML-файл по введенным размерам стен комнаты.
6. \*Напишите программу, которая создает VRML-файл с описанием шахматной доски, состоящей из 64 чередующихся черных и белых блоков (объектов **Box**).
7. Постройте простейший лабиринт из нескольких коридоров. Пройдите его от точки входа до точки выхода. Используя режим полёта (*Fly*), посмотрите на лабиринт сверху.
8. Используя комбинацию простейших геометрических тел, попробуйте создать какие-нибудь простые объемные предметы. Например, конус и пара цилиндров позволяет «построить» ракету, а из сфер разного радиуса можно создать модель планетной системы.
9. Используя блоки (параллелепипеды), постройте объемные буквы «Г», «Е» и «Ш».
10. \*Найдите информацию о полях узла **Material** и посмотрите, как их значения влияют на изображение объекта.
11. \*Найдите информацию об узле **Transform**. Примените режимы **rotation** и **scale**.
12. \*Напишите VRML-код, который строит снеговика.