Министерство науки и образования Украины

Славянский государственный педагогический университет

## Реферат

на тему:

Информация, информатика,

**представление информации.**

Студента 3 курса

Шрам Сергея

##### Славянск

2003

1. Информация, информатика,

представление информации.

Понятие об информации

Несмотря на то, что человеку постоянно приходится иметь дело с информацией (он получает ее с помощью органов чувств), строгого научного определения, что же такое *информация,* не существует. В тех случаях, когда наука не может дать четкого определения какому-то предмету или явлению, люди пользуются *понятиями.*

Понятия отличаются от определений тем, что разные люди при разных обстоятельствах могут вкладывать в них разный смысл. В бытовом смысле под информацией обычно понимают те сведения, которые человек получает от окружающей природы и общества с помощью органов чувств. Наблюдая за природой, общаясь с другими людьми, читая книги и газеты, просматривая телевизионные передачи, мы получаем информацию.

Математик рассмотрит это понятие шире и включит в него те сведения, которые человек не получал, а создал сам с помощью умозаключений. Биолог же пойдет еще дальше и отнесет к информации и те данные, которые человек не получал с помощью органов чувств и не создавал в своем уме, а хранит в себе с момента рождения и до смерти. Это генетический код, благодаря которому дети так похожи на родителей.

Итак, в разных научных дисциплинах и в разных областях техники существуют разные понятия об информации. Нам же, приступая к изучению информатики, надо найти что-то общее, что объединяет различные подходы. И такая общая черта есть. Все отрасли науки и техники, имеющие дело с информацией, сходятся в том, что информация обладает четырьмя свойствами. Информацию можно: *создавать, передавать* (и, соответственно, принимать), *хранить* и *обрабатывать.*

Каждая дисциплина решает эти вопросы по-разному. В нашем учебном пособии мы рассмотрим те средства, которые для этого предоставляет *информатика.*

Информатика

*Информатика — это техническая наука, систематизирующая приемы создания, хранения, обработки и передачи информации средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.*

Из этого определения видно, что информатика очень близка к *технологии,* поскольку отвечает на вопрос *как..?*

• Как принимать и хранить информацию?

• Как обрабатывать информацию и преобразовывать ее в форму, удобную для человека?

• Как использовать вычислительную технику с наибольшей эффективностью?

• Как использовать достижения других наук для создания новых средств вычислительной техники?

• Как управлять техническими средствами с помощью программ?

Поэтому не случайно предмет, изучением которого занимается информатика, нередко называют *информационной технологией* или *компьютерной технологией.*

Важным в нашем определении является то, что у информатики есть как бы две стороны. С одной стороны, она занимается изучением устройств и принципов действия средств вычислительной техники, а с другой стороны — систематизацией приемов и методов работы с программами, управляющими этой техникой.

Краткая история информатики

Корни информатики лежат в другой науке — *кибернетике.* Понятие «кибернетика» впервые появилось в первой половине XIX века, когда французский физик Андре Мари Ампер, известный нам из школьного курса физики по закону Ампера, решил создать единую классификацию всех наук, как существовавших в то время, так и гипотетических (которые не существовали, но, по его мнению, должны были бы существовать). Он предположил, что должна существовать некая наука, занимающаяся изучением искусства управления. Ампер не имел в виду управление техническими системами, поскольку сложных технических систем в те времена еще не было. Он имел в виду искусство управления людьми, то есть обществом. Эту несуществующую науку Ампер назвал кибернетикой от греческого слова *кибернетикос* (искусный в управлении). В Древней Греции этого титула удостаивались лучшие мастера управления боевыми колесницами.

**Впоследствии слово *кибернетикос* было заимствовано римлянами — так в латинском языке появилось слово *губернатор* (управляющий провинцией). Сегодня уже трудно догадаться, что слова «кибернетика» и «губернатор» имеют одно происхождение, но это так.**

С тех пор о кибернетике забыли более, чем на сто лет. В 1948 году выдающийся американский математик Норберт Винер, труды которого по математической логике легли в основу зарождавшегося тогда программирования вычислительной техники, вновь возродил термин «кибернетика» и определил ее как науку об управлении в живой природе и в технических системах. Это определение оказалось весьма спорным. Смешивание живой природы и технических систем в одной дисциплине привело к резкому неприятию такого определения учеными многих стран. Особенно сильной критике зарождавшаяся кибернетика подверглась в Советском Союзе. В большинстве стран мира научная дискуссия привела к расколу в научных кругах, а в СССР, где кибернетика получила даже политическое осуждение, работы в этой области были вообще прекращены на много лет, что болезненно сказывается и по сей день.

Так из-за спорного определения в молодой зарождающейся науке произошел раскол. Сегодня *кибернетика* продолжает изучать связь между психологией и математической логикой, разрабатывает методы создания искусственного интеллекта, но наряду с ней уже действует другая, отделившаяся от нее наука. Она занимается проблемами применения средств вычислительной техники для работы с информацией. В Великобритании и США эту науку называют *computer science (наука о вычислительной технике).* Во Франции она получила другое название — *informatique (информатика).* Оттуда это название и пришло к нам в Украину и Россию, а также в некоторые другие страны Восточной Европы.

Информация аналоговая и цифровая

Информацию можно классифицировать разными способами, и разные науки делают это по-разному. Например, в философии различают информацию *объективную* и *субъективную.* Объективная информация отражает явления природы и человеческого общества. Субъективная информация создается людьми и отражает их взгляд на объективные явления.

Для криминалистики, например, очень важно, что информация бывает *полной* и *неполной, истинной* и *ложной, достоверной* и *недостоверной.* Юристы рассматривают информацию как *факты.* Физики же рассматривают информацию как *сигналы —* для них наиболее важна передача информации, поскольку физика изучает законы природы, лежащие в основе распространения сигналов разных видов (оптических, звуковых, электромагнитных и других). Биология изучает методы обмена информацией между животными, генетика изучает передачу информации по наследству с помощью генов, а лингвистика изучает методы кодирования и выражения информации языковыми методами.

Каждая наука, занимающаяся вопросами, связанными с информацией, вводит свою систему классификации. Для информатики самым главным вопросом является то, каким образом используются средства вычислительной техники для создания, хранения, обработки и передачи информации, поэтому у информатики особый подход к классификации информации. В информатике отдельно рассматривают *аналоговую* информацию и *цифровую.* Это важно, поскольку человек благодаря своим органам чувств, привык иметь дело с аналоговой информацией, а вычислительная техника, наоборот, в основном работает с цифровой информацией.

Человек так устроен, что воспринимает информацию с помощью органов чувств. Свет, звук и тепло — это энергетические сигналы, а вкус и запах – это результат воздействия химических соединений, в основе которого тоже энергетическая природа. Человек испытывает энергетические воздействия непрерывно и может никогда не встретиться с одной и той же их комбинацией дважды. Мы не найдем двух одинаковых зеленых листьев на одном дереве и не услышим двух абсолютно одинаковых звуков — это информация *аналоговая.* Если же разным цветам дать номера, а разным звукам — ноты, то аналоговую информацию можно превратить в *цифровую.*

Музыка, когда мы ее слышим, несет аналоговую информацию, но стоит только записать ее нотами, как она становится цифровой. Мы легко различим разницу в одной и той же ноте, если исполнить ее на фортепиано и на флейте, хотя на бумаге эти ноты выглядят одинаково.

Разница между аналоговой информацией и цифровой прежде всего в том, что аналоговая информация *непрерывна,* а цифровая — *дискретна.* Если у художника в палитре только одна зеленая краска, то непрерывную бесконечность зеленых цветов листьев он передаст очень грубо, и все деревья на картине будут иметь одинаковый цвет. Если у художника три разные зеленые краски, то передача цвета уже будет чуть более точной. Для большей точности передачи аналоговой информации о живой природе художники смешивают разные краски и получают большое количество оттенков.

Аналого-цифровое преобразование

*Преобразование информации из аналоговой формы в цифровую называют* ***аналогово-цифровым преобразованием (АЦП).***

Примеры аналоговой информации известны нам из школьного курса математики. Графики непрерывных функций выражают аналоговую информацию.

1. На рисунке показан график функции **Y=X2.** Это график непрерывной функции.

2. Тот же самый график после преобразования в цифровую форму выглядит иначе — намного грубее.

3. Погрешность, которая возникает при таком преобразовании, называется *погрешностью оцифровки.*

4. Преобразование можно сделать менее грубым, если столбики диаграммы поставить почаще (так уменьшается *дискретность).*

5. Чем меньше дискретность, тем ближе цифровая информация к аналоговой и меньше погрешность оцифровки.

Вы видите, что при уменьшении дискретности на диаграмме становится больше столбиков. Если дискретность сделать очень маленькой, то точность представления непрерывной аналоговой информации в виде последовательности чисел можно сделать очень высокой, но и столбиков в диаграмме будет больше. Поэтому чем ближе цифровая информация приближается по качеству к аналоговой, тем больше вычислений приходится выполнять компьютеру, а значит, тем больше информации ему надо хранить и обрабатывать.

**Чем мощнее компьютер, тем больше информации он может обработать в единицу времени. Чем быстрее компьютер обрабатывает информацию, тем выше качество изображения, лучше звук и точнее результаты расчетов, но тем дороже обходится людям прием, передача и обработка информации.**

6. Справа показаны два «одинаковых» рисунка, полученные из Интернета. Затраты времени (и средств) на прием второго рисунка в десять раз выше, потому что он содержит больше информации.

Устройства аналоговые и цифровые

Органы чувств человека так устроены, что он способен принимать, хранить и обрабатывать аналоговую информацию. Многие устройства, созданные человеком, тоже работают с аналоговой информацией.

1. Телевизор — это аналоговое устройство. Внутри телевизора есть кинескоп. Луч кинескопа непрерывно перемещается по экрану. Чем сильнее луч, тем ярче светится точка, в которую он попадает. Изменение свечения точек происходит плавно и непрерывно.

2. Монитор компьютера тоже похож на телевизор, но это устройство цифровое. В нем яркость луча изменяется не плавно, а скачком (дискретно). Луч либо есть, либо его нет. Если он есть, мы видим яркую точку (белую или цветную). Если луча нет, мы видим черную точку. Поэтому изображения на экране монитора получаются более четкими, чем на экране телевизора.

3. Проигрыватель грампластинок — аналоговое устройство. Чем больше высота неровностей на звуковой дорожке, тем громче звучит звук.

4. Телефон — тоже аналоговое устройство. Чем громче мы говорим в трубку, тем выше сила тока, проходящего по проводам, тем громче звук, который слышит наш собеседник.

К цифровым устройствам относятся персональные компьютеры — они работают с информацией, представленной в цифровой форме. Цифровыми также являются музыкальные проигрыватели лазерных компакт-дисков, поэтому музыкальные компакт-диски можно воспроизводить на компьютере.

Недавно началось создание цифровой телефонной связи, а в ближайшие годы ожидается и появление цифрового телевидения. В некоторых городах Украины и России уже работают цифровые телевизионные станции. После того как телевидение станет цифровым, качество изображения на экране телевизора намного улучшится — оно станет ближе к качеству изображения на экране компьютерного монитора.

Понятие о кодировании информации

Информация передается в виде сигналов. Когда мы разговариваем с другими людьми, то улавливаем звуковые сигналы. Если мы смотрим в окно, наш глаз принимает световые потоки, отраженные от объектов окружающей природы. Световой поток — это тоже сигнал.

А как же информация хранится? Для того чтобы информацию сохранить, ее надо закодировать. Любая информация всегда хранится в виде *кодов.* Когда мы что-то пишем в тетради, мы на самом деле кодируем информацию с помощью специальных *символов.* Эти символы всем знакомы — они называются буквами. И система такого кодирования тоже хорошо известна — это обыкновенная азбука. Жители других стран те же самые слова запишут по-другому (другими буквами) — у них своя азбука. Можно сказать, что у них другая система кодирования. В некоторых странах вместо букв используют иероглифы — это еще более сложный способ кодирования информации.

Можно кодировать и звуки. С одной из таких систем кодирования вы тоже хорошо знакомы: мелодию можно записать с помощью нот. Это не единственная система кодирования музыки. В давние времена на Руси музыку записывали с помощью так называемых «крюков» — это особая форма записи.

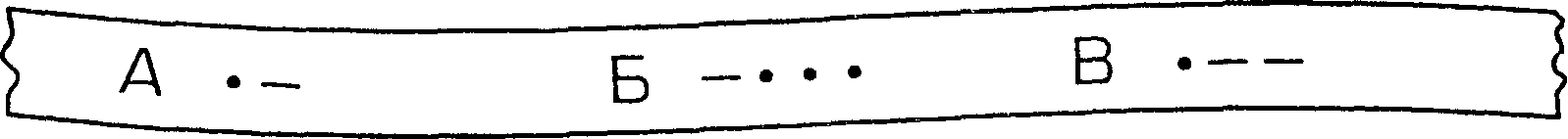
Хранить можно не только текстовую и звуковую информацию. В виде кодов хранятся и изображения. Если посмотреть на рисунок с помощью увеличительного стекла, то видно, что он состоит из точек — это так называемый *растр.* Координаты каждой точки можно запомнить в виде чисел. Цвет каждой точки тоже можно запомнить в виде числа. Эти числа могут храниться в памяти компьютера и передаваться на любые расстояния. По ним компьютерные программы способны изобразить рисунок на экране или напечатать его на принтере. Изображение можно сделать больше или меньше, темнее или светлее, его можно повернуть, наклонить, растянуть. Мы говорим о том, что на компьютере обрабатывается изображение, но на самом деле компьютерные программы изменяют числа, которыми отдельные точки изображения представлены в памяти компьютера.

Хранение цифровой информации. Бит

Вы уже знаете, что компьютеры предпочитают работать с цифровой информацией, а не с аналоговой. Так происходит потому, что цифровую информацию очень удобно кодировать, а значит, ее удобно хранить и обрабатывать.

Компьютер работает с информацией по принципу «разделяй и властвуй». Если это книга, то она делится на главы, разделы, абзацы, предложения, слова и буквы (то есть, символы). Компьютер отдельно работает с каждым символом. Если это рисунок, то компьютер работает с каждой точкой этого рисунка отдельно.

Спрашивается, а до каких же пор можно делить информацию? Буква — это самая маленькая часть информации? Оказывается, нет. Существует много различных букв, и, для того чтобы компьютер мог различать буквы, их тоже надо кодировать. В телеграфной азбуке, например, буквы кодирую г с помощью точек и тире:



Точки и тире — это действительно самая малая часть информации, но в информатике кодом телеграфной азбуки не пользуются. Вместо точек и тире применяют нули и единицы — такой код называется *двоичным.* По-английски двоичный знак звучит как *binary digit* Сокращенно получается *bit* (бит).

**Бит — это наименьшая единица информации, которая выражает логическое значение. *Да* или *Нет* и обозначается двоичным числом 1 или 0.**

Если какая-то информация представлена в цифровом виде, то компьютер легко превращает числа, которыми она закодирована, в последовательности нулей и единиц, а дальше уже работает с ними. Вы тоже можете преобразовать любое число в двоичную форму. Делается это следующим образом.

1. Берем, например, число 29. Поскольку это число нечетное, отнимаем от него единицу, записываем ее отдельно, а число делим пополам. Получилось 14.

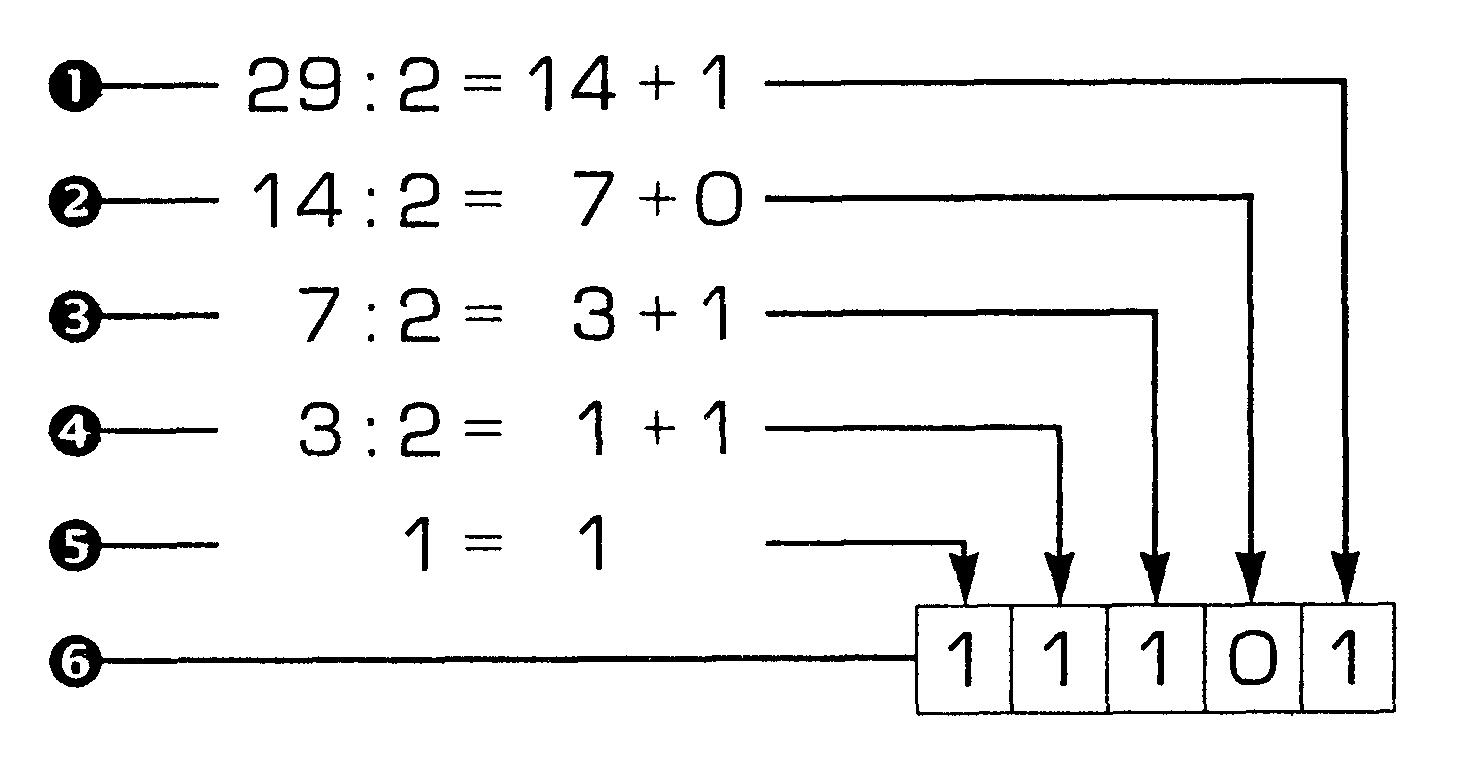
2. Число 14 — четное. Отнимать от него единицу не нужно, поэтому слева от «запомненной» единицы запишем 0. Число делим пополам, получаем 7.

3. Число 7 — опять нечетное. Отнимаем от него единицу, записываем ее отдельно и делим число пополам. Получается 3.

4. Число 3 — нечетное. Отнимаем единицу, записываем ее отдельно, и результат делим пополам — получаем 1.

5. Последнюю единицу уже не делим, а просто записываем слева от полученного результата.

6. Смотрим на результат. У нас получилось двоичное число 11101 — это и есть двоичный код числа 29.



Как видите, преобразовать число в двоичный код совсем не трудно. Отнять единичку и поделить число пополам может каждый, хотя приятной эту работу не назовешь. Для человека эта работа очень утомительна. Зато компьютер, который выполняет сотни миллионов операций в секунду, преобразовывает числа в двоичный код так быстро, что это совершенно не заметно со стороны.

**Когда какая-то операция выполняется незаметно, говорят, что она выполняется *прозрачно.* Мы думаем, что компьютер работает с текстами, графикой, музыкой и видео, а на самом деле он работает с нулями и единицами, но для нас эта работа *прозрачна,* мы ее не замечаем и можем о ней не думать.**

**От битов к байтам**

Бит — очень удобная единица для хранения информации в компьютере, но не очень удобная для обработки информации. Если мы покупаем в магазине хлеб, то продавец может выдать нам каждый батон отдельно, упаковав его в полиэтиленовый пакет. Но если мы покупаем орехи, разве он станет упаковывать отдельно каждый орех?

Бит — очень маленькая единица информации. Работать с каждым битом отдельно, конечно, можно, но это малопроизводительно. Обработкой информации в компьютере занимается специальная микросхема, которая называется *процессор.* Эта микросхема устроена так, что может обрабатывать группу битов одновременно (параллельно). В начале 70-х годов, еще до появления персональных компьютеров, были карманные электронные калькуляторы, в которых процессор мог одновременно работать с четырьмя битами. Такие процессоры называли *четырехразрядными.*

Один из первых персональных компьютеров (Altair, 1974 г.) имел восьмиразрядный процессор, то есть он мог параллельно обрабатывать восемь битов информации. Это в восемь раз быстрее, чем работать с каждым битом отдельно, поэтому в вычислительной технике появилась новая единица измерения информации — *байт.* Байт — это группа из восьми битов.

Мы знаем, что один бит может хранить в себе один двоичный знак — 0 или 1. Это наименьшая единица представления информации — простой ответ на вопрос Да или Нет. А что может хранить байт?

На первый взгляд кажется, что раз в байте восемь битов, то и информации он может хранить в восемь раз больше, чем один бит, но это не так. Дело в том, что в байте важно не только, включен бит или выключен, но и то, в каком месте стоят включенные биты. Байты 0000 0001, 0000 1000 и 1000 0000 — не одинаковые, а разные.

**Это должно быть понятно, если вспомнить, что числа 723, 732, 273, 237, 372 327 различны, хоть и записываются одинаковыми цифрами. Значения чисел зависят не только от того, какие цифры в них входят, но и от того в каких позициях эти цифры стоят.**

Если учесть что важны не только нули и единицы, но и позиции, в которых они стоят, то с помощью одного байта можно выразить 256 различных единиц информации (от 0 до 255).

0000 0000 = 0

0000 0001 *=* 1

0000 0010 = 2

0000 0011 = 3

0000 0100 = 4

0000 0101 = 5

1111 1100 = 252

1111 1101 = 253

1111 1110 = 254

1111 1111 = 255

**Всегда ли байты состояли из восьми битов? Нет, не всегда. Еще в 60-е годы, когда не было персональных компьютеров и все вычисления проводились на больших электронно-вычислительных машинах (ЭВМ), байты могли быть какими угодно. Наиболее широко были распространены ЭВМ, у которых байт состоял из шести битов, но были и такие, у которых он состоял из четырех и даже из семи битов.**

*Восьмибитный байт* появился достаточно поздно (в начале семидесятых годов), но быстро завоевал популярность. С тех пор понятие о байте, как о группе из восьми битов, является общепризнанным.

# Кодирование текстовой информации байтами

Одним битом можно закодировать два значения: Да или Нет (1 или 0).

Двумя битами можно закодировать уже четыре значения: 00, 01, 10, 11.

Тремя битами кодируются 8 разных значений.

Добавление одного бита удваивает количество значений, которое можно закодировать. При восьми битах уже можно закодировать 256 разных значений. Нетрудно догадаться, что если бы в байте было 9 битов, то одним байтом можно было бы закодировать 512 разных значений, а если бы в нем было 10 битов, то 1024 и т. д.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Биты | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Количество кодируемых значений | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |

Но все-таки в байте не 9 и не 10 битов, а только 8, и потому с его помощью кодируют 256 разных значений. Много это или мало? Смотря для чего. Давайте посмотрим, как с помощью байтов кодируют текстовую информацию.

В русском языке 33 буквы (символа) — для их кодирования достаточно 33 различных байтов. Если мы хотим различать прописные (заглавные) и строчные буквы, то потребуется 66 байтов. Для строчных и прописных букв английского языка хватит еще 52 символов — получается 118. Добавим сюда цифры (от 0 до 9), все возможные знаки препинания: точку, запятую, тире, восклицательный и вопросительный знаки. Добавим скобки: круглые, квадратные и фигурные, а также знаки математических операций: +, –, =, /, \*. Добавим специальные символы, например такие, как: %,$,&, @, — мы видим, что все их можно выразить восемью битами, и при этом еще останутся свободные коды, которые можно использовать для других целей.

Дело осталось за малым: надо всем людям мира договориться о том, каким кодом (от 0 до 255) должен кодироваться каждый символ. Если, например, все люди будут знать, что код 33 означает восклицательный знак, а код 63 — знак вопросительный, то текст, набранный на одном компьютере, всегда можно будет прочитать и распечатать на другом компьютере.

Такая всеобщая договоренность об одинаковом использовании чего-либо называется *стандартом.* Стандарт устанавливает таблицу, в которой записано, каким кодом должен кодироваться каждый символ. Такая таблица называется *таблицей кодов.* В этой таблице должно быть 256 строк, в которых записывается, какой байт какому символу соответствует.

Но здесь-то и начались проблемы. Дело в том, что символы, которые хороши для одной страны, не подходят для другой. В Греции используются одни буквы, в Турции — другие. То, что подходит для Америки, не годится для России, а то, что подходит для России, не подходит для Германии.

Поэтому было принято следующее решение. Таблицу кодов разделили пополам. Первые 128 кодов (с 0 до 127) должны быть стандартными и обязательными для всех стран и всех компьютеров, а во второй половине (с кода 128 до кода 255) каждая страна может делать все, что ей угодно, и создавать в этой половине свой стандарт — *национальный.*

Первую (международную) половину таблицы кодов называют *таблицей ASCII —* ее ввел американский институт стандартизации ANSI. В этой таблице размещаются прописные и строчные буквы английского алфавита, символы чисел от 0 до 9, все знаки препинания, символы арифметических операций и некоторые другие специальные коды.

За вторую половину кодовой таблицы (коды от 128 до 255) стандарт ASCII не отвечает Разные страны могут создавать здесь свои таблицы. Часто бывает, что даже в одной стране в этой половине действуют несколько разных стандартов, предназначенных для разных компьютерных систем. В России, например, содержание этой половины таблицы может подчиняться четырем разным стандартам, каждый из которых действует в какой-то своей, особой области.

**Коды ASCII по-русски произносят как а *эс-цэ-и,* а иногда (в просторечии) еще проще *аски-коды.***

**Стандартная кодировка ASCII**

В русском алфавите буква А имеет первое место, а буква Б – второе. У каждой буквы есть своя позиция. Буква Я имеет позицию номер 33. Мы можем считать, что алфавит — это таблица для кодирования букв.

Стандарт ASCII — это тоже как бы «алфавит», только компьютерный. Он тоже определяет номер каждого символа. Но символов больше, чем букв, потому что к ним относятся еще и цифры, и знаки препинания, и некоторые специальные символы.

Выше мы сказали, что с помощью одного байта можно закодировать 256 разных символов. Еще мы узнали, что стандарт ASCII определяет первую половину кодовой таблицы, то есть, кодировку символов, имеющих номера до 127. Но это не совсем так. На самом деле стандарт ASCII первые 32 кода (от 0 до 31) не определяет. Он оставляет их для так называемых *управляющих кодов,* которые не используются для представления информации, а применяются для управления компьютерами. Эти коды отданы на усмотрение производителей компьютерных систем (у них есть свои соглашения и свои стандарты по применению этих кодов). Еще несколько лет назад людям, работающим с компьютерами, стоило знать некоторые из этих кодов, но сегодня это уже не требуется.

Самый первый символ стандарта ASCII — это ПРОБЕЛ. Он имеет код 32.

За ним идут специальные символы и знаки препинания (коды с 33 по 47).

Далее идут десять цифр (коды 48-57).

Коды 58—64 используют некоторые математические символы и знаки препинания.

Самое интересное начинается с кодов 65—90. Ими обозначают прописные английские буква от А до Z

Коды 91—96 используются для специальных символов.

Коды 97—122 — строчные буквы английского алфавита.

Коды 123-127 — специальные символы.

**Коды верхней половины таблицы символов (128-255) отданы для национальных стандартов. Когда мы узнаем о компьютере немного больше, мы разберемся с несколькими российскими стандартами и посмотрим, как кодируются буквы русского языка.**

Имея под рукой кодовую таблицу символов, вы можете легко определить, какие слова закодированы следующими байтами

67 79 77 80 85 84 69 82 99 111 109 112 117 116 101 114

У вас раскодирование информации займет пару минут. Компьютер сделает это за несколько миллионных долей секунды.

Кодирование цветовой информации

С помощью одного байта можно закодировать 256 разных значений. Мы уже знаем, что этого вполне хватает и на русские, и на английские буквы и на знаки препинания. А давайте посмотрим, хватит ли этою для кодирования графической информации. И начнем с кодирования цвета.

Легко догадайся, что одним байтом можно закодировать 256 различных цветов. В принципе, этого достаточно для рисованных изображений типа тех, что мы видим в мульфильмах, но для полноцветных изображении живой природы — недостаточно. Человеческий глаз — не самый совершенный инструмент, но и он может различать десятки миллионов цветовых оттенков.

А что, если на кодирование цвета одной точки отдать не один байт, а два, то есть, не 8 битов, а 16. Мы уже знаем, что добавление каждого бита увеличивает в два раза количество кодируемых значений. Добавление восьми битов восемь раз удвоит это количество, то есть увеличит его в 256 раз (2х2х2х2х2х2х2х2=256) Двумя байтами можно закодировать 256х256=65 536 различных цветов. Это уже лучше и похоже на то, что мы видим на фотографиях и на картинках в журналах, но все равно хуже, чем в живой природе.

Если для кодирования цвета одной точки использовать 3 байта (24 бита), то количество возможных цветов увеличится еще в 256 раз и достигнет 16,5 миллионов. Этот режим позволяет хранить, обрабатывать и передавать изображения, не уступающие по качеству наблюдаемым в живой природе.

Возможно, вы знаете, что любой цвет можно представить в виде комбинации трех основных цветов: красного, зеленого и синего (их называют *цветовыми составляющими).* Если мы кодируем цвет точки с помощью трех байтов, то первый байт выделяется красной составляющей, второй — зеленой, а третий — синей. Чем больше значение байта цветовой составляющей, тем ярче этот цвет.

*Белый цвет.* Если точка имеет белый цвет, значит, у нее есть все цветовые составляющие, и они имеют полную яркость. Поэтому белый цвет колируется тремя полными байтами 255, 255, 255.

*Черный цвет.* Он означает отсутствие всех прочих цветов. Все цветовые составляющие равны нулю. Черный цвет кодируется байтами 0, 0, 0.

*Серый цвет.* Это цвет, промежуточный между черным и белым. В нем есть все цветовые составляющие, но они одинаковы и нейтрализуют друг друга. Например, серый цвет может быть таким 100, 100, 100 или таким: 150, 150, 150. Можно догадаться, что во втором случае яркость выше, и второй вариант серого цвета светлее первого.

*Красный цвет.* У него все составляющие, кроме красной, равны нулю. Это может быть, например, темно-красный цвет: 128, 0, 0 или ярко-красный: 255, 0, 0.

То же относится и к синему цвету (0, 0, 255) и к зеленому (0, 255, 0).

Задавая любые значения (от 0 до 255) для каждого из трех байтов, с помощью которых кодируется цвет, можно закодировать любой из 16,5 миллионов цветов.

Кодирование графической информации

Итак, мы уже умеем с помощью чисел кодировать цвет одной точки. На это необходимы один, два или три байта, в зависимости от того, сколько цветов мы хотим передать. А как закодировать целый рисунок?

Решение приходит само собой — надо рисунок разбить на точки. Чем больше будет точек и чем мельче они будут, тем точнее будет передача рисунка. А когда рисунок разбит на точки, то можно начать с его левого верхнего угла и, двигаясь по строкам слева направо, кодировать цвет каждой точки.

Взгляните на рисунок справа. Книжка у нас черно-белая, и цветной рисунок в ней показать нельзя, поэтому мы не будем кодировать точки этого рисунка тремя байтами — нам достаточно и одного байта на каждую точку.

Код 0 обозначает черную точку, код 255 — белую. Коды 1-254 обозначают серые точки. Чем выше значение кода, тем светлее точка.

Когда все точки рисунка закодированы, получается следующая последовательность байтов:

176, 176, 176, 128, 64, 64, 64, 80, 64, 64, 80, 80, 80, 80, 80…

**Если бы рисунок был цветным, то для каждой точки вместо одного байта стояло бы три байта и вся последовательность была бы втрое длиннее.**

Закодировать рисунок оказалось несложно, а вот как его раскодировать, чтобы опять получить то, что было? Если раскодировать байты по одному слева направо, то никогда не узнаешь, где кончается одна строка и начинается другая.

Это говорит о том, что нам чего-то не хватает. Значит, мы что-то важное упустили из виду. Если бы перед группой байтов приписать еще небольшой *заголовок,* из которого было бы ясно, как надо эти байты раскодировать, то все стало бы на свои места. Этот заголовок может быть, например таким: {8х8}. По нему можно догадаться, что рисунок должен состоять из восьми строк по восемь точек в каждой строке.

Заголовок можно сделать еще подробнее, например так: {8х8х3} — тогда можно догадаться, что это рисунок цветной, в котором на кодирование цвета каждой точки использовано три байта.

Заголовок помогает решить многие вопросы, но возникает новая проблема. Как компьютер разберется, где заголовок, а где сама информация? Ведь заголовок тоже должен быть записан в виде байтов. Сумеет ли компьютер отличить байты заголовка от байтов информации? Далее мы с этим разберемся.

# Понятие формата информации

Идея представить любую информацию в виде чисел и закодировать их байтами очень рациональна. Компьютеру удобно работать, когда тексты, звуки, рисунки и видеофильмы представлены в виде байтов со значениями от 0 до 255. Непонятно только, как он отличит, где и что записано.

Возьмем несколько байтов: 70, 79, 82, 77, 65, 84. Что здесь записано?

• Может быть, это две цветные точки: первая с цветом 70, 79, 82, а вторая: 77, 65, 84?

• Может быть, это шесть серых точек (одни чуть светлее, а другие чуть темнее).

• Может быть, этими байтами закодирована дата и время запуска очередного спутника Земли?

• Может быть, это начало какой-то музыкальной мелодии?

Это может быть вообще все, что угодно, в том числе и английское слово FORMAT, закодированное по стандарту ASCII (проверьте, не так ли это на самом деле).

Если компьютер не знает, что выражает каждая группа байтов, он не сможет ничего с ней сделать. Он должен различать, где байтами закодирован текст, а где музыка и рисунки. Тексты должны всегда оставаться текстами, числа — числами, даты — датами, рисунки — рисунками, музыка — музыкой, а деньги, хранящиеся в банковском компьютере в виде тех же самых байтов, должны оставаться деньгами и не превращаться в звук и музыку.

Решение этой проблемы опять-таки связано с заголовком. Если бы перед группой байтов стоял специальный заголовок, то компьютер точно знал бы, что эти байты обозначают. А чтобы компьютер знал, где кончаются байты заголовка и начинаются байты данных, заголовок и данные должны иметь строго определенный *формат.* Для разных видов информации используются разные форматы. Например, если это черно-белая картинка, то *каждый* байт после заголовка определяет яркость точки, а если это цветная картинка, то цвет одной точки может определять не один байт, а несколько байтов.

Понятие о файле

Итак, мы поняли, что любая последовательность байтов может выражать все что угодно, но надо знать, в каком *формате* информация записана, есть ли у нее *заголовок,* где он начинается и где заканчивается.

Если мы пишем контрольную работу на отдельных листочках, а потом сдаем ее учителю на проверку, то как учитель узнает, какой ученик написал какую работу? Очень просто — каждый ученик свою работу подписывает, то есть *выполнит регистрацию.*

Точно так же и в компьютере. Каждая последовательность байтов, содержащая информацию определенного типа, должна быть зарегистрирована. После регистрации эта последовательность получает уникальное имя и называется *файлом.* Любая информация, сохраняемая на компьютере, должна быть зарегистрирована как файл.

Мы уже говорили о том, что наименьшей единицей *представления* информации является бит. Наименьшей единицей *обработки* или *передачи* информации является байт. Теперь мы узнали наименьшую единицу *хранения* информации — это файл. Ни байт, ни бит нельзя сохранить в качестве информации, поскольку непонятно, что они обозначают (то ли буквы, то ли ноты, то ли еще что-то). Файл можно сохранить, потому что он регистрируется, даже если в нем только один байт.

**Простой пример. Если зайти в школьную библиотеку и попросить выдать букву «А», то библиотекарь этого сделать не сможет, хотя у него есть тысячи книг, в которых встречаются миллионы букв «А». Буквы в библиотеке не зарегистрированы. Другое дело — книги, журналы, газеты. Выбирайте любую по каталогу. Они зарегистрированы. В них вы найдете нужную информацию.**

*Файл — это наименьшая единица хранения информации, содержащая последовательность байтов и имеющая уникальное имя.*

По имени файла компьютер определяет, где файл находится, какая информация в нем содержится, в каком формате она записана и какими программами ее можно обработать. Имя файла имеет очень большое значение, и мы к нему обязательно вернемся, но сделаем это несколько позже.