

ИНФОРМАТИКА

класс

9



24 ЧАСА
ДО ЭКЗАМЕНА
И ГИА

Т.Е. Чуркина

ОТВЕТЫ
НА

ЭКЗАМЕНАЦИОННЫЕ
БИЛЕТЫ



+ ШПАРГАЛКА

РУССКИЙ ЯЗЫК ФИЗИКА ХИМИЯ ИСТОРИЯ
И ЯЗЫК ИСТОРИЯ ОБЩЕСТВОЗНАНИЕ
ИЯ ИСТОРИЯ РУССКИЙ ЯЗЫК ФИЗИКА
ГЕОГРАФИЯ ИНФОРМАТИКА БИОЛОГИЯ
ЯК ФИЗИКА ХИМИЯ ГЕОМЕТРИЯ ОБЖ



Т.Е. Чуркина

ИНФОРМАТИКА

ОТВЕТЫ
НА ЭКЗАМЕНАЦИОННЫЕ
БИЛЕТЫ

9 класс



ШПАРГАЛКИ К БИЛЕТАМ

Издательство
«ЭКЗАМЕН»

МОСКВА
2013

УДК 372.8:002

ББК 74.263.2

Ч 93

Чуркина, Т.Е.

Ч 93 Информатика. Ответы на экзаменационные билеты 9 класс / Т.Е. Чуркина.— М.: Издательство «Экзамен» 2013. — 126, [2] с. (Серия «24 часа до экзамена и ГИА»)

ISBN 978-5-377-05384-2

Данное пособие полностью соответствует федеральному государственному образовательному стандарту (второго поколения).

В данном пособии приводятся ответы на все вопросы экзаменационных билетов по информатике, предлагаемых Министерством образования и науки РФ для проведения устной итоговой аттестации выпускников 9 классов общеобразовательных школ. В ответах на практические вопросы рассмотрены примеры решения типовых задач и примеры выполнения типовых заданий.

Предлагаемые ответы полностью удовлетворяют требованиям предъявляемым на экзаменах в школах, и помогут школьникам быстро и эффективно подготовиться к экзаменам, систематизировать и укрепить свои знания.

В пособии содержатся шпаргалки к билетам.

Для простого и эффективного использования шпаргалки разрежьте каждую страницу на четыре части по пунктирной линии. Сложите полученные листы по порядку номеров — верхний левый, верхний правый, нижний левый, нижний правый. Для удобства использования можно скрепить получившуюся стопку степлером или скрепкой в верхнем левом углу.

Пособие предназначено преподавателям и учащимся 9 классов общеобразовательных школ.

Приказом № 729 Министерства образования и науки Российской Федерации учебные пособия издательства «Экзамен» допущены к использованию в общеобразовательных учреждениях.

УДК 372.8:002

ББК 74.263.2

Формат 84х108/32. Гарнитура «Таймс».

Бумага газетная. Уч.-изд. л. 7,18. Усл. печ. л. 6,72.

Тираж 20 000 экз. Заказ № 12402.

ISBN 978-5-377-05384-2

© Чуркина Т.Е., 2013

© Издательство «ЭКЗАМЕН», 2013

СОДЕРЖАНИЕ

Предисловие	4
Часть I. Ответы на теоретические задания билетов	5
Билет 1	5
Билет 2	6
Билет 3	8
Билет 4	11
Билет 5	13
Билет 6	14
Билет 7	17
Билет 8	20
Билет 9	22
Билет 10	25
Билет 11	28
Билет 12	30
Билет 13	32
Билет 14	34
Билет 15	36
Билет 16	39
Билет 17	41
Билет 18	42
Билет 19	44
Билет 20	47
Часть II. Примерные варианты ответов на практические задания билета	49
Шпаргалка	109

ПРЕДИСЛОВИЕ

Предлагаемое пособие адресовано учащимся 9 классов общеобразовательных школ для самостоятельной подготовки к итоговому экзамену по предмету «Информатика и ИКТ».

Пособие содержит ответы на типовые экзаменационные билеты разработанные Федеральной службой по надзору в сфере образования и науки для сдачи экзамена по выбору по предмету «Информатика и ИКТ» за 9 класс. Полный экзаменационный комплект состоит из 21 билета, в каждом из которых по два вопроса. Первый вопрос является теоретическим, второй практическим (заданием или задачей).

В первой части пособия излагаются ответы на теоретические вопросы и приводятся темы задач, рекомендованных для комплектации к каждому вопросом по теории. Выбор конкретной задачи, соответствующей уровню ученика или профилю школы, осуществляет учитель. Список типовых задач с подробным разбором их решений приведен во второй части пособия. Задачи разделены по тематикам (9 основных разделов). Звездочками над номерами задач обозначен уровень сложности задачи (* — низкий уровень сложности, **, *** — средний уровень сложности, ****, ***** — задачи повышенной сложности).

Решение задач и выполнение заданий практической части билета подразумевает использование компьютера и программных средств, установленных на нём (Word, Excel, Power Point, антивирусных программ, программ-архиваторов и др.). При самостоятельной подготовке практической части билета школьнику рекомендуется дома на компьютере неоднократно проделать все действия, описанные в пособии.

Решение всех типовых задач по программированию приведено на двух наиболее популярных для изучения в российских школах языках программирования Pascal и Basic. Все решения задач по программированию, приведенные на языке Pascal, снабжены подробными разъяснениями-комментариями, набранными в фигурных скобках (согласно синтаксису языка Pascal). Для практического выполнения заданий по программированию школьнику необходимо:

- запустить программную оболочку, входящую в систему программирования;
- набрать в текстовом редакторе этой оболочки текст программы;
- запустить компилятор с языка программирования (для языка Pascal) или интерпретатор (для языка Basic);
- исправить синтаксические ошибки, если таковые обнаружены;
- запустить программу на исполнение.

Описания заданий, выполняемых в программах Excel, Access, Word, PowerPoint, входящих в Microsoft Office, дано для версии Microsoft Office 2003.

ЧАСТЬ I

ОТВЕТЫ НА ТЕОРЕТИЧЕСКИЕ ЗАДАНИЯ БИЛЕТОВ

БИЛЕТ 1

1. Понятие информации. Виды информации. Роль информации в живой природе и в жизни людей. Язык как способ представления информации: естественные и формальные языки. Основные информационные процессы: хранение, передача и обработка информации.

Информация (от латинского «разъяснение», «набор сведений») — это связанные между собой, неизвестные до их получения сведения, дающие представление о явлениях или объектах. Информация бывает:

- *актуальная* (информация ценная в данный момент времени),
- *достоверная* (информация, полученная без искажений),
- *понятная* (информация, выраженная на языке понятном тому, кому она предназначена),
- *полная* (информация, достаточная для принятия правильного решения или понимания),
- *полезная* (полезность информации определяется субъектом, получившим информацию в зависимости от объёма возможностей её использования).

Восприятие информации может идти через органы слуха, зрения, обоняния, вкуса, осязания (органолептическая информация). Некоторые из этих видов информации могут воспринимать животные и растения.

Полученная человеком информация даёт ему знания об окружающем мире. Современный человек должен уметь получать информацию, её обрабатывать, передавать, оценивать её качества (полезность, достоверность и т.д.), уметь взаимодействовать в информационном поле с другими людьми. Современные компьютерные технологии предоставили человеку широкие возможности использования больших объёмов информации, находящихся на носителях на значительном удалении от самого человека.

Основными информационными процессами являются обработка, хранение и передача информации.

Передача информации обеспечивает её распространение. Устройствами передачи информации в современном мире являются телефон, радио, телевизионные передатчики, телекоммуникативные сети. Все это, так называемые, *каналы передачи информации*. Для

отправки информации необходимо на одном конце такого канала иметь устройство-передатчик, на другом — приемник. Одной из важнейших проблем в современном мире является обеспечение сохранности информации при её передаче, так как различные помехи на линиях связи могут исказить информацию и даже вести к её утере. Другая проблема — защита информации от хищения.

Обработка информации — это преобразование ранее полученной информации к иной форме или к иному содержанию. Преобразование формы информации означает, что человек по своему усмотрению может перевести информацию в символьную, текстовую, графическую или звуковую форму.

Хранение информации. Цель хранения информации — её передача во времени, а также обеспечение возможности её многократного использования.

2. Построение алгоритма (основные алгоритмические структуры) и его реализация в среде учебного исполнителя. Демонстрация полученного алгоритма в среде учебного исполнителя.

БИЛЕТ 2

1. Измерение информации: содержательный и алфавитный подходы. Единицы измерения информации.

В настоящее время существуют два подхода к измерению информации: содержательный и алфавитный.

При *содержательном подходе* объём информации приравнивается к совокупности знаний человека, и получение им новой информации ведет к расширению его знаний. Поэтому, если полученная объектом информация является для него новой и ранее не известной, то говорят, что сообщение информативно, (т.е. содержит ненулевую информацию). Поскольку каждый человек в силу своей специальности, возраста, уровня образования, начитанности, круга увлечений, места жительства и многих других факторов обладает своим специфическим набором сведений, то информативность сообщения — понятие субъективное (т.е. зависит от конкретного человека, воспринимающего данное сообщение). При восприятии информации человеком важна её понятность, т.е. информация должна быть в виде, доступном для понимания, иначе она перестаёт быть информативной. Для того чтобы информация была понятной, она должна быть выражена на языке, доступном для понимания человеку. При этом во многих случаях от него самого требуется иметь оп-

ределённые знания, необходимые для получения вновь поступающей информации. Данное свойство положено в основу процесса обучения человека, т.е. последовательного процесса получения информации от простого к более сложному.

Для количественного измерения поступающей информации была введена специальная единица измерения *бит*, являющаяся основной и наименьшей единицей измерения объёма информации. Другими (производными) единицами информации являются: байт = 8 бит, Кбайт = 1024 байта, Мбайт = 1024 Кбайта, Гбайт = 1024 Мбайта. Бит — это количество информации в сообщении, уменьшающем неопределённость знаний в два раза. Очевидно, что два сообщения, следующие друг за другом и уменьшающие неопределённость, имеющуюся на момент их получения, в 2 раза каждое (т.е. в целом в 4 раза), дадут 2 бита информации, три таких сообщения — 3 бита и т.д. В качестве примеров можно привести игру бинго или игру в орлянку. В первом случае ведущий случайным образом вытаскивает из мешочка специальный пронумерованный шар (например, один из 64), и выпадает одно из значений от 1 до 64, во втором при подбрасывании монеты выпадает либо орёл, либо решка. При бросании монеты человек получает одно из двух возможных информационных сообщений, что и является уменьшением неопределённости знаний в два раза (т.е. говорят, что получен один бит информации). При игре бинго исходов шестьдесят четыре, и неопределённость знания уменьшается в шестьдесят четыре раза, и тогда $2^x = 64$, где $x = 6$ — выраженное в битах количество полученной информации, являющееся решением приведённого уравнения. Заметим, что в общем случае полученное информационное сообщение может нести дробное количество информации; количество информации будет выражено целым числом, только если неопределённость знаний уменьшилась в число раз, являющееся целочисленной неотрицательной степенью числа 2.

Итак, количество информации i , содержащееся в сообщении о том, что произошло одно из N равновероятных событий, есть решение уравнения $N = 2^i$.

Другой подход к измерению информации, *алфавитный*, рассматривает сообщение как дискретную последовательность знаков из специального набора символов (алфавита), при этом количество информации (в битах) в данном сообщении подсчитывается как произведение количества символов в сообщении на количество битов, необходимых для кодирования одного символа в алфавите. В этом случае анализ содержания сообщения не происходит.

Полное количество символов алфавита N называется его *мощностью*. При этом в знакомых нам языках учитываются не только бу-

квы, но и знаки препинания, пробелы, специальные символы (№, \$, % и другие).

Для представления информации в различных технических устройствах используется её преобразование в последовательность нулей и единиц. При этом информация, закодированная знаками некоторого алфавита по строгому однозначному алгоритму, ставящему в соответствие букве алфавита двоичное число, может быть переведена в двоичный код (и при необходимости обратно). При этом минимальное количество битов, необходимое для кодирования одного символа алфавита, определяется как корень уравнения $N = 2^i$, (в случае если N не является степенью двойки, то следует на место N в уравнении поставить ближайшее большее N число, являющееся степенью числа 2).

Для примера посчитаем количество информации в сообщении «Жил старик со своею старухой у самого синего моря; они жили в ветхой землянке ровно тридцать лет и три года.», если предполагать, что каждый символ данного алфавита закодирован двухбайтным словом. Длина фразы составляет 108 символов (с учётом пробелов и знаков препинания!). Тогда информационный объём сообщения можно посчитать как $108 \cdot 2 = 216$ байта или, что то же самое, $216 \cdot 8 = 1728$ бита.

Отметим, что алфавитный подход к измерению информации, применяемый в технических устройствах, не только позволяет пользователю работать на ЭВМ с текстовой и графической информацией, но и оценивать объёмы необходимой для сообщения памяти на носителях информации или необходимые технические характеристики устройств для передачи данных на расстояние.

2. Создание и редактирование текстового документа (исправление ошибок, удаление или вставка текстовых фрагментов), в том числе использование элементов форматирования текста (установка параметров шрифта и абзаца, внедрение заданных объектов в текст).

БИЛЕТ 3

1. Дискретное представление информации: двоичные числа; двоичное кодирование текста в памяти компьютера. Информационный объём текста.

Вся информация, хранящаяся и обрабатываемая компьютером, представляется двоичным кодом, т.е. последовательностью из нулей и единиц. Для такого представления, а также для обратного преобразования кода в сообщение, понятное человеку, применяется *кодирование и декодирование* информации соответственно. В процессе

кодирования происходит преобразование входной информации (вводимой, например, человеком с клавиатуры компьютера) в двоичный код, в процессе декодирования происходит обратное действие. Выбор двоичного представления информации в ЭВМ обусловлен конструктивными особенностями вычислительной техники. Дело в том, что ячейка памяти, хранящая один бит информации и являющаяся минимальной в памяти компьютера, может в каждый момент времени находиться в одном из состояний: намагниченном (обозначается 1) или ненамагниченном (0). Передача информации идёт посредством электронных импульсов: 1 — есть сигнал, 0 — нет сигнала. Такое представление информации является дискретным (т.е. прерывистым), поскольку объём представленной информации обязательно равен целому числу бит.

Для получения компьютерного представления числа, записанного в привычной нам десятичной системе счисления, необходимо осуществить его перевод в двоичную систему. С этой целью для целых положительных чисел используется метод последовательного деления целой части частного на 2. Так, например, переведем в двоичную систему число 23:

$$23 : 2 = 11 + 1 : 2, \text{ т.е. в остатке } 1;$$

$$11 : 2 = 5 + 1 : 2, \text{ т.е. в остатке } 1;$$

$$5 : 2 = 2 + 1 : 2, \text{ т.е. в остатке } 1;$$

$$2 : 2 = 1 + 0 : 2, \text{ т.е. в остатке } 0;$$

$$1 : 2 = 0 + 1 : 2, \text{ т.е. в остатке } 1.$$

В частном оказалось число, меньшее делителя, поэтому процесс последовательного деления останавливаем. Далее записываем число 23 в двоичной системе счисления: последовательность из нулей и единиц, полученных в остатках от деления, но взятые в обратном порядке, по отношению к тому, как они возникали в процессе деления, т.е. $23_{10} = 10111_2$ (индекс внизу обозначает основание системы счисления, в которой записано число). На компьютере для хранения числа выделяется k -разрядная ячейка памяти, наиболее распространенной является 2-байтная ячейка, содержащая 16 бит. В этом случае при записи числа 23 в компьютер его двоичный эквивалент будет дополнен слева незначащими нулями, т.е. представление числа 23 на ЭВМ будет иметь вид 0000000000010111 (всего 16 нулей и единиц).

Как правило, в компьютерном представлении целого числа самый левый разряд означает его знак (0 — если число положительное, 1 — если число отрицательное). При этом для представления отрицательных чисел используется, так называемый, дополнительный код. Для его получения необходимо определить двоичное представление модуля числа, произвести инверсию (заменить в записи числа нули еди-

нищами, а единицы нулями и прибавить 1). Таким образом, для представления числа -23 нам необходимо инвертировать его двоичную запись 0000000000010111: получим 111111111101000. Прибавляя единицу, окончательно имеем 111111111101001.

Для представления вещественного числа в памяти компьютера отводится 4 байта (если требуется большая точность, то 8 байт). При этом число A представляется в виде произведения мантиссы на некоторую степень двойки: $A = m \cdot 2^n$, где m — мантисса (значащие разряды), n — порядок (т.е. положение запятой относительно первого разряда мантиссы). При этом в памяти компьютера среди выделенных 4 байт для представления числа отдельно отводятся биты для хранения мантиссы, порядка и по одному биту для хранения знака числа и знака порядка.

Рассмотрим двоичное кодирование текстовой информации на ЭВМ. Поскольку компьютер может воспринимать информацию только в виде двоичного кода, то для представления в компьютерной памяти текстов производится кодирование символов алфавита некоторыми целыми числами, для которых компьютерное представление нам уже известно. Для удобства кодирования символов целыми десятичными числами, как правило, составляются кодировочные таблицы, которые утверждены международными стандартами. В кодовых таблицах латинские прописные и строчные буквы располагаются в алфавитном порядке, а цифры упорядочены по возрастанию (так называемый принцип последовательного кодирования алфавитов).

В зависимости от числа битов отведенных под кодирование символов в памяти ЭВМ кодировки бывают 8- и 16-разрядные. 8-разрядной является, например, кодировка ASCII, 16-разрядной — кодировка Unicode, которые позволяют закодировать соответственно 256 и 65536 различных символов.

Если пользователю известна система кодирования, используемая в компьютере (а вернее количество разрядов отведенных под хранение одного символа), то ему не составит труда оценить объем информации, содержащейся в данном тексте. Приведем пример. Некоторая брошюра содержит 70 страниц печатного текста, на каждой странице в среднем 40 строк, в строке 60 символов (включая пробелы), используется 8-разрядная кодировка ASCII. В этом случае объем информации в книге будет составлять $70 \cdot 40 \cdot 60 = 168\ 000$ байт или, что то же самое, $168\ 000 \cdot 8 = 1\ 344\ 000$ бит.

2. Создание и обработка графических изображений средствами графического редактора. Ввод изображения через сканер или с цифрового фотоаппарата. Простейшая обработка цифрового изображения.

БИЛЕТ 4

1. Дискретное представление информации: кодирование цветного изображения в компьютере (растровый подход). Представление и обработка звука и видеонизображения. Понятие мультимедиа.

Существует несколько технических систем записи цвета, которые иногда называют цветовыми моделями. Рассмотрим основные из них.

RGB (Red-Green-Blue — *красный-зелёный-синий*). Цвет идентифицируется набором из трёх упорядоченных целых чисел, значения которых меняются от 0 до 255. Так красному соответствует набор (255,0,0), зелёному — (0,255,0), синему — (0,0,255), белый цвет состоит из всех трёх цветов максимальной интенсивности, т.е. (255,255,255); чёрный получается при выключении всех трёх базовых цветов — (0,0,0). Все другие цвета и оттенки получаются смешиванием трёх базовых цветов различной интенсивности (чем больше значение, тем ярче цвет). Для кодировки цвета конкретной точки изображения выделяются три байта памяти компьютера, а именно: для кодирования числа, отвечающего за интенсивность красного цвета отводится первый байт, для кодирования зелёного — второй, для кодирования синего — третий. RGB является аддитивной моделью, так как при наложении одной составляющей цвета на другую не только меняется его тон, но и увеличивается яркость суммарного цвета. Цветовая модель RGB применяется в телевизорах и в цветных мониторах компьютеров.

СМЯК (Cyan-Magenta-Yellow-black — *голубой-пурпурный-жёлтый-чёрный*). Цвет идентифицируется набором из четырех упорядоченных целых чисел, значения которых меняются от 0 до 100 и соответствуют интенсивности (в процентах) базовых цветов голубого, пурпурного, жёлтого и чёрного. Эти цвета (кроме чёрного) получаются при вычитании основных цветов, используемых в модели RGB, из белого цвета, чёрный цвет задается отдельно. Данная цветовая модель является субтрактивной, поскольку увеличение яркости основного цвета (краски) влечет снижение яркости картинки в целом. Такая цветовая модель получила наибольшее распространение в цветной полиграфии.

HSB — цветовая модель, наиболее удобная для восприятия человеком. В ней используются три параметра, характеризующие цвет: *тон* (Hue), *насыщенность* (Saturation), *яркость* (Brightness). Под *тоном* цвета обычно понимают близость оттенка к тем или иным цветам спектра. *Насыщенность* — это степень выраженности

тона в цвете. **Яркость** определяется уровнем действующего на глаз излучения.

В настоящее время существует большое количество программ, называемых, **графическими редакторами**, которые предоставляют человеку широкие возможности для работы как с цветными, так и с монохромными изображениями, позволяют их редактировать, преобразовывать, обрабатывать.

Отметим, что количество оттенков цвета, которые можно закодировать, обусловлено объёмом памяти, выделяемой для представления цвета каждой конкретной точки. Если выделяется один байт, то возможно закодировать 256 различных цветов, если 2 байта — то 65536. Чем больше объём выделяемой под цвет точки памяти, тем выше качество изображения.

Как известно, **звук** представляет собой волну, распространяющуюся в упругой среде. Звук обладает музыкальным *тоном* (т.е. определённой частотой волны), *длительностью* и *силой* (по-другому, *интенсивностью* или громкостью), зависящей от амплитуды колебаний. Для записи звука применяется **дискретизация**, т.е. в течение некоторого промежутка времени через определённые временные интервалы производится измерение амплитуды и её кодирование двоичным числом. Устройство, выполняющее такое преобразование, носит название **аналого-цифрового преобразователя**. Устройство, осуществляющее обратный процесс преобразования кода в звук, называется **цифроаналоговым преобразователем** и используется в комплексе с фильтрами, усилителями и динамиками, осуществляющими сглаживание ступенчатого сигнала, его усиление и воспроизведение. Качество воспроизведения записанного звука зависит от частоты дискретизации и от объёма памяти, выделенного для хранения его амплитуды. Сократить объём памяти, занятой звуковой записью, возможно, если хранить информацию не в виде отдельных данных, а записывать разность между двумя соседними значениями, которые часто отличаются друг от друга не слишком сильно, а следовательно, происходит экономия памяти компьютера. В настоящее время принято записывать звуковые (музыкальные) файлы в формате MIDI (музыка), в котором файл с музыкальной записью представляет собой последовательность закодированных сообщений синтезатору звука, разделённых также закодированными паузами. Сообщение синтезатору звука представляет собой ряд команд, задающих высоту, тембр и громкость звука.

Для записи **видеоизображения**, также как и для записи звука, применяется дискретизация. Дело в том, что при воспроизведении последовательности картинок с частотой более 15 кадров в секунду

человеческий глаз воспринимает эту последовательность как движение. Таким образом, принято кодировать видео изображение последовательностью стационарных картинок, отвечающих различным фазам движения объекта, воспроизводимыми с достаточно большой частотой. Запись изображения на компьютерные носители появилась сравнительно недавно, так как сама по себе требует больших объёмов памяти. Объём памяти, занимаемый видеозаписью, возможно сократить за счет хранения не всех следующих друг за другом кадров, а изменений к ним.

Возможность записи на компьютер звука, изображения и видеоизображения привела к созданию мультимедиа. *Мультимедиа* — это многофункциональная компьютерная система, включающая совокупность компьютерных продуктов в нескольких видах: изображение, звук, анимацию, текст и т.п., совмещаемая в интерактивном режиме с другими системами и используемая для разных целей: для создания базы данных справочно-энциклопедической информации, для её накопления, обработки, хранения, издания и т.п.

2. Работа с файловой системой, с графическим интерфейсом (выполнение стандартных операций с файлами: создание, копирование, переименование, удаление). Организация индивидуального информационного пространства (настройка элементов рабочего стола, проверка на вирусы, использование архиватора).

БИЛЕТ 5

1. Процесс передачи информации, источник и приёмник информации, канал передачи информации. Скорость передачи информации.

Передача информации обеспечивает её распространение. В процессе передачи информации осуществляется следующая цепочка: источник информации $\Rightarrow$ кодирующее устройство $\Rightarrow$ канал связи $\Rightarrow$ декодирующее устройство $\Rightarrow$ получатель. Источником информации может быть как техническое устройство, так и сам человек. *Кодирующее устройство* переводит исходное сообщение в пригодный для передачи по данному каналу связи вид. Примерами кодирующих устройств являются микрофон, видеокамера, фотоаппарат и др. *Декодирующее устройство* переводит переданное сообщение из кода в вид, понятный человеку (получателю). Естественными такими устройствами служат человеческое ухо и глаз.

Устройствами передачи информации в современном мире являются телефон, радио, телевизионные передатчики, телекоммуникационные сети. Все это, является *каналами передачи информации*, по которым информация передается посредством сигналов (механических, электрических, тепловых). Для отправки информации необходимо на одном конце такого канала иметь устройство-передатчик, на другом — приёмник.

Одной из важнейших проблем в современном мире является обеспечение сохранности информации при её передаче, так как различные помехи на линиях связи могут искажать информацию и даже вести к её утере. Другая проблема — *защита информации* от хищения. Для решения всех проблем, связанных с кодированием и декодированием информации, создана специальная наука *криптография*, она же создает методы защиты информации от понимания посторонними людьми (т.е. разрабатывает искусственные языки, понятные ограниченному кругу лиц).

Для характеристики скорости перемещения данных из одной области памяти в другую или с одного устройства на другое вводится понятие *скорости передачи данных*. Основной единицей измерения является бит/сек (бит в секунду). Производные единицы байт/сек=8 бит/сек, Кбайт/сек=1024 байт/сек. Характеристикой качества канала связи является его *пропускная способность* (т.е. максимальная скорость передачи данных).

2. Создание мультимедийной презентации на основе шаблонов. Выбор типа разметки слайда, применение таблицы оформления, цветовых схем и эффектов анимации. Показ презентации с использованием автоматической смены слайдов.

БИЛЕТ 6

1. Понятие алгоритма. Исполнитель алгоритма. Система команд исполнителя (на примере учебного исполнителя). Свойства алгоритма. Способы записи алгоритмов; блок-схемы.

Алгоритм — описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов. Процесс разработки алгоритма для решения задачи называется *алгоритмизацией*.

Основными свойствами алгоритма являются следующие: дискретность, детерминированность, результативность, конечность, массовость, корректность.

1) **Дискретность** алгоритма означает, что процесс решения задачи состоит из последовательности отдельных шагов, каждый из которых называется *командой*. Выполнение каждой новой команды начинается только после завершения предыдущей.

2) **Детерминированность** (определённость) алгоритма означает, что результат работы алгоритма для одного и того же набора начальных данных, будет оставаться постоянным вне зависимости от времени и места запуска программы, поскольку каждая команда алгоритма интерпретируется однозначным образом и выполняется в строго определённом случае. Наличие ошибок в алгоритме может привести к нарушению данного свойства.

3) **Результативность** алгоритма означает, что после выполнения команд, составляющих правильный алгоритм, должен получиться определённый результат. Завершение алгоритма должно произойти за конечное число шагов (*свойство конечности*). Данные свойства могут нарушаться при наличии ошибок в алгоритме.

4) **Массовость** алгоритма означает, что один и тот же алгоритм может быть применён для решения целого класса сходных задач, отличающихся исходными данными, которые выбираются из множества значений, допустимых для данного алгоритма (при этом в качестве исходных данных используются переменные). Массовость алгоритма подразумевает результативность его работы для любого набора начальных данных из некоторого допустимого множества (*свойство корректности*).

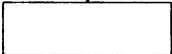
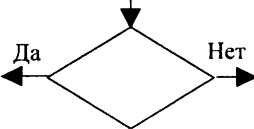
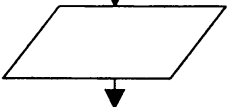
Алгоритмы бывают:

- **линейные** (действия выполняются однократно в заданном порядке);
- **циклические** (повторение действий до тех пор, пока не станет верным условие выхода из цикла);
- **разветвляющиеся** (выполняется та или иная последовательность действий, в зависимости от некоторого логического условия);
- **вспомогательные** (некоторое условное обозначение группы команд, выполняющих некоторое целостное действие).

По форме представления различают алгоритмы, представленные на естественном языке (в письменной или устной форме), при помощи формального языка или при помощи блок-схемы.

Блок-схема представляет собой систему из геометрических фигур (блоков), связанных стрелками, которые указывают порядок выполнения блоков, входящих в алгоритм.

Основные обозначения, применяемые в блок-схемах

Название	Обозначение	Характер совершаемых действий
Пуск/ остановка		Начало или конец алгоритма
Процесс		Действия, вычисления, операции
Принятие решения (ветвление)		Происходит проверка условия, вписанного в ромб, и выбор дальнейшего пути в зависимости от результата этой проверки
Ввод/вывод		Команда либо выводит информацию на экран или принтер, либо получает её извне (с клавиатуры, из внешнего файла и т.п.)

Объект (человек или машина), выполняющий алгоритм называется **исполнителем**. Его задача точно (в большинстве случаев формально) выполнить предписанный ему алгоритм. При этом не в его функциях следить за правильностью и корректностью алгоритма, он должен просто исполнять свою систему команд, хотя при возникновении команды, не являющейся для данного исполнителя допустимой, происходит отказ работы алгоритма.

Автоматическим исполнителем в современном мире является компьютер, а алгоритм, записанный в форме, доступной для «понимания» ЭВМ, называют **программой**.

Учебным исполнителем называется некоторый образ на мониторе компьютера, выполняющий некоторые простейшие действия с целью обучения школьника составлению алгоритмов. При этом исполнитель обладает допустимой для него системой команд и обитает в некоторой среде (на плоскости, на прямой, на клетчатом ограниченном поле, в лабиринте и т.д.). Рассмотрим пример. Исполнитель ЧЕРЕПАШКА может перемещаться на плоскости, оставляя непрерывный след. Допустимой системой команд являются движение ВПЕРЕД на x единиц и ПОВОРОТ (изменение направления движения) на y градусов по часовой стрелке. Тогда в результате выполнения алгоритма «ВПЕРЕД 2 ПОВОРОТ 90 ВПЕРЕД 2

ПОВОРОТ 90 ВПЕРЕД 2 ПОВОРОТ 90 ВПЕРЕД 2 ПОВОРОТ 30 ВПЕРЕД 2 ПОВОРОТ 120 ВПЕРЕД 2» на плоскости будет нарисована «домик», все линейные измерения которого равны 2.

2. Создание базы данных. Определение структуры базы данных: количество и типы полей, заполнение таблиц (или использование готовых). Организация поиска информации в базах данных. Создание запросов разной сложности.

БИЛЕТ 7

1. Основные алгоритмические структуры: следование, ветвление, цикл; изображение на блок-схемах. Разбиение задачи на подзадачи. Вспомогательные алгоритмы.

Основные виды алгоритмов (или как их ещё называют алгоритмических структур) следующие:

- линейный алгоритм (следование);
- циклический алгоритм;
- разветвляющийся алгоритм;
- вспомогательный алгоритм.

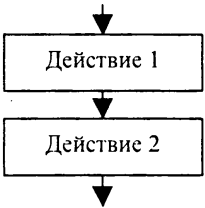
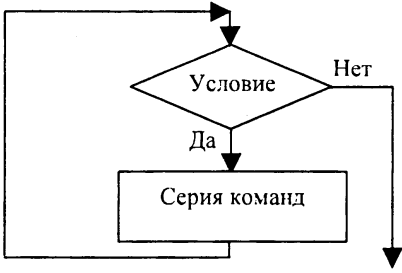
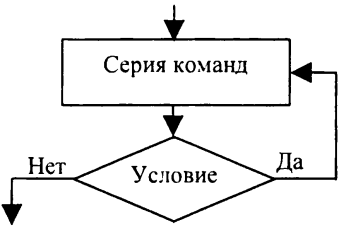
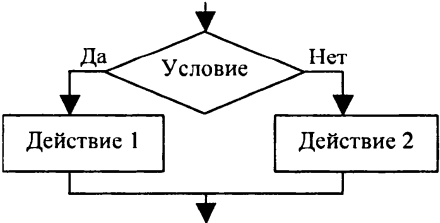
Рассмотрим последовательно каждый из них.

Линейный алгоритм — все этапы выполняются строго последовательно друг за другом, однократно и без пропусков. Изображением на блок-схеме линейного алгоритма является прямоугольник или ряд следующих друг за другом прямоугольников, соединённых стрелками. Внутри прямоугольников принято вписывать наименования выполняемых действий.

Циклический алгоритм применяется для обеспечения многократного выполнения некоторой последовательности действий (*тела цикла*). Индикатором прекращения или продолжения повторений является условие выхода из цикла, т.е. логическое выражение, принимающее значение «истина» или «ложь». Если оно истинно, то повторения прекращаются, в противном случае — продолжаются. Очевидно, что логическое выражение, отвечающее за прекращение работы цикла, должно составляться очень аккуратно, так как иначе велик риск попадания в бесконечный цикл с невозможностью выхода из него (зацикливания), что приведёт к некорректной работе программы. Условиями выхода из цикла являются превышение совершённых повторений некоторого фиксированного числа (*циклы со счётчиками*) или выполнение какого-либо условия (*циклы с условием*). Последние подразделяются на циклы с предусловием и циклы с постусловием. **Цикл с предусловием** сначала проверяет ис-

тинность логического выражения, после чего может начаться выполнение тела цикла. *Цикл с постусловием* сначала один раз выполняет тело цикла, затем проверяет логическое выражение и по результатам проверки может снова повториться тело цикла.

Примеры изображения алгоритмических структур на блок-схемах

Алгоритмическая структура	Блок-схема
Линейный алгоритм	
Циклический алгоритм с предусловием	
Циклический алгоритм с постусловием	
Разветвляющийся алгоритм	

Разветвляющийся алгоритм предоставляет возможность создать некоторый переключатель, который в зависимости от выполнения или невыполнения условия направляет работу алгоритма по одному из возможных путей, т.е. ведёт к выполнению определённой группы действий, продиктованных ситуацией. Отметим, что в некоторых случаях такая последовательность действий может быть пустой.

Вспомогательный алгоритм представляет собой поименованную некоторую группу команд, выполняющих некоторое целостное действие, состоящее из нескольких элементарных. В более крупные алгоритмы вспомогательный алгоритм входит под своим именем и его детализация (расшифровка) не указывается. Хотя предполагается, что ожидаемый результат его действия известен. Вспомогательный алгоритм, записанный на языке программирования, называют **подпрограммой**. При написании больших по объёму программ выделение отдельных кусков в подпрограммы делает структуру алгоритма более прозрачной и доступной для понимания. Кроме того, во многих случаях в программе вызов подпрограммы осуществляется не единожды, что также говорит о необходимости обозначения групп вызываемых команд одним именем. Во многих случаях некоторый алгоритм, оформленный в виде вспомогательного, может быть применён в различных алгоритмах, имеющих совсем разное предназначение.

Таким образом, в процессе алгоритмизации задачи, происходит разбиение её на более простые алгоритмы, которые либо уже существуют, либо могут быть легко созданы. В этом случае процесс алгоритмизации идёт от более общего к частному, от идейной структуры к детализации кусков алгоритма, составляющих основной. Такой метод называется **методом последовательной (пошаговой) детализации**.

Принципиально другим способом является **сборочный метод**. В этом случае большое количество уже готовых алгоритмов (программ) оформляется в виде библиотеки, компоненты которой используются для создания новых алгоритмов, содержание которых определяет пользователь.

2. Работа с электронной таблицей. Создание таблицы в соответствии с условием задачи, использование функций. Построение диаграмм и графиков по табличным данным.

БИЛЕТ 8

1. Величины: константы, переменные, типы величин. Присваивание, ввод и вывод величин. Линейные алгоритмы работы с величинами.

Для рассмотрения понятия «величина» рассмотрим некоторый алгоритмический язык программирования, поддерживаемый на компьютере. ЭВМ работает по определённой программе с набором данных, состоящих из множества величин, под которые отведены определённые объёмы памяти (ячейка или несколько ячеек). В любом языке программирования каждая величина имеет своё имя, тип и значение в данный момент. Имена переменных (или, как их ещё называют, *идентификаторы*) состоят из последовательности букв, цифр, специальных знаков (таких, например, как подчеркивание). Допустимые имена для величин могут слегка варьироваться в зависимости от конкретного языка программирования.

В нижеприведённой таблице на примере языка программирования Pascal перечислены наиболее распространенные типы данных, а также их наиболее характерные черты. Каждый тип данных характеризуется своим внутренним представлением на ЭВМ, множеством значений и операциями, которые допустимо для них производить. Множество значений величин диктуется объёмом памяти, отведённым для её хранения. Большинство языков программирования предоставляет программисту возможность конструировать новые типы данных на основе уже имеющихся.

Типы данных

Название	Описание
Целое число Integer	Целочисленные данные; во внутреннем представлении занимают 2 байта; диапазон возможных значений от -32768 до +32767. Представление точное. Возможные операции: +, -, *, /, целочисленное деление, все операции сравнения. К целым числам возможно применение математических функций (например, модуль). Отметим, что некоторые операции (например, деление) над самими целыми числами дают вещественный результат

Вещественное число Real	Вещественные данные; занимают 6 байт; диапазон возможных значений модуля числа от $2.9E-39$ до $1.7E+38$, точность представления данных составляет 11 значащих цифр. Возможные операции: +, -, *, /, все операции сравнения. К вещественным числам также возможно применение математических функций (корень, синус, модуль)
Символ Char	Символ; занимает 1 байт. Возможные операции: получение по символу его кода в таблице кодировки, получение прописной буквы, если задана строчная
Строка символов String	Строка символов. Её максимальная длина l должна быть задана заранее, объём отводимой памяти составляет $(l+1)$ байт (лишний байт необходим для хранения длины строки)
Логический тип Boolean	Принимает значение «истина» (true) или «ложь» (false), занимает 1 байт. Возможные операции: логическое «не», логическое «или», логическое «и», исключающее «или».

В языках программирования идентификаторы бывают *константами* и *переменными*. В отличие от константы переменная именуется величиной, которая может изменять своё значение в ходе работы программы. Программист должен заранее определить, с какой из этого типа величиной он будет работать.

Для работы с величинами необходимо освоить следующие команды: присваивание, ввод и вывод. **Присваивание** — это задание значения переменной. Осуществляет его оператор присваивания (в Pascal :=), который нельзя путать с операцией сравнения (=), осуществляющей проверку идентичности объектов. **Оператор присваивания**, а также **оператор ввода данных** — это два возможных способа задания значения переменных. При применении оператора ввода выполнение программы приостанавливается, и компьютер ждёт ввода данных с клавиатуры или с какого-либо другого устройства. При этом вводимые данные должны быть представлены в формате, соответствующему типу вводимой величины. До момента первого присваивания переменная значения не имеет и является неопределённой.

Для представления полученного результата используются команды **вывода**. По команде вывода значение переменной преобразуется в вид, понятный человеку, и выводится на монитор или принтер.

Все программы при своей работе производят действия над переменными, эти действия группируются в линейные алгоритмы, являющиеся составными частями более сложных алгоритмов с ветвлениями и циклами. Помимо этого, для вычисления значения выражений компьютер сам выполняет некоторый линейный алгоритм, обусловленный приоритетностью операций, входящих в это выражение. Так, например, вычисление значения выражения f , заданного формулой $a*b+a/b+abs(a)$, может происходить по следующему сценарию: найти произведение $a*b$ (значение присваивается некоторой промежуточной переменной $var1$) $\rightarrow$ найти частное a/b (значение присваивается некоторой промежуточной переменной $var2$) $\rightarrow$ найти модуль числа a , т.е. $abs(a)$, (значение присваивается некоторой промежуточной переменной $var3$) $\rightarrow$ найти сумму $var1$ и $var2$ (значение присваивается некоторой промежуточной переменной $var4$) $\rightarrow$ найти сумму $var4$ и $var3$ (что и дает ответ, т.е. значение f). Отметим, что в целях экономии памяти, система может производить вычисления с меньшим количеством вспомогательных переменных (например, можно было вместо $var4$, использовать $var1$, записать что $var1:=var1+var2$, в этом случае произошло бы сложение выражений в правой части, а далее результат был бы записан в область памяти, соответствующую переменной $var1$).

2. Поиск информации в Интернете с применением языка запросов.

БИЛЕТ 9

1. Логические величины, операции, выражения. Логические выражения в качестве условий в ветвящихся и циклических алгоритмах.

Элементами логических рассуждений являются утверждения, которые либо ложны (Л, 0), либо истинны (И, 1), но не то и другое вместе. Такие утверждения называют *высказываниями*. Из простых высказываний при помощи логических операций могут быть построены более сложные составные высказывания (или формулы).

Существует три основных логических операции:

- *инверсия* (логическое отрицание, не, $\neg$);
- *конъюнкция* (логическое умножение, и, $\&$, $\wedge$);
- *дизъюнкция* (логическое сложение, или, $\vee$).

Остальные введенные логические операции алгебры логики (*эквивалентность*, *импликация*, *исключающее или*) могут быть выражены через три основных.

Приведем таблицы истинности для простейших логических функций.

Отрицание (НЕ)	
A	$\neg A$
0	1
1	0

		Конъюнкция (И)				Дизъюнкция (ИЛИ)	
A	B	$A \wedge B$		A	B	$A \vee B$	
0	0	0		0	0	0	
1	1	1		1	1	1	
0	1	0		0	1	1	
1	0	0		1	0	1	

Эквивалентность			Импликация			Исключающее или		
A	B	$A \Leftrightarrow B$	A	B	$A \rightarrow B$	A	B	$A \text{ XOR } B$
0	0	1	0	0	1	0	0	0
1	1	1	1	1	1	1	1	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	0	1	0	1

Порядок выполнения операций в логическом выражении задается скобками. Если скобки отсутствуют, то приоритетность операций следующая: отрицание, конъюнкция, дизъюнкция, импликация, эквивалентность.

Основные законы алгебры логики

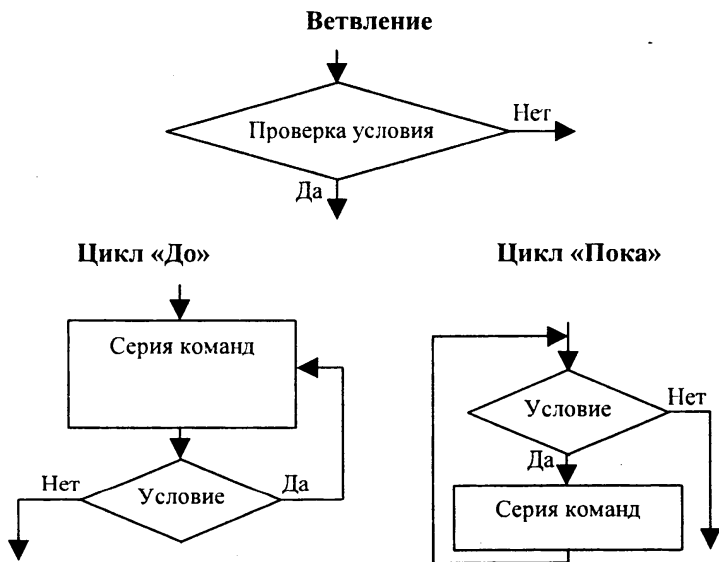
Название закона	Формулировка
Переместительный закон	$A \vee B = B \vee A, A \wedge B = B \wedge A$
Сочетательный закон	$(A \vee B) \vee C = A \vee (B \vee C),$ $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
Распределительный закон	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
Закон непротиворечия	$A \wedge \neg A = 0$
Закон исключённого третьего	$A \vee \neg A = 1$
Закон двойного отрицания	$\neg(\neg A) = A$
Законы де Моргана	$\neg(A \vee B) = \neg A \wedge \neg B$ $\neg(A \wedge B) = \neg A \vee \neg B$

Другие важные формулы:

$A \vee 0 = A$, $A \wedge 0 = 0$, $A \vee 1 = 1$, $A \wedge 1 = A$, $0 \rightarrow A = 1$, $A \rightarrow B = \neg A \vee B$.

Структура **ветвление** обеспечивает переход по одной из ветвей алгоритма в зависимости от выполнения или невыполнения некоторого логического условия. В блок-схеме ветвление условно обозначается ромбом с записанным внутри условием, принимающим значение «истина» или «ложь»; после вычисления значения этого логического выражения осуществляется выбор дальнейшего пути: по стрелке «Да», если значение выражения истинно, либо по стрелке «НЕТ», если значение ложно.

Цикл обеспечивает многократное выполнение некоторой последовательности действий (или тела цикла) до тех пор, пока не станет ложным некоторое логическое выражение. Различают два вида циклов: «До» и «Пока». В первом случае проверка условия осуществляется после выполнения серии команд. Если логическое выражение все ещё истинно, то серия команд повторяется опять, а далее снова идёт проверка логического условия выхода из цикла. В цикле «Пока» сначала проверяется условие, и только после этого, если оно верно, осуществляется серия команд, затем снова проверяется истинность выражения.



2. Форматирование текстового документа. Установка параметров страницы, вставка номеров страниц, колонтитулов, гиперссылок, изменение параметров шрифта и абзаца.

БИЛЕТ 10

1. Представление о программировании: язык программирования (на примере одного из языков высокого уровня); примеры несложных программ с линейной, ветвящейся и циклической структурой.

Язык программирования — это фиксированная система обозначений для описания алгоритмов и структур данных. Во всяком языке программирования имеется алфавит, т.е. набор символов, которые можно использовать в программе, существуют зарезервированные слова, имеющие вполне определённый смысл и определённое назначение и которые нельзя менять, предоставляется возможность вводить переменные, использовать константы (как встроенные, так и введённые самим программистом), использовать существующие и доопределять новые операторы (они задают те или иные действия, которые должна выполнять программа). **Система программирования** — некоторая единая структура, состоящая из компилятора (или транслятора) с языка программирования и некоторой инструментальной программной оболочки, способствующей повышению эффективности создания программ. **Компилятор** или **транслятор** — это программа, переводящая текст программы на языке программирования высокого уровня в эквивалентную программу на машинном языке.

Рассмотрим принципы построения программы на языке Pascal на нескольких примерах. В фигурных скобках даны комментарии к программам.

Программа с линейной структурой

Program volume; {Программа вычисления объёма шара по заданному радиусу, начало раздела описаний переменных и констант}

Const pi=3.14; {Вводится вещественная константа, имеющая смысл числа π }

Var R, V:real; {описываются две переменные, имеющие смысл радиуса и объёма шара}

Begin {начало раздела операторов}

Writeln('Введите радиус шара'); {Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры значения радиуса шара}

Readln(R); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

V:=4*pi*R*R*R/3; {вычисляется значение правой части выражения и срабатывает оператор присваивания переменной V посчитанного значения}
 Writeln('Объём шара равен ',V); {результат выводится на экран}

End.

Программа с ветвящейся структурой

Program abs_abs; {Программа вычисления модуля числа}
Var a:real; {описывается переменная, отождествляемая с числом, модуль которого мы вычисляем}
 b:real; {описывается переменная, представляющая собой модуль числа}

Begin

Writeln('Введите число');
 Readln(a);
 If (a>=0) then b:=a; else b:=-a; {Условный оператор if проверяет условие $a \geq 0$ и в зависимости от результатов проверки производит вычисление модуля числа согласно его определению}
 Writeln('Модуль числа равен ',b); {результат выводится на экран}
 b:=abs(a); {модуль числа вычисляется при помощи встроенной функции языка Pascal, возвращающей модуль вещественного числа}
 Writeln('Сравним результат. Модуль числа равен ',b); {на монитор выводится тот же результат, полученный другим способом}

End.

Программа с циклической структурой

Program sum_sum; {Программа суммы всех двузначных натуральных чисел}
Var a:integer; {описывается переменная, отождествляемая суммой всех двузначных натуральных чисел}
 i:integer; {описывается переменная, имеющая смысл счетчика цикла}

Begin

a:=0; {переменной присваивается начальное значение}
 for i:=10 to 99 do a:=a+i; {Срабатывает оператор повторений for. Происходит проверка условия принадлежности счетчика цикла определённому диапазону значений от начального значения 10 до конечного 99,

далее выполняется тело цикла, наращивается значение счетчика на единицу, затем снова проверка условия выхода из цикла. В теле цикла происходит суммирование «старой» суммы с следующим числом и результат присваивается снова переменной, обозначающей подсчитываемую сумму}

Writeln('Сумма всех двузначных чисел равна ',b);

End.

Помимо оператора повторений **for**, в языке программирования Pascal существуют конструкции **repeat ... until** <условие> и **while** <условие> **do...**, соответствующие циклам с постусловиями и предусловиями соответственно. Аналогичные конструкции есть и в языке Basic.

Те же программы на языке программирования Basic имеют следующий вид.

Программа с линейной структурой

REM Объем шара

pi=3.14

INPUT "Введите радиус шара"; R

V=4*pi*R*R*R/3;

PRINT "Объем шара равен "; V

END

Программа с ветвящейся структурой

REM Модуль числа

INPUT "Введите число"; a

IF a>=0 THEN

b=a

ELSE

b=-a

END IF

PRINT "Модуль числа равен "; b

b=ABS(a)

PRINT "Сравним результат. Модуль числа равен "; b

END

Программа с циклической структурой

REM сумма всех двузначных чисел

a=0

FOR i=10 TO 99

a=a+i

NEXT i

PRINT "Сумма всех двузначных чисел равна ", b

END

Современные языки программирования предоставляют возможность работать с процедурами и функциями, т.е. некоторыми подпрограммами, оформленными заранее и, возможно, записанными во внешний по отношению к текущей программе, файл (так называемая *модульность языка программирования*). Выполнение процедур и функций происходит по команде пользователя.

2. Работа с архиваторами и антивирусными программами. Создание многотомного архива, использование антивирусных программ.

БИЛЕТ 11

1. Основные компоненты компьютера, их функциональное назначение и принципы работы. Программный принцип работы компьютера.

Компьютер — это техническое устройство преобразования, хранения и обработки информации, осуществляющее заданную программную последовательность операций. Основными компонентами компьютера являются устройства ввода, память, процессор и устройства вывода.

Устройства ввода-вывода обеспечивают ввод информации в память компьютера и обратный вывод её в форме, доступной для понимания человека. Основными устройствами ввода являются *клавиатура, манипуляторы* (например, мыши), *сканеры*. Основные устройства вывода: *монитор, принтер*.

Процессор — это центральное устройство, отвечающее за обработку всех данных и координирующее работу всех узлов компьютера. Процессор состоит из арифметико-логического устройства, управляющего устройства, ячеек внутренней памяти. Ссылка на ячейку памяти, происходит по её адресу. Устройство управления обеспечивает передачу адресов ячеек памяти, команд и данных внутри процессора. Арифметико-логическое устройство производит логические и арифметические действия с данными. Основная характеристика процессора — тактовая частота, чем она выше, тем больше производительность компьютера.

Память компьютера бывает внутренняя и внешняя и предназначена для хранения программ и обработки информации. Её основной характеристикой является объём. Внешняя память предназначена для долговременного хранения данных. Её примерами могут служить компакт-диски, карты флэш-памяти, винчестеры. Конструкция внешних носителей памяти предусматривает наличие движущихся

механических частей, что влечёт снижение их быстродействия по сравнению с внутренней памятью, которая полностью электронная. Внутренняя память состоит из ОЗУ (оперативного запоминающего устройства) и ПЗУ (постоянного запоминающего устройства). Первое из них хранит данные, используемые в данный момент времени, второе хранит данные, предназначенные для загрузки компьютера (используются только для чтения). При выключении компьютера содержимое ОЗУ пропадает, а ПЗУ сохраняется. Отметим, что при хранении программ и данных используется принцип однородности памяти, введенный фон Нейманом. Это означает, что программы и данные хранятся в одной и той же памяти.

Принципиальная схема устройства компьютера



Программа — последовательность действий (команд), которую должен выполнить компьютер для решения задачи. **Программный принцип работы компьютера**, состоит в том, что компьютер выполняет действия по заранее заданной программе.

Базовой конфигурацией ПК называется минимальный комплект аппаратных средств, достаточный для начала работы с компьютером. Принято считать, что базовыми являются системный блок, монитор (как устройство вывода информации), клавиатура и мышь (как устройства ввода информации). Системный блок имеет, как правило, большое количество разъёмов, обеспечивающих доступ ко внутренним устройствам. Через эти разъёмы происходит подключение дополнительного периферийного оборудования. В данный момент наиболее популярным и удобным является разъём USB, посредством которого могут быть подключены, например, и принтер, и сканер, и мышь. Внутри системного блока расположена **Материнская плата** — самая большая плата компьютера. Она содержит магистрали, связывающие процессор с оперативной памятью. Из

схемы устройства компьютера видно, что обмен данными между периферийными устройствами и процессором идёт через магистраль.

2. Построение алгоритма для обработки величин с реализацией на языке программирования (ветвление, цикл, линейный массив или вспомогательные алгоритмы). Отладка программы и получение результатов.

БИЛЕТ 12

1. Программное обеспечение компьютера, состав и структура. Назначение операционной системы. Командное взаимодействие пользователя с компьютером. Графический пользовательский интерфейс.

Последовательность команд, которую выполняет компьютер в процессе исполнения алгоритма, называется *программой*. Все программы, хранящиеся на компьютере, составляют его программное обеспечение. Программы бывают системные и прикладные.

Системное программное обеспечение обеспечивает работу всех устройств и программ компьютера и взаимодействие человека с ним. Такие программы обеспечивают управление памятью, вводом и выводом информации, тестируют работу периферийных устройств. При запуске компьютера начинает работать важная системная программа, называемая операционной системой. Другие системные программы — это драйвера, архиваторы, антивирусные программы.

Операционная система — это совокупность программных средств, обеспечивающих управление аппаратными ресурсами ЭВМ и взаимодействие программных процессов с аппаратурой, другими процессами и пользователем. Операционная система выполняет следующие действия: управление памятью, управление вводом-выводом, управление файловой системой, управление взаимодействием процессов, защиту, учёт использования ресурсов, обработку командного языка. Часто многие отдельные сервисные программы (трансляторы, утилиты) считаются частью операционной системы.

Другой тип программ — *прикладные программы*. Такие программы предоставляют возможность проводить необходимые пользователю прикладные работы: использовать возможности компьютера по выполнению расчетов, созданию текстов и изображений, таблиц. К таким программам относятся текстовые и графические

редакторы, электронные таблицы, системы управления базами данных, браузеры, игры. Среди прикладных программ различают отдельно системы программирования, которые сами дают возможность создавать новые программы как прикладные, так и системные.

Для эффективной работы пользователя на ЭВМ ему необходимо инструктировать компьютер о действиях, которые необходимо произвести, т.е. подавать команды. Один из подходов к вводу команд — это *командная строка*, в которую вводится некоторая последовательность знаков, означающих конкретную команду.

Для удобства работы на компьютере составной частью современных операционных систем являются *графические интерфейсы пользователя*. В этом случае сложные команды можно вводить при помощи щелчков мыши на различных областях экрана, а не набирая команду вручную с клавиатуры в командной строке. Основой такого интерфейса являются диалоговые окна, меню и элементы управления (важнейшими из последних являются кнопки). В операционной системе Windows существуют следующие элементы графического интерфейса:

- *рабочий стол* (вид экрана компьютера, при котором на представленном поле расположены значки и иконки, обозначающие программы). На рабочем столе расположены либо сами программы, либо их ярлыки (т.е. некоторые указатели на место расположения самих программ).
- *значки* (некоторое специальное изображение, подобранное для определённого класса программ или файлов с данными).
- в нижней части экрана располагается *панель задач*, содержащая кнопку «Пуск», панель быстрого запуска (для часто используемых программ) и значки открытых окон программ, запущенных на текущий момент.
- *окно Windows* — это часть дисплея, с которой программа работает как с отдельным экраном. Важнейшее свойство окон — возможность изменения размера без нарушения работы самой выполняемой в нём программы.

2. Создание мультимедийной презентации на основе шаблонов. Выбор типа разметки слайда, применение шаблона оформления, цветовых схем и эффектов анимации. Демонстрация слайдов с использованием управляющих кнопок.

БИЛЕТ 13

1. Понятие файла и файловой системы организации данных (папка, иерархическая структура, имя файла, тип файла, параметры файла). Основные операции с файлами и папками, выполняемые пользователем. Понятие об архивировании и защите от вирусов.

Файл — это некоторая целостная совокупность данных или программа, имеющая имя и хранящаяся на внешнем носителе. Имя файла состоит из двух частей, разделённых точкой: имя файла и его расширение, позволяющее судить о его типе (программа или данные) и назначении. Пользователь сам выбирает имя файла, а его расширение, как правило, присваивается автоматически программой, в которой файл создается. Для упрощения работы с файлами приняты следующие соответствия некоторых файлов и их расширений:

Расширение	Тип хранящейся информации
doc,txt	Текстовая
bmp, gif, tif, jpg	Графическая
avi	Видеоизображение
wav	Звуковая
bak	Резервная копия (устаревшая версия) файла, сохраненного после внесения исправлений
exe, com	Исполняемый файл, запускающий некоторую программу
xls	Электронная таблица
rar, zip, arj	Архив
dbf	База данных
bas, pas, cp	Текст программы на языке программирования (Бейсик, Паскаль, С)
Sys	Системный файл
dll	Системная библиотека
hlp	Файл помощи
bat	Текстовый командный файл

Помимо этого файл характеризуется размером, датой и временем создания. Размер файла — это количество места, занимаемое им на диске (единицы измерения байт, килобайт, мегабайт).

Папка — объект, объединяющий в группу файлы и другие папки. Характеризуется своим именем (без расширения), размером (суммарным объёмом всех файлов и папок, находящихся в ней), датой и временем создания.

Каталог — структура данных, обеспечивающая поиск объекта по текстовому имени.

Файловой системой называют совокупность файлов и каталогов, а также управляющей информации для выполнения операций над ними. В современных операционных системах используется многоуровневая иерархическая файловая система, организованная в виде дерева:



Основными операциями с файлами и папками, которые может произвести пользователь из системной среды, являются

- копирование,
- перемещение,
- удаление,
- переименование.

Также файл может быть создан или закрыт (открыт) из программной среды.

Хранение и передача по компьютерным сетям файлов во многих случаях требует компактности представления данных, с целью уменьшения объема. Для этого используют программы архиваторы, такие как RAR или ARJ. Они обеспечивают сжатие и упаковку архивируемых данных, а также их восстановление в обратную сторону в случае необходимости. Принцип работы архиваторов — нахождение повторяющихся или однотипных кусков текста или кода и замещение их какой-то более компактной меткой.

Компьютерный вирус — это некоторая специальная программа, выполняющая на компьютере пользователя всевозможные вредные действия и, как правило запускающаяся без его ведома сама по себе. При этом в зависимости от типа вируса может происходить порча или несанкционированная передача посторонним лицам информации, хранящейся в памяти компьютера, исчезновение информации или программ, существенное замедление работы компьютера, некорректная работа некоторых приложений, появление посторонних

видео-аудио эффектов, остановка работы компьютера (т.е. его «зависание» или невозможность загрузки).

Для защиты от вирусов применяются специальные **антивирусные программы**, работа которых, как правило, заключается в постоянном мониторинге компьютерной системы на наличие вирусов. Результатом их работы является удаление, «лечение» или помещение «в карантин» зараженных файлов или файлов, которые потенциально могут являться вирусами. Большинство современных антивирусных программ (NOD32, Doctor Web, Symantec Antivirus, Kaspersky) предоставляют пользователю возможность осуществлять проверку на вирусы как постоянно (в режиме реального времени проверять входящие e-mail письма и передаваемые по сети данные, проверять подключаемые внешние носители памяти), так и по запросу самого пользователя, а также выбирать области памяти на диске, которые необходимо проверить. Отметим, что из-за постоянно вновь создающихся вирусов, а также модификации старых, необходимо периодическое обновление антивирусных программ, которое, как правило, осуществляется через Интернет путем скачивания с сайта разработчика обновлённых антивирусных баз.

2. Организация поиска информации в готовой базе данных с применением составного логического выражения.

БИЛЕТ 14

1. Информационные ресурсы общества. Основы информационной безопасности, этики и права.

Ресурс — это средства, имеющиеся в наличии, к которым можно прибегнуть в случае необходимости. В современном мире все большее и большее значение приобретают **информационные ресурсы**, под которыми понимают совокупность данных, организованных для эффективного получения достоверной информации. Юридически под информационными ресурсами понимается информация, зафиксированная на материальном носителе и хранящаяся в информационных системах (библиотеках, архивах, фондах, банках данных и др.). Свойствами информационных ресурсов являются постоянная пополняемость и возможность многократного использования.

Информационный ресурс может иметь своего обладателя (человек, государство или организация). Информация, находящаяся в свободном доступе, называется открытой. Информация, доступ к которой ограничен, относится либо к конфиденциальной информации, либо к государственной тайне.

Важнейшей характеристикой информации является её *достоверность*. Получение именно достоверной информации отвечает интересам государственной безопасности. Поэтому информационные ресурсы общества в настоящее время рассматриваются как стратегические ресурсы, аналогичные по значимости ресурсам материальным и сырьевым.

Основными хранилищами информации являются библиотеки (как классические, хранящие бумажные носители, так и современные электронные) и архивы. Они могут служить как справочными, так и образовательными ресурсами.

Хранение и распространение информации в современном мире регулируется государственными законами и нормативными актами. Они, с одной стороны, гарантируют свободу распространения открытой информации, а с другой — препятствуют распространению информации конфиденциального характера и информации, относящейся к государственной тайне. Однако, в некоторых случаях юридические правила не гарантируют соблюдения общепринятых моральных и этических норм, что особенно заметно в публикациях в Интернете, где применение юридических норм пока затруднительно.

Правовое регулирование в сфере распространения программного обеспечения в настоящее время обеспечивается покупкой лицензий на использование определённых программ. Использование программы, распространяемой не свободно, без покупки лицензии является преступлением и наказуемо. Однако в настоящее время существует большое количество программ, находящихся в свободном доступе и не требующих платы за пользование ими (так называемые *freeware* программы). Помимо этого можно встретиться с демoversиями лицензионных программных продуктов, также предлагающихся бесплатно. Однако, как правило, такие программы либо имеют ограниченный набор функций по сравнению с оригиналом, либо ограниченный срок работы (например, месяц).

На программные продукты, также как и на литературные произведения, распространяется авторское право. Для информирования о своих авторских правах на программном продукте ставится знак ©, пишется имя правообладателя и год выпуска. Автору программы принадлежат исключительные права на воспроизведение, распространение программы и на её модификацию.

Для предотвращения несанкционированного доступа к информации компьютерный пользователь вправе поставить на свой компьютер пароль, известный только ему. Такой же принцип использования пароля используется при доступе к электронной почте.

2. Работа с электронной таблицей. Проведение вычислительного эксперимента в среде электронной таблицы. Решение задачи с использованием электронной таблицы для изменяющихся начальных данных.

БИЛЕТ 15

1. Технологии работы с текстовыми документами. Текстовые редакторы и процессоры: назначение и возможности. Основные структурные элементы текстового документа. Шрифты, стили, форматы. Основные приёмы редактирования документа. Встраиваемые объекты. Понятие гипертекста.

Для создания новых документов и внесения изменений в ранее созданные на компьютере используются **текстовые редакторы**, позволяющие создавать, редактировать, форматировать, сохранять и распечатывать документы. Простые текстовые редакторы (например, стандартное приложение Windows Блокнот) позволяют редактировать текст, а также осуществлять простейшее форматирование. Более совершенные текстовые редакторы (например, Microsoft Word), называемые иногда текстовыми процессорами, имеют гораздо более широкий спектр возможностей по созданию документов (вставка списков и таблиц, средства проверки орфографии, сохранение исправлений и много другое).

В профессиональной издательской деятельности человека для подготовки к изданию книг, журналов и газет в процессе макетирования печатного издания используются мощные программы обработки текста — настольные издательские системы (например, Adobe PageMaker, Microsoft Office Publisher), а для подготовки к публикации в Интернете Web-страниц и Web-сайтов - специализированные приложения такие, как Microsoft FrontPage.

Итак, **текстовые редакторы** — это программы, обеспечивающие редактирование и форматирование текстов, программ и документов в соответствии с задаваемыми пользователем командами. **Редактирование** — это процесс внесения изменений в документ, обеспечивающий добавление, удаление, перемещение или исправление содержания документа. **Форматирование** — это процесс представления внешнего вида документа или отдельных его объектов в соответствии с предписанным форматом. При этом под форматом понимают способ расположения и представления данных. В процессе форматирования формируются абзацы, центрируются заголовки, выравниваются поля, происходит разбиение на страницы.

Каждый документ может быть сохранен в файле, отображен на экране или напечатан на бумаге (физической странице). Область физической страницы, в которой размещаются объекты документа, называют логической страницей. Форматирование страницы документа в любой прикладной среде предполагает установку ориентации страницы, полей, колонтитулов. **Ориентация страницы** — это положение листа бумаги в пространстве. Различают книжную и альбомную ориентации. **Поля** — это области физической страницы, используемые для размещения сопроводительной информации (сносок, колонтитулов и т.п.). Левое и правое поля обычно остаются незаполненными. В верхнем и нижнем полях могут располагаться колонтитулы. **Колонтитулы** представляют собой служебную информацию, размещаемую в верхнем или нижнем поле страницы (фамилия автора, название документа, номер страницы и т.п.). Все, что находится между колонтитулами, принято называть **телом страницы**: текст, подстрочные примечания, рисунки, таблицы.

Вид каждого объекта, составляющего документ, определяется его размером, формой объекта, местоположением в документе относительно других объектов. Размер измеряется по высоте и ширине, при этом существует возможность масштабирования (т.е. выбора масштаба). Местоположение объекта в документе определяется относительно края страницы. Положение объекта относительно других объектов документа задается пользователем.

Кроме перечисленных параметров, каждый объект может иметь другие параметры, доступные для форматирования: шрифтовое оформление; цветовое оформление и узор; толщина, тип. **Шрифт** — внешний вид символов алфавита. Каждую конкретную реализацию шрифта называют **начертанием**. Совокупность всех начертаний называют **гарнитурой** шрифта, которая определяет художественное решение, выделяющее шрифт среди других. Почти все семейства шрифтов имеют обычное (прямое) начертание. Курсивный, полужирный и полужирный курсивный шрифты используют для смыслового выделения текста. **Кегль** (т.е. размер шрифта) измеряют в печатных пунктах. Один пункт равен 0,35 мм. Помимо этого шрифт может характеризоваться контрастностью (отношением толщины горизонтальных и горизонтальных штрихов символа), наличием засечек (небольших черточек на концах линий), цветом и некоторыми другими параметрами.

Кроме текстовых символов форматированный текст содержит специальные невидимые коды, которые сообщают программе, как надо его отображать на экране и печатать на принтере: какой шрифт использовать, каким должно быть начертание и размер символов, как оформляются абзацы и заголовки.

При оформлении текстовых документов часто требуется добавлять в документ нетекстовые элементы или объекты. Современные текстовые редакторы позволяют это делать — они имеют широкие возможности по вставке в текст рисунков, диаграмм, формул и т.д.

Во многих электронных документах используется так называемое относительное форматирование. Оно обеспечивает возможность просмотра документа на любом компьютере с заранее не заданной величиной экрана и в любом окне.

Формат файла — это способ хранения текста в файле. Наиболее часто используемые форматы приведены в таблице.

Формат	Краткое описание
TXT	Файл, состоящий из числовых кодов символов текста, не поддерживается возможность хранения таких сложных объектов как таблицы или рисунки
DOC	Файл, состоящий из числовых кодов символов текста и дополнительных кодов команд форматирования, дает возможность вставлять в документ различные объекты, не являющимися текстовыми (рисунки, диаграммы)
RTF	Файл, состоящий из числовых кодов символов текста и дополнительных кодов команд форматирования, сохраняет информацию о форматировании произведенном в документе, но набор его возможностей беднее по сравнению с форматом DOC
HTM, HTML	Файл для хранения Web-страниц и гипертекста

Для того чтобы отформатированный документ выглядел аккуратно, принято сходные объекты оформлять единообразно, т.е. одним шрифтом, с одинаковыми отступами и т.п. Для обеспечения такого единообразия введено понятие стиля. **Стиль** — это совокупность параметров форматирования объекта документа. Простейший стиль, как правило предлагаемый самим текстовым редактором, называется **автоформатом**. Пользователь по своему желанию может изменять стиль оформления всего текстового документа или его частей по своему усмотрению (например, убрать нумерацию глав, оставив только заголовки, переместить номер страницы из верхнего колонтитула в нижний, задать нумерацию формул с размещением по правому краю страницы).

Для создания Web-страниц и для обеспечения удобства навигации по документам большого объема используется гипертекст. **Гипертекст** — это размеченный текст, содержащий в себе ссылки на другие документы или места документа. В более широком понимании термина, гипертекстом является любой словарь или энциклопедия, где встречаются отсылки к другим частям текста.

2. Построение алгоритма и реализация на изучаемом языке программирования или в среде учебного исполнителя. Демонстрация полученного алгоритма в среде учебного исполнителя или отладка программы и получение результатов.

БИЛЕТ 16

1. Технологии работы с графической информацией. Растровая и векторная графика. Аппаратные средства ввода и вывода графических изображений. Прикладные программы работы с графикой. Графический редактор. Основные инструменты и режимы работы.

Графический редактор — это программа, предназначенная для создания, редактирования и просмотра изображений. В настоящее время существует два принципа хранения изображений: растровый и векторный, используемые в различных графических редакторах. Сравнительный анализ этих двух способов проведен в таблице.

	Векторная графика	Растровая графика
Основные принципы	Основными структурами в векторной графике являются геометрические фигуры. При этом в памяти компьютера хранятся не сами фигуры, а формулы, их описывающие	Изображение формируется из точек (пикселей), в памяти компьютера отдельно для каждого пикселя хранится его цвет
Достоинства	Изменение масштаба не приводит к потере качества изображения, также не происходит существенного увеличения объема занимаемой памяти. Экономичные объемы требуемой под изображение памяти. Хорошее качество изображения. Возможность редактирования отдельных элементов изображения	Хорошая передача цвета и полутонов

Недостатки	Плохое качество передачи полутонов цвета. При наличии большого количества объектов, составляющих рисунок, увеличивается время загрузки файла. Не всякий объект природы может быть легко представлен в виде комбинации геометрических фигур	При увеличении размера изображения происходит потеря его качества. Большой объем занимаемого места на носителе
Применение	Для изображений чертежей, схем, диаграмм	Для работы с фотографиями и рисунками
Программы	CorelDraw, Adobe Illustrator	PhotoShop, Paint

Для создания картинки в графическом редакторе необходимо выбрать на панели инструментов «инструмент рисования»: условно изображенные картинки, после выбора которых пользователю предоставляется возможность нарисовать линию, если выбрана «линия», прямоугольник, если выбран «прямоугольник», ломаную, если выбран инструмент «ломаная». Наборы инструментов есть и в растровых графических редакторах, и в векторных. Однако в растровом редакторе нарисованный объект становится лишь изображением, тогда как в векторном у объекта целиком может быть в любой момент изменен размер, местоположение, цвет. Для редактирования изображения в графических редакторах существуют инструменты выделения, после чего пользователь может производить копирование, удаление, перемещение выделенных фрагментов. Существуют также возможности текстовых добавления надписей. Основными характеристиками рисуемых объектов являются цвет (контур и заливки), толщина и тип линий. Для более детального рассмотрения объектов используется инструмент «лупа».

Ввод изображений в компьютер возможен не только посредством прямого рисования объекта пользователем, но и при помощи таких устройств как сканер, цифровая камера или цифровой фотоаппарат. В этом случае полученное с данного устройства изображение в последствие, как и любое другое, может быть открыто в графическом редакторе и отредактировано. Для вывода изображений применяются мониторы, принтеры и плоттеры (для вывода широкоформатных рисунков).

2. Решение задачи по теме «Системы счисления» на изучаемом языке программирования или с использованием стандартной программы «Калькулятор».

БИЛЕТ 17

1. Табличные базы данных (БД): основные понятия (поле, запись, первичный ключ записи); типы данных. Системы управления базами данных и принципы работы с ними. Поиск, удаление и сортировка данных в БД. Условия поиска (логические выражения); порядок и ключи сортировки.

База данных (БД) — это определённым образом организованный набор информации из некоторой предметной области, которая упорядочена в виде набора элементов или записей одинаковой структуры. Базы данных относятся к компьютерной технологии сохранения, поиска и сортировки информации. Определение физической и логической структуры базы данных, ввод информации и доступ к ней осуществляется программной системой, называемой *системой управления базой данных (СУБД)*. Примером СУБД является Microsoft Office Access.

В зависимости от способа представления данных и отношений между ними база данных может иметь реляционную (табличную), сетевую или иерархическую структуры. Наибольшее распространение получили реляционные базы данных, имеющих табличную форму организации. Главное достоинство таблиц — их понятность, простота и удобство реализации на ЭВМ.

Обычно информация в базах данных хранится не в одной таблице, а в нескольких взаимосвязанных. Каждая таблица имеет уникальное имя, по которому на неё можно сослаться. Столбцы таблицы соответствуют тем или иным характеристикам объекта и называются полями. Каждое поле характеризуется именем и типом хранящихся данных (число, символ, дата, логическое поле). Каждая строка таблицы соответствует одному из объектов. Она называется записью и содержит значения всех полей, характеризующих объект. Каждая строка должна быть уникальной, поэтому одному полю или группе полей отводится роль первичного ключа, т.е. набор их значений уникален в таблице. Если ключом является одно поле, то ключ простой, если более — то составной. Строки в базах данных хранятся неупорядоченно, их упорядоченный вывод на экран или принтер, осуществляется только по запросу пользователя. При этом результатом работы такого запроса также является таблица, но отсортированная согласно пожеланию человека. Связь между несколькими таблицами в базе данных осуществляется через поля, имеющие одни и те же значения и смысл. Возможны следующие виды связей: «один-к-одному», «один-ко-многим», «многие-ко-многим».

Важнейшим элементом любой системы управления базами данных является наличие средств ускоренного поиска данных, по-

сколько поиск — самая распространенная операция при работе с таблицами. Этот механизм обычно реализуется введением так называемых **индексов**. Любая таблица базы данных может быть проиндексирована по нескольким полям и иметь любое число индексов. Индексы (а точнее индексные файлы) содержат информацию о расположении записей базы данных в определённом порядке (хронологическом, по возрастанию, алфавитном). Сортировка может быть произведена сразу по нескольким полям (ключам), в этом случае важен порядок указанных полей (их приоритетность).

Выборка информации из базы данных проводится с помощью запросов — логических выражений, реализованных, как правило, на различных модификациях специального языка SQL — структурированного языка запросов, применяемого для создания, модификации и управления данными в реляционных базах данных. Запросы позволяют проводить выборку, добавление, удаление и редактирование записей по заданным критериям. При формировании запроса над полями баз данных могут производиться определённые действия, соответствующие типу конкретного поля. Критерий выбора формируется в виде логического выражения, имеющего значение либо истина, либо ложь. По результатам работы запроса формируется новая таблица из записей, значения полей которой делают истинным значение этого логического выражения. При формировании логического выражения используются стандартные логические операции (не, или, и), знаки отношений (<, >, =) и встроенные функции преобразования данных, хранящихся в полях таблицы.

2. Построение алгоритма для обработки величин с реализацией на языке программирования (ветвление, цикл, линейный массив или вспомогательные алгоритмы). Отладка программы, получение результатов.

БИЛЕТ 18

1. Технология обработки информации в электронных таблицах (ЭТ). Структура электронной таблицы. Типы данных: числа, формулы, текст. Правила записи формул. Основные встроенные функции. Абсолютные и относительные ссылки. Графическое представление данных.

Электронная таблица — это удобная среда для работы с большими объёмами однотипной информации. Наиболее популярной электронной таблицей в России является Excel.

Таблица представляет собой рабочую область, разбитую на столбцы и строки, пересечения которых образуют **ячейки**, в которые записывается информация. Каждая строка имеет свой уникальный числовой номер, каждый столбец — своё уникальное буквенное имя. Для обеспечения обработки содержимого ячеек таблицы каждой ячейке присваивается также уникальный адрес (или имя). Адрес каждой ячейки формируется из номера буквенного имени столбца и номера строки, которым эта ячейка принадлежит (например, запись B7 означает, что это ячейка седьмой строки и столбца с именем «B»). В каждый момент времени активной (т.е. той, над которой производятся какие-либо действия) является только одна ячейка.

Любая ячейка электронной таблицы может содержать числовое значение, текстовую информацию или формулу.

Числовые данные, хранящиеся в электронной таблице, могут в зависимости от реализации электронной таблицы иметь различные формы представления (так в Excel принято использовать такие типы, как числовой десятичный, числовой экспоненциальный, денежный, типы дата, время, процент).

Текстовые данные, как правило, отображаются именно так, как были введены и обрабатываются как строки вне зависимости от их содержания.

Формулы в ячейках электронных таблиц позволяют содержимое текущей ячейки вычислять, используя значения других ячеек, при этом в формулах используются ссылки на адреса ячеек (т.е. их имена). Ссылки бывают относительные (обозначаются, например, B4) и абсолютные (\$B\$4). Относительная ссылка имеет смысл смещения ячейки, из которой необходимо подставить в формулу данные, относительно активной ячейки. При копировании формулы такие ссылки автоматически пересчитываются, сохраняя постоянным относительное расположение источника данных по отношению к ячейке, их принимающей и использующей. Абсолютная ссылка, в отличие от относительной, ссылается на конкретную ячейку в таблице и не меняется при перемещении самой формулы. Результат, отображающийся в ячейке, имеющей тип «формула», при изменении данных в ячейках, на которые производятся ссылки, пересчитывается автоматически.

Современные реализации электронных таблиц позволяют производить сортировку данных (в алфавитном порядке — для текстового формата ячеек, по возрастанию и убыванию — для числового), сортировка может производиться сразу по нескольким столбцам, но в этом случае необходимо задать приоритетность сортировки по каждому из столбцов или строк.

Электронные таблицы позволяют осуществлять поиск данных в больших объёмах информации, занесённых в них. Для осуществления поиска пользователь указывает условие или ряд условий, которым должны удовлетворять найденные данные. Такие запросы носят название фильтров и формируются при помощи специальных условий поиска, которыми могут служить математические отношения $<$, $>$, $=$ и логические операции (И, ИЛИ, НЕ).

Помимо этого, большинство современных электронных таблиц позволяет пользователю менять формат вывода данных в ячейках на экран, применять к числовым данным математические операции (как элементарные, так и операции высшей математики), получать статистические данные, искать максимальные и минимальные значения. Удобными и современными средствами, предоставляемыми электронными таблицами являются возможности создания диаграмм и таблиц по результатам анализа значений, находящихся в ячейках, из заданного диапазона столбцов и строк. Помимо этого следует отметить, что эффективность использования электронных таблиц особенно ощутима при работе с большими объёмами данных, которые можно вводить не вручную, а загружать из сформированных заранее простейших файлов.

2. Построение алгоритма для управления учебным исполнителем (основные алгоритмические структуры). Демонстрация полученного алгоритма в среде учебного исполнителя.

БИЛЕТ 19

1. Основные принципы организации и функционирования компьютерных сетей. Интернет. Информационные ресурсы и сервисы компьютерных сетей: Всемирная паутина, файловые архивы, интерактивное общение. Назначение и возможности электронной почты. Поиск информации в Интернете.

Компьютерная сеть представляет собой объединение информационных ресурсов, которые могут быть удалены друг от друга. Сети бывают:

- **локальные** (объединение компьютеров, расположенных на небольших расстояниях друг от друга, например, в пределах одного дома);

- **региональные** (объединение компьютеров и локальных сетей одного региона, области);
- **корпоративные** (объединение локальных сетей в пределах одной корпорации);
- **глобальные** (объединение компьютеров, расположенных на значительном расстоянии, обеспечивающее доступ к мировым информационным ресурсам).

Формирование сетей стало возможным, в частности, благодаря созданию унифицированных протоколов, т.е. совокупностей правил, регламентирующих формат и процедуры обмена информацией между двумя или несколькими устройствами, а также из-за создания большого количества устройств, способных передавать и получать большие объёмы информации на значительные расстояния. Примером таких устройств являются, например, модемы, передающие сигналы по телефонной линии, или сетевые адаптеры. В данный момент наиболее современными являются устройства, обеспечивающие беспроводной доступ к компьютерным сетям. Помимо этого широкому распространению компьютерных систем способствует появившееся большое количество различных программ (браузеров, почтовых программ, программ обмена короткими сообщениями), поддерживающих возможность работы в сети и облегчающих её для пользователя.

Интернет — это глобальная сеть, пользователям которой предоставлен доступ к мировым информационным ресурсам и сервисам. Базовыми его сервисами являются следующие нижеперечисленные службы.

- **Электронная почта** предоставляет возможность пересылки различного рода сообщений между пользователями сети. Пересылаемая информация может содержаться как в текстовых, так и в аудио-, видеофайлах.

- **Поисковые серверы** осуществляют в автоматическом режиме просмотр информации в сети и создание базы ссылок на неё. По запросу пользователя поисковый сервер предоставляет пользователю адреса интересующих его сайтов. Наиболее популярные в России поисковые серверы Yandex, Google, Rambler.

- **Телеконференции** предоставляют возможность пользователям обмениваться информацией на определённую тему. Для обеспечения работы конференции на ряде серверов выделяются специальные почтовые ящики, работа которых синхронизирована. Пользователи имеют доступ ко всей информации конференции вне зависимости от того, с каким сервером они связываются сами.

- **Справочные службы.**

- **Системы удаленного управления**, обеспечивающие оборот электронных денег, дают возможность совершать электронные платежи и управление банковскими счетами.

- **Файловые архивы** представляют собой серверы, на которых размещаются либо свободно распространяемые программные продукты, либо демоверсии программных продуктов, не находящихся в свободном распространении (последние, как правило, с ограниченным сроком действия или ограниченным набором функций).

- **WWW (World Wide Web)** — это огромное множество серверов в Интернете, содержащих так называемые Web-страницы, созданные на основе гипертекста. Для просмотра такой страницы необходимо подключиться к Интернету, запустить какой-либо браузер (например, Internet Explorer), в строке «адрес» набрать адрес необходимой страницы.

Остановимся подробнее на работе с электронной почтой. Для того чтобы пользователь мог эффективно принимать электронную почту ему необходимо завести почтовый ящик на почтовом сервере, а чтобы отправлять корреспонденцию — знать уникальный электронный адрес адресата. **Почтовый сервер** — это некоторый удаленный компьютер, подключенный к сети, обеспечивающий приём, отправку и хранение электронных писем. **Почтовый ящик** — это выделенная часть дискового пространства на почтовом сервере для хранения электронной почты адресата. Для доступа к корреспонденции, хранящейся в почтовом ящике, необходимо знать его имя и пароль.

Электронный адрес строится следующим образом: `username@hostname.domain`. Здесь `username` — сетевой псевдоним пользователя (или логин), `hostname` — имя почтового сервера («хоста»), `domain` — доменный адрес (обозначает группу ресурсов, находящихся в управлении одного «узла» в сети). Домены, как правило, бывают либо административными (`com` — коммерческая организация, `edu` — образовательное учреждение, `net` — сетевая организация), либо географическими (`ru` — Россия, `de` — Германия, `us` — США).

2. Обработка цифрового изображения и графическом редакторе. Например, устранение дефектов, ретуширование и тоновая коррекция фотографии.

БИЛЕТ 20

1. Понятие модели. Информационная модель. Виды информационных моделей (на примерах). Реализация информационных моделей на компьютере. Пример применения электронной таблицы в качестве инструмента математического моделирования.

На протяжении всей своей жизни человек постоянно сталкивается с различными копиями (или моделями) предметов и существ из окружающего его мира. Все эти копии не обладают всеми свойствами своего прототипа, но имеют сходные с ним характерные черты. Так детские игрушки передают внешний облик людей, автомобилей, животных, их пропорции, цвет, велотренажер имитирует поездку на настоящем велосипеде. Подбор черт, в которых необходимо осуществить схожесть оригинала и его модели зависит от цели создания модели. Например, и географическая карта, и глобус передают очертания береговой линии океанов, границ стран, и там и там обозначены русла рек и озера. Однако сферическая форма глобуса и его крепление в подставке под некоторым углом, передают форму земного шара в целом, а также угол наклона оси планеты к плоскости орбиты. Итак, *модель* — аналог (заменитель) оригинала, отражающий некоторые его характерные черты. Приведенные выше примеры моделей относятся к так называемым *материальным моделям*, так как сами являются материальными объектами.

Другой тип моделей — *абстрактные модели*. К ним относятся математические формулы, чертежи, таблицы, диаграммы. В этом случае может моделироваться не объект, а целый процесс, явление или межчеловеческие отношения. Например, самыми распространенными кинематическими моделями в физике являются модели равномерного и равноускоренного движений, описываемые соответствующими формулами для определения координат и скоростей, при этом предлагаемые формулы справедливы не для конкретного тела, а для широкого класса объектов.

При наличии достаточной и точной информации путем представления её в некотором виде, доступном для передачи, хранения, обработки или анализа, можно говорить о создании информационной модели. Таким образом, *информационная модель* — целенаправленно отобранная информация об объекте, представленная в некоторой форме. Возможны следующие формы представления информации: в виде жестов или сигналов, в устной или словесной форме, в виде символов (например, числа или ноты), графически или в виде таблицы.

Представление информационных моделей может быть осуществлено в *образной* или *знаковой* форме. Образная форма представления означает наличие некоторой визуальной информации, зафиксированной на каком-либо носителе. Знаковые информационные модели подразумевают использование некоторого языка записи данных (формул, языка программирования, таблицы).

Другое представление информационной модели на компьютере — это *электронная таблица*, позволяющая хранить и во многих случаях автоматически обрабатывать большие объёмы данных. Примерами применения таких таблиц могут быть электронный журнал успеваемости учеников в школе, таблицы прихода и расхода денежных средств, используемые в бухгалтерии, таблицы, содержащие сведения о расстоянии между населёнными пунктами, стоимости проезда и расписании движения в транспортных компаниях. Современные таблицы также позволяют ученым по экспериментальным данным (например, значению переменной времени и текущей координате) определить траекторию полета воздушного судна, определить предполагаемую его скорость и местоположение в некоторой промежуточной точке, не вошедшей в статистическую выборку.

2. Построение алгоритма для обработки величин с реализацией на языке программирования (ветвление, цикл, линейный массив или вспомогательные алгоритмы).

ЧАСТЬ II

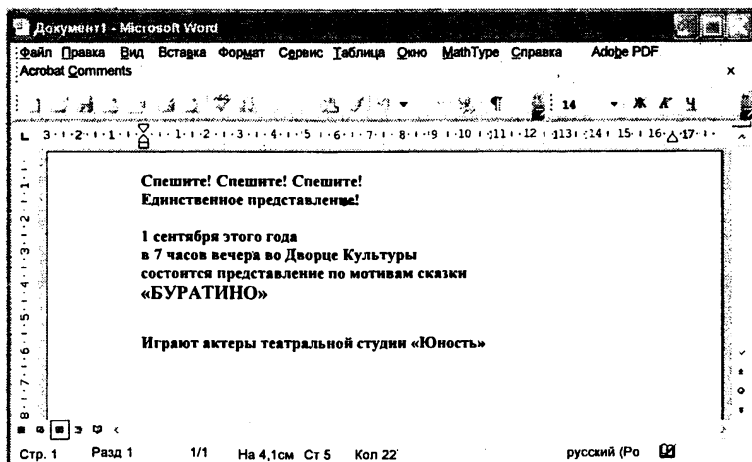
ПРИМЕРНЫЕ ВАРИАНТЫ ОТВЕТОВ НА ПРАКТИЧЕСКИЕ ЗАДАНИЯ БИЛЕТА

1. Создание и редактирование текстового документа (исправление ошибок, удаление или вставка текстовых фрагментов), в том числе использование элементов форматирования текста (установка параметров шрифта и абзаца, внедрение заданных объектов в текст).

Задание 1.** Подготовить афишу к спектаклю «Буратино», подготовленного театральной студией «Юность». При подготовке текстового документа использовать различные размеры, начертания и виды шрифтов. Список действующих лиц и исполнителей подготовить при помощи таблицы. Использовать внедрённые объекты: рисунок, символ, WordArt.

Для выполнения задания необходимо осуществить следующие действия.

Открыть текстовый редактор (например, Word). Набрать текст афиши.



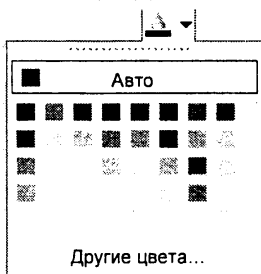
Путем выделения при помощи мыши «текстовых фрагментов» изменить атрибуты шрифта каждого фрагмента:

- размер



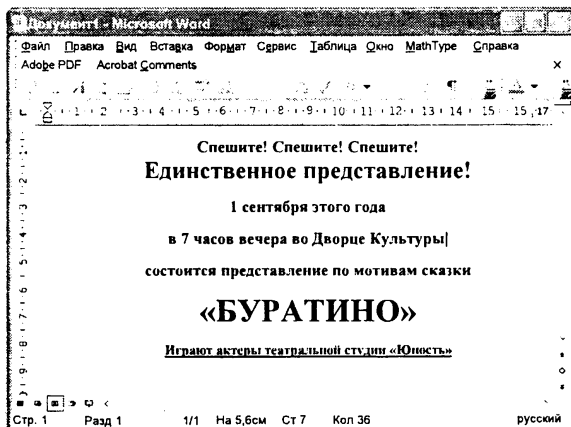
- начертание Ж К Ч (полужирный, курсив или подчеркнутый),

- цвет



Выбрав в меню команд «**Формат**», «**Шрифт**», можно воспользоваться расширенным списком атрибутов шрифта. Помимо этого следует выделить и отцентрировать текст при помощи инструмента «**По центру**» . Вставить нажатием клавиши Enter дополнительные пробелы между строками текста. Проверить орфографию, выбрав в меню «**Сервис**», «**Правописание**».

После редактирования текст примет следующий вид.

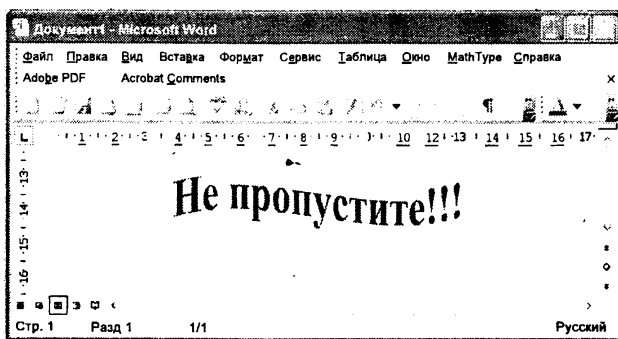


Далее следует выбрать в меню команд «Вставка», «Рисунок», «Из файла» и загрузить заранее заготовленный графический файл с эпизодом спектакля, который следует вставить.

Для печати списка играющих актеров выберем в меню «Таблица», «Вставить», «Таблица», зададим число строк (например, 6) и столбцов (3). Первая колонка предназначена для названия роли, вторая и третья для фамилии и имени исполнителя. Заполним строки и столбцы таблицы (например, Буратино ☺ | Иванов | Володя, Мальвина♥ | Петрова | Даша). После этогоотрегулируем ширину столбцов и самой таблицы согласно собственному вкусу: наведём указатель мыши на границу колонки или таблицы (он примет вид двух параллельных полосок) и перетащим её. При печати текста в строках использовались специальные символы (меню «Вставка», «Символ»). Выделив таблицу (ячейки закрашены темным цветом), щелкнем по ней правой кнопкой мыши и выберем «Свойства таблицы». Далее выберем вкладку «Таблица», нажмем кнопку «Границы и заливка», затем «Границы», «Тип», «Нет». В результате выполнения этих действий на экране останется только содержимое таблицы (без рамки).

Вставим в конец документа объект **WordArt**. Для этого последовательно выберем «Вставка», «Рисунок», «Объект WordArt». Выберем понравившийся тип изображения надписи и введем в текстовое поле слова «Не пропустите!!!».

Получится такая необычная надпись:



Задание 2*.** Разработать гипертекстовый документ «Видеотека», в котором содержится список видеофильмов (не менее трёх) и ссылки на данные об актёрах и режиссерах фильмов. В свою очередь, актёрские и режиссёрские страницы содержат ссылки на аннотации к фильмам.

Гиперссылка — это фрагмент документа, указывающий на другой файл, web-страницу или на другую страницу текущего документа. Гиперссылка для пользователя представляет собой графическое изображение или текст (как правило, выделенный цветом), позволяющие переходить к другим объектам при нажатии гиперссылки.

Для выполнения задания предварительно создадим в программе Word многостраничный документ следующего содержания. На первой странице перечислим названия популярных кинофильмов (например, «Небесный тихоход», «Бриллиантовая рука», «Война и мир»). Отдельно на следующих страницах документа (желательно на разных для каждого фильма) перечислим имена актеров, в этих фильмах игравших (Василий Меркурьев, Николай Крючков; Людмила Савельева, Вячеслав Тихонов; Андрей Миронов, Юрий Никулин, Нонна Мордюкова). На следующих страницах (также на разных) опишем краткое содержание фильмов (например, «Советский музыкальный художественный фильм про летчиков на войне», «Комедия про незадачливых мошенников и пострадавшего Семена Семеновича», «Экранизация романа Льва Толстого, режиссер Сергей Бондарчук»).

Для установления гиперссылок предварительно установим в получившемся документе закладки около имён и фамилий актёров в каждом фильме (т.е. отметим места в тексте для последующего лёгкого перехода к ним по имени ссылки). Для этого установим курсор около первой в перечислении фамилии артиста и выберем **«Вставка»**, **«Закладка»**, введём имя закладки (пусть, для определённости, это будут соответственно имена «артисты_небесного_тихохода», «артисты_бриллиантовой_руки», «артисты_войны_и_мира»). Около содержания фильмов тем же способом установим закладки «содержание_Небесный_тихоход», «содержание_Бриллиантовая_рука», «содержание_Война_и_мир».

Далее выделите текст, который должен представлять гиперссылку (в нашем случае это названия кинофильмов и фамилии артистов этих фильмов). На панели инструментов **Стандартные** нажмите кнопку **Добавление гиперссылки** , в списке **Связать** выберем параметр «с местом в этом документе». Выберем в списке соответствующую закладку для ссылки. В результате при установке указателя на гиперссылку на экране будет появляться всплывающая подсказка (например, «Нажмите CTRL и щелкните ссылку»). После выполнения указанного действия, будет осуществлен быстрый переход к новой странице документа.

2. Создание и обработка графических изображений средствами графического редактора. Ввод изображения через сканер или с цифрового фотоаппарата. Простейшая обработка цифрового изображения.

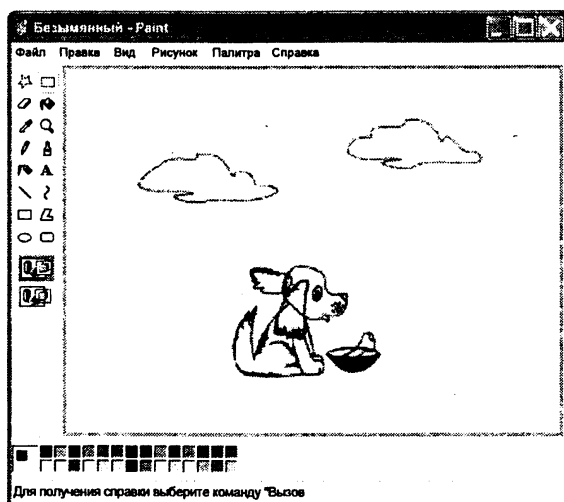
Задание 1*.

1. Запустить графический редактор PAINT.
2. Открыть файл *dog.jpg*.
3. Дорисовать будку для собаки и любые другие недостающие детали.
4. Раскрасить получившийся рисунок.

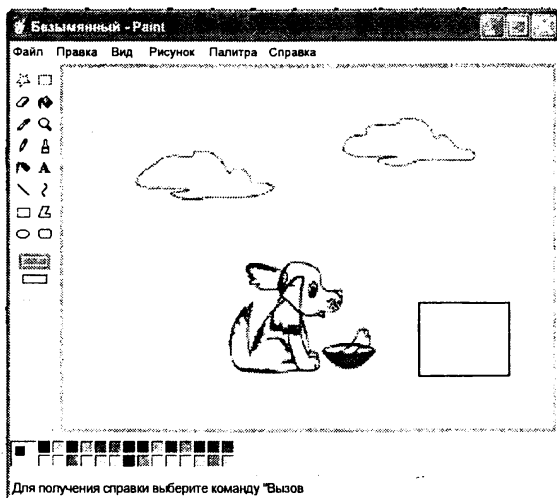
Пример выполнения задания следующий.

1. Запустить графический редактор PAINT, выбрав в меню «Пуск» последовательно «Все программы», «Стандартные», «Paint».

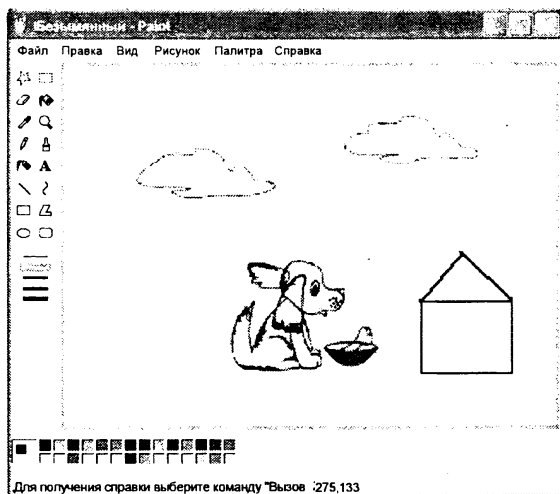
2. Открыть файл *dog.jpg*, указав заранее известный путь к данному файлу (меню «Файл», «Открыть»), затем указывается путь к файлу). Программа отобразит рисунок, содержащийся в файле.



3. Чтобы дорисовать будку для собаки, необходимо в наборе инструментов, расположенных в левой части экрана выбрать инструмент «Прямоугольник», подвести курсор к области рисунка, где желательно увидеть угол будки, нажать левую кнопку мыши и, не отпуская кнопки, «растянуть» возникший прямоугольник переменного размера до желаемых пропорций, отпустить кнопку мыши.

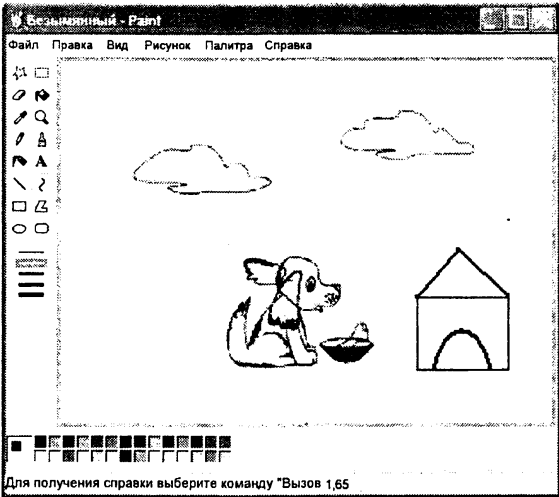


Выбрать инструмент «Линия», в появившемся снизу от набора инструментов наборе выбрать толщину линии. Нарисовать крышу будки, подведя курсор мыши к началу линии, нажав левую кнопку мыши и «растянув» появившийся отрезок до желаемой длины и в нужном направлении.

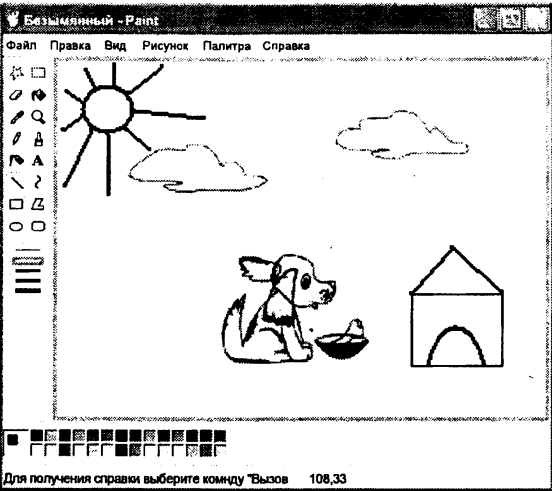


Выбрать инструмент «Кривая». Нарисовать вход в будку. Для этого подвести курсор мыши к краю будущего входа, расположенного на полу будки, щелкнуть левой кнопкой мыши, не отпуская её,

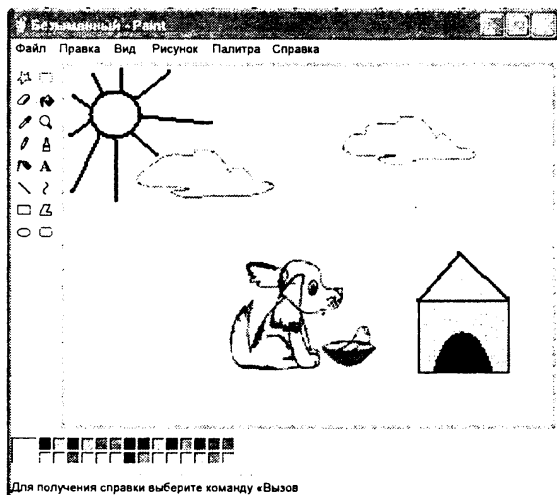
подвести курсор к симметричной точке входа в будку, отпустить кнопку. Затем необходимо щелкнуть левой кнопкой мыши в том месте, где линия должна изгибаться, и перетащить появившийся указатель до создания нужной кривизны.



Выбрать инструмент «Эллипс». Щелкнув левой кнопкой мыши и, не отпуская её, потянув, нарисовать солнце. Выбрать инструмент «Линия», и, задав желаемую толщину, нарисовать лучи солнца.



4. Раскрасить получившийся рисунок можно, выбрав инструмент «Заливка» и желаемый цвет из палитры внизу экрана. Например, выбрав цвет закрашивания жёлтый, можно раскрасить солнце, щелкнув левой кнопкой мыши внутри круга, изображающего солнце. Далее можно выбрать зелёный цвет и закрасить будку. Затем выбрать чёрный и закрасить вход в будку. Заметим, что заливка цветом происходит от выбранной точки до границ с другими фигурами. Если при этом граница закрашиваемой фигуры не является непрерывной, то выбранным цветом могут быть залиты лишние области рисунка. Если такое неприятное явление все же произошло, то следует выбрать в меню «Правка» команду «Отменить», после чего произойдет отмена последнего совершенного действия. Цвета линий (например, лучей солнца) также можно поменять (например, на жёлтый). Для этого необходимо произвести те же действия, что и для закрашивания области, только произвести щелчок кнопкой мыши, предварительно направив курсор на соответствующую линию.



Задание 2**.

1. Запустить графический редактор (Paint, PhotoShop, CorelDraw).
2. Создать по образцу рисунок, используя максимально возможное количество инструментов рисования. (Образец рисунка взять из предыдущего задания).

Приведём примерную последовательность действий для выполнения данного задания в графическом редакторе Paint. Подробные комментарии дадим только для инструментов, которые не были использованы при описании выполнения **Задания 1** (см. выше).

1. Запустить графический редактор Paint, выбрав в меню «Пуск» последовательно «Все программы», «Стандартные», «Paint».

2. Основным инструментом для рисования выберем «Карандаш». При нажатой левой кнопки мыши будем перемещать курсор по экрану повторяя основные контуры рисунка (контуры облаков и собачки). Детальная прорисовка собачки при этом не требуется и будет осуществляться позже при помощи других инструментов. При непроизвольном появлении лишних линий и штрихов осуществим их стирание при помощи «Ластика».

Прорисуем детали рисунка. Для этого при помощи инструмента «Масштаб» (или, как его иногда называют, «лупа») визуально приблизим область рисунка, где требуется произвести детализацию. Нарисуем глаз собачки, используя инструмент «Эллипс». Для изображения зрачка выберем инструмент «Кисть», зададим её форму в виде жирной точки, выбрав из предложенных под набором инструментов. Щелкнув на экране компьютера левой кнопкой мыши, нарисует сам зрачок. Произведём заштриховку пятнышек на хвостике, ухе и шее собачки при помощи того же инструмента, но в форме наклонного штриха. Сгладим острые углы, получившиеся на контурах рисунка, стерев их при помощи «Ластика» и исправив их форму при помощи «Карандаша». Использование карандаша при этом несколько иное, чем прежде. Ранее он использовался для рисования длинных линий, здесь он используется для практически поточечного рисования (кнопка мыши нажимается и тут же отпускается, появляющееся на экране изображение представляет собой точку).

Нарисуем миску, создав её при помощи «Эллипса» (верхняя часть) и «Кривой» (нижняя часть). Косточку нарисует, проведя две параллельные прямые (инструментом «Линия») и изобразив рядом на конце косточки две окружности. Далее опять применив «Карандаш» плавно соединим окружности между собой и с прямыми. Ластиком наименьшего размера сотрем внутренние линии, оставив только контур кости.

Раскрасим миску на рисунке, выбрав инструмент «Заливка» и синий цвет.

Задание 3***

1. Отсканировать предложенное учителем изображение или фото.
2. Сохранить изображение или фото в нужном формате.
3. Запустить графический редактор PhotoShop (или другую программу для просмотра фотографий).
4. Загрузить предложенный учителем файл с фотографией.
5. Выполнить устранение дефектов, имеющих на фотографии.
6. Выполнить ретуширование и тоновую коррекцию фото.
7. Выполнить кадрирование и сохранение изображения под другим именем.

Примерный порядок выполнения работы следующий.

1. Запустить программу сканирования, вложить в сканер фото или рисунок.

Если данная функция поддерживается используемой в классе программой сканирования, осуществить предварительный просмотр рисунка, выбрать область изображения, которую необходимо оцифровать (например, отдельную фигуру из группы на рисунке).

2. Указать папку и путь к ней для хранения отсканированного изображения, выбрать разрешение (чем оно выше, тем качественнее рисунок), указать тип рисунка (цветной или чёрно-белый), выбрать формат изображения (jpg).

3. Запустить графический редактор PhotoShop, в меню «**Файл**» выбрать «**Открыть**» и указать путь к сохранённому файлу.

4. Открывшийся файл сохранить в формате bmp («**Файл**», «**Сохранить как**», тип файла указываем bmp). Вновь открываем файл уже в формате bmp.

5. Устраним основные имеющиеся на фотографии изъяны (грязь, царапины, неровности на лицах). Для этого среди инструментов необходимо выбрать «**Healing Brush Tool**» («исцеляющая кисть»). В появившемся окне выбрать свойства кисти (толщину «пера», например 27, плавность линий «**Flow 50%**», коэффициент матовости рисунка «**Opacity 50%**»). Затем первым щелчком мыши выбирается участок картинка хорошего качества, последующие щелчки «исцеляют» участки ненадлежащего качества, беря за эталон первый выбранный участок рисунка. Устранить некоторые дефекты можно также при помощи фильтров «шум» и «смазывание» (**noise** и **blur**), варьируя их параметры.

6. Произведем ретуширование изображения. Для этого будем работать с так называемыми слоями. Продублируем первый слой (фон), для этого, например, можно нажать CTRL+J. Дальнейшие действия будем производить с продублированным слоем, не трогая оригинал (в этом случае при совершении ошибочных действий или ненадлежащем результате исправлений мы всегда сможем вернуться к первоначальному изображению). Выбираем для работы новый продублированный слой, отключив оригинал (для этого необходимо произвести щелчок по маленькой иконке с глазом напротив имени слоя в панели с закладками). Если необходимо произвести коррекцию над конкретными частями лица или фигуры, то их необходимо представить в виде самостоятельных слоёв, предварительно выделив при помощи инструмента «**Лассо**», и корректировать отдельно, отключив остальные слои. Опишем более подробно последовательность действий, необходимую для устранения эффекта «красных глаз» от вспышки фотоаппарата на фотографии: «красный» глаз выделяется при помощи инст-

румента «Эллипс» или «Лассо», выбирается инструмент «Тон/Насыщенность», устанавливается желаемый цвет глаз. Чтобы избежать присутствия на отредактированной фотографии красного ободка по краю зрачка лучше захватить в выделяемую область редактирования на пиксель больше, а для улучшения эффекта край области размыть (опция «Quick Mask» и фильтр «Gaussian Blur»). Для создания эффекта естественности можно добавить блики на зрачок (фильтр «Lens Flare»). Описанная процедура представляет собой ручную коррекцию изображения, однако, программа PhotoShop предоставляет возможность производить устранение данного дефекта полуавтоматически при помощи специальной опции, хотя при её использовании в некоторых случаях пользователь может получить эффект «мертвого» изображения (т. е. потери естественности).

Далее произведём цветокоррекцию фотографии (тонирование). Для этого следует наложить несколько слоев одного и того же изображения, изменяя их прозрачность и заливая их краской, прозрачность и цвет которой также можно варьировать в зависимости от эффекта, которого необходимо добиться.

7. Произведём кадрирование (т.е. обрезку изображения) до желаемого размера. Для этого необходимо выбрать инструмент **Crop** (Кадрировать) или **Trim** (Подрезать) и нарисовать рамку вокруг изображения, которое необходимо оставить. Можно также установить флажок **Shield cropped area** (Прикрыть область кадрирования), если необходимо, чтобы область за пределами рамки была затемнена.

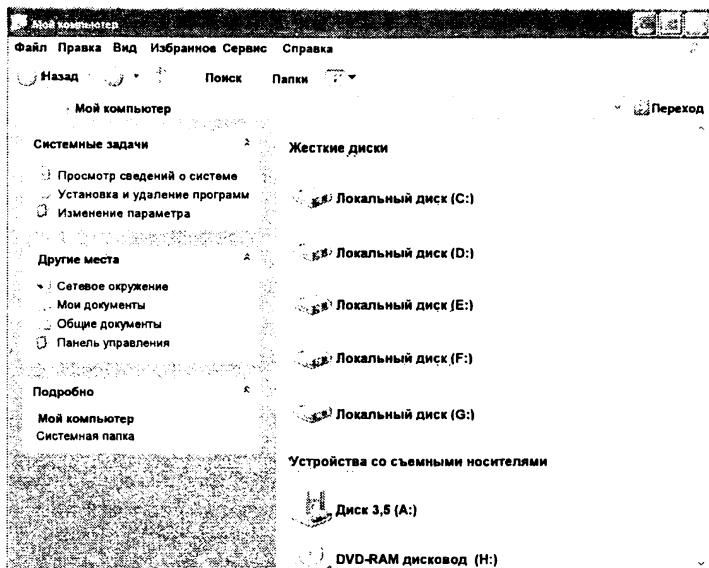
Сохраним изображение под другим именем («Файл», «Сохранить как», указать новое имя файла).

3. Работа с файловой системой, с графическим интерфейсом (выполнение стандартных операций с файлами: создание, копирование, переименование, удаление). Организация индивидуального информационного пространства (настройка элементов рабочего стола, проверка на вирусы, использование архиватора).

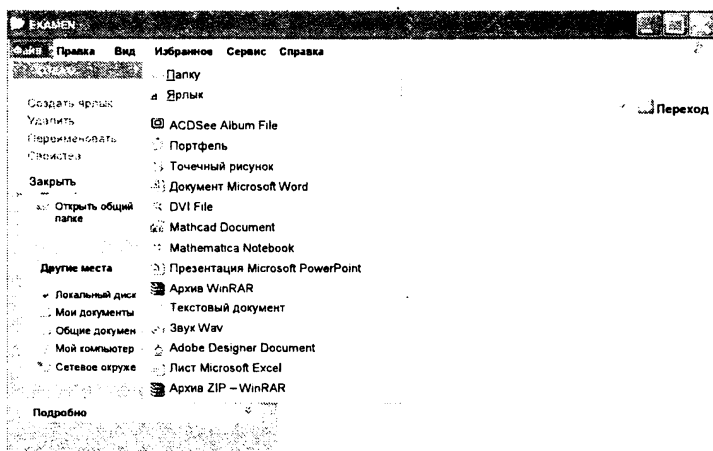
Задание 1.* (Состоит из 12 пунктов, необходимых для выполнения.)

1. Создать в папке C:\EXAMEN папку с именем NEW

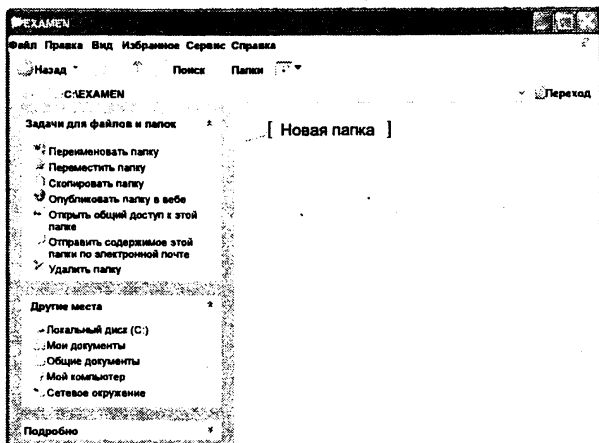
Подвести курсор мыши к иконке «Мой компьютер» на Рабочем столе. Достаточно быстро два раза нажать левую кнопку мыши («кликнуть»). На экране появится окно «Мой компьютер».



Подвести курсор к иконке «Локальный диск (C:)» и «кликнуть» (если диск C: в окне не виден, воспользоваться полосой прокрутки). Появится окно «Локальный диск (C:)». В этом окне найти папку EXAMEN, подвести на неё курсор и «кликнуть». Появится окно EXAMEN. Подвести курсор к пункту меню «Файл» и один раз нажать на левую кнопку мыши. Появится список доступных команд. В этом списке навести курсор на команду «Создать» (кнопки мыши при этом нажимать не надо!). Появится список с командой «Папку» (см. рис.).



Навести на неё курсор и один раз нажать левую кнопку мыши. Появится папка с названием «Новая папка», выделенным синим цветом.

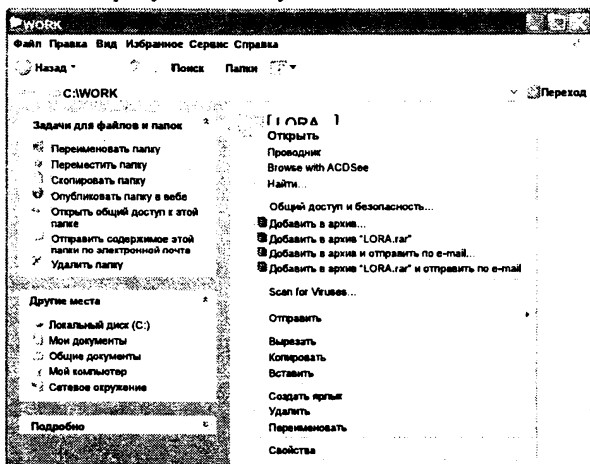


На клавиатуре набрать “NEW” и нажать ВВОД.

Примечание. Если в результате неосторожных или неаккуратных действий название папки перестало выделяться синим цветом до начала набора или исчезли в процессе набора квадратные скобки (т.е. папке уже присвоено имя), то папку можно переименовать (см. ниже). Кроме того, папку можно удалить (см. ниже) и повторить процесс заново.

2. Скопировать в созданную папку NEW из папки C:\WORK папку LORA и файлы *gramota.doc*, *uspev.xls*.

Открыть папку C:\WORK. Найти папку LORA. Навести на неё курсор и нажать правую клавишу мыши. Появится меню команд



В этом меню навести курсор на команду **«Копировать»** и нажать левую кнопку мыши.

Возможен также другой способ копирования. В папке WORK навести курсор на папку LORA и нажать левую кнопку мыши (далее это будем называть **«Выбрать»**). Затем в меню команд папки WORK нажать команду **«Правка»** и выбрать команду **«Копировать»**.

Чтобы вставить скопированную папку LORA в папку NEW, открыть папку NEW (если она была закрыта или не открывалась ранее), подвести курсор к любому свободному месту в окне (белый фон), нажать правую кнопку мыши, выбрать пункт **«Вставить»** и нажать левую кнопку мыши.

Возможен другой вариант вставки. В папке NEW в меню команд выбрать пункт **«Правка»**, команду **«Вставить»** и нажать левую кнопку мыши.

Аналогично копируются файлы *gramota.doc.uspev.xls*.

Можно копировать сразу несколько файлов следующим способом. Нажать клавишу Ctrl на клавиатуре и, не выпуская, выбрать все файлы и папки, которые следует скопировать. При этом они выделяются цветом. Наведя курсор на любой из файлов, нажать правую кнопку мыши, выбрать пункт **«Копировать»** и нажать левую кнопку мыши. Далее файлы вставляются в нужную папку как было описано выше.

3. Зайти в папку C:\EXAMEN и переименовать папку LORA в папку STAR.

Открыть папку C:\EXAMEN. Выбрать папку LORA. Папка выделится цветом. Навести курсор на имя папки и ещё раз нажать левую кнопку мыши. Название папки появится в квадратных скобках. На клавиатуре набрать STAR и **«Ввод»** (кнопка Enter).

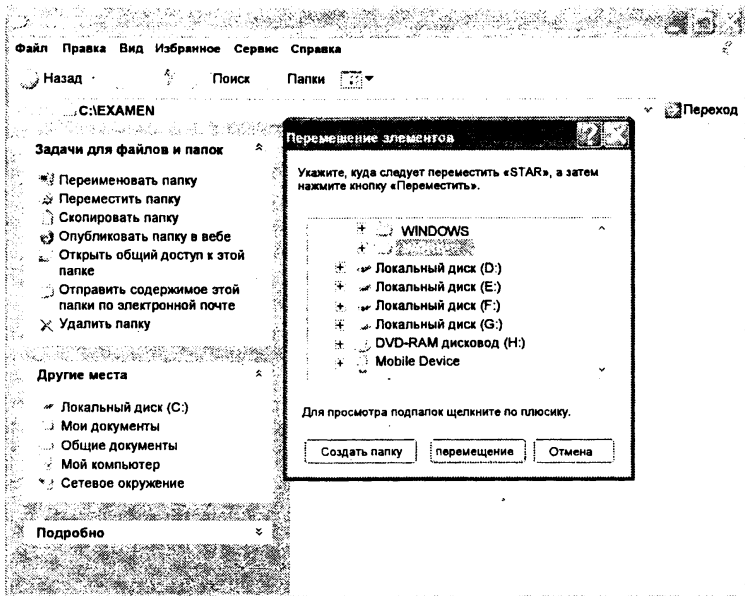
Другой метод: выбрать папку LORA. В меню команд окна выбрать **«Файл»**, **«Переименовать»**. Далее действовать как было описано выше.

4. Переместить папку STAR в папку C:\WORK.

Выбрать папку STAR. Нажать правую клавишу мыши и выбрать **«Вырезать»**. Папка исчезнет из окна.

В папке WORK проделать процедуру вставки, как было описано в пункте 2.

Другой вариант выполнения: выбрать папку STAR. В меню команд окна выбрать **«Правка»**, **«Переместить в папку»**. Появится меню (см. рис.).



В этом меню найти и выделить папку **WORK** и нажать «перемещение».

5. Удалить из папки **C:\EXAMEN** папку **NEW**, а из папки **C:\WORK** — папку **STAR**.

Выбрать соответствующую папку и нажать правую кнопку мыши, выбрать «Удалить».

Другой способ: выбрать папку и нажать на клавиатуре «Delete».

6. Очистить корзину.

На рабочем столе выбрать иконку «Корзина» и нажать правую кнопку мыши. Выбрать «Очистить корзину».

7. Вывести содержимое папки **C:\WORK\TEST\DATA** в полной форме (Таблица), отсортировав её по размеру файлов (по убыванию).

Открыть папку **C:\WORK\TEST\DATA**. В меню команд выбрать «Вид», «Таблица». В окне появится таблица. Найти столбец «Размер» и нажать на него. Если файлы отсортировались по возрастанию, а нужно отсортировать по убыванию, следует ещё раз нажать клавишу «Размер».

8. Определить, какой файл в папке C:\WORK\TEST\DATA имеет самый большой размер.

В отсортированном по убыванию содержимом папки самый верхний файл будет иметь самый большой размер.

9. Вывести содержимое папки C:\WORK\TEST\DATA в полной форме (Таблица), отсортировав её по времени создания файлов (по возрастанию).

Сортировка по времени создания осуществляется аналогично сортировке по размеру путём нажатия надписи столбца даты.

10. Создать в папке C:\EXAMEN папку с именем ARJ.

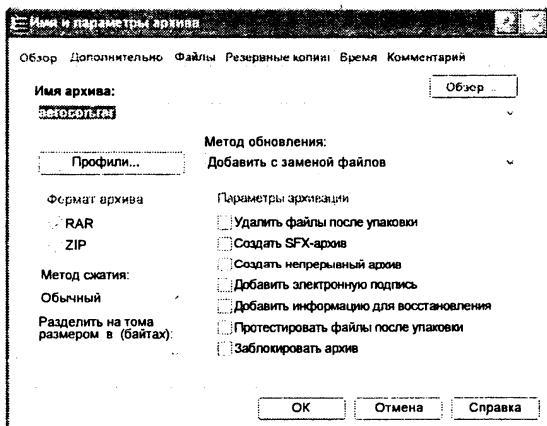
Выполняется аналогично пункту 1.

11. Проверить все файлы в папке DATA на наличие вирусов.

Для большинства современных антивирусных программ возможна следующая процедура. Выбрать папку DATA и нажать правую кнопку мыши. Выбрать пункт «Проверить на вирусы» (или «Scan for Viruses» или подобный пункт, в зависимости от установленной антивирусной программы).

12. Заархивировать все файлы из папки DATA и поместить архив в папку ARJ.

Для большинства современных архиваторов (например, WINRAR) возможна следующая процедура. Выбрать все файлы папки DATA (используя клавишу Ctrl) и нажать правую клавишу. Выбрать команду «Добавить в архив». В появившемся окне при необходимости изменить название архива, метод архивации и нажать «Ввод» (см. рис.).



Переместить архив в папку ARJ.

Для создания многотомного архива следует в появившемся окне (см. рис.) выбрать максимально допустимый размер архива при помощи **«Разделить на тома размером (в байтах)»**.

Можно также воспользоваться непосредственно установленным архиватором.

Для извлечения всех файлов из архива выбрать архив и нажать правую клавишу мыши. Выбрать **«Извлечь файлы»**, в появившемся окне при необходимости выбрать путь для извлечения.

Задание 2.** (Состоит из 13 пунктов, необходимых для выполнения.)

1. Зайти в папку MYFOLDER.

Последовательно выбирая имя диска и папок в поле **«Адрес»**, найти папку MYFOLDER (если её местоположение известно заранее).

Если расположение папки неизвестно, то воспользоваться **«Поиском»** (меню кнопки **«Запуск»**).

2. Заархивировать все файлы папки MYFOLDER с расширением doc в архив с именем DOCUM (тип архива — RAR).

Запустить программу WinRar, найти папку MYFOLDER, щелкнув правой кнопкой мыши по свободному пространству на поле экрана выбрать **«Отсортировать по типу»**. При нажатой клавише Shift выделить блок файлов с расширением *doc*, нажать **«Добавить в архив»**, установить флажок **«формат архива RAR»**, ввести имя архива, нажать **«ОК»**.

3. Просмотреть содержимое созданного архива.

Для просмотра содержимого архива необходимо **«кликнуть»** по его имени, после чего на экране появится список содержащихся в нём файлов (визуально это выглядит как содержимое папки, носящей имя архива).

4. Заархивировать все файлы с расширением doc в архив с именем DOCUM (тип архива — ZIP).

Последовательность действий аналогична пункту 2, однако следует установить флажок **«формат архива ZIP»**.

5. Сравнить размеры двух архивов и определить, какой из форматов лучше сжимает файлы данного типа.

Для просмотра размера файла необходимо нажать на его имени правой кнопкой мыши, выбрать **«Свойства»**, вкладку **Общие**, где и будет виден размер данного файла. Произвести сравнение размеров файлов для обоих архивов.

6. Добавить в архив DOCUM.ZIP все файлы с расширением *ppt*.

Выделить все файлы с расширением *ppt*, нажать кнопку **«Добавить в архив»** и выбрать в строке **«Имя архива»** уже существующий архив DOCUM.ZIP.

7. Удалить из архива DOCUM.ZIP все файлы с расширением *doc*.

Осуществить просмотр содержимого архива как в пункте 3 данного задания. Далее выделив все требуемые файлы, нажать правую кнопку мыши, выбрать удалить, подтвердить желание удалить файлы в ответ на уточняющий вопрос системы.

8. Создать в папке MYFOLDER папки NEW1 и NEW2.

Действия по созданию папки описаны в предыдущем задании в первом пункте.

9. Заархивировать все файлы из папки MYFOLDER в многотомный архив с именем MYARCHIV и поместить его в папке NEW1.

Выбрать все файлы папки MYFOLDER (меню **«Правка»**, **«Выделить все»**) и нажать правую клавишу. Выбрать команду **«Добавить в архив»**. В появившемся окне выбрать максимально допустимый размер архива при помощи **«Разделить на тома размером (в байтах)»**. Задать имя архива MYARCHIV, при помощи кнопки **«Обзор»** указать путь его расположения (папка NEW1). Запустить архиватор.

10. Извлечь файлы из архива MYARCHIV в папку NEW1.

Для извлечения всех файлов из архива необходимо выбрать архив и нажать правую клавишу мыши. Выбрать **«Извлечь файлы»**, в появившемся окне при необходимости выбрать путь для извлечения.

11. Заархивировать все файлы с расширением *jpg* в самораспаковывающийся архив с именем PAPER, поместив его в папке NEW2.

Действия по архивации те же, что и ранее, но необходимо установить флажок **«Создать SFX-архив»** (от английского *self-extracting archive*).

12. Извлечь файлы из архива PAPER в папку NEW2.

Выбрать архив PAPER, нажать клавишу Enter, указать путь к папке NEW2, запустить процесс извлечения файлов из архива.

13. Удалить папки NEW1 и NEW2, а также все созданные вами архивы в папке MYFOLDER.

Выделить соответствующие папки и архивы и нажать клавишу Delete на клавиатуре, либо, щелкнув правой кнопкой мыши на имени файла или папки, выбрать «Удалить» в появившемся списке.

4. Создание мультимедийной презентации на основе шаблонов. Выбор типа разметки слайда, применение таблицы оформления, цветовых схем и эффектов анимации. Показ презентации с использованием автоматической смены слайдов.

Задание 1*. Подготовить на основе готового шаблона презентацию по теме «Устройство компьютера», состоящую не менее чем из 5 слайдов. Применить к объектам эффекты анимации. Настроить автоматическую демонстрацию слайдов.

Презентация — это демонстрационные материалы для публичного выступления. Под компьютерной презентацией понимают файл, в котором такие материалы собраны. Готовая презентация представляет собой последовательность слайдов, на которых могут быть представлены текст, графика, диаграммы.

Для выполнения задания запустим программу PowerPoint и для предложенной «чистой» презентации произведем изменения для создания полноценной презентации на тему «Устройство компьютера». В большом правом поле экрана мы видим текущий слайд, слева на закладке «Слайды» расположена последовательность всех слайдов (они представляют собой уменьшенные копии больших слайдов). Выбор слайда для редактирования происходит щелчком левой кнопки мыши на соответствующем слайде слева (если он не виден, то следует воспользоваться полосой прокрутки).

Для начала сохраним файл под именем compute_structure.ppt (ppt — стандартное расширение для файла презентации MS PowerPoint).

Предположим, что наша презентация должна состоять из пяти слайдов, информация на которых должна быть следующей:

- 1) Основные принципы работы компьютера;
- 2) Конструктивное устройство элементов компьютера;
- 3) Взаимодействие устройств, составляющих ЭВМ;
- 4) Различные типы современных периферийных устройств;
- 5) Современные разъемы в ЭВМ.

Для оформления презентации, как правило, используются стандартные шаблоны, заранее разработанные дизайнерами. Для выбора

шаблона следует в меню **«Формат»** выбрать **«Оформление слайда»**, затем подвести курсор и «кликнуть» по тому шаблону, который более всего приглянулся. Отметим, что оформление может быть изменено как у всех слайдов одновременно, так и у некоторых в отдельности: необходимо на кнопке со стрелкой справа от шаблона щелкнуть левой кнопкой мыши и выбрать **«Применить ко всем слайдам»** или **«Применить к выделенным слайдам»**. Заметим, что при оформлении презентации не следует выбирать слишком кричащие тона фона, кроме этого желательно устанавливать контрастные цвета фона и текста и представлять информацию крупно и тезисно, а не выводить на экран полный текст доклада или реферата. Отметим, что хорошим тоном выступления на публике также считается наличие на первом слайде названия доклада и имени выступающего. Можно также на последнем слайде вывести благодарность слушателям за внимание: **«Спасибо за внимание!»** с установленным эффектом анимации.

Заготовим необходимое число пустых слайдов (5). Для этого правой кнопкой мыши щелкнем по уменьшенной копии слайда в левой части экрана и выберем **«Создать слайд»**. Если в дальнейшем возникнет необходимость вставки слайда между двух существующих, то указанной действие необходимо производить на слайде, после которого необходимо осуществить вставку. Вырезать и вставить слайд в другое место можно выполнив такую последовательность действий: щелкнуть правой кнопкой мыши по слайду слева, выбрать **«Вырезать»**, затем подведя курсор к зазору между слайдами и дождавшись появления разделительной линии, нажать **«Вставить»** из меню **«Правка»**.

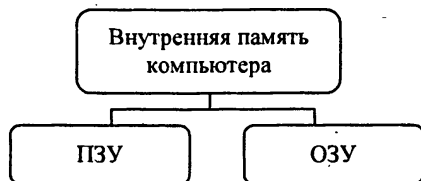
Для удобства работы лучше вывести на экран в правой части экрана панель **«Область задач»** (если эта панель скрыта), для этого необходимо кликнуть правой кнопкой мыши по свободному месту на любой панели и активизировать панель **«Область задач»** (т.е. поставить напротив неё галочку). Удобство этой панели в том, что она предоставляет готовые макеты наиболее популярных элементов презентации.

Далее на первом слайде в поле **«Заголовок слайда»** введем слова **«Основные принципы работы компьютера»**, на втором слайде — **«Конструктивное устройство элементов компьютера»** и т.д. Перетаскиваем при помощи мыши рамки с заголовками в верхние части слайдов.

На первом слайде, выбрав макет **«Заголовок и текст»**, в поле текста перечислим принципы работы компьютера (принцип программного управления, принцип однородности памяти, принцип ад-

ресности), после каждого из них при помощи нажатия кнопки Enter будет происходить печать следующего принципа с новой строки, начинающейся с маркера (жирной точки). Изменим размер рамки с текстом и отцентрируем её.

Отредактируем второй слайд. Выберем макет **«Заголовок, схема или организационная диаграмма»**, **«Отображение структурных отношений»**. Введём названия в появившиеся окошки: внутренняя память компьютера состоит из ПЗУ и ОЗУ. Результат будет примерно следующий:



Оформим третий слайд. Удалим рамку для поля текста слайда и при помощи инструментов на панели **«Рисование»** изобразим схему взаимодействия устройств компьютера.

На четвёртом слайде перечислим различные типы периферийных устройств (сканер, принтер и др.). Загрузим рисунки периферийных устройств из заранее заготовленных файлов. Для осуществления задуманного выберем макет **«Заголовок, текст и объект»**. Произведя щелчок кнопкой мыши на иконке **«Добавить рисунок»**, загрузим приготовленный рисунок.

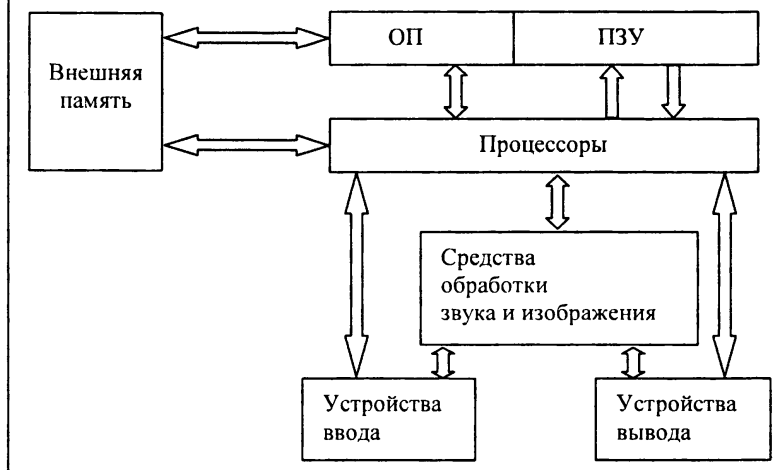
На пятом слайде перечислим основные современные разъемы (USB, COM-порт, S-Video). Аналогично предыдущему слайду выберем тот же макет, однако объектом для вставки выберем таблицу, заполнив её характеристиками существующих портов.

Создадим эффект анимации в презентации. Для этого выберем **«Показ слайдов»**, **«Эффекты анимации»**. Выберем, например, **«Появление»**.

Для настройки показа слайдов выберем **«Показ слайдов»**, **«Настройка презентации»**, **«Показ слайдов автоматический (полный экран)»**, помимо этого необходимо установить время смены слайдов: **«Показ слайдов»**, **«Смена слайдов»**, **«Автоматически»**, установить время и нажать **«Применить ко всем слайдам»**. Для запуска демонстрации созданных слайдов следует выбрать **«Вид»**, **«Показ слайдов»** (или нажать F5).

Один из перечисленных слайдов представлен на следующем рисунке.

Взаимодействие устройств, составляющих ЭВМ



Задание 2*.** Разработать презентацию «Видеотека», содержащую гиперссылки, в которых содержатся список видеофильмов (не менее трёх) и ссылки на данные об актёрах и режиссёрах фильмов. В свою очередь, актерские и режиссерские страницы содержат ссылки на аннотации к фильмам.

Для выполнения задания 2, как и в предыдущем задании, необходимо запустить программу PowerPoint и сохранить файл. Выбрать шаблон (как описано в инструкции выше). Создадим презентацию, руководствуясь рекомендациями, изложенными в предыдущем задании и сведениями о фильмах, изложенными в описании выполнения задания на создание гипертекстового документа (см. выше раздел 1 второй части пособия, задание 2).

Для создания гиперссылки необходимо выделить текстовый фрагмент, нажать «Вставка», «Гиперссылка», «Связать с местом в документе», выбрать необходимый слайд, на который нужно перейти. Желательно на слайде, на который осуществляется переход, создать аналогичную гиперссылку, предусматривающую возможность возвращения обратно. Если при показе презентация работает некорректно, необходимо, выделив гиперссылку, проверить настройки в «Показ слайдов», «Настройка действия». Помимо этого, желательно установить «Показ слайдов», «Настройка презентации», «Показ слайдов, управляемый докладчиком».

5. Создание базы данных. Определение структуры базы данных: количество и типы полей, заполнение таблиц (или использование готовых). Организация поиска информации в базах данных. Создание запросов разной сложности.

Задание 1*. (Состоит из 8 пунктов).

1. Создать структуру таблицы базы данных «Ученики», содержащую следующие поля: фамилия, имя, класс, адрес, дата рождения, вес.
2. Определить первичный ключ таблицы.
3. В режиме таблицы ввести в базу данных 10 записей об учениках вашей школы (значения полей можно задавать произвольно).
4. Добавить в структуру таблицы после поля «Дата рождения» поле «Рост».
5. Удалить из структуры поле «Вес».
6. Заполнить в таблице поле «Рост» (произвольно).
7. Вывести на экран поля «Фамилии», «Имя», «Класс» для учеников, рост которых выше 175 см (использовать запрос), отсортировав их в алфавитном порядке фамилий.
- 8***. Удалить из таблицы сведения об учениках с именем «Владимир».

Выполнение задания необходимо осуществлять в программе MS Access. Откроем программу, выберем в меню «Файл» «Создать», «Новую базу данных», сохраним её как «*pupils.mdb*» и перейдем к редактированию (основные понятия о табличных базах данных изложены в ответе на теоретическую часть билета 17 — см. Часть I пособия).

В открытом окне «Pupils: база данных» выберем объект «Таблицы», а далее «Создание таблицы в режиме конструктора». Введем названия полей: фамилия, имя, класс, адрес, дата рождения, вес. Для каждого поля установим тип вводимых данных, выбрав из предложенного списка. Соответствие полей и их типов приведено в таблице.

№	Поле	Тип
1	Фамилия	Текстовый
2	Имя	Текстовый
3	Класс	Числовое
4	Адрес	Текстовый
5	Дата рождения	Дата
6	Вес	Числовое

Закроем конструктор, откроем таблицу базы данных и заполним её.

Таблица1 : таблица							
Код	Фамилия	Имя	Класс	Адрес	Дата рождения	Вес	
1	Иванов	Иван	5		12.06.1997	46	
2	Петров	Владимир	10		14.07.1992	50	
3	Сидоров	Константин	10		16.01.1992	51	
4	Васильев	Владимир	11		15.10.1991	60	
5	Давыдов	Ян	11		08.12.1991	63	
6	Леночкин	Петр	9		07.07.1993	55	
7	Ильин	Владимир	8		01.04.1993	55	
8	Сергеев	Александр	7		19.10.1995	50	
9	Михайлов	Олег	1		05.05.2001	30	
10	Савельев	Илья	10		06.11.1992	170	
★	(Счетчик)		0			0	

Запись 10 из 10

Ключом (т.е. совокупностью полей, однозначно характеризующей запись), являются три поля: «Фамилия», «Имя» и «Дата рождения». Для этих полей, переведя таблицу в режим конструктора (щелчком правой кнопки мыши на заголовке окна таблицы открываем список, в котором выбираем «Конструктор»), установим маркер «Обязательное поле». Помимо этого программа автоматически создала ключевое поле «Код», являющееся аналогом нашего составного ключа (все значения в этом поле уникальны, контроль за уникальностью данного ключа осуществляет сама система управления базой данных Access).

В режиме конструктора добавим в самый конец поле «Рост» числового типа. Для перемещения поля «Рост» установим на него указатель и, нажав левую кнопку, мыши «потянем» его вверх на место, следующее за полем «Дата рождения». Наведя курсор на поле «Вес», нажмем правую кнопку мыши, а далее «Удалить строки». Закрыв режим «Конструктор», введем в таблицу данные о росте учащихся.

Таблица1 : таблица							
Код	Фамилия	Имя	Класс	Адрес	Дата рождения	Рост	
1	Иванов	Иван	5		12.06.1997	130	
2	Петров	Владимир	10		14.07.1992	170	
3	Сидоров	Константин	10		16.01.1992	160	
4	Васильев	Владимир	11		15.10.1991	167	
5	Давыдов	Ян	11		08.12.1991	180	
6	Леночкин	Петр	9		07.07.1993	160	
7	Ильин	Владимир	8		01.04.1993	160	
8	Сергеев	Александр	7		19.10.1995	155	
9	Михайлов	Олег	1		05.05.2001	120	
10	Савельев	Илья	10		06.11.1992	182	
★	(Счетчик)		0			0	

Запись 10 из 10

Закрыв таблицу, выберем объект «Запросы», «Создание запроса в режиме конструктора». Выберем в качестве источника данных созданную таблицу со сведениями об учениках (кнопка «Доба-

вить)). В первой строке в четырех первых столбцах конструктора запросов выберем соответственно поля «Фамилия», «Имя», «Класс», «Рост». В колонке «Рост» снимем отметку о выводе столбца на экран. В столбце «Фамилия» установим сортировку по возрастанию, а в столбце «Рост» обозначим «Условие отбора»: «>175». Закроем «Конструктор» и откроем созданный запрос. Результат его работы будет следующий:

Запрос1 : запрос на выборку

Фамилия	Имя	Класс
Давыдов	Ян	11
Савельев	Илья	10
		0

Запись: 1 из 3

Удалим из таблицы сведения об учениках с именем Владимир. Для этого в режиме конструктора создадим новый запрос (как и в предыдущем случае, источник данных та же таблица). На панели инструментов выберем «Тип запроса», «Удаление». В строке «Поле» в первой колонке укажем имя всей таблицы, во втором столбце название поля («Имя»), значение которого играет роль при отборе записей. Во втором столбце укажем условие отбора «=Владимир». В строке «Удаление» в первом столбце возникнет «Из», во втором «Условие». Закроем «Конструктор», выполним запрос, подтвердив наше желание удалить записи. Результат удаления записей из таблицы таков:

Таблица1 : таблица

Код	Фамилия	Имя	Класс	Адрес	Дата рождения	Рост
1	Иванов	Иван	5		12.06.1997	130
3	Сидоров	Константин	10		16.01.1992	160
5	Давыдов	Ян	11		08.12.1991	180
6	Леночкин	Петр	9		07.07.1993	160
8	Сергеев	Александр	7		19.10.1995	155
9	Михайлов	Олег	1		05.05.2001	120
10	Савельев	Илья	10		06.11.1992	182
*(Счетчик)			0			0

Запись: 1 из 7

Задание 2.** (Состоит из 7 пунктов.)

1. Открыть базу данных SPORT.
2. Изменить размер поля «Страна» на 20 мм.
3. Сформировать запрос для вывода на экран фамилий и страны для спортсменов из США.

4**. Сформировать запрос для вывода на экран номера, фамилий, оценок за кольца и перекладину для спортсменов, получивших на кольцах балл не ниже 9.500.

5**. Сформировать запрос для удаления всех спортсменов, получивших за кольца оценку ниже 9.000.

6***. Сформировать запрос, с помощью которого у всех спортсменов из России название страны будет изменено на название «Российская Федерация».

7***. Применить все созданные запросы и проверить правильность их работы.

Для выполнения задания необходимо запустить MS Access, открыть базу данных SPORT (расширение файла базы данных *mbf*). Открытая база данных может иметь типовой вид, приведенный на рисунке.

Номер	Фамилия	Имя	Страна	Переключатель	Кольца	Конь
1	Робсон	Джон	США	9,225	9	9,875
2	Леонидов	Сергей	Россия	9,5	9,225	9,875
3	Чугайнов	Андрей	Россия	9,225	9,775	9,925
4	Пьолин	Франсуа	Франция	8,5	8,9	8,225
5	Штольц	Андреас	Германия	9,775	9,225	9,1
6	Морозов	Олег	Украина	9,885	9,5	9,625
7	Джонсон	Фрэнк	США	8,885	9	9,325
8	Ли	Грег	США	9,5	9,5	9,225
9	Овсиенко	Геннадий	Украина	9,975	9	9,225
10	Кузартен	Пьер	Франция	9,925	8,775	9,5
Ж	(Счетчик)			0	0	0

Запись 1 из 10

Для изменения размера поля «Страна» необходимо открыть таблицу в режиме «Конструктор», Выделить поле «Страна», изменить значение открывшегося атрибута поля «Размер» на 20.

Для вывода на экран фамилий и страны для спортсменов из США выберем объект «Запросы», «Создание запроса в режиме конструктора». Выберем в качестве источника данных таблицу со сведениями о спортсменах (кнопка «Добавить»). В первой строке в двух первых столбцах конструктора запросов выберем соответственно поля «Фамилия», «Страна». В обеих колонках должен быть установлен флажок «Вывод на экран». В столбце «Страна» обозначим «Условие отбора»: «=США». Результат работы запроса приведен на рисунке.

Фамилия	Страна
Робсон	США
Джонсон	США
Ли	США

Запись 4 из 4

Таблица1 : таблица						
Номер	Фамилия	Имя	Страна	Перекладина	Кольца	Конь
1	Робсон	Джон	США	9,225	9	9,875
2	Леонидов	Сергей	Российская Федерация	9,5	9,225	9,875
3	Чугайнов	Андрей	Российская Федерация	9,225	9,775	9,925
5	Штольц	Андреас	Германия	9,775	9,225	9,1
6	Морозов	Олег	Украина	9,885	9,5	9,625
7	Джонсон	Фрэнк	США	8,885	9	9,325
8	Ли	Грег	США	9,5	9,5	9,225
9	Овсиенко	Геннадий	Украина	9,975	9	9,225
※ (Счетчик)				0	0	0
Запись: 3 из 8						

Задание 3***. (Состоит из 5 пунктов.)

1. Открыть базу данных SPORT.
2. Сформировать запрос для вывода на экран номера, фамилии, оценок за кольца и перекладину для спортсменов, получивших на кольцах балл не ниже 9.000, а на перекладине — не ниже 9.200.
3. Сформировать запрос для вывода на экран фамилии и названия страны для спортсменов из США и России.
4. Сформировать запрос для вывода на экран фамилии и названия страны для всех спортсменов, кроме спортсменов из России и Украины.
5. Сформировать запрос для вывода на экран фамилии, названия страны и суммы баллов для всех спортсменов, набравших в сумме более 28 баллов.

Для выполнения задания, как и раньше, запустим MS Access, откроем базу данных SPORT (см. предыдущее задание).

Формировать запросы будем теми же способами, что и в заданиях 1 и 2. Поэтому кратко опишем выводимые запросом поля, условия отбора данных, типы запросов.

Для формирования запроса для вывода на экран номера, фамилии, оценок за кольца и перекладину для спортсменов, получивших на кольцах балл не ниже 9.000, а на перекладине — не ниже 9.200, в «Конструкторе» запросов выберем поля «Номер», «Фамилия», «Кольца», «Перекладина». Условия отбора следующие: для поля «Кольца» «>=9,000», для поля «Перекладина» «>=9,2». Запрос выведет следующую таблицу:

Запрос1 : запрос на выборку				
Номер	Фамилия	Кольца	Перекладина	
1	Робсон	9	9,225	
2	Леонидов	9,225	9,5	
3	Чугайнов	9,775	9,225	
5	Штольц	9,225	9,775	
6	Морозов	9,5	9,885	
8	Ли	9,5	9,5	
9	Овсиенко	9	9,975	
※ (Счетчик)		0	0	
Запись: 1 из 7				

Для формирования запроса для вывода на экран фамилии и названия страны для спортсменов из США и России в «Конструкторе» запросов выберем поля «Фамилия», «Страна». Условие отбора следующее: для поля «Страна» («(Россия) Or («США»)»). Получим:

Запрос2 : запрос «...

Фамилия	Страна
Робсон	США
Леонидов	Россия
Чугайнов	Россия
Джонсон	США
Ли	США

Запись: 1 6 из

Осуществим выборку спортсменов, выступающих не за Россию и не за Украину. Для этого в «Конструкторе» укажем поля «Фамилия» и «Страна». Условие отбора для поля «Страна» имеет следующий вид: «<> («Россия») And <> («Украина»)». В результате имеем:

Запрос3 : запрос ...

Фамилия	Страна
Робсон	США
Штольц	Германия
Джонсон	США
Ли	США
Пьолин	Франция
Кузртен	Франция

Запись: 1 7 из

Запрос для вывода на экран фамилии, названия страны и суммы баллов для всех спортсменов, набравших в сумме более 28 баллов, сформируем следующим образом. В «Конструкторе» запросов выберем поля «Фамилия», «Страна», в первую строку третьей колонки введем «Конь+Кольца+Перекладина». Условие отбора для третьей колонки следующее: «>28»; флажок вывода на экран третьей колонки включен. Получим:

Запрос4 : запрос на выборку

Фамилия	Страна	Выражение 1
Робсон	США	28,1
Леонидов	Россия	28,6
Чугайнов	Россия	28,925
Штольц	Германия	28,1
Морозов	Украина	29,01
Ли	США	28,225
Овсиенко	Украина	28,2
Кузртен	Франция	28,2

Запись: 1 9 из 9

6. Работа с электронной таблицей. Создание таблицы в соответствии с условием задачи, использование функций. Построение диаграмм и графиков по табличным данным.

Задание 1*. В банк внесен вклад размером N под $P\%$ годовых. Определить ежегодный прирост вклада и величину вклада через K лет.

Пусть для определённости $N=1000$, $P=10\%$, $K=12$. В конце каждого года сумма на счету увеличивается в $(1+P/100)=(1+0,1)=1,1$ раза. Окончательная сумма S_K на счету через K лет может быть найдена по формуле сложных процентов: $S_K=N*(1+P/100)^K=S_{K-1}*(1+P/100)$.

В программе Excel составим следующую таблицу (см. рис.). Для заполнения ячеек подводим курсор к нужной ячейке, нажимаем левую кнопку мыши (ячейка выделяется жирной рамкой), вводим нужный текст или цифры и нажимаем «Ввод».

	A	B	C	D	E	F	G	H
1	Год	Сумма						
2	0	1000						
3	1			P				
4	2			10				
5	3							
6	4							
7	5							
8	6							
9	7							
10	8							
11	9							
12	10							
13	11							
14	12							
15								
16								

В ячейке D4 содержится значение P (текущее значение равно десяти). Ячейка B2 содержит начальную сумму вклада (1000). Выбираем строку B3, нажимаем знак « $=$ » и далее набираем « $B2*(1+\$D\$4/100)$ » и «Ввод».

Задание 2.** Для проведения эксперимента выбраны 10 районов Московской области. Известны засеваемые площади и средняя урожайность по району.

Название района	Площадь посева (га)	Средняя урожайность (т/га)
Дмитровский	93	12
Домодевский	65	17
Клинский	98	15
Лыткаринский	64	17
Люберецкий	102	17
Марьинский	155	14
Мытищинский	207	16
Орловский	307	19
Реутовский	134	21
Семеновский	45	14

Определить количество урожая, собранного в каждом районе и в целом по экспериментальным районам области, а также среднюю урожайность по всем районам. Выделить районы с самой низкой и самой высокой урожайностью.

Для выполнения работы заполним таблицу в программе Excel. В столбец D будем выводить данные по урожаю конкретно в каждом районе. Формула подсчета урожая такова: урожай = площадь посева*средняя урожайность, т.е. перемножаются данные из столбцов B и C. В ячейку D2 введем «=B2*C2». Затем необходимо выбрать ячейку D2 и подвести курсор к правому нижнему углу рамки. Курсор примет вид крестика.

	A	B	C	D
	Название района	Площадь посева (га)	Средняя урожайность (т/га)	Урожай в районе
1				
2	Дмитровский	93	12	1116
3	Домодевский	65	17	1105
4	Клинский	98	15	1470
5	Лыткаринский	64	17	1088
6	Люберецкий	102	17	1734
7	Марьинский	155	14	2170
8	Мытищинский	207	16	3312
9	Орловский	307	19	5833
10	Реутовский	134	21	2814
11	Семеновский	45	14	630
12		307		2127,2
13		45		2127,2

Готово: NUM

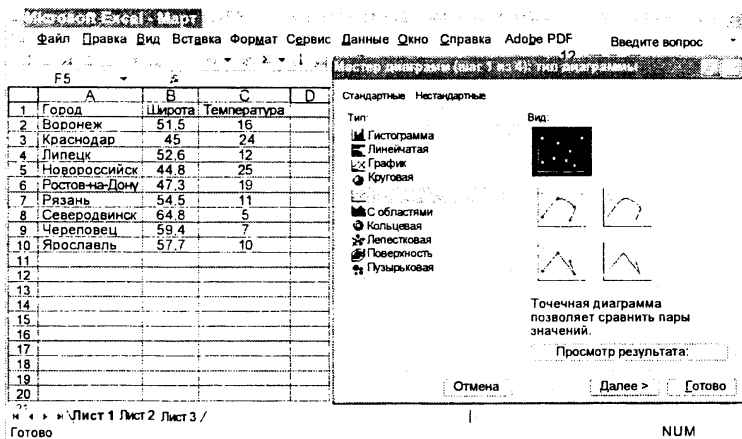
Затем следует нажать левую кнопку мыши и, не отпуская её, переместить курсор вниз до ячейки D11, а затем отпустить кнопку мыши. В ячейки D3–D11 будет скопирована формула из D2 (поскольку ссылка в формуле является относительной, то значения в ячейках будут вычисляться по формулам «=B3*C3», «=B4*C4», ...). В результате колонка D заполнится значениями, соответствующими урожаю по районам. В ячейку D12 выведем сумму чисел, стоящих в ячейках с D2 по D11 (т.е. найдем урожай в целом по указанным районам). Для этого выделим ячейку D12, нажмем на панели инструментов **Стандартные** кнопку **Автосумма** , а затем нажмем клавишу ВВОД (либо наберем в ячейке D12 «=СУММ(D2:D11)», указав через двоеточие диапазон ячеек, данные в которых необходимо просуммировать). Среднее значение урожайности по всем районам выведем в ячейку D13, задав её значение формулой «=СРЗНАЧ(D2:D11)».

Выделим районы с самой низкой и самой высокой урожайностью. Для этого в ячейки B12 (для поиска максимального значения) и B13 (для поиска минимального значения) введем соответственно формулы «=МАКС(B2:B11)», «=МИН(B2:B11)». После чего легко видеть, что самая низкая урожайность в Семеновском районе, самая высокая — в Орловском. Если таблица длинная и визуальный поиск осложнен, то можно воспользоваться в меню «**Правка**» командой «**Поиск**», предварительно выделив соответствующий столбец и задав уже известное число для поиска. Вторым вариантом решения данной проблемы может служить использования команды «**Сортировка**» из меню «**Данные**».

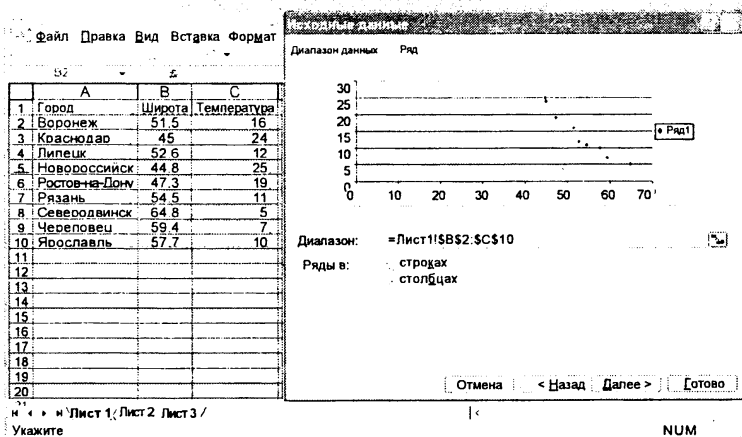
Задание 3*.** Построить график зависимости средней дневной температуры за последнюю неделю марта в различных городах европейской части России от географических широт этих городов.

Город	Широта	Температура
Воронеж	51,5	16
Краснодар	45	24
Липецк	52,6	12
Новороссийск	44,8	25
Ростов-на-Дону	47,3	19
Рязань	54,5	11
Северодвинск	64,8	5
Череповец	59,4	7
Ярославль	57,7	10

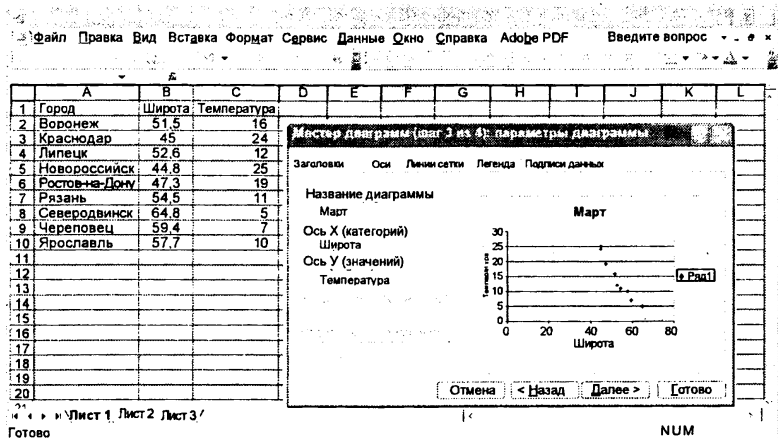
Заполнить таблицу в программе Excel. В меню выбрать «Вставка», «Диаграмма». Появится окно мастера диаграмм. В этом окне выбрать «Точечная», нужный вид и нажать «Далее».



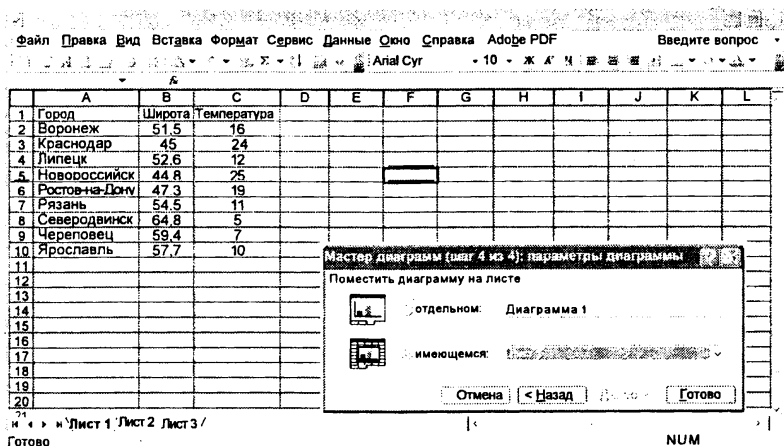
Появится окно «Исходные данные». В таблице выделить при помощи мыши данные для построения и нажать «Далее»:



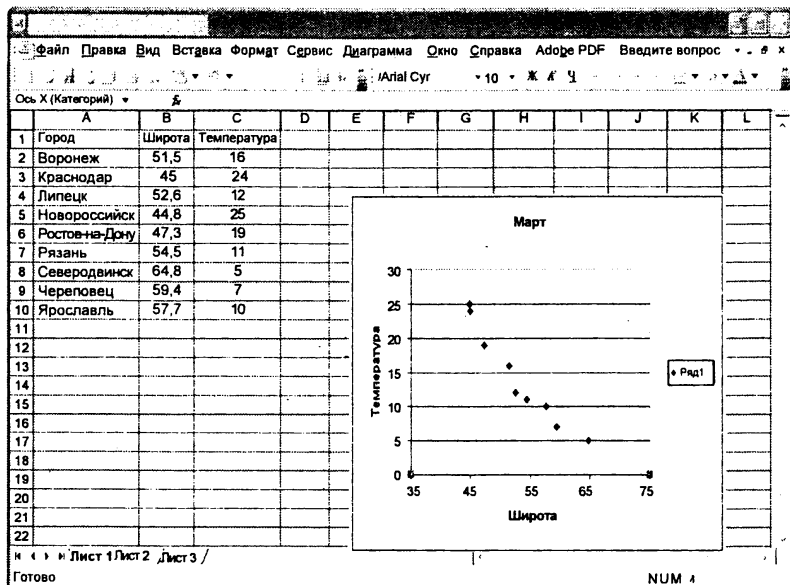
В соответствующих полях ввести название диаграммы и названия осей, затем нажать «Далее»



В появившемся поле выбрать место размещения диаграммы и нажать «Готово».



При необходимости можно перетащить диаграммы в нужное место таблицы и изменить размеры. Для удобства восприятия можно изменить диапазон значений, выводимых по оси x (щелкнуть по оси, выбрать закладку «Шкала», изменить минимальное значение по оси x с нулевого, например, на 35). Итоговый график будет иметь следующий вид.



7. Поиск информации в Интернете с применением языка запросов.

Задание 1*. Продемонстрировать поисковые возможности на примере школьного сайта. Поиск заданной страницы по критерию.

Например — кто стал победителем школьной олимпиады по информатике в прошлом году?

Поиск информации в Интернете во многих случаях осуществляется при помощи запросов поисковым системам (например, Yandex или Rambler), формирование которых осуществляется по специальным правилам. Для осуществления поиска необходимо указать ключевое слово или комбинацию ключевых слов. Основные правила построения запроса следующие:

- При поиске автоматически учитываются все формы слова.
- Если ищется фиксированная комбинация слов, то её необходимо заключить в кавычки: «Война и мир».
- При необходимости найти определённую форму слова перед ним ставится восклицательный знак: «!преподавателем».
- Набранные в запросе через пробел слова означают желание пользователя получить Интернет-страницы, где все эти слова

входят в одно предложение. Другим разделителем, выполняющим ту же функцию, является знак &.

- Синонимы в запросе разделяются вертикальной чертой (знак |).
- Использование в запросе знака тильда (~) между словами означает, что ведется поиск предложений, содержащих первое, но не содержащих второе слово.
- Знак «+» перед словом означает, что оно должно содержаться в документе, который пытаются найти, знак «-» — наоборот. Оба этих знака пишутся через пробел с предыдущим, но слитно с последующим словом.

Согласно выше приведенным правилам запрос, осуществляющий поиск информации о победителе школьной олимпиады по информатике в прошлом (2008) году на сайте школы (лицея, гимназии) №4 города Вологда, может выглядеть следующим образом:

«+(олимпиада)+(информатика)+(школа&4)+(Вологда)+(2008)+(победитель)»

Задание 2.** Найти ответы на вопросы, используя поисковый сервер Rambler (<http://www.rambler.ru>) или Yandex (<http://www.yandex.ru>). Указать адрес источника информации.

1) Где и когда проводилась последняя Всероссийская олимпиада по информатике? Кто стал победителем олимпиады?

2) Где и когда проводилась последняя международная олимпиада по информатике? Каков состав российской команды и её результат?

1) Запрос «"Всероссийская олимпиада по информатике"» дает ответ на вопрос где и когда была проведена Всероссийская олимпиада по информатике.

Запрос «("всероссийская олимпиада по информатике")+(победитель)+(2008)» (указан год, за который нас интересуют результаты) поможет ответить на вопрос о победителе этой олимпиады.

2) Задание может быть решено при помощи запросов

«"международная олимпиада по информатике"» и

«("международная олимпиада по информатике")+(2008)+(состав&команда)+(Россия)».

8. Построение алгоритма (основные алгоритмические структуры) и его реализация в среде учебного исполнителя. Демонстрация полученного алгоритма в среде учебного исполнителя.

Описание исполнителя

Имеется исполнитель ЧЕРЕПАШКА, которая «живет» на координатной плоскости (200×100). ЧЕРЕПАШКА может двигаться по прямой, оставляя за собой след, делать повороты на n градусов по часовой стрелке. В начальном положении она находится в левом нижнем углу и «смотрит» вверх.

Система команд исполнителя ЧЕРЕПАШКА:

Вперед N — движение вперед на N единиц.

Поворот N — поворот на N градусов по часовой стрелке.

Повторить <число> раз <команды, которые необходимо повторить>
Кц

Проц <заголовок вспомогательного алгоритма>
<тело вспомогательного алгоритма>

Кн Проц

Задача 1. Нарисовать прямоугольную рамку вдоль края экрана (поля рисунка), используя вспомогательный алгоритм (процедуру).

Для создания прямоугольной рамки по краю экрана ЧЕРЕПАШКЕ необходимо обойти все четыре стороны прямоугольника. Для этого создается процедура, отвечающая за обход двух последовательных сторон. В тексте программы выполнение этой процедуры осуществляется два раза.

Проц Две_стороны _пройдены

Вперед 100;

Поворот 90;

Вперед 200;

Поворот 90;

Кн Проц;

Повторить 2 раз Две_стороны_пройдены;

Кц.

Задача 2. Расчертить весь экран (поле рисунка) горизонтальными линиями на расстоянии одного шага друг от друга, используя вспомогательные алгоритмы (процедуры).

Для решения данной задачи ЧЕРЕПАШКЕ необходимо начать совершать следующие действия:

- 1) сдвинуться на одну единицу вдоль боковой рамки экрана, повернуться,
- 2) пройти по горизонтальной прямой из конца в конец экрана (200 шагов), опять повернуться,
- 3) сдвинуться вверх на одну единицу, осуществить ещё один поворот,

- 4) ещё раз пересечь экран из конца в конец, повернуться,
5) ...

Поскольку «высота» поля, на котором «живет» ЧЕРЕПАШКА равна 100, то для того, чтобы расчертить весь экран необходимо выписанную последовательность действий повторить в цикле 50 раз.

Проц Туда_обратно

Вперед 1;

Поворот 90;

Вперед 200;

Поворот 270;

Вперед 1;

Поворот 270;

Вперед 200;

Поворот 90;

Кн Проц;

Повторить 50 раз Туда_обратно;

Кц.

Задача 3. КЕНГУРУ стоит на нулевом делении. Все деления не закрашены. Она должна закрасить деления, как показано на рисунке (закрашенные деления обозначены жирной линией) и закончить движение в точке 12.

Написать программу, по которой Кенгуру выполнит эту задачу, используя следующий вспомогательный алгоритм:



Проц ШАГИ Вперед 3; Назад 2; Кн Проц;

Описание исполнителя

Исполнитель КЕНГУРУ, «живет» на числовой оси, на которой проставлены деления, отмечающие позиции целых чисел (см. рисунок). КЕНГУРУ может прыгать по делениям и закрашивать их.

Система команд исполнителя КЕНГУРУ:

Вперед N — прыгает вдоль числовой оси вперед на N единиц.

Назад N — прыгает вдоль числовой оси назад на N единиц.

Закрась — закрасить текущую точку на числовой прямой.

Повторить <число> раз <команды, которые необходимо повторить>

Кц

Проц <заголовок вспомогательного алгоритма>

<тело вспомогательного алгоритма>

Кн Проц

Заметим, что все деления, которые закрашиваются, выбираются строго определённо: остаются не закрашенными деления, номера кото-

рых кратны трем. Такое повторение рисунка (через каждые три деления) приводит к необходимости применения циклов к решению задачи.

Описанная в задании процедура осуществляет следующие действия: сдвиг на 3 единицы вперед, а затем на 2 единицы назад. Если КЕНГУРУ стартует из начального положения на нулевом делении, то выполнение данной процедуры приведет кенгуру на первое деление, которое необходимо закрасить. Далее, закрасив текущее деление и ещё одно следующее за ним, КЕНГУРУ может сдвинуться на одно деление вперед, после чего она будет находиться в тех же условиях, что и до начала прыжков. Таким образом, для перехода КЕНГУРУ с нулевого деления на двенадцатое необходимо полностью повторить описанные действия 4 раза.

Программа действий для исполнителя КЕНГУРУ имеет следующий вид

Проц ШАГИ Вперед 3; Назад 2; Кн Проц;

Повторить 4 раз

ШАГИ;

Повторить 2 раз

Закрась;

Вперед 1;

Кц;

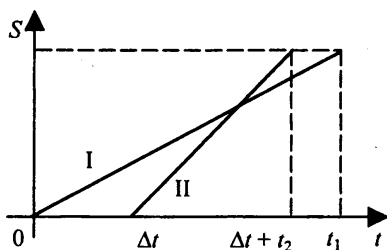
Кц.

9. Построение алгоритма для обработки величин с реализацией на языке программирования (ветвление, цикл, линейный массив или вспомогательные алгоритмы). Отладка программы и получение результатов.

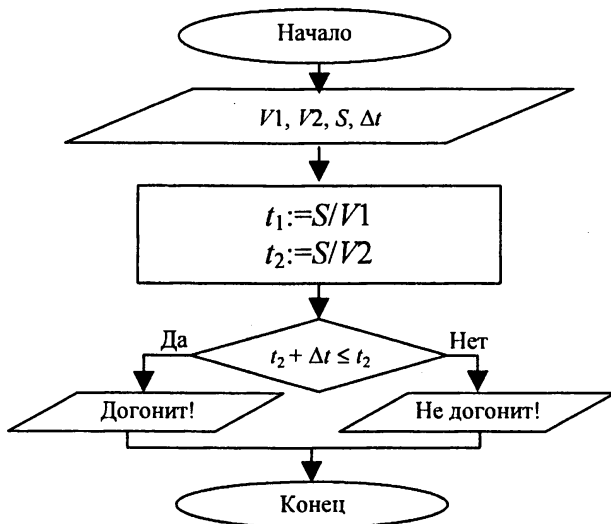
Задача 1*. **Равномерное движение.** Из пункта А в пункт Б движется велосипедист с постоянной скоростью V км/ч. Через 1,5 часа после его старта из пункта А стартовал второй велосипедист, который двигался с постоянной скоростью 11 км/ч. Построить алгоритм (нарисовать блок-схему) и составить программу на языке программирования для определения: догонит ли второй велосипедист первого во время своего движения при данных значениях расстояния между пунктами А и Б и скоростей V . Проверить программу на тестах при разных вариантах значений исходных данных (скоростей велосипедистов, времени между их стартами и расстояния между пунктами А и Б). Считать, что если оба велосипедиста приехали в пункт Б одновременно, то второй догнал первого.

Из физики известно, что равномерное движение описывается формулой $S=V \cdot t$, где S — расстояние пройденной телом, V — его скорость, t — прошедшее с момента начала движения время. Тогда, если известно расстояние, то время, за которое тело может его преодолеть, может быть найдено как $t=S/V$.

График, описывающий движение в задаче 1, изображен на рисунке. Из него легко видеть, что второй велосипедист догонит первого, если сумма его времени в пути (t_2) и времени задержки его выезда ($\Delta t = 1.5$) не превзойдет времени в пути (t_1) для первого велосипедиста: $t_1 + t_2 \leq t_1$.



Блок-схема для данной задачи имеет следующий вид.



Program motion;

{Программа, позволяющая ответить на вопрос: «Догонит ли первый велосипедист второго?»; начало раздела описаний переменных и констант}

Var V1, V2, delta_t, S : real; {описываются четыре переменные, имеющие соответственно смысл скоростей первого и второго велосипедистов, разницы во времени между их стартами и расстояния между пунктами А и Б}

t1, t2 : real; {описываются две переменные, имеющие соответственно смысл времени в пути для первого и второго велосипедистов}

Begin {начало раздела операторов}

Writeln('Введите скорость первого велосипедиста, км/ч ');
{Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры значения V1 }

```

Readln(V1); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}
Writeln('Введите скорость второго велосипедиста, км/ч ');
    {На экран выводится просьба о вводе с клавиатуры значения V2 }
Readln(V2); {Оператор ввода данных}
Writeln('Введите разницу во времени между стартом первого и второго, ч '); {На экран выводится просьба о вводе с клавиатуры значения delta_t}
Readln(delta_t); {Оператор ввода данных}
Writeln('Введите расстояние между пунктами А и Б, км ');
    {На экран выводится просьба о вводе с клавиатуры значения S}
Readln(S); {Оператор ввода данных }
t1:=S/V1; {вычисляется время в пути для первого велосипедиста}
t2:=S/V2; {вычисляется время в пути для второго велосипедиста}
if (t2+delta_t<=t1) then
Writeln('Второй догонит первого! ')
else
Writeln('Второй не догонит первого! '); {Происходит проверка неравенства  $t_2 + \Delta t \leq t_1$ , результат проверки выводится на экран}

End.

```

Аналогичная программа на языке Basic имеет следующий вид.

```

REM Движение
INPUT "Введите скорость первого велосипедиста, км/ч "; V1
INPUT "Введите скорость второго велосипедиста, км/ч "; V2
INPUT "Введите разницу во времени между стартом первого и второго, ч"; delta_t
INPUT ""Введите расстояние между пунктами А и Б, км "; S
t1=S / V1
t2=S / V2
IF t2+delta_t<=t1 THEN
PRINT "Второй догонит первого! "
ELSE
PRINT "Второй не догонит первого! "
END IF
END

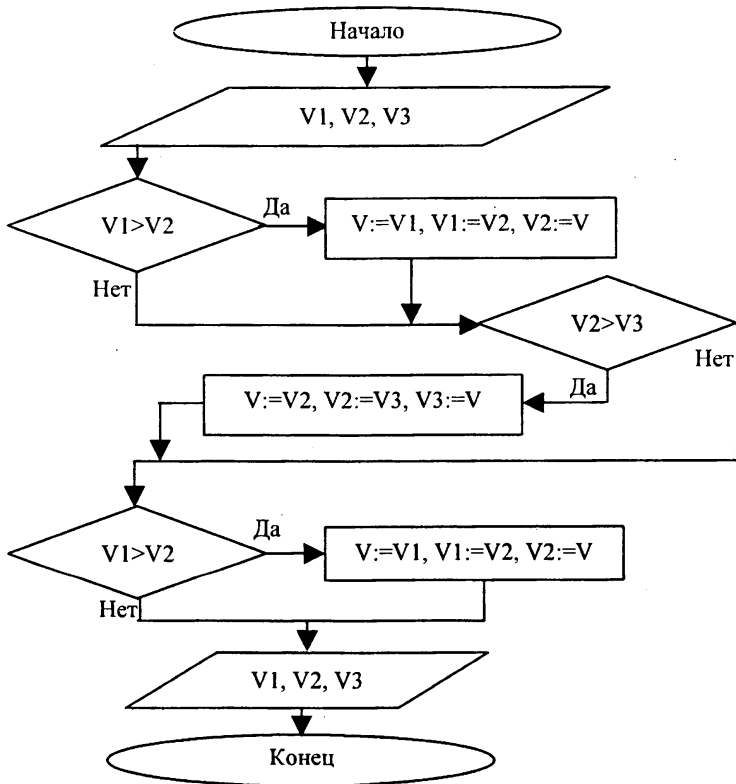
```

Результат работы программы при различных начальных данных приведен в таблице.

	V1, км/ч	V2, км/ч	S, км	Δt , ч	Ответ
1	11	11	22	1.5	Не догонит
2	12	11	22	1.5	Не догонит
3	9	11	22	1.5	Не догонит
4	6	11	22	1.5	Догонит

Задача 2.** Сортировка чисел. Построить алгоритм (нарисовать блок-схему) и составить программу на языке программирования, осуществляющую сортировку значений трёх переменных A, B, C по возрастанию. Проверить программу на тестах при разных вариантах значений исходных данных.

Блок-схема для данной задачи имеет следующий вид.



Program sort;

{Программа, сортирующая три действительных числа по возрастанию; начало раздела описаний переменных и констант}

Var V1, V2, V3, V: **real**; {описываются три переменные, имеющие смысл значений трёх сортируемых чисел и одна вспомогательная}

Begin {начало раздела операторов}

Writeln('Введите первое число '); {Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры значения V1 }

Readln(V1); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

Writeln('Введите второе число '); {На экран выводится просьба о вводе с клавиатуры значения V2 }

Readln(V2); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

Writeln('Введите третье число '); {На экран выводится просьба о вводе с клавиатуры значения V3}

Readln(V3); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

If (V1>V2) **then** {Сравниваем первое и второе числа, если второе меньше, то меняем их местами, соответственно меняются и их номера}

Begin

V:=V1;

V1:=V2;

V2:=V;

End;

if (V2>V3) **then** {Сравниваем третье и второе числа, если третье меньше, то меняем их местами}

begin

V:=V2;

V2:=V3;

V3:=V;

end;

If (V1>V2) **then**{Сравниваем первое и второе числа, если второе меньше, то меняем их местами}

```

Begin
    V:=V1;
    V1:=V2;
    V2:=V;
End;
Writeln("Числа отсортированы по возрастанию: ", V1, V2, V3);
End.

```

Аналогичная программа на языке Basic имеет следующий вид.

```

REM сортировка
INPUT "Введите первое число "; V1
INPUT "Введите второе число "; V2
INPUT "Введите третье число "; V3
IF V1>V2 THEN
    V=V1
    V1=V2
    V2=V
END IF
IF V2>V3 THEN
    V=V2
    V2=V3
    V3=V
END IF
IF V1>V2 THEN
    V=V1
    V1=V2
    V2=V
END IF
PRINT "Числа отсортированы по возрастанию: "; V1; V2; V3
END

```

Результат работы программы при различных начальных данных приведен в таблице.

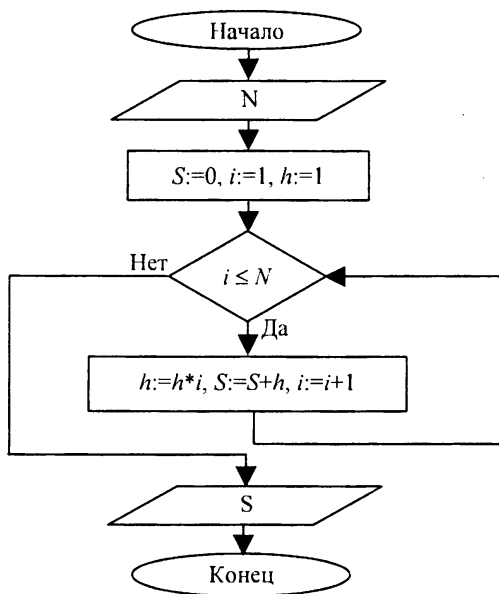
	V1	V2	V3	Ответ
1	1	2	3	1 2 3
2	3	2	1	1 2 3
3	10	8	12	8 10 12
4	12	8	10	8 10 12

Задача 3**. Ветвление и цикл

1. Дано натуральное число N . Вычислить: $S = 1! + 2! + 3! + \dots + N!$
- 2.***** Перевести заданное натуральное число из десятичной системы счисления в двоичную.
- 3.**** Перевести заданное число из двоичной системы счисления в десятичную.

1. Обозначение $n!$ (читается «эн факториал») представляет собой краткую математическую запись произведения всех натуральных чисел от единицы до n , т.е. $n! = 1 * 2 * 3 * \dots * (n-1) * n$. При реализации алгоритма нам также потребуется очевидная формула $n! = (n-1)! * n$

Блок-схема для данной задачи имеет следующий вид.



Program summing_up;

{Программа, подсчитывающая значение выражения $S = 1! + 2! + 3! + \dots + N!$; начало раздела описаний переменных и констант}

Var N, S : integer; {описываются две целочисленные переменные, имеющие смысл значения самого натурального числа N и суммы S соответственно }

i : integer; {целочисленная переменная, счетчик циклов}

h : integer; {вспомогательная переменная, имеющая смысл произведения чисел}

Begin {начало раздела операторов}

Writeln('Введите натуральное число N '); {Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры значения N }

Readln(N); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

S:=0; {Обнуляется начальное значение суммы}

h:=1; {Начальное значение переменной, значение которой на каждом шаге в цикле равно факториалу счетчика цикла i }

for i:=1 to N **do** {Организован цикл для подсчета суммы}

begin

h:=h*i; {Домножаем $(n-1)!$ на n , чтобы получить $n!$ }

S:=S+h; {К старому значению суммы прибавляется вновь подсчитанный факториал}

end;

Writeln('Сумма S равна : ', S);

End.

Аналогичная программа на языке Basic имеет следующий вид.

REM суммирование факториалов

INPUT "Введите натуральное число N "; N

S=0

h=1

FOR i=1 **TO** N

h=h*i

S=S+h

NEXT i

PRINT "Сумма S равна: "; S

END

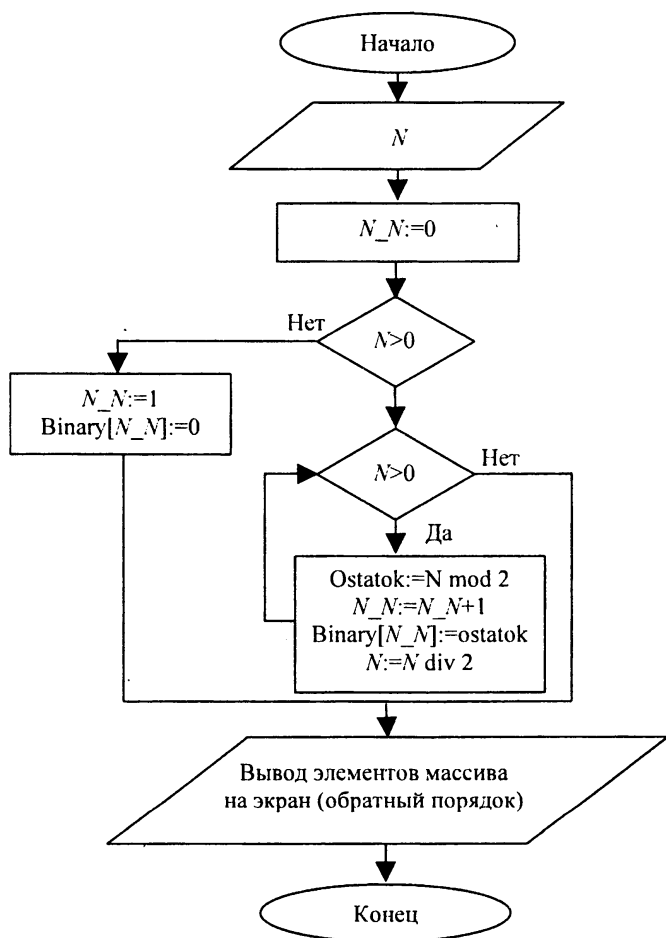
Заметим, что реализация данной программы возможна также с использованием вложенных циклов (внутри цикла со счетчиком i , «отвечающего» за суммирование, возможно организовать цикл каждый раз заново подсчитывающий значение произведения, т.е. факториала), результат работы программы в этом случае будет тот же, но возрастет количество шагов ведущих к результату. При желании школьника организовать работу программы именно таким образом более целесообразно блок команд, отвечающих за подсчет значения факториала, оформить в виде отдельной функции.

Результат работы программы при различных начальных данных приведен в таблице.

	N	Ответ
1	1	1
2	2	3
3	3	9
4	5	153

2. Для реализации данной программы на языке программирования Pascal будем считать, что нам задано целое десятичное число N . При переводе этого числа в двоичную систему счисления (алгоритм перевода десятичного числа в двоичное описан в теоретической части билета 3) будем записывать значения цифры по порядку в массив `binary` из целочисленных элементов. При распечатке ответа необходимо элементы массива вывести в обратном порядке следования.

Блок-схема алгоритма перевода числа из десятичной в двоичную систему счисления имеет следующий вид.



Program decimal_to_binary;

{Программа, переводящая заданное десятичное число в двоичное; начало раздела описаний переменных и констант}

Var *N*, *N_N* : **integer**; {описываются две целочисленные переменные, имеющие смысл значения самого натурального числа *N* и числа его разрядов при двоичном представлении соответственно }

i : **integer**; {целочисленная переменная, счетчик цикла}

ostatok : **integer**; {вспомогательная переменная, имеющая смысл остатка от деления числа *N* на 2}

binary : **array**[1..1000] **of integer**; {целочисленный массив, каждый элемент которого представляет собой один разряд в двоичном представлении числа; значения элементов данного массива — либо 1, либо 0}

Begin {начало раздела операторов}

Writeln('Введите целое неотрицательное число *N* '); {Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры значения *N* }

Readln(*N*); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

N_N:=0; {Обнуляется начальное значение числа разрядов в двоичном представлении числа}

If (*N*>0) **then** {Если *N*>0, то выполняется следующий блок в программе}

Begin

While (*N*>0) **do**

Begin

Ostatok:=*N mod* 2; {вычисляется остаток от деления числа *N* на 2}

N_N:=*N_N*+1; {Увеличивается количество двоичных разрядов в числе, т.е. на единицу увеличивается число элементов массива}

Binary[*N_N*]:=ostatok; {остаток заносится в массив}

N:=*N div* 2; {выполняется целочисленное деление числа *N* на 2, т.е. результатом является целая часть частного}

End;

End

Else {Если значение введенного числа *N* нулевое, то и двоичное представление такое же, т.е. 0}

Begin

N_N:=1; {Число элементов массива равно 1}

Binary[N_N]:=0; {Единственный элемент массива нулевой}

End;

WriteIn('Двоичное представление числа ', S, ' таково: ');

for i:=N_N **downto** 1 **do** **Write**(Binary[i]);

{Организован цикл для распечатки результата, порядок печати — обратный к порядку следования элементов в массиве}

End.

Аналогичная программа на языке Basic имеет следующий вид.
(В языке Basic «/» — оператор деления, «\» — оператор получения целой части от деления.)

REM десятичное в двоичное

INPUT "Введите целое неотрицательное число N "; N

DIM binary[N]

N_N=0

IF N>0 **THEN**

WHILE N>0

Ostatok=N - (N \ 2)*2

N_N=N_N+1

Binary[N_N]=ostatok;

N=N \ 2

WEND

ELSE

N_N=1

Binary(N_N)=0

END IF

PRINT "Двоичное представление числа "; S; " таково: "

FOR i=N_N **TO** 1 **STEP** -1

PRINT **Binary**(i)

NEXT i

END

Результат работы программы при различных начальных данных приведен в таблице.

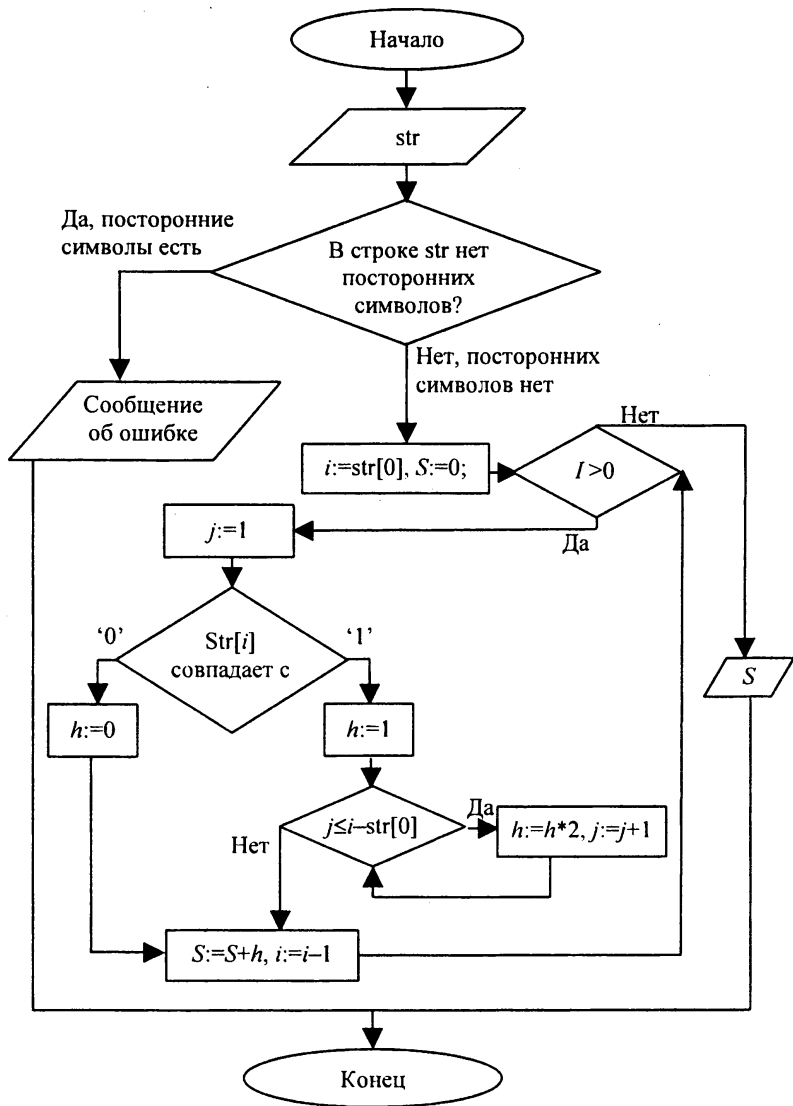
	N	Ответ
1	0	0
2	1	1
3	2	10
4	4	100
5	10	1010
6	121	1111001

Проверку результата работы программы по переводу чисел из десятичной системы счисления в двоичную и наоборот можно легко осуществить при помощи стандартной программы Windows «Калькулятор». Для этого необходимо запустить программу «Калькулятор», выбрать в меню Вид команду **Инженерный**, ввести число. Дальнейший перевод числа из одной системы счисления в другую осуществляется путем выбора системы счисления в левой части рабочего окна программы (Dec — десятичная, Bin — двоичная).

3. Для реализации данной программы на языке программирования Pascal будем считать, что в массиве из N элементов хранятся цифры целого двоичного числа (в виде символов). Тип данных, наиболее подходящий для описания такого массива, — String. Преимущество данного типа по сравнению с типом array состоит в следующем: массив array имеет фиксированную длину, определённую при описании, а строка str типа string характеризуется текущей длиной (т.е. количеством символов в массиве в данный момент времени), которая хранится в нулевом элементе строки. Максимальная длина строки составляет 255 элементов, доступ к ним осуществляется по тем же правилам, как и для массивов (т.е. по номеру элемента в строке).

Алгоритм перевода двоичного числа в десятичное следующий: предположим, что есть двоичное число $\overline{a_n a_{n-1} \dots a_3 a_2 a_1}$ (здесь a_i — либо 0, либо 1), тогда его десятичный эквивалент может быть посчитан по формуле:

$$S = a_n * 2^{n-1} + a_{n-1} * 2^{n-2} + \dots + a_3 * 2^2 + a_2 * 2^1 + a_1 * 2^0.$$



Program binary_to_decimal;

{Программа, переводящая заданное двоичное число в десятичное; начало раздела описаний переменных и констант}

Var S : integer; {описывается целочисленная переменная, имеющая смысл десятичного числа}

Str : **string**; {строка содержащая двоичное представление числа}

i, j : **integer**; {целочисленные переменные, счетчики циклов}

N : **integer**; {целочисленная переменная, имеющая смысл числа символов в строке, отличных от нулей и единиц}

h : **integer**; {вспомогательная переменная, равная одному из слагаемых в сумме}

$$S = a_n * 2^{n-1} + a_{n-1} * 2^{n-2} + \dots + a_3 * 2^2 + a_2 * 2^1 + a_1 * 2^0 \}$$

Begin {начало раздела операторов}

Writeln('Введите двоичное число '); {Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры двоичного числа }

Readln(str); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

N:=0; {Начальное значение количества символов в строке, отличных от нулей и единиц, равно нулю}

For i:=1 to str[0] do

If (not ((str[i]='1') or (str[i]='0'))) **then N:=N+1**;

{Осуществляется проверка корректности введения двоичного числа, проверяется наличие посторонних символов в строке, если посторонний символ встречается, то значение счетчика *N* увеличивается на единицу}

If (N=0) then {Ниже следующий блок выполняется, если число введено корректно}

Begin

S:=0; {Начальное значение десятичного числа}

For i:=str[0] to 1 do {Последовательно "перебираем" ряды двоичного числа, начиная с младших}

Begin

If str[i]='1' then

Begin

h:=1;

for j:=1 to str[0]-i do h:=h*2; {Вычисляем соответствующие степени числа 2}

End

Else h:=0;

S:=S+h; {Суммирование в соответствии с формулой}

$$S = a_n * 2^{n-1} + a_{n-1} * 2^{n-2} + \dots + a_3 * 2^2 + a_2 * 2^1 + a_1 * 2^0 \}$$

End;

WriteLn('введенное двоичное число в десятичном представлении имеет вид ', S); {Ответ выводится на экран}

End

Else WriteLn('Ошибка в вводе двоичного числа, в строке найдены посторонние символы'); {Если число введено некорректно, то выводится сообщение об ошибке}

End.

Аналогичная программа на языке Basic имеет следующий вид.

REM двоичное в десятичное

INPUT "Введите двоичное число "; str\$

N=0

FOR i=1 **TO** LEN(str\$)

IF (**NOT** ((MID\$(str\$,i,1) = "1") **OR** (MID\$(str\$,i,1) = "0"))) **THEN**

N=N+1

NEXT i

IF N=0 **THEN**

S=0

FOR i=LEN(str\$) **TO** 1 **STEP** -1

IF str[i] = "1" **THEN**

h=1

FOR j=1 **TO** LEN(str\$)-i

h=h*2

NEXT j

ELSE

h=0

END IF

S=S+h

NEXT i

PRINT "введенное двоичное число в десятичном представлении имеет вид "; S

ELSE

PRINT "Ошибка в вводе двоичного числа, в строке найдены посторонние символы"

END IF

END

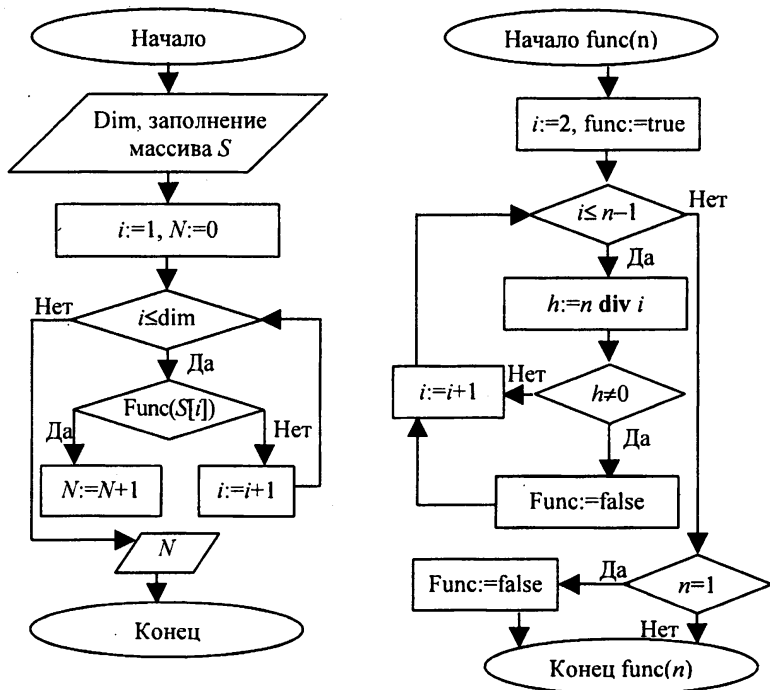
Результат работы программы при различных начальных данных приведен в таблице.

	Двоичное число	Ответ
1	0	0
2	1	1
3	10	2
4	100	4
5	1010	10
6	1111001	121

Задача 4*. Массивы и функции.** Подсчитать количество простых чисел в одномерном массиве, состоящем из целых чисел, полученных случайным образом.

Для реализации данной задачи введем функцию, возвращающую значение логического типа и проверяющую является ли число простым. Напомним, что простым называют число, которое имеет ровно два различных делителя (единицу и самого себя).

Блок-схема алгоритма подсчета количества простых чисел в массиве имеет следующий вид.



Program prime;

{Программа, подсчитывающая количество простых чисел
массиве; начало раздела описаний переменных и констант}

Var s : array[0..1000] of integer; {массив чисел}

N : integer; {целочисленная переменная, имеющая смысл ко-
личества простых чисел в массиве}

dim : integer; {количество элементов в массиве}

function func(a : integer) : boolean; {Функция проверки «являет-
ся ли число простым», аргумент — проверяемое натурально
число, возвращаемое значение равно true, если число просто
и false, если число простым не является }

var k, h: integer; {Описываются локальные переменные — счет-
чик цикла и остаток от деления проверяемого числа на друго-
е целое число}

func1 : Boolean; {Вспомогательная логическая переменная
имеющая смысл ответа на вопрос: «Число простое?». }

begin

func1:=true; {Предполагаем сначала, что проверяемое число
простое }

for k:=2 to (a-1) do {Проверяем делимость числа *a* последо-
вательно на все числа от 2 до самого числа *a* (само *a* не
включая) }

begin

h:=a mod k; {Вычисляется остаток от деления}

if (h<>0) then func1:=false; {Если остаток нулю не равен,
то число составное}

end;

if (a=1) then func1:=false; {Число *a* простым не является}

func:=func1; {Функция возвращает значение, равное func1}

end;

Begin {начало раздела операторов}

**Writeln('Введите желаемое число случайных элементов в
массиве');** {Оператор вывода данных, на экран выводится
просьба о вводе с клавиатуры числа dim}

Readln(dim); {Оператор ввода данных, в этот момент испол-
нение программы приостанавливается и она ожидает вво-
да данных}

N:=0; {Начальное значение количества простых чисел в мас-
сиве равно нулю}

For i:=1 to dim do

Begin

s[i]:=round(random(1)*1000); {Массив заполняется случайными целыми числами}

write(s[i]); {Элементы массива выводятся на печать}

If (func(s[i])) then N:=N+1;{Количество простых чисел в массиве увеличивается на 1, если число простое; проверку осуществляет функция func, аргументом которой является проверяемое число, а её значение имеет логический тип}

end;

Writeln('Количество простых чисел в массиве равно ', N);{Ответ выводится на экран}

End.

Аналогичная программа на языке Basic имеет следующий вид.

REM простые числа (на языке QBasic)

DEFINT N, dim1

DIM S(100) **AS** **INTEGER**

FUNCTION **FUNC**(a **AS** **INTEGER**)

DEFINT k, h

DEFBOOL func1

func1=true

FOR k=2 **TO** a-1

h=a - (a \ k)*k

IF h<>0 **THEN** func1=false

NEXT k

IF a=1 **THEN** func1=false

FUNC=func1

END FUNCTION

INPUT "Введите желаемое число случайных элементов в массиве"; dim1

N=0

FOR i=1 **TO** dim1

S(i)=INT(RND*1000)

PRINT S(i)

IF **FUNC**(S(i)) **THEN** N=N+1

NEXT i

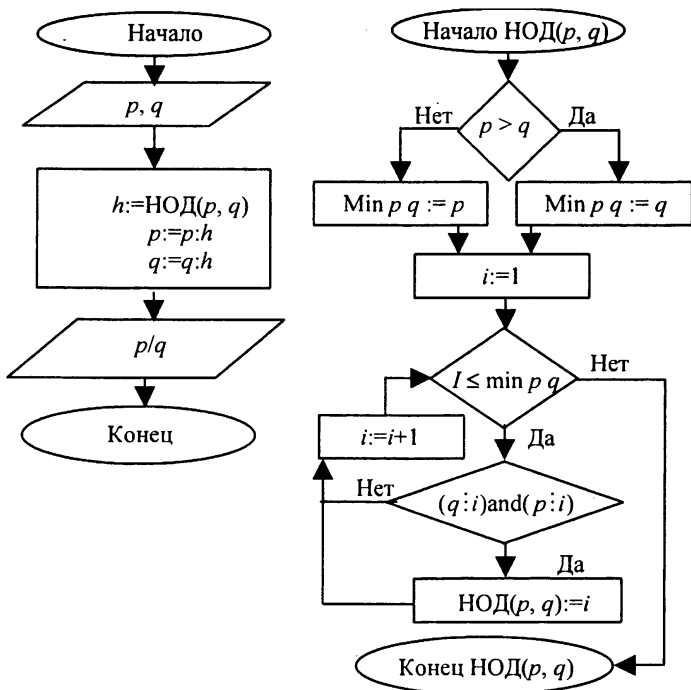
PRINT "Количество простых чисел в массиве равно "; N

END

Задача 5*. Сокращение дробей.** Построить алгоритм (нарисовать блок-схему) и составить программу на языке программирования, осуществляющую сокращение простой дроби p/q .

Для сокращения дроби необходимо найти НОД (наибольший общий делитель) двух чисел p и q . Новая дробь будет иметь вид $(p:\text{НОД}(p,q)) / (q:\text{НОД}(p,q))$. Напомним, что НОД — это наибольшее натуральное число, на которое делятся без остатка и p , и q .

Блок-схема алгоритма такова.



Program p_q;

{Программа, производящая сокращение дроби p/q }

Var p,q : integer; {Соответственно числитель и знаменатель дроби }

h : integer; {Наибольший общий делитель}

function NOD(a,b : integer):integer; {Функция возвращает НОД двух своих натуральных аргументов}

var k, minpq: integer; {Локальные переменные, счетчик цикла и минимум из двух аргументов функции}

begin

if (a>b) **then** minpq:=b **else** minpq:=a; {Определяем минимальный из двух аргументов функции}

for k:=1 **to** minpq **do** {Последовательно перебираем числа от 1 до минимального аргумента функции в поисках максимального делителя}

if ((a mod k =0)and(b mod k=0)) **then** NOD:=k; {Как только и *a* и *b* оба делятся на проверяемое число, то НОД увеличивается; отметим, что НОД будет определено всегда, т.е. если даже у *a* и *b* нет общих делителей, отличных от единицы, то значение НОД будет получено уже на первом шаге цикла}

end;

Begin {начало раздела операторов}

Writeln('Введите значение числителя дроби (не нуль!!!)');
{Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры числа *p*}

Readln(p); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

Writeln('Введите значение знаменателя дроби (не нуль!!!)');
{На экран выводится просьба о вводе с клавиатуры числа *q*}

Readln(q);

h:=NOD(p,q); {Вычисляем НОД *p* и *q*}

p:=p div h; {Сокращаем число *p* на НОД}

q:=q div h; {Сокращаем число *q* на НОД}

Writeln('Несократимая дробь имеет вид ', p, ' / ', q);{Ответ выводится на экран}

End.

Аналогичная программа на языке Basic имеет следующий вид.

REM сокращение p/q (на языке QBasic)

DEFINT p, q, h

FUNCTION NOD(a, b AS INTEGER)

DEFINT k, minpq

IF a>b **THEN**

minpq=b

ELSE

minpq=a

END IF


```

FOR k=1 TO minpq
IF ((a - (a \ k)*k=0) AND (b - (b \ k)*k=0)) THEN NOD=k
NEXT k
END FUNCTION
INPUT "Введите значение числителя дроби (не нуль!!!)"; p
INPUT "Введите значение знаменателя дроби (не нуль!!!)"; q
h=NOD(p,q)
p=p \ h
q=q \ h
PRINT "Несократимая дробь имеет вид "; p; " / "; q
END

```

Результат работы программы при различных начальных данных приведен в таблице.

	p	q	Ответ
1	1	2	$\frac{1}{2}$
2	3	2	$\frac{3}{2}$
3	5	10	$\frac{1}{2}$
4	10	5	$\frac{2}{1}$
5	121	11	$\frac{11}{1}$
6	6	36	$\frac{1}{6}$

СОДЕРЖАНИЕ

БИЛЕТ 13

1. Понятие информации. Виды информации. Роль информации в живой природе и в жизни людей. Язык как способ представления информации: естественные и формальные языки.

Основные информационные процессы: хранение, передача и обработка информации.

БИЛЕТ 24-5

1. Измерение информации: содержательный и алфавитный подходы. Единицы измерения информации.

БИЛЕТ 36-7

1. Дискретное представление информации: двоичные числа; двоичное кодирование текста в памяти компьютера. Информационный объем текста.

БИЛЕТ 48-9

1. Дискретное представление информации: кодирование цветного изображения в компьютере (растровый подход). Представление и обработка звука и видеоизображения. Понятие мультимедиа.

БИЛЕТ 510

1. Процесс передачи информации, источник и приемник информации, канал передачи информации. Скорость передачи информации.

БИЛЕТ 611-12

1. Понятие алгоритма. Исполнитель алгоритма. Система команд исполнителя (на примере учебного исполнителя). Свойства алгоритма. Способы записи алгоритмов; блок-схемы.

БИЛЕТ 1324-25

1. Понятие файла и файловой системы организации данных (папка, иерархическая структура, имя файла, тип файла, параметры файла). Основные операции с файлами и папками, выполняемые пользователем. Понятие об архивировании и защите от вирусов.

БИЛЕТ 1426

1. Информационные ресурсы общества. Основы информационной безопасности, этики и права.

БИЛЕТ 1527-29

1. Технологии работы с текстовыми документами. Текстовые редакторы и процессоры: назначение и возможности. Основные структурные элементы текстового документа. Шрифты, стили, форматы. Основные приемы редактирования документа. Встраиваемые объекты. Понятие гипертекста.

БИЛЕТ 1630-31

1. Технологии работы с графической информацией. Растровая и векторная графика. Аппаратные средства ввода и вывода графических изображений. Прикладные программы работы с графикой. Графический редактор. Основные инструменты и режимы работы.

БИЛЕТ 1732

1. Табличные базы данных (БД): основные понятия (поле, запись, первичный ключ записи); типы данных. Системы управления базами данных и принципы работы с ними. Поиск, удаление и сортировка данных в БД. Условия поиска (логические выражения); порядок и ключи сортировки.

БИЛЕТ 1

1. Понятие информации. Виды информации. Роль информации в живой природе и в жизни людей. Язык как способ представления информации: естественные и формальные языки. Основные информационные процессы: хранение, передача и обработка информации.

Информация (от латинского «разъяснение», «набор сведений») — это связанные между собой, неизвестные до их получения сведения, дающие представление о явлениях или объектах. Информация бывает:

- **актуальная** (информация ценная в данный момент времени),
- **достоверная** (информация, полученная без искажений),
- **понятная** (информация, выраженная на языке понятном тому, кому она предназначена),
- **полная** (информация, достаточная для принятия правильного решения или понимания),
- **полезная** (полезность информации определяется субъектом, получившим информацию в зависимости от объема возможностей её использования). Восприятие информации может идти через органы слуха, зрения, обоняния, вкуса, осязания (органолептическая информация). Некоторые из этих видов информации могут воспринимать животные и растения.

Полученная человеком информация даёт ему знания об окружающем мире. Современный человек должен уметь получать информацию, её обрабатывать, передавать, оценивать её качества (полезность, достоверность и т.д.), уметь взаимодействовать в информационном поле с другими людьми. Современные компьютерные технологии предоставили человеку ши-

БИЛЕТ 2

1. Измерение информации: содержательный и алфавитный подходы. Единицы измерения информации.

В настоящее время существуют два подхода к измерению информации: содержательный и алфавитный.

При **содержательном подходе** объём информации приравнивается к совокупности знаний человека, и получение им новой информации ведет к расширению его знаний. Поэтому, если полученная объектом информация является для него новой и ранее не известной, то говорят, что сообщение информативно, (т.е. содержит ненулевую информацию). Поскольку каждый человек в силу своей специальности, возраста, уровня образования, начитанности, круга увлечений, места жительства и многих других факторов обладает своим специфическим набором сведений, то информативность сообщения — понятие субъективное (т.е. зависит от конкретного человека, воспринимающего данное сообщение). При восприятии информации человеком важна её понятность, т.е. информация должна быть в виде, доступном для понимания, иначе она перестает быть информативной. Для того чтобы информация была понятной, она должна быть выражена на языке, доступном для понимания человеку. При этом во многих случаях от него самого требуется иметь определённые знания, необходимые для получения вновь поступающей информации.

БИЛЕТ 18 33

1. Технологии обработки информации в электронных таблицах (ЭТ). Структура электронной таблицы. Типы данных: числа, формулы, текст. Правила записи формул. Основные встроены функции. Абсолютные и относительные ссылки. Графическое представление данных.

БИЛЕТ 19 34-35

1. Основные принципы организации и функционирования компьютерных сетей. Интернет. Информационные ресурсы и сервисы компьютерных сетей: Всемирная паутина, файловые архивы, интерактивное общение. Назначение и возможности электронной почты. Поиск информации в Интернете.

БИЛЕТ 20 36

1. Понятие модели. Информационная модель. Виды информационных моделей (на примерах). Реализация информационных моделей на компьютере. Пример применения электронной таблицы в качестве инструмента математического моделирования.

БИЛЕТ 7 13-14

1. Основные алгоритмические структуры: следование, ветвление, цикл; изображение на блок-схемах. Разбиение задачи на подзадачи. Вспомогательные алгоритмы.

БИЛЕТ 8 15-16

1. Величины: константы, переменные, типы величин. Присваивание, ввод и вывод величин. Линейные алгоритмы работы с величинами.

БИЛЕТ 9 17-18

1. Логические величины, операции, выражения. Логические выражения в качестве условий в ветвящихся и циклических алгоритмах.

БИЛЕТ 10 19-21

1. Представление о программировании: язык программирования (на примере одного из языков высокого уровня); примеры несложных программ с линейной, ветвящейся и циклической структурой.

БИЛЕТ 11 22

1. Основные компоненты компьютера, их функциональное назначение и принципы работы. Программный принцип работы компьютера.

БИЛЕТ 12 23

1. Программное обеспечение компьютера, состав и структура. Назначение операционной системы. Командное взаимодействие пользователя с компьютером. Графический пользовательский интерфейс.

Данное свойство положено в основу процесса обучения человека, т.е. последовательного процесса получения информации от простого к более сложному.

Для количественного измерения поступающей информации была введена специальная единица измерения бит, являющаяся основной и наименьшей единицей измерения объема информации. Другими (производными) единицами информации являются: байт = 8 бит, Кбайт = 1024 байта, Мбайт = 1024 Кбайта, Гбайт = 1024 Мбайта. Бит — это количество информации в сообщении, уменьшающем неопределенность знаний в два раза. Очевидно, что два сообщения, следующие друг за другом и уменьшающие неопределенность, имеющуюся на момент их получения, в 2 раза каждое (т.е. в целом в 4 раза), дадут 2 бита информации, три таких сообщения — 3 бита и т.д. В качестве примеров можно привести игру бинго или игру в орлянку. В первом случае ведущий случайным образом вытаскивает из мешочка специальный пронумерованный шар (например, один из 64), и выпадает одно из значений от 1 до 64, во втором при подбрасывании монеты выпадает либо орёл, либо решка. При бросании монеты человек получает одно из двух возможных информационных сообщений, что и является уменьшением неопределенности знаний в два раза (т.е. говорят, что получен один бит информации). При игре бинго исходов шестьдесят четыре, и неопределенность знания уменьшается

рокие возможности использования больших объемов информации, находящихся на носителях на значительном удалении от самого человека.

Основными информационными процессами являются обработка, хранение и передача информации. **Передача** информации обеспечивает её распространение. Устройствами передачи информации в современном мире являются телефон, радио, телевизионные передатчики, телекоммуникативные сети. Все это, так называемые, **каналы передачи информации**. Для отправки информации необходимо на одном конце такого канала иметь устройство-передатчик, на другом — приемник. Одной из важнейших проблем в современном мире является обеспечение сохранности информации при её передаче, так как различные помехи на линиях связи могут искажать информацию и даже вести к её утере. Другая проблема — защита информации от хищения.

Обработка информации — это преобразование ранее полученной информации к иной форме или иному содержанию. Преобразование формы информации означает, что человек по своему усмотрению может перевести информацию в символическую, текстовую, графическую или звуковую форму.

Хранение информации. Цель хранения информации — её передача во времени, а также обеспечение возможности её многократного использования.

в шестьдесят четыре раза, и тогда $2x = 64$, где $x = 6$ — выраженное в битах количество полученной информации, являющееся решением приведённого уравнения. Заметим, что в общем случае полученное информационное сообщение может нести дробное количество информации: количество информации будет выражено целым числом, только если неопределённость знаний уменьшилась в число раз, являющееся целочисленной неотрицательной степенью числа 2.

Итак, количество информации i , содержащееся в сообщении о том, что произошло одно из N равновероятных событий, есть решение уравнения $N = 2^i$.

Другой подход к измерению информации, **алфавитный**, рассматривает сообщение как дискретную последовательность знаков из специального набора символов (алфавита), при этом количество информации (в битах) в данном сообщении подсчитывается как произведение количества символов в сообщении на количество битов, необходимых для кодирования одного символа в алфавите. В этом случае анализ содержания сообщения не происходит.

Полное количество символов алфавита N называется его **мощностью**. При этом в знакомых нам языках учитываются не только буквы, но и знаки препинания, пробелы, специальные символы (№, \$, % и другие).

Для представления информации в различных технических устройствах используется её преобразование в последовательность нулей и единиц.

целого числа самый левый разряд означает его знак (0 — если число положительное, 1 — если число отрицательное). При этом для представления отрицательных чисел используется, так называемый, дополнительный код. Для его получения необходимо определить двоичное представление модуля числа, произвести инверсию (заменить в записи числа нули единицами, а единицы нулями и прибавить 1). Таким образом, для представления числа -23 нам необходимо инвертировать его двоичную запись 0000000000010111: получим 111111111101000. Прибавляя единицу, окончательно имеем 111111111101001.

Для представления вещественного числа в памяти компьютера отводится 4 байта (если требуется большая точность, то 8 байт). При этом число A представляется в виде произведения мантиссы на некоторую степень двойки: $A = m \cdot 2^n$, где m — мантисса (значащие разряды), n — порядок (т.е. положение значатой относительно первого разряда мантиссы). При этом в памяти компьютера среди выделенных 4 байт для представления числа отдельно отводятся биты для хранения мантиссы, порядка и по одному биту для хранения знака числа и знака порядка.

Рассмотрим двоичное кодирование **текстовой информации** на ЭВМ. Поскольку компьютер может воспринимать информацию только в виде двоичного кода, то для представления в компьютерной памяти текстов производится кодирование символов алфавита некоторыми целыми числами, для которых компьютерное представ-

БИЛЕТ 3

1. Дискретное представление информации: двоичные числа; двоичное кодирование текста в памяти компьютера. Информационный объём текста.

Вся информация, хранящаяся и обрабатываемая компьютером, представляется двоичным кодом, т.е. последовательностью из нулей и единиц. Для такого представления, а также для обратного преобразования кода в сообщение, понятное человеку, применяется **кодирование** и **декодирование** информации соответственно. В процессе кодирования происходит преобразование входной информации (вводимой, например, человеком с клавиатуры компьютера) в двоичный код, в процессе декодирования происходит обратное действие. Выбор двоичного представления информации в ЭВМ обусловлен конструктивными особенностями вычислительной техники. Дело в том, что ячейка памяти, хранящая один бит информации и являющаяся минимальной в памяти компьютера, может в каждый момент времени находиться в одном из состояний: намагниченном (обозначается 1) или ненамагниченном (0). Передача информации идёт посредством электронных импульсов: 1 — есть сигнал, 0 — нет сигнала. Такое представление информации является дискретным (т.е. прерывистым), поскольку объём представленной информации обязательно равен целому числу бит.

БИЛЕТ 4

1. Дискретное представление информации: кодирование цветного изображения в компьютере (растровый подход). Представление и обработка звука и видеоизображения. Понятие мультимедиа.

Существует несколько технических систем записи цвета, которые иногда называют цветowymi моделями. Рассмотрим основные из них.

RGB (Red-Green-Blue — красный-зелёный-синий). Цвет идентифицируется набором из трёх упорядоченных целых чисел, значения которых меняются от 0 до 255. Так красному соответствует набор (255,0,0), зелёному — (0,255,0), синему — (0,0,255), белый цвет состоит из всех трёх цветов максимальной интенсивности, т.е. (255,255,255), чёрный получается при выключении всех трёх базовых цветов — (0,0,0). Все другие цвета и оттенки получают смешиванием трех базовых цветов различной интенсивности (чем больше значение, тем ярче цвет). Для кодировки цвета конкретной точки изображения выделяются три байта памяти компьютера, а именно: для кодирования числа, отвечающего за интенсивность красного цвета отводится первый байт, для кодирования зелёного — второй, для кодирования синего — третий. RGB является аддитивной моделью, так как при наложении одной составляющей цвета на другую не только меняется его тон, но и увеличивается яркость суммарного цвета. Цветовая модель RGB применяется в телевизорах и в цветных мониторах компьютеров.

Для получения компьютерного представления числа, записанного в привычной нам десятичной системе счисления, необходимо осуществить его перевод в двоичную систему. С этой целью для целых положительных чисел используется метод последовательного деления целой части частного на 2. Так, например, переведем в двоичную систему число 23:

$23 : 2 = 11 + 1 : 2$, т.е. в остатке 1;
 $11 : 2 = 5 + 1 : 2$, т.е. в остатке 1;
 $5 : 2 = 2 + 1 : 2$, т.е. в остатке 1;
 $2 : 2 = 1 + 0 : 2$, т.е. в остатке 0;
 $1 : 2 = 0 + 1 : 2$, т.е. в остатке 1.

В частном оказалось число, меньшее делителя, поэтому процесс последовательного деления останавливаем. Далее записываем число 23 в двоичной системе счисления: последовательность из нулей и единиц, полученных в остатках от деления, но взятые в обратном порядке, по отношению к тому, как они возникали в процессе деления, т.е. $23_{10} = 10111_2$ (индекс внизу обозначает основание системы счисления, в которой записано число). На компьютере для хранения числа выделяется k -разрядная ячейка памяти, наиболее распространенной является 2-байтная ячейка, содержащая 16 бит. В этом случае при записи числа 23 в компьютер его двоичный эквивалент будет дополнен слева незначащими нулями, т.е. представление числа 23 на ЭВМ будет иметь вид 0000000000010111 (всего 16 нулей и единиц).

Как правило, в компьютерном представлении

При этом информация, закодированная знаками некоторого алфавита по строгому однозначному алгоритму, ставящему в соответствие букве алфавита двоичное число, может быть переведена в двоичный код (и при необходимости обратно). При этом минимальное количество битов, необходимое для кодирования одного символа алфавита, определяется как корень уравнения $N = 2^i$, (в случае если N не является степенью двойки, то следует на место N в уравнении поставить ближайшее большее N число, являющееся степенью числа 2).

Для примера посчитаем количество информации в сообщении «Жил старик со своею старухой у самого синего моря; они жили в ветхой землянке ровно тридцать лет и три года.», если предполагать, что каждый символ данного алфавита закодирован двухбайтным словом. Длина фразы составляет 108 символов (с учётом пробелов и знаков препинания!). Тогда информационный объём сообщения можно посчитать как $108 \cdot 2 = 216$ байта или, что то же самое, $216 \cdot 8 = 1728$ бита.

Отметим, что алфавитный подход к измерению информации, применяемый в технических устройствах, не только позволяет пользователю работать на ЭВМ с текстовой и графической информацией, но и оценивать объёмы необходимой для сообщения памяти на носителях информации или необходимые технические характеристики устройств для передачи данных на расстояние.

CMYK (Cyan-Magenta-Yellow-black — голубой-пурпурный-жёлтый-чёрный). Цвет идентифицируется набором из четырех упорядоченных целых чисел, значения которых меняются от 0 до 100 и соответствуют интенсивности (в процентах) базовых цветов голубого, пурпурного, жёлтого и чёрного. Эти цвета (кроме чёрного) получаются при вычитании основных цветов, используемых в модели RGB, из белого цвета, чёрный цвет задается отдельно. Данная цветовая модель является субтрактивной, поскольку увеличение яркости основного цвета (краски) влечет снижение яркости картинки в целом. Такая цветовая модель получила наибольшее распространение в цветной полиграфии.

HSB — цветовая модель, наиболее удобная для восприятия человеком. В ней используются три параметра, характеризующие цвет: *тон* (Hue), *насыщенность* (Saturation), *яркость* (Brightness). Под *тоном* цвета обычно понимают близость оттенка к тем или иным цветам спектра. *Насыщенность* — это степень выраженности тона в цвете. *Яркость* определяется уровнем действующего на глаз излучения.

В настоящее время существует большое количество программ, называемых **графическими редакторами**, которые предоставляют человеку широкие возможности для работы как с цветными, так и с монохромными изображениями, позволяют их редактировать, преобразовывать, обрабатывать.

Отметим, что количество оттенков цвета, которые можно закодировать, обусловлено объёмом памяти, выделяемой для представления цвета каждой конкретной точки. Если выделяется один

ление на уже известно. Для удобства кодирования символов целыми десятичными числами, как правило, составляются кодировочные таблицы, которые утверждены международными стандартами. В кодовых таблицах латинские прописные и строчные буквы располагаются в алфавитном порядке, а цифры упорядочены по возрастанию (так называемый принцип последовательного кодирования алфавитов).

В зависимости от числа битов отведенных под кодирование символов в памяти ЭВМ кодировки бывают 8- и 16-разрядные. 8-разрядной является, например, кодировка ASCII, 16-разрядной — кодировка Unicode, которые позволяют закодировать соответственно 256 и 65536 различных символов.

Если пользователю известна система кодирования, используемая в компьютере (а вернее количество разрядов отведенных под хранение одного символа), то ему не составит труда оценить объём информации, содержащейся в данном тексте. Приведем пример. Некоторая брошюра содержит 70 страниц печатного текста, на каждой странице в среднем 40 строк, в строке 60 символов (включая пробелы), используется 8-разрядная кодировка ASCII. В этом случае объём информации в книге будет составлять $70 \cdot 40 \cdot 60 = 168\,000$ байт или, что то же самое, $168\,000 \cdot 8 = 1\,344\,000$ бит.

байт, то возможно закодировать 256 различных цветов, если 2 байта — то 65536. Чем больше объем выделяемой под цвет точки памяти, тем выше качество изображения.

Как известно, звук представляет собой волну, распространяющуюся в упругой среде. Звук обладает музыкальным *тоном* (т.е. определённой частотой волны), *длительностью* и *силой* (по-другому, *интенсивностью* или громкостью), зависящей от амплитуды колебаний. Для записи звука применяется *дискретизация*, т.е. в течение некоторого промежутка времени через определённые временные интервалы производится измерение амплитуды и её кодирование двоичным числом. Устройство, выполняющее такое преобразование, носит название *аналого-цифрового преобразователя*. Устройство, осуществляющее обратный процесс преобразования кода в звук, называется *цифроаналоговым преобразователем* и используется в комплексе с фильтрами, усилителями и динамиками, осуществляющими сглаживание ступенчатого сигнала, его усиление и воспроизведение. Качество воспроизведения записанного звука зависит от частоты дискретизации и от объёма памяти, выделенного для хранения его амплитуды. Сократить объём памяти, занятой звуковой записью, возможно, если хранить информацию не в виде отдельных данных, а записывать разность между двумя соседними значениями, которые часто отличаются друг от друга не слишком сильно, а следовательно, происходит экономия памяти компьютера. В настоящее время принято записывать звуковые (музыкальные) файлы в формате MIDI.

9

БИЛЕТ 5

10

1. Процесс передачи информации, источник и приёмник информации, канал передачи информации. Скорость передачи информации.

Передача информации обеспечивает её распространение. В процессе передачи информации осуществляется следующая цепочка: источник информации $\Rightarrow$ кодирующее устройство $\Rightarrow$ канал связи $\Rightarrow$ декодирующее устройство $\Rightarrow$ получатель. Источником информации может быть как техническое устройство, так и сам человек. *Кодирующее устройство* переводит исходное сообщение в пригодный для передачи по данному каналу связи вид. Примерами кодирующих устройств являются микрофон, видеокамера, фотоаппарат и др. *Декодирующее устройство* переводит переданное сообщение из кода в вид, понятный человеку (получателю). Естественными такими устройствами служат человеческое ухо и глаз.

Устройствами передачи информации в современном мире являются телефон, радио, телевизионные передатчики, телекоммуникационные сети. Все это, являясь *каналами передачи информации*, по которым информация передается посредством сигналов (механических, электрических, тепловых). Для отправки информации необходимо на одном конце такого канала иметь устройство-передатчик, на другом — приёмник.

БИЛЕТ 6

11

1. Понятие алгоритма. Исполнитель алгоритма. Система команд исполнителя (на примере учебного исполнителя). Свойства алгоритма. Способы записи алгоритмов; блок-схемы.

Алгоритм — описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов. Процесс разработки алгоритма для решения задачи называется *алгоритмизацией*.

Основными свойствами алгоритма являются следующие: дискретность, детерминированность, результативность, конечность, массовость, корректность.

1) **Дискретность** алгоритма означает, что процесс решения задачи состоит из последовательности отдельных шагов, каждый из которых называется *командой*. Выполнение каждой новой команды начинается только после завершения предыдущей.

2) **Детерминированность** (определённость) алгоритма означает, что результат работы алгоритма для одного и того же набора начальных данных, будет оставаться постоянным вне зависимости от времени и места запуска программы, поскольку каждая команда алгоритма интерпретируется однозначным образом и выполняется в строго определённом случае. Наличие ошибок в алгоритме может привести к нарушению данного свойства.

(в письменной или устной форме), при помощи формального языка или при помощи блок-схемы.

Блок-схема представляет собой систему из геометрических фигур (блоков), связанных стрелками, которые указывают порядок выполнения блоков, входящих в алгоритм.

Основные обозначения, применяемые в блок-схемах

Название	Обозначение	Характер совершаемых действий
Пуск/остановка		Начало или конец алгоритма
Процесс		Действия, вычисления, операции
Принятие решения (ветвление)		Происходит проверка условия, вписанного в ромб, и выбор дальнейшего пути в зависимости от результата этой проверки
Ввод/вывод		Команда либо выводит информацию на экран или принтер, либо получает её извне (с клавиатуры, из внешнего файла и т.п.)

Объект (человек или машина), выполняющий алгоритм называется *исполнителем*. Его задача точно (в большинстве случаев формально) выполнить предписанный ему алгоритм. При этом

Одной из важнейших проблем в современном мире является обеспечение сохранности информации при её передаче, так как различные помехи на линиях связи могут искажать информацию и даже вести к её утере. Другая проблема — защита информации от хищения. Для решения всех проблем, связанных с кодированием и декодированием информации, создана специальная наука **криптография**, она же создает методы защиты информации от понимания посторонними людьми (т.е. разрабатывает искусственные языки, понятные ограниченному кругу лиц).

Для характеристики быстроты перемещения данных из одной области памяти в другую или с одного устройства на другое вводится понятие **скорости передачи данных**. Основной единицей измерения является бит/сек (бит в секунду). Производные единицы байт/сек=8 бит/сек, Кбайт/сек=1024 байт/сек. Характеристикой качества канала связи является его **пропускная способность** (т.е. максимальная скорость передачи данных).

(музыка), в котором файл с музыкальной записью представляет собой последовательность закодированных сообщений синтезатору звука, разделенных также закодированными паузами. Сообщение синтезатору звука представляет собой ряд команд, задающих высоту, тембр и громкость звука.

Для записи **видеоизображения**, также как и для записи звука, применяется дискретизация. Дело в том, что при воспроизведении последовательности картинок с частотой более 15 кадров в секунду человеческий глаз воспринимает эту последовательность как движение. Таким образом, принято кодировать видео изображение последовательностью стационарных картинок, отвечающих различным фазам движения объекта, воспроизводимыми с достаточно большой частотой. Запись изображения на компьютерные носители появилась сравнительно недавно, так как сама по себе требует больших объемов памяти. Объем памяти, занимаемый видеозаписью, возможно сократить за счет хранения не всех следующих друг за другом кадров, а изменений к ним.

Возможность записи на компьютер звука, изображения и видеоизображения привела к созданию мультимедиа. **Мультимедиа** — это многофункциональная компьютерная система, включающая совокупность компьютерных продуктов в нескольких видах: изображение, звук, анимацию, текст и т.п., совмещаемая в интерактивном режиме с другими системами и используемая для разных целей: для создания базы данных справочно-энциклопедической информации, для её накопления, обработки, хранения, издания и т.п.

не в его функциях следить за правильностью и корректностью алгоритма, он должен просто исполнять свою систему команд, хотя при возникновении команды, не являющейся для данного исполнителя допустимой, происходит отказ работы алгоритма.

Автоматическим исполнителем в современном мире является компьютер, а алгоритм, записанный в форме, доступной для «понимания» ЭВМ, называют **программой**.

Учебным исполнителем называется некоторый образ на мониторе компьютера, выполняющий некоторые простейшие действия с целью обучения школьника составлению алгоритмов. При этом исполнитель обладает допустимой для него системой команд и обитает в некоторой среде (на плоскости, на прямой, на клетчатом ограниченном поле, в лабиринте и т.д.). Рассмотрим пример. Исполнитель ЧЕРЕПАШКА может перемещаться на плоскости, оставляя непрерывный след. Допустимой системой команд являются движение **ВПЕРЕД** на x единиц и **ПОВОРОТ** (изменение направления движения) на y градусов по часовой стрелке. Тогда в результате выполнения алгоритма «**ВПЕРЕД 2 ПОВОРОТ 90 ВПЕРЕД 2 ПОВОРОТ 90 ВПЕРЕД 2 ПОВОРОТ 30 ВПЕРЕД 2 ПОВОРОТ 120 ВПЕРЕД 2**» на плоскости будет нарисована «домика», все линейные измерения которого равны 2.

3) **Результативность** алгоритма означает, что после выполнения команды, составляющих правильный алгоритм, должен получиться определенный результат. Завершение алгоритма должно произойти за конечное число шагов (**свойство конечности**). Данные свойства могут нарушаться при наличии ошибок в алгоритме.

4) **Массовость** алгоритма означает, что один и тот же алгоритм может быть применен для решения целого класса сходных задач, отличающихся исходными данными, которые выбираются из множества значений, допустимых для данного алгоритма (при этом в качестве исходных данных используются переменные). Массовость алгоритма подразумевает результативность его работы для любого набора начальных данных из некоторого допустимого множества (**свойство корректности**).

Алгоритмы бывают:

- **линейные** (действия выполняются однократно в заданном порядке);
- **циклические** (повторение действий до тех пор, пока не станет верным условие выхода из цикла);
- **разветвляющиеся** (выполняется та или иная последовательность действий, в зависимости от некоторого логического условия);
- **вспомогательные** (некоторое условное обозначение группы команд, выполняющих некоторое целостное действие).

По форме представления различают алгоритмы, представленные на естественном языке

БИЛЕТ 7

13

1. Основные алгоритмические

структуры: следование, ветвление, цикл; изображение на блок-схемах. Разбиение задачи на подзадачи. Вспомогательные алгоритмы.

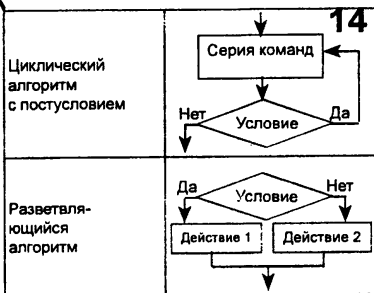
Основные виды алгоритмов (или как их ещё называют алгоритмических структур) следующие:

- линейный алгоритм (следование);
- циклический алгоритм;
- разветвляющийся алгоритм;
- вспомогательный алгоритм.

Рассмотрим последовательно каждый из них.

Линейный алгоритм — все этапы выполняются строго последовательно друг за другом, однократно и без пропусков. Изображением на блок-схеме линейного алгоритма является прямоугольник или ряд следующих друг за другом прямоугольников, соединённых стрелками. Внутри прямоугольников принято вписывать наименования выполняемых действий.

Циклический алгоритм применяется для обеспечения многократного выполнения некоторой последовательности действий (*тела цикла*). Индикатором прекращения или продолжения повторений является условие выхода из цикла, т.е. логическое выражение, принимающее значение «истина» или «ложь». Если оно истинно, то повторения прекращаются, в противном случае — продолжают. Очевидно, что логическое выражение, отвечающее за прекращение работы цикла, должно составляться очень аккуратно, так как иначе велик риск попадания в бесконечный цикл с невозможностью выхода из него (зацикливания).



Разветвляющийся алгоритм предоставляет возможность создать некоторый переключатель, который в зависимости от выполнения или невыполнения условия направляет работу алгоритма по одному из возможных путей, т.е. ведёт к выполнению определённой группы действий, продиктованных ситуацией. Отметим, что в некоторых случаях такая последовательность действий может быть пустой.

Вспомогательный алгоритм представляет собой поименованную некоторую группу команд, выполняющих некоторое целостное действие, состоящее из нескольких элементарных. В более крупные алгоритмы вспомогательный алгоритм входит под своим именем и его дета-

БИЛЕТ 8

15

1. Величины: константы, переменные, типы величин. Присваивание, ввод и вывод величин. Линейные алгоритмы работы с величинами.

Для рассмотрения понятия «величина» рассмотрим некоторый алгоритмический язык программирования, поддерживаемый на компьютере. ЭВМ работает по определённой программе с набором данных, состоящих из множества величин, под которые отведены определённые объёмы памяти (ячейка или несколько ячеек). В любом языке программирования каждая величина имеет своё имя, тип и значение в данный момент. Имена переменных (или, как их ещё называют, **идентификаторы**) состоят из последовательности букв, цифр, специальных знаков (таких, например, как подчеркивание). Допустимые имена для величин могут слегка варьироваться в зависимости от конкретного языка программирования.

В нижеприведённой таблице на примере языка программирования Pascal перечислены наиболее распространённые типы данных, а также их наиболее характерные черты. Каждый тип данных характеризуется своим внутренним представлением на ЭВМ, множеством значений и операциями, которые допустимо для них производить. Множество значений величин диктуется объёмом памяти, отведённым для её хранения. Большинство языков программирования предоставляет программисту возможность конструи-

В языках программирования идентификаторы бывают константами и переменными.

В отличие от константы переменная изменяет величину, которая может изменять своё значение в ходе работы программы. Программист должен заранее определить, с какой из этого типа величиной он будет работать.

Для работы с величинами необходимо освоить следующие команды: присваивание, ввод и вывод. **Присваивание** — это задание значения переменной. Осуществляет его оператор присваивания (в Pascal :=), который нельзя путать с операцией сравнения (=), осуществляющей проверку идентичности объектов. **Оператор присваивания**, а также **оператор ввода данных** — это два возможных способа задания значения переменных. При применении оператора ввода выполнение программы приостанавливается, и компьютер ждёт ввода данных с клавиатуры или с какого-либо другого устройства. При этом вводимые данные должны быть представлены в формате, соответствующему типу вводимой величины. До момента первого присваивания переменная значения не имеет и является неопределённой.

Для представления полученного результата используются команды **вывода**. По команде вывода значение переменной преобразуется в вид, понятный человеку, и выводится на монитор или принтер.

Все программы при своей работе производят действия над переменными, эти действия группируются в линейные алгоритмы, являющиеся

лизация (расшифровка) не указывается. Хотя предполагается, что ожидаемый результат его действия известен. Вспомогательный алгоритм, записанный на языке программирования, называют **подпрограммой**. При написании больших по объёму программ выделение отдельных кусков в подпрограммы делает структуру алгоритма более прозрачной и доступной для понимания. Кроме того, во многих случаях в программе вызов подпрограммы осуществляется не единожды, что также говорит о необходимости обозначения групп вызываемых команд одним именем. Во многих случаях некоторый алгоритм, оформленный в виде вспомогательного, может быть применён в различных алгоритмах, имеющих совсем разное предназначение.

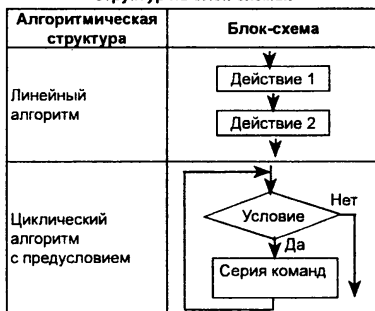
Таким образом, в процессе алгоритмизации задачи, происходит разбиение её на более простые алгоритмы, которые либо уже существуют, либо могут быть легко созданы. В этом случае процесс алгоритмизации идёт от более общего к частному, от идейной структуры к детализации кусков алгоритма, составляющих основную. Такой метод называется **методом последовательной (пошаговой) детализации**.

Принципиально другим способом является **сборочный метод**. В этом случае большое количество уже готовых алгоритмов (программ) оформляется в виде библиотеки, компоненты которой используются для создания новых алгоритмов, содержание которых определяет пользователь.

составными частями более сложных алгоритмов с ветвлениями и циклами. Помимо этого, для вычисления значения выражений компьютер сам выполняет некоторый линейный алгоритм, обусловленный приоритетностью операций, входящих в это выражение. Так, например, вычисление значения выражения f , заданного формулой $a*b+a/b+abs(a)$, может происходить по следующему сценарию: найти произведение $a*b$ (значение присваивается некоторой промежуточной переменной var1) → найти частное a/b (значение присваивается некоторой промежуточной переменной var2) → найти модуль числа a , т.е. $abs(a)$, (значение присваивается некоторой промежуточной переменной var3) → найти сумму var1 и var2 (значение присваивается некоторой промежуточной переменной var4) → найти сумму var4 и var3 (что и даёт ответ, т.е. значение f). Отметим, что в целях экономии памяти, система может производить вычисления с меньшим количеством вспомогательных переменных (например, можно было вместо var4, использовать var1, записать что $var1:=var1+var2$, в этом случае произошло бы сложение выражений в правой части, а далее результат был бы записан в область памяти, соответствующую переменной var1).

что приведёт к некорректной работе программы. Условиями выхода из цикла являются превышении совершенных повторений некоторого фиксированного числа (**циклы со счетчиками**) или выполнение какого-либо условия (**циклы с условием**). Последние подразделяются на циклы с предусловием и циклы с постусловием. **Цикл с предусловием** сначала проверяет истинность логического выражения, после чего может начаться выполнение тела цикла. **Цикл с постусловием** сначала один раз выполняет тело цикла, затем проверяет логическое выражение и по результатам проверки может снова повториться тело цикла.

Примеры изображения алгоритмических структур на блок-схемах



создавать новые типы данных на основе уже имеющихся.

Типы данных

Название	Описание
Целое число Integer	Целочисленные данные; во внутреннем представлении занимают 2 байта; диапазон возможных значений от -32768 до +32767. Представление точное. Возможные операции: +, -, *, /, целочисленное деление, все операции сравнения. К целым числам возможно применение математических функций (например, модуль). Отметим, что некоторые операции (например, деление) над самими целыми числами дают вещественный результат
Вещественное число Real	Вещественные данные; занимают 6 байт; диапазон возможных значений модуля числа от 2.9E-39 до 1.7E+38, точность представления данных составляет 11 значащих цифр. Возможные операции: +, -, *, /, все операции сравнения. К вещественным числам также возможно применение математических функций (корень, синус, модуль)
Символ Char	Символ; занимает 1 байт. Возможные операции: получение по символу его кода в таблице кодировки, получение прописной буквы, если задана строчная
Строка символов String	Строка символов. Её максимальная длина должна быть задана заранее, объём отводимой памяти составляет (n+1) байт (лишний байт необходим для хранения длины строки)
Логический тип Boolean	Принимает значение «истина» (true) или «ложь» (false), занимает 1 байт. Возможные операции: логическое «и», логическое «или», логическое «не», логическое «исключающее «или».

1. Логические величины, операции, выражения. Логические выражения в качестве условий в ветвящихся и циклических алгоритмах.

Элементами логических рассуждений являются утверждения, которые либо ложны (Л, 0), либо истинны (И, 1), но не то и другое вместе. Такие утверждения называются *высказываниями*. Из простых высказываний при помощи логических операций могут быть построены более сложные составные высказывания (или формулы).

Существует три основных логических операции:

- **инверсия** (логическое отрицание, не, $\neg$);
- **конъюнкция** (логическое умножение, и, $\&$, $\wedge$);
- **дизъюнкция** (логическое сложение, или, $\vee$).

Остальные введенные логические операции алгебры логики (*эквивалентность, импликация, исключающее или*) могут быть выражены через три основных.

Приведем таблицы истинности для простейших логических функций.

A	$\neg A$
0	1
1	0

A	B	$A \wedge B$	$A \vee B$
0	0	0	0
1	1	1	1
0	1	0	1
1	0	0	1

$$\begin{aligned} A \vee 0 &= A, A \wedge 0 = 0, A \vee 1 = 1, A \wedge 1 = A, \\ 0 \rightarrow A &= 1, A \rightarrow B = \neg A \vee B. \end{aligned}$$

Структура **ветвление** обеспечивает переход по одной из ветвей алгоритма в зависимости от выполнения или невыполнения некоторого логического условия. В блок-схеме ветвление условно обозначается ромбом с записанным внутри условием, принимающим значение «истина» или «ложь»; после вычисления значения этого логического выражения осуществляется выбор дальнейшего пути: по стрелке «Да», если значение выражения истинно, либо по стрелке «НЕТ», если значение ложно.

Цикл обеспечивает многократное выполнение некоторой последовательности действий (или тела цикла) до тех пор, пока не станет ложным некоторое логическое выражение. Различают два вида циклов: «До» и «Пока». В первом случае проверка условия осуществляется после выполнения серии команд. Если логическое выражение все еще истинно, то серия команд повторяется опять, а далее снова идет проверка логического условия выхода из цикла. В цикле «Пока» сначала проверяется условие, и только после этого, если оно верно, осуществляется серия команд, затем снова проверяется истинность выражения.

1. Представление о программировании: язык программирования (на примере одного из языков высокого уровня); примеры несложных программ с линейной, ветвящейся и циклической структурой.

Язык программирования — это фиксированная система обозначений для описания алгоритмов и структур данных. Во всяком языке программирования имеется алфавит, т.е. набор символов, которые можно использовать в программе, существуют зарезервированные слова, имеющие вполне определенный смысл и определенное назначение и которые нельзя менять, предоставляется возможность вводить переменные, использовать константы (как встроенные, так и введенные самим программистом), использовать существующие и доопределять новые операторы (они задают те или иные действия, которые должна выполнять программа). **Система программирования** — некоторая единая структура, состоящая из компилятора (или транслятора) с языка программирования и некоторой инструментальной программной оболочки, способствующей повышению эффективности создания программ. **Компилятор** или **транслятор** — это программа, переводящая текст программы на языке программирования высокого уровня в эквивалентную программу на машинном языке.

Программа с ветвящейся структурой

Program abs_abs; {Программа вычисления модуля числа}

Var a:real; {описывается переменная, описываемая с числом, модуль которого мы вычисляем}

b:real; {описывается переменная, представляющая собой модуль числа}

Begin

WriteIn('Введите число');

ReadIn(a);

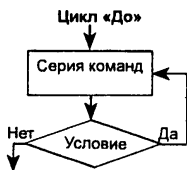
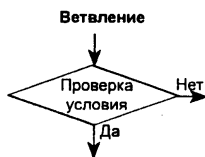
If (a>=0) **then** b:=a; **else** b:=-a; {Условный оператор **If** проверяет условие $a \geq 0$ и в зависимости от результатов проверки производит вычисление модуля числа согласно его определению}

WriteIn('Модуль числа равен ',b); {результат выводится на экран}

b:=abs(a); {модуль числа вычисляется при помощи встроенной функции языка Pascal, возвращающей модуль вещественного числа}

WriteIn('Сравним результат. Модуль числа равен ',b); {на монитор выводится тот же результат, полученный другим способом}

End.



Эквивалентность			Импликация			Исключающее или	
A	B	$A \leftrightarrow B$	A	B	$A \Rightarrow B$	A	$B \oplus A$
0	0	1	0	0	1	0	0
1	1	1	1	1	1	1	0
0	1	0	0	1	1	0	1
1	0	0	1	0	0	1	1

Порядок выполнения операций в логическом выражении задается скобками. Если скобки отсутствуют, то приоритетность операций следующая: отрицание, конъюнкция, дизъюнкция, импликация, эквивалентность.

Основные законы алгебры логики

Название закона	Формулировка
Переместительный закон	$A \vee B = B \vee A, A \wedge B = B \wedge A$
Сочетательный закон	$(A \vee B) \vee C = A \vee (B \vee C), (A \wedge B) \wedge C = A \wedge (B \wedge C)$
Распределительный закон	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C), A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
Закон непротиворечия	$A \wedge \neg A = 0$
Закон исключённого третьего	$A \vee \neg A = 1$
Закон двойного отрицания	$\neg \neg A = A$
Законы де Моргана	$\neg (A \vee B) = \neg A \wedge \neg B, \neg (A \wedge B) = \neg A \vee \neg B$

Программа с циклической структурой

Program sum_sum; {Программа суммы всех двузначных натуральных чисел}

Var a:integer; {описывается переменная, отождествляемая суммой всех двузначных натуральных чисел}

i:integer; {описывается переменная, имеющая смысл счетчика цикла}

Begin

a:=0; {переменной присваивается начальное значение}

for i:=10 **to** 99 **do** a:=a+i; {Срабатывает оператор повторений **for**.

Происходит проверка условия принадлежности счетчика цикла определённому диапазону значений от начального значения 10 до конечного 99, далее выполняется тело цикла, наращивается значение счетчика на единицу, затем снова проверка условия выхода из цикла. В теле цикла происходит суммирование «старой» суммы с следующим числом и результат присваивается снова переменной, обозначающей подсчитываемую сумму}

Writeln('Сумма всех двузначных чисел равна ',b);

End.

Рассмотрим принципы построения программы на языке Pascal на нескольких примерах. В фигурных скобках даны комментарии к программам.

Программа с линейной структурой

Program volume; {Программа вычисления объема шара по заданному радиусу, начало раздела описаний переменных и констант}

Const pi:=3.14; {Вводится вещественная константа, имеющая смысл числа π }

Var R, V:real; {описываются две переменные, имеющие смысл радиуса и объема шара}

Begin {начало раздела операторов}

Writeln('Введите радиус шара'); {Оператор вывода данных, на экран выводится просьба о вводе с клавиатуры значения радиуса шара}

Readln(R); {Оператор ввода данных, в этот момент исполнение программы приостанавливается и она ожидает ввода данных}

$V:=4*pi*R*R*R/3$; {вычисляется значение правой части выражения и срабатывает оператор присваивания переменной V посчитанного значения}

Writeln('Объем шара равен ',V); {результат выводится на экран}

End.

Помимо оператора повторений **for**, з языке программирования Pascal существуют конструкции **repeat ... until <условие>** и **while <условие> do ...**, соответствующие циклам с постусловиями и предусловиями соответственно. Аналогичные конструкции есть и в языке Basic.

Те же программы на языке программирования Basic имеют следующий вид.

Программа с линейной структурой

```
REM Объем шара
pi=3.14
INPUT "Введите радиус шара"; R
V=4*pi*R*R*R/3;
PRINT "Объем шара равен "; V
END
```

Программа с ветвящейся структурой

```
REM Модуль числа
INPUT "Введите число"; a
IF a>=0 THEN
b=a
ELSE
b=-a
END IF
PRINT "Модуль числа равен "; b
b=ABS(a)
PRINT "Сравним результат. Модуль числа
равен "; b
END
```

БИЛЕТ 12

23

1. Программное обеспечение компьютера, состав и структура. Назначение операционной системы. Командное взаимодействие пользователя с компьютером. Графический пользовательский интерфейс.

Последовательность команд, которую выполняет компьютер в процессе исполнения алгоритма, называется **программой**. Все программы, хранящиеся на компьютере, составляют его программное обеспечение. Программы бывают системные и прикладные.

Системное программное обеспечение обеспечивает работу всех устройств и программ компьютера и взаимодействие человека с ним. Такие программы обеспечивают управление памятью, вводом и выводом информации, тестируют работу периферийных устройств. При запуске компьютера начинает работать важнейшая системная программа, называемая операционной системой. Другие системные программы — это драйверы, архиваторы, антивирусные программы.

Операционная система — это совокупность программных средств, обеспечивающих управление аппаратными ресурсами ЭВМ и взаимодействие программных процессов с аппаратурой, другими процессами и пользователем. Операционная система выполняет следующие действия: управление памятью, управление вводом-выводом, управление файловой системой, управление взаимодействием процессов, защиту, учёт использования ресурсов, обработку командного языка. Часто многие отдельные сервисные программы (трансляторы, утилиты) считаются частью операционной системы.

Другой тип программ — **прикладные программы**. Такие программы предоставляют возможность проводить необходимые пользователю прикладные работы: использовать возможности компьютера по выполнению расчетов, созданию текстов и изображений, таблиц. К таким программам относятся текстовые и графические редакторы, электронные таблицы, системы управления базами данных, браузеры, игры. Среди

БИЛЕТ 11

22

1. Основные компоненты компьютера, их функциональное назначение и принципы работы. Программный принцип работы компьютера.

Компьютер — это техническое устройство преобразования, хранения и обработки информации, осуществляющее заданную программную последовательность операций. Основными компонентами компьютера являются устройства ввода, память, процессор и устройства вывода.

Устройства ввода-вывода обеспечивают ввод информации в память компьютера и обратный вывод её в форме, доступной для понимания человека. Основными устройствами ввода являются **клавиатура, манипуляторы** (например, мыши), **сканеры**. Основные устройства вывода: **монитор, принтер**.

Процессор — это центральное устройство, отвечающее за обработку всех данных и координирующее работу всех узлов компьютера. Процессор состоит из арифметико-логического устройства, управляющего устройством, ячеек внутренней памяти. Ссылка на ячейку памяти, происходит по её адресу. Устройство управления обеспечивает передачу адресов ячеек памяти, команд и данных внутри процессора. Арифметико-логическое устройство производит логические и арифметические действия с данными. Основная характеристика процессора — тактовая частота, чем она выше, тем больше производительность компьютера.

Память компьютера бывает внутренняя и внешняя и предназначена для хранения программ и обработки информации. Её основной характеристикой является объём. Внешняя память предназначена для долговременного хранения данных. Её примерами могут служить компакт-диск, карты флэш-памяти, винчестеры. Конструкция внешних носителей памяти предусматривает наличие движущихся механических частей, что влечёт снижение их быстродействия по сравнению с внутренней памятью, которая полностью электронная. Внутренняя память состоит из ОЗУ (оперативного запоминающего устройства) и ПЗУ (постоянного запоминающего устройства). Первое из них хранит данные, используемые в данный момент

БИЛЕТ 13

24

1. Понятие файла и файловой системы организации данных (папка, иерархическая структура, имя файла, тип файла, параметры файла). Основные операции с файлами и папками, выполняемые пользователем. Понятие об архивировании и защите от вирусов.

Файл — это некоторая целостная совокупность данных или программа, имеющая имя и хранящаяся на внешнем носителе. Имя файла состоит из двух частей, разделённых точкой: имя файла и его расширение, позволяющее судить о его типе (программа или данные) и назначении. Пользователь сам выбирает имя файла, а его расширение, как правило, присваивается автоматически программой, в которой файл создается. Для упрощения работы с файлами приняты следующие соответствия некоторых файлов и их расширений:

Расширение	Тип хранящейся информации
doc, txt	Текстовая
bmp, gif, tif, jpg	Графическая
avi	Видеоизображение
wav	Звуковая
bak	Резервная копия (устаревшая версия) файла, сохранённого после внесения исправлений
exe, com	Исполняемый файл, запускающий некоторую программу
xls	Электронная таблица
rar, zip, arj	Архив
dbf	База данных
bas, pas, cp	Текст программы на языке программирования (Бейсик, Паскаль, С)

времени, второе хранит данные, предназначенные для загрузки компьютера (используются только для чтения). При выключении компьютера содержимое ОЗУ пропадает, а ПЗУ сохраняется. Отметим, что при хранении программ и данных используется принцип однородности памяти, введенный фон Нейманом. Это означает, что программы и данные хранятся в одной и той же памяти.

Принципиальная схема устройства компьютера



Программа — последовательность действий (команд), которую должен выполнить компьютер для решения задачи. Программный принцип работы компьютера, состоит в том, что компьютер выполняет действия по заранее заданной программе.

Базовая конфигурация ПК называется минимальный комплект аппаратных средств, достаточный для начала работы с компьютером. Принято считать, что базовыми являются системный блок, монитор (как устройство вывода информации), клавиатура и мышь (как устройства ввода информации). Системный блок имеет, как правило, большое количество разъемов, обеспечивающих доступ ко внутренним устройствам. Через эти разъемы происходит подключение дополнительного периферийного оборудования. В данный момент наиболее популярным и удобным является разъем USB, посредством которого могут быть подключены, например, и принтер, и сканер, и мышь. Внутри системного блока расположена **Материнская плата** — самая большая плата компьютера. Она содержит магистрали, связывающие процессор с оперативной памятью. Из схемы устройства компьютера видно, что обмен данными между периферийными устройствами и процессором идет через магистраль.

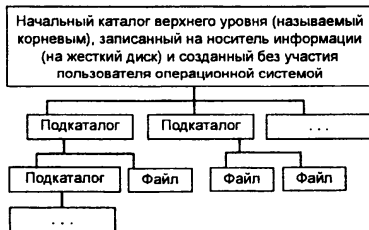
Sys	Системный файл
dll	Системная библиотека
hlp	Файл помощи
bat	Текстовый командный файл

Помимо этого файл характеризуется размером, датой и временем создания. Размер файла — это количество места, занимаемое им на диске (единицы измерения байт, килобайт, мегабайт).

Папка — объект, объединяющий в группу файлы и другие папки. Характеризуется своим именем (без расширения), размером (суммарным объемом всех файлов и папок, находящихся в ней), датой и временем создания.

Каталог — структура данных, обеспечивающая поиск объекта по текстовому имени.

Файловой системой называют совокупность файлов и каталогов, а также управляющей информации для выполнения операций над ними. В современных операционных системах используется многоуровневая иерархическая файловая система, организованная в виде дерева:



Программа с циклической структурой

REM сумма всех двузначных чисел

a=0

FOR i=10 TO 99

a=a+i

NEXT i

PRINT "Сумма всех двузначных чисел равна ", b

END

Современные языки программирования предоставляют возможность работать с процедурами и функциями, т.е. некоторыми подпрограммами оформленными заранее и, возможно, записанными во внешней по отношению к текущей программе, файл (так называемая **модульность языка программирования**). Выполнение процедур и функций происходит по команде пользователя.

Прикладные программы различают отдельно системы программирования, которые сами дают возможность создавать новые программы как прикладные, так и системные.

Для эффективной работы пользователя на ЭВМ ему необходимо инструктировать компьютер о действиях, которые необходимо произвести, т.е. подавать команды. Один из подходов к вводу команд — это **командная строка**, в которую вводится некоторая последовательность знаков, означающих конкретную команду.

Для удобства работы на компьютере составной частью современных операционных систем являются **графические интерфейсы пользователя**. В этом случае сложные команды можно вводить при помощи щелчков мыши на различных областях экрана, а не набирая команду вручную с клавиатуры в командной строке. Основой такого интерфейса являются диалоговые окна, меню и элементы управления (важнейшими из последних являются кнопки). В операционной системе Windows существуют следующие элементы графического интерфейса:

- **рабочий стол** (вид экрана компьютера, при котором на представленном поле расположены значки и иконки, обозначающие программы). На рабочем столе расположены либо сами программы, либо их ярлыки (т.е. некоторые указатели на место расположения самих программ);
- **значки** (некоторое специальное изображение, подобранное для определённого класса программ или файлов с данными);
- в нижней части экрана располагается **панель задач**, содержащая кнопку «Пуск», панель быстрого запуска (для часто используемых программ) и значки открытых окон программ, запущенных на текущий момент;
- **окно Windows** — это часть дисплея, с которой программа работает как с отдельным экраном. Важнейшее свойство окон — возможность изменения размера без нарушения работы самой исполняемой в нём программы.

Основными операциями с файлами и папками, которые может произвести пользователь из системной среды, являются

- копирование,
- перемещение,
- удаление,
- переименование.

Также файл может быть создан или закрыт (открыт) из программной среды.

Хранение и передача по компьютерным сетям файлов во многих случаях требует компактности представления данных, с целью уменьшения объема. Для этого используют программы архиваторы, такие как RAR или ARJ. Они обеспечивают сжатие и упаковку архивируемых данных, а также их восстановление в обратную сторону в случае необходимости. Принцип работы архиваторов — нахождение повторяющихся или однотипных кусков текста или кода и замещение их какой-то более компактной меткой.

Компьютерный вирус — это некоторая специальная программа, выполняющая на компьютере пользователя всевозможные вредные действия и, как правило запускающаяся без его ведома сама по себе. При этом в зависимости от типа вируса может происходить порча или несанкционированная передача посторонним лицам информации, хранящейся в памяти компьютера, исчезновение информации или программ, существенное замедление работы компьютера, некорректная работа некоторых приложений, появление посторонних видео-аудио эффектов, остановка работы компьютера (т.е. его «зависание» или невозможность загрузки).

Для защиты от вирусов применяются специальные **антивирусные программы**, работа которых,

25

БИЛЕТ 14

26

1. Информационные ресурсы общества.

Основы информационной безопасности, этики и права.

Ресурс — это средства, имеющиеся в наличии, к которым можно прибегнуть в случае необходимости. В современном мире все большее и большее значение приобретают **информационные ресурсы**, под которыми понимают совокупность данных, организованных для эффективного получения достоверной информации. Юридически под информационными ресурсами понимается информация, зафиксированная на материальном носителе и хранящаяся в информационных системах (библиотеках, архивах, фондах, банках данных и др.). Свойствами информационных ресурсов являются постоянная пополняемость и возможность многократного использования.

Информационный ресурс может иметь своего обладателя (человека, государство или организацию). Информация, находящаяся в свободном доступе, называется открытой. Информация, доступ к которой ограничен, относится либо к конфиденциальной информации, либо к государственной тайне.

Важнейшей характеристикой информации является её **достоверность**. Получение именно достоверной информации отвечает интересам государственной безопасности. Поэтому информационные ресурсы общества в настоящее время рассматриваются как стратегические ресурсы, аналогичные по значимости ресурсам материальным и сырьевым.

Основными хранилищами информации являются библиотеки (как классические, хранящие бумажные носители, так и современные электронные) и архивы. Они могут служить как справочными, так и образовательными ресурсами.

Хранение и распространение информации в современном мире регулируется государственными законами и нормативными актами. Они, с одной стороны, гарантируют

БИЛЕТ 15

27

1. Технологии работы с текстовыми

документами. Текстовые редакторы и процессоры: назначение и возможности. Основные структурные элементы текстового документа. Шрифты, стили, форматы. Основные приёмы редактирования документа. Встраиваемые объекты. Понятие гипертекста.

Для создания новых документов и внесения изменений в ранее созданные на компьютере используются **текстовые редакторы**, позволяющие создавать, редактировать, форматировать, сохранять и распечатывать документы. Простые текстовые редакторы (например, стандартное приложение Windows Блокнот) позволяют редактировать текст, а также осуществлять простейшее форматирование. Более совершенные текстовые редакторы (например, Microsoft Word), называемые иногда текстовыми процессорами, имеют гораздо более широкий спектр возможностей по созданию документов (вставка списков и таблиц, средства проверки орфографии, сохранение исправлений и много другое).

В профессиональной издательской деятельности человека для подготовки к изданию книг, журналов и газет в процессе макетирования печатного издания используются мощные программы обработки текста — настольные издательские системы (например, Adobe PageMaker, Microsoft

физической страницы, используемые для размещения сопроводительной информации (сносок, колонтитулов и т.п.). Левое и правое поля обычно остаются незаполненными. В верхнем и нижнем полях могут располагаться колонтитулы. **Колонтитулы** представляют собой служебную информацию, размещаемую в верхнем или нижнем поле страницы (фамилия автора, название документа, номер страницы и т.п.). Все, что находится между колонтитулами, принято называть **телом страницы**: текст, подстрочные примечания, рисунки, таблицы.

Вид каждого объекта, составляющего документ, определяется его размером, формой объекта, местоположением в документе относительно других объектов. Размер измеряется по высоте и ширине, при этом существует возможность масштабирования (т.е. выбора масштаба). Местоположение объекта в документе определяется относительно края страницы. Положение объекта относительно других объектов документа задается пользователем.

Кроме перечисленных параметров, каждый объект может иметь другие параметры, доступные для форматирования: шрифтовое оформление; цветовое оформление и узор; толщина, тип. **Шрифт** — внешний вид символов алфавита. Каждую конкретную реализацию шрифта называют **начертанием**. Совокупность всех начертаний называют **гарнитурой** шрифта, которая определяет художественное решение,

свободу распространения открытой информации, а с другой — препятствуют распространению информации конфиденциального характера и информации, относящейся к государственной тайне. Однако, в некоторых случаях юридические правила не гарантируют соблюдения общепринятых моральных и этических норм, что особенно заметно в публикациях в Интернете, где применение юридических норм пока затруднительно.

Правовое регулирование в сфере распространения программного обеспечения в настоящее время обеспечивается покупкой лицензий на использование определенных программ. Использование программы, распространяемой не свободно, без покупки лицензии является преступлением и наказуемо. Однако в настоящее время существует большое количество программ, находящихся в свободном доступе и не требующих платы за пользование ими (так называемые *freeware* программы). Помимо этого можно встретить с демо-версиями лицензионных программных продуктов, также предлагающихся бесплатно. Однако, как правило, такие программы либо имеют ограниченный набор функций по сравнению с оригиналом, либо ограниченный срок работы (например, месяц).

На программные продукты, также как и на литературные произведения, распространяется авторское право. Для информирования о своих авторских правах на программном продукте ставится знак ©, пишется имя правообладателя и год выпуска. Автору программы принадлежат исключительные права на воспроизведение, распространение программ и на её модификацию.

Для предотвращения несанкционированного доступа к информации компьютерный пользователь вправе поставить на свой компьютер пароль, известный только ему. Такой же принцип использования пароля используется при доступе к электронной почте.

как правило, заключается в постоянном мониторинге компьютерной системы на наличие вирусов. Результатом их работы является удаление, «лечение» или помещение «в карантин» зараженных файлов или файлов, которые потенциально могут являться вирусами. Большинство современных антивирусных программ (NOD32, Doctor Web, Symantec Antivirus, Kaspersky) предоставляют пользователю возможность осуществлять проверку на вирусы как постоянно (в режиме реального времени проверять входящие e-mail письма и передаваемые по сети данные, проверять подключаемые внешние носители памяти), так и по запросу самого пользователя, а также выбирать области памяти на диске, которые необходимо проверить. Отметим, что из-за постоянно вновь создающихся вирусов, а также модификации старых, необходимо периодическое обновление антивирусных программ, которое, как правило, осуществляется через Интернет путем скачивания с сайта разработчика обновлённых антивирусных баз.

выделяющее шрифт среди других. Почти все семейства шрифтов имеют обычное (прямое) начертание. Курсивный, полужирный и полужирный курсивный шрифты используют для смыслового выделения текста. *Ке́льз* (т.е. размер шрифта) измеряют в печатных пунктах. Один пункт равен 0,35 мм. Помимо этого шрифт может характеризоваться контрастностью (отношением толщины горизонтальных и горизонтальных штрихов символа), наличием засечек (небольших черточек на концах линий), цветом и некоторыми другими параметрами.

Кроме текстовых символов форматированный текст содержит специальные невидимые коды, которые сообщают программе, как надо его отображать на экране и печатать на принтере: какой шрифт использовать, каким должно быть начертание и размер символов, как оформляются абзацы и заголовки.

При оформлении текстовых документов часто требуется добавлять в документ нетекстовые элементы или объекты. Современные текстовые редакторы позволяют это делать — они имеют широкие возможности по вставке в текст рисунков, диаграмм, формул и т.д.

Во многих электронных документах используется так называемое относительное форматирование. Оно обеспечивает возможность просмотра документа на любом компьютере с заранее не заданной величиной экрана и в любом окне.

Office Publisher), а для подготовки к публикации в Интернете Web-страниц и Web-сайтов — специализированные приложения такие, как Microsoft FrontPage.

Итак, *текстовые редакторы* — это программы, обеспечивающие редактирование и форматирование текстов, программ и документов в соответствии с задаваемыми пользователем командами. *Редактирование* — это процесс внесения изменений в документ, обеспечивающий добавление, удаление, перемещение или исправление содержания документа. *Форматирование* — это процесс представления внешнего вида документа или отдельных его объектов в соответствии с предписанным форматом. При этом под форматом понимают способ расположения и представления данных. В процессе форматирования формируются абзацы, центрируются заголовки, выравниваются поля, происходит разбиение на страницы.

Каждый документ может быть сохранен в файле, отображен на экране или напечатан на бумаге (физической странице). Область физической страницы, в которой размещаются объекты документа, называют логической страницей. Форматирование страницы документа в любой прикладной среде предполагает установку ориентации страницы, полей, колонтитулов. *Ориентация страницы* — это положение листа бумаги в пространстве. Различают книжную и альбомную ориентации. *Поля* — это области

Формат файла — это способ хранения текста в файле. Наиболее часто используемые форматы приведены в таблице.

29

Формат	Краткое описание
TXT	Файл, состоящий из числовых кодов символов текста, не поддерживается возможность хранения таких сложных объектов как таблицы или рисунки
DOC	Файл, состоящий из числовых кодов символов текста и дополнительных кодов команд форматирования, дает возможность вставлять в документ различные объекты, не являющимися текстовыми (рисунки, диаграммы)
RTF	Файл, состоящий из числовых кодов символов текста и дополнительных кодов команд форматирования, сохраняет информацию о форматировании произведенном в документе, но набор его возможностей беднее по сравнению с форматом DOC
HTM, HTML	Файл для хранения Web-страниц и гипертекста

Для того чтобы отформатированный документ выглядел аккуратно, принято сходные объекты оформлять единообразно, т.е. одним шрифтом, с одинаковыми отступами и т.п. Для обеспечения такого единообразия введено понятие стиля.

Для создания картинка в графическом редакторе необходимо выбрать на панели инструментов «инструмент рисования»: условно изображенные картинки, после выбора которых пользователю предоставляется возможность нарисовать линию, если выбрана «линия», прямоугольник, если выбран «прямоугольник», ломаную, если выбран инструмент «ломаная». Наборы инструментов есть и в растровых графических редакторах, и в векторных. Однако в растровом редакторе нарисованный объект становится лишь изображением, тогда как в векторном у объекта целиком может быть в любой момент изменен размер, местоположение, цвет. Для редактирования изображения в графических редакторах существуют инструменты выделения, после чего пользователь может производить копирование, удаление, перемещение выделенных фрагментов. Существуют также возможности текстовых добавления надписей. Основными характеристиками рисуемых объектов являются цвет (контур и заливки), толщина и тип линий. Для более детального рассмотрения объектов используется инструмент «лупа».

Ввод изображений в компьютер возможен не только посредством прямого рисо-

31

БИЛЕТ 16

30

1. Технологии работы с графической информацией. Растровая и векторная графика. Аппаратные средства ввода и вывода графических изображений. Прикладные программы работы с графикой. Графический редактор. Основные инструменты и режимы работы.

Графический редактор — это программа, предназначенная для создания, редактирования и просмотра изображений. В настоящее время существует два принципа хранения изображений: растровый и векторный, используемые в различных графических редакторах. Сравнительный анализ этих двух способов проведен в таблице.

	Векторная графика	Растровая графика
Основные принципы	Основными структурами в векторной графике являются геометрические фигуры. При этом в памяти компьютера хранятся не сами фигуры, а формулы, их описывающие	Изображение формируется из точек (пикселей), в памяти компьютера отдельно для каждого пикселя хранится его цвет

БИЛЕТ 17

32

1. Табличные базы данных (БД): основные понятия (поле, запись, первичный ключ записи); типы данных. Системы управления базами данных и принципы работы с ними. Поиск, удаление и сортировка данных в БД. Условия поиска (логические выражения); порядок и ключи сортировки.

Базы данных (БД) — это определённым образом организованный набор информации из некоторой предметной области, которая упорядочена в виде набора элементов или записей одинаковой структуры. Базы данных относятся к компьютерной технологии сохранения, поиска и сортировки информации. Определение физической и логической структуры базы данных, ввод информации и доступ к ней осуществляется программной системой, называемой **системой управления базой данных** (СУБД). Примером СУБД является Microsoft Office Access.

В зависимости от способа представления данных и отношений между ними база данных может иметь реляционную (табличную), сетевую или иерархическую структуры. Наибольшее распространение получили реляционные базы данных, имеющих табличную форму организации. Главное достоинство таблиц — их понятность, простота и удобство реализации на ЭВМ.

Обычно информация в базах данных хранится не в одной таблице, а в нескольких взаимосвязанных. Каждая таблица имеет уникальное имя, по которому на неё можно сослаться. Столбцы таблицы соответствуют тем или иным характеристикам объекта и называются полями. Каждое поле характеризуется именем и типом хранящихся данных (число, символ, дата, логическое поле). Каждая строка таблицы соответствует одному из объектов. Она называется записью и содержит значения всех полей, характеризующих объект. Каждая строка должна быть уникальной, поэтому одному полю или группе полей отводится роль первичного ключа, т.е. набор их значений уникален в таблице. Если ключом является одно поле, то

Достоинства	Изменение масштаба не приводит к потере качества изображения, также не происходит существенного увеличения объема занимаемой памяти. Экономичные объемы требуемой под изображения памяти. Хорошее качество изображения. Возможность редактирования отдельных элементов изображения	Хорошая передача цвета и полутонов
Недостатки	Плохое качество передачи полутонов цвета. При наличии большого количества объектов, составляющих рисунок, увеличивается время загрузки файла. Не всякий объект природы может быть легко представлен в виде комбинации геометрических фигур	При увеличении размера изображения происходит потеря его качества. Большой объем занимаемого места на носителе
Применение	Для изображений чертежей, схем, диаграмм	Для работы с фотографиями и рисунками
Программы	CorelDraw, Adobe Illustrator	PhotoShop, Paint

Стиль — это совокупность параметров форматирования объекта документа. Простейший стиль, как правило предлагаемый самим текстовым редактором, называется автоформатом. Пользователь по своему желанию может изменять стиль оформления всего текстового документа или его частей по своему усмотрению (например, убрать нумерацию глав, оставив только заголовки, переместить номер страницы из верхнего колонтитула в нижний, задать нумерацию формул с размещением по правому краю страницы).

Для создания Web-страниц и для обеспечения удобства навигации по документам большого объема используется гипертекст. *Гипертекст* — это размеченный текст, содержащий в себе ссылки на другие документы или места документа. В более широком понимании термина, гипертекстом является любая словарь или энциклопедия, где встречаются отсылки к другим частям текста.

ключ простой, если более — то составной. Строки в базах данных хранятся неупорядоченно, их упорядоченный вывод на экран или принтер, осуществляется только по запросу пользователя. При этом результатом работы такого запроса также является таблица, но отсортированная согласно желанию человека. Связь между несколькими таблицами в базе данных осуществляется через поля, имеющие одни и те же значения и смысл. Возможны следующие виды связей: «один-к-одному», «один-ко-многим», «многие-ко-многим».

Важнейшим элементом любой системы управления базами данных является средство ускоренного поиска данных, поскольку поиск — самая распространенная операция при работе с таблицами. Этот механизм обычно реализуется введением так называемых *индексов*. Любая таблица базы данных может быть проиндексирована по нескольким полям и иметь любое число индексов. Индексы (а точнее индексные файлы) содержат информацию о расположении записей базы данных в определенном порядке (хронологическом, по возрастанию, алфавитном). Сортировка может быть произведена сразу по нескольким полям (ключам), в этом случае важен порядок указанных полей (их приоритетность).

Выборка информации из базы данных проводится с помощью запросов — логических выражений, реализованных, как правило, на различных модификациях специального языка SQL — структурированного языка запросов, применяемого для создания, модификации и управления данными в реляционных базах данных. Запросы позволяют проводить выборку, добавление, удаление и редактирование записей по заданным критериям. При формировании запроса над полями баз данных могут производиться определенные действия, соответствующие типу конкретного поля. Критерий выбора формируется в виде логического выражения, имеющего значение либо истина, либо ложь. По результатам работы запроса формируется новая таблица из записей, значения полей которой делают истинным значение этого логического выражения. При формировании логического выражения используются стандартные логические операции (не, или, и), знаки отношений (<, >, =) и встроенные функции преобразования данных, хранящихся в полях таблицы.

вания объекта пользователем, но и при помощи таких устройств как сканер, цифровая камера или цифровой фотоаппарат. В этом случае полученное с данного устройства изображение в последствие, как и любое другое, может быть открыто в графическом редакторе и отредактировано. Для вывода изображений применяются мониторы, принтеры и плоттеры (для вывода широкоформатных рисунков).

1. Технология обработки информации в электронных таблицах (ЭТ). Структура электронной таблицы. Типы данных: числа, формулы, текст. Правила записи формул. Основные встроенные функции. Абсолютные и относительные ссылки. Графическое представление данных.

Электронная таблица — это удобная среда для работы с большими объемами однотипной информации. Наиболее популярной электронной таблицей в России является Excel.

Таблица представляет собой рабочую область, разбитую на столбцы и строки, пересечения которых образуют **ячейки**, в которые записывается информация. Каждая строка имеет свой уникальный числовой номер, каждый столбец — своё уникальное буквенное имя. Для обеспечения обработки содержимого ячеек таблицы каждой ячейке присваивается также уникальный адрес (или имя). Адрес каждой ячейки формируется из номера буквенного имени столбца и номера строки, которым эта ячейка принадлежит (например, запись B7 означает, что это ячейка седьмой строки и столбца с именем «В»). В каждый момент времени активной (т.е. той, над которой производятся какие-либо действия) является только одна ячейка.

Любая ячейка электронной таблицы может содержать числовое значение, текстовую информацию или формулу. Числовые данные, хранящиеся в электронной таблице, могут в зависимости от реализации электронной таблицы иметь различные формы представления (так в Excel принято использовать такие типы, как числовой десятичный, числовой экспоненциальный, денежный, типы даты, время, процент).

Текстовые данные, как правило, отображаются именно так, как были введены и обрабатываются как строки вне зависимости от их содержания.

Формулы в ячейках электронных таблиц позволяют содержать текущей ячейки вычислять, используя значения других ячеек, при этом в формулах используются ссылки на адреса ячеек (т.е. их имена). Ссылки бывают относительные (обозначаются, например, B4) и абсолютные (\$B\$4).

синхронизирована. Пользователи имеют доступ ко всей информации конференции вне зависимости от того, с каким сервером они связываются сами.

- **Справочные службы.**

- **Системы удаленного управления,** обеспечивающие оборот электронных денег, дают возможность совершать электронные платежи и управление банковскими счетами.

- **Файловые архивы** представляют собой серверы, на которых размещаются либо свободно распространяемые программные продукты, либо демоверсии программных продуктов, не находящихся в свободном распространении (последние, как правило, с ограниченным сроком действия или ограниченным набором функций).

- **WWW (World Wide Web)** — это огромное множество серверов в Интернете, содержащих так называемые Web-страницы, созданные на основе гипертекста. Для просмотра такой страницы необходимо подключиться к Интернету, запустить какой-либо браузер (например, Internet Explorer), в строке «адрес» набрать адрес необходимой страницы.

Остановимся подробнее на работе с электронной почтой. Для того чтобы пользователь мог эффективно принимать электронную почту ему необходимо завести почтовый ящик на почтовом сервере, а чтобы отправлять корреспонденцию — знать уникальный электронный адрес адресата.

Почтовый сервер — это некоторый удаленный компьютер, подключенный к сети, обеспечиваю-

1. Основные принципы организации и функционирования компьютерных сетей. Интернет. Информационные ресурсы и сервисы компьютерных сетей: Всемирная паутина, файловые архивы, интерактивное общение. Назначение и возможности электронной почты. Поиск информации в Интернете.

Компьютерная сеть представляет собой объединение информационных ресурсов, которые могут быть удалены друг от друга. Сети бывают:

- **локальные** (объединение компьютеров, расположенных на небольших расстояниях друг от друга, например, в пределах одного дома);
- **региональные** (объединение компьютеров и локальных сетей одного региона, области);
- **корпоративные** (объединение локальных сетей в пределах одной корпорации);
- **глобальные** (объединение компьютеров, расположенных на значительном расстоянии, обеспечивающее доступ к мировым информационным ресурсам).

Формирование сетей стало возможным, в частности, благодаря созданию унифицированных протоколов, т.е. совокупностей правил, регламентирующих формат и процедуры обмена информацией между двумя или несколькими устройствами, а также из-за создания большого количества устройств, способных передавать и получать большие объёмы информации на значительные расстояния. Примером таких устройств

1. Понятие модели. Информационная модель. Виды информационных моделей (на примерах). Реализация информационных моделей на компьютере. Пример применения электронной таблицы в качестве инструмента математического моделирования.

На протяжении всей своей жизни человек постоянно сталкивается с различными копиями (или моделями) предметов и существ из окружающего его мира. Все эти копии не обладают всеми свойствами своего прототипа, но имеют сходные с ним характерные черты. Так детские игрушки передают внешний облик людей, автомобилей, животных, их пропорции, цвет, велосипед имитирует поездку на настоящем велосипеде. Подбор черт, в которых необходимо осуществить схожесть оригинала и его модели зависит от цели создания модели. Например, и географическая карта, и глобус передают очертания береговой линии океанов, границ стран, и там и там обозначены русла рек и озера. Однако сферическая форма глобуса и его крепление в подставке под некоторым углом, передают форму земного шара в целом, а также угол наклона оси планеты к плоскости орбиты. Итак, **модель** — аналог (заместитель) оригинала, отражающий некоторые его характерные черты. Приведенные выше примеры моделей относятся к так называемым **материальным моделям**, так как сами являются материальными объектами.

Другой тип моделей — **абстрактные модели**. К ним относятся математические формулы, чертежи, таблицы, диаграммы. В этом случае может моделироваться не объект, а целый процесс, явление или межличностные отношения. Например, самыми распространенными кинематическими моделями в физике являются модели равномерного и равноускоренного движений, описываемые соответствующими формулами для определения координат и скорости,

являются, например, модемы, передающие сигналы по телефонной линии, или сетевые адаптеры. В данный момент наиболее современными являются устройства, обеспечивающие беспроводной доступ к компьютерным сетям. Помимо этого широкому распространению компьютерных систем способствует появившееся большое количество различных программ (браузеров, почтовых программ, программ обмена короткими сообщениями), поддерживающих возможность работы в сети и облегчающих её для пользователя.

Интернет — это глобальная сеть, пользователям которой предоставлен доступ к мировым информационным ресурсам и сервисам. Базовыми его сервисами являются следующие нижеперечисленные службы.

- **Электронная почта** предоставляет возможность пересылки различного рода сообщений между пользователями сети. Пересылаемая информация может содержаться как в текстовых, так и в аудио-, видеофайлах.

- **Поисковые серверы** осуществляют в автоматическом режиме просмотр информации в сети и создание базы ссылок на неё. По запросу пользователя поисковый сервер предоставляет пользователю адреса интересующих его сайтов. Наиболее популярные в России поисковые серверы Yandex, Google, Rambler.

- **Телеконференции** предоставляют возможность пользователям обмениваться информацией на определённую тему. Для обеспечения работы конференции на ряде серверов выделяются специальные почтовые ящики, работа которых

при этом предлагаемые формулы справедливы не для конкретного тела, а для широкого класса объектов.

При наличии достаточной и точной информации путем представления её в некотором виде, доступном для передачи, хранения, обработки или анализа, можно говорить о создании информационной модели. Таким образом, **информационная модель** — целенаправленно отобранная информация об объекте, представленная в некоторой форме. Возможны следующие формы представления информации: в виде жестов или сигналов, в устной или словесной форме, в виде символов (например, числа или ноты), графически или в виде таблицы.

Представление информационных моделей может быть осуществлено в **образной** или **знаковой** форме. Образная форма представления означает наличие некоторой визуальной информации, зафиксированной на каком-либо носителе. Знаковые информационные модели подразумевают использование некоторого языка записи данных (формулы, языка программирования, таблицы).

Другое представление информационной модели на компьютере — это **электронная таблица**, позволяющая хранить и во многих случаях автоматически обрабатывать большие объёмы данных. Примерами применения таких таблиц могут быть электронный журнал успеваемости учеников в школе, таблицы прихода и расхода денежных средств, используемые в бухгалтерии, таблицы, содержащие сведения о расстоянии между населёнными пунктами, стоимости проезда и расписании движения в транспортных компаниях. Современные таблицы также позволяют ученым по экспериментальным данным (например, значению леремантовой времени и текущей координате) определить траекторию полета воздушного судна, определить предполагаемую его скорость и местоположение в некоторой промежуточной точке, не вошедшей в статистическую выборку.

Относительная ссылка имеет смысл смещения ячейки, из которой необходимо подставить в формулу данные, относительно активной ячейки. При копировании формулы такие ссылки автоматически пересчитываются, сохраняя постоянным относительное расположение источника данных по отношению к ячейке, их принимающей и использующей. Абсолютная ссылка, в отличие от относительной, ссылается на конкретную ячейку в таблице и не меняется при перемещении самой формулы. Результат, отображающийся в ячейке, имеющей тип «формула», при изменении данных в ячейках, на которые производятся ссылки, пересчитывается автоматически.

Современные реализации электронных таблиц позволяют производить сортировку данных (в алфавитном порядке — для текстового формата ячеек, по возрастанию и убыванию — для числового), сортировка может производиться сразу по нескольким столбцам, но в этом случае необходимо задать приоритетность сортировки по каждому из столбцов или строк.

Электронные таблицы позволяют осуществлять поиск данных в больших объёмах информации, занесённых в них. Для осуществления поиска пользователь указывает условие или ряд условий, которым должны удовлетворять найденные данные. Такие запросы носят название фильтров и формируются при помощи специальных условий поиска, которыми могут служить математические отношения $<$, $>$, $=$ и логические операции (И, ИЛИ, НЕ).

Помимо этого, большинство современных электронных таблиц позволяет пользователю менять формат вывода данных в ячейках на экран, применяя к числовым данным математические операции (как элементарные, так и операции высшей математики), получать статистические данные, искать максимальные и минимальные значения. Удобными и современными средствами, предоставляемыми электронными таблицами являются возможности создания диаграмм и таблиц по результатам анализа значений, находящихся в ячейках, из заданного диапазона столбцов и строк. Помимо этого следует отметить, что эффективность использования электронных таблиц особенно ощутима при работе с большими объёмами данных, которые можно вводить не вручную, а загружать из сформированных заранее простейших файлов.

Почтовый приём, отправление и хранение электронных писем. **Почтовый ящик** — это выделенная часть дискового пространства на почтовом сервере для хранения электронной почты адресата. Для доступа к корреспонденции, хранящейся в почтовом ящике, необходимо знать его имя и пароль.

Электронный адрес строится следующим образом: `username@hostname.domain`. Здесь `username` — сетевой псевдоним пользователя (или логин), `hostname` — имя почтового сервера («хоста»), `domain` — доменный адрес (обозначает группу ресурсов, находящихся в управлении одного «узла» в сети). Домены, как правило, бывают либо административными (`com` — коммерческая организация, `edu` — образовательное учреждение, `net` — сетевая организация), либо географическими (`ru` — Россия, `de` — Германия, `us` — США). P

Справочное издание

Чуркина Татьяна Евгеньевна

ИНФОРМАТИКА

**Ответы на экзаменационные билеты
9 класс**

Издательство «ЭКЗАМЕН»

Гигиенический сертификат
№ РОСС RU. АЕ51. Н 16054 от 28.02.2012 г.

Главный редактор *Л.Д. Лаппо*
Редактор *Г.А. Лонцова*
Корректор *О.А. Андрейчик*
Дизайн обложки *М.Н. Ершова*
Компьютерная верстка *Е.Ю. Лысова*

105066, Москва, ул. Нижняя Красносельская, д. 35, стр. 1.
www.examen.biz

Е-mail: по общим вопросам: info@examen.biz;
по вопросам реализации: sale@examen.biz
тел./факс 641-00-30 (многоканальный)

Общероссийский классификатор продукции
ОК 005-93, том 2; 953005 — книги, брошюры, литература учебная

ОАО «Владимирская книжная типография»
600000, г. Владимир, Октябрьский проспект, д. 7.
Качество печати соответствует качеству предоставленных диапозитивов

**По вопросам реализации обращаться по тел.:
641-00-30 (многоканальный).**