

" Предмет и основные понятия информатики"

Информатика — это комплексная, техническая наука, которая изучает и систематизирует законы и приемы создания, сохранения, воспроизведения, получения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ними.

Термин "*информатика*" происходит от французского слова *Informatique* и образован из двух слов: информация и автоматика. Термин *информация* произошел от латинского слова *informatio* — разъяснение, осведомление. Этот термин введен во Франции в середине 60—х лет XX ст., когда началось широкое использование вычислительной техники. Тогда в англоязычных странах вошел в употребление термин "*Computer Science*" для обозначения науки о преобразовании информации, которая базируется на использовании вычислительной техники. Теперь эти термины являются синонимами.

Появление информатики обусловлено возникновением и распространением новой технологии сбора, обработки и передачи информации, связанной с фиксацией данных на машинных носителях.

Предмет информатики как науки составляют:

- аппаратное обеспечение средств вычислительной техники;
- программное обеспечение средств вычислительной техники;
- средства взаимодействия аппаратного и программного обеспечения;
- средства взаимодействия человека с аппаратными и программными средствами.

Средства взаимодействия в информатике принято называть **интерфейсом**. Поэтому средства взаимодействия аппаратного и программного обеспечения иногда называют также **программно-аппаратным интерфейсом**, а средства взаимодействия человека с аппаратными и программными средствами — **интерфейсом пользователя**.

Задачи информатики как науки составляют:

- изучить структуру, общие свойства информации, исследовать законы и методы создания, преобразования, накопления, передачи и использования информации.
- систематизация приемов и методов работы с аппаратными и программными средствами вычислительной техники. Цель систематизации состоит в том, чтобы выделять, внедрять и развивать передовые, более эффективные технологии автоматизации этапов работы с данными, а также методически обеспечивать новые технологические исследования.

В составе основной задачи сегодня можно выделить такие основные направления информатики для практического применения :

- архитектура вычислительных систем (приемы и методы построения систем, предназначенных для автоматической обработки данных);
- интерфейсы вычислительных систем (приемы и методы управления аппаратным и программным обеспечением);
- программирование (приемы, методы и средства разработки комплексных задач);
- преобразование данных (приемы и методы преобразования структур данных);
- защита информации (обобщение приемов, разработка методов и средств защиты данных);
- автоматизация (функционирование программно—аппаратных средств без участия человека);
- стандартизация (обеспечение совместимости между аппаратными и программными средствами, между форматами представления данных, относящихся к разным типам вычислительных систем).

На всех этапах технического обеспечения информационных процессов для информатики ключевым вопросом есть эффективность. Для аппаратных средств под эффективностью понимают соотношение производительности оснащение к его стоимости. Для программного обеспечения под эффективностью принято понимать производительность работающих с ним пользователей. В программировании под эффективностью понимают объем программного кода, созданного программистами за единицу времени. В информатике все жестко ориентированно на эффективность. Вопрос как осуществить ту или другую операцию, для информатики важный, но не основной. Основным есть вопрос — как совершить данную операцию эффективно.

В рамках информатики, как технической науки можно сформулировать понятия информации, информационной системы и информационной технологии.

Данные

Данные (data)— это такое же первоначальное понятие, как, скажем, в математике "точка": попытка дать определение начальным понятиям приводит к необходимости дополнительно определять

использованные термины. Итак, будем считать, что данные — это те или иные сведения (необязательно несущие смысловую нагрузку).

Пример данных:

812, 930, 944.

(для человека это ничего не значит, если нет объяснения, что обозначают эти цифры).

01000001 01101100 01101100 01100001

(для человека это ничего не значит, но в ASCII – это слово Alla).

Во время информационного процесса данные преобразовываются из одного вида в другого с помощью методов. Обработка данных включает в себя множество разных операций. Основными операциями есть:

- сбор данных — накопление информации с целью обеспечения достаточной полноты для принятия решения;
- формализация данных — приведение данных, которые поступают из разных источников к единой форме;
- фильтрация данных — устранение лишних данных, которые не нужны для принятия решений;
- сортировка данных — приведение в порядок данных за заданным признаком с целью удобства использования;
- архивация данных — сохранение данных в удобной и доступной форме;
- защита данных — комплекс мер, направленных на предотвращение потерь, воспроизведения и модификации данных;
- транспортирование данных — прием и передача данных между отдаленными пользователями информационного процесса. Источник данных принят называть сервером, а потребителя — клиентом;
- преобразование данных — преобразование данных с одной формы в другую, или с одной структуры в другую, или изменение типа носителя.

Информация

Информация (information)— это данные, сопровождающиеся смысловой нагрузкой.

При этом, очевидно, то, что *для одних* является данными, *для других* вполне может быть информацией. Но всегда можно точно сказать, что нужно предпринять для того, чтобы те или иные данные стали информативными для наибольшей аудитории: их нужно снабдить смысловым содержанием. Чем более полным будет это содержание, тем более информативной будет соответствующее сообщение.

Информация которая:

воспринимается из окружающей среды называется *входной* информацией,

выдается в окружающую среду называется *исходная* информацией.

Информация существует в виде документов, чертежей, рисунков, текстов, звуковых и световых сигналов, электрических и нервных импульсов и т.п..

Важнейшие свойства информации:

- объективность и субъективность;
- полнота;
- достоверность;
- адекватность;
- доступность;
- актуальность.

Пример информации: 812 рублей, 930 рублей, 944 рубля.

Более информативное сообщение: 812 рублей, 930 рублей, 944 рубля — цены на бальзам после бритья.

Ещё более информативное: 812 рублей, 930 рублей, 944 рубля — цены на бальзам после бритья "Dune", 100 мл. в Москве.

В итоге мы имеем следующую простую формулу:

информация = данные + смысл

Знание

Знание — есть переживание, сравнённое с другими переживаниями это определение, которое дал Лосский Н. О. в своей работе "Обоснование интуитивизма", является наиболее точным.

Пример знания:

Цены на бальзам после бритья "Dune", 100 мл. 8 апреля 2000 г. (по Москве)

В салоне Christian Dior (Тверская ул.):	812 рублей;
В салоне Rivoli (Торговый комплекс на Манежной площади):	930 рублей;
В салоне Л'Этуаль (Рамстор на Молодёжной):	944 рубля.

Я думаю, что теперь без труда можно понять, какие знания будут более полными (более качественными) по сравнению с приведёнными: можно расширить список сравнения по другим торговым точкам, по другим товарам, ввести динамику цен по датам, привести данные не только в рублях, но и в валюте и т. д.

Сравнительная информация (знание) обладает теми качествами, на которые как раз указывали процитированные мною авторы. В частности, она позволяет принимать решения. Причём, если это покупатель, то он может принять решение о покупке товара по более подходящей для него цене (при этом могут подключаться знания о качестве сервиса — сравнение информации; знания о надёжности торговой компании — сравнение информации о том, где могут продавать качественную, а где — поддельную продукцию и т. д.). Если решение принимает представитель торговой компании, то знания о конкурентах могут ему позволить более выгодно вести бизнес. Важно подчеркнуть, что каждый раз знание получается именно на основе сравнения информации.

В итоге мы имеем следующую простую формулу:

знание = информация + сравнение
--

Информационная система

В информатике понятие "система" чаще используют относительно набора технических средств и программ. Системой называют также аппаратную часть компьютера. Дополнение понятия "система" словом "информационная" отображает цель ее создания и функционирования.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, используемая для сохранения, обработки и выдачи информации с целью решения конкретной задачи.

Современное понимание информационной системы предусматривает использование компьютера как основного технического средства обработки информации. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом информационной системы.

В работе информационной системы можно выделить следующие этапы:

1. Зарождение данных — формирование первичных сообщений, которые фиксируют результаты определенных операций, свойства объектов и субъектов управления, параметры процессов, содержание нормативных и юридических актов и т.п..
2. Накопление и систематизация данных — организация такого их размещения, которое обеспечивало бы быстрый поиск и отбор нужных сведений, методическое обновление данных, защита их от искажений, потери, деформирование целостности и др.
3. Обработка данных — процессы, вследствие которых на основании прежде накопленных данных формируются новые виды данных: обобщающие, аналитические, рекомендательные, прогнозные. Производные данные тоже можно обрабатывать, получая более обобщенные сведения.
4. Отображение данных — представление их в форме, пригодной для восприятия человеком. Прежде всего — это вывод на печать, то есть создание документов на так называемых твердых (бумажных) носителях. Широко используют построение графических иллюстративных материалов (графиков, диаграмм) и формирование звуковых сигналов.

Сообщения, которые формируются на первом этапе, могут быть обычным бумажным документом, сообщением в "машинном виде" или тем и другим одновременно. В современных информационных системах сообщения массового характера большей частью имеют "машинный вид". Аппаратура, которая используется при этом, имеет название средства регистрации первичной информации.

Потребности второго и третьего этапов удовлетворяются в современных информационных системах в основном средствами вычислительной техники. Средства, которые обеспечивают доступность информации для человека, то есть средства отображения данных, являются компонентами вычислительной техники.

Подавляющее большинство информационных систем работает в режиме диалога с пользователем. Типичные программные компоненты информационных систем включают: диалоговую подсистему ввода—вывода, подсистему, которая реализует логику диалога, подсистему прикладной логики обработки данных, подсистему логики управления данными. Для сетевых информационных систем важным элементом является коммуникационный сервис, обеспечивающий взаимодействие узлов сети при общем решении задачи. Значительная часть функциональных возможностей информационных систем закладывается в

системном программном обеспечении: операционных системах, системных библиотеках и конструкциях инструментальных средств разработки. Кроме программной составной информационных систем важную роль играет информационная составная, которая задает структуру, атрибутику и типы данных, а также тесно связана с логикой управления данными.

Информационные технологии

В широком смысле слово технология — это способ освоения человеком материального мира с помощью социально организованной деятельности, которая включает три компонента: информационную (научные принципы и обоснование), материальную (орудие работы) и социальную (специалисты, имеющие профессиональные навыки). Эта триада составляет сущность современного понимания понятия технологии.

Понятие информационной технологии появилось с возникновением информационного общества, основой социальной динамики в котором являются не традиционные материальные, а информационные ресурсы: знания, наука, организационные факторы, интеллектуальные способности, инициатива, творчество и т.д. К сожалению, это понятие настолько общее и всеохватывающее, что до сих пор специалисты не пришли к четкой, формализованной формулировке. Наиболее удачным определением понятия информационной технологии дано академиком Глушковым В.М., который трактовал ее как

человеко-машинную технологию сбора, обработки и передачи информации, которая основывается на использовании вычислительной техники.

Эта технология быстро развивается, охватывая все виды общественной деятельности: производство, управление, науку, образование, финансово—банковские операции, медицину, быт и др.

Основные представления теории информации

Понятие об информации и ее свойствах

Современная наука о свойствах информации и закономерностях информационных процессов называется **теорией информации**. Рассмотрим смысл понятия информация на примере двух подходов к измерению количества информации: подходов Хартли и Шеннона. Первый базируется на принципах теории множеств и комбинаторики, а для второго фундаментом служит теория вероятностей.

В основе всей теории информации лежит открытие, сделанное Р. Хартли в 1928 году, и состоящее в том, что информация допускает количественную оценку. К. Шеннон в 1948 году придал этой теории завершенность. Большой вклад в дальнейшее развитие и обобщение теории информации внесли отечественные ученые А.Н. Колмогоров, А.А. Харкевич, Р.Л. Стратанович. Сравнительно недавно исследователи советских архивов сообщили о том, что теория, известная сегодня как теория Шеннона, была создана А.Н. Колмогоровым еще в 1938 году, но была засекречена, так как использовалась в военных разработках.

Подход Р. Хартли

Подход Р. Хартли базируется на фундаментальных теоретико-множественных, по существу комбинаторных основаниях, а также некоторых интуитивно очевидных предположениях.

Итак, будем считать, что если существует множество элементов и осуществляется выбор одного из них, то этим самым сообщается или генерируется определенное количество информации. Эта информация состоит в том, что если до выбора не было известно, какой элемент будет выбран, то после выбора это становится известным.

Найдем вид функции, связывающей количество информации, получаемой при выборе некоторого элемента из множества, с количеством элементов в этом множестве, иначе говоря — с мощностью множества.

Если множество элементов, из которых осуществляется выбор, состоит из одного единственного элемента, то его выбор предопределен, т.е. никакой неопределенности выбора нет. Таким образом, если мы узнаем, что выбран этот единственный элемент, то, очевидно, при этом мы не получаем никакой новой информации, т.е. получаем нулевое количество информации.

Если множество состоит из двух элементов, то неопределенность выбора существует, но ее значение минимально. В этом случае минимально и количество информации, которое мы получаем, узнав, что совершен выбор одного из элементов.

Чем больше элементов содержит множество, тем больше неопределенность выбора, т.е. тем больше информации мы получаем, узнав о том, какой выбран элемент.

Из этих очевидных соображений следует первое требование: информация есть монотонная функция мощности исходного множества — чем больше элементов содержит множество, тем больше заключено в нем информации.

Обычно мы представляем числа в десятичной системе счисления, или системе счисления с основанием 10.

Например, целое число представляется как сумма степеней (от нулевой и выше) числа 10 с соответствующими коэффициентами от 0 до 9:

$$1365=1\cdot 10^3+3\cdot 10^2+6\cdot 10^1+5\cdot 10^0$$

В компьютерных технологиях опираются на представления чисел в двоичной (цифры 0 и 1) и шестнадцатеричной системах, в последней к цифрам 0123456789 добавлены "цифры" ABCDEF (10 11 12 13 14 15)

В двоичной системе целое число представляется как сумма степеней числа 2 с соответствующими коэффициентами от 0 до 1, а в шестнадцатеричной — как сумма степеней числа 16 с соответствующими коэффициентами от 0 до 15 (но вместо чисел от 10 до 15 используются буквы A — F).

Например:

Десятичное $13=1\cdot 2^3+1\cdot 2^2+0\cdot 2^1+1\cdot 2^0=1101$ в двоичной системе.

Десятичное $65=4\cdot 16^1+1\cdot 16^0=41$ в шестнадцатеричной системе.

Шестнадцатеричное $15AF=1\cdot 16^3+5\cdot 16^2+A\cdot 16^1+F\cdot 16^0=4096+1280+160+15=5551$ в десятичной системе.

Рассмотрим множество, состоящее из чисел в двоичной системе счисления длиной i двоичных разрядов. При этом каждый из разрядов может принимать значения только 0 и 1:

Кол-во двоичных разрядов (i)	Кол-во состояний, которое можно пронумеровать i -разрядными двоичными числами (N)
1	2 (0 1)
2	4 (00 01 10 11)
3	8 (000 001 010 011 100 101 110 111)
4	16 (0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111)
i	2^i (000...00 000...01 111...11)

Рассмотрим процесс выбора чисел из рассмотренного множества. До выбора вероятность выбрать любое число одинакова. Существует объективная неопределенность в вопросе о том, какое число будет выбрано. Эта неопределенность тем больше, чем больше N — количество чисел, а чисел тем больше — чем больше разрядность i этих чисел.

Примем, что выбор одного числа дает нам следующее количество информации:

$$i = \text{Log}_2(N).$$

Таким образом, количество информации, содержащейся в двоичном числе, равно количеству двоичных разрядов в этом числе.

Это выражение и представляет собой формулу Хартли для количества информации.

При увеличении длины числа в два раза количество информации в нем также должно возрасти в два раза, несмотря на то, что количество чисел во множестве возрастает при этом по показательному закону (в квадрате, если числа двоичные), т.е. если

$$N2=(N1)^2,$$

то

$$I2 = 2 * I1,$$

$$F(N1 * N1) = F(N1) + F(N1).$$

Это невозможно, если количество информации выражается линейной функцией от количества элементов во множестве. Но известна функция, обладающая именно таким свойством: это Log :

$$\text{Log}_2(N2) = \text{Log}_2(N1)^2 = 2 * \text{Log}_2(N1).$$

Это второе требование называется требованием *аддитивности*.

Таким образом, логарифмическая мера информации, предложенная Хартли, одновременно удовлетворяет условиям монотонности и аддитивности. Сам Хартли пришел к такому представлению меры информации на основе эвристических соображений, подобных только что изложенным. Но в настоящее время строго доказано, что логарифмическая мера для количества информации однозначно следует из этих двух постулированных им условий.

Минимальное количество информации получается при выборе одного из двух равновероятных вариантов. Это количество информации принято за единицу измерения и называется бит (от англ. binary digit — двоичная цифра, то есть два возможных значения).

Подход К. Шеннона

Клод Шеннон основывается на теоретико-вероятностном подходе. Это связано с тем, что исторически шенноновская теория информации выросла из потребностей теории связи, имеющей дело со статистическими характеристиками передаваемых сообщений и каналов связи.

Пусть существует некоторое конечное множество событий (состояний системы): $X=\{x_1, x_2, \dots, x_N\}$, которые могут наступать с вероятностями: $p(x_i)$, соответственно, причем множество вероятностей удовлетворяет естественному условию нормировки:

Исходное множество событий характеризуется некоторой неопределенностью, т.е. энтропией Хартли, зависящей, как мы видели выше, только от мощности множества. Но Шеннон обобщает это понятие, учитывая, что различные события в общем случае не равновероятны. Например, неопределенность системы событий: {монета упала "орлом", монета упала "решкой"}, значительно выше, чем неопределенность событий: {монета упала "орлом", монета упала "ребром"}, так как в первом случае варианты равновероятны, а во втором случае вероятности вариантов сильно отличаются:

$$H(X) = -\sum_{i=1}^N p(x_i) \text{Log}_2 p(x_i). \quad (4) \quad . 1)$$

Если измерять количество информации изменением степени неопределенности, то шенноновское количество информации численно совпадает с энтропией исходного множества:

$$I(X) = -\sum_{i=1}^N p(x_i) \text{Log}_2 p(x_i). \quad (4) \quad . 2)$$

4.2.3. СВЯЗЬ ФОРМУЛ К. ШЕННОНА И Р. ХАРТЛИ

Следуя [391], приведем вывод выражения Шеннона (4.2) непосредственно из выражения Хартли для количества информации: $I = \text{Log}_2(N)$.

Пусть события исходного множества мощности N равновероятны:

$$p(x_i) = \frac{1}{N},$$

тогда учитывая, что

$$\begin{aligned} \text{Log} \frac{1}{N} &= \text{Log} 1 - \text{Log} N = -\text{Log} N, \\ \sum_{i=1}^N \frac{1}{N} &= 1, \end{aligned}$$

непосредственно из формулы Хартли получаем

$$I(x) = -\sum_{i=1}^N \frac{1}{N} \text{Log}_2 \frac{1}{N} = -\sum_{i=1}^N p(x_i) \text{Log}_2 p(x_i).$$

Остается предположить, что это выражение верно и для случая, когда события неравновероятны [391]. В этом предположении и состоит обобщение Клода Шеннона, составившее целую эпоху в развитии современной теории информации.

В 1948 году Шеннон предложил следующий способ измерения количества информации. Пусть X - случайная величина, принимающая значения $x_1, x_2, x_3, \dots, x_n$ с вероятностью $p_1, p_2, p_3, \dots, p_n$, и Y - случайная величина, принимающая значения $y_1, y_2, y_3, \dots, y_n$ с вероятностью $q_1, q_2, q_3, \dots, q_n$. Тогда информация $I(X, Y)$ относительно Y , содержащаяся в X , определяется формулой

$$I(X, Y) = \sum_{ij} p_{ij} \log_2 \frac{p_{ij}}{p_i q_j}$$

где p_{ij} — вероятность совмещения событий $X = x_i$ и $Y = y_j$.

Свойства информации:

5. $I(X, Y) \geq 0$,
6. $I(X, Y) = 0$ при $p_{ij} = p_i q_j$ т.е. X, Y – независимые события,
7. $I(X, Y) = I(Y, X)$,
8. $I(X, Y) = H(X) + H(Y) - H(X, Y)$, где H – информационная энтропия,
 $H(X) = \sum p_i \log_2 (1/p_i)$, $H(Y) = \sum q_j \log_2 (1/q_j)$, $H(X, Y) = \sum p_{ij} \log_2 (1/p_{ij})$.

Величина энтропии показывает среднее число знаков, необходимых для различия (записи) возможных значений случайной величины. Это позволяет понять роль количества информации при хранении информации в запоминающих устройствах.

Если величины X, Y – независимые, то для записи X требуется в среднем $H(X)$ двоичных знаков, для записи Y – $H(Y)$, а для пары (X, Y) надо $H(X) + H(Y)$ двоичных знаков.

Если величины X, Y зависимы, то среднее число двоичных знаков оказывается меньше:
 $H(X, Y) = H(X) + H(Y) - I(X, Y)$.

Сравнение подходов Р. Хартли и К. Шеннона

Чрезвычайно важным и принципиальным является то обстоятельство, что для построения меры Хартли используется лишь понятие многообразия, которое накладывает на элементы исходного множества лишь одно условие (ограничение): должна существовать возможность отличать эти элементы один от другого.

В теории Шеннона существенным образом используется статистика, причем предполагается, что случайные события (состояния системы) распределены по нормальному закону.

Таким образом, различие между подходами Хартли и Шеннона к построению теории информации соответствует различию между непараметрическими и параметрическими методами в статистике.

Если говорить более конкретно, то, очевидно, что мера Шеннона асимптотически переходит в меру Хартли при условии, что вероятности всех событий (состояний) равны.

В статистике доказано фундаментальное свойство энтропии случайного процесса, состоящее в том, что при условии нормальности распределения и достаточно больших выборках все множество событий можно разделить на две основные группы:

- высоковероятные события (считающиеся заслуживающими изучения);
- маловероятные события (считаются не заслуживающими особого внимания).

Причем высоковероятные события с высокой точностью равновероятны. При увеличении размерности выборки доля "заслуживающих внимания" событий неограниченно уменьшается, и мера Шеннона асимптотически переходит в меру Хартли. Поэтому можно считать, что при больших нормально распределенных выборках мера Хартли является оправданным упрощением меры Шеннона.

Кодирование информации

Очевидно, что для кодирования информации достаточно использовать любые два различных значения, например, цифры 0 и 1 (да и нет, истина и ложь и т. п.). В этом случае применительно к кодировке чисел мы имеем дело с так называемой двоичной системой счисления. В общем случае, необходимо говорить о кодировке любых символов (не чисел!), не смешивая эти две разные задачи.

Для измерения количества информации используют 1 бит – один символ двухбуквенного алфавита, состоящего из нулей и единиц $\{0, 1\}$.

Кроме этого используются также следующие единицы:

- 1 байт=8 бит,
- 1024 байт=1 Кбайт,
- 1024 Кбайт=1 Мбайт,
- 1024 Мбайт=1 Гбайт.

Кодировка чисел

Целые числа в системе счисления с любым основанием (2, 10, 16 и т.д.) всегда представляются точно. Так, любое целое число представляется в виде суммы степеней числа, лежащего в основании системы счисления с соответствующими коэффициентами — от 0 до 9 в десятичной, 0 и 1 в двоичной, от 0 до F (15) в шестнадцатеричной системах. Для определенности остановимся на двоичном представлении.

При этом, как показано выше, чем большее количество двоичных разрядов отводится под хранение целого числа, тем больше возможный диапазон изменения его значений.

Минимальное количество используемых двоичных разрядов (применительно к компьютерным технологиям) равно 8, что составляет 1 байт.

Легко убедиться, что 8 двоичных разрядов позволяют закодировать числа от 0 до 255. Если же отвести один из разрядов под хранение знака числа, то те же 8 разрядов обеспечат возможность кодировки целых чисел от -128 до 127.

Для кодировки вещественных чисел, когда необходимо учесть и десятичную часть числа, используется особая форма представления — с плавающей точкой:

$$X = M * 2^P$$

Здесь M — так называемая мантисса, P — порядок.

При хранении чисел с плавающей точкой выделяется определенное количество разрядов под хранение мантиссы и под хранение порядка. Точность представления вещественного числа зависит от разрядности мантиссы, а возможный диапазон изменения — от количества разрядов, выделенных для хранения порядка.

Кодировка символов

Кодировка символов применительно к компьютерным приложениям базируется обычно на системе кодировки ASCII (American Standard Code for Information Interchange- Американский стандартный код для обмена информацией) — 256 различных, упорядоченных по коду (номеру) символов. Значения кода — целые числа от 0 до 255.

Для кодирования одного символа используется 8 битов, или 1 байт. В этом легко убедиться, если записать значения кодов в двоичной системе. Существует много систем кодировки, основанных на выделении 1 байта для кодирования одного символа (ANSI, KOI-8 и т.д.). В последнее время получила распространение система Unicode, использующая для кодировки одного символа 2 байта. Это резко расширило возможности, позволив кодировать более чем 65000 различных символов уникальными кодами.

Кодировка изображений

Под изображением будем понимать прямоугольную область, закрашенную непрерывно изменяющимся цветом. Поэтому для представления изображений в целых числах необходимо отдельно дискретизировать прямоугольную область и цвет.

Для описания области она разбивается на множество точечных элементов – пикселей [pixel]. Само множество называется растром [bit map, dot matrix, raster] (см. рис. 1.3), а изображения, которые формируются на основе растра, называются растровыми.

Рис. 1.4 Дискретизация области изображения

Число пикселей называется разрешением [resolution]. Часто встречаются значения 640x480, 800x600, 1024x768, 1280x1024. Каждый пиксел нумеруется, начиная с нуля слева направо и сверху вниз.

Для представления цвета используются цветовые модели.

Цветовая модель [color model] это правило, по которому может быть вычислен цвет. Самая простая цветовая модель – битовая. В ней для описания цвета каждого пиксела (чёрного или белого) используется всего один бит. Для представления полноцветных изображений используются несколько более сложных моделей.

Известно, что любой цвет может быть представлен как сумма трёх основных цветов: красного, зелёного и синего. Если интенсивность каждого цвета представить числом, то любой цвет будет выражаться через набор из трёх чисел. Так определяется наиболее известная цветовая RGB-модель. На каждое число отводится один байт. Так можно представить 224 цвета, то есть примерно 16,7 млн. цветов. Белый цвет в этой модели представляется как (1,1,1), чёрный – (0,0,0), красный (1,0,0), синий (0,0,1). Жёлтый цвет является комбинацией красного и зелёного и потому представляется как (1,1,0).

Цветовая модель RGB [Red-Green-Blue] была стандартизирована в 1931 г. и впервые использована в цветном телевидении. Модель RGB является аддитивной моделью, то есть цвет получается в результате сложения базовых цветов. Существуют и другие цветовые модели, которые для ряда задач оказываются более предпочтительными, чем RGB-модель.

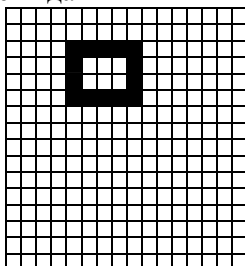
Например, для представления цвета в принтере используется субтрактивная CMY-модель [Cyan-Magenta-Yellow], цвет в которой получается в результате вычитания базовых цветов из белого цвета. Белому цвету в этой модели соответствует (0,0,0), чёрному - (1,1,1), голубому - (1,0,0), сиреневому - (0,1,0), жёлтому - (0,0,1).

В цветовой модели HSV [Hue-Saturation-Value] цвет представляется через цвет, насыщенность и значение, а в модели HLS [Hue-Lightness-Saturation] через оттенок, яркость и насыщенность. Современные графические редакторы, как правило, могут работать с несколькими цветовыми моделями.

Наиболее популярными являются графические форматы: BMP [Bit Map], PCX, GIF [Graphics Interchange Format], TIFF [Tagged Image File Format], JPEG [Joint Photographic Experts Group], которые по существу различаются между собой используемыми методами сжатия. Форматы BMP и PCX используют RLE-алгоритм, форматы GIF и TIFF – LZW-алгоритм, JPEG использует одноимённый алгоритм сжатия.

Пример

Пусть имеется изображение вида



Будем считать, что белый цвет представляется нулём, а чёрный – единицей, тогда в однобитовой модели такое изображение представится в виде:


```

0000000000000000
0000000000000000
0000111110000000
0000100010000000
0000100010000000
0000111110000000
0000000000000000
0000000000000000

```

В шестнадцатеричном виде этот двоичный набор будет выглядеть так:

```
00 00 00 00 0F 80 08 80 08 80 0F 80 00 00 00 00
```

Всего для хранения такого изображения потребуется 16 байт.

Данное изображение легко преобразовать в RGB-модель. Достаточно заменить все нули тройками (1,1,1), а все единицы - тройками (0,0,0). Тогда получим следующее шестнадцатеричное представление изображения:

```

FF FF FF FF FF FF FF FF FF FF FF FF FF F0 00 1F FF FF
FF F1 FF 1F FF FF FF F1 FF 1F FF FF FF F0 00 1F FF FF
FF FF FF FF FF FF FF FF FF FF FF FF

```

Для хранения такого изображения потребуется 48 байт.

Цифровое представление звука

Звук можно описать в виде совокупности синусоидальных волн определённых частоты и амплитуды. Частота волны определяет высоту звукового тона, амплитуда – громкость звука. Частота измеряется в герцах (Гц [Hz]). Диапазон слышимости для человека составляет от 20 Гц до 17000 Гц (или 17 кГц). Рис. 1.5 Дискретизация звукового сигнала

Задача цифрового представления звука, таким образом, сводится к задаче описания синусоидальной кривой. Принцип такого представления изображён на рис. 1.5.

Каждой дискретной выборке присваивается целое число – значение амплитуды. Количество выборок в секунду называется частотой выборки [sampling rate]. Количество возможных значений амплитуды называется точностью выборки [sampling size]. Таким образом, звуковая волна представляется в виде ступенчатой кривой. Ширина ступеньки тем меньше, чем больше частота выборки, а высота ступеньки тем меньше, чем больше точность выборки.

Пример

Возможности наиболее распространённой современной аппаратуры предусматривают работу с частотой выборки до 44,1 кГц, что позволяет правильно описывать звук частотой до 22,05 кГц. Точность выборки имеет всего два значения 8 бит и 16 бит. То есть для представления амплитуды 8-битного звука используется $2^8 = 256$ уровней амплитуды.

Контрольные вопросы

1. Что изучает наука информатика?
2. Что понимают под интерфейсом пользователя?
3. Что такое информация? Какими свойствами она владеет?
4. Из каких операций складывается процесс обработки данных?
5. Что такое информационная система?
6. Из каких этапов складывается работа информационных систем?
7. Какой смысл вкладывается в понятия "информационная технология"?

Список рекомендованной литературы

1. Информатика. Компьютерная техника. Компьютерные технологии. / Пособие под ред. О.И.Пушкаря.— Издательский центр "Академия", Киев, — 2001 г.
2. Коцюбинский А.О., Грошев С.В. Современный самоучитель профессиональной работы на компьютере. — Г.: Триумф, 1999 г.
3. Информатика. Базовый курс. / Под ред. С.В.Симоновича. — Спб., 2000 г.
4. <http://lc.kubagro.ru/aidos/aidos99/4.htm>
5. <http://victoria.lviv.ua/html/informatika/lecture9.htm>

"Сжатие данных"

Характерной особенностью большинства типов данных является их избыточность. Степень избыточности данных зависит от типа данных. Например, для видеоданных степень избыточности в несколько раз больше чем для графических данных, а степень избыточности графических данных, в свою очередь, больше чем степень избыточности текстовых данных. Другим фактором, влияющим на степень избыточности является принятая система кодирования. Примером систем кодирования могут быть обычные языки общения, которые являются ни чем другим, как системами кодирования понятий и идей для высказывания мыслей. Так, установлено, что кодирование текстовых данных с помощью средств русского языка дает в среднем избыточность на 20—25% большую чем кодирование аналогичных данных средствами английского языка.

Для человека избыточность данных часто связана с качеством информации, поскольку избыточность, как правило, улучшает понятность и восприятие информации. Однако, когда речь идет о хранении и передаче информации средствами компьютерной техники, то избыточность играет отрицательную роль, поскольку она приводит к возрастанию стоимости хранения и передачи информации. Особенно актуальной эта проблема стает в случае обработки огромных объемов информации при незначительных объемах носителей данных. В связи с этим, постоянно возникает проблема уменьшения избыточности или сжатия данных. Если методы сжатия данных применяются к готовым файлам, то часто вместо термина "сжатие данных" употребляют термин "архивация данных", сжатый вариант данных называют **архивом**, а программные средства, которые реализуют методы сжатия называются **архиваторами**.

В зависимости от того, в каком объекте размещены данные, подлежащие сжатию различают:

1. Сжатие (архивация) файлов: используется для уменьшения размеров файлов при подготовке их к передаче каналами связи или к транспортированию на внешних носителях малой емкости;
2. Сжатие (архивация) папок: используется как средство уменьшения объема папок перед долгим хранением, например, при резервном копировании;
3. Сжатие (уплотнение) дисков: используется для повышения эффективности использования дискового пространства путем сжатия данных при записи их на носителе информации (как правило, средствами операционной системы).

Существует много практических алгоритмов сжатия данных, но все они базируются на трех теоретических способах уменьшения избыточности данных. Первый способ состоит в изменении содержимого данных, второй — в изменении структуры данных, а третий — в одновременном изменении как структуры, так и содержимого данных.

Если при сжатии данных происходит изменение их содержимого, то метод сжатия называется **необратимым**, то есть при восстановлении (разархивировании) данных из архива не происходит полное восстановление информации. Такие методы часто называются методами сжатия с регулирурованными потерями информации. Понятно, что эти методы можно применять только для таких типов данных, для которых потеря части содержимого не приводит к существенному искажению информации. К таким типам данных относятся видео— и аудиоданные, а также графические данные. Методы сжатия с регулирующими потерями информации обеспечивают значительно большую степень сжатия, но их нельзя применять к текстовым данным. Примерами форматов сжатия с потерями информации могут быть:

- JPEG — для графических данных;
- MPG — для видеоданных;
- MP3 — для аудиоданных.

Если при сжатии данных происходит только изменение структуры данных, то метод сжатия называется **обратимым**. В этом случае, из архива можно восстановить информацию полностью. Обратимые методы сжатия можно применять к любым типам данных, но они дают меньшую степень сжатия по сравнению с необратимыми методами сжатия. Примеры форматов сжатия без потери информации:

- GIF, TIFF — для графических данных;
- AVI — для видеоданных;
- ZIP, ARJ, RAR, CAB, LH — для произвольных типов данных.

Существует много разных практических методов сжатия без потери информации, которые, как правило, имеют разную эффективность для разных типов данных и разных объемов. Однако, в основе этих методов лежат три теоретических алгоритма:

- алгоритм RLE (Run Length Encoding);
- алгоритмы группы KWE(KeyWord Encoding);
- алгоритм Хаффмана.

Алгоритм RLE

В основе алгоритма RLE лежит идея выявления повторяющихся последовательностей данных и замены их более простой структурой, в которой указывается код данных и коэффициент повторения. Например, пусть задана такая последовательность данных, что подлежит сжатию:

1 1 1 1 2 2 3 4 4 4

В алгоритме RLE предлагается заменить ее следующей структурой: 1 4 2 2 3 1 4 3, где первое число каждой пары чисел — это код данных, а второе — коэффициент повторения. Если для хранения каждого элемента данных входной последовательности отводится 1 байт, то вся последовательность будет занимать 10 байт памяти, тогда как выходная последовательность (сжатый вариант) будет занимать 8 байт памяти. Коэффициент сжатия, характеризующий степень сжатия, можно вычислить по формуле:



где V_x — объем памяти, необходимый для хранения выходной (результатирующей) последовательности данных, V_n — входной последовательности данных.

Чем меньше значение коэффициента сжатия, тем эффективней метод сжатия. Понятно, что алгоритм RLE будет давать лучший эффект сжатия при большей длине повторяющейся последовательности данных. В случае рассмотренного выше примера, если входная последовательность будет иметь такой вид: 1 1 1 1 1 3 4 4 4, то коэффициент сжатия будет равен 60%. В связи с этим большая эффективность алгоритма RLE достигается при сжатии графических данных (в особенности для однотонных изображений).

Алгоритмы группы KWE

В основе алгоритма сжатия по ключевым словам положен принцип кодирования лексических единиц группами байт фиксированной длины. Примером лексической единицы может быть обычное слово. На практике, на роль лексических единиц выбираются повторяющиеся последовательности символов, которые кодируются цепочкой символов (кодом) меньшей длины. Результат кодирования помещается в таблице, образуя так называемый словарь.

Существует довольно много реализаций этого алгоритма, среди которых наиболее распространенными являются алгоритм Лемпеля—Зива (алгоритм LZ) и его модификация алгоритм Лемпеля—Зива—Велча (алгоритм LZW). Словарем в данном алгоритме является потенциально бесконечный список фраз. Алгоритм начинает работу с почти пустым словарем, который содержит только одну закодированную строку, так называемая NULL—строка. При считывании очередного символа входной последовательности данных, он прибавляется к текущей строке. Процесс продолжается до тех пор, пока текущая строка соответствует какой—нибудь фразе из словаря. Но рано или поздно текущая строка перестает соответствовать какой—нибудь фразе словаря. В момент, когда текущая строка представляет собой последнее совпадение со словарем плюс только что прочитанный символ сообщения, кодер выдает код, который состоит из индекса совпадения и следующего за ним символа, который нарушил совпадение строк. Новая фраза, состоящая из индекса совпадения и следующего за ним символа, прибавляется в словарь. В следующий раз, если эта фраза появится в сообщении, она может быть использована для построения более длинной фразы, что повышает меру сжатия информации.

Алгоритм LZW построен вокруг таблицы фраз (словаря), которая заменяет строки символов сжимаемого сообщения в коды фиксированной длины. Таблица имеет так называемое свойством опережения, то есть для каждой фразы словаря, состоящей из некоторой фразы w и символа K , фраза w тоже заносится в словарь. Если все части словаря полностью заполнены, кодирование перестает быть адаптивным (кодирование происходит исходя из уже существующих в словаре фраз).

Алгоритмы сжатия этой группы наиболее эффективны для текстовых данных больших объемов и малоэффективны для файлов маленьких размеров (за счет необходимости сохранения словаря).

Алгоритм Хаффмана

В основе алгоритма Хаффмана лежит идея кодирования битовыми группами. Сначала проводится частотный анализ входной последовательности данных, то есть устанавливается частота вхождения каждого символа, встречающегося в ней. После этого, символы сортируются по уменьшению частоты вхождения.

Основная идея состоит в следующем: чем чаще встречается символ, тем меньшим количеством бит он кодируется. Результат кодирования заносится в словарь, необходимый для декодирования. Рассмотрим простой пример, иллюстрирующий работу алгоритма Хаффмана.

Пусть задан текст, в котором буква 'A' входит 10 раз, буква 'B' — 8 раз, 'C' — 6 раз, 'D' — 5 раз, 'E' и 'F' — по 4 раза. Тогда один из возможных вариантов кодирования по алгоритму Хаффмана приведен в таблице 1.

Таблица 1.

Символ	Частота вхождения	Битовый код
--------	-------------------	-------------

A	10	00
B	8	01
C	6	100
D	5	101
E	4	110
F	4	111

Как видно из таблицы 1, размер входного текста до сжатия равен 37 байт, тогда как после сжатия — 93 бит, то есть около 12 байт (без учета длины словаря). Коэффициент сжатия равен 32%. Алгоритм Хаффмана универсальный, его можно применять для сжатия данных любых типов, но он малоэффективен для файлов маленьких размеров (за счет необходимости сохранения словаря).

На практике программные средства сжатия данных синтезируют эти три "чистых" алгоритмы, поскольку их эффективность зависит от типа и объема данных. В таблице 2 приведены распространенные форматы сжатия и соответствующие им программы—архиваторы, использующиеся на практике.

Таблица 2.

Формат сжатия	Операционная система MS DOS		Операционная система Windows	
	Программа архивации	Программа разархивации	Программа архивации	Программа разархивации
ARJ	Arj.exe	Arj.exe	WinArj.exe	WinArj.exe
RAR	Rar.exe	Unrar.exe	WinRar.exe	WinRar.exe
ZIP	Pkzip.exe	Pkunzip.exe	WinZip.exe	WinZip.exe

Кроме того, современные архиваторы предоставляют пользователю полный спектр услуг для работы с архивами, основными из которых являются:

1. создание нового архива;
2. добавление файлов в существующий архив;
3. распаковывание файлов из архива;
4. создание самораспаковывающихся архивов (self—extractor archive);
5. создание распределенных архивов фиксированного размера для носителей маленькой емкости;
6. защита архивов паролями от несанкционированного доступа;
7. просмотр содержимого файлов разных форматов без предварительного распаковывания;
8. поиск файлов и данных внутри архива;
9. проверка на вирусы в архиве к распаковыванию;
10. выбор и настройка коэффициента сжатия.

Контрольные вопросы

1. Какие факторы влияют на степень избыточности данных?
2. Что такое архив? Какие программные средства называются архиваторами?
3. Почему методы сжатия, при которых происходит изменение содержимого данных, называются необратимыми?
4. Приведите примеры форматов сжатия с потерями информации.
5. В чем состоит преимущество обратимых методов сжатия над необратимыми? А недостаток?
6. Которая существует зависимость между коэффициентом сжатия и эффективностью метода сжатия?
7. В чем состоит основная идея алгоритма RLE?
8. В чем состоит основная идея алгоритмов группы KWE?
9. В чем состоит основная идея алгоритма Хаффмана?
10. Какие вы знаете программы—архиваторы? Коротко охарактеризуйте их.

Список рекомендованной литературы

1. Информатика. Базовый курс. / Под ред. С.В.Симоновича. — СПб., 2000 г.
2. А.П.Микляев, Настольная книга пользователя IBM PC 3—издание М.:, "Солон—Р", 2000, 720 с.
3. Симонович С.В., Евсеев Г.А., Мураховский В.И. Вы купили компьютер: Полное руководство для начинающих в вопросах и ответах. — М.: АСТ—ПРЕСС КНИГА; Инфорком—Пресс, 2001.— 544 с.: ил. (1000 советов).
4. Ковтанюк Ю.С., Соловьян С.В. Самоучитель работы на персональном компьютере — К.:Юниор, 2001.— 560с., ил.

" Программное обеспечение "

В основу работы компьютеров положен программный принцип управления, состоящий в том, что компьютер выполняет действия по заранее заданной программе. Этот принцип обеспечивает универсальность использования компьютера: в определенный момент времени решается задача соответственно выбранной программе. После ее завершения в память загружается другая программа и т.д.

Программа — это запись алгоритма решения задачи в виде последовательности команд или операторов языком, который понимает компьютер. Конечной целью любой компьютерной программы является управление аппаратными средствами.

Для нормального решения задач на компьютере нужно, чтобы программа была отлажена, не требовала доработок и имела соответствующую документацию. Поэтому, относительно работы на компьютере часто используют термин программное обеспечение (software), под которым понимают совокупность программ, процедур и правил, а также документации, касающихся функционирования системы обработки данных.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и взаимодействии. Состав программного обеспечения вычислительной системы называется программной конфигурацией. Между программами существует взаимосвязь, то есть работа множества программ базируется на программах низшего уровня.

Междупрограммный интерфейс — это распределение программного обеспечения на несколько связанных между собою уровней. Уровни программного обеспечения представляют собой пирамиду, где каждый высший уровень базируется на программном обеспечении предшествующих уровней. Схематично структура программного обеспечения приведена на рис. 1.

Прикладной уровень
Служебный уровень
Системный уровень
Базовый уровень

Базовый уровень

Базовый уровень является низшим уровнем программного обеспечения. Отвечает за взаимодействие с базовыми аппаратными средствами. Базовое программное обеспечение содержится в составе базового аппаратного обеспечения и сохраняется в специальных микросхемах постоянного запоминающего устройства (ПЗУ), образуя базовую систему ввода—вывода BIOS. Программы и данные записываются в ПЗУ на этапе производства и не могут быть изменены во время эксплуатации.

Системный уровень

Системный уровень — является переходным. Программы этого уровня обеспечивают взаимодействие других программ компьютера с программами базового уровня и непосредственно с аппаратным обеспечением. От программ этого уровня зависят эксплуатационные показатели всей вычислительной системы. При подсоединении к компьютеру нового оборудования, на системном уровне должна быть установлена программа, обеспечивающая для остальных программ взаимосвязь с устройством. Конкретные программы, предназначенные для взаимодействия с конкретными устройствами, называют драйверами.

Другой класс программ системного уровня отвечает за взаимодействие с пользователем. Благодаря ему, можно вводить данные в вычислительную систему, руководить ее работой и получать результат в удобной форме. Это средства обеспечения пользовательского интерфейса, от них зависит удобство и производительность работы с компьютером.

Совокупность программного обеспечения системного уровня образует ядро операционной системы компьютера. Наличие ядра операционной системы — это первое условие для возможности практической работы пользователя с вычислительной системой. Ядро операционной системы выполняет такие функции: управление памятью, процессами ввода—вывода, файловой системой, организация взаимодействия и диспетчеризация процессов, учет использования ресурсов, обработка команд и т.д.

Служебный уровень

Программы этого уровня взаимодействуют как с программами базового уровня, так и с программами системного уровня. Назначение служебных программ (утилит) состоит в автоматизации работ по проверке и настройке компьютерной системы, а также для улучшения функций системных программ. Некоторые служебные программы (программы обслуживания) сразу входят в состав операционной системы, дополняя ее ядро, но большинство являются внешними программами и расширяют функции операционной системы. То есть, в разработке служебных программ отслеживаются два направления: интеграция с операционной системой и автономное функционирование.

Классификация служебных программных средств

1. **Диспетчеры файлов** (файловые менеджеры). С их помощью выполняется большинство операций по обслуживанию файловой структуры: копирование, перемещение, переименование файлов, создание каталогов (папок), уничтожение объектов, поиск файлов и навигация в файловой структуре. Базовые программные средства содержатся в составе программ системного уровня и устанавливаются вместе с операционной системой.

2. **Средства сжатия данных** (архиваторы). Предназначены для создания архивов. Архивные файлы имеют повышенную плотность записи информации и соответственно, эффективнее используют носители информации.

3. **Средства диагностики**. Предназначены для автоматизации процессов диагностики программного и аппаратного обеспечения. Их используют для исправления ошибок и для оптимизации работы компьютерной системы.

4. **Программы инсталляции** (установки). Предназначены для контроля за добавлением в текущую программную конфигурацию нового программного обеспечения. Они следят за состоянием и изменением окружающей программной среды, отслеживают и протоколируют образование новых связей, утраченных во время уничтожения определенных программ. Простые средства управления установлением и уничтожением программ содержатся в составе операционной системы, но могут использоваться и дополнительные служебные программы.

5. **Средства коммуникации**. Разрешают устанавливать соединение с удаленными компьютерами, передают сообщения электронной почты, пересылают факсимильные сообщения и т.п..

6. **Средства просмотра и воспроизведения**. Преимущественно, для работы с файлами, их необходимо загрузить в "родную" прикладную программу и внести необходимые исправления. Но, если редактирование не нужно, существуют универсальные средства для просмотра (в случае текста) или воспроизведения (в случае звука или видео) данных.

7. **Средства компьютерной безопасности**. К ним относятся средства пассивной и активной защиты данных от повреждения, несанкционированного доступа, просмотра и изменения данных. Средства пассивной защиты — это служебные программы, предназначенные для резервного копирования. Средства активной защиты применяют антивирусное программное обеспечение. Для защиты данных от несанкционированного доступа, их просмотра и изменения используют специальные системы, базирующиеся на криптографии.

Прикладной уровень

Программное обеспечение этого уровня представляет собой комплекс прикладных программ, с помощью которых выполняются конкретные задачи (производственных, творческих, развлекательных и учебных). Между прикладным и системным программным обеспечением существует тесная взаимосвязь. Универсальность вычислительной системы, доступность прикладных программ и широта функциональных возможностей компьютера непосредственно зависят от типа имеющейся операционной системы, системных средств, помещенных в ее ядро и взаимодействии комплекса человек—программа—оборудование.

Классификация прикладного программного обеспечения

1. **Текстовые редакторы**. Основные функции — это ввод и редактирование текстовых данных. Для операций ввода, вывода и хранения данных текстовые редакторы используют системное программное обеспечение. С этого класса прикладных программ начинают знакомство с программным обеспечением и на нем приобретают первые привычки работы с компьютером.

2. **Текстовые процессоры**. Разрешают форматировать, то есть оформлять текст. Основными средствами текстовых процессоров являются средства обеспечения взаимодействия текста, графики, таблиц и других объектов, составляющих готовый документ, а также средства автоматизации процессов редактирования и форматирования. Современный стиль работы с документами имеет два подхода: работа с бумажными документами и работа с электронными документами. Приемы и методы форматирования таких документов различаются между собой, но текстовые процессоры способны эффективно обрабатывать оба вида документов.

3. **Графические редакторы**. Широкий класс программ, предназначенных для создания и обработки графических изображений. Различают три категории:

- растровые редакторы;
- векторные редакторы;
- 3—D редакторы (трехмерная графика).

В **растровых редакторах** графический объект представлен в виде комбинации точек (растров), которые имеют свою яркость и цвет. Такой подход эффективный, когда графическое изображение имеет много цветов и информация про цвет элементов намного важнее, чем информация про их форму. Это

характерно для фотографических и полиграфических изображений. Применяют для обработки изображений, создания фотоэффектов и художественных композиций.

Векторные редакторы отличаются способом представления данных изображения. Объектом является не точка, а линия. Каждая линия рассматривается, как математическая кривая III порядка и представлена формулой. Такое представление компактнее, чем растровое, данные занимают меньше места, но построение объекта сопровождается пересчетом параметров кривой в координаты экранного изображения, и соответственно, требует более мощных вычислительных систем. Широко применяются в рекламе, оформлении обложек полиграфических изданий.

Редакторы трехмерной графики используют для создания объемных композиций. Имеют две особенности: разрешают руководить свойствами поверхности в зависимости от свойств освещения, а также разрешают создавать объемную анимацию.

4. **Системы управления базами данных (СУБД)**. Базой данных называют большие массивы данных, организованные в табличные структуры. Основные функции СУБД:

- создание пустой структуры базы данных;
- наличие средств ее заполнения или импорта данных из таблиц другой базы;
- возможность доступа к данным, наличие средств поиска и фильтрации.

В связи с распространением сетевых технологий, от современных СУБД требуется возможность работы с удаленными и распределенными ресурсами, которые находятся на серверах Интернета.

5. **Электронные таблицы**. Предоставляют комплексные средства для хранения разных типов данных и их обработки. Основной акцент смещен на преобразование данных, предоставлен широкий спектр методов для работы с числовыми данными. Основная особенность электронных таблиц состоит в автоматическом изменении содержимого всех ячеек при изменении отношений, заданных математическими или логическими формулами.

Широкое применение находят в бухгалтерском учете, анализе финансовых и торговых рынков, средствах обработки результатов экспериментов, то есть в автоматизации регулярно повторяемых вычислений больших объемов числовых данных.

6. **Системы автоматизированного проектирования (САД—системы)**. Предназначены для автоматизации проектно—конструкторских работ. Применяются в машиностроении, приборостроении, архитектуре. Кроме графических работ, разрешают проводить простые расчеты и выбор готовых конструктивных элементов из существующей базы данных.

Особенность САД—систем состоит в автоматическом обеспечении на всех этапах проектирования технических условий, норм и правил. САПР являются необходимым компонентом для гибких производственных систем (ГВС) и автоматизированных систем управления технологическими процессами (АСУ ТП).

7. **Настольные издательские системы**. Автоматизируют процесс верстки полиграфических изданий. Издательские системы отличаются расширенными средствами управления взаимодействия текста с параметрами страницы и графическими объектами, но имеют более слабые возможности по автоматизации ввода и редактирования текста. Их целесообразно применять к документам, которые предварительно обработаны в текстовых процессорах и графических редакторах.

8. **Редакторы HTML (Web—редакторы)**. Особый класс редакторов, объединяющих в себе возможности текстовых и графических редакторов. Предназначены для создания и редактирования Web—страниц Интернета. Программы этого класса можно использовать при подготовке электронных документов и мультимедийных изданий.

9. **Браузеры** (средства просмотра Web—документов). Программные средства предназначены для просмотра электронных документов, созданных в формате HTML. Воспроизводят, кроме текста и графики, музыку, человеческий язык, радиопередачи, видеоконференции и разрешают работать с электронной почтой.

10. **Системы автоматизированного перевода**. Различают электронные словари и программы перевода языка.

Электронные словари — это средства для перевода отдельных слов в документе. Используются профессиональными переводчиками, которые самостоятельно переводят текст.

Программы автоматического перевода используют текст на одном языке и выдают текст на другом, то есть автоматизируют перевод. При автоматизированном переводе невозможно получить качественный исходный текст, поскольку все сводится к переводу отдельных лексических единиц. Но, для технического текста, этот барьер снижен.

Программы автоматического перевода целесообразно использовать:

- при абсолютном незнании иностранного языка;
- при необходимости быстрого ознакомления с документом;

- для перевода на иностранный язык;
- для создания черновика, который потом будет подправлен полноценным переводом.

11. **Интегрированные системы делопроизводства.** Средства для автоматизации рабочего места руководителя. В частности, это функции создания, редактирования и форматирования документов, централизация функций электронной почты, факсимильной и телефонной связи, диспетчеризация и мониторинг документооборота предприятия, координация работы подразделов, оптимизация административно—хозяйственной деятельности и поставка оперативной и справочной информации.

12. **Бухгалтерские системы.** Имеют функции текстовых, табличных редакторов и СУБД. Предназначены для автоматизации подготовки начальных бухгалтерских документов предприятия и их учета, регулярных отчетов по итогам производственной, хозяйственной и финансовой деятельности в форме, приемлемой для налоговых органов, внебюджетных фондов и органов статистического учета.

13. **Финансовые аналитические системы.** Используют в банковских и биржевых структурах. Разрешают контролировать и прогнозировать ситуацию на финансовых, торговых рынках и рынках сырья, выполнять анализ текущих событий, готовить отчеты.

14. **Экспертные системы.** Предназначены для анализа данных, содержащихся в базах знаний и выдачи результатов, при запросе пользователя. Такие системы используются, когда для принятия решения нужны широкие специальные знания. Используются в медицине, фармакологии, химии, юриспруденции. С использованием экспертных систем связана область науки, которая носит название инженерии знаний.

Инженеры знаний — это специалисты, являющиеся промежуточным звеном между разработчиками экспертных систем (программистами) и ведущими специалистами в конкретных областях науки и техники (экспертами).

15. **Геоинформационные системы (ГИС).** Предназначены для автоматизации картографических и геодезических работ на основе информации, полученной топографическим или аэрографическими методами.

16. **Системы видеомонтажа.** Предназначены для цифровой обработки видеоматериалов, монтажа, создания видеозффектов, исправления дефектов, добавления звука, титров и субтитров. Отдельные категории представляют учебные, справочные и развлекательные системы и программы. Характерной особенностью являются повышенные требования к мультимедийной составляющей.

17. **Инструментальные языки и системы программирования.** Эти средства служат для разработки новых программ. Компьютер "понимает" и может выполнять программы в машинном коде. Каждая команда при этом имеет вид последовательности нулей и единиц. Писать программы на машинном языке крайне неудобно. Поэтому программы разрабатываются на языке, понятном человеку (инструментальный язык или алгоритмический язык программирования), после чего, специальной программой, которая называется транслятором, текст программы переводится (транслируется) на машинный код.

Трансляторы бывают двух типов:

- интерпретаторы,
- компиляторы.

Интерпретатор читает один оператор программы, анализирует его и сразу выполняет, после чего переходит к обработке следующего оператора.

Компилятор сначала читает, анализирует и переводит на машинный код всю программу и только после завершения всей трансляции эта программа выполняется.

Инструментальные языки делятся на языки низкого уровня (близкие к машинному языку) и языки высокого уровня (близкие к человеческим языкам). К языкам низкого уровня принадлежат ассемблеры, а высокого — Pascal, Basic, C/C++, языки баз данных и т.д. В систему программирования, кроме транслятора, входит текстовый редактор, компоновщик, библиотека стандартных программ, отладчик, визуальные средства автоматизации программирования. Примерами таких систем являются Delphi, Visual Basic, Visual C++, Visual FoxPro и др.

Контрольные вопросы

1. Что принято понимать под термином 'software'?
2. На какие уровни делится программное обеспечение?
3. Совокупность программ какого уровня образуют ядро операционной системы?
4. Какие функции выполняет ядро операционной системы?
5. Для чего предназначены программы базового уровня?
6. Какие классы программ служебного уровня вы знаете?
7. С помощью программ какого класса можно осуществлять ввод, редактирование и оформление текстовых данных?
8. Какие вы знаете категории графических редакторов?
9. В каких случаях целесообразно использовать системы автоматизированного перевода?

5. Что такое интерпретатор и компилятор? Какая между ними разница?