

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Тамбовский государственный университет имени Г.Р. Державина»
Педагогический институт
Отделение допрофессионального образования
Кафедра профильной довузовской подготовки**

«УТВЕРЖДАЮ»

Директор Педагогического института

Гущина Т.И.

«27» января 2022 г.




РАБОЧАЯ ПРОГРАММА
учебного предмета
«Информатика»
для 10-11 классов, год набора: 2022 г.
уровень образования: среднее общее
уровень изучения: углубленный
профиль обучения: технологический

Тамбов, 2022

Разработчики/составители:

 /Белова Е.А. к.п.н., доцент кафедры
профильной довузовской подготовки ТГУ имени Г.Р. Державина.

Эксперты:

 / Самохвалов А.В., к.п.н., доцент кафедры
математического моделирования и информационных технологий Института
математики, физики и информационных технологий ТГУ имени Г.Р.
Державина;

Рабочая программа учебного предмета/курса разработана/составлена на
основе ФГОС СОО и утверждена на заседании кафедры профильной
дovuзовской подготовки 24 января 2022 г., протокол №6.

Заведующий кафедрой

 А.А. Андреева

С О Д Е Р Ж А Н И Е

1. Пояснительная записка.....	4 стр.
– основы разработки рабочей программы учебного предмета;	
– общая характеристика учебного предмета;	
– формы организации образовательного процесса по учебному предмету, предпочтительные формы контроля и оценки;	
– место учебного предмета в учебном плане основной образовательной программы среднего общего образования. (далее – ООП СОО).	
2. Планируемые результаты освоения учебного предмета.....	8 стр.
3. Содержание учебного предмета.....	17 стр.
4. Тематическое планирование учебного предмета.....	23 стр.
5. Учебно-методическое и материально-техническое обеспечение реализации учебного предмета.....	32 стр.
6. Приложение 1 (календарно-тематическое планирование).....	36 стр.
7. Приложение 2 (оценочные материалы).....	57 стр.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Информатика - предмет, непосредственно востребованный во всех видах профессиональной деятельности и в различных траекториях продолжения обучения. Подготовка по этому предмету на углубленном уровне обеспечивает эту потребность наряду с фундаментальной научной и общекультурной подготовкой в данном направлении. Изучение предмета содействует дальнейшему развитию таких умений, как критический анализ информации, поиск информации в различных источниках; формирование представлений своих мыслей и взглядов; моделирование, прогнозирование, организация собственной и коллективной деятельности. При этом эффективность обучения повышается, если оно осуществляется в новой информационной образовательной среде.

Реализация программы учебного предмета «Информатика» допускается как в очном формате обучения, так и в формате электронного обучения с использованием дистанционных образовательных технологий.

Нормативные, концептуальные и иные основы для разработки рабочей программы учебного предмета:

1. Федеральный закон Российской Федерации от 29.12.2012г. №273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);
2. Приказ Министерства образования и науки Российской Федерации от 17 мая 2012 г. № 413 «Об утверждении федерального государственного образовательного стандарта среднего общего образования» (с изменениями и дополнениями);
3. Примерная основная образовательная программа среднего общего образования, одобренная решением Федерального учебно-методического объединения по общему образованию (Протокол от 28 июня 2016 г. № 2/16-з);
4. Приказ Министерства просвещения Российской Федерации от 28 августа 2020 г. № 442 «Об утверждении Порядка организации и осуществления образовательной деятельности по основным общеобразовательным программам - образовательным программам начального общего, основного общего и среднего общего образования» (с изменениями и дополнениями);
5. Приказ Министерства просвещения Российской Федерации от 20.05.2020 № 254 «Об утверждении федерального перечня учебников, допущенных к использованию при реализации имеющих государственную аккредитацию образовательных программ начального общего, основного общего, среднего общего образования организациями, осуществляющими образовательную деятельность»;
6. Методические рекомендации по реализации образовательных программ начального общего, основного общего и среднего общего образования, образовательных программ среднего профессионального образования и дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий Министерства просвещения Российской Федерации от 19.03.2020;
7. Постановлением Главного государственного санитарного врача от 28.09.20 N 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодежи»;
8. Устав и иные локальные акты Университета, регламентирующих основные положения, специфику, особенности организации обучения по программам среднего общего образования.

Программа по учебному предмету «Информатика» разработана на основе примерной рабочей программы «Информатика» к УМК К.Ю. Полякова (Примерная рабочая программа «Информатика». 10-11 классы. Базовый и углубленный уровни / К.Ю. Поляков, Е.А. Еремин.- М.: БИНОМ. Лаборатория знаний, 2016) и соответствует требованиям и положениям основной образовательной программы (Тамбовского государственного университета имени Г.Р.Державина).

Программа настоящего учебного предмета реализуется при помощи/с использованием следующих учебно-методических разработок (комплексов):

1. Информатика. 10 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 1 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019.

2. Информатика. 10 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 2 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019.

3. Информатика. 11 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 1 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019.

4. Информатика. 11 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 2 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019.

Программа конкретизирует содержание предметных тем образовательного стандарта: введение, информация и информационные процессы, данные; тексты и кодирование, передача данных; дискретизация; системы счисления; элементы комбинаторики, теории множеств и математической логики; дискретные объекты; алгоритмы и структуры данных; языки программирования; разработка программ; элементы теории алгоритмов; математическое моделирование; аппаратное и программное обеспечение компьютера; подготовка текстов и демонстрационных материалов; работа с аудиовизуальными данными; электронные (динамические) таблицы; базы данных; подготовка и выполнение исследовательского проекта; системы искусственного интеллекта и машинное обучение; компьютерные сети; деятельность в сети Интернет; социальная информатика; информационная безопасность.

Общая характеристика учебного предмета	
Цель реализации ООП СОО по предмету	обеспечение дальнейшего развития информационных компетенций выпускника, готового к работе в условиях развивающегося информационного общества и возрастающей конкуренции на рынке труда; создание условий для осуществления научной и исследовательской деятельности в области информатики и смежных наук
Задачи реализации ООП СОО по предмету	<p>владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;</p> <p>овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;</p> <p>владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;</p> <p>владение навыками и опытом разработки программ в выбранной среде программирования, включая</p>

	<p>тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;</p> <p>сформированность представлений о важнейших видах дискретных объектов и об их простейших свойствах, алгоритмах анализа этих объектов, о кодировании и декодировании данных и причинах искажения данных при передаче; систематизацию знаний, относящихся к математическим объектам информатики; умение строить математические объекты информатики, в том числе логические формулы;</p> <p>сформированность представлений об устройстве современных компьютеров, о тенденциях развития компьютерных технологий; о понятии "операционная система" и основных функциях операционных систем; об общих принципах разработки и функционирования интернет-приложений;</p> <p>сформированность представлений о компьютерных сетях и их роли в современном мире; знаний базовых принципов организации и функционирования компьютерных сетей, норм информационной этики и права, принципов обеспечения информационной безопасности, способов и средств обеспечения надежного функционирования средств ИКТ;</p> <p>владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними;</p> <p>владение опытом построения и использования компьютерно-математических моделей, проведения экспериментов и статистической обработки данных с помощью компьютера, интерпретации результатов, получаемых в ходе моделирования реальных процессов; умение оценивать числовые параметры моделируемых объектов и процессов, пользоваться базами данных и справочными системами;</p> <p>сформированность умения работать с библиотеками программ; наличие опыта использования компьютерных средств представления и анализа данных.</p>		
Наличие межпредметных связей	Предмет (название темы)	Отражение в предмете Информатика	
	Физика (Равноускоренное движение, Движение, Взаимодействие тел, Механика)	Моделирование	

	Математика (Уравнения и неравенства, Теория множеств, Алгебра высказываний, Теория графов, Графики функций)	Алгоритм, программа Запросы для поисковых систем Проверка истинности логических выражений, построение таблиц истинности, логические уравнения Поиск путей в графе, анализ информационных моделей
Интеграция с внеурочной деятельностью, в том числе с проектной	<p>Учебный предмет «Информатика» интегрирован с курсом внеурочной деятельности «Компьютерные технологии в науке и образовании», в которых учащимся могут быть предложены темы проектов в рамках предметной области «Математика и информатика».</p> <p>Выполненные учащимися индивидуальные и групповые проекты могут быть представлены на конференциях и конкурсах различного уровня («Педагогический Олимп», «Международные образовательные дистанционные конкурсы «Олимпис», Всероссийская олимпиада учащихся «IT в области автоматизации производственных процессов в сельском хозяйстве», «Межрегиональная многопрофильная олимпиада школьников» и др.).</p>	

Формы организации образовательного процесса по учебному предмету, предпочтительные формы контроля и оценки	
Основная форма организации образовательного процесса по учебному предмету	Урок в очном, онлайн и офлайн форматах
Типы занятий	<p>Уроки формирования новых знаний (урок-лекция, в том числе с использованием дистанционных образовательных технологий.)</p> <p>Уроки применений знаний на практике (практическое занятие, в том числе с использованием дистанционных образовательных технологий.);</p> <p>Уроки повторения, систематизации и обобщения знаний, закрепления умений (комбинированный урок, в том числе с использованием дистанционных образовательных технологий);</p> <p>Уроки контроля и проверки знаний и умений (урок - практическая работа).</p>
Виды и формы контроля и оценки знаний, умений обучающихся	<p>Виды контроля: <i>текущий:</i> входной, поурочный, тематический; <i>промежуточная аттестация:</i> по результатам текущего контроля (10 класс, I, II полугодие; 11 класс, II полугодие) практическая работа (11 класс I полугодие) Формы контроля:</p>

	Практическая работа, в том числе с использованием дистанционных образовательных технологий.
--	---

Место учебного предмета в учебном плане ООП СОО

Учебный предмет «Информатика» в учебном плане ООП СОО является учебным предметом по выбору из обязательных предметных областей и изучается на углубленном уровне.

Объем учебного предмета и распределение его по классам (годам) обучения		
Год обучения/класс	Общее количество часов	Недельная нагрузка в часах
10 класс/1 год	140	4
11 класс/2 год	140	4

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОГО ПРЕДМЕТА

Виды образовательных результатов	Планируемые результаты		Раздел/тема учебного предмета
	10 класс/ 1 год обучения		
	Углубленный уровень		
	обучающийся/ученик научится	обучающийся/ученик получит возможность научиться	
предметные	- записывать натуральные числа в системе счисления с данным основанием; использовать при решении задач свойства позиционной записи числа, в частности признак делимости числа на основание системы счисления; -записывать действительные числа в экспоненциальной форме; применять знания о представлении чисел в памяти компьютера;		Системы счисления
	- кодировать и декодировать тексты по заданной кодовой таблице; строить неравномерные коды, допускающие однозначное декодирование сообщений, используя условие Фано; понимать задачи построения кода, обеспечивающего по возможности меньшую	- применять коды, исправляющие ошибки, возникшие при передаче информации; определять пропускную способность и помехозащищенность канала связи, искажение информации при передаче по каналам связи, а также использовать алгоритмы сжатия данных (алгоритм	Тексты и кодирован и. Передача данных

	среднюю длину сообщения при известной частоте символов, и кода, допускающего диагностику ошибок;	<i>LZW и др.);</i>	
	<p>- строить логические выражения с помощью операций дизъюнкции, конъюнкции, отрицания, импликации, эквиваленции; выполнять эквивалентные преобразования этих выражений, используя законы алгебры логики (в частности, свойства дизъюнкции, конъюнкции, правила де Моргана, связь импликации с дизъюнкцией);</p> <p>- строить таблицу истинности заданного логического выражения; строить логическое выражение в дизъюнктивной нормальной форме по заданной таблице истинности; определять истинность высказывания, составленного из элементарных высказываний с помощью логических операций, если известна истинность входящих в него элементарных высказываний; исследовать область истинности высказывания, содержащего переменные; решать логические уравнения;</p>		Элементы комбинаторики, теории множеств и математической логики
	- формализовать понятие "алгоритм" с помощью одной из универсальных моделей вычислений (машина Тьюринга, машина Поста и др.); понимать содержание тезиса Черча-Тьюринга;	<p>- использовать методы машинного обучения при анализе данных; использовать представление о проблеме хранения и обработки больших данных;</p> <p>- использовать знания о методе "разделяй и властвуй"</p>	Элементы теории алгоритмов
	- понимать и использовать	- приводить примеры	Алгоритм

	<p>основные понятия, связанные со сложностью вычислений (время работы и размер используемой памяти при заданных исходных данных; асимптотическая сложность алгоритма в зависимости от размера исходных данных); определять сложность изучаемых в курсе базовых алгоритмов;</p> <ul style="list-style-type: none"> - анализировать предложенный алгоритм, например определять, какие результаты возможны при заданном множестве исходных значений и при каких исходных значениях возможно получение указанных результатов; - создавать, анализировать и реализовывать в виде программ базовые алгоритмы, связанные с анализом элементарных функций (в том числе приближенных вычислений), записью чисел в позиционной системе счисления, делимостью целых чисел; линейной обработкой последовательностей и массивов чисел (в том числе алгоритмы сортировки), анализом строк, а также рекурсивные алгоритмы; - создавать собственные алгоритмы для решения прикладных задач на основе изученных алгоритмов и методов 	<p><i>различных алгоритмов решения одной задачи, которые имеют различную сложность; использовать понятие переборного алгоритма;</i></p> <ul style="list-style-type: none"> - <i>использовать понятие универсального алгоритма и приводить примеры алгоритмически неразрешимых проблем;</i> 	<p>ы и структуры данных</p>
	<ul style="list-style-type: none"> - выполнять отладку и тестирование программ в выбранной среде программирования; использовать при разработке программ стандартные библиотеки языка программирования и 	<ul style="list-style-type: none"> - <i>осознанно подходить к выбору ИКТ-средств и программного обеспечения для решения задач, возникающих в ходе учебы и вне ее, для своих учебных и иных целей;</i> 	<p>Аппаратно е и программн ое обеспече ние компьютер а</p>

	<p>внешние библиотеки программ; создавать многокомпонентные программные продукты в среде программирования;</p> <ul style="list-style-type: none"> - устанавливать и деинсталлировать программные средства, необходимые для решения учебных задач по выбранной специализации; - понимать основные принципы устройства и функционирования современных стационарных и мобильных компьютеров; выбирать конфигурацию компьютера в соответствии с решаемыми задачами; - понимать назначение, а также основные принципы устройства и работы современных операционных систем; знать виды и назначение системного программного обеспечения; - владеть принципами организации иерархических файловых систем и именования файлов; использовать шаблоны для описания группы файлов; - проектировать собственное автоматизированное место; следовать основам безопасной и экономичной работы с компьютерами и мобильными устройствами; соблюдать санитарно-гигиенические требования при работе за персональным компьютером в соответствии с нормами действующих СанПиН. 		
	<ul style="list-style-type: none"> - использовать динамические (электронные) таблицы, в том числе формулы с использованием 		Электронные(динамические) таблицы

	абсолютной, относительной и смешанной адресации, выделение диапазона таблицы и упорядочивание (сортировку) его элементов; построение графиков и диаграмм;		
	<ul style="list-style-type: none"> - использовать компьютерные сети для обмена данными при решении прикладных задач; - организовывать на базовом уровне сетевое взаимодействие (настраивать работу протоколов сети TCP/IP и определять маску сети); - понимать структуру доменных имен; принципы IP-адресации узлов сети; - представлять общие принципы разработки и функционирования интернет-приложений (сайты, блоги и др.); 		Компьютерные сети
метапредметные	<p>1. Умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; <i>Достигается за счет выполнения проектных заданий</i></p> <p>2. Умение продуктивно общаться в процессе совместной деятельности, эффективно разрешать конфликты <i>Достигается за счет выполнения заданий поискового, дискуссионного содержания</i></p> <p>3. Владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем;</p> <p>4. Готовность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации; <i>Достигается за счет выполнения проектных заданий требующих самостоятельного сбора информации и освоения новых программных средств.</i></p> <p>5. Умение использовать средства информационных и коммуникационных технологий в решении коммуникативных и организационных задач с соблюдением норм информационной безопасности. <i>Достигается за счет деления заданий практикума на уровни сложности:</i> 1-й уровень — репродуктивный; 2-й уровень — продуктивный; 3-й уровень — творческий</p>		

личностные	<p>1. Сформированность мировоззрения, соответствующего современному уровню развития науки и общественной практики</p> <p><i>Информация рассматривается как одно из базовых понятий современной науки наряду с материей и энергией. Рассматриваются различные подходы к понятию информации в философии, кибернетике, биологии.</i></p> <p>2. Сформированность навыков сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, общественно полезной, учебно-исследовательской, проектной и других видах деятельности</p> <p><i>За счет выполнения заданий, ориентированных на коллективное обсуждение, дискуссии, выработку коллективного мнения.</i></p> <p>3. 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; осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов</p> <p><i>Ряд проектных заданий требует осознания недостаточности имеющихся знаний, самостоятельного изучения нового для учеников теоретического материала, ориентации в новой предметной (профессиональной) области, поиска источников информации, приближения учебной работы к формам производственной деятельности. В ряде глав учебников имеются разделы, в которых рассказывается о профессиях в области ИКТ</i></p>	
11 класс/ 2 год обучения		
предметные	<p>- описывать графы с помощью матриц смежности с указанием длин ребер (весовых матриц); решать алгоритмические задачи, связанные с анализом графов, в частности задачу построения оптимального пути между вершинами ориентированного ациклического графа и определения количества различных путей между вершинами;</p> <p>- применять метод сохранения промежуточных результатов (метод динамического программирования) для</p>	<p>- использовать графы, деревья, списки при описании объектов и процессов окружающего мира; использовать префиксные деревья и другие виды деревьев при решении алгоритмических задач, в том числе при анализе кодов;</p> <p>Дискретные объекты</p>

	<p>создания полиномиальных (непереборных) алгоритмов решения различных задач; примеры: поиск минимального пути в ориентированном ациклическом графе, подсчет количества путей;</p> <p>- применять при решении задач структуры данных: списки, словари, деревья, очереди; применять при составлении алгоритмов базовые операции со структурами данных;</p>		
	<p>- владеть основными сведениями о табличных (реляционных) базах данных, их структуре, средствах создания и работы, в том числе выполнять отбор строк таблицы, удовлетворяющих определенному условию; описывать базы данных и средства доступа к ним; наполнять разработанную базу данных;</p>	<p>- создавать многотабличные базы данных; работе с базами данных и справочными системами с помощью веб-интерфейса.</p>	Базы данных
	<p>- использовать основные понятия, конструкции и структуры данных последовательного программирования, а также правила записи этих конструкций и структур в выбранном для изучения языке программирования;</p> <p>- использовать в программах данные различных типов; применять стандартные и собственные подпрограммы для обработки символьных строк; выполнять обработку данных, хранящихся в виде массивов различной размерности; выбирать тип цикла в зависимости от решаемой подзадачи;</p>	<p>- использовать второй язык программирования; сравнивать преимущества и недостатки двух языков программирования;</p>	Языки программирования

	<p>составлять циклы с использованием заранее определенного инварианта цикла; выполнять базовые операции с текстовыми и двоичными файлами; выделять подзадачи, решение которых необходимо для решения поставленной задачи в полном объеме; реализовывать решения подзадач в виде подпрограмм, связывать подпрограммы в единую программу; использовать модульный принцип построения программ; использовать библиотеки стандартных подпрограмм;</p> <p>- применять алгоритмы поиска и сортировки при решении типовых задач;</p>		
	<p>- выполнять объектно-ориентированный анализ задачи: выделять объекты, описывать на формальном языке их свойства и методы; реализовывать объектно-ориентированный подход для решения задач средней сложности на выбранном языке программирования;</p> <p>- пользоваться навыками формализации задачи; создавать описания программ, инструкции по их использованию и отчеты по выполненным проектным работам;</p>	<p>- <i>создавать программы для учебных или проектных задач средней сложности;</i></p>	Разработка программ
	<p>- строить дерево игры по заданному алгоритму; строить и обосновывать выигрышную стратегию игры;</p> <p>- разрабатывать и использовать компьютерно-математические модели; анализировать соответствие модели</p>	<p>- <i>использовать информационно-коммуникационные технологии при моделировании и анализе процессов и явлений в соответствии с выбранным профилем;</i></p> <p>- <i>проводить (в несложных случаях) верификацию (проверку надежности и</i></p>	Математическое моделирование

	<p>реальному объекту или процессу; проводить эксперименты и статистическую обработку данных с помощью компьютера; интерпретировать результаты, получаемые в ходе моделирования реальных процессов; оценивать числовые параметры моделируемых объектов и процессов;</p>	<p><i>согласованности) исходных данных и валидацию (проверку достоверности) результатов натурных и компьютерных экспериментов; использовать пакеты программ и сервисы обработки и представления данных, в том числе - статистической обработки</i></p>	
	<p>- применять на практике принципы обеспечения информационной безопасности, способы и средства обеспечения надежного функционирования средств ИКТ; соблюдать при работе в сети нормы информационной этики и права (в том числе авторские права);</p>		Информационная безопасность
	<p>- использовать на практике общие правила проведения исследовательского проекта (постановка задачи, выбор методов исследования, подготовка исходных данных, проведение исследования, формулировка выводов, подготовка отчета); планировать и выполнять небольшие исследовательские проекты;</p>		Подготовка и выполнение исследовательского проекта
метапредметные	<p>1. Умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; <i>Достигается за счет выполнения проектные задания</i> 2. Умение продуктивно общаться в процессе совместной деятельности, эффективно разрешать конфликты <i>Достигается за счет выполнения заданий поискового, дискуссионного содержания</i> 3. Владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; 4. Готовность к самостоятельной информационно-познавательной деятельности, включая умение ориентиро-</p>		

	<p>ваться в различных источниках информации;</p> <p><i>Достигается за счет выполнение проектных заданий требующих самостоятельного сбора информации и освоения новых программных средств.</i></p> <p>5. Умение использовать средства информационных и коммуникационных технологий в решении коммуникативных и организационных задач с соблюдением норм информационной безопасности.</p> <p><i>Достигается за счет деление заданий практикума на уровни сложности:</i></p> <p><i>1-й уровень — репродуктивный;</i></p> <p><i>2-й уровень — продуктивный;</i></p> <p><i>3-й уровень — творческий</i></p>	
личностные	<p>1. Сформированность мировоззрения, соответствующего современному уровню развития науки;</p> <p>2. Способность вести диалог с другими людьми, достигать в нём взаимопонимания, находить общие цели и сотрудничать для их достижения;</p> <p>3. Навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, общественно полезной, учебно-исследовательской, проектной и других видах деятельности;</p> <p>4. 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;</p> <p>5. Осознанный выбор будущей профессии; отношение к профессиональной деятельности как возможности участия в решении проблем.</p>	

3. СОДЕРЖАНИЕ УЧЕБНОГО ПРЕДМЕТА

Название раздел(а)/глав (ы) и/или тем(ы) учебного предмета	Краткое содержание программы разделов(а)/ глав(ы) и/или тем(ы) учебного предмета
Введение. Информация и информационные процессы	Способы представления данных. Различия в представлении данных, предназначенных для хранения и обработки в автоматизированных компьютерных системах и предназначенных для восприятия человеком. Системы. Компоненты системы и их взаимодействие. Информационное взаимодействие в системе, управление. Разомкнутые и замкнутые системы управления. Математическое и компьютерное моделирование систем управления.
Тексты и кодирование. Передача данных	Знаки, сигналы и символы. Знаковые системы. Равномерные и неравномерные коды. Префиксные коды. Условие Фано. Обратное условие Фано. Алгоритмы декодирования при использовании префиксных кодов. Сжатие данных. Учет частотности символов при выборе неравномерного кода. Оптимальное кодирование Хаффмана. Использование программ-архиваторов.

	<p>Алгоритм LZW. Передача данных. Источник, приемник, канал связи, сигнал, кодирующее и декодирующее устройства. Пропускная способность и помехозащищенность канала связи. Кодирование сообщений в современных средствах передачи данных.</p> <p>Искажение информации при передаче по каналам связи. Коды с возможностью обнаружения и исправления ошибок.</p> <p>Способы защиты информации, передаваемой по каналам связи. Криптография (алгоритмы шифрования). Стеганография.</p>
Дискретизация	<p>Измерения и дискретизация. Частота и разрядность измерений. Универсальность дискретного представления информации. Дискретное представление звуковых данных. Многоканальная запись. Размер файла, полученного в результате записи звука. Дискретное представление статической и динамической графической информации.</p> <p>Сжатие данных при хранении графической и звуковой информации.</p>
Системы счисления	<p>Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления. Алгоритм перевода десятичной записи числа в запись в позиционной системе с заданным основанием. Алгоритмы построения записи числа в позиционной системе счисления с заданным основанием и вычисления числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием.</p> <p>Арифметические действия в позиционных системах счисления. Краткая и развернутая форма записи смешанных чисел в позиционных системах счисления. Перевод смешанного числа в позиционную систему счисления с заданным основанием. Представление целых и вещественных чисел в памяти компьютера. Компьютерная арифметика.</p>
Элементы комбинаторики, теории множеств и математической логики	<p>Операции "импликация", "эквиваленция". Логические функции. Законы алгебры логики. Эквивалентные преобразования логических выражений. Логические уравнения. Построение логического выражения с данной таблицей истинности. Дизъюнктивная нормальная форма. Конъюнктивная нормальная форма. Логические элементы компьютеров. Построение схем из базовых логических элементов. Дискретные игры двух игроков с полной информацией. Выигрышные стратегии.</p>
Дискретные объекты	<p>Решение алгоритмических задач, связанных с анализом графов (примеры: построения оптимального пути между вершинами ориентированного ациклического графа; определения количества различных путей между вершинами). Обход узлов дерева в глубину. Упорядоченные деревья (деревья, в которых упорядочены ребра, выходящие из одного узла). Использование деревьев при решении алгоритмических задач (примеры: анализ работы рекурсивных алгоритмов, разбор арифметических и логических выражений). Бинарное дерево. Использование деревьев при хранении данных. Использование графов, деревьев, списков при описании объектов и процессов окружающего мира.</p>
Алгоритмы и структуры данных	<p>Алгоритмы исследования элементарных функций, в частности - точного и приближенного решения квадратного уравнения с</p>

	<p>целыми и вещественными коэффициентами, определения экстремумов квадратичной функции на отрезке.</p> <p>Алгоритмы анализа и преобразования записей чисел в позиционной системе счисления.</p> <p>Алгоритмы, связанные с делимостью целых чисел. Алгоритм Евклида для определения НОД двух натуральных чисел.</p> <p>Алгоритмы линейной (однопроходной) обработки последовательности чисел без использования дополнительной памяти, зависящей от длины последовательности (вычисление максимума, суммы; линейный поиск и т.п.). Обработка элементов последовательности, удовлетворяющих определенному условию (вычисление суммы заданных элементов, их максимума и т.п.).</p> <p>Алгоритмы обработки массивов. Примеры: перестановка элементов данного одномерного массива в обратном порядке; циклический сдвиг элементов массива; заполнение двумерного числового массива по заданным правилам; поиск элемента в двумерном массиве; вычисление максимума и суммы элементов двумерного массива. Вставка и удаление элементов в массиве.</p> <p>Рекурсивные алгоритмы, в частности: нахождение натуральной и целой степени заданного ненулевого вещественного числа; вычисление факториалов; вычисление n-го элемента рекуррентной последовательности (например, последовательности Фибоначчи). Построение и анализ дерева рекурсивных вызовов.</p> <p>Возможность записи рекурсивных алгоритмов без явного использования рекурсии.</p> <p>Сортировка одномерных массивов. Квадратичные алгоритмы сортировки (пример: сортировка пузырьком). Слияние двух отсортированных массивов в один без использования сортировки.</p> <p>Алгоритмы анализа отсортированных массивов. Рекурсивная реализация сортировки массива на основе слияния двух его отсортированных фрагментов.</p> <p>Алгоритмы анализа символьных строк, в том числе: подсчет количества появлений символа в строке; разбиение строки на слова по пробельным символам; поиск подстроки внутри данной строки; замена найденной подстроки на другую строку.</p> <p>Построение графика функции, заданной формулой, программой или таблицей значений.</p> <p>Алгоритмы приближенного решения уравнений на данном отрезке, например, методом деления отрезка пополам. Алгоритмы приближенного вычисления длин и площадей, в том числе: приближенное вычисление длины плоской кривой путем аппроксимации ее ломаной; приближенный подсчет методом трапеций площади под графиком функции, заданной формулой, программой или таблицей значений. Приближенное вычисление площади фигуры методом Монте-Карло. Построение траекторий, заданных разностными схемами. Решение задач оптимизации. Алгоритмы вычислительной геометрии. Вероятностные алгоритмы.</p> <p>Сохранение и использование промежуточных результатов. Метод динамического программирования.</p> <p>Представление о структурах данных. Примеры: списки, словари, деревья, очереди. Хэш-таблицы.</p>
--	--

Языки программирования	<p>Подпрограммы (процедуры, функции). Параметры подпрограмм.</p> <p>Рекурсивные процедуры и функции.</p> <p>Логические переменные. Символьные и строковые переменные.</p> <p>Операции над строками.</p> <p>Двумерные массивы (матрицы). Многомерные массивы.</p> <p>Средства работы с данными во внешней памяти. Файлы.</p> <p>Подробное знакомство с одним из универсальных процедурных языков программирования. Запись алгоритмических конструкций и структур данных в выбранном языке программирования.</p> <p>Обзор процедурных языков программирования.</p> <p>Представление о синтаксисе и семантике языка программирования.</p> <p>Понятие о непроцедурных языках программирования и парадигмах программирования. Изучение второго языка программирования.</p>
Разработка программ	<p>Этапы решения задач на компьютере.</p> <p>Структурное программирование. Проверка условия выполнения цикла до начала выполнения тела цикла и после выполнения тела цикла: постусловие и предусловие цикла. Инвариант цикла.</p> <p>Методы проектирования программ "сверху вниз" и "снизу вверх". Разработка программ, использующих подпрограммы.</p> <p>Библиотеки подпрограмм и их использование.</p> <p>Интегрированная среда разработки программы на выбранном языке программирования. Пользовательский интерфейс интегрированной среды разработки программ.</p> <p>Понятие об объектно-ориентированном программировании. Объекты и классы. Инкапсуляция, наследование, полиморфизм.</p> <p>Среды быстрой разработки программ. Графическое проектирование интерфейса пользователя. Использование модулей (компонентов) при разработке программ.</p>
Элементы теории алгоритмов	<p>Формализация понятия алгоритма. Машина Тьюринга - пример абстрактной универсальной вычислительной модели. Тезис Чёрча-Тьюринга.</p> <p>Другие универсальные вычислительные модели (пример: машина Поста). Универсальный алгоритм. Вычислимые и невычислимые функции. Проблема остановки и ее неразрешимость.</p> <p>Абстрактные универсальные порождающие модели (пример: грамматики).</p> <p>Сложность вычисления: количество выполненных операций, размер используемой памяти; их зависимость от размера исходных данных. Сложность алгоритма сортировки слиянием (MergeSort).</p> <p>Примеры задач анализа алгоритмов: определение входных данных, при которых алгоритм дает указанный результат; определение результата алгоритма без его полного пошагового выполнения.</p> <p>Доказательство правильности программ.</p>
Математическое моделирование	<p>Практическая работа с компьютерной моделью по выбранной теме. Проведение вычислительного эксперимента. Анализ достоверности (правдоподобия) результатов компьютерного экс-</p>

	<p>перимента.</p> <p>Представление результатов моделирования в виде, удобном для восприятия человеком. Графическое представление данных (схемы, таблицы, графики).</p> <p>Построение математических моделей для решения практических задач.</p> <p>Имитационное моделирование. Моделирование систем массового обслуживания.</p> <p>Использование дискретизации и численных методов в математическом моделировании непрерывных процессов.</p> <p>Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности.</p> <p>Компьютерный (виртуальный) и материальный прототипы изделия. Использование учебных систем автоматизированного проектирования.</p>
Аппаратное и программное обеспечение компьютера	<p>Аппаратное обеспечение компьютеров. Персональный компьютер.</p> <p>Многопроцессорные системы. Суперкомпьютеры. Распределенные вычислительные системы и обработка больших данных.</p> <p>Мобильные цифровые устройства и их роль в коммуникациях.</p> <p>Встроенные компьютеры. Микроконтроллеры. Роботизированные производства.</p> <p>Соответствие конфигурации компьютера решаемым задачам.</p> <p>Тенденции развития аппаратного обеспечения компьютеров.</p> <p>Программное обеспечение (ПО) компьютеров и компьютерных систем. Классификация программного обеспечения. Многообразие операционных систем, их функции. Программное обеспечение мобильных устройств.</p> <p>Модель информационной системы "клиент-сервер". Распределенные модели построения информационных систем. Использование облачных технологий обработки данных в крупных информационных системах.</p> <p>Инсталляция и деинсталляция программного обеспечения. Системное администрирование.</p> <p>Тенденции развития компьютеров. Квантовые вычисления.</p> <p>Техника безопасности и правила работы на компьютере. Гигиена, эргономика, ресурсосбережение, технологические требования при эксплуатации компьютерного рабочего места. Проектирование автоматизированного рабочего места в соответствии с целями его использования.</p> <p>Применение специализированных программ для обеспечения стабильной работы средств ИКТ. Технология проведения профилактических работ над средствами ИКТ: диагностика неисправностей.</p>
Подготовка текстов и демонстрация материалов	<p>Технологии создания текстовых документов. Вставка графических объектов, таблиц. Использование готовых шаблонов и создание собственных.</p> <p>Средства поиска и замены. Системы проверки орфографии и грамматики. Нумерация страниц. Разработка гипертекстового документа: определение структуры документа, автоматическое формирование списка иллюстраций, сносок и цитат, списка ис-</p>

	<p>пользуемой литературы и таблиц. Библиографическое описание документов. Коллективная работа с документами. Рецензирование текста.</p> <p>Средства создания и редактирования математических текстов. Технические средства ввода текста. Распознавание текста. Распознавание устной речи. Компьютерная верстка текста. Настольно-издательские системы.</p>
Работа с аудиовизуальными данными	<p>Технические средства ввода графических изображений. Кадрирование изображений. Цветовые модели. Коррекция изображений. Работа с многослойными изображениями.</p> <p>Работа с векторными графическими объектами. Группировка и трансформация объектов.</p> <p>Технологии ввода и обработки звуковой и видеоинформации.</p> <p>Технологии цифрового моделирования и проектирования новых изделий. Системы автоматизированного проектирования. Разработка простейших чертежей деталей и узлов с использованием примитивов системы автоматизированного проектирования. Аддитивные технологии (3D-печать).</p>
Электронные (динамические) таблицы	<p>Технология обработки числовой информации. Ввод и редактирование данных. Автозаполнение. Форматирование ячеек. Стандартные функции. Виды ссылок в формулах. Фильтрация и сортировка данных в диапазоне или таблице. Коллективная работа с данными. Подключение к внешним данным и их импорт. Решение вычислительных задач из различных предметных областей.</p> <p>Компьютерные средства представления и анализа данных. Визуализация данных.</p>
Базы данных	<p>Понятие и назначение базы данных (далее - БД). Классификация БД. Системы управления БД (СУБД). Таблицы. Запись и поле. Ключевое поле. Типы данных. Запрос. Типы запросов. Запросы с параметрами. Сортировка. Фильтрация. Вычисляемые поля.</p> <p>Формы. Отчеты.</p> <p>Многотабличные БД. Связи между таблицами. Нормализация.</p>
Подготовка и выполнение исследовательского проекта	<p>Технология выполнения исследовательского проекта: постановка задачи, выбор методов исследования, составление проекта и плана работ, подготовка исходных данных, проведение исследования, формулировка выводов, подготовка отчета. Верификация (проверка надежности и согласованности) исходных данных и валидация (проверка достоверности) результатов исследования.</p> <p>Статистическая обработка данных. Обработка результатов эксперимента.</p>
Системы искусственного обучения и машинное обучение	<p>Машинное обучение - решение задач распознавания, классификации и предсказания. Искусственный интеллект. Анализ данных с применением методов машинного обучения. Экспертные и рекомендательные системы.</p> <p>Большие данные в природе и технике (геномные данные, результаты физических экспериментов, интернет-данные, в частности данные социальных сетей). Технологии их обработки и хранения.</p>
Компьютерные сети	<p>Принципы построения компьютерных сетей. Аппаратные ком-</p>

	<p>поненты компьютерных сетей. Проводные и беспроводные телекоммуникационные каналы. Сетевые протоколы. Принципы межсетевого взаимодействия. Сетевые операционные системы. Задачи системного администрирования компьютеров и компьютерных сетей.</p> <p>Интернет. Адресация в сети Интернет (IP-адреса, маски подсети). Система доменных имен.</p> <p>Технология WWW. Браузеры.</p> <p>Веб-сайт. Страница. Взаимодействие веб-страницы с сервером. Язык HTML. Динамические страницы.</p> <p>Разработка веб-сайтов. Язык HTML, каскадные таблицы стилей (CSS). Динамический HTML. Размещение веб-сайтов.</p> <p>Использование сценариев на языке Javascript. Формы. Понятие о серверных языках программирования.</p> <p>Сетевое хранение данных. Облачные сервисы.</p>
Деятельность в сети Интернет	<p>Расширенный поиск информации в сети Интернет. Использование языков построения запросов.</p> <p>Другие виды деятельности в сети Интернет. Сервисы Интернета. Геолокационные сервисы реального времени (локация мобильных телефонов, определение загруженности автомагистралей и т.п.); интернет-торговля; бронирование билетов и гостиниц и т.п. Облачные версии прикладных программных систем.</p> <p>Новые возможности и перспективы развития Интернета: мобильность, облачные технологии, виртуализация, социальные сервисы, доступность. Технологии "Интернета вещей". Развитие технологий распределенных вычислений.</p>
Социальная информатика	<p>Социальные сети - организация коллективного взаимодействия и обмена данными. Проблема подлинности полученной информации. Государственные электронные сервисы и услуги. Мобильные приложения. Открытые образовательные ресурсы. Информационная культура. Информационные пространства коллективного взаимодействия. Сетевой этикет: правила поведения в киберпространстве.</p> <p>Стандартизация и стандарты в сфере информатики и ИКТ докомпьютерной эры (запись чисел, алфавитов национальных языков, библиотечного и издательского дела и др.) и компьютерной эры (языки программирования).</p>
Информационная безопасность	<p>Средства защиты информации в автоматизированных информационных системах (АИС), компьютерных сетях и компьютерах. Общие проблемы защиты информации и информационной безопасности АИС. Компьютерные вирусы и вредоносные программы. Использование антивирусных средств.</p> <p>Электронная подпись, сертифицированные сайты и документы. Правовые нормы использования компьютерных программ и работы в Интернете. Законодательство РФ в области программного обеспечения.</p> <p>Техногенные и экономические угрозы, связанные с использованием ИКТ. Правовое обеспечение информационной безопасности.</p>

4. ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ УЧЕБНОГО ПРЕДМЕТА

№	Раздел / глава / тема учебно-го предмета	Кол-во часов	Основные виды деятельности обучающихся (на уровне универсальных учебных действий)
10 класс/ 1 год обучения (140 часов)			
	Раздел 1. Информация и информационные процессы	6	
1.	Диагностический срез.	1	- знать три философские концепции информации
2.	Информатика и информация	1	- оперировать понятием информация в частных науках: нейрофизиологии, генетике, кибернетике, теории информации;
3.	Структура информации.	4	-строить граф-модели (деревья, сети) по вербальному описанию системы; - строить табличные модели по вербальному описанию системы;
	Раздел 2. Кодирование информации	16	
4.	Дискретное кодирование	1	-решать задачи на измерение информации, заключенной в тексте, с алфавитной т.з. (в приближении равной вероятности символов)
5.	Равномерное и неравномерное кодирование	2	- решать несложные задачи на измерение информации, заключенной в сообщении, используя содержательный подход (в равновероятном приближении) - вычислять размет цветовой палитры по значению битовой глубины цвета; - знать понятия «кодирование» «декодирование» информации - приводить примеры примеров технических систем кодирования информации: азбука Морзе, телеграфный код Бодо; - оперировать понятиями «шифрование», «дешифрование»; - использовать принципы представления вещественных чисел; представление текста; представление изображения; цветовые модели; - знать дискретное (цифровое) представление звука;
6.	Декодирование	1	
7.	Алфавитный подход к оценке количества информации	2	
8.	Системы счисления	6	
9.	Кодирование текстов.	1	
10.	Кодирование графической информации	1	
11.	Кодирование звуковой и видеоинформации	2	
	Раздел 3. Логические осно-	26	

	вы компьютеров		
12.	Логические операции и выражения	5	<ul style="list-style-type: none"> - знать основные логические операции (инверсия, конъюнкция, дизъюнкция, импликация, эквивалентность), их свойства и обозначения; - представлять логические выражения в виде формул и таблиц истинности; - объяснять назначение основных логических устройств ЭВМ (регистр, сумматор); - вычислять логическое значение сложного высказывания по известным значениям элементарных высказываний; - уметь строить логические схемы из основных логических элементов по формулам логических выражений.
13.	Упрощение логических выражений	4	
14.	Логические уравнения	5	
15.	Синтез логических выражений	2	
16.	Множества и логика	6	
17.	Предикаты и кванторы	1	
18.	Логические элементы компьютера	3	
	Раздел 4. Компьютерная арифметика	5	
19.	Особенности представления чисел в компьютере	1	<ul style="list-style-type: none"> - иметь представление о системах счисления; - знать правила выполнения арифметических операций в двоичной системе счисления; - применять основные принципы представления данных в памяти компьютера; диапазоны представления целых чисел без знака и со знаком
20.	Хранение в памяти целых чисел	1	
21.	Операции с целыми числами	1	
22.	Хранение в памяти вещественных чисел	1	
23.	Операции с вещественными числами	1	
	Раздел 5. Устройство компьютера	5	
24.	Современные компьютерные системы	1	<ul style="list-style-type: none"> - знать архитектуру персонального компьютера; что такое контроллер внешнего устройства ПК; назначение шины; в чем заключается принцип открытой архитектуры ПК; основные виды памяти ПК; что такое системная плата, порты ввода-вывода; назначение дополнительных устройств: сканер, средства мультимедиа, сетевое оборудование и др.;
25.	Принципы устройства компьютеров	1	
26.	Магистрально-модульная организация компьютера	1	
27.	Процессор, память, устройства ввода и вывода	2	
	Раздел 6. Программное обеспечение	18	
28.	Программы для обработки документов	4	<ul style="list-style-type: none"> - знать что такое программное обеспечение ПК; структура ПО ПК; прикладные программы и их назначение; системное ПО; функции операционной системы; что такое системы программирования;
29.	Пакеты прикладных программ	8	
30.	Программы для создания презентаций	2	

31.	Системное программное обеспечение	3	- подбирать конфигурацию ПК в зависимости от его назначения
32.	Системы программирования	1	- соединять устройства ПК; - производить основные настройки БИОС; - работать в среде операционной системы на пользовательском уровне
	Раздел 7. Компьютерные сети	7	
33.	Локальные сети	1	- знать идею распараллеливания вычислений; что такое многопроцессорные вычислительные комплексы; какие существуют варианты их реализации; назначение и топологии локальных сетей;
34.	Сеть Интернет	3	технические средства локальных сетей (каналы связи, серверы, рабочие станции); основные функции сетевой операционной системы; историю возникновения и развития глобальных сетей; то такое Интернет; систему адресации в Интернете (IP-адреса, доменная система имен); способы организации связи в Интернете; принцип пакетной передачи данных и протокол TCP/IP; основные понятия WWW: web-страница, web-сервер, web-сайт, web-браузер, HTTP-протокол, URL-адрес; что такое поисковый каталог: организацию, назначение; что такое поисковый указатель: организацию, назначение;
35.	Электронная коммерция	1	- работать с электронной почтой;
36.	Личное информационное пространство	2	- извлекать данные из файловых архивов; - осуществлять поиск информации в Интернете с помощью поисковых каталогов и указателей.
	Раздел 8. Алгоритмизация и программирование	43	
37.	Алгоритмы	1	- знать основные типы задач обработки информации;
38.	Оптимальные линейные алгоритмы	1	- оперировать понятием исполнителя обработки информации; понятием алгоритма обработки информации;
39.	Анализ алгоритмов с ветвлениями и циклами	1	- применять на практике определение и свойства алгоритма управления алгоритмической машиной;
40.	Введение в язык Python	1	- использовать способы описания
41.	Вычисления	1	

42.	Ветвления	3	алгоритмов: блок-схемы, учебный алгоритмический язык;
43.	Циклические алгоритмы	4	-применять алгоритм последовательного поиска; алгоритм поиска половинным делением; блочный поиск
44.	Циклы с переменной	2	
45.	Процедуры	4	- составлять алгоритмы решения несложных задач для управления машиной Поста;
46.	Функции	3	- строить алгоритмы управления учебными исполнителями;
47.	Рекурсия	2	- осуществлять трассировку алгоритма работы с величинами путем заполнения трассировочной таблицы;
48.	Массивы	5	- определять этапы решения задачи на компьютере;
49.	Сортировка	3	- определять понятия исполнитель алгоритмов, система команд исполнителя;
50.	Двоичный поиск	2	- понимать возможности компьютера как исполнителя алгоритмов;
51.	Символьные строки	2	- понимать систему команд компьютера;
52.	Матрицы	1	- классифицировать структуры алгоритмов;
53.	Работа с файлами	6	- понимать основные принципы структурного программирования;
			- знать систему типов данных в Питоне, операторы ввода и вывода, правила записи арифметических выражений на Питоне, оператор присваивания, структуру программы на Питоне;
			- анализировать типы данных, логический тип данных, логические величины, логические операции;
			- понимать правила записи и вычисления логических выражений;
			- знать понятия вспомогательного алгоритма и подпрограммы, правила описания и использования подпрограмм-функций, правила описания и использования подпрограмм-процедур;
			- знать правила описания массивов, правила организации ввода и вывода значений массива, правила программной обработки массивов;
			- понимать правила описания символьных величин и символьных строк, основные функции и процедуры для работы с символьной информацией.
			- описывать алгоритмы на языке блок-схем и на учебном алгоритмиче-

			<p>ском языке;</p> <ul style="list-style-type: none"> – составлять программы линейных вычислительных алгоритмов; – разрабатывать и отлаживать типовые программы, обрабатывающие числовые данные; – разрабатывать и отлаживать простейшие программы, реализующие основные алгоритмические конструкции; – разрабатывать и отлаживать типовые программы, реализующие основные методы и алгоритмы обработки массивов: заполнение массива, поиск и подсчет элементов, нахождение максимального и минимального значений, сортировки массива и др.; – программировать циклы, выделять подзадачи и описывать вспомогательные алгоритмы; – описывать функции и процедуры на, записывать в программах обращения к функциям и процедурам; – тестировать и отлаживать программы <p>- работать с файлами и строками.</p>
	Раздел 9. Решение вычислительных задач на компьютере	9	
54.	Точность вычислений	1	<ul style="list-style-type: none"> - решать уравнения, используя различные методы - вычислять длины кривой и площадей фигур - обрабатывать результатов эксперимента
55.	Решение уравнений	3	
56.	Дискретизация	2	
57.	Оптимизация	1	
58.	Статистические расчеты	1	
59.	Обработка результатов эксперимента	1	
	Раздел 10. Информационная безопасность	5	
60.	Вредоносные программы	1	<ul style="list-style-type: none"> - защищать личное пространство в ин-

61.	Шифрование	1	тернете.
62.	Хеширование и пароли	1	
63.	Стеганография/ Безопасность в интернете	1	
64.	Обобщающий урок по теме: Информация	1	
11 класс/ 2 год обучения (136 часов)			
	Раздел 1 Информация и информационные процессы	13	
65.	Количество информации	4	- знать сущность содержательного (вероятностного) подхода к измерению информации -передавать и сжимать данные -использовать алгоритма Хаффмана; RLE;LZW - иметь представление о системном управлении, информационном обществе, информационных технологиях.
66.	Передача и сжатие данных	6	
67.	Информация и управление	2	
68.	Информационное общество	1	
	Раздел 2 Моделирование	13	
69.	Модели и моделирование	2	- иметь представление о модели и моделировании - определять выигрышные стратегии в игре; - знать этапы моделирования; - моделировать движение; - составлять модели неорганического и органического роста; - оперировать вероятностными моделями.
70.	Игровые модели	1	
71.	Модели мышления	1	
72.	Этапы моделирования	1	
73.	Моделирование движения	2	
74.	Математические модели в биологии	4	
75.	Вероятностные модели	2	
	Раздел 3. Базы данных	10	
76.	Многотабличные базы данных	1	- создавать многотабличную БД средствами конкретной СУБД; - реализовывать простые запросы на выборку данных в конструкторе запросов; - реализовывать запросы со сложными условиями выборки.
77.	Реляционная модель данных	1	
78.	Работа с таблицей	2	
79.	Запросы	1	
80.	Формы	2	

81.	Отчёты	1	
82.	Нереляционные базы данных	1	
83.	Экспертные системы	1	
	Раздел 4. Создание веб-сайтов	16	
84.	Веб-сайты и веб-страницы.	1	<ul style="list-style-type: none"> - знать какие существуют средства для создания web-страниц; в чем состоит проектирование web-сайта; что значит опубликовать web-сайт; - создавать несложный web-сайт с помощью редактора сайтов. - применять основные конструкции языка гипертекстовой разметки.
85.	Текстовые веб-страницы	2	
86.	Оформление веб-страниц	2	
87.	Рисунки, звук, видео	1	
88.	Таблицы	2	
89.	Блоки	2	
90.	XML и XHTML	2	
91.	Динамический HTML	2	
92.	Размещение веб-сайтов	2	
	Раздел 5. Элементы теории алгоритмов	5	
93.	Уточнение понятия алгоритма	2	<ul style="list-style-type: none"> - составлять алгоритмы решения несложных задач для управления машиной Поста; - строить алгоритмы управления учебными исполнителями; - осуществлять трассировку алгоритма работы с величинами путем заполнения трассировочной таблицы;
94.	Алгоритмически неразрешимые задачи	1	
95.	Сложность вычислений	1	
96.	Доказательство правильности программ	1	
	Раздел 6. Алгоритмизация и программирование	39	
97.	Целочисленные алгоритмы	2	<ul style="list-style-type: none"> - применять словари, стеки, очереди, деки для решения задач обработки данных; - использовать деревья для организации данных; - применять методы описания графов и некоторыми популярными
98.	Практическая работа по итогам полугодия.	1	
99.	Структуры	3	
100.	Словари	1	

101.	Стек, очередь, дек	12	алгоритмами на графах; - использовать динамическое программирование для решения комбинаторных и оптимизационных задач
102.	Деревья	8	
103.	Графы	10	
104.	Динамическое программирование	2	
	Раздел 7. Объектно-ориентированное программирование	16	
105.	Создание объектов в программе	2	- использовать объектно-ориентированные подходы к разработке программ; - выполнять объектно-ориентированный анализ задачи, - выделять свойства и методы объектов; - использовать инкапсуляцию для защиты данных объектов; - проектировать несложные иерархии классов для прикладных задач
106.	Скрытие внутреннего устройства	2	
107.	Иерархия классов	2	
108.	Программы с графическим интерфейсом	2	
109.	Графический интерфейс: основы	2	
110.	Использование компонентов (виджетов)	2	
111.	Совершенствование компонентов	2	
112.	Модель и представление	2	
	Раздел 8 Обработка изображений	12	
113.	Ввод изображений	1	- определять особенности, достоинства и недостатки растровой графики; особенности, достоинства и недостатки векторной графики; - знать методы описания цветов в компьютерной графике — цветовые модели; способы получения цветовых оттенков на экране монитора и принтере; способы хранения изображений в файлах растрового и векторного форматов; Назначение и функции различных графических программ; использовать методы сжатия графических файлов; - обрабатывать графическую информацию с помощью растровых
114.	Коррекция изображений	1	
115.	Работа с областями	2	
116.	Многослойные изображения	2	
117.	Каналы	1	
118.	Иллюстрации для веб-сайтов	2	
119.	Анимация	2	
120.	Векторная графика	1	

			<p>программ, а именно: выделять фрагменты изображений с использованием различных инструментов (Область, Лассо, волшебная палочка и др.); перемещать, дублировать, вращать выделенные области; редактировать фотографии с использованием различных средств художественного оформления; сохранять выделенные области для последующего использования; монтировать фотографии (создавать многослойные документы) раскрашивать черно-белые эскизы и фотографии; применять к тексту различные эффекты; выполнять цветовую и тоновую коррекцию фотографий; ретушировать фотографии</p>
	Раздел 9. Трёхмерная графика	16	
121.	Работа с объектами	1	<p>- знать понятие 3D-графики. Проекция. - уметь работать с объектами, а именно: преобразовать объекты, связывать объекты. - применять редактирование сетки, деление рёбер и граней, выдавливание, сглаживание, деформацию, отражение света, рендеринг.</p>
122.	Сеточные модели	1	
123.	Модификаторы	2	
124.	Кривые	1	
125.	Материалы и текстуры	2	
126.	Рендеринг	1	
127.	Анимация . Язык VRML	3	
128.	Обобщающий урок по теме: Программирование	2	
129.	Обобщающий урок по теме: Алгоритмизация	2	
130.	Обобщающий урок по теме: Основы программирования	2	
131.	Обобщающий урок по теме: Информация	1	

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ РЕАЛИЗАЦИИ УЧЕБНОГО ПРЕДМЕТА

5.1. Основная учебная литература

- 5.1.1 Информатика. 10 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 1 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019.
- 5.1.2 Информатика. 10 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 2 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019.
- 5.1.3 Информатика. 11 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 1 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019.
- 5.1.4 Информатика. 11 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 2 / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2019

5.2. Дополнительная учебная литература

- 5.2.1 Информатика. 10–11 классы. Углубленный уровень: практикум в 2 ч. Ч. 1 / [И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шеина и др.]. – М.: БИНОМ. Лаборатория знаний, 2016.
- 5.2.2 Информатика. 10–11 классы. Углубленный уровень: практикум в 2 ч. Ч. 2 / [И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шеина и др.]. – М.: БИНОМ. Лаборатория знаний, 2016.
- 5.2.3 Игошин В.И. Математическая логика и теория алгоритмов / В.И. Игошин. – М.: Издательский центр «Академия», 2008.
- 5.2.4 Компьютерные сети. Принципы, технологии, протоколы / [В.Г. Олифер, Н.А. Олифер]. – СПб. Питер, 2010. – 944 с.
- 5.2.5 Залогова Л.А. Компьютерная графика: Практикум./ Л.А.Залогова – М.: ЛБЗ, 2009
- 5.2.6 Чекмарев Ю.В.Вычислительные системы, сети и телекоммуникации: Учебное пособие / Ю.В. Чекмарев. – ДМК Пресс, 2009
- 5.2.7 Ярославские олимпиады по информатике. Сборник задач с решениями. Учебное пособие [С. Г. Волченков, П. А. Корнилов, Ю. А. Белов]. — М.: БИНОМ. Лаборатория знаний, 2010.
- 5.2.8 Окулов С. М. Абстрактные типы данных. Учебное пособие / С.А. Окулова. — М.: БИНОМ. Лаборатория знаний, 2009
- 5.2.9 Анеликова Л. Информатика. Упражнения по текстовому редактору Word / Л. Анеликова. — М.: СОЛОН-ПРЕСС, 2010
- 5.2.10 Макдональд М. Создание Web-сайта. Недостающее руководство / М.Макдональд. — СПб.: БХВ-Петербург, 2013
- 5.2.11 Самоучитель Access 2010 [Ю. Бекаревич, Н. Пушкина]. — СПб.: БХВ-Петербург, 2011 г.
- 5.2.12 Цымбал В.П. Теория информации и кодирование. Задачник. Учебное пособие / В.П. Цымбал. — М.: Ленанд, 2014
- 5.2.13 Алгоритмы компьютерной арифметики / [С.Окулов, А. Лялин]. — М.: БИНОМ. Лаборатория знаний, 2014
- 5.2.14 Васильев А. Числовые расчеты в Excel. Учебное пособие / А. Васильев. — СПб.: Лань, 2014 г.

5.3. Учебно-методическая литература

- 5.3.1 Информатика. 10–11 классы. Базовый и углубленный уровни: методическое пособие / [К.Ю. Поляков, Е.А. Еремин]. – М.: БИНОМ. Лаборатория знаний, 2016.

5.4. Перечень рекомендуемых технических средств обучения, в том числе тренажеров

- 5.4.1. Интерактивная доска/ экран.
- 5.4.2. Компьютер/ноутбук.
- 5.4.3. Мультимедиапроектор.
- 5.4.4 Принтер
- 5.4.5 Сканер
- 5.4.6 Телевизор

5.5. Перечень рекомендуемых электронных библиотечно-образовательных ресурсов

5.5.1 Единая коллекция цифровых образовательных ресурсов. – Режим свободного доступа: <http://school-collection.edu.ru>

5.5.2 Федеральный центр информационно-образовательных ресурсов (ФЦИОР). – Режим свободного доступа: <http://fcior.edu.ru>

5.5.3 Образовательный портал «решу ЕГЭ» – Режим свободного доступа: <https://sdamgia.ru>

5.5.4 Система программирования Python». – Режим свободного доступа: <https://repl.it/>

5.5.5 Образовательный портал К.Ю. Полякова – Режим свободного доступа: <https://kpolyakov.spb.ru/school/ege.htm>

5.5.6 Компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещённый на сайте авторского коллектива: <http://kpolyakov.spb.ru/school/probook.htm>

5.5.7 Электронный задачник-практикум с возможностью автоматической проверки решений задач по программированию: <http://informatics.mccme.ru/course/view.php?id=666>

5.5.8 Материалы для подготовки к итоговой аттестации по информатике в форме ЕГЭ, размещённые на сайте материалы, размещённые на сайте <http://kpolyakov.spb.ru/school/ege.htm>;

5.5.9 сетевая методическая служба авторского коллектива для педагогов на сайте издательства <http://metodist.lbz.ru/authors/informatika/7/>.

Средства коммуникации для проведения онлайн - и офлайн уроков

1. Дневник.ру
2. Социальная сеть ВКонтакте
3. Сервис Zoom видеоконференция

Электронные образовательные ресурсы для проведения онлайн- и офлайн уроков:

1. kpolyakov.spb.ru
2. kompege.ru

5.6. Материально-технические средства и оборудования для обучения

5.6.1. Оборудование учебного кабинета:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- комплект учебно-наглядных пособий по дисциплине;
- методические указания и задания для выполнения практических работ;
- рабочие тетради для выполнения домашнего задания.

5.6.2. Технические средства обучения:

- программное обеспечение общего и профессионального назначения, комплект учебно-методической документации.
- Операционная система Windows 10
- Векторный редактор Corel Draw
- Растровый редактор Adobe Photoshop
- Антивирус Касперского
- Программа-архиватор WinRAR / 7-Zip
- Офисное приложение MicrosoftOffice

- Adobe Dreamweaver
- PyCharm
- Python
- КОМПАС 3D
- Blender

5.6.3. Для обеспечения обучения с использованием дистанционных образовательных технологий необходимы:

- компьютеры или иные технические устройства учеников и педагога с высокоскоростным выходом в Интернет, обеспечивающие возможность работы с мультимедийным контентом: воспроизведение видеоизображений, качественный стереозвук в наушниках, речевой ввод с микрофона и др.

**КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ
УЧЕБНОГО ПРЕДМЕТА «ИНФОРМАТИКА» (280 ч.)**

10 класс/ 1 год обучения (140 часов)

№	Раздел / глава / тема учебного предмета	Кол- во часов	Тип занятия	Формы и виды контроля	Календарные сроки проведения
	Глава 1. Информация и информационные процессы	6			
1	Диагностический срез	1	Урок контроля и проверки знаний и умений	Входной контроль: практическая работа.	
	Структура информации.				
2	Информатика и информация.	1	Урок формирования новых знаний		
3	Практическая работа Деревья.	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
4	Графы. Количество маршрутов. Оптимальные маршруты	1	Урок формирования новых знаний		
5	Анализ информационных моделей	1	Урок формирования новых знаний		
6	Практическая работа Анализ информационных моделей и поиск путей в графе	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
	Глава 2. Кодирование информации	16			
7	Дискретное кодирование	1	Урок формирования новых знаний		

	Равномерное и неравномерное кодирование				
8	Равномерное кодирование	1	Урок формирования новых знаний		
9	Неравномерное кодирование	1	Урок формирования новых знаний		
10	Практическая работа Декодирование	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
	Алфавитный подход к оценке количества информации				
11	Оценка количества информации. Общие понятия	1	Урок формирования новых знаний		
12	Оценка количества информации. Решение задач	1	Урок применения знаний на практике		
	Системы счисления				
13	Системы счисления. Общие понятия	1	Урок формирования новых знаний		
14	Двоичная система счисления. Перевод чисел.	1	Урок применения знаний на практике		
15	Двоичная система счисления. Арифметические операции	1	Урок формирования новых знаний		
16	Двоичная система счисления. Дробные числа	1	Урок формирования новых знаний		
17	Восьмеричная, шестнадцатеричная и другие системы счисления	1	Урок применения знаний на практике		
18	Практическая работа Системы счисления	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
19	Кодирование текстов.	1	Урок формирования новых		

			знаний		
20	Кодирование графической информации	1	Урок формирования новых знаний		
21	Кодирование звуковой и видеоинформации	1	Урок формирования новых знаний		
22	Практическая работа Кодирование информации	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
Глава 3. Логические основы компьютеров		26			
	Логические операции и выражения				
23	Логические операции «И», «ИЛИ», «НЕ», «исключающее ИЛИ»	1	Урок формирования новых знаний		
24	Импликация и эквиваленция	1	Урок формирования новых знаний		
25	Другие логические операции	1	Урок формирования новых знаний		
26	Практическая работа Логические выражения. Доказательства логических тождеств	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
27	Практическая работа Запросы в поисковых системах. Исследование запросов для поисковых систем	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
	Упрощение логических выражений				
28	Логические формулы. Законы алгебры логики	1	Урок формирования новых знаний		
29	Упрощение логических выражений с	1	Урок формирования новых		

	двумя / тремя переменными		знаний		
30-31	Упрощение логических выражений с четырьмя переменными и более	2	Урок формирования новых знаний		
	Логические уравнения				
32-33	Решение логических уравнений с x и y методом «дерева»	2	Урок формирования новых знаний		
34-35	Решение логических уравнений с x и y методом «отображений»	2	Урок формирования новых знаний		
36	Решение логических уравнений с x , y и z методом «отображений»	1	Урок формирования новых знаний		
37	Практическая работа Синтез логических выражений	1	Урок применения знаний на практике		
38	Практическая работа Решение логических выражений	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
	Множества и логика				
39-40	Логические функции на области числовых значений. Отрезки. Графики	2	Урок формирования новых знаний		
41-42	Логические функции на области числовых значений. Множества. Делимость	2	Урок формирования новых знаний		
43	Логические функции на области числовых значений. Задачи с поразрядными операциями	1	Урок формирования новых знаний		
44	Предикаты и кванторы	1	Урок формирования новых знаний		

			знаний		
	Логические элементы компьютера				
45-46	Логические схемы. Основные понятия	2	Урок формирования новых знаний		
47	Анализ логических схем	1	Урок формирования новых знаний		
48	Практическая работа Построение схем на логических элементах		Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
Глава 4. Компьютерная арифметика		5			
49	Особенности представления чисел в компьютере	1	Урок формирования новых знаний		
50	Хранение в памяти целых чисел	1	Урок формирования новых знаний		
51	Операции с целыми числами	1	Урок формирования новых знаний		
52	Хранение в памяти вещественных чисел	1	Урок формирования новых знаний		
53	Практическая работа Операции с вещественными числами	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
Глава 5. Устройство компьютера		5			
54	Современные компьютерные системы	1	Урок формирования новых знаний		
55	Принципы устройства компьютеров	1	Урок формирования новых знаний		
56	Магистрально-модульная организация компьютера	1	Урок формирования новых знаний		
	Процессор, память,				

	устройства ввода и вывода				
57	Процессор, системная плата, внутренняя память	1	Урок формирования новых знаний		
58	Внешние устройства ПК. Практическая работа Выбор конфигурации компьютера	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
Глава 6. Программное обеспечение		18			
	Программы для обработки документов				
59	Текстовые редакторы и процессоры	1	Урок формирования новых знаний		
60	Специальные тексты	1	Урок формирования новых знаний		
61	Использование систем двуязычного перевода и электронных словарей.	1	Урок применения знаний на практике		
62	Практическая работа Коллективная работа над документами	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
	Пакеты прикладных программ				
63-64	Программы для дизайна и вёрстки	2	Урок применения знаний на практике		
65-66	САПР 2D	2	Урок применения знаний на практике		
67-68	САПР 3D	2	Урок применения знаний на практике		
69	Обработка звука. Практическая работа Знакомство с аудиоредактором	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	

70	Обработка видео. Практическая работа Знакомство с видеоредактором	1	Урок приме- нения знаний на практике	Поурочный контроль: практическая работа	
	Программы для создания презентаций				
71	Правила создания презентаций	1	Урок формиро- вания новых знаний		
72	Разработка презентаций Практическая работа Онлайн-сервисы для разработки презентаций	1	Урок приме- нения знаний на практике	Поурочный контроль: практическая работа	
	Системное программное обеспечение				
73	Функции операционной системы по управлению компьютером.	1	Урок формиро- вания новых знаний		
74	Файлы и файловая система.	1	Урок формиро- вания новых знаний		
75	Файловая система. Решение задач	1	Урок приме- нения знаний на практике		
76	Системы программирования	1	Урок формиро- вания новых знаний		
Глава 7. Компьютерные сети		7			
77	Локальные сети	1	Урок формиро- вания новых знаний		
	Сеть Интернет				
78	Сеть Интернет. Адреса в Интернете.	1	Урок формиро- вания новых знаний		
79	Установки протоколов	1	Урок приме- нения знаний на		

	ТСР/IP		практике		
80	Разбиение сети на подсети. Метод квадратов	1	Урок применения знаний на практике		
81	Электронная коммерция	1	Урок формирования новых знаний		
82	Личное информационное пространство	1	Урок формирования новых знаний		
83	Практическая работа Тестирование сети	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
Глава 8. Алгоритмизация и программирование		43			
84	Алгоритмы	1	Урок формирования новых знаний		
85	Оптимальные линейные алгоритмы	1	Урок формирования новых знаний		
86	Анализ алгоритмов с ветвлениями и циклами	1	Урок формирования новых знаний		
87	Введение в язык Python	1	Урок формирования новых знаний		
88	Вычисления	1	Урок формирования новых знаний		
	Ветвления				
89	Условный оператор. Решение задач	1	Урок применения знаний на практике		
90	Множественный выбор. Решение задач	1	Урок применения знаний на практике		
91	Использование ветвлений.	1	Урок применения знаний на практике		

	Циклические алгоритмы				
92	Поиск максимальной цифры	1	Урок применения знаний на практике		
93	Алгоритм Евклида	1	Урок формирования новых знаний		
94-95	Нахождение НОД	2	Урок формирования новых знаний		
	Циклы с переменной				
96-97	Цикл с переменной.	2	Урок формирования новых знаний		
98-99	Вложенные циклы. Практическая работа Основы языка программирования	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
	Процедуры				
100	Процедуры. Основные понятия	1	Урок формирования новых знаний		
101-102	Процедуры. Анализ программ с циклами и подпрограммами	2	Урок применения знаний на практике		
103	Процедуры. Решение задач	1	Урок применения знаний на практике		
	Функции				
104-105	Функции. Анализ программ с циклами и подпрограммами	2	Урок формирования новых знаний		
106	Функции. Решение задач	1	Урок применения знаний на практике		
	Рекурсия				
107	Рекурсия. Общие понятия	1	Урок формирования новых знаний		

108	Рекурсия. Решение задач	1	Урок применения знаний на практике		
	Массивы				
109	Массивы. Общие понятия	1	Урок формирования новых знаний		
110-111	Массивы. Перебор элементов массива.	2	Урок применения знаний на практике		
112-113	Отбор элементов массива по условию. Практическая работа Массивы	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
114	Линейный поиск в массиве.	1	Урок формирования новых знаний		
	Сортировка				
115	Сортировка массивов. Метод пузырька.	1	Урок формирования новых знаний		
116	Сортировка массивов. Метод выбора.	1	Урок формирования новых знаний		
117	Сортировка массивов. Быстрая сортировка.	1	Урок формирования новых знаний		
	Двоичный поиск				
118	Поиск максимального элемента в массиве.	1	Урок формирования новых знаний		
119	Преобразования «строка-число».	1	Урок формирования новых знаний		
	Символьные строки				
120	Сравнение и сортировка строк.	1	Урок формирования новых знаний		
121	Обработка	1	Урок применения знаний на	Поурочный контроль:	

	символьных строк. Практическая работа Символьные строки		практике	практическая работа	
122	Матрицы	1	Урок приме- нения знаний на практике		
	Работа с файлами				
123	Файловый ввод и вывод.	1	Урок формиро- вания новых знаний		
124	Обработка массивов, записанных в файле.	1	Урок формиро- вания новых знаний		
125- 126	Обработка строк, записанных в файле. Практическая работа Файлы	2	Урок контроля и проверки знаний и уме- ний	Тематический контроль: практическая работа.	
Глава 9. Решение вычислительных задач на компьютере		10			
	Решение уравнений				
127	Точность вычислений	1	Урок формиро- вания новых знаний		
128	Практическая работа Решение уравнений. Метод перебора	1	Урок приме- нения знаний на практике	Поурочный контроль: практическая работа	
129	Практическая работа Решение уравнений. Метод деления отрезка пополам	1	Урок приме- нения знаний на практике	Поурочный контроль: практическая работа	
130	Практическая работа Решение уравнений с табличных процессорах	1	Урок приме- нения знаний на практике	Поурочный контроль: практическая работа	
	Дискретизация				
131	Вычисление длины	1	Урок приме- нения знаний на практике		

	кривой		ния знаний на практике		
132	Практическая работа Вычисление площадей фигур	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
133	Практическая работа Оптимизация	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
134	Статистические расчеты	1	Урок применения знаний на практике		
135	Обработка результатов эксперимента	1	Урок применения знаний на практике		
136	Практическая работа Обработка результатов эксперимента	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
Глава 10. Информационная безопасность		4			
137	Вредоносные программы	1	Урок формирования новых знаний		
138	Шифрование. Хеширование и пароли.	1	Урок формирования новых знаний		
139	Практическая работа Стеганография. Безопасность в интернете	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
140	Обобщающий урок по теме: Информация	1	Урок повторения, систематизации и обобщения знаний, закрепления умений		

11 класс/ 2 год обучения (140 часов)

№	Раздел / глава / тема	Кол-во	Тип занятия	Формы и виды контроля	Календарные сроки
----------	----------------------------------	---------------	--------------------	------------------------------	--------------------------

	учебного предмета	часов			проведения
Раздел 1. Информация и информационные процессы		13			
	Количество информации				
1-2	Измерение информации. Содержательный подход	2	Урок формирования новых знаний		
3-4	Вероятность и информация	2	Урок формирования новых знаний		
	Передача и сжатие данных				
5	Передача данных	1	Урок формирования новых знаний		
6	Практическая работа Помехоустойчивые коды	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
7	Сжатие данных	1	Урок формирования новых знаний		
8	Алгоритм Хаффмана	1	Урок формирования новых знаний		
9	Программы-архиваторы	1	Урок формирования новых знаний		
10	Сжатие данных с потерями	1	Урок формирования новых знаний		
	Информация и управление				
11	Системы	1	Урок формирования новых знаний		
12	Системы управления	1	Урок формирования новых знаний		

13	Информационное общество	1	Урок формирования новых знаний		
Раздел 2. Моделирование		13			
14-15	Модели и моделирование	2	Урок формирования новых знаний		
16	Игровые модели	1	Урок формирования новых знаний		
17	Модели мышления	1	Урок формирования новых знаний		
18	Этапы моделирования	1	Урок формирования новых знаний		
19-20	Практическая работа Моделирование движения	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
	Математические модели в биологии				
21	Модели ограниченного и неограниченного роста.	1	Урок формирования новых знаний		
22	Практическая работа Моделирование эпидемии.	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
23	Модель «хищник-жертва».	1	Урок применения знаний на практике		
24	Обратная связь. Саморегуляция.	1	Урок применения знаний на практике		
	Вероятностные модели				
25	Методы Монте-Карло	1	Урок применения знаний на практике		
26	Практическая работа Системы массового обслуживания	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа	

			ний	та.	
Раздел 3. Базы данных		10			
27	Многотабличные базы данных	1	Урок формирования новых знаний		
28	Реляционная модель данных	1	Урок формирования новых знаний		
	Работа с таблицей				
29	Практическая работа Работа с готовой таблицей	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
30	Практическая работа. Создание базы данных.	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
31	Практическая работа Запросы. Создание запросов	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
	Формы				
32	Практическая работа Формы для ввода данных	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
33	Практическая работа Кнопочные формы	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
34	Практическая работа Отчёты	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
35	Практическая работа Нереляционные базы данных	1	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
36	Экспертные системы	1	Урок формирования новых знаний		
Раздел 4. Создание веб-сайтов		16			
37	Веб-сайты и веб-страницы.	1	Урок применения знаний на практике		

38-39	Практическая работа Текстовые веб-страницы	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
40-41	Практическая работа Оформление веб-страниц	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
42	Рисунки, звук, видео. Практическая работа Вставка звука и видео в документ	1	Урок применения знаний на практике	Поурочный контроль: практическая работа	
43-44	Практическая работа Таблицы	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
45-46	Практическая работа Блоки	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
47-48	Практическая работа XML и XHTML	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
49-50	Динамический HTML	2	Урок применения знаний на практике		
52-52	Размещение веб-сайтов. Практическая работа Услуги хостинга	2	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
Раздел 5. Элементы теории алгоритмов		5			
	Уточнение понятия алгоритма				
53	Машина Поста	1	Урок формирования новых знаний		
54	Нормальные алгоритмы Маркова	1	Урок формирования новых знаний		
55	Алгоритмически неразрешимые задачи	1	Урок формирования новых знаний		
56	Практическая работа Сложность	1	Урок контроля и проверки	Тематический контроль: прак-	

	вычислений		знаний и умений	тическая работа.	
57	Доказательство правильности программ	1	Урок формирования новых знаний		
Раздел 6. Алгоритмизация и программирование		39			
	Целочисленные алгоритмы				
58	Решето Эратосфена	1	Урок формирования новых знаний		
59	«Длинные» числа. Квадратный корень	1	Урок формирования новых знаний		
60	Практическая работа по итогам полугодия.	1	Урок контроля и проверки знаний и умений	Промежуточная аттестация: практическая работа.	
	Структуры				
61	Структуры. Общие понятия	1	Урок формирования новых знаний		
62	Работа с файлами	1	Урок применения знаний на практике		
63	Сортировка	1	Урок формирования новых знаний		
64	Словари	1	Урок формирования новых знаний		
	Стек, очередь, дек				
65-66	Стек. Общие понятия	2	Урок формирования новых знаний		
67-68	Практическая работа Использование стека	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
69-70	Очередь. Общие понятия	2	Урок формирования новых		

			знаний		
71-72	Очередь. Решение задач	2	Урок применения знаний на практике		
73-74	Дек. Общие понятия	2	Урок формирования новых знаний		
75-76	Дек. Решение задач	2	Урок применения знаний на практике		
	Деревья				
77-78	Деревья. Общие понятия	2	Урок формирования новых знаний		
79-80	Практическая работа Обход дерева	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
81-82	Вычисление арифметических выражений.	2	Урок применения знаний на практике		
83-84	Хранение двоичного дерева в массиве.	2	Урок применения знаний на практике		
	Графы				
85-86	Графы. Общие понятия	2	Урок формирования новых знаний		
87-88	Задача Прима-Крускала Практическая работа Алгоритм Прима-Крускала	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
89-90	Алгоритм Дейкстры	2	Урок формирования новых знаний		
91-92	Практическая работа Алгоритм Флойда-Уоршелла	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
93-94	Использование графов	2	Урок применения знаний на		

			практике		
	Динамическое программирование				
95	Задачи оптимизации	1	Урок применения знаний на практике		
96	Количество решений	1	Урок применения знаний на практике		
Раздел 7. Объектно-ориентированное программирование		16			
97-98	Создание объектов в программе	2	Урок применения знаний на практике		
99-100	Скрытие внутреннего устройства	2	Урок применения знаний на практике		
101-102	Иерархия классов	2	Урок применения знаний на практике		
103-104	Программы с графическим интерфейсом	2	Урок применения знаний на практике		
105-106	Графический интерфейс: основы	2	Урок применения знаний на практике		
107-108	Использование компонентов (виджетов)	2	Урок применения знаний на практике		
109-110	Практическая работа Совершенствование компонентов	2	Урок контроля и проверки знаний и умений	Тематический контроль: практическая работа.	
111-112	Модель и представление	2	Урок применения знаний на практике		
Раздел 8. Обработка изображений		16			
113	Ввод изображений	1	Урок формирования новых		

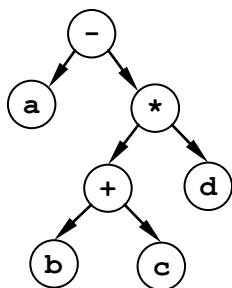
			знаний		
114	Коррекция изображений	1	Урок формирования новых знаний		
115-116	Практическая работа Работа с областями	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
117-118	Многослойные изображения	2	Урок применения знаний на практике		
119	Каналы	1	Урок применения знаний на практике		
120-121	Практическая работа Иллюстрации для веб-сайтов	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
122-123	Анимация	2	Урок применения знаний на практике		
124	Векторная графика	1	Урок применения знаний на практике		
Раздел 9. Трёхмерная графика		12			
125	Работа с объектами	1	Урок формирования новых знаний		
126	Сеточные модели	1	Урок формирования новых знаний		
127-128	Практическая работа Модификаторы	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
129	Кривые	1	Урок формирования новых знаний		
130-131	Практическая работа Материалы и текстуры	2	Урок применения знаний на практике	Поурочный контроль: практическая работа	
132	Рендеринг	1	Урок приме-		

			ния знаний на практике		
133-134	Анимация	2	Урок применения знаний на практике		
135	Язык VRML	1	Урок применения знаний на практике		
136-137	Обобщающий урок по теме: Алгоритмизация	2	Урок повторения, систематизации и обобщения знаний, закрепления умений		
138-139	Обобщающий урок по теме: Основы программирования	2	Урок повторения, систематизации и обобщения знаний, закрепления умений		
140	Обобщающий урок по теме: Информация	1	Урок повторения, систематизации и обобщения знаний, закрепления умений		

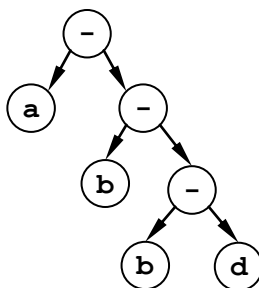
ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
Демонстрационная версия практической работы
Практическая работа
Деревья

1. 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».
2. 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)».
3. 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».
4. 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.

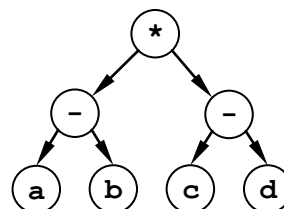
а)



б)



в)



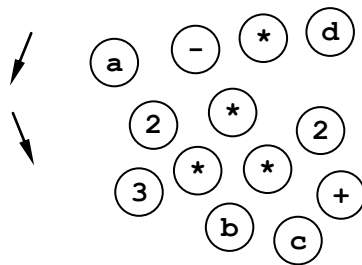
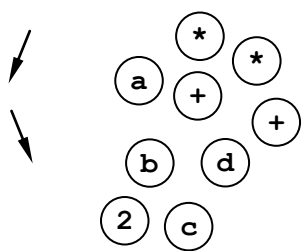
Инфиксная форма:

Постфиксная форма:

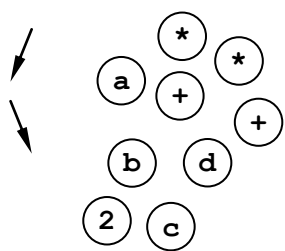
а)	а)
б)	б)
в)	в)

5. Постройте деревья, соответствующие следующим арифметическим выражениям:

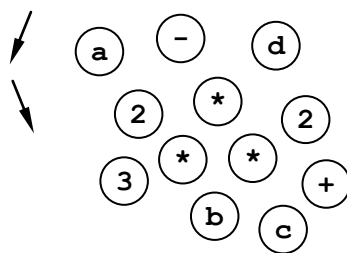
а) $(a+b)*(c+2*d)$ б) $(2*a-3*d)*c+2*b$ а)	в) $(a+b+2*c)*d$ г) $3*a-(2*b+c)*d$ б)
--	--



в)



г)



Запишите эти выражения в префиксной и постфиксной формах.

Префиксная форма:

Постфиксная форма:

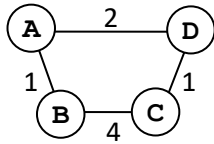
а)		а)	
б)		б)	
в)		в)	
г)		а)	

Практическая работа

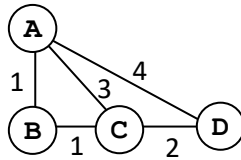
Анализ информационных моделей и поиск путей в графе

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

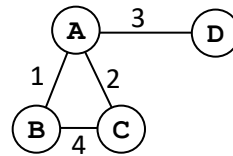
а)



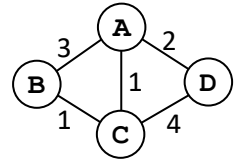
б)



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	

а)

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	

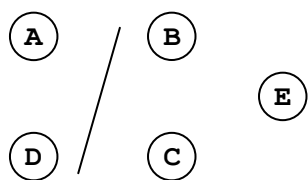
в)

	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	

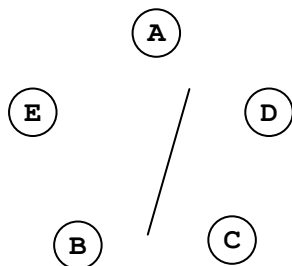
б)

г)

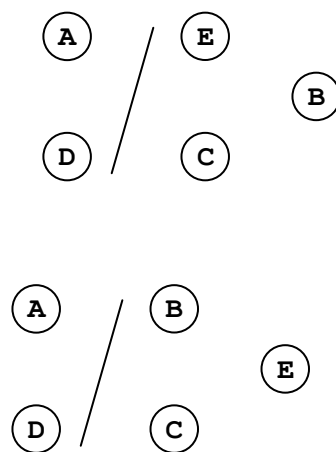
	A	B	C	D	E
A		0	0	1	0
B	0		1	0	1
C	0	1		1	1
D	1	0	1		0
E	0	1	1	0	



В)



Г)

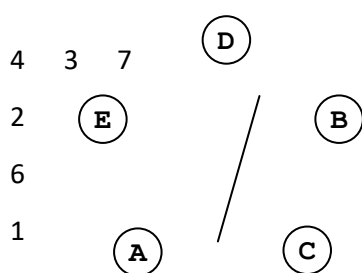


3. Постройте графы, соответствующие каждой из весовых матриц:

а)

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

а)

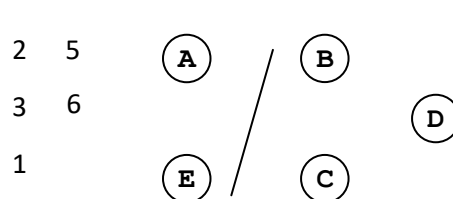


б)

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

в)

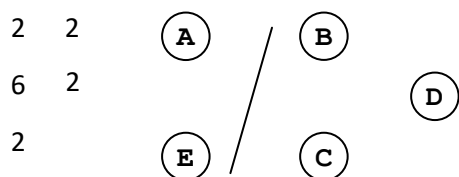
б)



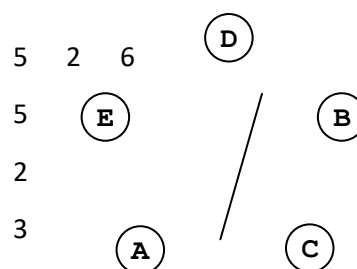
г)

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

В)



Г)

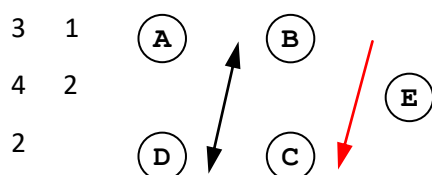


4. Постройте орграф, соответствующий каждой из весовых матриц.

а)

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

а)



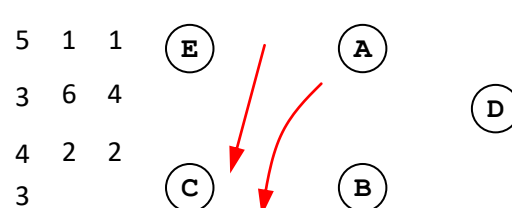
б)

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

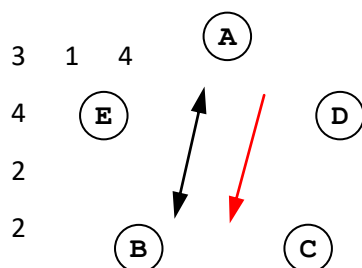
в)

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

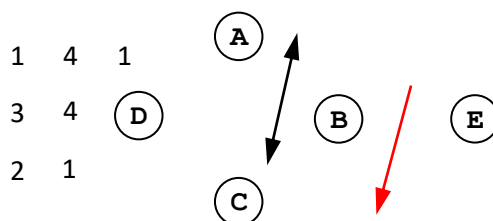
б)



в)



г)



Практическая работа

Декодирование

1. Для кодирования сообщения используется таблица¹

Вариант 1:

A	Б	В	Г	Д
10	11	001	010	01

Сообщение: 0101110010110

(Ответы: ГБАДДА, ДДБВДА)

Вариант 2:

A	Б	В	Г	Д
01	11	110	010	101

Сообщение: 01011100101101 (Ответы: ААВААД, ААВГБА)

Вариант 3:

¹ Выберите вариант по указанию учителя.

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

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) $40C_{16}$

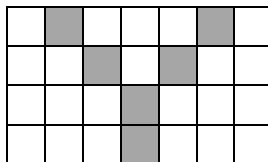
+ 10001010_2 + 5514_8 + $BD6_{16}$ - 10111100_2 - 7760_8 - $C92_{16}$

Практическая 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:



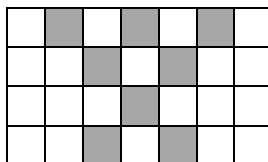
Запишите шестнадцатеричный код, соответствующий этому рисунку.

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?

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



Запишите шестнадцатеричный код, соответствующий этому рисунку.

- 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.

Практическая работа

Доказательство логических тождеств

Используя таблицы истинности, докажите или опровергните тождества:

Вариант 1

$$1) A \rightarrow B = \bar{B} \rightarrow \bar{A} \qquad 2) A \leftrightarrow B = (A + \bar{B}) \cdot (\bar{A} + B) \qquad 3) A + B = A + \bar{A} \cdot B$$

Вариант 2

$$1) A \rightarrow B = B \rightarrow A \qquad 2) A \leftrightarrow B = \bar{A} \cdot \bar{B} + A \cdot B \qquad 3) A + A \cdot B = A$$

Практическая работа

Исследование запросов для поисковых систем

- 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 Яндекса, млн страниц	Выдача Google, млн страниц
яблоки		
груши		
яблоки & груши		
яблоки груши		
Отклонение от формулы Δ		

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

Запрос	Выдача Яндекса, млн страниц	Выдача Google, млн страниц
"яблоки"		
"груши"		
"яблоки" & "груши"		

"яблоки" "груши"		
Отклонение от формулы Δ		

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

Запрос	Выдача Яндекса, тыс. страниц	Выдача Google, тыс. страниц
"яблоки"		
"груши"		
"яблоки" & "груши"		
"яблоки" "груши"		
Отклонение от формулы Δ		

4. Сделайте выводы по результатам исследования двух поисковых систем.

Практическая работа

Синтез логических выражений

Постройте и упростите логические выражения, соответствующие приведённым таблицам истинности. В каждом случае выбирайте наиболее простой способ синтеза. В вашем решении опишите все шаги алгоритма.

Вариант 1

1)

A	B	X
0	0	1
0	1	1
1	0	0

2)

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0

3)

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1

Вариант 2

1)

A	B	X
0	0	1
0	1	0
1	0	1

2)

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0

3)

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0

Практическая работа

Решение логических уравнений

- 1) Сколько различных решений имеет система логических уравнений

$$(x_1 \equiv y_1) \equiv ((x_2 \equiv y_2) \equiv (x_1 \equiv y_1)) = 1$$

$$(x_2 \equiv y_2) \equiv ((x_3 \equiv y_3) \equiv (x_2 \equiv y_2)) = 1$$

$$(x_3 \equiv y_3) \equiv ((x_4 \equiv y_4) \equiv (x_3 \equiv y_3)) = 1$$

$$(x_4 \equiv y_4) \equiv ((x_5 \equiv y_5) \equiv (x_4 \equiv y_4)) = 1$$

$$(x_5 \equiv y_5) \equiv ((x_6 \equiv y_6) \equiv (x_5 \equiv y_5)) = 1$$

$$x_6 \equiv y_6 = 1$$

- 2) Сколько различных решений имеет система логических уравнений

$$(x_1 \wedge y_1) \equiv (\neg x_2 \vee \neg y_2)$$

$$(x_2 \wedge y_2) \equiv (\neg x_3 \vee \neg y_3)$$

...

$$(x_6 \wedge y_6) \equiv (\neg x_7 \vee \neg y_7)$$

- 3) Упростить выражение: $((A + B) \rightarrow C) \cdot (B \rightarrow C \cdot D) \cdot C.$

Практическая работа

Построение схем на логических элементах

1. Постройте схему, соответствующую заданной логической функции, на логических элементах «И», «ИЛИ» и «НЕ». Предварительно преобразуйте выражение так, чтобы количество использованных логических элементов было минимальным.
2. Постройте схему, соответствующую заданной логической функции, на логических элементах «И», «ИЛИ» и «НЕ» (в базисе «И-ИЛИ-НЕ»). Предварительно преобразуйте выражение так, чтобы количество использованных логических элементов было минимальным.
3. * Используя формулу $A + B = \overline{\overline{A} \cdot \overline{B}}$ (она следует из закона двойного отрицания и закона де Моргана), постройте схему, соответствующую логической функции из п. 1, используя только логические элементы «И-НЕ» (в базисе «И-НЕ»).
4. * Используя формулу $A \cdot B = \overline{\overline{A} + \overline{B}}$ (она следует из закона двойного отрицания и закона де Моргана), постройте схему, соответствующую логической функции из п. 1, используя только логические элементы «ИЛИ-НЕ» (в базисе «ИЛИ-НЕ»).

Вариант 1

1. $X = A \cdot \overline{B} + A \cdot \overline{C}$
2. $X = (A \leftrightarrow B) \cdot (B \rightarrow C)$

Вариант 3

1. $X = \overline{A} \cdot \overline{B} + \overline{A} \cdot C$
2. $X = (\overline{B} \leftrightarrow \overline{C}) \cdot (\overline{C} \rightarrow A)$

Вариант 5

1. $X = A + \overline{B} \cdot \overline{C}$
2. $X = C \cdot (\overline{A} \rightarrow \overline{B}) \cdot (\overline{B} \rightarrow C)$

Вариант 2

1. $X = \overline{A} \cdot B + \overline{C} \cdot \overline{A}$
2. $X = (A \leftrightarrow \overline{C}) \cdot (\overline{C} \rightarrow B)$

Вариант 4

1. $X = \overline{A} \cdot \overline{B} \cdot \overline{C}$
2. $X = B \cdot (C \rightarrow A) \cdot (A \rightarrow B)$

Вариант 6

1. $X = \overline{C} + \overline{A} \cdot \overline{B}$
2. $X = A \cdot (B \rightarrow C) \cdot (\overline{A} \rightarrow \overline{C})$

Практическая работа

Операции с вещественными числами

1. Покажите все этапы выполнения операции сложения для приведенных чисел:

- переведите слагаемые в двоичный код, выполните нормализацию

Ответ:

- выполните приведение к одинаковому порядку

Ответ:

- выполните сложение

Ответ:

- выполните нормализацию результата

Ответ:

- переведите результат в десятичную систему счисления

Ответ:

2. Покажите все этапы выполнения операции вычитания для приведенных чисел.

Ответ:

3. *Покажите все этапы выполнения операции умножения для приведенных чисел.

Ответ:

4. *Покажите все этапы выполнения операции деления для приведенных чисел.

Ответ:

Практическая работа

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

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.


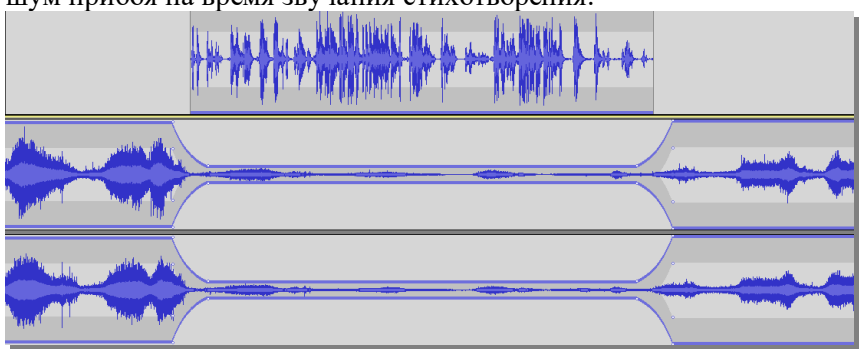
6. Разбейтесь на группы по 3-4 человека. Задача группы – подготовить сообщение на какую-то тему в форме небольшого реферата или презентации.
7. Выберите руководителя, который будет организовывать работу группы.
8. Разделите работу между участниками: подбор материала по одному разделу сообщения, подбор 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

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

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/veopad/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

Онлайн-сервисы для разработки презентаций

Работа в группах по 3-4 человека. Используя онлайн-сервисы для разработки презентаций, например,

- презентации Google
- prezi.com
- 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.

Практическая работа

Тестирование сети

1. 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	

2. С какими из этих компьютеров² 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

3. Определите IP-адреса сайтов:

Сайт	IP-адрес
www.lenta.ru	
www.google.ru	
www.yale.edu	
resh.edu.ru	
fcior.edu.ru	

4. Определите, сколько «прыжков» до следующих сайтов:

Сайт	Число «прыжков»
www.ozon.ru	
www.yandex.ru	

² Учитель должен изменить эти адреса локальной сети так, чтобы часть компьютеров конкретной локальной сети были доступны, а часть – нет.

5. Найдите в Интернете сайты Московского и Санкт-Петербургского государственного университетов и определите их IP-адреса

Сайт	IP-адрес

Практическая работа

Основы языка программирования

Уровень А.

- 11) С клавиатуры вводится трёхзначное число. Нужно вывести ответ «Да», если все его цифры четные, и ответ «Нет» в остальных случаях.
- 12) Определите значение переменной **s** после выполнения фрагмента программы:

```
s = 0
k = 0
while s < 1024:
    s = s + 10
    k += 1
```

```
s:=0;
k:=0;
while s < 1024 do begin
    s:=s+10;
    k:=k+1;
end;
```

- 13) Определите значение переменной **s** после выполнения фрагмента программы:

```
s = 0
for k in range(-3,6):
    s = s + k
```

```
s:=0;
for k:=-3 to 5 do begin
    s:=s+k;
end;
```

- 14) Определите результат работы функции при входном значении, равном 123456:

```
def qq(x):
    s = 0
    for i in range(1, 4):
        s = s + x % 10
        x = x // 10
    return s
```

```
function qq(x: integer):integer;
var i, s: integer;
begin
    s:=0;
    for i:=1 to 3 do begin
        s := s + x mod 10;
        x := x div 10;
    end;
    qq := s;
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
  while x > 0:
    s = 10*s + x % 10
    x = x // 10
  return s
```

```
function qq(x: integer):integer;
var i, s: integer;
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;
```

Практическая 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.

Практическая 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).

Практическая 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

...

Практическая работа

Решение уравнений методом перебора

Напишите программу, которая находит все решения заданного вам уравнения на интервале $[-5;5]$. Программа должна выполнить следующие действия:

6. Определяет и выводит на экран интервалы, на которых расположены корни уравнения.
7. 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

Практическая работа

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

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.

Варианты заданий по теме «Решение уравнений»

№	Уравнение	Ответы		
1	$x^3 - 8x + 1 = 5 \sin x$	-2.945	0.077	2.854
2	$x^3 - 8x + 1 = -2 \sin x - 10 \cos x$	-2.233	1.088	3.286
3	$x^3 - 8x + 1 = -12 \sin x - 10 \cos x$	-1.332	1.975	3.379
4	$x^3 - 8x + 1 = 5 \sin x + 12 \cos x$	-3.377	-0.683	2.347
5	$x^3 - 8x + 1 = -5 \sin x + 12 \cos x$	-3.468	-1.210	1.798
6	$x^3 + 2x^2 - 8x + 1 = -2 \sin x - 15 \cos x$	-3.123	1.493	2.638
7	$x^3 + 2x^2 - 8x + 1 = -2 \sin x + 5 \cos x$	-4.200	-0.497	1.639
8	$x^3 + 2x^2 - 8x + 1 = -12 \sin x + 8 \cos x$	-4.495	-1.386	0.850
9	$x^3 + 2x^2 - 8x + 1 = 5 \sin x + 12 \cos x$	-4.133	-0.638	1.941
10	$x^3 + 2x^2 - 8x + 1 = -5 \sin x + 12 \cos x$	-4.356	-1.042	1.440
11	$x^3 + 3x^2 - 9x - 5 = -\cos x$	-4.690	-0.406	2.212
12	$x^3 + 3x^2 - 9x - 20 = -8 \sin x + 8 \cos x$	-4.488	-2.100	2.363
13	$x^3 + 3x^2 - 9x - 5 = -5 \sin x + 5 \cos x$	-4.834	-1.066	1.776
14	$x^3 + 3x^2 - 9x - 5 = -10 \sin x + 10 \cos x$	-4.931	-1.441	1.448
15	$x^3 + 3x^2 - 9x - 20 = -18 \sin x + 25 \cos x$	-4.756	-2.101	1.720
16	$x^3 + 2x^2 - 6x - 5 = -\sin x - \cos x$	-3.285	-0.707	2.039
17	$x^3 + 2x^2 - 6x - 5 = -\sin x - 15 \cos x$	-1.986	1.086	2.704
18	$x^3 + 2x^2 - 6x - 5 = -10 \sin x - 20 \cos x$	-1.281	1.973	2.772
19	$x^3 + 2x^2 - 6x - 5 = -10 \sin x + 10 \cos x$	-4.040	-1.866	1.316
20	$x^3 + 2x^2 - 6x - 5 = -18 \sin x + 25 \cos x$	-4.395	-1.968	1.196

Практическая работа

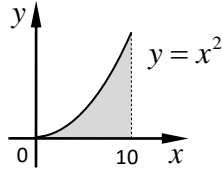
Дискретизация

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 ОХ, на отрезке $[0; 10]$.



Ответ:

Уровень В.

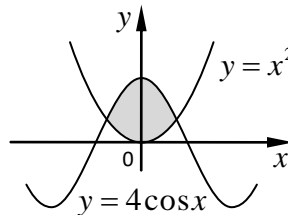
3. Для примера «Полет мяча» вычислите длину траектории движения шарика для углов вылета $35,5^\circ$ и $65,8^\circ$. Сравните полученные результаты.

Ответ:

угол $35,5^\circ \Rightarrow L = \dots$

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 Рамануджана:

$$L = \pi \left[3(a+b) - \sqrt{(3a+b)(a+3b)} \right].$$

Ответ:

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

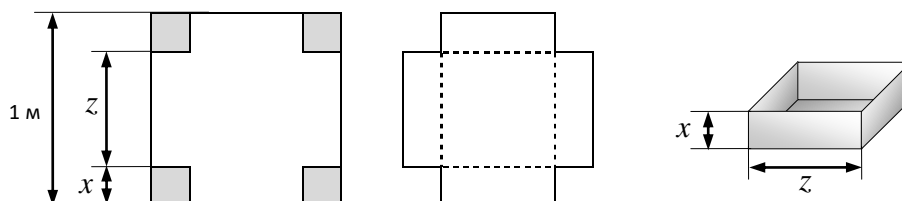
Оптимизация

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
Всего				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 см²;

Ответ:

$$R = \dots, H = \dots$$

- 2) найдите размеры банки, которая имеет минимальную площадь полной поверхности при объеме 500 см³.

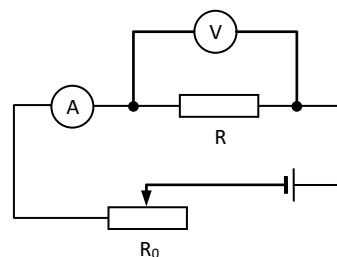
Ответ:

$$R = \dots, H = \dots$$

Практическая работа

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

Использование стеганографии

Используется программа **stegano.exe**.

1. Извлеките текстовую информацию из файла **baloon.bmp**. Число измененных бит на пиксель определите подбором.

Скрытый текст:

2. Работа в парах.
 - а) Найдите в Интернете небольшой рисунок и текст, который вы хотели бы передать своему напарнику.
 - б) Закодируйте текст внутри рисунка. Выберите количество изменяемых бит на пиксель так, чтобы искажение рисунка было почти незаметным.
 - в) Передайте рисунок своему напарнику (например, через разделяемую сетевую папку).

г) Расшифруйте текст, который был скрыт в рисунке, который передал вам напарник:

Скрытый текст:

Практическая работа

Помехоустойчивые коды

1. Для передачи данных используется 8-битовый код, в котором младшие 7 бит – это ASCII-код символа, а старший бит – контрольный бит чётности. Определите коды символов, которые были переданы без ошибок, а ошибочные замените нулями. Используя таблицу ASCII (А – 65, В – 66, ..., Z – 90), определите правильно принятые символы. Попробуйте восстановить известное английское слово, связанное с информатикой, которое пытались передать.
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, приведённые в задании. Для этого набора кодовых слов выполнено такое свойство: любые два слова из набора отличаются не менее чем в трёх позициях. Какое из перечисленных кодовых слов можно использовать для буквы Г, чтобы указанное свойство выполнялось для всех четырёх кодовых слов?
3. 7-битовый код Хэмминга для 4-битовых данных (с 3-мя контрольными битами) строится следующим образом. Биты, номера которых равны степеням двойки – контрольные (это биты 1, 2 и 4, выделенные жёлтым фоном), остальные биты – биты данных:

1 2 3 4 5 6 7

--	--	--	--	--	--	--

Номера остальных битов раскладываются на сумму степеней двойки, например: $5 = 4 + 1$.

Разложение справа даёт номера контрольных битов, которые проверяют этот бит данных. Так 5-й бит проверяется контрольными битами 1 и 4.

Значение контрольного бита вычисляется как бит чётности для всех битов, которые он контролирует. Например, бит 1 контролирует биты 3, 5 и 7 (выделены голубым фоном, в их разложении есть 1). Пусть четверка данных – это 1100:

1 2 3 4 5 6 7

		1		1	0	0
--	--	---	--	---	---	---

Тогда контрольный бит 1 – это бит чётности для цепочки 110, он равен 0, поскольку число единиц в блоке – чётное.

1 2 3 4 5 6 7

0	1	1	1	1	0	0
---	---	---	---	---	---	---

Задание: используйте код Хэмминга, постройте кодовые слова для заданных данных.

*Постройте таблицу кодов Хэмминга для всех двоичных кодов, соответствующих числам от 0 до 15. Для этого используйте электронные таблицы (*Excel* или *OpenOffice.Calc*). Для вычисления бита чётности примените функцию вычисления остатка от деления, которая в *Excel* называется **ОСТАТ**, а в *OpenOffice.Calc* – **MOD**.

4. Код Хэмминга позволяет исправить одну ошибку и обнаружить две. Признаком ошибки (или ошибок) служит несовпадение контрольных битов со значением, которые вычислено по полученным битам данных. Например, пусть приняты данные

1 2 3 4 5 6 7

0	1	1	1	1	1	0
---	---	---	---	---	---	---

По битам данных (с номерами 3, 5, 6 и 7) рассчитываем значения контрольных битов, которые получаются при безошибочной передаче:

$$\text{бит } 1 = (\text{бит } 3 + \text{бит } 5 + \text{бит } 7) \bmod 2 = (1 + 1 + 0) \bmod 2 = 0$$

$$\text{бит } 2 = (\text{бит } 3 + \text{бит } 6 + \text{бит } 7) \bmod 2 = (1 + 1 + 0) \bmod 2 = 0 \neq 1$$

$$\text{бит } 4 = (\text{бит } 5 + \text{бит } 6 + \text{бит } 7) \bmod 2 = (1 + 1 + 0) \bmod 2 = 0 \neq 1$$

Видим, что полученные значения контрольных битов 2 и 4 не совпадают с вычисленными, поэтому при передаче были ошибки. Если предположить, что была только одна ошибка, то номер ошибочного бита вычисляется как сумма номером несовпавших контрольных битов, в данном примере это $2 + 4 = 6$. Таким образом, 6-й бит принят неверно, исправленные данные выглядят так:

1	2	3	4	5	6	7
0	1	1	1	1	0	0

Это код Хэмминга для числа $1100_2 = 12$.

Задание: устройство приняло приведенные в задании 7-битовые блоки, в каждом из которых не более одной ошибки. Восстановите правильные данные и запишите в десятичной системе счисления числовую последовательность, которую пытались передать.

* Используя электронные таблицы, автоматизируйте исправление ошибок: при вводе 7-битового кода Хэмминга в некоторой ячейке должен появляться номер ошибочного бита или 0, если ошибок нет.

Вариант 1.

1. 11010100 01010111 11001001 11010100 11010100 01000101 11010010
2. А – 11111, Б – 11000, В – 00100, Г – ?
1) 00000 2) 00011 3) 11100 4) не подходит ни одно из указанных слов
3. 10, 12
4. 1100001 0101110 1001101 0001001

Вариант 2.

1. 01010011 01001111 11001100 01000001 01010010 11001001 01010011
2. А – 00110, Б – 11000, В – 10011, Г – ?
1) 01101 2) 01001 3) 00011 4) не подходит ни одно из указанных слов
3. 5, 15
4. 0101001 1010011 0100111 1011000

Вариант 3.

1. 11010111 11001001 11001110 01000100 01001111 11010111 01010011
2. А – 11100, Б – 00110, В – 01011, Г – ?
1) 11001 2) 10010 3) 10001 4) не подходит ни одно из указанных слов
3. 4, 11
4. 1101010 0001100 1111000 0110111

Вариант 4.

1. 01000111 01001111 11001111 11000111 11001100 11000101
2. А – 01101, Б – 00110, В – 10001, Г – ?

- 1) 11111 2) 11010 3) 01000 4) не подходит ни одно из указанных слов

3. 6, 10

4. 1111011 0011100 0011000 1101101

Вариант 5.

1. 11011001 01000001 01001110 11000100 11000101 11011000

2. А – 00101, Б – 01011, В – 10110, Г – ?

- 1) 10000 2) 01110 3) 11000 4) не подходит ни одно из указанных слов

3. 7, 13

4. 0011010 1100000 1100100 0000011

Вариант 6.

1. 01000001 11001101 01000001 01011010 11001111 11001110

2. А – 01010, Б – 11001, В – 10100, Г – ?

- 1) 00000 2) 00111 3) 01101 4) не подходит ни одно из указанных слов

3. 8, 14

4. 0001001 0001011 0101101 0101011

Практическая работа

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 м². 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° 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

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 моделей 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

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 делает 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 Δt 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:

$$P_{\max} = 4, T_{\min} = 1, T_{\max} = 9, M = 15.$$

Практическая работа

Работа с готовой таблицей

1. Откройте базу данных *Учебники.odb*³ и изучите её структуру. Сколько таблиц она содержит?

Ответ:

2. Откройте таблицу *Учебники*.
3. Найдите любой учебник, выпущенный издательством «Вита-Пресс».
4. Используя «быстрый фильтр» (фильтр по выделенному), найдите все учебники, которые выпустило издательство «Вита-Пресс». Сколько записей было отобрано?

Ответ:

5. 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.
6. Измените фильтр так, чтобы отобрать все учебники по русскому языку издательства «Дрофа». Сколько записей было отобрано?

Ответ:

7. 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.*
8. Используя фильтр и сортировку⁴, определите, сколько учебников выпустили издательства «Дрофа», «Вита-Пресс» и «Вентана-Граф» для 1-ого класса. Подсказка: программа автоматически считает число выделенных записей.

Ответ:

Практическая работа

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.

³ Если на уроках используется программа *Microsoft Access*, нужно открывать файлы с теми же именами и расширением **.mdb**.

⁴ В программе *Microsoft Access* для этого достаточно использовать расширенный фильтр.

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.

Заказано
Код
Номер заказа
Код блюда

Блюда
Код
Название
Цена

2. Запишите *SQL*-запрос для создания таблицы *Заказы*:

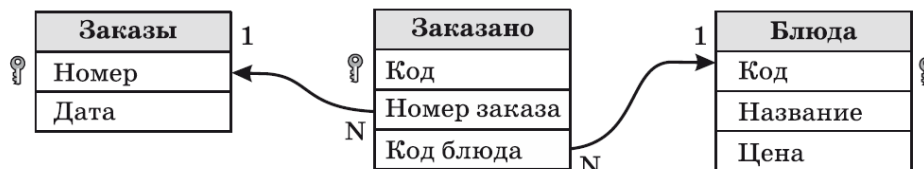
Заказы
Номер
Дата

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

Создание запросов

Уровень А.

При выполнении этой работы используется база данных *Футбол*, с которой вы работали на предыдущем уроке.

1. Постройте запрос с именем *Запрос.Лучшие*, 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.

ЗапросЗаказы			
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.

ЗапросКОплате		
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:

 Аметист	
Победы	10
Ничьи	7
Поражения	3
Зарплата	13 200 руб.

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

Формы для ввода данных

Уровень А.

При выполнении этой работы используется база данных *Футбол*, с которой вы работали на предыдущих уроках.

- Постройте форму на основе таблицы *Футбол* с помощью мастера.
- Добавьте на форму изображение футбольного мяча (файл **ball.jpg**).
- Удалите надпись «Команда» и выделите название команды жирным шрифтом увеличенного размера. Внешний вид формы должен получиться примерно такой, как на рисунке справа.
- 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.

10. Проверьте, появились ли эти записи в таблице *Футбол*.
11. 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

Дата

	Блюдо	Цена
▶	борщ	80 руб.
	гуляш	70 руб.
	чай	10 руб.

Запись из

К оплате

Практическая работа

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:

База данных «Футбол»

Лучшие

Полная таблица

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, постройте кнопочную форму, показанную на рисунке:

База данных «Кафе»

Блюда

Заказы

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

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

Команда	Победы	Ничьи	Поражения	Очки	Место
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.

Заказы		
Исполнитель: Василий Петров		
Дата: 15.12.12		
Дата 11.12.12		
Заказ 1		
<i>Блюдо</i>	<i>Цена</i>	
гуляш	40 руб.	
борщ	30 руб.	
рассольник	20 руб.	
Дата 12.12.12		
Заказ 2		
<i>Блюдо</i>	<i>Цена</i>	
компот	10 руб.	

Практическая работа

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 образом⁵:

```
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).

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 идентификаторы⁶:

```
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." } )
```

⁵ 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*.

⁶ 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')
```



```
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 предыдущем 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:

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 сработает).

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:

Практическая работа

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.

Работа Василия Иванова

Мраморный карьер Рускеала

Рускеальские каменоломни — это памятник природы и горного дела. Они находятся в Сортавальском районе, в 24 километрах к северу от города Сортавала, рядом с поселком Рускеала. Мрамор тут добывали ещё 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 «Ладжская»

Добраться до карьера Рускеала от Петрозаводска не так просто:

1. вначале нужно ехать по трассе **E105** (M18) до **Пряжи**;
2. свернуть направо и двигаться по трассе Р21 в сторону **Сортавала**;
3. не доезжая до Сортавала, свернуть на **Вартсилу**;
4. проехать поселок **Рускеала**;
5. затем будет мост через речку Тохмайоки, сразу за которым будет указатель налево к Мраморному каньону.

Текст с сайтов www.all-karelia.ru, tourskaza.ru, ruskeala.info.

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.

Практическая работа

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 заголовок 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

Вставка звука и видео в документ

1. 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**.
2. Добавьте на веб-страницу звуковой файл **hcopter.wav** из каталога **audio** (шум вертолёт). Используйте тэг **embed**, отключите автозапуск проигрывания, установите ширину проигрывателя 300 пикселей и высоту 20 пикселей.
3. 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.
4. 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:

- С помощью тэга **embed** добавьте на страницу флэш-ролик **cube.swf** (вращающийся кубик) из каталога **flash**. Установите размеры 257 на 200 пикселей.
- 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.
- 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.
- 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.

- 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.

- Обновите страницу в браузере и выясните, поддерживает ли ваш браузер тэг **video**, и с какими форматами он умеет работать.

Ответ:

Практическая работа

Таблицы

- 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.
- 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».



Практическая работа

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](#).

Выводите 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^А 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:

Практическая работа

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 файлы 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.xml**. 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*⁷.

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>
```

⁷ Для того, чтобы выполнить работу в *Google Chrome*, нужно запустить его с ключом **--allow-file-access-from-files**.

```

<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'?>
```

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 &gt;	больше или равно (англ. <i>greater than</i>)
>= или &gt;=	больше или равно (англ. <i>greater than</i>)
&lt;	меньше (англ. <i>less than</i>)
&lt;=	меньше или равно

Знаки > и < в коде веб-страниц заменяют соответственно на **>** и **<**; 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 км², 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 < 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 отобрать страны, население которых меньше 20 млн. человек, а также все страны, площадь которых больше 100 тыс. км².

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.xsl**, назвав копию **europe2.xsl**. 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 ссылки⁸

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 ****, 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:

<xsl:attribute name="src">

<xsl:value-of select="@Фото" />

</xsl:attribute>

Элемент **xsl:attribute** изменяет значение атрибута **src** тэга ****, 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" />

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.

⁸ Дополнительное задание для любознательных.

Использование Javascript⁹

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-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-таблицы¹⁰. 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**:

⁹ Дополнительное задание для любознательных.

¹⁰ Изучая этот файл, вы заметите, что для браузера *Internet Explorer* предусмотрен особый вариант действий, потому что он не поддерживает команды, работающие в остальных браузерах.

```

<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

```


<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).



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 с 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 сине-зелёным 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^А класс

Практическая работа

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/yca/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/yca/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:

Практическая работа

Сложность вычислений

1. 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:

2. 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:

3. 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:

4. 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:

5. 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:

6. 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:

7. 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:

Практическая работа

Использование стека

Уровень А.

1. Напишите программу, которая вычисляет с помощью стека значение арифметического выражения, записанного в постфиксной форме. Выражение вводится с клавиатуры в виде символьной строки.

Уровень В.

2. Напишите программу, которая вычисляет значение арифметического выражения, записанного в префиксной форме. Выражение вводится с клавиатуры в виде символьной строки.

Уровень С.

3. 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.

4. Найдите в литературе или в Интернете алгоритм перевода арифметического выражения из инфиксной формы в постфиксную, и напишите программу, которая решает эту задачу.

Практическая работа

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

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

Алгоритм Прима-Крускала

Уровень А.

1. Напишите программу, в которой задаётся весовая матрица графа размером 6×6 , 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

Совершенствование компонентов

1. Постройте программу с компонентом **TIntEdit**, описание которой приведено в § 49 учебника.
2. *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).

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**.

Практическая работа


Работа с областями

Сглаживание и растушевка

Сначала нужно разобраться в том, что такое сглаживание и растушевка при выделении областей.

1. Создайте новый документ размером 300 на 100 пикселей с белым фоном.
2. Установите черный цвет фона (это можно сделать, нажав клавишу X – поменять основной и фоновый цвета).
3. Включите инструмент  *Эллипс*, в свойствах инструмента выключите режимы *Сглаживание* и *растушевать края*.
4. В левой части выделите круг размером примерно 80 на 80 пикселей и залейте его фоновым цветом, нажав клавишу *Delete*.
5. 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.
6. 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.

7. Откройте файл **baikal.jpg**.
8. Выберите инструмент  *Эллипс*, включите режим растушевки краев с радиусом 100 пикселей. Выделите центральную часть фотографии.
9. Сделайте инверсию выделения, нажав клавиши *Ctrl+I* (при этом выделенные области становятся невыделенными и наоборот).
10. 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.

11. Откройте файл **scan-multi.jpg**.
12. Включите инструмент  *Лассо*. Выделите одну из фотографий, щелкая в её углах и затем в первой точке, чтобы замкнуть контур выделения.
13. Скопируйте выделенную область в буфер обмена (*ПКМ – Правка – Копировать*).
14. Создайте новое изображение из рисунка, который содержится в буфере обмена (*ПКМ – Правка – Вставить как – Новое изображение* или *Shift+Ctrl+V*).
15. Поверните рисунок, если нужно, и обрежьте поля. Сохраните его в виде отдельного файла с именем **photo-1.jpg**.

16. Сохраните таким же способом остальные фотографии.

Практическая работа

Иллюстрации для веб-сайтов

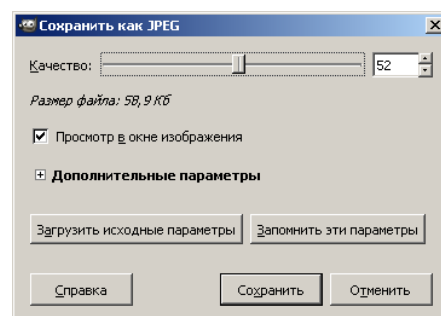
В этой работе мы научимся сохранять изображения, предназначенные для размещения на веб-сайтах. Главная задача – добиться минимального размера и сохранить приемлемое качество.

16. Откройте файл **travel.xcf**.

17. 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.

18. Выберите пункт меню *Файл – Сохранить копию*, в появившемся окне введите имя файла **travel.jpg**.

19. 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.



20. Выберите минимальное качество, которое вы считаете допустимым, и сохраните файл **travel.jpg**.

Какой уровень качества вы считаете допустимым?

Ответ:

21. 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.

22. Откройте файл **travel.gif**, увеличьте изображение и посмотрите, устраивает ли вас его качество. Сделайте выводы. Закройте файл.

Ответ:

23. 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:


24. По результатам этого эксперимента сделайте выводы о том, в каком формате лучше всего готовить фотографии для веб-сайта.

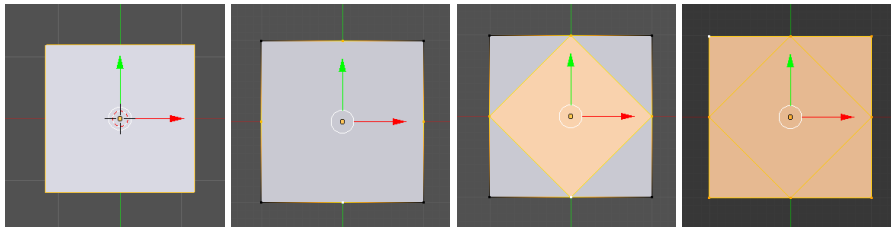
Ответ:



Практическая работа

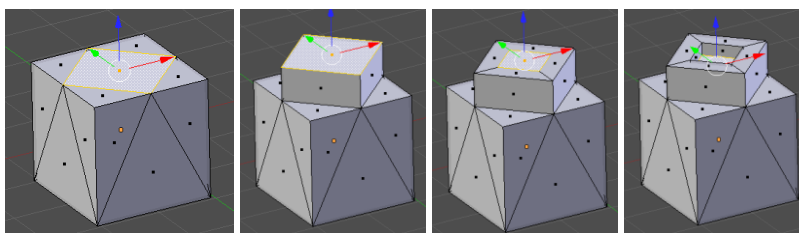
Модификаторы

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:




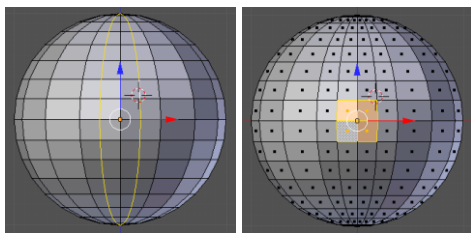
2. Перейдите в режим редактирования (клавиша *Tab*) и отмените выделение всех граней (клавиша *A*). Проверьте, чтобы кнопка  в нижней части рабочей области, которая ограничивает выделение только видимыми гранями (запрещает выделение на обратной стороне фигуры) была включена.
3. Перейдите к ортографической 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*).



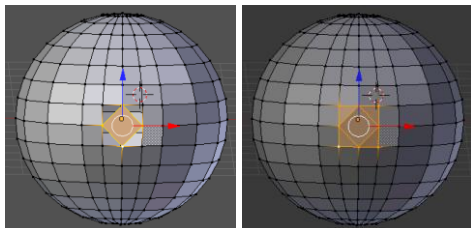
7. Сохраните модель под именем **mesh.blend**.

Модель головы слоника

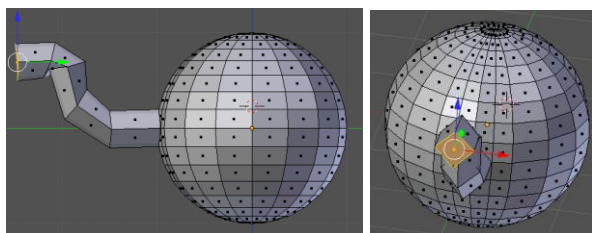
1. Создайте новую сцену. Выделите куб и удалите его (клавиша *Delete*).
2. Добавьте на сцену новый объект – сферу (*UV-sphere*) и уменьшите число её сегментов до 16.
3. Включите вид спереди (клавиша *Num1*), ортографическую проекцию (*Num5*) и перейдите в режим редактирования (*Edit Mode*) с помощью клавиши *Tab*.
4. Отмените выделение (клавиша *A*).
5. Включите режим работы с гранями (*Ctrl+Tab* или ). С помощью инструмента *Разрезать петлей 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.





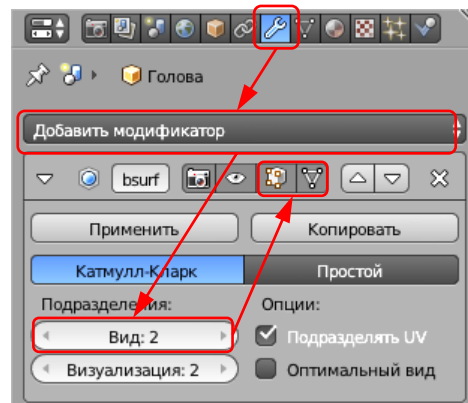
8. Переключитесь в режим редактирования граней, выделите центральную грань и перейдите к виду справа (*Num3*). Нажмите клавишу *Ctrl* и, удерживая её, щелчками мыши постройте хобот:



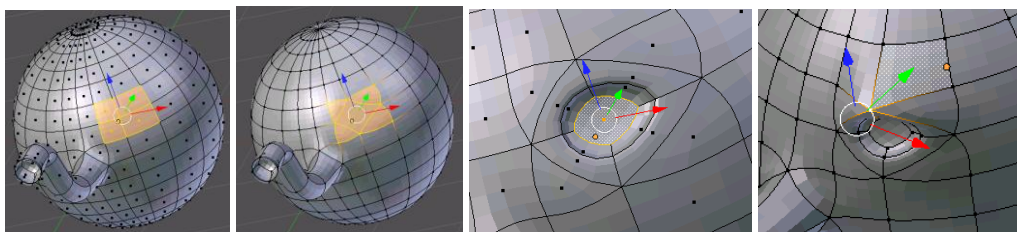
9. Выделите грань в торце хобота, нажмите клавише *E* (выдавливание) и сразу же клавишу *Enter* (будет создана копия всего контура грани). Затем снова нажмите клавишу *E* и вдавите торец внутрь хобота.

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*).

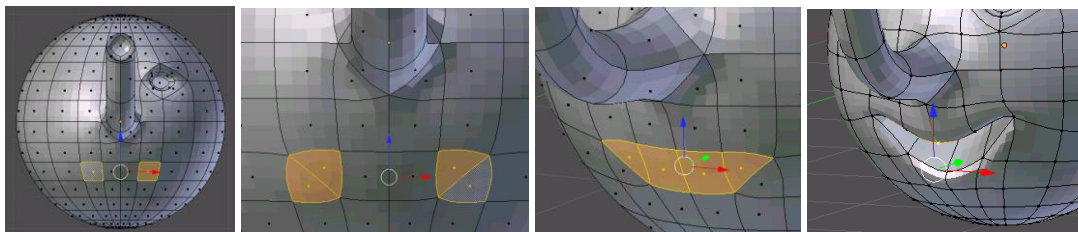
11. В свойствах модификатора в поле *Вид (View)* увеличьте число делений грани при просмотре до 2. Для того, чтобы модификатор применялся в режиме редактирования сеточной модели, включите кнопки-выключатели  и .



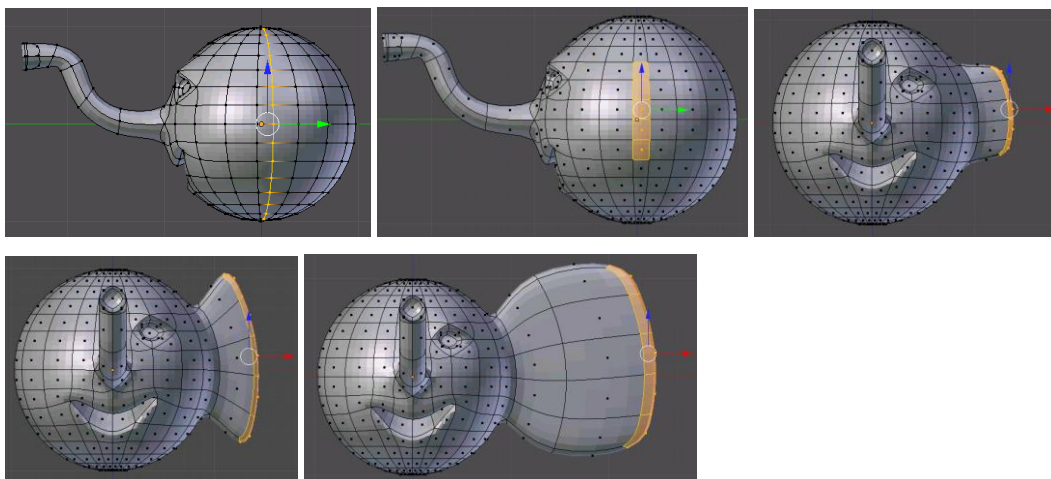
12. Выделите 4 грани из которых будем строить глаз. Удалите их и постройте такую же сетку, как для выдавливания хобота:




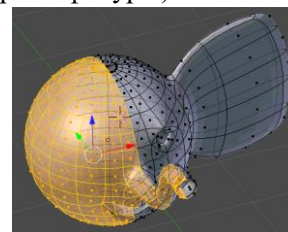
13. Выделите центральную грань отмеченной части, нажмите клавише *E* (выдавливание) и сразу же клавишу *Enter* (будет создана копия всего контура грани). Затем постройте внутренний контур, используя масштабирование (клавиша *E*). Затем снова нажмите клавишу *E* и вдавите центральную грань немного внутрь.
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.

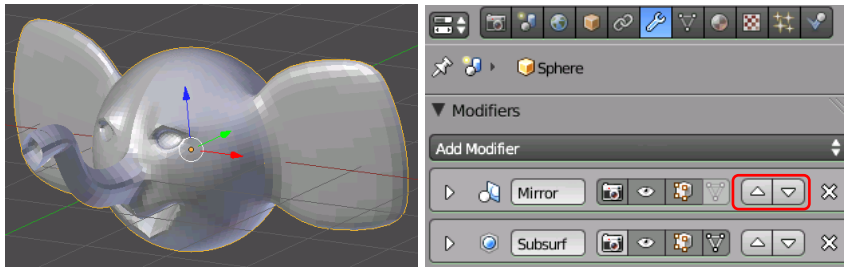


16. Выделите 4 грани (см. рисунок) и вдавите их внутрь. Перейдите к виду спереди и измените положение вершин так, чтобы рот стал улыбающимся.
17. Перейдите к виду справа и примените инструмент *Разрезать петлей со сдвигом (Loop Cut and Slide)* для одной из граней. Затем выделите 5 граней и примените выдавливание (клавиша *E*). После этого используйте масштабирование (клавиша *S*) и затем снова выдавливание.

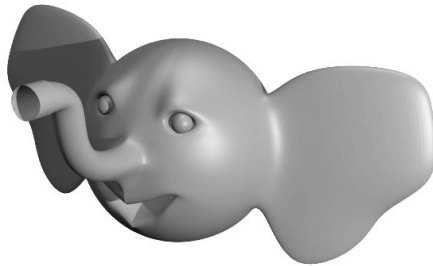


18. Перейдите к виду спереди. Проверьте, чтобы кнопка , которая ограничивает выделение только видимыми гранями (запрещает выделение на обратной стороне фигуры) была отключена. Включите режим работы с гранями и нажмите клавишу *B* (выделение прямоугольником) и выделите все грани левой части фигуры. Повернув 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.







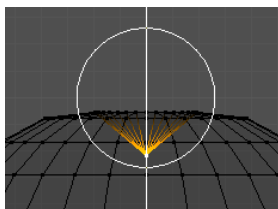
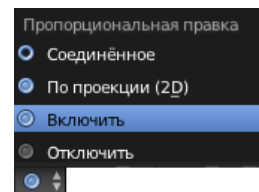
21. Добавьте две небольшие сферы и установите их внутрь глазных впадин.
22. Перемещая вершины, исправьте форму ушей так, как показано на рисунке:



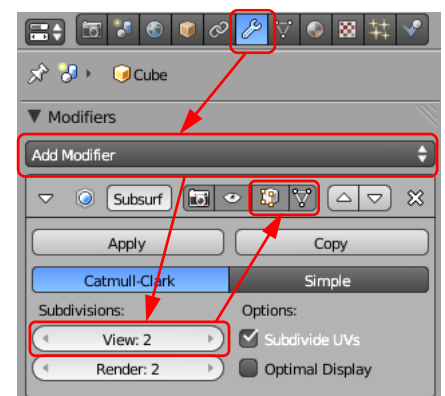
23. Сохраните модель под именем **slon.blend**.
24. Выберите положение камеры и источника света. Выполните рендеринг и сохраните картинку под именем **slon.png**.



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).

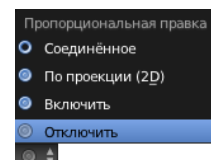


8. Включите режим просмотра поверхностей *Сплошной* (*Solid*, клавиша *Z*) и посмотрите, что получилось.
9. Перейдите к режиму работы с объектами (*Object Mode*, клавиша *Tab*) и примените модификатор *Подразделение поверхности* (*Subdivision surface*). Увеличьте число деления граней при просмотре до 2 (поле *View*). Для

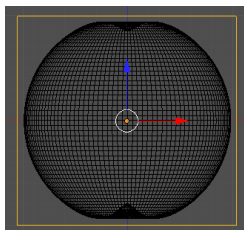


того, чтобы модификатор применялся в режиме редактирования сеточной модели, включите кнопки-выключатели  и .

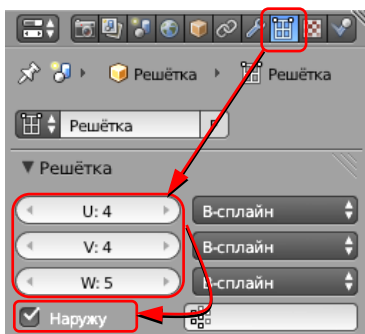
10. Отключите режим пропорционального редактирования, выбрав вариант *Отключить* (*Disable*) в списке элемента управления .



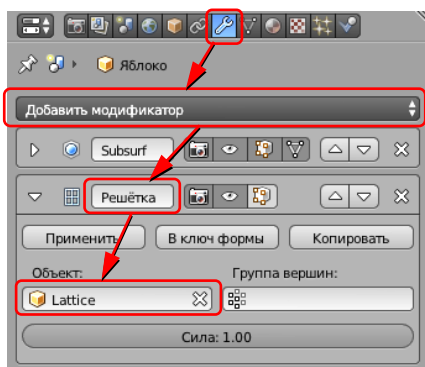
11. Переключитесь в режим выделения объектов (*Object Mode*, клавиша *Tab*) и включите просмотр каркаса (*Wireframe*, клавиша *Z*).
12. Нажмите клавиши *Shift-A* и создайте новую решётку (*Добавить – Решётка*, *Add – Lattice*). Разместите ее так, чтобы она охватывала все яблоко.



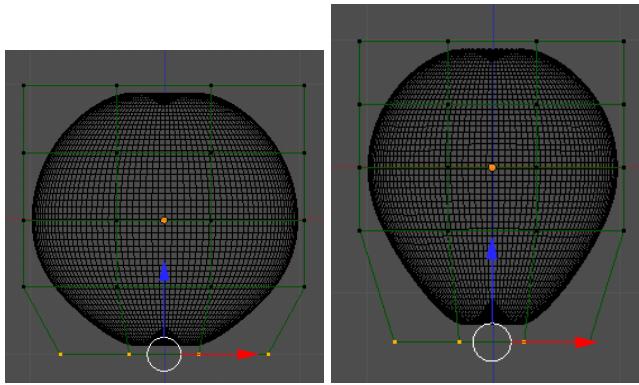
13. На странице свойств *Lattice* увеличьте количество разбиений решетки по осям. Отметьте флажок *Наружа* (*Outside*), чтобы все внутренние вершины и ребра решётки были удалены (остаются только вершины на наружных стенках).



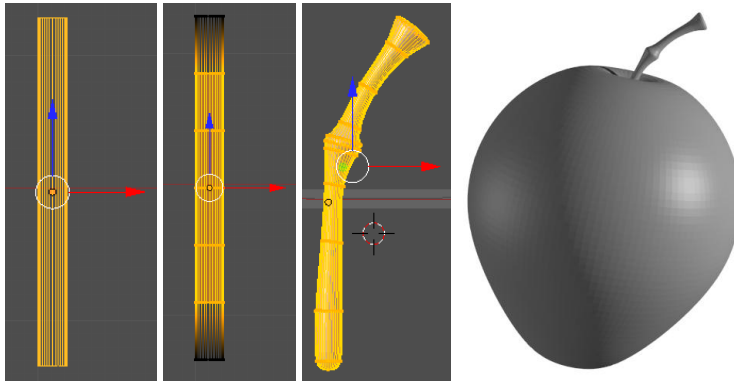
14. Выделите яблоко и примените к нему модификатор *Решётка*. В поле *Object* (объект) на странице свойств нужно выбрать созданную решётку (она по умолчанию имеет имя *Lattice*).



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.





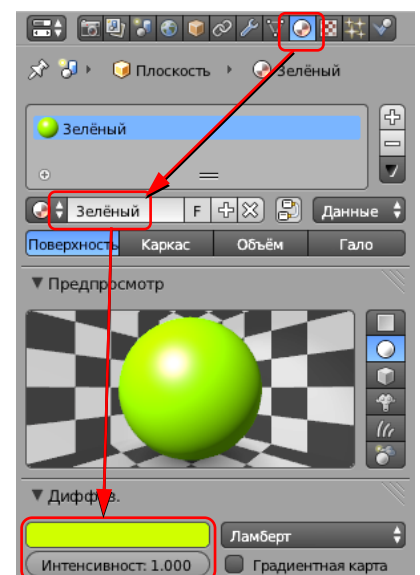
18. Перейдите в режим редактирования и с помощью инструмента *Разрезать петлей со сдвигом* (*Loop Cut and Slide*) добавьте еще 8-9 горизонтальных ребер по контуру. Используя перемещение, поворот и масштабирование сечений, постройте черенок яблока, как на рисунке.
19. Перейдите в режим работы с объектами 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

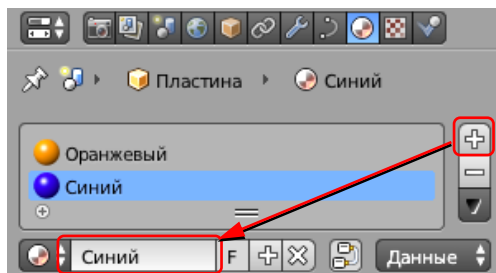
Материалы и текстуры

Материалы

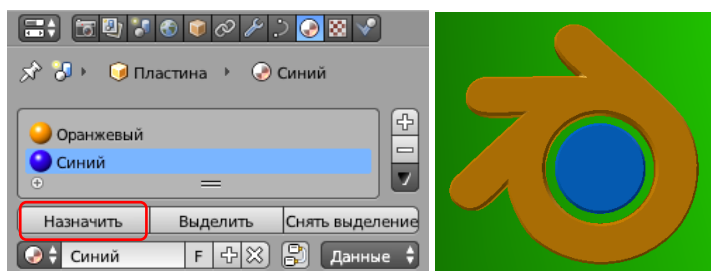
2. Запустите *Blender* и откройте файл **blender.blend**, в котором построен логотип *Blender* и плоскость.
3. С помощью элемента управления  в нижней части окна установите режим просмотра *Текстура* (*Textured*).
4. Выделите плоскость, перейдите на страницу свойств  *Материал* (*Material*) и щелкните 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*):






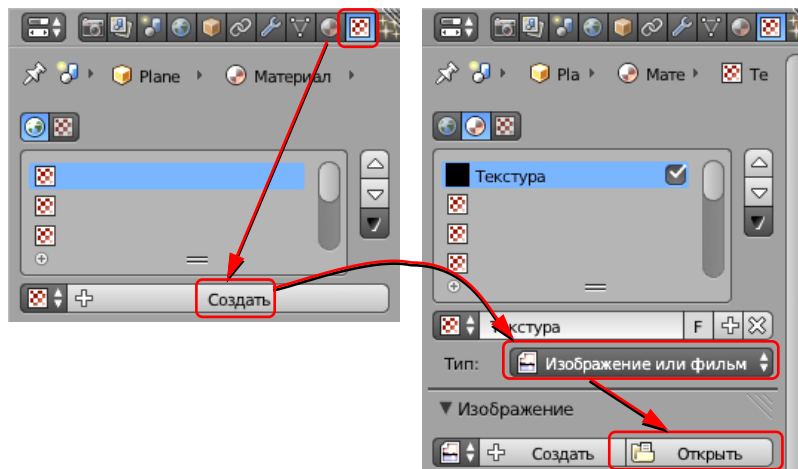
7. Перейдите в режим редактирование (*Edit Mode*, клавиша *Tab*). Выделите все вершины внутренней окружности и щелкните по кнопке *Назначить* (*Assign*). Средняя часть должна стать синей.





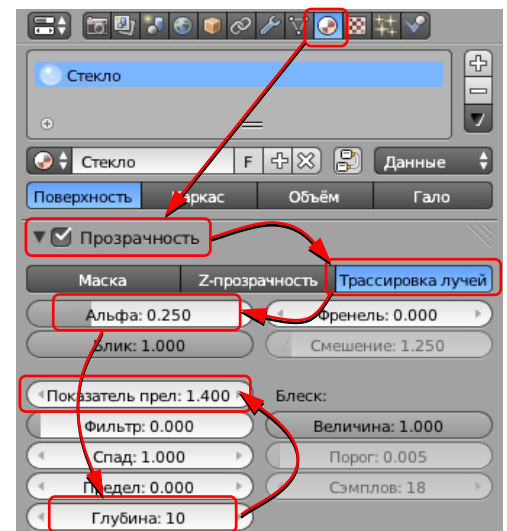
8. Сохраните модель под именем **blender1.blend** (меню *File* – *Save as...*). Выполните рендеринг и сохраните картинку в файле **blender1.png**.

Текстуры

25. Запустите *Blender* и откройте файл **wineglass.blend**, в котором на сцену помещены бокал и две плоскости.
26. С помощью элемента управления  в нижней части окна установите режим просмотра *Текстура* (*Textured*).
27. 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.
28. 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**.






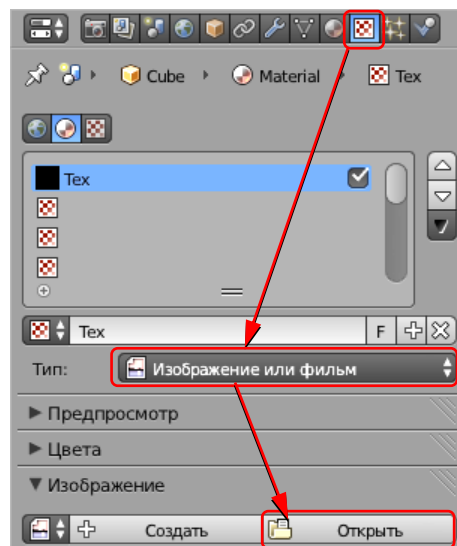
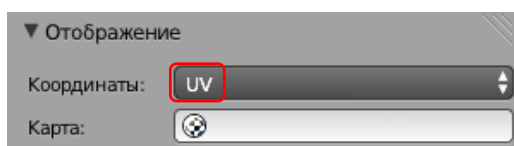
29. Щелкните по кнопке  слева от имени файла, чтобы рисунок был «упакован» в blend-файл.
30. Аналогично назначьте вертикальной плоскости новый материал с текстурой из файла **red-white.png**.
31. Выделите бокал, перейдите на страницу свойств  *Материал* (*Material*) и щелкните по кнопке *Создать* (*New*). Назовите его *Стекло* (*Glass*).
32. Включите флажок *Прозрачность* (*Transparency*), режим *Трассировка лучей* (*Raytrace*) и установите параметр *Альфа* (*Alpha*, непрозрачность) равным 0,25. Выполните рендеринг (это может занять около минуты).
33. Увеличьте значение параметра *Глубина* (*Depth*) до 10 и выполните рендеринг. Что изменилось?
34. Увеличьте значение параметра *Показатель преломления* (*IOR*, *Index of Refraction*) до 1.4 и выполните рендеринг. Что изменилось?
35. Сохраните модель под именем **wineglass1.blend**.
36. Сохраните картинку под именем **wineglass.png**.





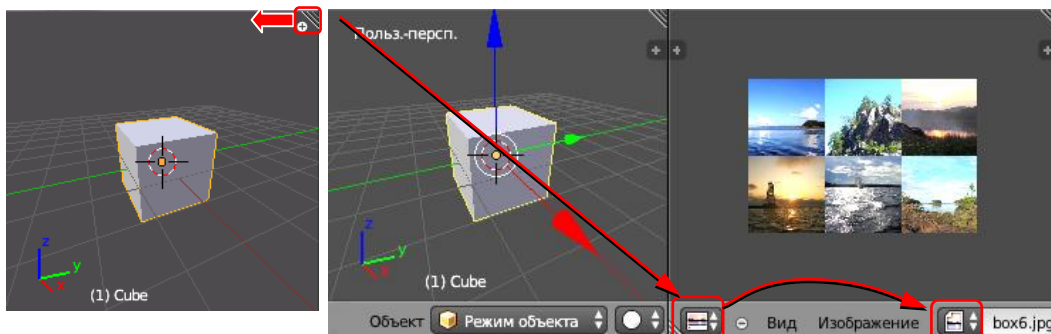
Практическая работа (дополнительная)




UV-развёртка (куб)

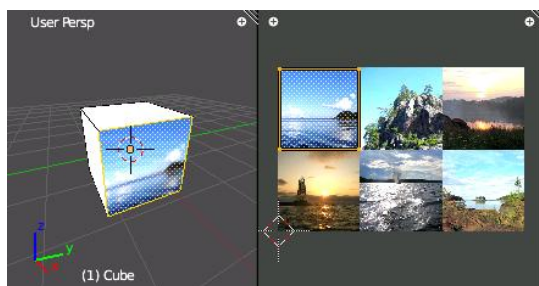
1. Запустите *Blender* и выделите куб, созданный автоматически.
2. Перейдите на страницу свойств  *Материал* (*Material*) и убедитесь, что для него автоматически создан новый материал.
3. Перейдите на страницу свойств  *Текстура* (*Texture*). В поле *Тип* (*Type*) выберите вариант *Изображение или видео* (*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**.



Демонстрационная версия промежуточной аттестации (практическая работа)

- 1) По каналу связи передаются сообщения, содержащие только семь букв: А, Б, В, Г, Д, Е и Ж. Для передачи используется двоичный код, удовлетворяющий условию Фано. Для буквы А используется кодовое слово 1; для буквы Б используется кодовое слово 01. Какова минимальная общая длина кодовых слов для всех семи букв?
- 2) Камера делает фотоснимки размером 640×480 пикселей. На хранение одного кадра отводится 250 Кбайт. Найдите максимально возможное количество цветов в палитре изображения.
- 3) Музыкальный фрагмент был записан в формате моно, оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла – 10 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате стерео (двухканальная запись) и оцифрован с разрешением в 3 раза выше и частотой дискретизации в 2,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи.
- 4) Данные объемом 60 Мбайт передаются из пункта А в пункт Б по каналу связи, обеспечивающему скорость передачи данных 2^{20} бит в секунду, а затем из пункта Б в пункт В по каналу связи, обеспечивающему скорость передачи данных 2^{23} бит в секунду. От начала передачи данных из пункта А до их полного получения в пункте В прошло 10 минут. Сколько времени в секундах составила задержка в пункте Б, т.е. время между окончанием приема данных из пункта А и началом передачи данных в пункт В?
- 5) Вася составляет 5-буквенные слова, в которых есть только буквы Б, А, Л, К, О, Н, причём буква Б используется в каждом слове хотя бы 1 раз. Каждая из других допус-

тимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

- 6) При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 12 символов и содержащий символы латинского алфавита (заглавные и строчные), десятичные цифры, а также не менее 6 специальных символов из набора \$, #, @, ^. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения. Для хранения сведений о 40 пользователях выделили 1080 байт. Сколько байт можно использовать для хранения дополнительных сведений о каждом пользователе?
- 7) Определите количество натуральных чисел, удовлетворяющих неравенству:
 $(170_8 + FE_{16}) \leq x \leq (200_8 + 11111111_2)$.
- 8) Сколько единиц в двоичной записи числа $8^{1341} - 4^{1342} + 2^{1343} - 1344$?
- 9) Решите уравнение $222_x + 4 = 1100_5$. Ответ запишите в троичной системе счисления
- 10) Логическая функция F задаётся выражением $(x \rightarrow y) \wedge (y \rightarrow z)$. На рисунке приведён фрагмент таблицы истинности функции F. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z.

?	?	?	F
1	0	0	0
1	0	1	1

В ответе напишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы.

- 11) В таблице приведены запросы и количество страниц, которые нашел поисковый сервер по этим запросам в некотором сегменте Интернета:

<i>Запрос</i>	<i>Количество страниц (тыс.)</i>
<i>декабрь & январь & февраль</i>	<i>113</i>
<i>декабрь & январь</i>	<i>225</i>
<i>декабрь & (январь / февраль)</i>	<i>645</i>

Сколько страниц (в тысячах) будет найдено по запросу

декабрь & февраль

- 12) Определите наименьшее натуральное число A, такое что выражение
 $(x \& 43 = 0) \vee ((x \& 49 = 0) \rightarrow (x \& A = 0))$

тождественно истинно (то есть принимает значение 1 при любом натуральном значении переменной x)?

- 13) Сколько различных решений имеет система уравнений?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5) = 1$$

$$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) \wedge (y_4 \rightarrow y_5) = 1$$

$$x_1 \vee y_1 = 1$$

где $x_1, x_2, \dots, x_5, y_1, y_2, \dots, y_5$ – логические переменные? В ответе не нужно перечислять все различные наборы значений переменных, при которых выполнено данное равенство. В качестве ответа нужно указать количество таких наборов.

- 14) В ячейки диапазонов C2:F6 и B3:B6 электронной таблицы записаны числа, как показано на рисунке. В ячейке A1 записали формулу $=E\$4 + \$D5$. После этого ячейку A1 скопировали в ячейку B2. Какое число будет показано в ячейке B2?

	A	B	C	D	E	F
1						
2			1	2	3	4
3		1	1	2	3	4
4		2	2	4	6	8
5		3	3	6	9	12
6		4	4	8	12	16

- 15) Дан фрагмент электронной таблицы:

	A	B	C
1	4	2	???
2	$=A1+C1$	$=A1+B1$	$=3*C1$



Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку? Известно, что все значения диапазона, по которым построена диаграмма, имеют один и тот же знак.

- 16) Два узла, находящиеся в одной сети, имеют IP-адреса 215.171.155.54 и 215.171.145.37. Укажите наибольшее возможное значение третьего слева байта маски сети. Ответ запишите в виде десятичного числа

Демонстрационная версия диагностического среза

- 1) В одной из кодировок Unicode каждый символ кодируется 16 битами. Вова написал текст (в нём нет лишних пробелов):

«Ёж, лев, слон, олень, тюлень, носорог, крокодил, аллигатор — дикие животные».

Ученик вычеркнул из списка название одного из животных. Заодно он вычеркнул ставшие лишними запятые и пробелы — два пробела не должны идти подряд.

При этом размер нового предложения в данной кодировке оказался на 16 байт меньше, чем размер исходного предложения. Напишите в ответе вычеркнутое название животного.

- 2) Валя шифрует русские слова (последовательности букв), записывая вместо каждой буквы её код:

А	Д	К	Н	О	С
01	100	101	10	111	000

Некоторые цепочки можно расшифровать не одним способом. Например, 00010101 может означать не только СКА, но и СНК. Даны три кодовые цепочки:

100101000
101111100
100111101

Найдите среди них ту, которая имеет только одну расшифровку, и запишите в ответе расшифрованное слово.

- 3) Между населёнными пунктами А, В, С, D, Е построены дороги, протяжённость которых (в километрах) приведена в таблице:

	А	В	С	D	Е
А		1			
В	1		2	2	7
С		2			3
D		2			4
Е		7	3	4	

Определите длину кратчайшего пути между пунктами А и Е. Передвигаться можно только по дорогам, протяжённость которых указана в таблице.

- 4) Доступ к файлу **slon.txt**, находящемуся на сервере **circ.org**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

- А) .txt
Б) ://
В) http
Г) circ
Д) /
Е) .org
Ж) slon

- 5) В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» — символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

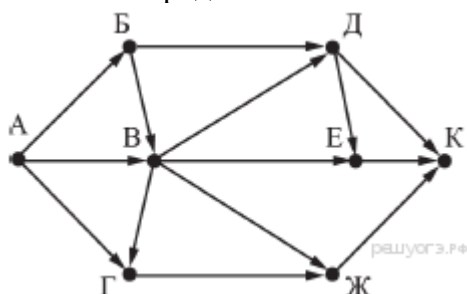
Запрос	Найдено страниц (в тысячах)
Рыбак Рыбка	780
Рыбак	260
Рыбак & Рыбка	50

Какое количество страниц (в тысячах) будет найдено по запросу

Рыбка?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

- 6) На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж и К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К, проходящих через город В?



- 7) Среди приведённых ниже трёх чисел, записанных в различных системах счисления, найдите максимальное и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

23_{16} , 32_8 , 11110_2 .

- 8) У исполнителя Альфа две команды, которым присвоены номера:

1. прибавь 1;

2. умножь на b

(b — неизвестное натуральное число; $b \geq 2$).

Выполняя первую из них, Альфа увеличивает число на экране на 1, а выполняя вторую, умножает это число на b . Программа для исполнителя Альфа — это последователь-

ность номеров команд. Известно, что программа 11211 переводит число 6 в число 82. Определите значение b .

Критерии выставления отметок за практические работы

Отметка «5» («отлично») ставится в случае, если обучающийся:

самостоятельно выбрал и подготовил для работы необходимое оборудование;
выполнил работу в полном объеме с соблюдением рациональной последовательности проведения опытов и измерений и с безусловным соблюдением правил техники безопасности;

научно, грамотно и логично описал ход эксперимента (опыта) и сформулировал выводы из результатов эксперимента (опыта);

в представленном отчете правильно и аккуратно выполнил все записи, таблицы, рисунки, чертежи, графики, вычисления и выводы;

правильно выполнил анализ погрешностей;

проявил организационно-трудовые умения (поддерживал чистоту рабочего места и порядок на столе, экономно использовал расходные материалы).

Отметка «4» («хорошо») ставится в случае, если обучающийся:

правильно определил цель работы;

самостоятельно выбрал и подготовил для работы необходимое оборудование;

выполнил работу в полном объеме с безусловным соблюдением правил техники безопасности;

но:

опыт проводил в условиях, не обеспечивающих достаточной точности измерений;

либо допустил два-три недочета;

либо допустил не более одной негрубой ошибки и одного недочета;

либо в описании наблюдений допустил неточности, сделал неполные выводы из результатов эксперимента (опыта).

Отметка «3» («удовлетворительно») ставится в случае, если обучающийся:

правильно определил цель работы, выполнил работу правильно не менее чем наполовину с безусловным соблюдением правил техники безопасности, однако объем выполненной части таков, что позволяет получить правильные результаты и выводы по основным, принципиально важным задачам работы;

но:

подбор оборудования, а также работы по началу эксперимента (опыта) провел с помощью педагога;

либо эксперимент (опыт) проводил в нерациональных условиях, что привело к получению результатов с большой погрешностью;

либо допустил не более двух грубых ошибок;

либо допустил не более одной грубой, одной негрубой ошибки и одного недочета;

либо допустил не более трех негрубых ошибок;

либо допустил не более одной негрубой ошибки и трех недочетов;

либо допустил не более пяти недочетов;

либо не выполнил совсем или выполнил неверно анализ погрешностей.

Отметка «2» («неудовлетворительно») ставится в случае, если обучающийся:

не смог определить цель работы и подготовить для работы необходимое оборудование, выполнил работу менее чем на половину и объем выполненной части работы не позволяет сделать правильных выводов;

либо допустил однократное нарушение правил техники безопасности;
либо эксперимент (опыты), измерения, вычисления, наблюдения производились неправильно;

либо допустил в работе число ошибок и недочетов превосходящее норму, при которой может быть выставлена отметка «3» («удовлетворительно»).

Отметка «1» («плохо») ставится в случае, если обучающийся:

не приступал к выполнению работы;

либо не соблюдал или неоднократно грубо нарушал требования правил техники безопасности и был отстранен от выполнения работы.

Примечание:

1. Педагог вправе поставить обучающемуся отметку на один балл выше той, которая предусмотрена нормами, в случае, когда обучающийся показал оригинальный и наиболее рациональный подход к выполнению работы.

2. Отметки с анализом доводятся до сведения обучающихся, как правило, не позднее недели после сдачи отчета.