§ 5. Кодирование информации

Кодирование – любое преобразование информации, идущей от источника, в форму, пригодную для ее передачи по каналу связи.

Язык – система кодирования понятий

Азбука – кодирование слов языка

Чертежи, ноты,

математические выкладки

Декодирование - это процесс восстановления содержания закодированной информации.

Двоичное кодирование - универсальная форма представления различных видов информации для дальнейшей ее обработки средствами ВТ.

Способ кодирования зависит от цели, ради которой осуществляется.

Три основных способа кодирования информации:

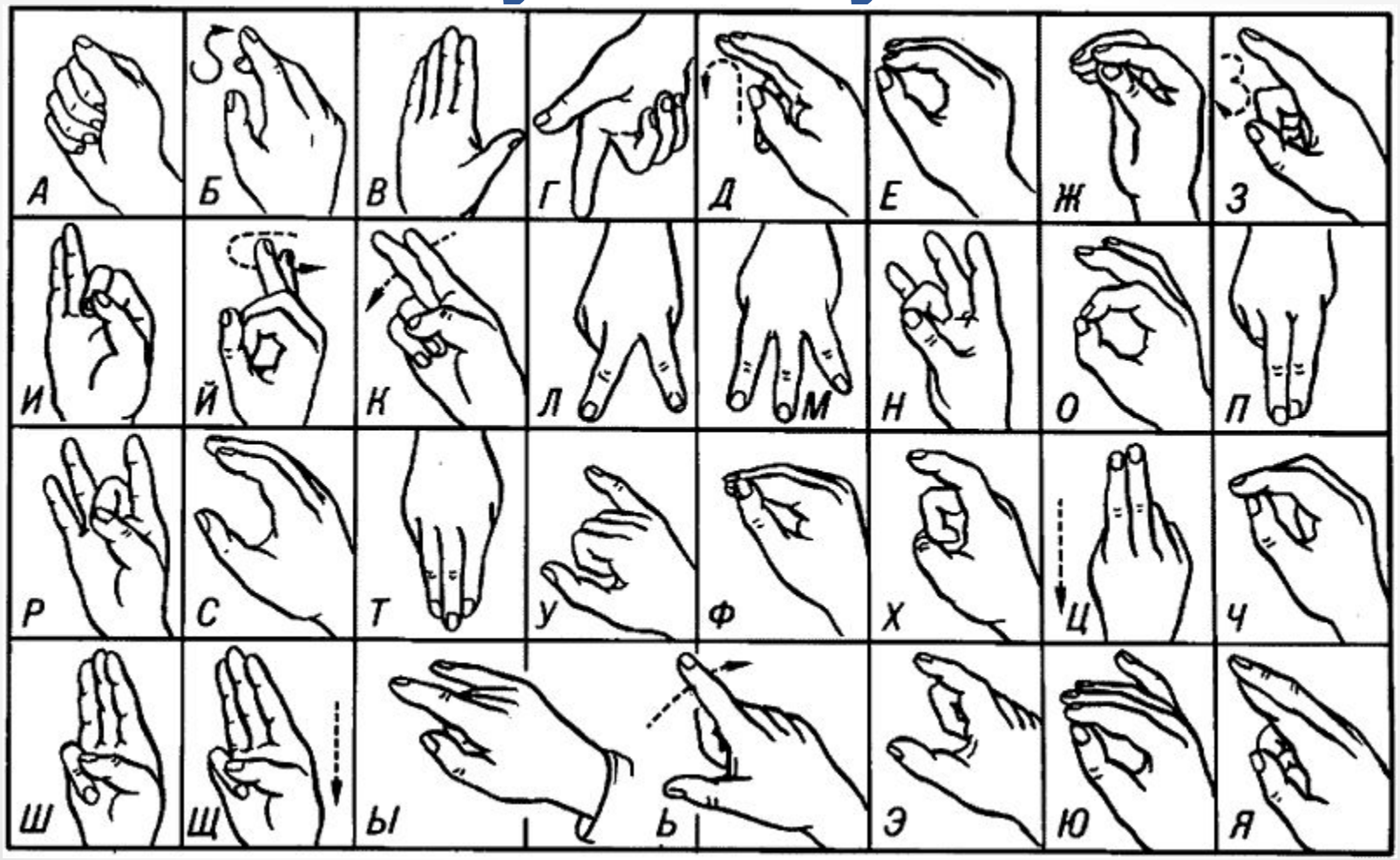
- **графический** - с помощью рисунков или значков;
- **числовой** - с помощью чисел;
- **символьный** - с помощью символов того же алфавита, что и текст.

Множество кодов очень прочно вошло в нашу жизнь. **Числовая** информация кодируется арабскими или римскими цифрами.

Примеры кодирования

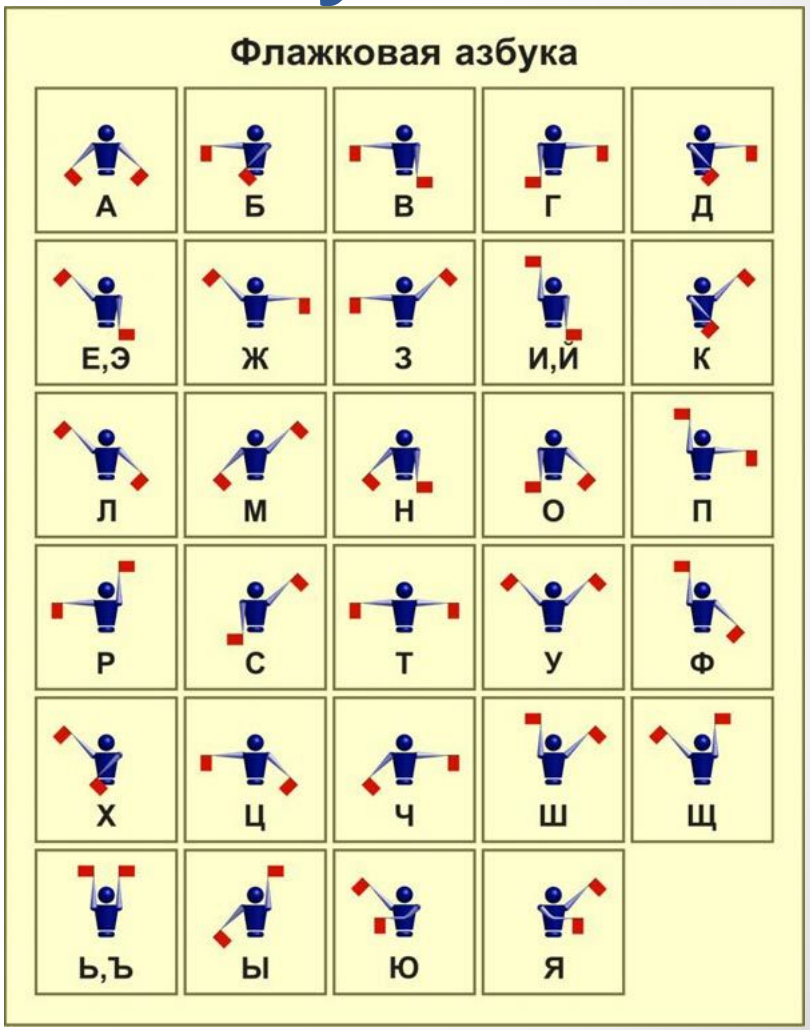
В середине 19 века французский педагог Луи Брайль придумал специальный шрифт для слепых. Буквы этого шрифта выдавливались на листках плотной бумаги. Проводя пальцами по образовавшимся от уколов выступам, люди учатся различать буквы.

Азбука глухих



Флажковая азбука

- Семафорная азбука
- Существующую сегодня на флоте русскую семафорную азбуку разработал в 1895 году вице-адмирал С.О.Макаров
- Не содержит цифр и знаков препинания



5.1. Кодирование текстовой информации

- **Код** – уникальное беззнаковое целое двоичное число, поставленное в соответствие некоторому символу
- **Алфавит** – совокупность вводимых и отображаемых символов

  Алфавит компьютерной системы включает в себя:

  арабские цифры, буквы латинского алфавита, знаки препинания, буквы национального алфавита, символы псевдографики
- **Кодовая таблица** – система, в которой каждому символу алфавита поставлен в соответствие уникальный код

Кодовая таблица ASCII

- ASCII – American Standard Code of Information Interchange
- принят в 1981г. Институтом стандартизации США
- для хранения кода одного символа отвели 1 байт
- таблица состоит из двух частей:
  - **основная** (не может быть изменена) включает арабские цифры, буквы латинского алфавита, знаки препинания
  - **расширенная** часть отдана национальным алфавитам, символам псевдографики

Кодовая таблица ASCII

0 -	16 - ▶	32 -	48 - 0	64 - @	80 - P	96 - '	112 - p
1 - ☺	17 - ◀	33 - !	49 - 1	65 - A	81 - Q	97 - a	113 - q
2 - ☻	18 - ↕	34 - "	50 - 2	66 - B	82 - R	98 - b	114 - r
3 - ♥	19 -‼	35 - #	51- 3	67 - C	83 - S	99 - c	115 - s
4 - ♦	20 - ¶	36 - $	52 - 4	68 - D	84 - T	100 - d	116 - t
5 - ♣	21 - §	37 - %	53 - 5	69 - E	85 - U	101 - e	117 - u
6 - ♣	22 - ▬	38 -&	54 - 6	70 - F	86 - V	102 - f	118 - v
7 -	23 - ↕	39 - '	55 - 7	71 - G	87 - W	103 - g	119 - w
8 -	24 - ↑	40 - (	56 - 8	72 - H	88 - X	104 - h	120 - x
9 -	25 - ↓	41 -)	57 - 9	73 - I	89 - Y	105 - i	121 - y
10 -	26 - →	42 - *	58 - :	74 - J	90 - Z	106 - j	122 - z
11 -	27 - ←	43 - +	59 - ;	75 - K	91 - [	107 - k	123 - {
12 -	28 - ∟	44 - ,	60 - <	76 - L	92 - \	108 - l	124 - |
13 -	29 - ↔	45 - -	61 - =	77 - M	93 - j	109 - m	125 - }
14 - ♫	30 - ▲	46 - .	62 ->	78 - N	94 - ^	110 - n	126 - ~
15 - ☼	31 - ▼	47 - /	63 - ?	79 - O	95 - ÷	111 - o	127 - ⌂
16 - ▶	32 -	48 -0	64 - @	80 - P	96 -	112 - p	

128 -А	144 - Р	160 - а	176 - ▒	192 - └	208 - ╨	224 -р	240 - Ё
129 - Б	145 - С	161 - б	177 - ▒	193 - ┴	209 - ╤	225 -с	241- ё
130 -В	146 - Т	162 - в	178 - ▓	194 - ┬	210 - ╥	226 -т	242 - Є
131 - Г	147 - У	163 - г	179 - |	195 - ├	211 - ╙	227 -у	243 - є
132 -Д	148 - Ф	164 - д	180 - ┤	196 - —	212 - ╘	228 -ф	244 - Ї
133 -Е	149 - Х	165 - е	181 - ╡	197 - ┼	213 - ╒	229 - х	245 - ї
134 - Ж	150 - Ц	166 - ж	182 - ╢	198 - ╞	214 - ╓	230 - ц	246 - Ў
135 -З	151 - Ч	167 - з	183 - ╖	199 - ╟	215 - ╫	231 - ч	247 - ў
136 -И	152 -Ш	168 - и	184 - ╕	200 - ╚	216 - ╪	232 - ш	248 - °
137 - Й	153 -Щ	169 - й	185 - ╣	201 - ╔	217 - ┘	233 - щ	249 - ●
138- К	154 - Ъ	170 - к	186 - ║	202 - ╩	218 - ┌	234 - ъ	250 - .
139 -Л	155 - Ы	171 - л	187 - ╗	203 - ╦	219 - ■	235 - ы	251 - √
140 -М	156 - Ь	172 - м	188 - ╝	204 - ╠	220 - ▄	236 - ь	252- №
141 -Н	157 - Э	173 - н	189 - ╜	205 - ═	221 - ▌	237 - э	253 - ¤
142 -О	158 - Ю	174 - о	190 - ╛	206 - ╬	222 - ▐	238 - ю	254 - ▮
143- П	159 - Я	175 - п	191 - ┐	207 - ╧	223 - ▀	239 - я	255 -
144 - Р	160 - а	176 - ▒	192 - └	208 - ╨	224 - р	240 - Ё	

Кодовые страницы для русского языка

Замену символов во второй половине кодовой таблицы можно произвести разными способами ⇒ существует несколько разных таблиц кодировки символов кириллицы. Все они одинаково изображают символы первой половины таблицы (от 0 до 127) и различаются представлением символов русского алфавита и псевдографики.

- Windows-1251 (CP1251)
- КОИ-8 (KOI8-RU) – российский сектор Интернета
- MacCyrillic

Unicode

- Принята в 1991г.
- Содержит два раздела:
  - универсальный набор символов UCS
  - семейство кодировок UTF
- Для хранения кода одного символа отвели 2 байта (65 536 символов).
- К настоящему времени задействовано около 49 000 кодов. Последнее значительное изменение - введение символа валюты EURO в сентябре 1998г.).

5.2. Кодирование числовой информации

Особенности представления чисел в компьютере

1. предельные значения чисел

Переполнение разрядной сетки — это ситуация, когда число, которое требуется сохранить, не умещается в имеющемся количестве разрядов вычислительного устройства.

Переполнение вызвано **ограниченным** числом разрядов. Аналогично для отрицательных чисел.

(англ. overflow — переполнение “сверху”).

Особенности представления чисел в компьютере

2. Не все вещественные числа могут быть представлены в компьютере точно.

При ограниченном числе разрядов дробной части существует некоторое минимальное ненулевое значение C_{min}, которое можно записать в памяти. Любое значение, меньшее чем C_{min}, неотличимо от нуля. Такой эффект принято называть **антипереполнением** (англ. underflow - переполнение «снизу»).

Особенности представления чисел в компьютере

Т.к. вещественные числа хранятся в памяти приближенно, сравнивать их (особенно если они являются результатами сложных расчетов) необходимо с большой осторожностью. Нужно проверять условие $|1 - X \times Y| < \varepsilon$, где ε — малая величина, которая задает нужную точность вычислений. Для большинства практических задач достаточно взять ε порядка 10^{-2}... 10^{-4}, а ошибка компьютерных расчетов обычно значительно меньше (не более 10^{-7}).

Особенности представления чисел в компьютере

3. Целые и вещественные числа в компьютере хранятся и обрабатываются по-разному:
   - исключить все проблемы, связанные с неточностью представления;
   - операции с целыми числами выполняются значительно быстрее, чем с вещественными. В ядре современных процессоров реализованы только целочисленные арифметические действия, а для вещественной арифметики используется специализированный встроенный блок — **математический сопроцессор**.
   - позволяет экономить память, используя **типы данных**

Тип данных

Тип данных помогает решить задачу интерпретации кодов (обратную задаче представления данных в двоичном коде).

Тип данных определяет:

- **множество значений**, которые могут иметь данные этого типа;
- **правила машинного** (битового/байтового) **представления** допустимых значений;
- **набор операций**, которые можно выполнять над данными этого типа.

Действительное число

Числа могут быть:

- целые точные,
- дробные точные,
- рациональные,
- иррациональные,
- дробные приближенные,
- положительные,
- отрицательные,
- «карликами» (масса атома),
- «гигантами» (масса Земли).

Представление целых чисел

1. **Двоично-десятичное** представление - форма записи целых чисел, когда каждый десятичный разряд числа записывается в виде его четырёхбитного двоичного кода.

   311_{10}=0011 0001 $0001_{2\text{-}10}$

   Применяется в калькуляторах.
2. Представление в **прямом** коде: числа представляются непосредственно в двоичной системе счисления, знак числа можно закодировать отдельным битом.

   311_{10}= 0001 0011 0111_{2}

   -311_{10}=1001 0011 0111_{2}

Представление целых чисел

Прямой код не применяется в компьютерах для представления целых чисел (используется для вещественных), т.к.

- число 0 кодируется двумя способами
- для разных сочетаний знаков действия над числами выполняются по разному.

3.Представление в **дополнительном** коде:

- положительные числа - в прямом коде
- отрицательные числа получаются по следующему алгоритму:
  - Вычислить число X – 1 и перевести его в двоичную систему.
  - Выполнить инверсию каждого разряда результата.

Представление целых чисел

Пример. Получить десятичное значение числа по его дополнительному коду 10010111_2 .

1. Задано отрицательное число, так как в старшем разряде стоит 1;
2. Инвертируем полученный код: 01101000_2 (получили модуль отрицательного числа);
3. Переведем в десятичную систему счисления: $01101000_2=1\cdot2^6+1\cdot2^5+1\cdot2^3=64+32+8=104_{10}$;
4. К полученному числу добавим 1:

$$104_{10}+1_{10}=105_{10};$$

- Записать число со знаком «-». Ответ: $X=-105_{10}$.

Представление целых чисел

Название типа	Число бит	Max число	Min число
Byte	8	0	255
Shortint	8	127	-128
Word	16	0	65 535
Integer	16	32 767	-32 768
Longint	32	2 147 483 647	-2 147 483 648
	64	9 223 372 036 854 775 807	- 9 223 372 036 854 775 808

Представление вещественных чисел

Применяются:

- кодирование с **фиксированной** запятой

Применялось в первых ЭВМ. Положение запятой, отделяющей целую часть от дробной, было жестко закреплено в разрядной сетке конкретной ЭВМ - раз и навсегда для всех чисел и для всех технических устройств этой машины. Программист подготавливал данные, масштабируя их.

- кодирование с **плавающей** запятой

Представление вещественных чисел

Т.к. вещественные числа хранятся в памяти приближенно, сравнивать их (особенно если они являются результатами сложных расчетов) необходимо с большой осторожностью. Нужно проверять условие $|1 - X \times Y| < \varepsilon$, где ε — малая величина, которая задает нужную точность вычислений. Для большинства практических задач достаточно взять ε порядка 10^{-2}... 10^{-4}, а ошибка компьютерных расчетов обычно значительно меньше (не более 10^{-7}).

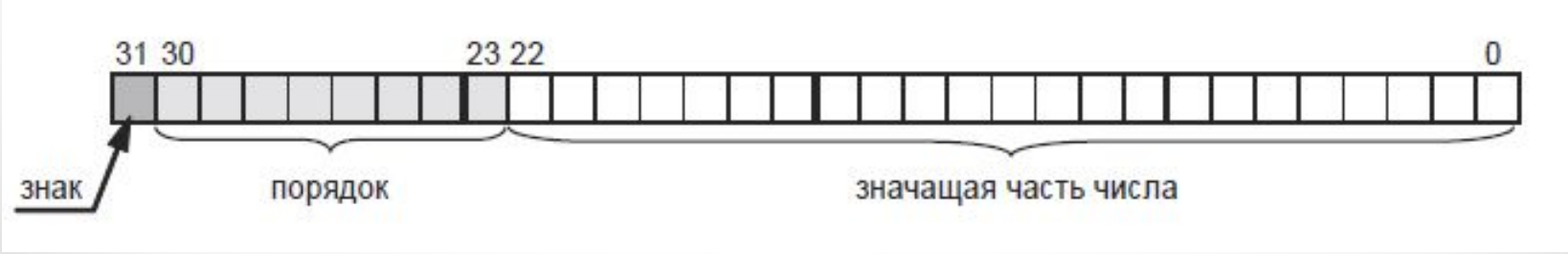
Представление вещественных чисел

Существуют два способа представления чисел с плавающей запятой:

- целая часть нулевая, первая цифра дробной части ненулевая – оптимально с теоретической точки зрения
- целая часть состоит из единственной ненулевой цифры – оптимально с практической точки зрения. Стандарт IEEE 754 (арифметика вещественных чисел в современных компьютерах)

Представление вещественных чисел

В компьютере используется такое представление вещественных чисел с плавающей запятой, при котором значащая часть Z удовлетворяет условию $1 \le Z < B$, где B - основание системы счисления. Такое представление называется **нормализованным**. Фактически хранится две величины числа: значащая часть (точность вычислений) и порядок (диапазон представления).



Представление вещественных чисел

Характеристики стандартных вещественных типов данных, используемых в математическом сопроцессоре Intel

Тип	Диапазон	Число десятичных значащих цифр	Размер (байт)
single (одинарная точность)	$1{,}4 \cdot 10^{-45} – 3{,}4 \cdot 10^{38}$	7–8	4
double (двойная)	$4{,}9 \cdot 10^{-324} – 1{,}8 \cdot 10^{308}$	15-16	8
extended (расширенная)	$3{,}6 \cdot 10^{-4951} – 1{,}2 \cdot 10^{4932}$	18-19	10

Планировка бит для вещественных типов

Точность	Одинарная	Двойная	Расширенная
Размер (байты)	4	8	10
Обозначение полей	S-E-M	S-E-M	S-E-I-M
Размеры полей (бит)	1-8-23	1-11-52	1-15-1-63
Смещение	127	1023	32766

Представление вещественных чисел

Пример. Представить число $-312{,}3125_{10}$ в формате с плавающей запятой с двойной точностью.

В формате с двойной точностью порядок занимает 11 бит ($2^{11}=2048_{10}$ с учетом цифры 0 и симметричного распределения положительных и отрицательных диапазонов (2048-2)/2=1023) и имеет диапазон от 2^{-1023} до 2^{1023}, поэтому смещение равно

$$1023_{10} = 1111111111_2$$

Перевод в двоичную систему счисления:

$$-312{,}3125_{10}=100111000{,}0101_2.$$

Нормализация:

$$100111000{,}0101_2=1{,}00111000010_2\cdot 2^8.$$

Представление вещественных чисел

Вычисление смещения:

$$8+1023=1031=10000000111_2.$$

Представление в памяти ЭВМ:

63	62..52	51..0
Знак	Смещение	Значащая часть + 40 нулей
1	10000000111	00111000010100000000000000000000000000000000000000..0

Записываем компактно в шестнадцатиричной форме (деля разряды справа налево по 4):

1100.0000.0111.0011.1000.0101.0000.0000.0000.0000.0000..

C0.73.85.00.00.00.00.00

5.3. Кодирование графической информации

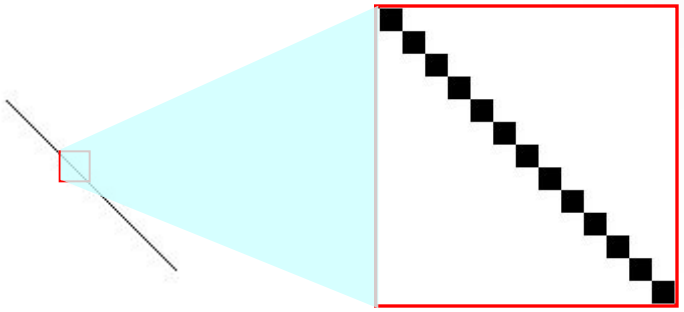
Изображения в компьютере закодированы в виде двоичных последовательностей.

Используют два принципиально разных метода кодирования:

Два типа кодирования рисунков

- **растровое кодирование**

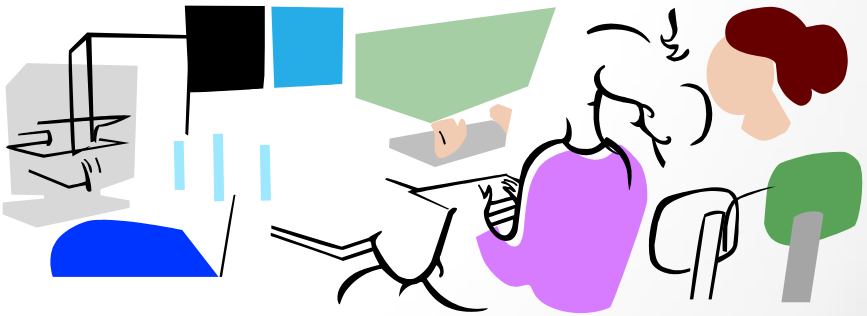
точечный рисунок, состоит из **пикселей**



фотографии, размытые изображения

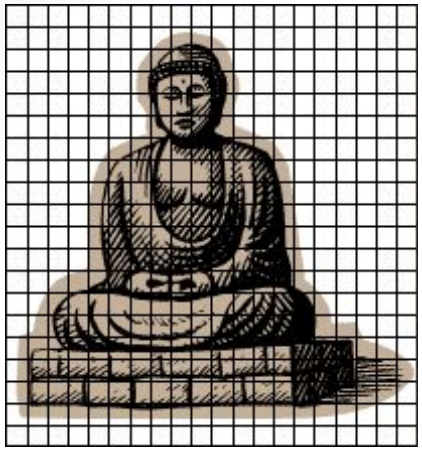
- **векторное кодирование**

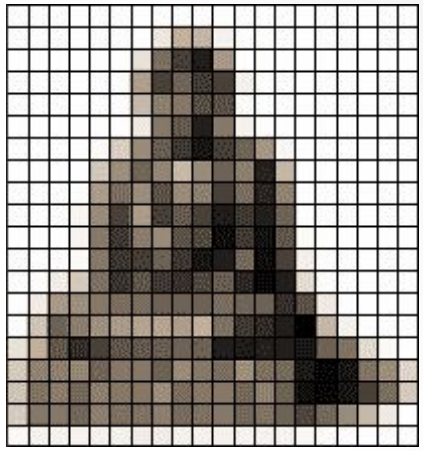
рисунок, состоит из **отдельных геометрических фигур**



чертежи, схемы, карты

Растровое кодирование





Шаг 1. Дискретизация: разбивка на *пиксели*.

Пиксель – это наименьший элемент рисунка, для которого можно независимо установить цвет.

Шаг 2. Для каждого пикселя определяется **единый цвет**.

Есть потеря информации!

Разрешение – количество пикселей, приходящихся на единицу линейного размера изображения
число пикселей на дюйм, *pixels per inch (ppi)*

- **экран 96** ppi, **печать 300-600** ppi, **типография 1200** ppi

Кодирование цвета

Человек воспринимает свет как множество электромагнитных волн. Определенная длина волны соответствует некоторому цвету.

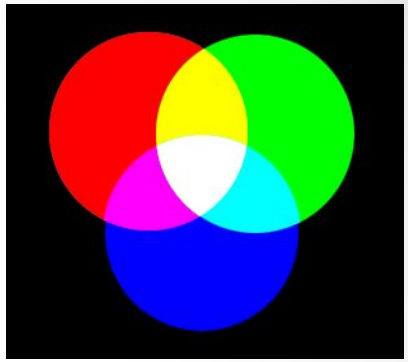
зеленый 500-565нм

Для представления цвета в виде числового кода используются так называемые **цветовые модели**.

Цветовая модель – способ представления большого количества цветов посредством разложения его на простые составляющие.

Растровое кодирование (*True Color*)

Шаг 3. От цвета – к числам: модель RGB



цвет = **R** + **G** + **B**

red	***green***	***blue***
красный	зеленый	синий
0..255	0..255	0..255

R = 218
G = 164
B = 32

R = 135
G = 206
B = 250

Шаг 4. Числа – в двоичную систему.

Сколько разных цветов можно кодировать?

Глубина цвета

256·256·256 = **16 777 216 (*True Color*)**

Сколько памяти нужно для хранения цвета 1 пикселя?

R: 256=2^8 вариантов, нужно 8 бит = 1 байт

R G B: всего 3 байта

Растровое кодирование *(High Color)*

На красную и синюю составляющие отводится по 5 битов, на зеленую, к которой человеческий глаз более чувствителен, - 6 битов.

В режиме High Color можно закодировать

$$2^5 \cdot 2^5 \cdot 2^6 = 65\,536$$ **различных цветов**

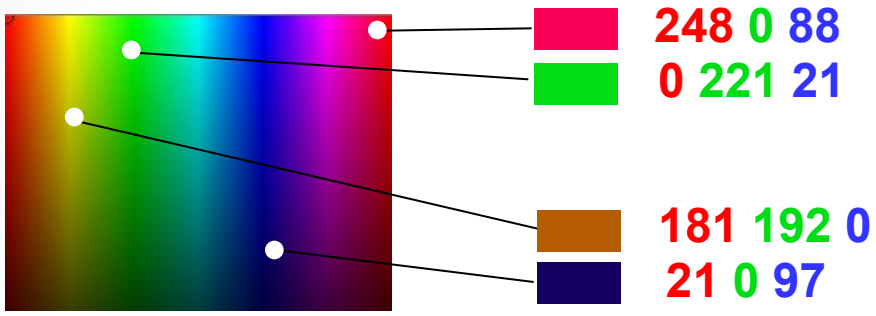
В мобильных телефонах иногда применяют 12-битное кодирование цвета:

4 бита на канал, 4096 цветов

Растровое кодирование с палитрой

Шаг 1. Выбрать количество цветов: 2, 4, … **256.**

Шаг 2. Выбрать 256 цветов из палитры:



Шаг 3. Составить палитру (каждому цвету – номер 0..255)

палитра хранится в начале файла

0	1		254	255
248 0 88	0 221 21	…	181 192 0	21 0 97

Шаг 4. Код пикселя = номеру его цвета в палитре

2	45	65	14	…	12	23

Растровое кодирование с палитрой

Файл с палитрой:

Сколько занимает палитра и основная часть?

Один цвет в палитре: **3 байта (RGB)**

256 = 2^8 цветов:

палитра	256·3 = 768 байт
рисунок	8 бит на пиксель

16 цветов:

палитра	16·3 = 48 байт
рисунок	4 бита на пиксель

2 цвета:

палитра	2·3 = 6 байт
рисунок	1 бит на пиксель

Глубина цвета

Пример решения задачи

Для хранения растрового изображения размером 128 ×128 пикселей отвели 4 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

Решение:

Общее количество информации:

$$4 \text{ Кб} = 2^2 \times 2^{10} \text{ байт} = 2^{12} \text{ байт}$$

Количество пикселей в изображении:

$$128 \times 128 = 2^7 \times 2^7 = 2^{14}$$

Объем информации на 1 пиксель:

2^{12} байт/ 2^{14} = 1 байт / 2^2 = 2^3 бита / 2^2 = 2 бита

2 бита может закодировать 2^2 = 4 цвета

Форматы файлов (растровые рисунки)

- ✔ **BMP** (bitmap) – стандартный формат растровых изображений в ОС Windows; поддерживает кодирование с палитрой и в режиме истинного цвета;
- ✔ **JPEG** (Joint Photographic Experts Group) – формат, разработанный специально для кодирования фотографий; для уменьшения объема файла используется сильное сжатие, при котором изображение немного искажается.

Не рекомендуется использовать для рисунков с четкими границами

Форматы файлов (растровые рисунки)

- **GIF** (Graphics Interchange Format) – формат, поддерживающий только кодирование с палитрой (от 2 до 256 цветов);

части рисунка могут быть прозрачными, т.е. на веб-странице через них будет «просвечивать» фон;

в современном варианте формата GIF можно хранить анимированные изображения;

используется сжатие без потерь.

Форматы файлов (растровые рисунки)

- **PNG** (Portable Network Graphics) – формат, поддерживающий как режим истинного цвета, так и кодирование с палитрой;

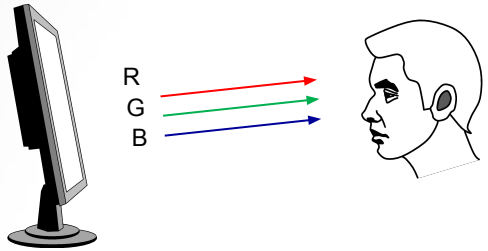
части рисунка могут быть прозрачными и даже полупрозрачными (32-битное кодирование RGBA, где четвертый байт задает прозрачность);

используется сжатие без потерь.

Форматы файлов (растровые рисунки)

Формат	True Color	Палитра	Прозрачность
BMP			
JPG			
GIF			
PNG			

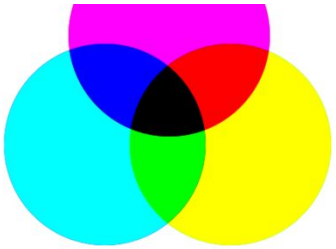
Кодирование цвета при печати



Белый – красный = голубой **C = Cyan**

Белый – зелёный = пурпурный **M = Magenta**

Белый – синий = желтый **Y = Yellow**



Модель CMY

C	M	Y	
0	0	0	
255	255	0	
255	0	255	
0	255	255	
255	255	255	

Модель CMYK: + *Key color*

Меньший расход краски и лучшее качество для чёрного и серого цветов.

Растровые рисунки

- лучший способ для хранения **фотографий** и изображений без четких границ
- **спецэффекты** (тени, ореолы, и т.д.)

- есть **потеря информации** (почему?)
- при изменении размеров рисунка он **искажается**
- **размер файла** не зависит от сложности рисунка (а от чего зависит?)

Какие свойства цифрового рисунка определяют его качество?

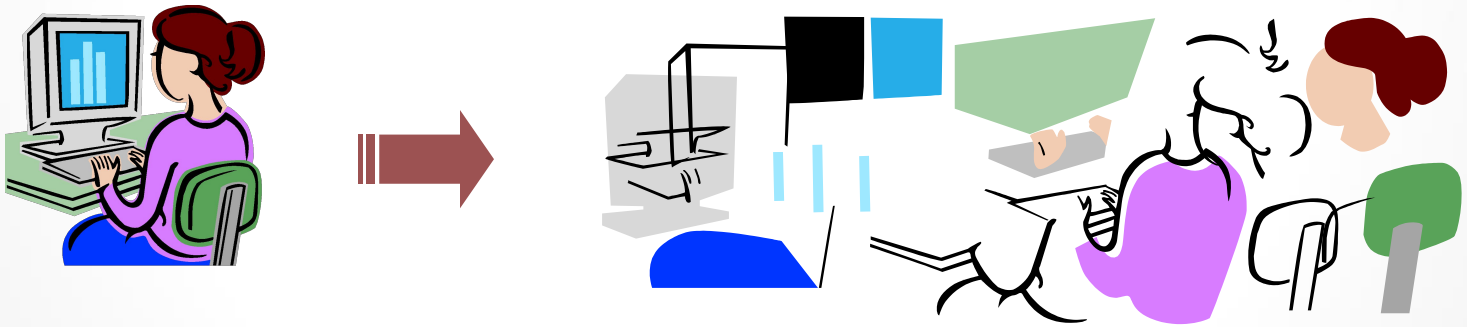
Векторные рисунки

Строятся из геометрических фигур:

- отрезки, ломаные, прямоугольники
- окружности, эллипсы, дуги
- сглаженные линии (кривые Безье)

Для каждой фигуры в памяти хранятся:

- размеры и координаты на рисунке
- цвет и стиль границы
- цвет и стиль заливки (для замкнутых фигур)

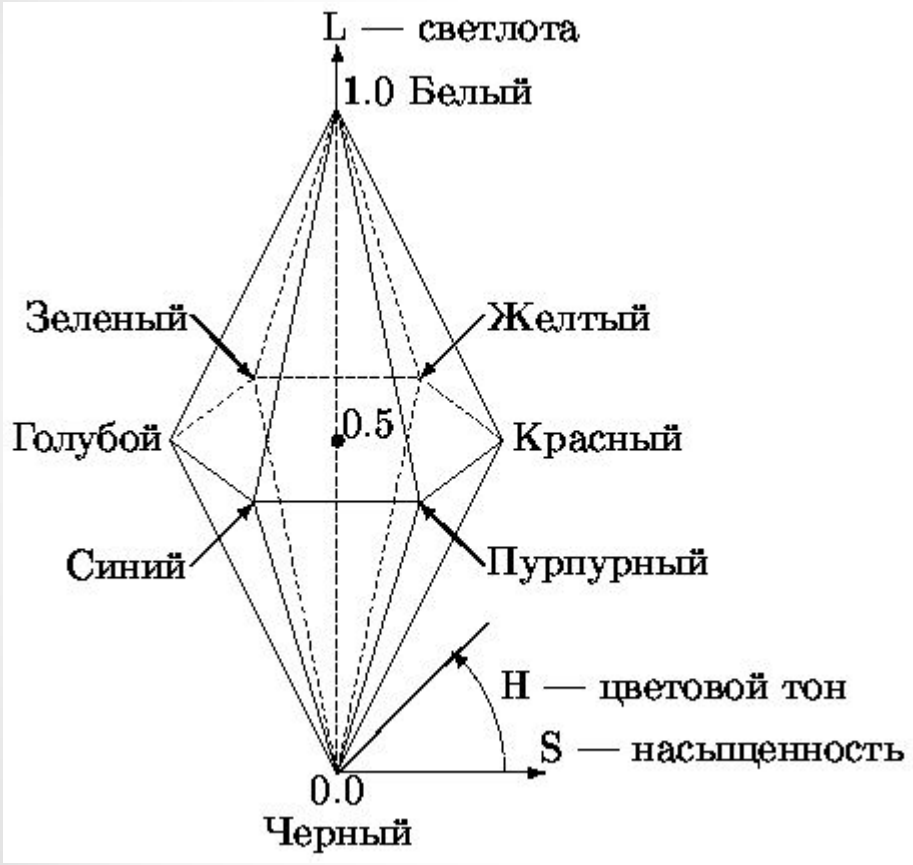


Форматы файлов:

- **WMF** *(Windows Metafile)*
- **CDR** *(CorelDraw)*
- **AI** *(Adobe Illustrator)*
- **SVG** *(Inkscape)*

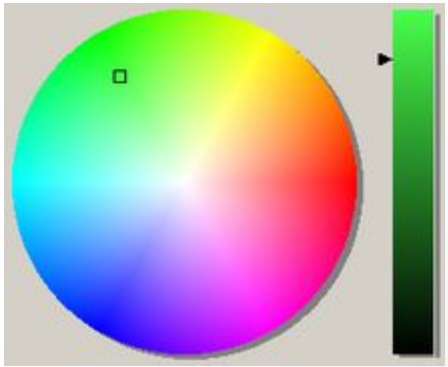
для Web

Модель HSB



- Hue – тон, оттенок
- Saturation – насыщен-ность
- Brightness – яркость

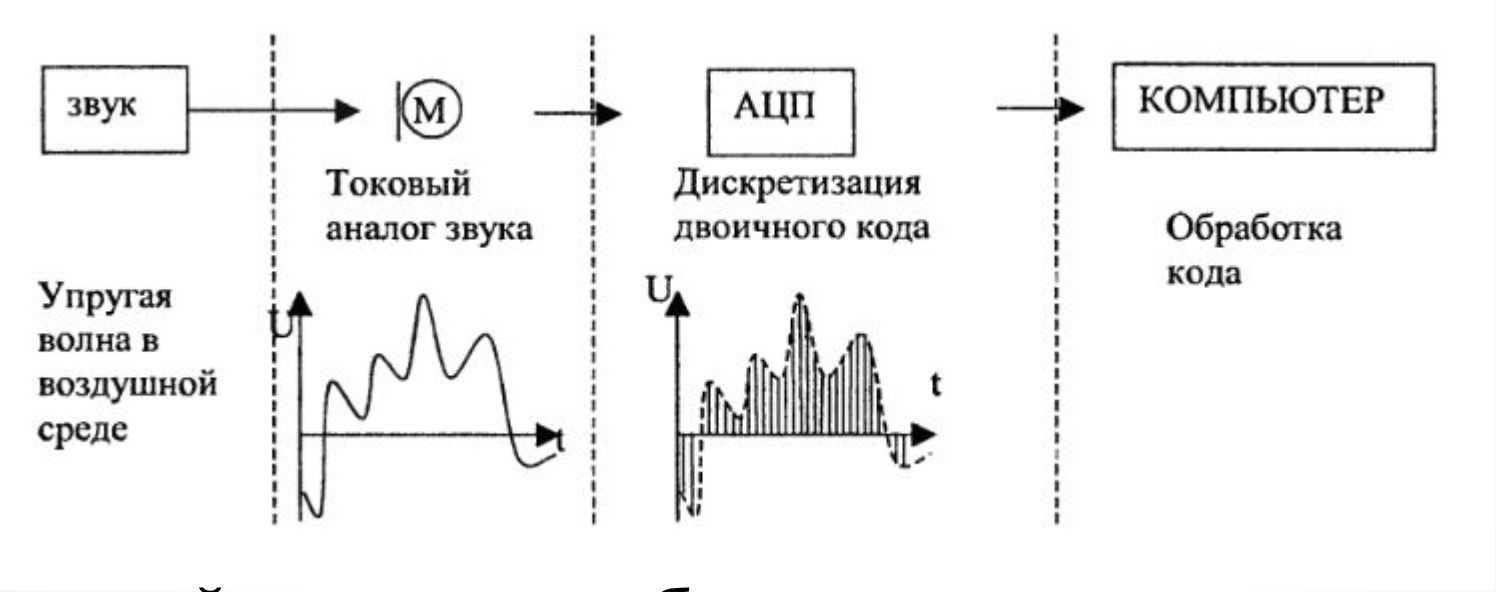
Модель HSB декла-рирована как аппарат-но-независимая.



5.4. Кодирование звуковой информации

Звук – колебания воздуха или воды, которые воспринимает человеческое ухо.

Чтобы представить звуковую информацию в виде, читаемом компьютером, необходимо выполнить следующие преобразования:

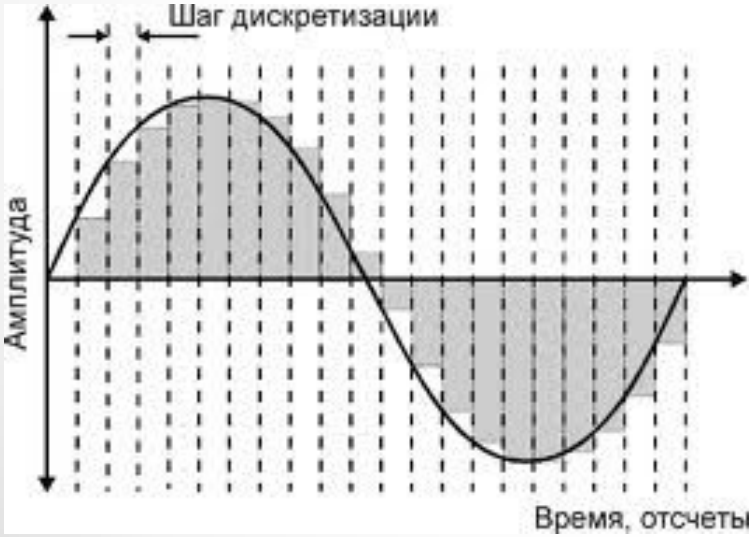


Звуковой сигнал преобразовывают в электрический аналог звука с помощью микрофона. Электрический аналог получается в непрерывной форме.

Чтобы перевести сигнал в цифровой код, надо пропустить его через **аналого-цифровой преобразователь** (АЦП). При воспроизведении происходит обратное преобразование – **цифро-аналоговое** (через ЦАП). Конструктивно АЦП и ЦАП находятся в звуковой карте компьютера. В современных ПК функции звуковой карты часто выполняет специальная микросхема материнской платы – аппаратный аудиокодек.

Оцифровка – преобразование аналогового сигнала в цифровой код. Во время этой процедуры сигнал **дискретизируется** по времени и по уровню.

Дискретизация по времени выполняется следующим образом: весь период времени T разбивается на малые интервалы времени Δt. Предполагается, что в течение интервала Δt уровень сигнала изменяется незначительно.



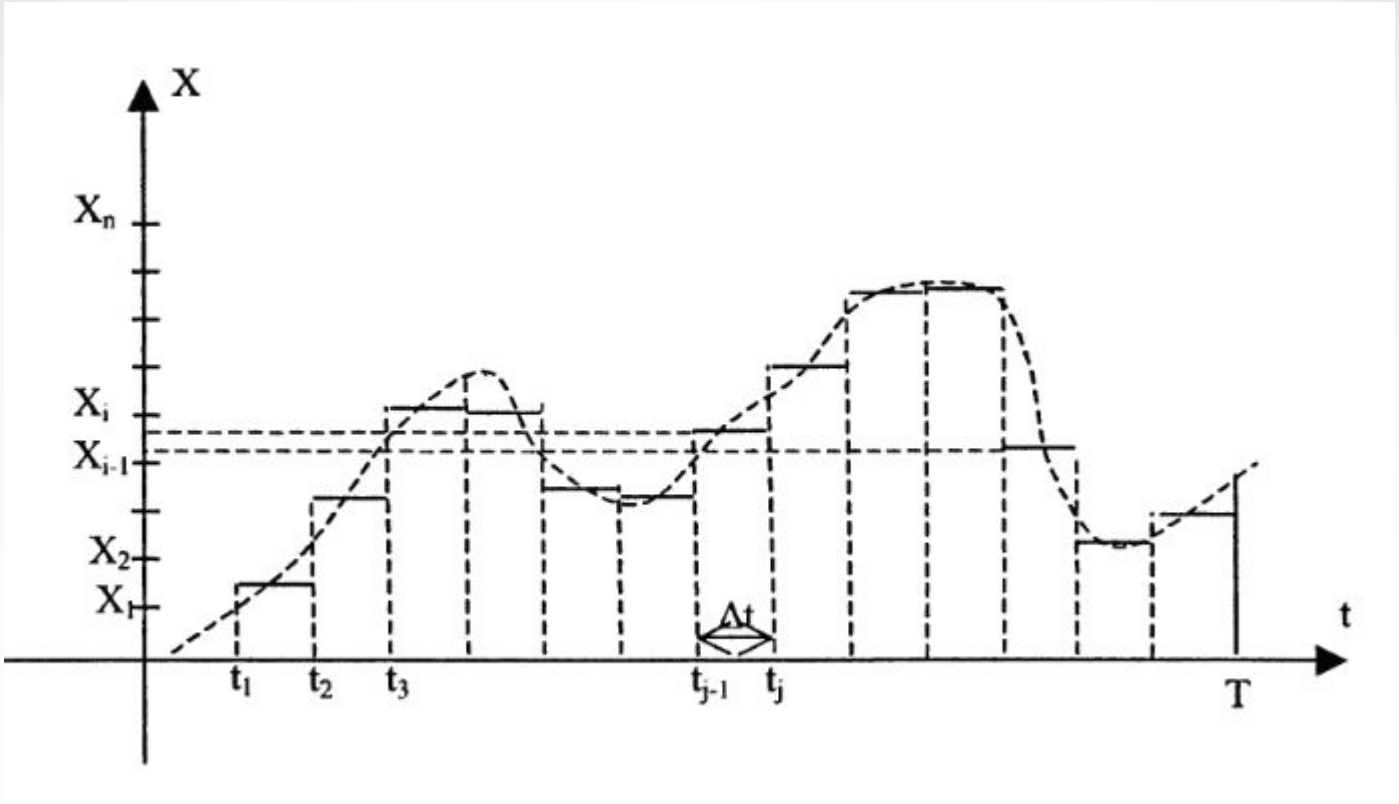
приближенно, сравнивать их (особенно если они являются результатами сложных расчетов) необходимо с большой осторожностью. Нужно проверять условие $|1 - X \times Y| < \varepsilon$, где ε — малая величина, которая задает нужную точность вычислений. Для большинства практических задач достаточно взять ε порядка 10^{-2}... 10^{-4}, а ошибка компьютерных расчетов обычно значительно меньше (не более 10^{-7}).

Для кодирования звука в компьютерах используются следующие частоты:

- 8 кГц - минимальное качество, достаточное для распознавания речи
- 11 кГц, 22 кГц, 44.1 кГц - звуковые компакт-диски
- 48 кГц – фильмы в формате DVD
- 96 кГц, 192 кГц – высококачественный звук в формате DVD-audio

Звук, проигрываемый через звуковую карту существенно отличается от исходного. Для повышения качества сигнал сглаживается с помощью специальных фильтров.

Средний **человек** слышит только звуки с частотами **от 16 Гц до 20 кГц**. Все частоты свыше 20 кГц можно «потерять», по теореме Котельникова-Шеннона, удвоив эту частоту, получаем **оптимальную частоту** дискретизации около **40 кГц** – наилучшее качество, различимое на слух. Более низкие частоты применяют, когда важно уменьшить объем звуковых данных (трансляция радиопередач через Интернет).



Дискретизация по уровню (квантование) – преобразование измеренного значения сигнала в целое число. Область изменения сигнала от самого малого значения до самого большого разбивается на N равных квантов. Каждый квант связывается с его порядковым номером, т.е. целым числом.

Разрядность кодирования – число битов, используемое для хранения одного отсчета.

- 16-18 бит - недорогие звуковые карты
- 24 бита (16 777 216 различных уровней) – большинство современных карт

Объем данных, полученный после оцифровки звука, **зависит от разрядности кодирования и частоты дискретизации**.

Пример. Используется 16-разрядное кодирование с частотой 44 кГц. За 1 сек выполняется 44 000 измерений сигнала, каждое значение занимает 16 бит.

За 1 сек: 44 000×2=88 000 байт

За 1 мин: 88 000×60≈5 Мб.

Если записывается стереозвук, то это число нужно удвоить.

Инструментальное кодирование звука

- можно применить только для кодирования инструментальных мелодий;
- основан на стандарте MIDI – **M**usical **I**nstrument **D**igital **I**nterface;
- в таком формате хранятся последовательность нот, коды инструментов, громкость, тембр, время затухания каждой ноты и т.д.;
- файл имеет значительно меньший объем по сравнению с оцифрованным звуком той же длительности;
- для проигрывания MIDI-файла используют синтезаторы (простейший – звуковая карта компьютера).

5.5. Кодирование видеоинформации

Для **сохранения видео** в памяти компьютера нужно **закодировать звук и изменяющееся изображение**, причем требуется обеспечить их **синхронность**.

Видеоинформация – трехмерный массив цветных пикселей, координатами которого являются

- разрешение кадра по горизонтали и вертикали,
- время, с которым связан отдельный кадр.

Кадр – массив значений пикселей, видимых камерой в отдельный момент времени.

768×576 точек (стандарт PAL/SECAM)

глубина цвета 24 цвета

1 минута (без звука) займет ≈1.85 Гб

Объем видео чрезвычайно большой, поэтому для его хранения и передачи выполняется **сжатие**. Некоторые незначительные детали теряются, но «обычный» человек (непрофессионал) не почувствует существенного ухудшения качества.

Основная **идея**: за короткое время изображение изменяется очень мало, поэтому можно запомнить базовый кадр, а затем сохранять только изменения.

Характеристики видеосигнала:

- **частота кадров** - число неподвижных изображений, сменяющих друг друга при показе 1 секунды видеозаписи (кинематограф – 24 кадра/с, компьютерное видео – 30 кадров/с)

- **соотношение сторон экрана** – соотношение ширины и высоты кадра

  фильмы «классического» формата 4:3

  цифровое телевидение стандартной чёткости 16:9
- **количество цветов и цветовое разрешение** описывается цветовыми моделями
- **битрейт** - количество обрабатываемых бит видеоинформации за секунду времени (бит/с)

  для DVD ≈5 Мбит/с

  формат цифрового телевидения HDTV ≈ 10 Мбит/с
- **разрешающая способность** измеряется в пикселах