

Примерные ответы на теоретические вопросы билетов

Билет 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

В настоящее время понятие «информация» стало очень популярным. Смысл, вкладываемый в него, сильно расширился и изменился. Нет, пожалуй, такой области человеческой деятельности, где это понятие не используется. Это доказывает возрастающую роль информации в жизни и деятельности людей. В то же время однозначного определения этого понятия пока не существует. Все подходы к определению информации имеют право на существование, они исследуются в различных науках.

Философы под информацией понимают *отраженное многообразие*, т. е. отражение в сознании человека происходящих вокруг изменений. Пример информации: *пришла зима*. В технических науках информацией считают *коды, знаки и сигналы*, которые можно передавать и получать с помощью технических устройств, при этом смысл сообщения значения не имеет. Пример информации: *00111000110011*. Смысл информации важен для журналистов, генетиков, биологов. Пример информации: *открытие генома человека*. В теории информации (по К. Шеннону) информацией считают *сведения, снимающие полностью или частично существующую неопределенность знания*. Пример информации: *из двух возможных вариантов падения монеты выпала «решка»*.

В информатике информацию можно рассматривать как продукт взаимодействия данных и методов их обработки, адекватных решаемой задаче [1]. Пример информации: *компьютерная программа*.

Существует множество классификаций видов информации по разным основаниям. Первое основание классификации — по способу восприятия информации человеком:

- визуальная;
- аудиальная (слуховая);
- тактильная;
- обонятельная;
- вкусовая.

Информацию необходимо как-то представить. Отсюда классификация по способу представления:

- текстовая;
- числовая;
- графическая;
- звуковая;
- комбинированная.

Информация необходима не только отдельному человеку, но и коллективам людей и обществу в целом, поэтому существует еще одна классификация информации — по общественному значению:

- личная;
- специальная;
- массовая.

Человек научился представлять информацию с помощью сигналов и знаков, создал множество технических устройств для работы с этой информацией.

В информатике и смежных науках информация рассматривается, прежде всего, в системе «источник — канал связи — приемник», и для функционирования системы ключевым является обмен информацией посредством передачи сигналов.

Сигналы делятся на аналоговые (значения их параметров изменяются непрерывно) и дискретные (параметры принимают значения из конечного набора). Эта классификация интересует

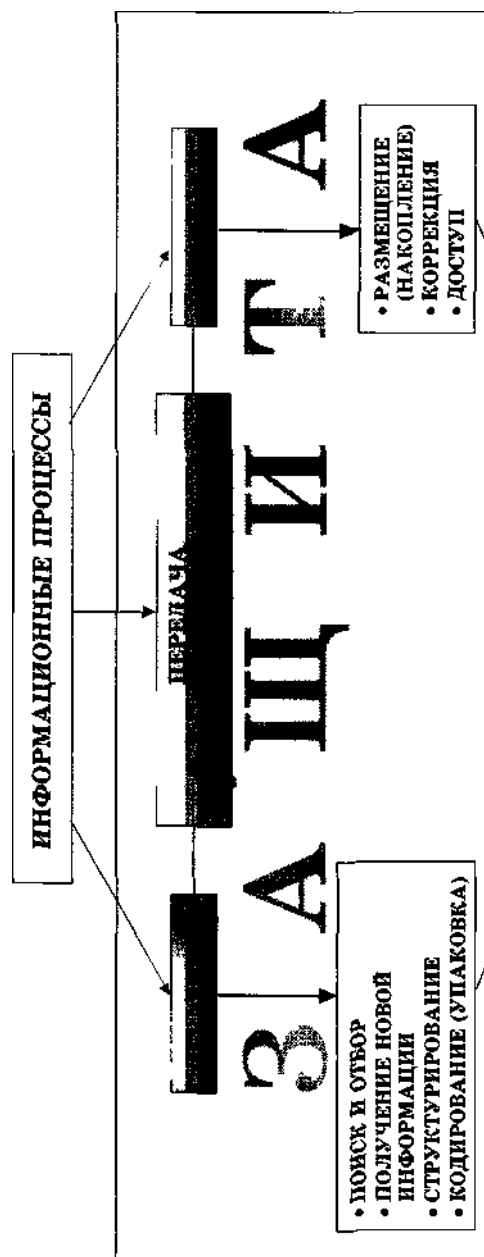
нас с точки зрения возможности кодирования сигналов (цветовых, звуковых) для их обработки с помощью компьютера. Большинство аналоговых сигналов понятны человеку, поскольку он их воспринимает органами чувств, которые к этому приспособлены. Но информации становится настолько много, что человеку приходится пользоваться мощным инструментом — компьютером. Компьютер же хранит, обрабатывает и передает информацию в дискретной (цифровой) форме.

Любые знаки являются элементами какого-либо языка. Язык является средством представления информации.

С точки зрения информатики, языки можно разделить на **естественные** и **формальные**. Естественные языки — это языки общения народов, имеющие устную и письменную форму. Они возникли в результате исторического развития человеческого общества. Формальные языки представления информации были созданы людьми в процессе их деятельности. Это системы обозначений для определенных областей знаний (нотная грамота, азбука Морзе, формулы по математике, физике, химии, шифры и пр.). Любой язык состоит из трех составляющих: набора кодов (алфавита), синтаксиса и семантики. *Алфавитом* называют используемый для кодирования конечный набор отличных друг от друга знаков, расположенных в определенном порядке, *синтаксисом* — правила построения информационной конструкции, а *семантикой* — смысловое содержание информационной конструкции (информационной конструкцией удобно называть сообщение, содержащее информацию).

Информационный процесс в учебниках информатики определяется как совокупность действий, проводимых над информацией для получения какого-либо результата (другой информации). В настоящее время выделены типы информационных процессов, общие для различных систем: обработка, передача, хранение информации. В информатике говорят об общих закономерностях, присущих информационным процессам, происходящим в любой системе: технической (компьютере), биологической (живой природе) или социальной (общественной жизни).

Рассмотрим информационные процессы с точки зрения деятельности человека (схема на с. 42).



Процесс **обработки информации** может представлять собой:

- **Поиск информации** в различных источниках и **отбор**. Поиск информации также можно считать процессом обработки, поскольку при его осуществлении, независимо от того, вручную это производится или с помощью компьютера, происходит процесс сопоставления имеющейся (найденной) информации с требуемой в соответствии с определенными критериями поиска. По такому же принципу происходит отбор необходимой информации.
- **Получение новой информации**. При решении задач любой дисциплины человек, обрабатывая имеющиеся исходные данные в соответствии с требуемым результатом, получает некоторую новую информацию. Получение новой по содержанию информации из исходной возможно путем как математических вычислений, так и логических рассуждений.
- **Структурирование информации**, т. е. изменение формы информации без изменения ее содержания. Если процесс обработки информации связан с тем, что изменяется только форма представления, но не содержание, то говорят, что происходит упорядочивание, систематизация или структурирование информации.
- **Кодирование** (в частности, **упаковка**) информации. В настоящее время достаточно распространен процесс кодирования, т. е. преобразования информации из одной знаковой формы в другую, удобную для ее обработки, хранения или передачи. К этой деятельности можно отнести упаковку (*архивирование*), шифрование с использованием различных алгоритмов.

Процесс **передачи информации** представляет собой создание копии информации на расстоянии от исходного места хранения. В процессе передачи информации обязательно участвуют **источник** и **приемник информации**. Между ними действует **канал связи**. В процессе передачи информация может частично теряться или искажаться. На устранение этого, в частности, могут быть направлены методы **защиты** при передаче информации. Еще при передаче должна обеспечиваться **информационная безопасность**. Передача информации в социальных, биологических и технических системах, с точки

зрения информатики, осуществляется по общей схеме: источник—канал—приемник. Различие в том, что в таких системах понимают под информацией. «В социальных науках под информацией понимают сведения, данные понятия, отраженные в нашем сознании и изменяющие наши представления о реальном мире. Эту информацию, передающуюся в человеческом обществе и участвующую в формировании общественного сознания, называют социальной информацией. Инженеры, биологи, генетики, психологи отождествляют информацию с теми сигналами, импульсами, кодами, которые наблюдают в технических и биологических системах. Содержание принимаемых и обрабатываемых сигналов инженера не интересует» [1], а генетиков и биологов может интересовать.

К процессу **хранения информации** можно отнести:

- **Размещение (накопление).** Информация, полученная в результате поиска, размещается на каком-либо носителе информации, происходит ее накопление. Процесс, в результате которого информация оказывается на носителе в виде, пригодном для последующего извлечения, называется размещением. Таким образом, мы создаем некоторый **информационный ресурс**. Основное отличие информационных ресурсов от других видов ресурсов состоит в том, что информация после использования не исчезает. Поэтому важнейшей задачей является создание таких хранилищ, которые совмещали бы процессы защиты, структурирования, поиска, извлечения, передачи в автоматическом режиме для увеличения доступности информации.
- **Коррекцию.** Информация в хранилищах нуждается в коррекции по различным причинам, таким как: механические повреждения или изменения свойств носителя, устаревание информации, модернизация структуры для оптимизации доступа к информации и пр. С этой целью выполняется процесс коррекции информации.
- **Доступ.** Организация оптимального доступа к различной по ценности информации с использованием процедур защиты от несанкционированного доступа может быть отнесена к процессу хранения [4].

Билет 2

Измерение информации: содержательный и алфавитный подходы. Единицы измерения информации

Известно, что для измерения чего-либо необходимо ввести единицу измерения. Минимальная единица измерения информации — **бит**.

Вероятностный, или содержательный подход

Попытаться объяснить данный подход можно, допустив, что для каждого человека можно условно выделить (например, в виде окружности) область его знания. Всё, что будет находиться за пределами окружности, можно назвать информационной неопределенностью. Постепенно, в процессе обучения или иной деятельности происходит переход от незнания к знанию, т. е. неопределенность уменьшается. Именно такой подход к информации как мере уменьшения неопределенности знания позволяет ее количественно оценить (измерить).

Сообщение, уменьшающее неопределенность знания в 2 раза, несет один бит информации.

Например: при подбрасывании монеты может выпасть либо «орел», либо «решка». Это два возможных события. Они равновероятны. Сообщение о том, что произошло одно из двух равновероятных событий (например, выпала «решка»), уменьшает неопределенность нашего знания (перед броском монеты) в два раза.

Минимальное количество событий для выбора — два (иначе нет выбора), поэтому бит — минимальная единица информации.

Математики рассматривают идеальный вариант, что возможные события равновероятны. Если даже события неравновероятны, то возможен подсчет вероятности выпадения каждого события.

Под неопределенностью знания здесь понимают количество возможных событий, их может быть больше, чем два.

Например, количество оценок, которые может получить студент на экзамене, равно четырем. Сколько информации содержится в сообщении о том, что он получил «4»? Рас-

суждая, с опорой на приведенное выше определение, можем сказать, что если сообщение об одном из двух возможных событий несет 1 бит информации, то выбор одного из четырех возможных событий несет 2 бита информации. Можно прийти к такому выводу, пользуясь методом половинного деления. Сколько вопросов необходимо задать, чтобы выяснить необходимое, столько битов и содержит сообщение. Вопросы должны быть сформулированы так, чтобы на них можно было ответить «да» или «нет», тогда каждый из них будет уменьшать количество возможных событий в 2 раза.

Очевидна связь количества возможных равновероятных событий и количества информации:

$$\text{Количество событий} = 2^{\text{Количество информации (в битах)}}$$

Заполним по формуле таблицу:

Количество битов	0	1	2	3	4	5	6	7	8	9	10
Количество событий	1	2	4	8	16	32	64	128	256	512	1024

Чтобы пользоваться рассмотренным подходом, необходимо вникать в содержание сообщения.

Алфавитный подход к измерению информации

Этот подход основан на подсчете числа символов в сообщении. Он не связывает количество информации с содержанием сообщения, позволяет реализовать передачу, хранение и обработку информации с помощью технических устройств.

Алфавит любого языка включает в себя конечный набор символов. Исходя из вероятностного подхода к определению количества информации, появление знаков алфавита в тексте можно рассматривать как различные возможные собы-

тия. Количество таких событий (знаков) N называют мощностью алфавита. Количество информации i , которую несет каждый из N знаков, согласно вероятностному подходу, определяется из формулы:

$$2^i = N.$$

Остается подсчитать количество символов в тексте из k символов:

$$i = k \cdot \log_2 N.$$

Алфавитный подход является объективным способом измерения информации и используется в технических устройствах.

Переход к более крупным единицам измерения

Ограничения на максимальную мощность алфавита не существует, но есть алфавит, который можно считать достаточным (на современном этапе) для работы с информацией как для человека, так и для технических устройств. Он включает в себя: латинский алфавит, алфавит языка страны, числа, спецсимволы — всего около 200 знаков. По приведенной выше таблице можно сделать вывод, что 7 битов информации недостаточно, требуется 8 битов, чтобы закодировать любой символ такого алфавита, $256 = 2^8$ битов образуют 1 байт. То есть для кодирования символа компьютерного алфавита используется 1 байт. Укрупнение единиц измерения информации аналогично применяемому в физике — используют приставки «кило», «мега», «гига». При этом следует помнить, что основание не 10, а 2. 1 килобайт (Кб) — это $2^{10} = 1024$ байтов, 1 мегабайт (Мб) = 2^{10} Кб = 2^{20} байтов и т. д.

$$1 \text{ Кбайт} = 2^{10} \text{ байт} = 1024 \text{ байт};$$

$$1 \text{ Мбайт} = 2^{10} \text{ Кбайт} = 1024 \text{ Кбайт} = 2^{20} \text{ байт};$$

$$1 \text{ Гбайт} = 2^{10} \text{ Мбайт} = 1024 \text{ Мбайт} = 2^{30} \text{ байт};$$

$$1 \text{ Терабайт} = 2^{10} \text{ Гбайт} = 1024 \text{ Гбайт} = 2^{40} \text{ байт};$$

$$1 \text{ Петабайт} = 2^{10} \text{ Тбайт} = 1024 \text{ Тбайт} = 2^{50} \text{ байт};$$

$$1 \text{ Эксабайт} = 2^{10} \text{ Пбайт} = 1024 \text{ Пбайт} = 2^{60} \text{ байт}.$$

Билет 3

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

Многие годы люди работали с информацией «вручную», прежде чем был изобретен компьютер, позволяющий автоматизировать процессы обработки, передачи и хранения информации. Любая информация (данные) в компьютере представлена дискретно — последовательностью отделенных друг от друга элементов. Значит, информацию для компьютера необходимо закодировать. Кодирование — это преобразование информации из одной знаковой формы в другую, удобную для ее обработки, хранения или передачи. Используемый для кодирования конечный набор знаков называют алфавитом. Кодирование осуществляется по принятым правилам. Правило кодирования называется **кодом** (от французского code — кодекс, свод законов). Длина кода — количество знаков алфавита, используемых для кодирования.

При кодировании информации для технических устройств важное значение имеют алфавиты, состоящие всего из двух знаков. Такие алфавиты называют двоичными. Они наиболее просты для кодирования. Чем меньше знаков в алфавите, тем проще устроена «машина» для распознавания (дешифрования) информационного сообщения. Однако чем меньше знаков в алфавите, тем большее их количество требуется для кодирования, следовательно, больше длина кода.

Легко рассчитать количество M различных сообщений, которые можно закодировать, используя код постоянной длины n и алфавит из R знаков: $M = Rn$. Если мы используем двоичный алфавит, то $M = 2^n$.

При конструировании компьютеров был выбран двоичный алфавит $\{0, 1\}$, что позволило использовать достаточно простые устройства для представления и автоматического распознавания программ и данных. Именно простота сделала этот принцип кодирования таким распространенным. Наряду с этим свойством двоичное кодирование обеспечивает удобство физической реализации, универсальность представления любого вида информации, уменьшение избыточности сообщения, обеспечение защиты от случайных искажений или нежелательного доступа.

Для совместимости компьютеров при обработке текстовой информации принят международный стандарт кодирования символов — код ASCII (American Standard Code for Information Interchange), который устанавливает соответствие между символами и их порядковыми номерами в компьютерном алфавите. В таблице ASCII для кодирования одного символа используется 1 байт (8 битов). Стандартными являются первые 128 символов (0-127), сюда входят буквы латинского алфавита, цифры, знаки препинания, спецсимволы и управляющие коды или операции (0-32). Остальные символы (128-255) используются для кодирования национальных алфавитов, научных символов и символов псевдографики. С 1997 года введен новый стандарт Unicode, где под каждый символ отводится 2 байта.

Для подсчета информационного объема текста пользуются алфавитным способом измерения информации. Принимая, что каждый символ занимает 1 байт информации (при мощности алфавита 256), для определения объема текста необходимо подсчитать количество символов в нем.

Билет 4

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

Вся информация в компьютере кодируется двоичными числами, в том числе графическая, а также звук и видео. Рассмотрим, как создается модель **изображения**, годная для обработки компьютером. Разобьем картинку вертикальными и горизонтальными линиями на маленькие прямоугольники. Полученный двумерный массив прямоугольников называется **растром**, а сами прямоугольники — **элементами растра**, или **пикселями** (это слово произошло от английского picture's element - элемент картинки). Далее кодируем числами цвета пикселей. Перечислим по порядку (например, слева направо и сверху вниз) коды цветов пикселей. Получим представление (код) картинки в компьютере.

Разумеется, часть информации о картинке при таком кодировании потеряется. Потери будут тем меньше, чем мельче прямоугольники и чем точнее закодирован цвет каждого из них.

Рассмотрим, как же кодировать цвет элемента изображения. Во-первых, в понятие «цвет элемента» включается и его яркость. Во-вторых, для единообразия говорят и о цветах черно-белого изображения. В этом случае цвет (оттенок серого цвета) просто сводится к яркости.

Кодирование черно-белых (на самом деле серых полутоновых) изображений

Яркость описывается числом. Для кодирования яркости пикселей отводятся ячейки фиксированного размера, чаще всего от 1 до 8 битов; черный цвет кодируется нулем, а чисто белый — максимальным числом N , которое может быть записано в ячейку. Для одноразрядной ячейки $N = 1$, а для 8-разрядной $N = 255$. Для практических приложений 8-разрядных ячеек вполне достаточно (человеческий глаз в состоянии различить не более одной-двух сотен разных оттенков серого цвета) [3].

Кодирование цветных изображений — метод RGB

Кодирование цветных изображений сложнее. Человеческий глаз различает огромное количество разных цветов и оттенков. Воспринимаемый цвет представляют в компьютере как сумму трех цветов — красного, зеленого и синего с различной яркостью каждого из них. Например, сиреневый цвет — это сумма красного и синего, желтый цвет — сумма красного и зеленого, оранжевый — тоже сумма красного и зеленого, но в другой пропорции. Поэтому цвет пикселя кодируется тремя числами — яркостью его красной, зеленой и синей составляющих.

Этот способ кодирования цветов называется RGB — по первым буквам английских слов Red, Green, Blue — красный, зеленый, синий.

Кодирование звука

По своей природе **звук** является непрерывным сигналом. Для кодирования звука надо непрерывный сигнал превратить в дискретный — последовательность нулей и единиц. Делают это следующим образом.

С помощью микрофона звук превращается в колебания электрического тока. Для этого измеряют амплитуду колебаний через равные *промежутки* времени (на практике не-

сколько десятков тысяч раз в секунду). Каждое измерение фиксируется с ограниченной точностью и записывается в двоичном виде. Этот процесс называется дискретизацией.

Устройство для выполнения дискретизации называется аналого-цифровым преобразователем (АЦП). АЦП измеряет электрическое напряжение в каком-то диапазоне и выдает ответ в виде многоразрядного двоичного числа. Например, типичный 8-битовый АЦП преобразует напряжения в диапазоне $[-500 \text{ мВ}, 500 \text{ мВ}]$ в 8-разрядные двоичные числа в диапазоне $[-127, +127]$.

Воспроизведение закодированного таким образом звука производится при помощи цифро-аналогового преобразователя (ЦАП). Двоичные числа, кодирующие звук, подаются на вход ЦАП с точно такой же частотой, как и при дискретизации, и ЦАП преобразует их в электрические напряжения обратно тому, как это делал АЦП. Например, двоичные числа из диапазона $[-127, +127]$ преобразуются в напряжения из диапазона $[-500 \text{ мВ}, 500 \text{ мВ}]$. Полученный на выходе ЦАП ступенчатый сигнал сначала сглаживается с помощью аналогового фильтра, а затем преобразуется в звук при помощи усилителя и динамика.

На качество последующего воспроизведения звука влияют в основном два параметра: **частота дискретизации** — количество измерений в секунду (измеряется в герцах — Гц) и ее **разрешение** — размер ячейки, отводимый под запись значения амплитуды.

Например, при записи на компакт-диски (CD) используются 16-разрядные значения, а частота дискретизации равна 44 032 Гц. Эти параметры обеспечивают прекрасное качество звучания речи и музыки.

Выбор частоты дискретизации объясняется тем, что максимальная частота звука, который может слышать человек, не превосходит 22 кГц. Чтобы удержать при дискретизации информацию о колебании в 22 кГц, на каждом периоде должно записываться по крайней мере два значения. То есть нужна вдвое большая частота дискретизации, а именно 44 кГц. Эта частота обеспечивает запись любых слышимых человеком звуков. В тех случаях, когда столь высокое качество не требуется, можно использовать меньшие частоты дискретизации: 11 кГц, 5,5 кГц и т. д. Чтобы частоты, получаемые последовательным делением исходной частоты

вдвое, оказались целыми, удобно взять исходную частоту в виде произведения целого числа на степень двойки. Этим и объясняется выбор частоты $17 \cdot 2^8 = 44032$ Гц, с которой кодируется звук на аудио-CD.

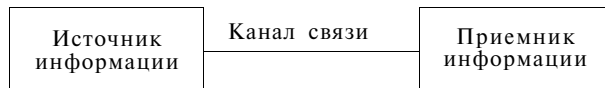
Однако во многих случаях качество CD не требуется. Для записи и передачи речи достаточна частота дискретизации 8 кГц. Несмотря на то, что составляющие человеческого голоса с частотой свыше 4 кГц не могут быть зарегистрированы при такой частоте дискретизации, закодированную речь легко понять [3].

В настоящее время широкое распространение получают **мультимедийные** компьютерные системы, основанные на возможности анимировать графическое изображение и интегрировать его *со звуком, текстом* и прочими эффектами (англ. multi — много, media — среда). Такие системы являются интерактивными, поскольку обеспечивают возможность моделирования реальной среды, событий с участием пользователя. Поэтому сферы применения мультимедийных продуктов разнообразны: от научных исследований, обучения до простого досугового использования.

Билет 5

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

Процесс **передачи** информации представляет собой создание копии информации на расстоянии от исходного места хранения. В процессе передачи информации обязательно участвуют **источник** и **приемник информации**. Между ними действует **канал связи**.

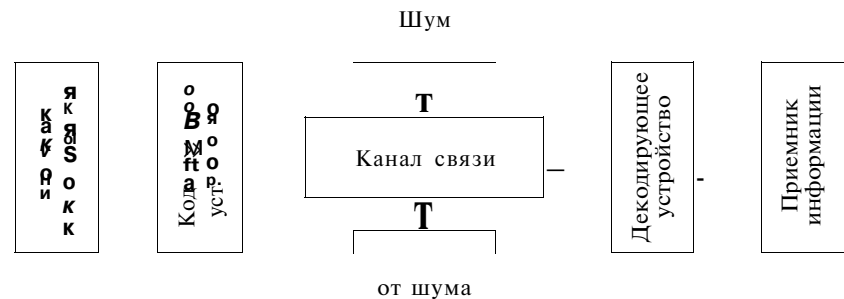


Передача информации в социальных, биологических и технических системах с точки зрения информатики осуществляется по общей схеме: источник—канал—приемник. Различие в том, что в таких системах понимают под инфор-

мацией. «В социальных науках под информацией понимают сведения, данные понятия, отраженные в нашем сознании и изменяющие наши представления о реальном мире. Эту информацию, передающуюся в человеческом обществе и участвующую в формировании общественного сознания, называют социальной информацией. Инженеры, биологи, генетики, психологи отождествляют информацию с теми сигналами, импульсами, кодами, которые наблюдают в технических и биологических системах. Содержание принимаемых и обрабатываемых сигналов инженера не интересует», а генетиков и биологов может интересовать [1].

В процессе передачи информация может частично теряться или искажаться. На устранение этого, в частности, могут быть направлены методы защиты при передаче информации.

Схема процесса передачи информации по техническим каналам связи по К. Шеннону выглядит следующим образом:



Современная компьютерная телекоммуникационная сеть (как пример технической системы) представляет собой комплекс узлов и каналов связи — аппаратуры и программ, обеспечивающих прием и передачу данных.

Комплекс из линии связи и устройств, передающих и принимающих информацию, называется каналом связи. Существует несколько типов линий связи между компьютерами в сети: соединение с помощью кабелей — телефонных, электрических, оптоволоконных и радиосвязь. Каждая из линий связи имеет свои преимущества и недостатки.

К основным характеристикам любого канала связи относят:

- **пропускную способность.** Объем информации, переданный в единицу времени, называется *скоростью передачи информации*. Она измеряется в бит/с, байт/с и т. д. Максимально допустимая скорость передачи информации называется *пропускной способностью* канала связи. В некоторых случаях указывают два числа — пропускную способность при приеме и при передаче данных;
- **надежность канала.** Под надежностью канала понимают вероятность возникновения ошибки при передаче данных. Чем меньше вероятность, тем надежнее канал;
- **максимальную дальность.** В зависимости от используемой технологии передачи данных канал связи может иметь некоторую максимальную длину. Например, в сети, построенной по стандартам FastEthernet, максимальная длина медного кабеля между узлом и устройством коммутации сети — 100 метров.

Для организации обмена информацией в сети требуется ряд аппаратных средств, позволяющих организовать специальные или использовать уже существующие линии связи для приема и передачи цифровой информации.

При использовании аналоговой линии связи (например, телефонной) для обмена данными требуется устройство *модем* (*модулятор—дежодулятор*), выполняющее преобразование цифровых сигналов в аналоговые и наоборот. Пример такого устройства — модем для коммутируемой телефонной линии.

При использовании цифровой линии связи (например, специализированной компьютерной сети) у компьютера должно быть устройство-адаптер, выполняющее преобразование кодов в стандарт, используемый сетью. Примером такого устройства может служить адаптер для подключения к локальной сети стандарта Ethernet.

Основными характеристиками технологий обмена данными, применяемыми при создании сетей, являются характеристики максимальной пропускной способности, количества объединяемых в сеть компьютеров и максимального расстояния, на котором возможен обмен данными [5].

Билет 6

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

Слово «алгоритм» (algorithm) происходит от имени выдающегося ученого IX века Мухаммада ибн Мусы ал-Хорезми (в переводе с арабского Мухаммад, сын Мусы из Хорезма). По латинскому переводу его труда (XII век) Западная Европа познакомилась с десятичной позиционной системой счисления и правилами (algorismi) выполнения в ней арифметических действий.

Несмотря на то, что понятие «алгоритм» давно и прочно вошло в употребление, его определение различается в зависимости от сферы деятельности, где оно используется. В своей деятельности, в частности в сфере обработки информации, человек сталкивается с различными способами или методами решения задач. Они определяют порядок выполнения действий для получения желаемого результата — мы можем трактовать это как первоначальное или интуитивное определение алгоритма.

В школе используются следующие определения алгоритма:

1. Алгоритм — это понятное и точное указание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

2. Алгоритм — это конечное предписание на некотором языке, задающее конечную последовательность выполнимых элементарных операций для решения задачи, общее для класса возможных исходных данных.

Таким образом, формально каждый **алгоритм** — это правила, описывающие процесс преобразования исходных данных в необходимый результат.

Алгоритм предполагает наличие **исполнителя** — человека или технического устройства (автомат, робот, компьютер) со строго определенным набором возможных команд. Совокупность команд, которые могут быть выполнены исполнителем, называется **системой команд исполнителя** (СКИ).

Для того чтобы произвольное описание последовательности действий было алгоритмом, оно должно обладать следующими свойствами.

Свойства алгоритмов

- **Дискретность.** Алгоритм должен состоять из последовательных команд, только выполнив одну команду, исполнитель может приступить к выполнению следующей, т. е. структура алгоритма является дискретной (прерывной).
- **Конечность.** Алгоритм должен содержать конечное количество элементарных выполнимых предписаний, т. е. удовлетворять требованию конечности записи. Исполнитель алгоритма должен выполнять конечное количество шагов при решении задачи, т. е. алгоритм должен удовлетворять требованию конечности действий.
- **Точность (определенность).** Каждая команда алгоритма должна определять однозначное действие исполнителя. Этим свойством часто не обладают предписания и инструкции, которые составляются для людей.
- **Понятность.** Каждая команда алгоритма должна быть понятна исполнителю, т. е. должны использоваться только команды СКИ. Алгоритм не рассчитан на принятие самостоятельных решений исполнителем, не предусмотренных составителем алгоритма.
- **Универсальность (массовость).** Алгоритм должен быть универсальным для некоторого класса однотипных задач.

Из перечисленных свойств вытекает возможность формального исполнения алгоритма, а из нее — очень важное следствие: поскольку осознавать содержание алгоритма не требуется, его исполнение вполне можно доверить автомату или компьютеру. Таким образом, составление алгоритма является обязательным этапом автоматизации любого процесса.

Однозначно определенных способов записи алгоритмов не существует. Различные авторы выделяют разные способы, например:

- словесный (словами устно или письменно);
- словесно-формульный (с использованием в основном формул со словесными комментариями);
- на учебном алгоритмическом языке (псевдокоде, языке учебных исполнителей);

- графический (блок-схема) (с использованием картинок, условных обозначений или блоков);
- на языке программирования высокого уровня (с использованием команд, понятных процессору компьютера).

Словесное описание. Применяется для описания несложных алгоритмов, иначе описание становится слишком громоздким.

Пример. Представим этим способом алгоритм нахождения наибольшего общего делителя двух чисел M и N (алгоритм Евклида).

$$\begin{aligned} &\text{НОД}(M - N, N) \text{ при } M > N, \\ &\text{НОД}(M, N) = \begin{cases} M & \text{при } M = N, \\ \text{НОД}(N - M, M) & \text{при } M < N. \end{cases} \end{aligned}$$

Словесное описание алгоритма Евклида:

1. Если $M > N$, то перейти к п. 4, иначе перейти к п. 2. '•
2. Если $M < N$, то перейти к п. 5, иначе перейти к п. 3.
3. Считать, что $\text{НОД}(M, N) = M$. Конец.
4. Из M вычесть N и впредь считать, что эта разность является значением M . Возвратиться к п. 1.
5. Из N вычесть M и впредь считать, что эта разность является значением N . Возвратиться к п. 1.

Описание на учебном алгоритмическом языке — это описание с помощью слов естественного языка, но в специальной форме, отображающей структуру алгоритма.

Пример

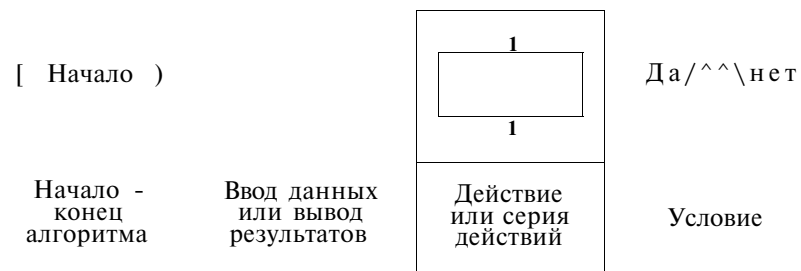
```

начало
    нц пока M ^ N
        если M > N
            то M := M - N
            иначе N := N - M
        все
    кц
    Z := M
конец

```

Описание в графической форме в виде блок-схемы. Блок-схема алгоритма — это графический способ записи алгоритма с использованием геометрических фигур (функциональных блоков), соединенных между собой стрелками, указывающими последовательность выполнения действий. Приняты определенные стандарты графического изображения блоков.

Основные блоки, используемые при графической форме записи алгоритмов:



Пример

Начало

Ввод A, B

$X := A * B$

Вывод X

Конец

Описание в виде программы для компьютера на языке программирования. Язык программирования объединяет систему кодирования предписаний и правила их использования. Написанию программы может предшествовать представление алгоритма в виде блок-схемы или текста на учебном алгоритмическом языке.

Пример

```
program Tutor;
begin
  Writeln(5*6);
  Writeln('Привет');
End.
```

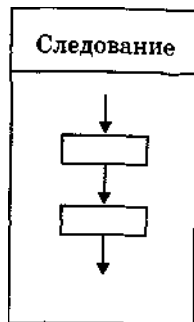
Билет 7

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

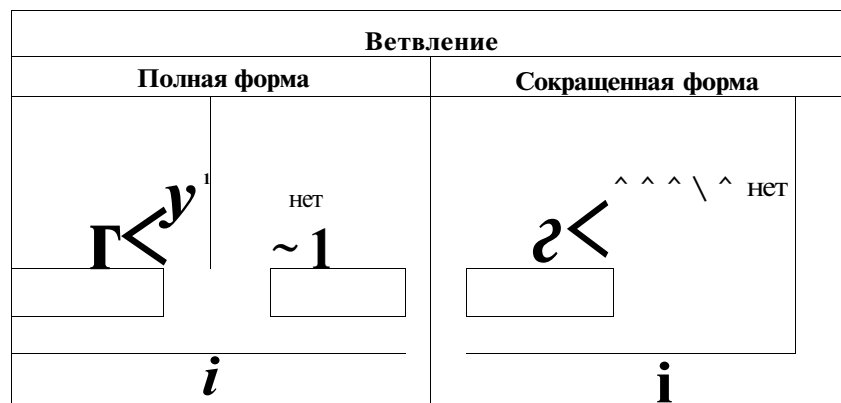
Блок-схема алгоритма — это графический способ записи алгоритма с использованием геометрических фигур (функциональных блоков), соединенных между собой стрелками, указывающими последовательность выполнения действий. Приняты определенные стандарты графического изображения блоков (см. билет 6).

Структурной элементарной единицей алгоритма является **простая команда**, обозначающая один элементарный шаг работы с информацией. Простая команда на языке блок-схем изображается в виде одного функционального блока, который имеет только один вход и один выход. Из простых команд и проверки условий образуются **составные команды** алгоритма, имеющие более сложную структуру. Как и у простых команд, у составных тоже только один вход и один выход. Выделяют всего три базовые алгоритмические структуры (три типа составных команд алгоритма) — следование, ветвление (в полной и сокращенной формах), повторение (с постусловием и предусловием). С помощью соединения только этих базовых конструкций (последовательно или вложением) можно собрать алгоритм любой степени сложности.

Следование — это составная команда алгоритма, в которой действия следуют одно за другим. При исполнении алгоритма действия выполняются всегда в одном и том же порядке, как они записаны. Под действием понимается либо простая, либо составная команда. **Линейные алгоритмы** состоят только из команд следования.



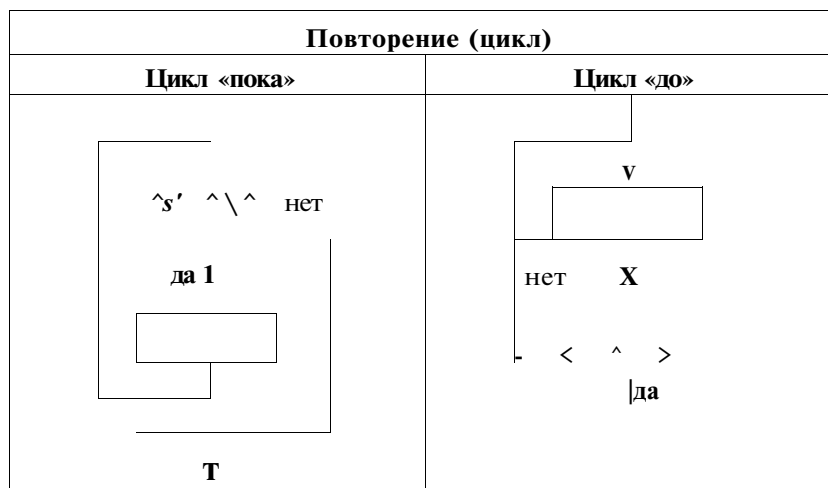
Ветвление — это составная команда алгоритма, в которой в зависимости от условия предусмотрен переход либо на одно, либо на другое действие. Действия могут быть простыми или составными командами алгоритма.



Команда ветвления может использоваться в сокращенной форме, когда в случае несоблюдения условия никакое действие не выполняется. В этом случае в блок-схеме команды

ветвления действие отсутствует всегда справа (путь «нет»). Под действием понимается либо простая команда, либо составная команда алгоритма. **Разветвляющиеся алгоритмы** (алгоритмы ветвления) состоят из команд ветвления и могут быть дополнены командами следования.

Повторение — это составная команда алгоритма, в которой в зависимости от соблюдения условия может повторяться выполнение некоторых действий. Под действием, как и прежде, понимается простая или составная команда.



В цикле с предусловием (цикл «пока») сначала проверяется условие. В том случае, когда условие соблюдается (путь «да»), выполняется действие. Затем снова проверяется условие. *Пока* условие соблюдается, выполняется действие (тело цикла). **В цикле с постусловием (цикл «до»)** условие проверяется после выполнения действия. Повторение выполнения действия (тела цикла) происходит в том случае, когда условие не соблюдено (путь «нет»), т. е. повторение производится *до* соблюдения условия. В команде повторения с постусловием тело цикла выполняется хотя бы один раз. В команде повторения с предусловием оно может ни разу не выполниться.

Цикл с параметром (цикл «для») и цикл «повторить *n* раз» рассматриваются как частные случаи команды повто-

рения с предусловием. Циклические алгоритмы состоят из команд повторения и могут быть дополнены командами следования [4].

Значительное увеличение сложности задач, решаемых с помощью компьютера, приводит к увеличению сложности алгоритмов и программ, что порождает дополнительные трудности при их разработке и отладке. Для упрощения работы задачу разбивают на подзадачи. Решение каждой подзадачи оформляется в виде вспомогательного алгоритма, а основной алгоритм организует связь между ними. Таким образом, текст основной программы сокращается, задача решается по частям, а в основной программе прописывается последовательность решения. Каждый вспомогательный алгоритм имеет свое имя и называется подпрограммой (процедурой) или функцией. Обратиться к вспомогательному алгоритму (т. е. процедуре или подпрограмме) можно из любого места программы и неограниченное количество раз. Любая подпрограмма обладает той же структурой, что и основная программа. При вызове подпрограммы выполнение основной программы приостанавливается и управление передается подпрограмме. По окончании работы подпрограммы управление возвращается основной программе.

Отличие подпрограмм-процедур от подпрограмм-функций в том, что процедуры служат для задания совокупности действий, направленной на изменение внешней по отношению к ним программной обстановки, а функции, являясь частным случаем процедур, обязательно возвращают в точку их вызова в основной программе единственный результат. Функции бывают стандартные и определяемые пользователем.

Примечание. Далее предлагается материал для тех, кто изучал язык программирования в достаточном объеме.

Процедура начинается с заголовка, за ним следует имя, а в круглых скобках — список формальных параметров.

Процедура может получить из основной программы несколько переменных для изменения их значений. Чтобы выполнить процедуру, к ней следует обратиться из основной программы с фактическими параметрами. Количество, тип и порядок следования формальных и фактических параметров должны совпадать.

Turbo PascalQBasicОписание процедуры

Procес1иге<имя> [(<формальные параметры>)] ;	SUB <имя>[(<формальные параметры>)]
«Объявление переменных»	«Объявление переменных»
Begin	<Тело процедуры>
<Тело процедуры>	EXIT SUB
End;	END SUB

Обращение к процедуре

<имя>[(фактические параметры)] ;	CALL <имя>[(фактические параметры)]
--------------------------------------	---

Примечание. В квадратных скобках указываются необязательные части оператора, т. е. список формальных параметров может отсутствовать. Параметры процедуры можно разделить на входные и выходные, т. е. результаты

Функция предназначена для того, чтобы возвращать одно значение, оно и является ее результатом. Описание функции начинается с заголовка, за ним следует имя, а в круглых скобках — список формальных параметров.

Turbo PascalQBasicОписание функции

Function <имя>[(<формальные параметры>)];	Function <имя>[(<формальные параметры>)] : <тип результата>
«Объявление. переменных»	<Объявление переменных>
Begin	<Тело функции>
<Тело функции>	<имя>:=<значение>
<имя>:=<значение>;	END FUNCTION
End;	

Обращение к функции

P:=<имя>[^фактические параметры>)]/	P=<имя>[(<фактические параметры>)]
--	---------------------------------------

Примечание. <значение> — это результат выполнения арифметического выражения

Функция или процедура — самостоятельная часть программы, имеющая собственные переменные, которым отводится отдельное, не зависящее от основной программы, место в памяти компьютера.

Turbo Pascal	QBasic
Program <имя>(input, output); Label — раздел меток Const — раздел констант Type — раздел типов Var — раздел переменных Procedure, Function — раздел процедур и функций Begin Раздел операторов End.	Const — раздел констант Type — раздел типов DEF — раздел переменных SUB, FUNCTION — раздел процедур и функций Раздел операторов END
	<i>Примечание. Через главное меню процедуры и функции записываются отдельно, при распечатке программы они печатаются после основной программы</i>

Turbo Pascal	QBasic
Begin p:=1; for i:=1 to n do p:=p * x; End; <i>(конец процедуры)</i>	P=1 FOR 1=1 TO N P=P*X NEXT END SUB <i>REM конец процедуры</i>
Begin <i>{начало тела основной программы}</i> clrscr; Writeln('Введите число y и степень k'); Readln(y,k); stepen(y,k,p); <i>{обращение к процедуре с фактическими параметрами Y,k,p}</i> Writeln('p=' , p); Readln; End.	CLS INPUT "Введите число Y и степень K ";Y,K <i>REM обращение к процедуре с фактическими параметрами Y,K,P</i> CALL STEPEN(Y,K,P) PRINT "P="; P END

Turbo Pascal	QBasic
Задача 1. Написать процедуру воешения в целую степень iV некоторого числа X, N > 0, X > 0. Показать использование процедуры, оба числа вводятся с клавиатуры.	
Program power my; Uses crt; Var y, k: integer; p: longint; <i>{Описание процедуры}</i> <i>{формальные параметры x,n,p}</i> Procedure stepen(x,n, integer;var p: longint); {x, n - входные параметры} {p - выходной параметр} Var i: integer; {i - локальная переменная}	DIM Y, K AS INTEGER DIM P AS LONG <i>REM Описание процедуры</i> <i>REM формальные параметры x,n,p</i> SUB STEPEN(X, N AS INTEGER, P AS LONG) DEFINT I <i>REM X, N - входные параметры,</i> <i>REM P - выходной параметр</i> <i>REM I - локальная переменная</i>

* Задача 2. Нужно определить шанс угадать выигрышный лотерейный номер. Лотерейный номер состоит из трех чисел, каждое из которых может быть от 0 до 99 включительно. Значит, надо вычислить количество размещений из 100 объектов по 3:	
$D_{00}^3 = \frac{100!}{(100-3)! \cdot 3!} = \frac{100!}{97! \cdot 6}$. Следовательно, для решения задачи разумно написать функцию, вычисляющую факториал некоторого числа <i>m</i> (m!=1-2-3-4-...-m)	
Program Loto; Var n, k: integer; a, pi, p2: longint; <i>{Функция вычисления факториала}</i> <i>{m - формальный параметр}</i> Function factorial(m: integer): longint; Var i: integer; mult: longint; <i>{локальные параметры}</i>	DEFINT K, N DEFLNG A,P1, P2 <i>REM Функция вычисления факториала</i> <i>REM M - формальный параметр</i> FUNCTION FACTORIAL(M AS INTEGER) DEFINT I DEFLNG MULT

<pre> begin mult:=1; for i:=1 to m do mult:=mult*i; factorial:=mult;; end; begin clrscr; writeln('Введите количество чисел в билете K и количество объектов N'); readln(k,n); pi:=factorial (n); {обращение к функции} p2:=factorial (n-k); in, n-k - фактические параметры} a:=p1 div p2; writeln('Количество размещений a=',a); readln; end. </pre>	<pre> MULT:=1 FOR 1=1 TO M MULT=MULT*I NEXT FACTORIAL=MULT END FUNCTION CLS INPUT "Введите число номеров K и число объектов N";K,N REM обращение к функции P1=FACTORIAL (N) P2=FACTORIAL (N-K) REM N, N-K - фактические параметры A=P1\ P2 PRINT "Количество размещений A=";A END </pre>
Примечание, div — функция деления нацело	\ — операция деления нацело

Метод программирования, при котором сначала пишется основная программа, затем записываются обращения к пока еще не составленным подпрограммам, а потом описываются эти подпрограммы, называется **методом последовательной детализации** (нисходящим программированием, программированием «сверху вниз»). Причем количество шагов детализации может быть гораздо больше, поскольку сами подпрограммы могут содержать в себе обращения к другим подпрограммам.

Возможен и другой подход к построению программы. Первоначально составляется множество подпрограмм, которые могут понадобиться при решении задачи, а затем пишется основная программа, содержащая обращения к ним. Подпрограммы могут быть объединены в библиотеку стандартных программ и дополняться по мере необходимости. Этот метод называют **сборочным программированием** (восходящим программированием, программированием «снизу вверх»).

Билет 8

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

Величиной называют объект, с которым связывают определенное множество значений. Величина имеет имя. Величины делятся на постоянные и переменные.

Константой (постоянной) называется величина, имеющая неизменяющееся значение.

Переменной называется величина, которая может принимать различные значения. Переменные величины различаются по своему типу. Тип величины определяет множество значений, которые может принимать величина, и множество операций над этими значениями. Например, в языке программирования величины 205 и -45 относятся к целочисленному типу и их можно складывать, умножать, делить и выполнять другие арифметические операции. Величины "цвет" и "ок" относятся к строковому типу и их можно сцеплять (складывать), но нельзя делить или вычитать.

Приведенное выше определение понятия «величина» в общем виде не позволяет в нужном смысле описать процессы, протекающие в компьютере. Более точным было бы понятие «данные», поскольку все величины, с которыми работает компьютер, принято называть данными. При решении задач на компьютере следует определить данные задачи и их тип. **Данные** (англ. data) — это информация, необходимая для выполнения программы,*представленная в формальном виде, который обеспечивает возможность ее хранения, обработки и передачи.

Теперь мы можем определить понятия «переменная» и «константа» в контексте представления данных в компьютере.

Переменная — это именованная область оперативной памяти (ячейка), в которой может храниться нужная информация. Для задания способа хранения переменной приписывают тип, он определяет, как эта область памяти организована.

В ячейку-переменную заносится значение переменной. Для того чтобы это сделать, используют операцию присваивания :=. Например: K:=15. Знак := делит команду присваивания на левую и правую части. В левой части может стоять любая величина (ее имя), а в правой — значение этой величины (текущее). Знак присваивания := не следует путать (заменять) со знаком равенства =. Знак присваивания указывает на действие (присваивание), знак равенства действия не предполагает.

Пример: y:=y+6 читается как: переменной y присваивается значение, равное предыдущему ее значению, увеличенному на 6. Данное выражение со знаком равенства бессмысленно.

^Константа — это именованное значение, которое не может изменяться на протяжении всего времени выполнения программы. *Числовая константа* представляет собой какое-либо число (7; 3.14), а *строковая константа* — произвольную строку ("количество учеников в классе").

Пример

```
const
    MyName=' Kate ';           -'>
    Pi=3.14;
```

При разработке сложных программ трудно представить все данные с помощью отдельных переменных — это существенно затруднило бы их обработку. Для разрешения этой проблемы применяют сложные, **структурные типы данных**. Наиболее распространенным видом таких типов данных являются массивы — индексированные наборы переменных. В языках программирования Basic и Pascal это наборы однотипных переменных.

Для хранения одних и тех же чисел можно использовать разные типы данных. Гибкое применение разных типов данных позволяет существенно экономить объем памяти, занимаемой переменными (особенно при обработке больших массивов данных) и контролировать правильность использования данных на этапе трансляции программы [5].

Типы данных в языках Basic и Pascal:

Turbo Pascal

Простые	Структурированные
<i>Целые:</i> integer(2 байта) longint(4 байта)	<i>Массивы:</i> array (могут быть целыми, символьными, вещественными, логическими)
<i>Вещественные:</i> real (6 байтов) Single(4 байта) Double (8 байтов) Extended (10 байтов)	<i>Строковые:</i> string (1 байт на символ)
<i>Символьные:</i> char (1 байт)	Файлы: file
<i>Логические:</i> Boolean(1 байт)	Записи: record
<i>Пользовательский</i>	

Q Basic

Простые	Структурированные
<i>Целые:</i> INTEGER (2 байта) LONG (4 байта)	<i>Массивы:</i> DIM (могут быть целыми, вещественными и логическими)
<i>Вещественные:</i> SINGLE (4 байта) DOUBLE (8 байтов)	<i>Строковые:</i> STRING (1 байт на символ)
<i>Логические:</i> BOOLEAN (2 байта)	

Ввод данных, вывод результатов проиллюстрирован в следующем примере линейной программы на языке Pascal:

Программа	Значения строк
<pre> Program myprogram; {примерная программа} var X, Y: integer; begin readln Y; X:=Y * 3; write (X); End.</pre>	- название программы; - комментарий к программе; - определение типов переменных; - начало тела программы; - ввод данных с клавиатуры; - присваивание; - вывод результатов на той же строке; - конец тела программы.

Билет 9

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

В билете 8 мы определили понятие «величина». Здесь мы рассмотрим **логические величины**, т. е. величины логического типа. Они рассматриваются **алгеброй логики**. Существуют логические переменные и логические константы.

Логическая переменная может принимать два значения: «истина» и «ложь», логическая константа представляет собой одно из этих двух значений. Иначе их можно записывать как TRUE и FALSE, или Т и F, или 1 и 0, или И и Л и т. д.

Над логическими величинами определяются **логические операции**, в результате которых получаются **логические выражения**.

Рассмотрим базовые логические операции.

Логическое умножение (конъюнкция) — это логическая операция, ставящая в соответствие каждому двум логическим величинам логическое выражение, являющееся истинным тогда и только тогда, когда обе исходные величины истинны. Обозначается: A и B , $A \& B$, $A \text{ л } B$, AB , $A \cdot B$, $A \text{ AND } B$, где A , B — логические величины.

Логическое сложение (дизъюнкция) — это логическая операция, ставящая в соответствие каждому двум логическим величинам логическое выражение, являющееся истинным тогда и только тогда, когда истинна хотя бы одна из величин. Обозначается: A или B , $A \vee B$, $A \text{ OR } B$, $A + B$.

Логическое отрицание (инверсия) — это логическая операция, применяемая к одной логической величине. Обозначается: A , $\neg A$, NOT A , НЕ A . A ложно, когда A истинно, и истинно, когда A ложно.

Логическое следование (импликация). Полученное выражение ложно тогда и только тогда, когда первая величина истинна, а вторая ложна. Обозначается: $A \rightarrow B$.

Эквивалентность. Полученное выражение истинно тогда и только тогда, когда обе логические величины истинны или обе ложны. Обозначается: $A \sim B$, $A \langle \$ \rangle B$, $A = B$.

Логическое выражение также будем называть **формулой** алгебры логики.

Каждая формула определяет некоторую функцию, аргументами которой являются логические переменные. Такую функцию будем называть **логической функцией**.

Логические переменные могут принимать два значения: «истина» и «ложь». Логическая функция также может принимать два значения: «истина» и «ложь».

Таблица простейших логических функций:

Отрицание		Конъюнкция			Дизъюнкция			Следование			Эквивалентность		
A	$\neg A$	A	B	$A \& B$	A	B	$A \vee B$	A	B	$A \rightarrow B$	A	B	$A \sim B$
0	1	0	0	0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	0	1	1	0	1	1	0	1	0
		1	0	0	1	0	1	1	0	0	1	0	0
		1	1	1	1	1	1	1	1	1	1	1	1

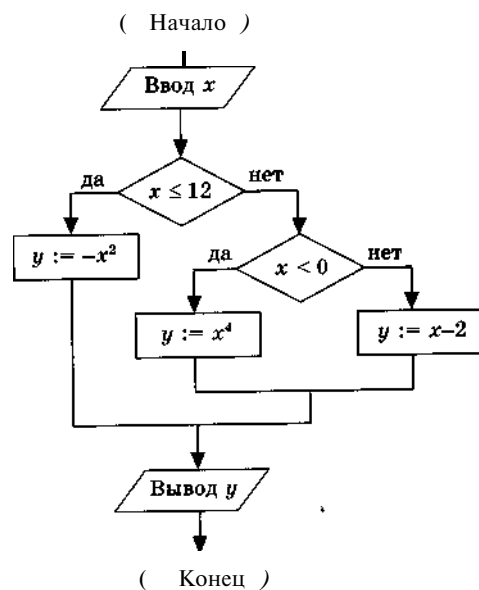
Аргументами логических функций могут быть как логические переменные, так и логические выражения.

Логические выражения используются в качестве условий в ветвящихся и циклических алгоритмах. Заметим, что операция импликации в условиях записывается ЕСЛИ...ТО или IF...THEN.

Пример 1 (ветвящийся алгоритм). Составить алгоритм вычисления значения функции

$-x$ при $x < -12$,
 x^4 при $-12 < x < 0$,
 $x - 2$ при $x > 0$.

Использовать алгоритмический язык или блок-схему.



1. Ввести x

2. Если $x < (-12)$ или $x = (-12)$, то $y := -x^2$, перейти к п. 5

3. Если $x < 0$, то $y := x^4$, перейти к п. 5

4. $y := x - 2$

5. Вывести y

6. Конец

Пример 2 (ветвящийся алгоритм). Составить программу на языке Pascal, которая определяет, являются ли введенные числа величинами углов треугольника.

Для составления программы используют полную форму условного оператора:

IF <логическое условие> THEN <действие 1> ELSE <действие 2>

```

Program program1;
{углы треугольника}
var A, B, C: integer;
begin
  readln A, B, C;
  IF (A>0) AND (B>0) AND (C>0) AND (A+B+C=180)
  THEN Writeln
    ('Являются углами треугольника')
  ELSE Writeln
    ('Не являются углами треугольника');
end.
  
```

Пример 3 (циклический алгоритм). Даны натуральные числа N и M . Определить их наименьшее общее кратное НОК [2].

```

Program NOK;
var I, n, m, k, nod, nok: integer;
begin
  repeat
    write
      ('Введите натуральное число n = ');
    readln (n);
    write
      ('Введите натуральное число m = ');
    readln (m);
  until (n>0) and (m>0);
  
```

```

if n>m then k:=m else k:=n;
for i:=1 to k do
  if (n mod i=0) and (m mod i=0) then nod:=i;
nok:=nod*(n div nod) * (m div nod);
writeln ('Наименьшее общее кратное чисел');
writeln (n, 'и', m, 'nok=', nok)
end.

```

Билет 10

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

Для решения задачи на компьютере необходимо алгоритм ее решения записать на **языке программирования** — составить **программу** (последовательность команд, входящих в систему команд компьютера).

Процесс составления программы на основе некоторого алгоритма называют **программированием** [7].

Можно выделить четыре подхода к программированию: структурное (процедурное), объектно-ориентированное, логическое и функциональное. В школе, как правило, изучаются структурное (процедурное) и объектно-ориентированное программирование.

Структурное (процедурное) программирование основывается на двух основных принципах: последовательной детализации алгоритма и использовании набора базовых структур. Алгоритм любой сложности можно построить с помощью основных алгоритмических структур. В качестве примера языка программирования, реализующего такой подход, можно привести язык Pascal.

Суть **объектно-ориентированного программирования** состоит в разработке программы как модели взаимодействующих объектов. Объект здесь — это некоторая структура данных (в том числе разных типов) и процедур их обработки, к которой можно обращаться по имени и описывать как еди-

ное целое. Формальное описание такого объекта называется классом. Объект в этом случае считается экземпляром некоторого класса. Примеры языков программирования, основанных на таком подходе: Java, C++.

Поскольку компьютер (а точнее, его процессор) оперирует не конструкциями языка программирования, а двоичными командами, то перед выполнением программа должна быть превращена в машинный код. Выполняют эту операцию программы-трансляторы.

Транслятор языка программирования (интерпретатор или компилятор) входит в состав **среды (системы) программирования**. Это программные комплексы, включающие специализированные текстовые редакторы, трансляторы, средства отладки и контроля за исполнением программ, средства разработки дополнительных элементов программ, библиотеки компонентов и многие другие средства, используемые профессиональными разработчиками системного и прикладного программного обеспечения.

Компилятор — это программа, автоматически преобразующая (транслирующая, компилирующая) исходный код языка высокого уровня в машинный код и создающая таким образом исполняемый файл. Он может быть запущен на исполнение операционной системой. В операционной системе Microsoft Windows такие файлы могут иметь расширение exe, com, dll [5].

Интерпретатор — это программа, преобразующая код языка высокого уровня в машинный код шаг за шагом, т. е. каждая команда преобразуется интерпретатором и выполняется компьютером, затем интерпретатор преобразует следующую команду, компьютер ее выполняет и т. д. (интерпретатор — «синхронный переводчик»). Недостаток интерпретаторов — низкая скорость выполнения программ. Примеры: интерпретаторы языков Basic и Java Script. Для языка Basic в настоящее время существуют как интерпретаторы, так и компиляторы [5].

Реализация основных алгоритмических структур на языке программирования:

Задача 1. Скорость света C приблизительно равна $3 \cdot 10^8$ м/с. Найти примерную длину светового года L , выразив ее в метрах L_m , футах L_{ph} , милях L_{ml} ($1 \text{ м} = 3,28 \text{ фута}$, $1 \text{ миля} = 1,61 \text{ км}$)

Блок-схема	Язык Pascal	Язык Basic
Линейная структура	<pre> Program Svet_God; Var C, Lm, Lph, Lml: real; Begin Writeln('Введите скорость света C '); Readln(C); Lm:=C*365*24*60*60; Lph:=Lm*3.28; Lml:=Lm/(1.61*10^3); Writeln('Длина светового года = ', Lm, ' м или ', Lph, ' футов или ', Lml, ' миль'); Readln; End.</pre>	<pre> REM Длина светового года L REM Скорость света C INPDT "Введите C"; C Lm=C*365*24*60*60 Lph=Lm*3.28 Lml=Lm/(1.61*1000) Print "Длина светового года ="; Lm; " м или "; Lph; " футов или "; Lml; " миль" END</pre>
Начало		
Ввод C		
$L_m = 0365 \cdot 24 \cdot 60 \cdot 60$		
$L_{ph} = L_m \cdot 3,28$		
$L_{ml} = L_m / (1,61 \cdot 10^3)$		
Вывод		
Ан' А Ан!/ Конец		

Примечание. Константу $3 \cdot 10^8$ следует вводить с клавиатуры в формате 3E8

Задача 2. Решить квадратное уравнение вида $ax^2 + bx + c = 0$ и проверить для различных значений a, b, c

Блок-схема	Язык Pascal	Язык Basic
Структура принятия решения (разветвленная структура). Полное ветвление	<pre> Program sq_equation; Var a, b, c, d, g, x1, x2: real; Begin Writeln('Введите коэффициенты a, b, c квадратного уравнения ax^2+bx+c=0'); Readln(a,b,c); d:=b*b-4*a*c; g:=2*a; If d >= 0 Then Begin x1:=(-b+sqrt(d))/g; x2:=(-b-sqrt(d))/g; WriteIn('x1=',x1, ' x2= ', x2); End Else Writeln('Решений нет'); Readln; End.</pre>	<pre> REM ax^2+bx+c=0 INPUT "введите a, b, c="; a,b,c d=b*b-4*a*c g=2*a IF D<0 THEN PRINT "Решений нет" GOTO 100 x1=(-b+sqrt(d))/g x2=(-b-sqrt(d))/g Print "x1="; x1; " x2="; x2 100 END</pre>
Начало		
Ввод a, b, c		
$d = b^2 - 4ac$		
$d \geq 0?$		
да		
нет		
$x_{1,2} = \frac{-b \pm \sqrt{d}}{2a}$		
Т		
Вывод x_1, x_2		
/ Вывод "Решений нет/"		
Конец		

Примечание. В приведенном примере рассмотрен частный случай нет проверки деления на ноль ($a = 0$)

Задача 3. Найти сумму значений элементов одномерного массива размера 10^j (например, стоимость 10 покупок)

Блок-схема	Язык Pascal	Язык Basic
Циклическая структура со счетчиком	<pre> Program Summa; Var s: real; k: integer; A: array[1..10] of real; </pre>	<pre> REM Сумма элементов массива INPUT "Введите N"; N DIM A(N) S=0 </pre>
Начало)	<pre> Begin S:=0; For k:=1 to 10 do Begin </pre>	<pre> FOR I = 1 TO N INPUT "Введите элемент "; A (I) S=S+A (I) NEXT I </pre>
$S := 0$	<pre> Write('Введите ', k, '-й элемент массива '); </pre>	<pre> Print "Сумма ="; S END </pre>
$k := 1 \text{ to } 10$	<pre> Readln(A[k]); S:=S+A[k]; End; </pre>	
Ввод $A[k]$	<pre> Writeln ('S=',S) ; Readln; End. </pre>	
$S := S + A[k]$		
Вывод S /		
(Конец j		

Примечание. Здесь для обозначения цикла с параметром использовали блок

Билет 11

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

Компьютер — это универсальное (многофункциональное) программно управляемое устройство для хранения, обработки и передачи информации.

/

Архитектура компьютера — это общее описание структуры и функций компьютера на уровне, достаточном для понимания принципов его работы.

В 1945 году в своем докладе математик Джон фон Нейман описал, как должен быть устроен компьютер для того, чтобы быть универсальным устройством для работы с информацией. Это описание называют архитектурой фон Неймана.

Принципы фон Неймана:

- принцип программного управления, согласно которому программа состоит из набора команд, которые выполняются процессором в определенной последовательности;
- принцип однородности памяти, согласно которому программы и данные хранятся в одной и той же памяти (оперативном запоминающем устройстве, ОЗУ);
- принцип адресности, согласно которому память состоит из пронумерованных ячеек и процессору в любой момент доступна любая ее ячейка.

Конструктивно современные компьютеры реализуются в виде взаимодействующих специализированных устройств, созданных из микросхем, напаянных на печатные платы.

Процессор — центральное устройство, выполняющее все арифметические и логические операции. Конструктивно центральный процессор в современных ПК чаще всего один. Данные и программы для их обработки представляются в оперативной памяти в виде двоичного кода. Для сохранения данных и программ при отключении электропитания используются различные устройства внешней памяти.

Характеристиками процессора являются тактовая частота и разрядность. Микросхема генератора *тактовой частоты* задает ритм работы процессора, который сейчас измеряется в гигагерцах. Максимальную длину двоичного кода, который может передаваться или обрабатываться процессором одновременно, называют *разрядностью*. Современные процессоры имеют разрядность 32 или 64 бита.

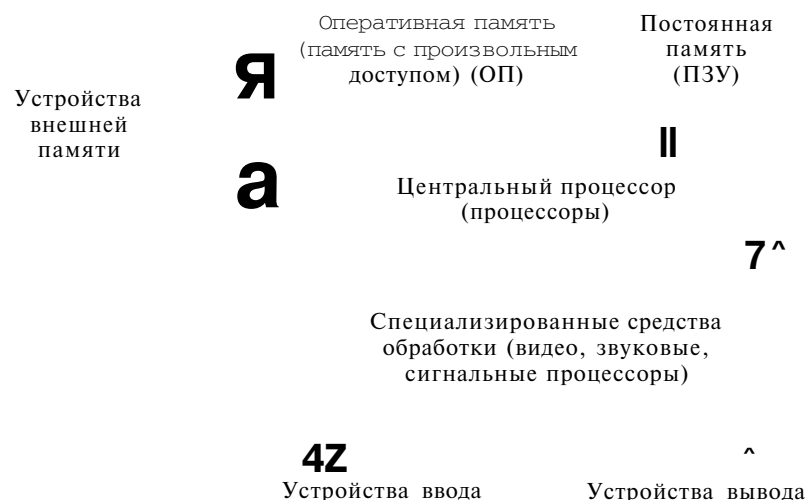
Внутренняя память компьютера делится на две части: ОЗУ и ПЗУ.

ОЗУ (оперативное запоминающее устройство) — быстрая полупроводниковая энергозависимая память, хранит данные и команды, с которыми работает процессор. Позволяет читать и записывать данные в ячейки памяти. Характеристиками оперативной памяти являются ее объем и время доступа к ячейке. В современных компьютерах *объем оперативной памяти* достигает 512 Мб — 4 Гб, а *время доступа к ячейке* — менее 10 нс.

ПЗУ (постоянное запоминающее устройство) — энергонезависимая память. В ПЗУ хранятся программы контроля оборудования и первоначальной загрузки ОС. ПЗУ — это память только для чтения, микросхема программируется один раз в заводских условиях. *Основные характеристики компьютера* — это объем оперативной памяти, тактовая частота и разрядность процессора.

Периферийные устройства служат для повышения функциональных возможностей компьютера, удобства ввода и вывода информации [5].

Основные виды устройств компьютера и их взаимодействие показаны на следующей схеме:



Примечание. Далее дана подробная информация о периферийных устройствах, которая может не использоваться в ответе на билет (на усмотрение учителя информатики).

Устройства внешней памяти рассчитаны на основные типы *носителей*: жесткий диск, гибкий магнитный диск, оптический диск и микросхемы энергонезависимой памяти.

Жесткий диск устанавливается в системный блок (в настоящее время жесткие диски могут не встраиваться в системный блок, а подключаться через порты). Жесткий диск считается несъемным, поскольку он чувствителен к вибрации. Он представляет собой конструкцию из самого носителя и устройства чтения/записи с необходимым управлением в едином защищенном корпусе. Носителем информации являются магнитные диски, собранные в пакет. Вращается пакет дисков общим мотором. Считывание и запись выполняются с помощью набора головок. Управление процессом считывания и записи выполняется электроникой диска, смонтированной в том же корпусе. Информация на диске располагается дорожками, соответствующими углу поворота головок. Важная особенность: из-за высокой плотности записи и скорости вращения крайне важно полное отсутствие пыли, она может привести к повреждению носителя.

Гибкий магнитный диск заключен в защитный корпус, с помощью которого диск защищается от прямого света и повреждений. Диск вращается внутри корпуса. Для считывания и записи диск вставляется в устройство — дисковод, при этом сдвигается защитная шторка и магнитная головка получает доступ к диску. Головка движется по направляющим с помощью шагового двигателя. Важная особенность: магнитный слой на диске чувствителен к перепадам температуры, влажности, магнитного поля и вибрации, а поэтому дискета — ненадежный носитель, часто выходящий из строя. Применяется из-за дешевизны, компактности и из-за распространенности устройств.

Оптический диск. Чтение/запись информации выполняется с помощью отражения от поверхности лазерного луча. Существует два основных типа носителей: «только чтение» (штампуются на заводе) и «чтение и запись» (записываются

на компьютере пользовате_ля в зависимости от типа один или несколько раз). Диски: меньше подвержены внешним воздействиям, не требуют корпуса. Устройства, позволяющие не только читать, но и: записывать информацию, сложнее и дороже. Диски различаются по емкости и особенностям чтения/записи, но общие принципы сохраняются.

Основные характеристики современных устройств памяти:

Вид внешней (долговременной) памяти	Скорость передачи данных	Емкость (минимум/максимум)
<i>Магнм-тные диски</i>		
Дискета	75 Кб/с	1,44 Мб—2,88 Мб
Жесткий диск	33 Мб/с—320 Мб/с	Ю Гб — 400 Гб
<i>Оптические диски</i>		
CD-ROM/CD-R/CD-RW	150 Кб/с—7,6 Мб/с	650 Мб — 800 Мб (5,25") 210 Мб — 280 Мб (3,25")
1 DVD-ROM/DVD-R/DVD-RW DVD+R/DVD+RW DVD-RAM	1,38 Мб/с — 22,08 Мб/с	4,7 Гб — 8,5Гбна одной стороне
<i>Микросхемы памяти</i>		
USB Flash Drive	До 24 Мб/с — чтение, до 16 Мб/с — запись	16 Мб — 4 Гб
Compact Flash	До 12,3 Мбит/с — чтение, до Ю, 4 Мбит/с — запись	16 Мб — 8 Гб
SecureDigital (SD)	До 23 Мбнст/с — чтение, до 20 Мбит/с — запись	16 Мб — 4 Гб
xD	До 5 Мб/с — чтение, до 2,7 Мб/с — запись	16 Мб — 1 Гб

Микросхемы энергонезависимой памяти. Это специализированные микросхемы, сохраняющие информацию длительное время без внешнего питания. Наиболее употребительные из них — флэш-брелоки, подключаемые через USB-накопители (микросхемы с электроникой чтения/записи и интерфейсом смонтированы в корпусе), и различные карты памяти (устройство для чтения и записи CardReader является внешним для компьютера). Карты памяти (CompactFlash, SD, xD) часто используются во всевозможных цифровых устройствах. Скорость работы с тем или иным носителем зависит от его собственных характеристик (максимальной скорости записи, например), от параметров устройства и от способа подключения к системной псине. Существует два основных типа таких шин — параллельная (ATA, SCSI) и последовательная (SerialATA, USB, FireWire). Более современные шины — последовательные, это позволяет упростить изготовление кабелей и увеличить скорость обмена [4].

Для ввода данных в компьютер и демонстрации результатов обработки применяются различные устройства **ввода/вывода** (табл. на с. 84-85).

Для подавляющего большинства современных устройств разного назначения предусмотрены стандартные средства подключения, адаптеры и контроллеры. Место для их установки предусмотрено на системной плате компьютера.

По определению, данному в ответе на билет, компьютер является универсальным, программно управляемым устройством. **Универсальность** достигается применением большого количества самого разного программного обеспечения, которое и обеспечивает обработку информации их всех предметных областей, представленную в различных видах: числовом, текстовом, графическом, звуковом.

Информация, обрабатываемая компьютером, называется **данными**. Последовательность команд, выполняемую компьютером в процессе обработки данных, называют **программой**. Как программы, так и данные внутри компьютера представлены в двоичном коде.

Работа компьютера обеспечивается единством двух составляющих: аппаратуры и программного обеспечения.

Основные характеристики устройств ввода/вывода:

Вид устройства	Характеристика	Диапазон или возможные варианты значений
Монитор	Тип матрицы	ЭЛТ или ЖК
	Размер области отображения	До 22"
	Размер пикселя	От 0,27 до 0,24 мм
	Частота обновления	До 180 Гц для ЭЛТ, для ЖК параметр менее важен
Принтер	Способ печати	Матричный, струйный, лазерный
	Размер отпечатка	До А0
	Разрешение	До 4800 dpi (точек на дюйм)
	Цветность	До примерно 4 млрд оттенков (теоретически)
Телевизор	Размер диагонали	До 24"
Колонки	Диапазон воспроизводимых частот	Примерно от 25 Гц до 40 КГц
	Максимальная мощность	До 250 Вт
	Уровень шумов (соотношение сигнал/шум)	До 100 Дб
Цифровой фотоаппарат	Разрешение матрицы	До 16,7 Мпикс
	Тип памяти	CompactFlash, Secure Digital, xD, MemoryStick
	Фотохарактеристики	Характеристики объектива и системы фокусировки
Цифровая видеокамера	Размер матрицы	До 3 Мпикс
	Тип носителя	Кассеты DV, DVD-R-диски
	Оптические и звуковые характеристики	Характеристики объектива и микрофона

Вид устройства	Характеристика	Диапазон или возможные варианты значений
Web-камера	Разрешение	До 1,3 Мпикс
	Количество цветов	До 16,7 млн
Сканер	Способ сканирования	Ручной, планшетный, барабанный, фотокамера
	Размер области	До А0 (для сканирования чертежей)
	Оптическое разрешение	До 9600 Dpi (для сканеров фотопленок)
Графический планшет	Разрешение	До 5000 lpi (линий на дюйм)
	Чувствительность	До 1024 уровней нажатия
	Размер рабочей области	До А1 — для чертежных моделей
Микрофон (для нестудийных моделей)	Чувствительность	58 db +/- 2 db
	Диапазон частот	50-16000 Гц
Музыкальная клавиатура	Количество октав	До 5 октав, 61 клавиша
Клавиатура	Способ подключения	USB, PS/2, IrDA, Bluetooth
Мышь	Разрешение	Около 400 lpi
	Принцип фиксации	Механический или оптический
	Способ подключения	USB, PS/2
Джойстик	Способ подключения	USB, RS232

Билет 12

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

Современные компьютеры можно считать универсальными, поскольку они применяются для автоматизации обработки и обмена самой разной информацией практически во всех отраслях современной жизни. Эта универсальность достигается в первую очередь применением огромного количества программного обеспечения, реализующего собственно операции по обработке информации.

Вопрос состава и структуры программного обеспечения (ПО) компьютера однозначно не определен. В различных источниках данный вопрос раскрывается по-разному.

Мы предлагаем деление ПО на два крупных блока — системное ПО и прикладное ПО.

Системное программное обеспечение. Сюда относятся программы, обеспечивающие выполнение общих для всех программ технических задач, взаимодействие с аппаратурой, т. е. обеспечение функционирования компьютера, работу с файлами, защиту программ и данных, возможность запуска и работы других программ.

а) *Ядро операционной системы.* Функции этих программ, данных и библиотек функций — управление выполнением программ, оперативной памятью, обеспечение взаимодействия программ.

б) *Системные библиотеки функций.* Сюда входят библиотеки и программы, обеспечивающие работу с устройствами внешней памяти (файловые системы), устройств ввода/вывода (обеспечение интерфейса с пользователем), и др. В большинстве современных ОС в ядро или важные системные библиотеки также входят библиотеки функций для работы с сетями.

в) *Драйверы.* Это программное обеспечение, разрабатываемое поставщиками аппаратных средств и в операционной системе управляющее нестандартными (не предусмотренными при разработке ОС) устройствами. Драйвер обеспечивает вы-

полнение стандартных для класса устройств функций, что позволяет разрабатывать новые, более совершенные устройства и применять их без принципиальных модификаций ОС.

г) *Утилиты.* Это небольшие программы, реализующие частные технические задачи по обслуживанию компьютера: архивирование, контроль состояния устройств внешней памяти, поиск нужных файлов и пр.

Прикладное программное обеспечение. К прикладному ПО относятся программные комплексы, обеспечивающие выполнение различных прикладных задач пользователей. Множество таких программ и комплексов огромно и исчерпывающей классификации не поддается. Можно выделить несколько часто используемых видов прикладного ПО.

- а) *Офисные пакеты.* Это комплексы программ, помогающих решать основные задачи делопроизводства — подготовку документов, выполнение расчетов, презентации, ведение переписки и организацию работы и др.
- б) *Системы управления базами данных (СУБД), справочные системы и оболочки автоматизированных информационных систем.* Эти программы позволяют организовать ввод, хранение и работу с большими объемами специализированных данных. СУБД часто являются общими компонентами, обеспечивающими работу большого количества специализированных комплексов.
- в) *Программы обработки графической информации.* Это крупный класс программ, целью применения которых является формирование или анализ изображений. Среди них можно упомянуть программы обработки фотоизображений, издательские комплексы, системы подготовки реалистичных трехмерных изображений и др.
- г) *Среды программирования.* Это программные комплексы, включающие трансляторы, специализированные текстовые редакторы, средства отладки и контроля за исполнением программ, средства разработки дополнительных элементов программ, библиотеки компонентов и многие другие средства, используемые профессиональными разработчиками системного и прикладного программного обеспечения [5].

Назначение операционной системы

Для обеспечения работы любых прикладных комплексов необходима **операционная система (ОС)**. От нее в значительной мере зависит устойчивость работы, специфика применяемых программных средств, особенности построения прикладных систем. При включении компьютера операционная система загружается в память раньше остальных программ и затем служит платформой и средой для их работы.

Во-первых, задачей операционной системы является управление ресурсами: процессорным временем, оперативной памятью, доступом к устройствам внешней памяти.

Во-вторых, ОС реализует взаимодействие программ (между собой и аппаратурой).

В-третьих, современные операционные системы обеспечивают взаимодействие компьютера с пользователем (человеко-машинный интерфейс) и с другими компьютерами — в сетях.

Основной компонент операционной системы — ядро. Ядро — это комплекс программ, постоянно присутствующих в оперативной памяти и выполняющих задачи управления процессами и памятью. Помимо этого, операционная система содержит библиотеки функций, выполняющих конкретные задачи. К этим библиотекам обращаются прикладные программы для выполнения типовых задач. Пример такой задачи — управление данными на внешних носителях (поддержка файловой системы).

Как правило, операционная система поставляется с набором прикладных программ, которые обеспечивают выполнение некоторых полезных функций: архивирование, поиск нужных файлов и др. Такие вспомогательные программы называются утилитами.

Существует большое количество классификаций операционных систем. Наиболее популярны следующие основные классификации.

1. По способу управления выполняемыми процессами:

а) *однозадачные*. В один момент времени выполняется одна задача, только по окончании исполнения управление передается следующей задаче (или процессору команд ОС);

б) *многозадачные*. В оперативной памяти присутствует несколько программ, которые с точки зрения пользователя выполняются одновременно. На практике ОС постоянно переключается от одной программы к другой.

2. По разделению среды пользователей:

а) *однопользовательские* — ОС, не предусматривающие разделение ресурсов между пользователями (т. е. не различающими пользователей);

б) *многопользовательские* — ОС, имеющие в своем составе средства разграничения ресурсов между пользователями.

В настоящий момент активно используются две линии операционных систем:

1) операционные системы линии UNIX;

2) операционные системы линии Windows.

Все эти ОС являются многозадачными, многопользовательскими, обеспечивают взаимодействие с широким спектром аппаратуры, предусматривают расширение возможностей за счет разработки самых разных программ. В составе этих ОС в том или ином виде поставляются средства организации **графического интерфейса пользователя**. Все они имеют свои достоинства и недостатки, выбор конкретной ОС обуславливается конкретной задачей и предпочтениями пользователя.

Командное взаимодействие пользователя с компьютером

Для решения задачи обеспечения взаимодействия с пользователем (человеко-машинного интерфейса) используется командный способ (иначе его называют диалоговым или интерактивным). Операционная система выдает на экран приглашение в определенной форме. В ответ вводится команда, определяющее следующее действие компьютера. Выполнив очередную команду пользователя, операционная система вновь выдает приглашение. Внешний вид команды определяется используемой операционной системой — это может быть командная строка или графический интерфейс.

Слово **интерфейс** (англ. interface) означает место или способ соединения/соприкосновения/связи. Этот термин широ-

ко распространен и используется в *самых разных* контекстах в зависимости от отрасли знаний.

Интерфейсы являются основой взаимодействия всех современных информационных и коммуникационных систем.

В современных информационных системах взаимодействие может осуществляться на пользовательском, программном и физическом уровнях. В соответствии с этой классификацией можно выделить:

1) интерфейс пользователя — совокупность средств, при помощи которых пользователь управляет работой компьютера. Это может быть:

- *интерфейс командной строки*: инструкции компьютеру даются путем ввода с клавиатуры текстовых строк (команд);
- *графический интерфейс*: программные функции представляются графическими элементами экрана.

2) физический интерфейс — способ взаимодействия физических устройств. Чаще всего речь идет о компьютерных портах. Различают также **сетевой интерфейс** и **интерфейсы в программировании**.

Билет 13

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

Информация, представляемая для обработки на компьютере, называется данными. Для организации хранения на устройствах внешней памяти данные записывают в файлы.

Файл — именованная область внешней памяти.

Способ сохранения как служебной, так и пользовательской информации о файлах на носителях называют файловой системой. Используется и другое эквивалентное объяснение файловой системы как физической организации дис-

кового пространства, соответствующей файловой структуре или ее части [11]. Файловая система определяет, в частности, правила именования файлов. Программные средства, необходимые для выполнения операций с файлами и носителями, входят в состав операционных систем. Такие программные средства не обращаются к содержимому файлов и не изменяют его, а работают с файлами просто как с целым, непрерывным массивом данных.

Имя файлу присваивает пользователь, или программа, создающая файл, предлагает имя в автоматическом режиме. По историческим причинам, для пользователя имя файла в операционных системах корпорации Microsoft состоит из двух частей, разделенных точкой: собственно имени и расширения (в последних версиях ОС расширение отделяется последней точкой). Тип файла определяется по его расширению, которое задает программа, сохраняющая файл, исходя из вида сохраняемой информации.

С точки зрения прикладных программ, файл представляет собой некоторую последовательность байтов. Такой подход позволяет интерпретировать некоторые устройства (принтер, модем, клавиатуру, монитор) как файлы.

В некоторых операционных системах предусмотрена такая организация работы и со служебной информацией носителей. Для носителей предусмотрены специальные, зарезервированные системой, имена файлов.

Для удобства хранения и работы файловые структуры организуются с помощью системы вложенных каталогов (папок).

Каталог — специальный системный файл, в котором хранится служебная информация о других файлах.

На каждом носителе может быть множество каталогов. В каждом каталоге может быть зарегистрировано много файлов, но каждый файл регистрируется только в одном каталоге (для ОС DOS и Windows).

На каждом носителе присутствует один главный, или корневой, каталог. В нем регистрируются файлы и подкаталоги (каталоги 1-го уровня). В каталогах 1-го уровня регистрируются файлы и каталоги 2-го уровня и т. д. Получается **иерархическая (древовидная)** структура каталогов.

Каталог, с которым работает пользователь в настоящий момент, называется текущим.

Когда используется файл не из текущего каталога, программе, осуществляющей доступ к файлу, необходимо указать, где именно этот файл находится. Это делается с помощью указания пути к файлу.

Путь к файлу — это последовательность из имен каталогов, в операционных системах Windows разделенных символом «\» (в ОС линии UNIX используется символ «/»). Этот путь задает маршрут к тому каталогу, в котором находится нужный файл. **Полным именем файла** называют имя файла, дополненное путем к нему. Например, запись D:\KLASS10\DOCS\START2\text.doc означает, что файл text.doc находится в подкаталоге START2, который находится в каталоге DOCS, а он, в свою очередь, находится в каталоге KLASS10 корневого каталога тома с названием D:. Если перед первым символом «обратный слэш» стоит точка, то отсчет ведется от текущего каталога. Символ «..» означает каталог, в который входит текущий каталог (каталог уровнем выше).

Над файлами можно производить следующие **основные операции: копирование, перемещение, удаление, переименование** и пр. Эти операции могут быть выполнены независимо от содержимого файлов средствами, входящими в поставку любой ОС.

При работе с информационными носителями необходимо уметь архивировать информацию. **Архивирование файлов** представляет собой уменьшение объема, занимаемого файлом на носителе, без значимой для человека потери исходных данных. Архивируют файлы специализированные приложения — архиваторы или файловые менеджеры. В настоящее время разработано много алгоритмов архивирования информации (без потери информации, с потерей информации). Пример: уменьшение объема файла путем замены последовательности одинаковых байтов (символов) двумя байтами — самым символом и числом его повторений.

Обеспечение **сохранности информации** также является актуальной задачей для любого пользователя. Один из необ-

ходимых компонентов процедуры обеспечения сохранности — ее защита от вирусов. **Компьютерные вирусы** (включая «черви» и «троянские» программы) — это программы или фрагменты программного кода, которые, попав на компьютер, могут вопреки воле пользователя выполнять различные операции: создавать или удалять объекты, модифицировать файлы данных или программные файлы, и осуществлять действия по собственному распространению по локальным вычислительным сетям или по сети Интернет.

Существует большое количество вирусов, классифицируемых по различным основаниям. Для борьбы с подавляющим большинством таких программ применяются различные антивирусные программы. Наиболее распространены следующие из них.

1. **Программы-сканеры (полифаги).** Эти программы после запуска анализируют файлы на диске на предмет обнаружения программного кода вирусных программ. При их обнаружении полифаги принимают меры к удалению вредоносного кода, его блокированию или удалению всей вредоносной программы. Корректность и эффективность работы такой программы зависят от ее своевременного обновления (программы обнаруживают и удаляют, в основном, известные вирусы и их модификации) и настройки параметров сканирования и удаления. Полифаги не обеспечивают мониторинга в режиме реального времени, для чего требуют наличия программ-мониторов.
2. **Программы-мониторы.** Проверяют файлы, запускаемые, открываемые или модифицируемые во время работы системы. Способ проверки сходен с принципом работы полифагов, зачастую они используют общие базы данных о вирусах и механизмы их удаления. Позволяют принимать меры более оперативно, но не выявляют вирусы, уже имеющиеся на диске (например, пропущенные устаревшей версией монитора — до его обновления). Дополняют полифаги.
3. **Программы-фильтры.** Эти программы проверяют поток данных, принимаемых системой по определенному протоколу (электронной почты, Web-страниц и пр.). Позволяют защитить компьютер от получения вредоносных программ из сети.

4. Программы-детекторы нежелательного программного обеспечения (ПО). С многими свободно распространяемыми программами или свободно доступными Web-страницами связаны формально не вредоносные программы, которые тем не менее могут затруднять работу пользователя, используя его компьютер для нежелательных операций или разглашая личные данные пользователей. Значительная часть таких программ выявляется антивирусами-полифагами, но иногда это не программы, а настройки уже имеющегося ПО. В таких ситуациях полифаги бесполезны. Выявляют такие настройки и устраняют их программы-детекторы (Anti-Spy Ware).

Все эти программы не могут полноценно противостоять распространяющимся с помощью уязвимостей в сетевом программном обеспечении вирусам-червям. Для защиты от таких программ необходимо своевременно обновлять уже установленное ПО (обновлениями, выпущенными производителями), а также применять программы контроля работы с сетями — **брандмауэры**.

Таким образом, защита — комплексная задача, требующая грамотного применения набора взаимодействующих программ, а также своевременного их обновления [5].

Билет 14

Информационные ресурсы общества. Основы информационной безопасности, этики и права

Ресурсом принято считать некоторый запас или источник каких-либо средств, использование которых может принести некоторую выгоду или необходимо для решения тех или иных задач. Ресурс может принадлежать государству, компании или частному лицу.

В настоящее время наряду с материальными ресурсами для государства большое значение приобретают информационные ресурсы. Информационными ресурсами считают накапливаемое содержимое специальных хранилищ и источники общественно-значимой информации.

Как и для материальных ресурсов, для информационных ресурсов остро стоит проблема обеспечения сохранности. Для решения проблемы охраны информационных ресурсов существует юридическое определение понятия «информационный ресурс»: «Информационные ресурсы — отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах)» [8].

В информационном обществе значимость информационных ресурсов возрастает. Информационные ресурсы государства становятся товаром, совокупная стоимость которого на мировом рынке сопоставима со стоимостью традиционных ресурсов. Информационные ресурсы государства можно считать стратегическими ресурсами, поскольку по значимости их можно приравнять к ресурсам материальным, сырьевым, энергетическим, финансовым или трудовым.

Отличительная особенность информационных ресурсов в том, что после использования они не исчезают, ими можно пользоваться многократно. Из описанной особенности вытекает необходимость хранения информации таким образом, чтобы она была легко доступна для использования. Эта особенность способствует формированию и расширению рынка информационных услуг. К ним относятся поиск и подбор информации по заданным критериям, консалтинг, обучение, телекоммуникации и пр.

Информационные услуги невозможно представить без развитых средств телекоммуникаций. Кроме того, решающее значение для формирования рынка информационных услуг имело и имеет создание баз данных по различным отраслям профессиональной и научной деятельности. Огромные информационные потоки нуждаются в систематизации для их хранения и использования. Поэтому информационные ресурсы классифицируют по какому-либо основанию. Это может быть отраслевой принцип (по виду науки, промышленности, социальной сферы и пр.) или по форме представления (по виду носителей, степени формализации, наличию дополнительных возможностей и пр.).

К образовательным ресурсам можно отнести любые информационные ресурсы, используемые в образовательных целях. Образование в современном обществе становится непрерывным. Для разных категорий обучаемых требуются различные виды образовательных ресурсов. К наиболее востребованным образовательным ресурсам можно отнести библиотечные образовательные ресурсы, архивные ресурсы и информацию по различным научным отраслям.

На современном этапе развития информационных технологий необходимо обратить внимание на вопросы защиты информации. Защита информации представляет собой самостоятельный информационный процесс, но сегодня более важной становится организация защиты информации как важнейшего компонента процессов хранения, обработки, передачи информации в системах любого типа, особенно в социальных и технических.

На рынке информационных услуг возникают новые отношения между его участниками, нуждающиеся в правовом регулировании со стороны государства. Но правовое регулирование этой сферы отстает от реальных потребностей общества. Поэтому на первое место выходят этические аспекты, определяющие нормы поведения в сфере использования информационных ресурсов. Применяя современные телекоммуникации в ежедневной работе, следует не использовать нелегальные копии программных продуктов или отдельных информационных ресурсов, не рассылать многочисленные рекламные объявления без получения предварительного согласия получателей, не допускать действия, являющиеся запрещенной пропагандой, не отклоняться от обсуждаемых тем в форумах и телеконференциях и т. д.

Базовым законом в информационной области можно считать Федеральный закон «Об информации, информационных технологиях и о защите информации» № 149-ФЗ от 27 июля 2006 г. В законе содержатся определения терминов, которые мы привыкли использовать. Их толкование с юридической точки зрения поможет при необходимости отстаивать свои права. Для отдельного человека это прежде всего право на неприкосновенность его частной жизни. Мы живем в такое время, когда каждое произнесенное слово, отправленное

электронное письмо, информация о покупке по кредитной карте, участие в компании и прочие действия сохраняются и могут быть использованы против нас.

Пункт 7 статьи 3 этого закона одним из основных принципов правового регулирования отношений в сфере информации, информационных технологий и защиты информации определяет неприкосновенность частной жизни, недопустимость сбора, хранения, использования и распространения информации о частной жизни лица без его согласия. Статьей 9 предусмотрены ограничения доступа к информации о частной жизни гражданина.

Ранее этот закон содержал понятие «персональные данные». В настоящее время принят отдельный Федеральный закон «О персональных данных» от 27 июля 2006 года № 152-ФЗ.

Информационные и коммуникационные технологии проникают во все сферы жизни, и учесть все это в одном или двух законах невозможно, поэтому в существующие законы, регулирующие определенную деятельность, вносятся необходимые поправки, которые не должны противоречить Конституции Российской Федерации. В п. 2 ст. 23 Конституции Российской Федерации говорится: «Каждый имеет право на тайну переписки, телефонных переговоров, почтовых, телеграфных и иных сообщений. Ограничение этого права допускается только на основании решения суда».

В настоящее время назвать преимущества электронной переписки может любой из нас, поэтому остановимся на рисках использования данной услуги. Переписка по электронной почте не гарантирует конфиденциальность, она открыта. Современные технические и программные средства позволяют легко контролировать содержимое всей или части переписки любого пользователя. Подобный контроль осуществляется как в государственных, так и в корпоративных системах под предлогом защиты от утечки конфиденциальной информации. Во всех странах такая деятельность является незаконной. Для государственных структур для осуществления этой деятельности требуется соответствующее решение суда. Однако здесь подстерегает несогласование нескольких законов. Решение суда на перлюстрацию почто-

вой корреспонденции можно получить только в соответствии с Федеральным законом «О почтовой связи» от 17 июля 1999 г. № 176-ФЗ, в котором пока отсутствует понятие «электронное письмо». Поэтому решение суда на перлюстрацию электронных писем получить не представляется возможным. Необходимо внести поправки в закон.

Таким образом, ради скоростных преимуществ пользователи электронной почты отказываются от конфиденциальности.

Большую проблему для активных пользователей электронной почты представляет собой спам. Сейчас наказать спамеров в России возможно на основании Федерального закона «О рекламе», новая редакция которого принята 13 марта 2006 г. Рассылку коммерческих предложений по электронной почте можно расценивать как рекламу. В соответствии со статьей 3 этого закона, «Реклама — информация, распространенная *любым способом, в любой форме и с использованием любых средств*, адресованная неопределенному кругу лиц и направленная на привлечение внимания к объекту рекламирования, формирование или поддержание интереса к нему и его продвижения на рынке». Действия лиц, осуществляющих рассылку, регулируются статьей 8 как «реклама товаров при дистанционном способе их продажи». В рекламе товаров при дистанционном способе их продажи должны быть указаны сведения о продавце таких товаров: наименование, место нахождения и государственный регистрационный номер записи о создании юридического лица; фамилия, имя, отчество, основной государственный регистрационный номер записи о государственной регистрации физического лица в качестве индивидуального предпринимателя. Отсутствие таких сведений позволяет квалифицировать рассылку как «ненадлежащую рекламу» и применять соответствующие административные санкции в виде штрафа или лишения лицензии на осуществление предпринимательской деятельности.

Билет 15

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

Информационные технологии, связанные с обработкой текстовой информации, позволяют создавать документы и вести издательскую деятельность.

Для работы с текстовой информацией мы пользуемся текстовыми редакторами и текстовыми процессорами.

Текстовый редактор — программа для ввода и редактирования текста. Пример текстового редактора — программа Блокнот, входящая в комплект ОС Windows.

Текстовый процессор, помимо редактирования, позволяет еще и форматировать текст и записывать в файл информацию о форматировании. Эта информация может быть записана по-разному, отсюда несовместимость текстовых процессоров друг с другом. Предусмотрена *возможность* преобразования текста из одного формата в другой. (Текстовые же редакторы сохраняют в файле текст, не содержащий команд разметки, поэтому они совместимы друг с другом.) Примером текстового процессора является программа Microsoft Word.

Основными объектами (структурными элементами) текстового документа являются: **символ, слово, строка, абзац, страница, документ.**

Редактирование — это преобразование документа, обеспечивающее набор, исправление ошибок, копирование, вставку, удаление, перемещение фрагментов текста. Всё, что позволяет делать с текстом или его фрагментами команды меню **Правка, Вставка** и некоторые кнопки панели инструментов **Стандартная**, является редактированием. Редактирование может быть применено к любому объекту документа.

Форматирование — это изменение параметров шрифта, абзаца и внедренных объектов, т. е. такое преобразование, которое меняет форму представления документа. Возможны следующие виды форматирования:

1. Задание параметров страницы (размер, поля, ориентация и др.).
2. Форматирование абзаца:
 - выравнивание (по ширине, центру, правому и левому краю);
 - установка отступа первой строки;
 - установка отступов/интервалов между строками и абзацами.
3. Форматирование символа:
 - Изменение гарнитуры шрифта: моноширинный (Courier), рубленый (Arial) или с засечками (Times New Roman).
 - Изменение начертания шрифта (обычное, **полужирное**, *курсивное*, *полужирное курсивное*, варианты подчеркивания, цвет символов, нижний или верхний индекс).
 - Изменение размера (кегля) символов (единицей измерения размера шрифта является пункт, 1 пт = 0,376 мм).
4. Формирование списков.
5. Оформление границ и заливки.
6. Расположение текста в несколько колонок.
7. Форматирование таблиц (расположение текста, объединение ячеек, ширина столбцов и высота строк и т. д.).

Большую помощь при форматировании оказывает инструмент **Формат по образцу**.

При сохранении текстовых документов в текстовых редакторах (процессорах) можно задавать различные форматы:

- только текст, *.txt - сохраняет текст без форматирования;
- формат RTF, *.rtf — сохраняет форматирование, универсальный, ориентированный на обмен файлами;
- документ Word, *.doc — сохраняет форматирование;

- HTML-документ, *.htm или *.html — сохраняет Web-страницы, содержит управляющие тэги языка разметки гипертекста и другие.

Встраиваемые объекты. Используя команды меню **Вставка**, можно встроить (внедрить) в текстовый документ *рисунки* из готовой коллекции или из файла, *символы*, сформированное с помощью *графических примитивов* изображение, *таблицу*, *объекты Chart*, *формулу* или *диаграмму*. По своей сути, вставка — это помещение в документ объекта, который отображается и обрабатывается внешней программой.

Понятие гипертекста. Для отображения в линейном тексте смысловых связей между понятиями или целыми разделами документа используют **гиперссылки**. Гипертекстом называют структурированный текст с возможностью перехода по выделенным элементам-ссылкам к необходимому фрагменту текста, рисунку или иному объекту. В широком смысле **гипертекст** — информационный массив, в котором заданы и автоматически поддерживаются ассоциативные и смысловые связи между выделенными элементами, понятиями, терминами или разделами. Именно таким образом формируется единое информационное пространство в сети Интернет. Web-страницы сайтов представляют собой документы с гипертекстовой разметкой. Поэтому так легко путешествовать по Всемирной паутине (WWW — Worl Wide Web), переходя с сайта на сайт по гиперссылкам.

Билет 16

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

Существует два основных подхода к представлению графической информации для ее обработки компьютером — растровый и векторный. В растровом представлении изображение — это совокупность данных о координатах и цветах

пикселей рисунка. В векторной графике — это данные, математически описывающие графические примитивы (линии, дуги, окружности, прямоугольники и пр.), составляющие рисунок.

К достоинствам растровой графики относят: возможность представления изображения фотографического качества, простоту вывода на внешние устройства. Существенными недостатками растровой графики являются: большой объем памяти, требуемый для хранения изображений, и ограниченные возможности изменения. Достоинством векторного представления является относительно небольшой объем памяти и возможность выполнять изменения без потери качества. Но при этом векторная графика не позволяет получать изображения фотографического качества.

К аппаратным средствам работы с графической информацией относятся:

- мониторы и видеокарты, поддерживающие графический режим отображения;
- манипуляторы «мышь» — для ввода изображения;
- сканеры — для ввода изображения;
- световое перо, графический планшет — для рисования движениями пера;
- принтеры — для вывода изображения;
- графопостроители (плоттеры) — для вывода изображения.

Для формирования изображения на экране монитора в составе всех современных ПК предусмотрен видеоадаптер (видеокарта). Это отдельное устройство, состоящее из видеопамати (для хранения двоичного кода изображения, выводимого на экран) и графического процессора (формирует и считывает состояние видеопамати и в соответствии с ним формирует изображение для монитора).

Для работы с графикой существует множество прикладных программ. Распространенное название этих программ — графические редакторы, его применяют по отношению к прикладным программам, не имеющим какой-либо специализированной ориентации и используемым для рисования или

редактирования отсканированных изображений. Для создания иллюстраций обычно используются редакторы векторной графики. Улучшение качества изображений, а также монтаж фотографий выполняется в редакторах растровой графики.

Рассмотрим, что представляет собой среда большинства графических редакторов. Основное место экрана программы занимает рабочее поле для создания изображения. Для удобства рисования предусмотрены инструменты рисования, включающие средства для рисования в виде карандаша, пера, кисти, а также линейки, ластика, штампа, лупы, лейки для заливки. Имеются готовые графические примитивы — различные линии и фигуры. Есть цветовая палитра. Графические редакторы позволяют включать в рисунок тексты. При этом можно управлять шрифтом, задавать нужные размеры, цвет и эффекты. Обязательным компонентом любой прикладной программы, в том числе и графического редактора, является меню команд в двух возможных форматах: текстовом и в виде пиктограммы. С помощью меню выполняются необходимые действия по сохранению, изменению, печати созданного графического файла.

Билет 17

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

То огромное количество информации, которое требуется человеку для осуществления любой деятельности, следует хранить в структурированном виде, чтобы можно было быстро находить необходимую информацию, делать любые выборки, пополнять, изменять, сортировать информацию. Организованные массивы информации определенного назначения (тематики) называют базами данных (БД). Примеры баз данных: база данных кинотеатров города, база данных

книжного фонда библиотеки, база данных нормативных правовых актов в сфере образования. Программы, обеспечивающие работу с базами данных, называют **системами управления базами данных**.

Для структурирования большого объема данных чаще всего применяется табличная форма. В таблице легко ориентироваться, даже если информации о каком-либо объекте очень много. Каждый столбец таблицы может содержать некоторую характеристику объекта, а каждая строка — описывать один объект.

База данных является основой любой информационной системы. **Информационная система** — это комплекс баз данных различного назначения (тематики) и аппаратно-программных средств (компьютеров и программного обеспечения) для хранения, изменения, поиска различной информации по запросу пользователя.

Роль компьютерных информационных систем в нашей жизни сильно возросла за последние 10-20 лет. Любая отрасль деятельности или жизнеобеспечения базируется на таких системах. Примеры информационных систем: система продажи билетов (авиационных и железнодорожных), система распределения электроэнергии и тепла, организация прямых трансляций соревнований из любой страны и пр.

Ранее существовало деление баз данных на фактографические (краткие сведения об описываемых объектах) и документальные (содержащие документы полного формата). Но современные информационные системы позволяют подключать к любой фактографической базе данных любые документы (текстовые, графические, звуковые и видеофайлы). В настоящее время более актуальна классификация по способу хранения данных: централизованные (на одном компьютере, сервере) и распределенные (различные по назначению базы хранятся на разных компьютерах, связанных в сеть).

Процесс создания базы данных (проектирования) включает, прежде всего, создание структуры таблиц, установку связей между этими таблицами, создание других вспомогательных объектов: запросов, форм, отчетов.

Выделяют следующие объекты, связанные с БД.

Таблица. В СУБД вся информация хранится в таблицах. Это базовый объект БД, все остальные объекты создаются на основе существующих таблиц (производные объекты). Каждая строка в таблице — **запись БД**, а столбец — **поле**. Запись содержит набор данных об одном объекте, а поле — однородные данные обо всех объектах. Поля различаются именами. Если в таблице предусмотрено поле (совокупность полей), значение которого уникально идентифицирует каждую запись, то его называют ключевым.

Типы полей задаются разработчиком; часто используемые типы: числовой, символьный, логический, дата. Тип поля определяет множество значений, которое может принимать данное поле в различных записях.

Запросы. В СУБД запросы служат для выборки записей, обновления таблиц и включения в них новых записей. С помощью запросов можно просматривать данные из нескольких таблиц. Они также используются в качестве способа получения данных для форм и отчетов. Но главное предназначение запросов — это отбор данных на основании критериев и математическая обработка данных (вычисляемые поля). В любой момент можно выбрать из БД необходимую информацию и создать вычисляемое поле.

Формы. Они предназначены для ввода данных в таблицу, для открытия других форм и отчетов (кнопочные формы), а также с их помощью можно ограничить объем информации, доступной пользователям, обращающимся к БД (маска). Другими словами, форма представляет собой бланк, подлежащий заполнению, или маску, накладываемую на набор данных. Большая часть данных, представленных в форме, берется из таблицы или запроса. Другая информация, не связанная ни с таблицей, ни с запросом, хранится в макете формы (например, кнопки, вычисляемые поля и т. п.).

Отчеты. Они служат для отображения итоговых данных из таблиц и запросов в удобном для просмотра виде. В отчетах, так же, как и в формах, часть данных берется из таблиц и запросов, другая часть информации хранится в макете отчета. (*Примечание.* Формы и отчеты только для Access являются частью базы данных.)

Табличная форма хранения данных позволяет легко осуществить поиск нужной информации. Поиск осуществляется с помощью запросов или с помощью фильтра, если надо напечатать только нужную информацию из таблицы. **Фильтры** позволяют отобрать записи, которые удовлетворяют определенным условиям. Если условие поиска задается для одного поля, то используют простой фильтр. Если требуется найти информацию, по составному условию для нескольких полей, то используется составной фильтр. Удалить данные в базе можно созданием запроса на удаление данных по заданному условию, простому или составному.

Часто возникает необходимость упорядочивания записей в результатах запросов. Эта операция называется **сортировкой**. Она производится также по какому-либо набору полей таблицы. В процессе сортировки структура записей сохраняется, т. е. строки таблицы перемещаются целиком. Поля, по значениям которых производится сортировка, называют ключом сортировки.

В качестве условий для производимых операций в базах данных используются логические выражения: простые (содержащие операции отношения) и составные (содержащие логические операции):

Операции отношения	Способ записи
Равно	=
Не равно	o
Меньше	<
Меньше или равно	<=
Больше	>
Больше или равно	>=
Логические операции	Знак операции
Операция логического умножения (конъюнкция)	И (AND)
Операция логического сложения (дизъюнкция)	ИЛИ (OR)
Операция отрицания	НЕ (NOT)

Билет 18

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

Электронные таблицы (табличный процессор) — это работающая в диалоговом режиме программа хранения и обработки данных, представленных в прямоугольных таблицах.

Структура электронной таблицы. Табличный процессор Excel фирмы Microsoft реализован в виде **рабочей книги** (самая крупная единица данных), которая состоит из прямоугольных таблиц (**рабочих листов**). Листам можно давать произвольные имена (по умолчанию Лист 1, Лист 2 и т. д.). Лист разбит на **строки** и **столбцы**. Заголовки столбцов обозначены буквами от А до IV, строки пронумерованы. Всего может быть 256 столбцов и 65 536 строк. Место пересечения строки и столбца называют **ячейкой**. Ячейка — основная единица хранения данных. **Адрес ячейки** (ссылка на ячейку) образуется из имени столбца и номера строки: А1, В1, D3, E5, АК10454 и т. п. Выделенная на экране ячейка является **активной**. Данные можно вводить только в активную ячейку, их можно видеть также в строке формул. Активную ячейку можно выделить щелчком мышью. Кроме того, можно работать с **диапазоном** (фрагментом, блоком) таблицы, который задается адресом верхней левой и нижней правой ячеек.

Типы данных: числа, формулы, текст. В работе с электронными таблицами можно выделить три типа данных: **число, текст, формулу**. Любой набор символов из букв, цифр и пробелов в ячейке воспринимается как текст и выравнивается по левому краю. Числа в электронных таблицах могут быть записаны в *обычном* или *экспоненциальном формате*. В обычном, если значение помещается в ячейку, например: -192,5. Если число слишком велико или мало, то в экспоненциальном: -1,925E+02 (мантисса, порядок). Числа по

умолчанию выравниваются по правому краю. Существуют и специализированные форматы, которые можно выбрать из меню **Формат, Ячейки**. Текст и числа рассматриваются как константы. Изменить их можно путем редактирования ячеек.

Электронные таблицы (ЭТ) позволяют автоматически проводить пересчет значений формул при изменении исходных данных, используемых в этих формулах, и предназначены для математических, финансовых, статистических расчетов, ведения простейших баз данных и построения диаграмм для наглядного воспроизведения данных.

Правила записи формул. Формула начинается со знака равенства и может содержать числа, адреса ячеек и знаки арифметических операций, выполняемых в порядке приоритета. Примеры формул:

$=A2*B3^2$ — содержимое ячейки A2 умножается на квадрат содержимого ячейки B3;

$-(D4-F5)/5$ — разность содержимого ячеек D4 и F5 делится на константу 5.

При вводе формулы в режиме отображения значений в ячейке отображается не сама формула, а результат вычислений по ней. Сама формула отображается в строке формул при активизации ячейки. При изменении данных в ячейках, имена которых входят в формулу, результат автоматически пересчитывается.

Редактирование данных в ячейках. Это удаление содержимого ячейки, отмена последнего изменения, перемещение данных, копирование данных, замена содержимого ячейки целиком, исправление данных в ячейке (двойной щелчок левой кнопкой мыши в редактируемой ячейке или в строке формул).

Основные встроенные функции. Функции — это заранее подготовленные формулы, к которым надо обращаться по заданному имени и с определенным количеством и типом данных. Microsoft Excel предлагает более 300 функций. Вызов функции осуществляется либо через текстовое меню **Вставка, Формула**, либо щелчком на кнопке *fx*.

Часто используемые функции:

Имя функции	Назначение функции
СУММ Пример: =СУММ(A1:A5) равносильно формуле: =A1+A2+A3+A4+A5	Суммирует аргументы
МАКС Пример: =МАХ(A1:A20)	Возвращает максимальное значение из списка аргументов
СРЗНАЧ Пример: =СРЗНАЧ(A1:F1) равносильно формуле: =(A1+B1+C1+D1+F1)/5	Возвращает среднее арифметическое аргументов
СЧЕТ Пример: =СЧЕТ(A1:A20)	Подсчитывает количество чисел в списке аргументов

Абсолютные и относительные ссылки. В формулах используют ссылки на адреса ячеек. Существуют два основных типа ссылок: относительная и абсолютная. Относительная используется в формуле для указания адреса ячейки, зависящего от адреса ячейки, в которой находится формула. При перемещении или копировании формулы из активной ячейки относительные ссылки автоматически обновляются в зависимости от нового положения формулы. Если формулу $=B1+B2$ поместить в ячейку B4, то Excel интерпретирует формулу как «прибавить содержимое ячейки, расположенной тремя строками выше, к содержимому ячейки, расположенной двумя строками выше». Абсолютную ссылку используют, когда нужно указать точное положение ячейки в таблице. Абсолютная ссылка не изменяется при ее переносе в другую ячейку. Для записи абсолютной ссылки используют знак \$ (\$A\$3). Используют и смешанную ссылку, когда изменяется один параметр — либо столбец, либо строка (\$A3, A\$3).

В Excel любой ячейке можно присвоить имя: выбрать команду **Вставка, имя, определить**, и на экране появится диалоговое окно с полем ввода, где необходимо ввести имя ячейки. Именованную ячейку удобно впоследствии исполь-

зовать в формулах, так как можно заменить адрес ячейки, ничего не говорящей о ее содержимом, на мнемоническое имя.

В Excel имеется множество *дополнительных возможностей*, ускоряющих и облегчающих работу с таблицами:

- автозаполнение данными;
- автоматический подбор высоты строк и ширины столбцов;
- объединение ячеек и перенос слов в ячейке;
- наложение защиты на лист и книгу;
- возможность сортировки данных таблицы;
- удобная справочная система;
- использование векторного редактора;
- использование буфера обмена.

Графическое представление данных. При решении аналитических задач, задач планирования, построении математических моделей очень важна наглядность представления данных. Здесь незаменимым средством являются диаграммы. «Диаграмма — средство графического представления количественной информации, предназначенное для сравнения значений величин или нескольких значений одной величины, отслеживания изменений значений» [6]. Для построения диаграммы в среде табличного процессора следует указать блок данных, по которым будет строиться диаграмма, и определить последовательность выбора данных из блока, все остальное программа сделает автоматически. Наиболее часто используемые типы диаграмм: гистограмма, график и круговая.

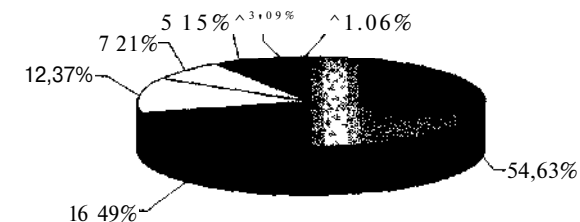
Пример

При проведении социологического опроса выпускники школ отвечали на вопрос «Какой фактор является для вас наиболее значимым при выборе профессии, и, соответственно вуза?» (журнал «Обучение и карьера», № 4, 2006) [5].

Была получена таблица:

Фактор	Процент выбравших данный фактор
Интерес к определенной деятельности	54,63
Престиж и доходность выбираемой специальности	16,49
Возможность последующего трудоустройства	12,37
Другое	7,21
Предназначение	5,25
Возможности родителей	3,09
Советы родителей и друзей	1,06

Диаграмма в Excel:



- Интерес к определенной деятельности
- Престиж и доходность выбираемой специальности
- Возможность последующего трудоустройства
- Другое
- Предназначение
- Возможности родителей
- Советы родителей и друзей

Билет 19

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

Одной из наиболее полезных возможностей, предоставляемых современными компьютерами, является возможность использования его для автоматизированного обмена информацией с другими компьютерами по линиям связи. Реализуется это с помощью компьютерных сетей. Под **компьютерной телекоммуникационной (вычислительной) сетью** понимается программно-аппаратный комплекс, обеспечивающий автоматизированный обмен данными между компьютерами по линиям связи. Современная сеть, состоящая из компьютеров (возможно специализированных), представляет собой комплекс узлов и каналов связи — аппаратуры и программ, обеспечивающих прием и передачу данных.

Основными характеристиками технологий обмена данными в сетях являются максимальная пропускная способность, количество объединяемых в сеть компьютеров и максимальное расстояние, на котором возможен обмен данными.

По этим характеристикам среди технологий выделяют:

1. **Локальные сети.** Это сети с потенциально ограниченным числом компьютеров и/или небольшой длиной линий связи. Такие сети обладают высокой скоростью при сравнительно небольшой цене, но могут увеличиваться только до определенных пределов.
2. **Территориальные (региональные) сети.** Это сети с потенциально неограниченным числом компьютеров, но сравнительно небольшой длиной линий связи (т. е. расположенные на сравнительно небольшой территории — одного города, например).
3. **Глобальные сети.** Это сети, объединяющие большие территории (континенты, например). Глобальные сети, как правило, создаются крупными организациями для предо-

ставления услуг связи. В них используют высокоскоростные специализированные линии дальней связи. Такие сети организуют взаимодействие между более мелкими территориальными сетями.

Большая часть возможностей, обеспечиваемых современными телекоммуникационными сетями, опирается на то, что они могут обмениваться данными между собой, создавая межсетевую среду. Самое крупное такое объединение сетей — это межсетевая среда Интернет.

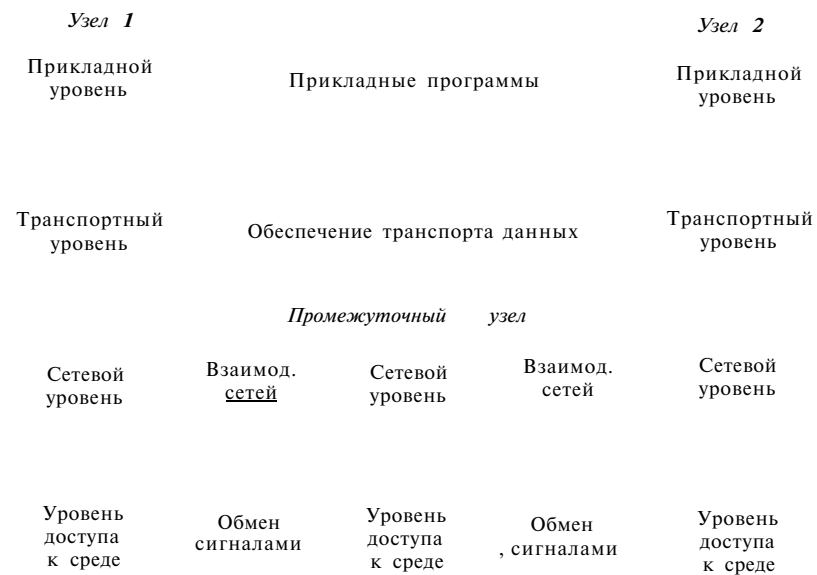
Возможности, реализованные в современных сетях:

- открытость, т. е. возможность разработки различных сетевых приложений, в том числе не предусмотренных ранее;
- использование для обмена данными сетей на базе различных технологий, с самыми разными каналами связи;
- возможность подключения новых абонентов и новых сетей, а также расширения существующих без принципиальной перестройки;
- возможность обеспечения автоматического перепланирования схемы обмена (изменение маршрутов) при возникновении технической необходимости (например, отказе канала связи);
- контроль обмена данными и минимизация потери в случае возникновения ошибок.

Основным принципом, лежащим в основе современных телекоммуникационных сетей, является принцип **пакетной коммутации**. Он состоит в том, что для доставки данные разбиваются на независимые фрагменты (пакеты), каждый пакет снабжается служебной информацией и передается отдельно от других пакетов. Итоговое сообщение (последовательность пакетов) восстанавливается при необходимости в конечной точке.

Основой конструирования и функционирования современных сетей являются их модели. **Модель сети** — это схема разделения функций между компонентами сети, определяющая основы их взаимодействия.

В основе Интернета лежит сетевая модель DOD (Department of Defence, министерство обороны США). Эта модель подразумевает, что все функции сети делятся на четыре уровня — от непосредственных клиентских программ до средств обмена сигналами. За каждым уровнем закреплены определенные задачи, выполняя их, уровень обменивается данными только с выше- и нижележащим уровнями.



С логической точки зрения, каждый уровень одного узла посылает данные такому же уровню другого узла.

За каждым уровнем закреплены следующие функции:

1. За уровнем доступа к среде — функции приема и передачи сигналов, преобразования их в цифровую форму и/или перекодирование. Этот уровень характеризует каждую сетевую технологию, применяемую для создания отдельной сети. Именно здесь осуществляются фактические прием и передача сигнала.

2. За сетевым уровнем закреплены функции организации межсетевого взаимодействия, он связывает отдельные сети. Протоколы этого уровня определяют маршруты следования пакетов (т. е. выполняют операцию перенаправления пакетов из одной сети в другую) и обмениваются необходимой технической информацией о пересылке. Узлы, выполняющие пересылку, называются шлюзами, выполняющие обработку пакетов и маршрутизацию — маршрутизаторами. У таких специализированных узлов могут быть не задействованы другие уровни, кроме первых двух).
3. За транспортным уровнем — задачи транспорта данных. Этот уровень связывает между собой программы, запущенные на конкретных узлах, т. е. позволяет определить, какая конкретно программа должна обработать полученный поток данных или обеспечить программе возможность отправки данных. Некоторые протоколы этого уровня позволяют также определять, получены ли данные.
4. К прикладному уровню отнесены все программы, взаимодействующие с пользователем и формирующие поток данных для обмена.

Основным набором протоколов, реализующих эту модель и обеспечивающим работу среды Интернет, является стек **протоколов TCP/IP**. Он предусматривает механизм, позволяющий разрабатывать различные сетевые приложения и использовать для их работы существующую инфраструктуру без модернизации. На основе этого стека протоколов уже разработано большое количество программ, реализующих разнообразные сервисы (службы, программы для предоставления специализированных услуг).

Большинство телекоммуникационных служб предполагают обмен данными между двумя основными типами программ: **клиентом** (программа, взаимодействующая с пользователем) и **сервером** (программа, обслуживающая запросы). Клиент и сервер могут располагаться на разных узлах сети, что и позволяет выстраивать обмен различными данными.

В настоящее время наиболее популярны следующие сетевые службы.

1. **Служба электронной почты.** Этот сервис позволяет обмениваться сообщениями в асинхронном режиме, т. е. отправить сообщение можно, даже если адресат не работает с сетью в момент отправки и получит его только через некоторое время. Основу инфраструктуры электронной почты составляют почтовые отделения — серверы, принимающие почту и обрабатывающие ее (либо пересылая, либо помещая в хранилище до обращения пользователя). Индивидуальный «раздел» пользователя в такой системе называется **почтовым ящиком**. Для обращения к содержимому своего почтового ящика или отправки новой почты пользователь должен использовать специальную программу-клиент (почтовую программу). Во многих случаях этот клиент реализован как Web-приложение.
2. **Служба World Wide Web (Всемирная паутина).** В этой службе информация представляется в виде гипертекста (см. билет 15). Для описания объектов и связей между ними применяется специальный язык разметки гипертекста — HTML. Основу функционирования этой службы составляют программы, выдающие такие объекты-страницы (и/или их составляющие) по запросу клиента (Web-серверы), и программы, демонстрирующие полученные страницы (браузеры). Данную службу можно применять как универсальное средство для создания специализированных приложений. В качестве примера таких приложений можно привести форумы (приложения, в которых посетители обмениваются мнениями по различным вопросам, оставляя в выделенных разделах, хранящихся в БД на серверах, свои комментарии), чаты (приложения, позволяющие организовать беседу в режиме реального времени, непосредственно передавая всем пользователям отправляемые каждым сообщения). *
3. **Телеконференции и Интернет-телефония.** Эти сервисы опираются на возможность передавать по сети потоки (т. е. не сообщения фиксированной длины, а некоторое количество данных за определенное время) аудио- и видеоинформации. С помощью телеконференций организуется обсуждение в режиме реального времени (это проще и эффективнее во многих случаях чатов и пр.), а с помо-

щью Интернет-телефонии — передача звуковой информации телефонной сети между узлами Интернета в виде потока IP-пакетов. Такими узлами могут быть специальные телефонные станции или просто компьютеры. Передача такого потока обходится значительно дешевле использования междугородного канала телефонной связи.

Служба WWW, выполняя роль универсального интерфейса к большей части современных сетевых служб, позволяет добиться схождения в представлении информации из разных источников.

Тем не менее, поскольку сам язык подготовки страниц ориентирован на оформление текста, а не на отражение его логической структуры, и в каждом конкретном случае авторы отдельных наборов страниц сами принимают решение о том, что и как отображать на своих страницах, не уведомляя об этом никого, то возникает проблема поиска информации в среде Интернет. С ростом общего количества страниц и объема представленной в такой форме информации, эта проблема становится все острее.

Для **поиска нужной информации в среде Интернет** применяют несколько способов.

1. **Применение классификации и каталогизации.** В соответствии с этим подходом создаются специальные ресурсы-рубрикаторы, на которых аннотированные ссылки разносят по некоторым категориям. Поиск информации в этом случае осуществляется постепенным уточнением области до тех пор, пока набор страниц не будет сокращен до обозримого минимума. Но сам поисковый рубрикатор пополняется не автоматически, а с помощью людей. В результате в таких рубрикаторах не упоминается подавляющее большинство страниц.
2. **Применение методов полнотекстового поиска и поиска по ключевым словам.** В этом случае автоматизированными средствами готовится поисковый индекс — фактически база данных, содержащая информацию о наполнении страниц. Пользователь, обращаясь к этому индексу, просит найти страницу, на которой содержатся определенные сло-

ва. Постепенно уточняя область поиска, снова получаем обозримый набор документов. Поисковый индекс формируется в основном автоматически, что резко увеличивает количество участвующей в поиске информации. К сожалению, это также увеличивает и количество «мусора», попадающего в результаты обработки запросов. Для борьбы с этим применяют систему ранжирования результатов на основе *релевантности* — некоторого вычисленного коэффициента соответствия найденной страницы запросу.

3. *Поиск информации в специализированных источниках.* Если точно известна область поиска, то вполне возможно, что существуют специализированные хранилища информации, имеющие специально разработанные поисковые системы. Такая система позволит найти нужную информацию существенно быстрее, чем использование поисковых систем общего назначения [5].

Билет 20

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

Понятие модели. Информационная модель

Моделирование является одним из ключевых видов деятельности человека и всегда в той или иной форме предшествует другим ее видам. Конечный этап моделирования — принятие решения.

Моделью принято считать материальный или мысленно представляемый объект, который замещает объект-оригинал с целью его исследования, сохраняя некоторые важные для данного исследования типичные черты и свойства оригинала, т. е. его существенные стороны [5].

Модель, как правило, доступнее для исследования, чем реальный объект (например, такой, как экономика страны,

Солнечная система, ядерный взрыв, извержение вулкана и т. п.). Модель также позволяет научиться управлять объектом или понять закономерности его изменения, что важно в тех случаях, когда экспериментировать с объектом бывает неудобно, трудно или невозможно (например, когда эксперимент имеет большую продолжительность или когда существует риск привести объект в нежелательное или необратимое состояние).

Таким образом, мы можем сделать вывод, что модель необходима для того, чтобы:

- понять, как устроен конкретный объект — каковы его структура, основные свойства, законы развития и взаимодействия с окружающим миром;
- научиться управлять объектом или процессом и определить наилучшие способы управления при заданных целях и критериях (оптимизация);
- прогнозировать прямые или косвенные последствия реализации заданных способов и форм воздействия на объект.

Интересно, что хорошо построенная модель обладает удивительным свойством: ее изучение может дать некоторые новые знания об объекте-оригинале.

Адекватность модели объекту предполагает воспроизведение моделью с необходимой полнотой всех характеристик объекта, *существенных для целей моделирования*. Один и тот же объект может иметь множество различных моделей, и одна и та же модель может описывать разные объекты.

Процесс построения и изучения модели называется **моделированием**.

Существуют материальные и информационные модели. **Материальные модели** воспроизводят физические, геометрические и другие свойства объекта. Примеры: глобус, скелет, макеты зданий и мостов, модели самолетов, кораблей, автомобилей.

Предметом изучения информатики являются информационные модели. **Информационные модели** представляют объекты в образной или знаковой форме.

Виды информационных моделей

Объектом информационного моделирования могут быть физические (падение тел), химические (реакции горения), биологические (фотосинтез в листьях растений) процессы, метеорологические явления (гроза, торнадо), экономические (падение курса валюты), социальные (миграция, рост населения) процессы и т. д. *Знаковая информационная модель* может быть представлена в виде текста (программа на языке программирования), формулы (второй закон Ньютона $F = ma$), таблицы (Периодическая система элементов Д. И. Менделеева), карты, схемы, чертежа (используется язык графических элементов). Естественные языки используются для создания *описательных информационных моделей* (гелиоцентрическая модель мира Коперника). С помощью формальных языков строятся *формальные информационные модели* (математические, логические). Модели, построенные с использованием математических понятий и формул, называются *математическими моделями*. В физике рассматривается много различных функциональных зависимостей, выраженных на языке алгебры, которые представляют собой математические модели изучаемых явлений или процессов. *Информатика изучает общие методы и средства создания и использования информационных моделей.*

Назначение информационных моделей чаще всего состоит в получении данных, которые могут быть использованы для подготовки и принятия решений экономического, социального, организационного или технического характера, для достижения наилучших показателей деятельности моделируемого объекта.

Моделируемый объект можно рассматривать как систему. **Система** — это сложный объект, состоящий из взаимосвязанных частей (элементов) и существующий как единое целое. Всякая система имеет определенное назначение (функцию, цель). Структура — это совокупность связей между элементами системы, т. е. внутренняя организация системы.

Для отражения состояния систем используются статические и динамические модели.

Модели, описывающие состояние системы в определенный момент времени, называются *статическими информационными моделями*. Примеры: модель строения молекулы, модель строения Солнечной системы, «Система природы» К. Линнея.

Модели, описывающие процессы изменения и развития систем, называются динамическими информационными моделями. Примеры: модели процесса протекания химической реакции, ядерной реакции, движения тел, развития организмов и популяций.

Для отражения систем с различными структурами используются различные виды информационных моделей.

- **Табличные модели** применяются для описания свойств объектов, обладающих одинаковыми наборами свойств. Могут быть динамическими и статическими. Свойства объекта представлены в виде списка, а их значения размещаются в ячейках прямоугольной таблицы (Периодическая система элементов Д. И. Менделеева).
- **В иерархических моделях** объекты распределены по уровням. Каждый элемент более высокого уровня может состоять из элементов нижнего уровня, а элемент нижнего уровня может входить в состав только одного элемента более высокого уровня (генеалогическое дерево, классификация компьютеров).
- **Сетевые модели** применяются для отражения таких систем, в которых связи между элементами имеют сложную структуру (сеть Интернет, телефонная сеть, процесс передачи мяча в коллективной игре, например в футболе). Могут быть статическими и динамическими [5].

Реализация информационных моделей на компьютере

Рассмотрим **основные этапы компьютерного моделирования**.

1. **Этап постановки задачи** характеризуется описанием моделируемого объекта в общем виде, определением конечного результата моделирования и имеющихся условий (возможно, ограничений).

- 2. Определение цели моделирования.** От выбранной цели зависит, какие характеристики объекта моделирования считать существенными, какие методы лучше подходят для решения данной задачи.
- 3. Анализ объекта моделирования для выделения существенных свойств с точки зрения цели моделирования.** Нет единственно правильного способа выделения существенных свойств объекта моделирования, поскольку объекты моделирования серьезно отличаются. Это может быть материальный объект, некая сложная информационная система, промышленный процесс и пр. Иногда необходимые свойства объекта могут быть очевидны, а иногда придется перебрать множество вариантов, прежде чем будет достигнута цель моделирования. Поэтому к этапам п. 2 и п. 3 можно возвращаться многократно. Адекватность модели объекту моделирования будет зависеть еще и от того, как выделенные существенные свойства мы сможем формализовать, т. е. в какой форме их отобразим.

Примечание. Иногда все эти три этапа объединены в этап постановки задачи.

- 4. Формализация** (определение и приведение к выбранной форме). Важный этап моделирования, влияющий на результат. От выбранной формы представления данных зависит, насколько точен будет конечный результат, в какой степени построенная модель будет соответствовать объекту. Формами представления моделей могут быть: словесное описание, чертеж, таблица, формула, схема, алгоритм, компьютерная программа и т. п.

Итак, форма представления модели определена, и данные формализованы для обработки. Конечной целью этого этапа является создание **информационной модели**.

- 5. Разработка компьютерной модели для проведения эксперимента:**

- создание математической или имитационной модели для исследования с помощью компьютера;
- проверка данных и условий на непротиворечивость;
- планирование эксперимента.

Существует множество программных комплексов, которые позволяют строить и исследовать самые разные по назначению модели. Разнообразное программное обеспечение позволяет преобразовать исходную информационную модель в компьютерную и провести компьютерный эксперимент.

В процессе разработки компьютерной модели исходная информационная модель будет претерпевать некоторые изменения по форме представления, так как должна ориентироваться на определенную программную среду и инструментарий. После внесения уточнений необходимо перепроверять данные и условия на непротиворечивость. В результате проверки может возникнуть необходимость возврата к п. 2 или п. 3.

План эксперимента должен четко отражать последовательность работы с моделью.

6. Компьютерный эксперимент:

- исследование модели;
- анализ полученных результатов **на соответствие цели** моделирования;
- уточнение модели.

Компьютерный эксперимент включает в себя исследование модели в соответствии с поставленной целью.

Конечная цель моделирования — принятие решения, которое должно быть выработано на основе всестороннего анализа результатов моделирования. Анализ полученных результатов на соответствие цели моделирования — решающий этап для дальнейшей работы. Если результаты не соответствуют целям поставленной задачи, значит, на предыдущих этапах были допущены ошибки. Если такие ошибки выявлены, то требуется уточнение (корректировка) модели. Корректировка может быть незначительной в рамках проведения самого эксперимента или существенной, требующей возврата к предыдущим этапам [4].

Процесс повторяется до тех пор, пока результаты не будут удовлетворять цели моделирования и их можно будет использовать для принятия решений.

Пример применения электронной таблицы моделирования в качестве инструмента моделирования

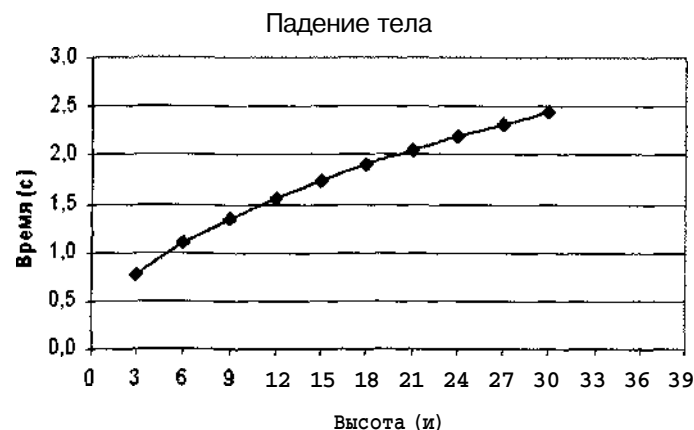
Задача. Определить экспериментально зависимость времени падения тела от начальной высоты.

Решение. Зависимость между величинами может быть представлена в виде *математической формулы* $t = \sqrt{2H/g}$ (без учета сопротивления воздуха), где g — ускорение свободного падения (константа).

Результаты эксперимента — бросания шарика с высоты представим в таблице:

H , м	J	6	12	18	24	30
t , с	J^LJ	1,1	1,3	1,5	1,7	1,9

По результатам, помещенным в таблицу, можно построить в электронных таблицах график зависимости времени падения t от начальной высоты падения H .



На данном примере рассмотрены три способа отображения зависимости между величинами: аналитический (математическая формула), табличный и графический [5].

1. Бешенков С.А., Ракитина Е.Н. Информатика. Систематический курс : Учебник для 10 класса. — М.: БИНОМ. Лаборатория знаний, 2006. — 432 с: ил.
2. Гусева А. И. Учимся информатике: задачи и методы их решения. — М. : Диалог-МИФИ, 2001.
3. Кушниренко А. Г., Леонов А. Г., Эпиктетов М.Г., Борисенко В. В., Кузьменко М.А., Назаров Б. А., Ханжин С. Б. Информационная культура: Кодирование информации. Информационные модели: 9-10 классы. 2-е изд. — М. : Дрофа, 1996.
4. Самылкина Н.Н., Калинин И.А., Левченко И. В. Методические рекомендации по использованию плакатов «Информатика и ИКТ. Основная школа». — М. : БИНОМ. Лаборатория знаний, 2005.
5. Самылкина Н.Н., Калинин И. А., Островская Е.М. Материалы для подготовки к экзамену по информатике. — М. : БИНОМ. Лаборатория знаний, 2006.
6. Семакин И. Г. и др. Информатика : Базовый курс : Учебники для 8 и 9 кл. — М. : БИНОМ. Лаборатория знаний, 2007.
7. Радченко Н.П., Козлов О.А. Школьная информатика: экзаменационные вопросы и ответы. — М. : Финансы и статистика, 1998.
8. Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27 июля 2006 г. № 149-ФЗ.
9. Федеральный закон «О почтовой связи» от 17 июля 1999 г. № 176-ФЗ.
10. Федеральный закон «О персональных данных» от 27 июля 2006 года № 152-ФЗ.
11. <http://ru.wikipedia.org/>