

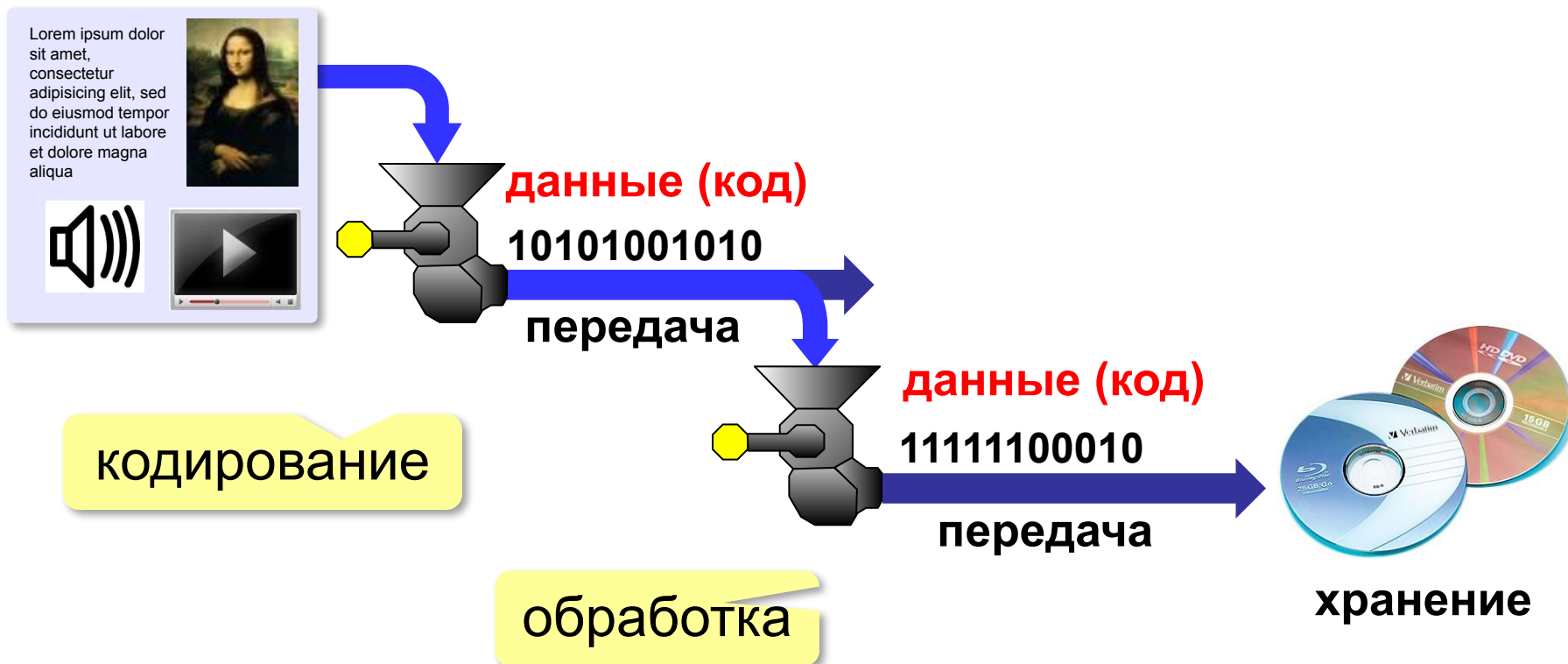
# Кодирование информации

- § 15. Кодирование символов
- § 16. Кодирование графической информации
- § 17. Кодирование звуковой и видеоинформации

# Зачем кодировать информацию?

**Кодирование** — это представление информации в форме, удобной для её хранения, передачи и обработки.

В компьютерах используется двоичный код:

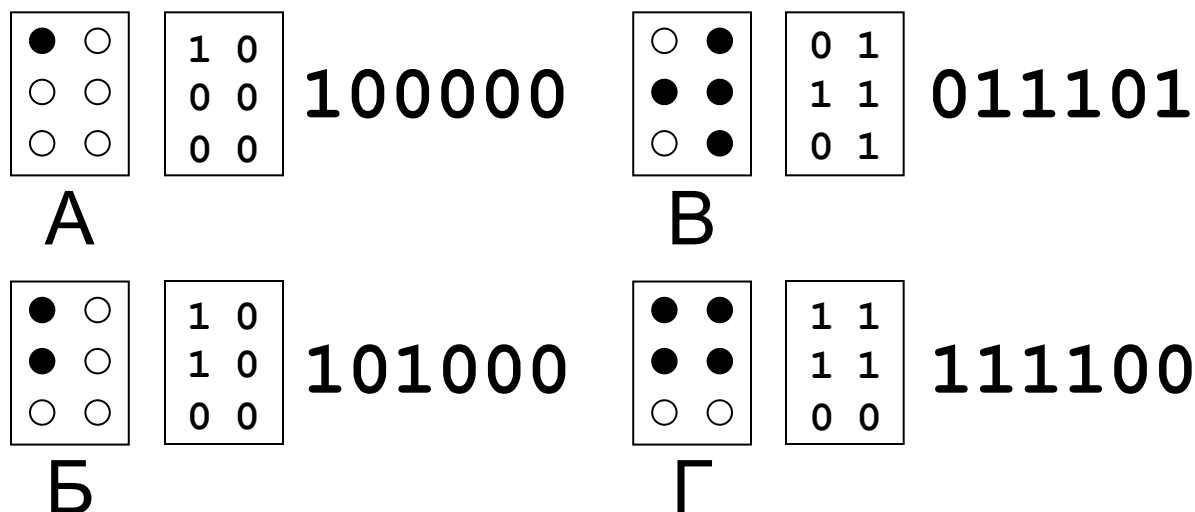


# Кодирование информации

## § 15. Кодирование символов

# Кодирование символов

Система Брайля:



**Общий подход:**

- нужно использовать  $N$  символов
- выберем число битов  $k$  на символ:  $2^k \geq N$
- сопоставим каждому символу код – число от 0 до  $2^k - 1$
- переведем коды в двоичную систему



Откуда формула?

# Кодирование символов

## Текстовый файл

- на экране (символы)
- в памяти — коды



$1000001_2$	$1000010_2$	$1000011_2$	$1000100_2$
65	66	67	68



В файле хранятся не изображения символов, а их числовые коды!

Файлы со шрифтами: **\*.fon**, **\*.ttf**, **\*.otf**

# Кодировка ASCII (7-битная)

---

**ASCII** = *American Standard Code for Information Interchange*

**Коды 0-127:**

**0-31 управляющие символы:**

7 – звонок, 10 – новая строка,

13 – возврат каретки, 27 – Esc.

32 пробел

**знаки препинания:** . , : ; ! ?

**специальные знаки:** + - \* / ( ) { } [ ]

48-57 цифры **0..9**

65-90 заглавные латинские буквы **A-Z**

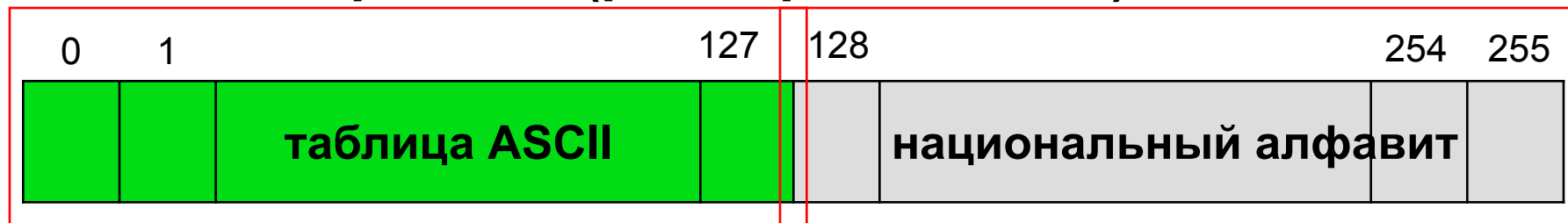
97-122 строчные латинские буквы **a-z**



Где русские буквы?

# 8-битные кодировки

## Кодовые страницы (расширения ASCII):



Для русского языка:

**CP-866** для *MS DOS*

**CP-1251** для *Windows* (Интернет)

**KOI8-R** для *UNIX* (Интернет)

**MacCyrillic** для компьютеров *Apple*

Проблема:

*Windows-1251*

Привет, Вася!

рТЙЧЕФ,

чБУС!

*KOI8-R*



оПХБЕР,

бЮЯЪ!

Привет, Вася!

## 8-БИТНЫЕ КОДИРОВКИ

---

-  1 байт на символ – файлы небольшого размера!
  - просто обрабатывать в программах
-  нельзя использовать символы разных кодовых страниц одновременно (русские и французские буквы, и т.п.)
  - неясно, в какой кодировке текст (перебор вариантов!)
  - для каждой кодировки нужен свой шрифт (изображения символов)



# Стандарт UNICODE

---

**1 112 064** знаков, используются около **100 000**

*Windows:* **UTF-16**

16 битов на распространённые символы,  
32 бита на редко встречающиеся

*Linux:* **UTF-8**

8 битов на символ для ASCII,  
от 16 до 48 бита на остальные



- совместимость с ASCII
- более экономична, чем UTF-16, если много символов ASCII

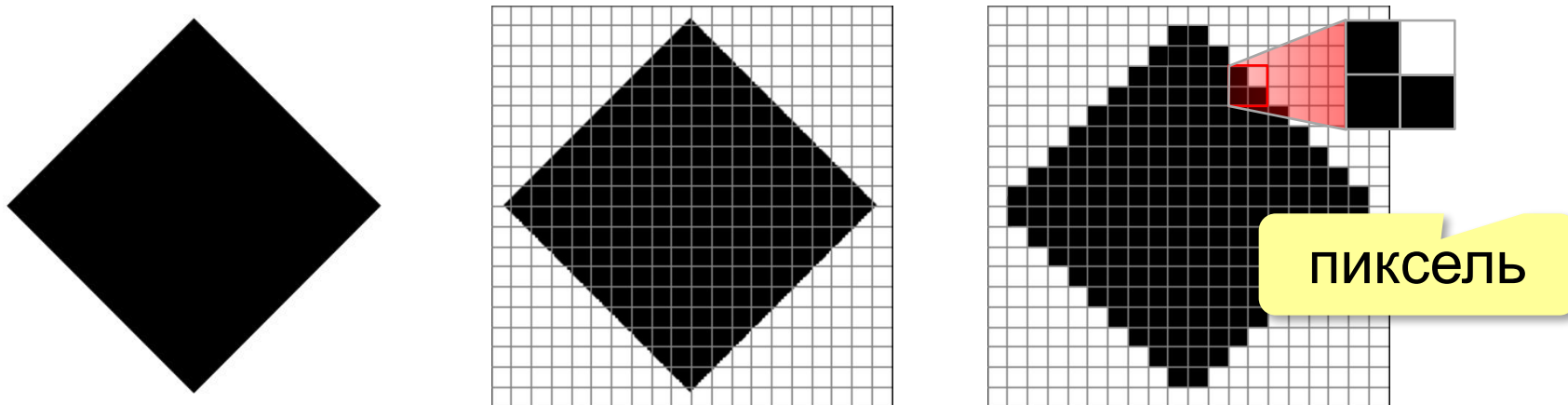


2010 г. – 50% сайтов использовали UTF-8!

# Кодирование информации

## § 16. Кодирование графической информации

# Растровое кодирование



дискретизация

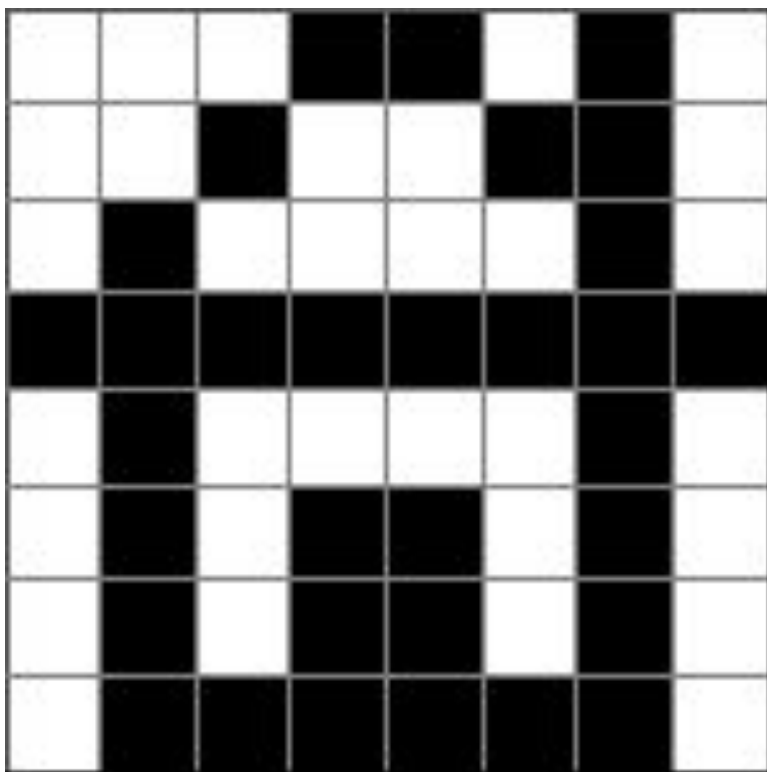


Рисунок искажается!

**Пиксель** – это наименьший элемент рисунка, для которого можно задать свой цвет.

**Растровое изображение** – это изображение, которое кодируется как множество пикселей.

# Растровое кодирование



0	0	0	1	1	0	1	0	1A
0	0	1	0	0	1	1	0	26
0	1	0	0	0	0	1	0	42
1	1	1	1	1	1	1	1	FF
0	1	0	0	0	0	1	0	42
0	1	0	1	1	0	1	0	5A
0	1	0	1	1	0	1	0	5A
0	1	1	1	1	1	1	0	7E

**1A2642FF425A5A7E**<sub>16</sub>

# Разрешение

**Разрешение** – это количество пикселей, приходящихся на дюйм размера изображения.

*ppi* = *pixels per inch*, пикселей на дюйм

1 дюйм = 2,54 см



300 ppi

печать

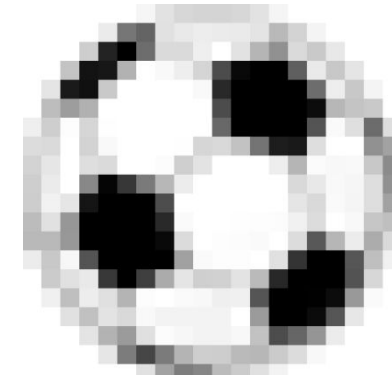


96 ppi

экран



48 ppi



24 ppi

# Разрешение

---

**Задача 1.** Какой размер в пикселях должен иметь закодированный рисунок с разрешением **300 ppi**, чтобы с него можно было сделать отпечаток размером **10×15 см**?

$$\text{высота} \quad \frac{10 \text{ см} \times 300 \text{ пикселей}}{2,54 \text{ см}} \approx \mathbf{1181 \text{ пиксель}}$$

$$\text{ширина} \quad \frac{15 \text{ см} \times 300 \text{ пикселей}}{2,54 \text{ см}} \approx \mathbf{1771 \text{ пиксель}}$$

## Разрешение

---

**Задача 2.** Закодированный рисунок имеет размеры  $5760 \times 3840$  пикселей и разрешение  $600 \text{ ppi}$ . Какой размер будет у изображения, отпечатанного на принтере?

$$\text{ширина} \quad \frac{5760 \text{ пикселей} \times 2,54 \text{ см}}{600 \text{ пикселей}} \approx \mathbf{24,4 \text{ см}}$$

$$\text{высота} \quad \frac{3840 \text{ пикселей} \times 2,54 \text{ см}}{600 \text{ пикселей}} \approx \mathbf{16,3 \text{ см}}$$

# Цветовая модель RGB

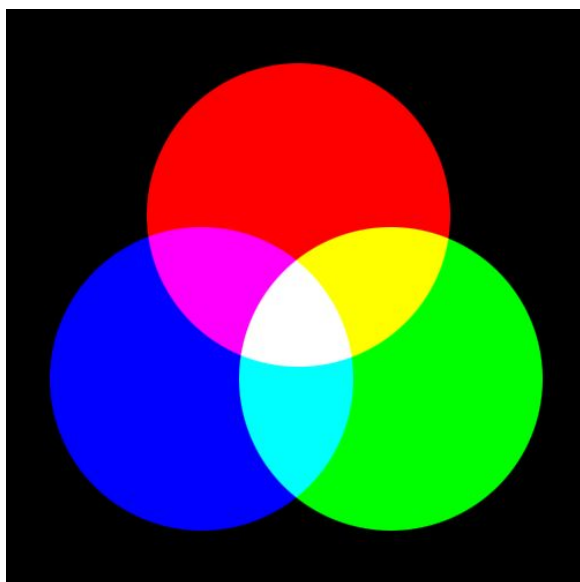
Д. Максвелл, 1860

цвет = ( **R**, **G**, **B** )

*red*      *green*      *blue*

красный    зеленый    синий

0..255      0..255      0..255



■ (0, 0, 0)

□ (255, 255, 255)

■ (255, 0, 0)

■ (255, 150, 150)

■ (0, 255, 0)

■ (0, 255, 255)

■ (0, 0, 255)

■ (100, 0, 0)



Сколько разных цветов можно кодировать?

$256 \cdot 256 \cdot 256 = 16\ 777\ 216$  (*True Color*, «истинный цвет»)

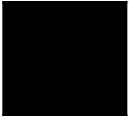








RGB – цветовая модель для устройств, излучающих свет (мониторов)!



# Цветовая модель RGB

(**255**, **255**, **0**) → #**FFFF00**

	RGB	Веб-страница
	(0, 0, 0)	#000000
	(255,255,255)	#FFFFFF
	(255, 0, 0)	#FF0000
	(0, 255, 0)	#00FF00
	(0, 0, 255)	#0000FF
	(255, 255, 0)	#FFFF00
	(204,204,204)	#CCCCCC

# Глубина цвета

**Глубина цвета** — это количество битов, используемое для кодирования цвета пикселя.



Сколько памяти нужно для хранения цвета 1 пикселя в режиме *True Color*?

**R** (0..255) 256 =  $2^8$  вариантов 8 битов = 1 байт

**R G B**: 24 бита = 3 байта

*True Color*  
(ИСТИННЫЙ ЦВЕТ)

**Задача.** Определите размер файла, в котором закодирован растровый рисунок размером **20×30 пикселей** в режиме истинного цвета (*True Color*)?

$20 \cdot 30 \cdot 3 \text{ байта} = \mathbf{1800}$

**байт**

# Кодирование с палитрой

---



Как уменьшить размер файла?

- уменьшить разрешение
- уменьшить глубину цвета

снижается  
качество

**Цветовая палитра** – это таблица, в которой каждому цвету, заданному в виде составляющих в модели RGB, сопоставляется числовой код.

# Задачи

---

















**Задача 1.** В процессе преобразования растрового графического файла количество цветов уменьшилось с 512 до 8. Во сколько раз уменьшился информационный объем файла?

**Задача 2.** Разрешение экрана монитора – 1024 x 768 точек, глубина цвета – 16 бит. Каков необходимый объем видеопамяти для данного графического режима?

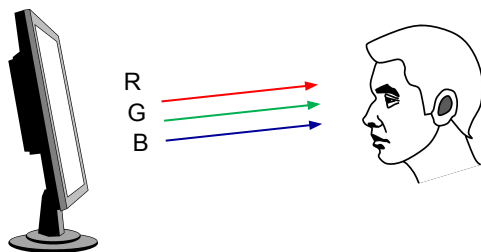
**Задача 3.** Сколько байт будет занимать код рисунка размером 40×50 пикселей в режиме истинного цвета? при кодировании с палитрой 256 цветов?

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

# Растровые рисунки: форматы файлов

Формат	True Color	Палитра	Прозрачность	Анимация
<b>BMP</b>				
<b>JPG</b>				
<b>GIF</b>				
<b>PNG</b>				

# Кодирование цвета при печати (СМУК)



Белый – красный

= голубой

**C = Cyan**

Белый – зелёный

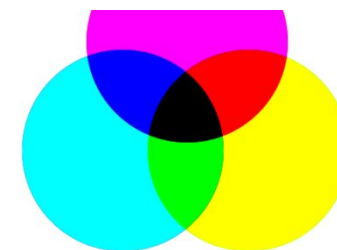
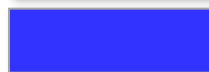
= пурпурный

**M = Magenta**

Белый – синий

= желтый

**Y = Yellow**



Модель СМУ

Модель СМУК: + **Key color**



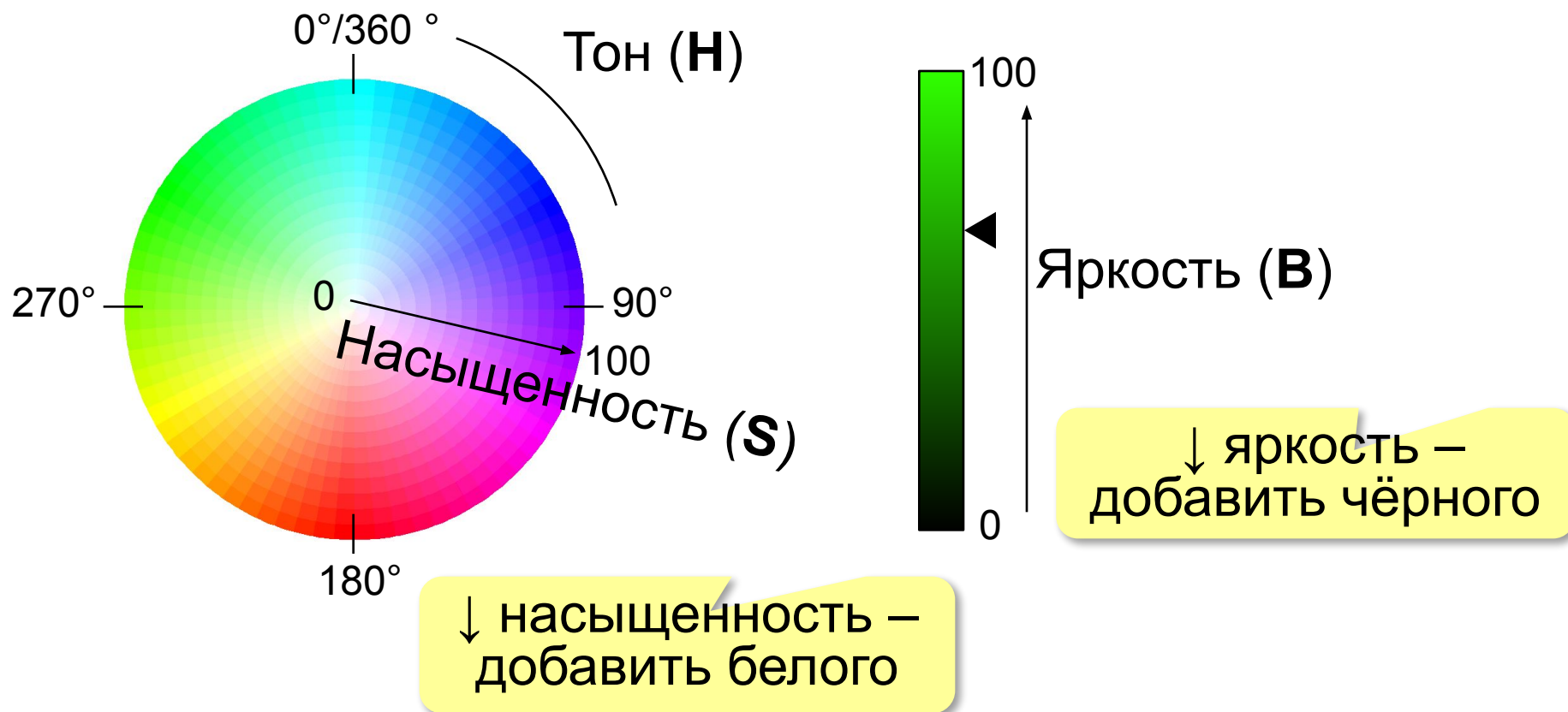
- меньший расход краски и лучшее качество для чёрного и серого цветов

# Цветовая модель HSB (HSV)

**HSB** = *Hue* (тон, оттенок)

*Saturation* (насыщенность)

*Brightness* (яркость) или *Value* (величина)



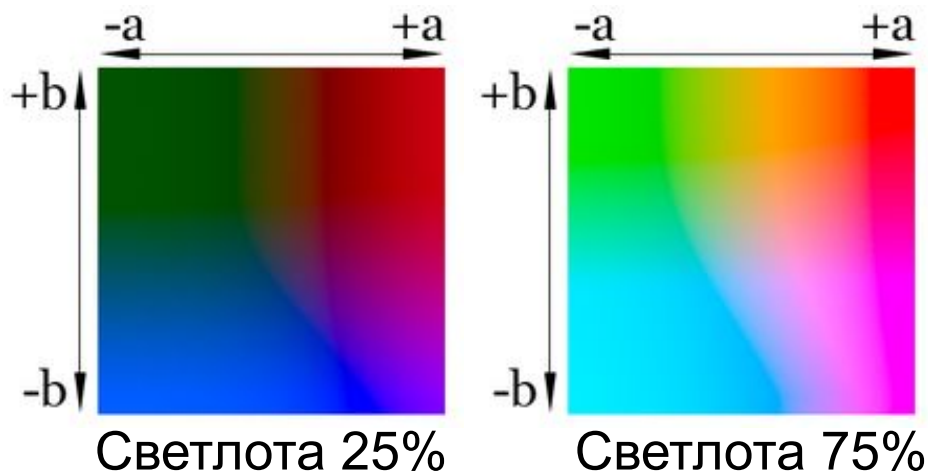
# Цветовая модель Lab

Международный стандарт кодирования цвета, независимого от устройства (1976 г.)

Основана на модели восприятия цвета человеком.

**Lab** = *Lightness* (светлота)

**a, b** (задают цветовой тон)



- для перевода между цветовыми моделями: RGB → Lab → CMYK
- для цветокоррекции фотографий



# Профили устройств

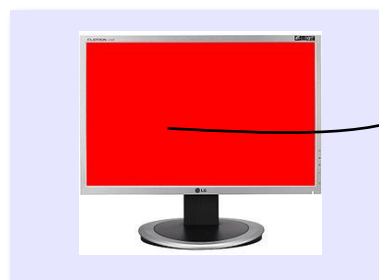


Какой цвет увидим?

**RGB(255,0,0)**



**RGB(255,0,0)**



как  $\lambda \approx 680\text{nm}$

профиль  
монитора

$\lambda \approx 680\text{nm}$



**RGB(225,10,20)**

профиль  
сканера



**CMYK(0,100,100,0)**



профиль  
принтера

## Растровое кодирование: итоги

---

-  универсальный метод (можно закодировать любое изображение)
- единственный метод для кодирования и обработки размытых изображений, не имеющих чётких границ (фотографий)
-  есть **потеря информации** (почему?)
- при изменении размеров цвет и форма объектов на рисунке **искажаются**
- **размер файла** не зависит от сложности рисунка (а от чего зависит?)

# Векторное кодирование

---

## Рисунки из геометрических фигур:

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

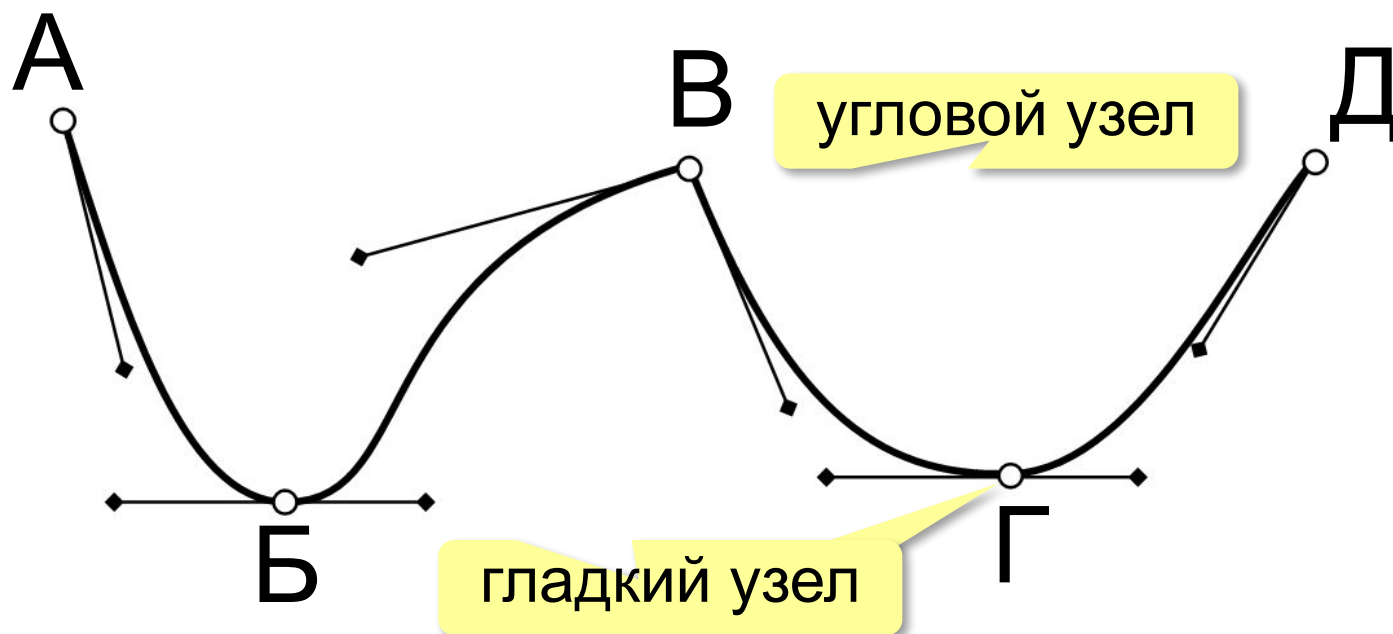
## Для каждой фигуры в памяти хранятся:

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



# Векторное кодирование

## Кривые Безье:



Хранятся координаты узлов и концов «рычагов»  
(3 точки для каждого узла, кривые 3-го порядка).

# Векторное кодирование (итоги)

---



- лучший способ для хранения **чертежей, схем, карт**
- при кодировании **нет потери информации**
- при изменении размера **нет искажений**



растровый  
рисунок



векторный  
рисунок

- меньше **размер файла**, зависит от сложности рисунка



- неэффективно использовать для **фотографий** и **размытых изображений**

# Векторное кодирование: форматы файлов

---

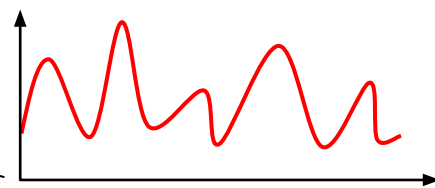
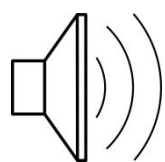
- **WMF** (*Windows Metafile*)
- **EMF** (*Windows Metafile*)
- **CDR** (программа *CorelDraw*)
- **AI** (программа *Adobe Illustrator*)
- **SVG** (*Scalable Vector Graphics*, масштабируемые векторные изображения)

для веб-страниц

# Кодирование информации

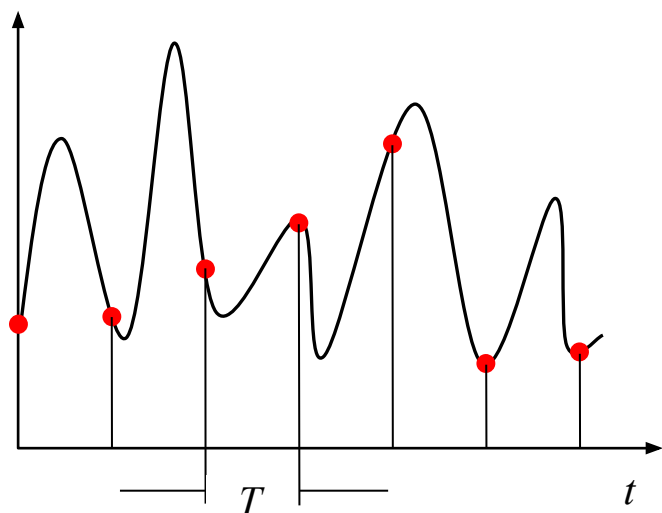
## § 17. Кодирование звуковой и видеоинформации

# Оцифровка звука



аналоговый  
сигнал

**Оцифровка** – это преобразование аналогового сигнала в цифровой код (дискретизация).



$T$  – интервал дискретизации (с)  
 $f = \frac{1}{T}$  – частота дискретизации (Гц, кГц)

8 кГц – минимальная частота для распознавания речи

11 кГц, 22 кГц,

44,1 кГц – качество CD-дисков

48 кГц – фильмы на DVD

96 кГц, 192 кГц

**Человек слышит**

**16 Гц ... 20 кГц**



# Оцифровка звука: квантование

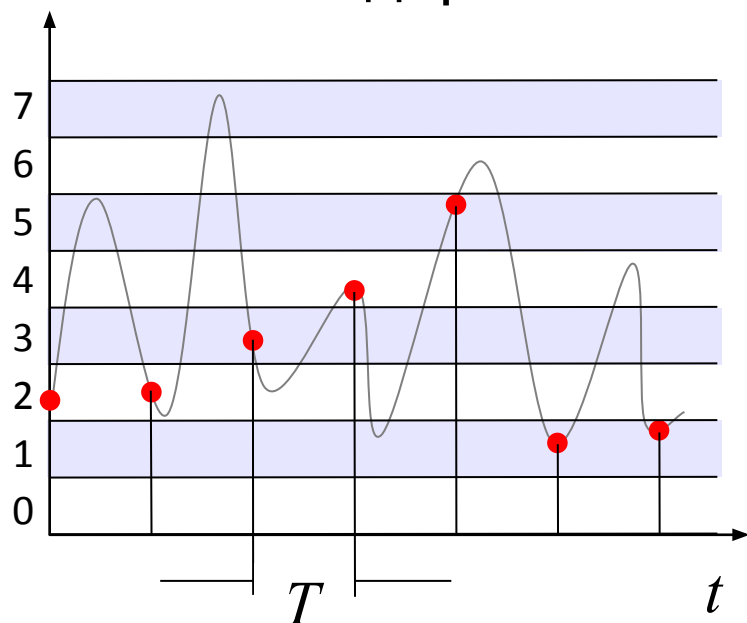


Сколько битов нужно, чтобы записать число 0,6?

**Квантование** (дискретизация по уровню) – это представление числа в виде цифрового кода конечной длины.

**АЦП** = Аналого-Цифровой Преобразователь

3-битное кодирование:



8 битов = 256 уровней

16 битов = 65536 уровней

24 бита =  $2^{24}$  уровней

**Разрядность кодирования** — это число битов, используемое для хранения одного отсчёта.

# Оцифровка звука

---

**Задача.** Определите информационный объем данных, полученных при оцифровке звука длительностью **1 минута** с частотой **44 кГц** с помощью **16-битной** звуковой карты. Запись выполнена в режиме «стерео».

За 1 сек *каждый канал* записывает **44000** значений,  
каждое занимает **16 битов = 2 байта**  
всего  **$44000 \cdot 2$  байта = 88000 байтов**

С учётом «стерео»

всего  **$88000 \cdot 2 = 176000$  байтов**

За 1 минуту

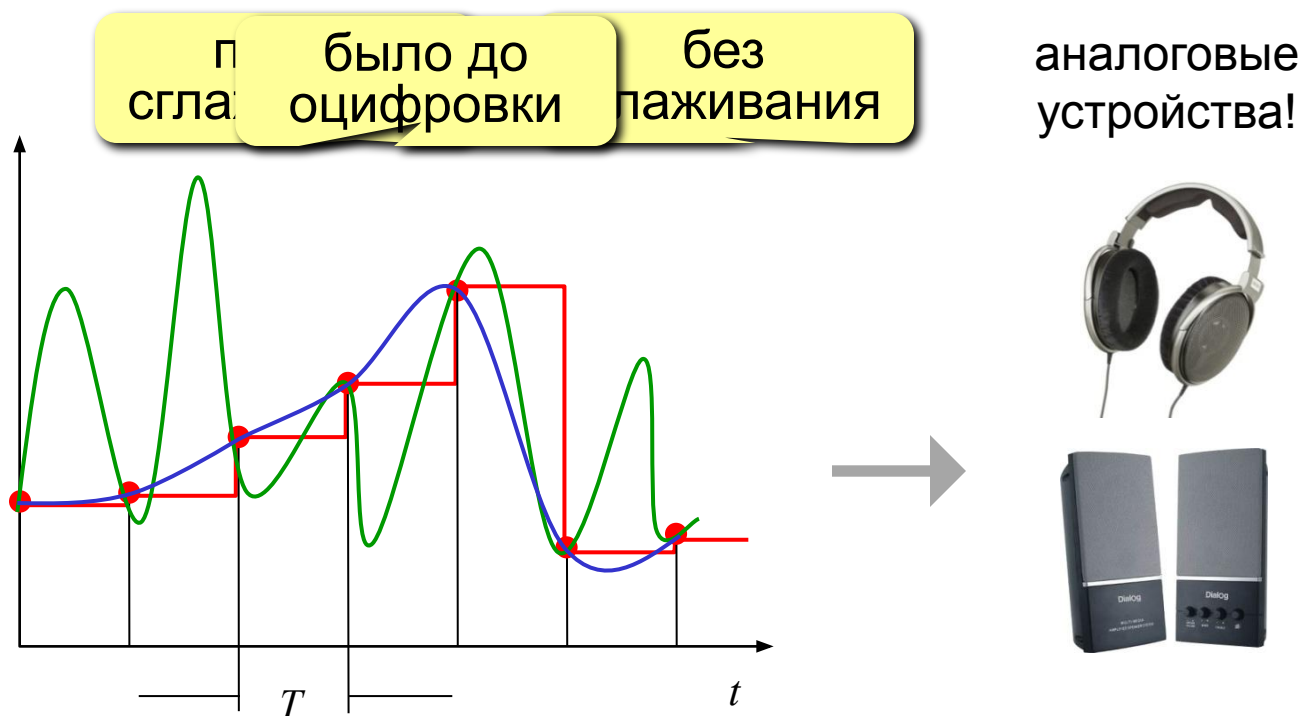
**$176000 \cdot 60 = 1056000$  байтов**

**$\approx 10313$  Кбайт  $\approx 10$  Мбайт**

# Оцифровка звука

Как восстановить сигнал?

**ЦАП** = Цифро-Аналоговый Преобразователь



**?** Какой улучшить качество?

уменьшать  $T$

**?** Что при этом ухудшится?


↑ размер файла

# Оцифровка – итог

---

 можно закодировать **любой звук** (в т.ч. ГОЛОС, СВИСТ, шорох, ...)

 • есть **потеря информации**  
• большой **объем файлов**

 Какие свойства оцифрованного звука определяют качество звучания?

## Форматы файлов:

**WAV** (*Waveform audio format*), часто без сжатия (размер!)

**MP3** (*MPEG-1 Audio Layer 3*, сжатие с учётом восприятия человеком)

**AAC** (*Advanced Audio Coding*, 48 каналов, сжатие)

**WMA** (*Windows Media Audio*, потоковый звук, сжатие)

**OGG** (*Ogg Vorbis*, открытый формат, сжатие)

# Инструментальное кодирование

**MIDI** (*Musical Instrument Digital Interface* — цифровой интерфейс музыкальных инструментов).

в файле **.mid**:

- нота (высота, длительность)
- музыкальный инструмент
- параметры звука (громкость, тембр)
- до 1024 каналов

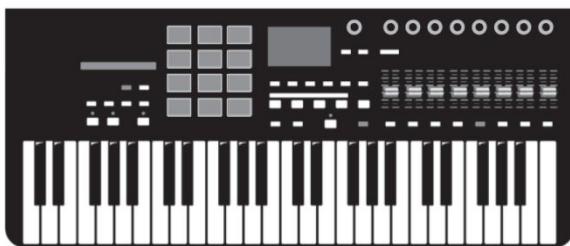
128 мелодических  
и 47 ударных

программа для  
звуковой карты!

в памяти звуковой карты:

- образцы звуков (волновые таблицы)

**MIDI-клавиатура:**



- **нет потери информации** при кодировании инструментальной музыки
- **небольшой размер файлов**



невозможно закодировать нестандартный звук, голос

# Трекерная музыка

---

В файле (модуле):

- образцы звуков (*сэмплы*)
- нотная запись, трек (*track*) – дорожка
- музыкальный инструмент
- до 32 каналов

## Форматы файлов:

**MOD** разработан для компьютеров *Amiga*

**S3M** оцифрованные каналы + синтезированный звук, 99 инструментов

**XM, STM, ...**

**Использование:** демосцены (важен размер файла)

# Кодирование видео



Видео = изображения + звук Синхронность!

изображен

- $\geq 25$  кадр
- **PAL:** 768  
за 1  
за 1
- **HDTV:** 1
- **ИСХОДНЫ**
- **сжатие (**
- **DivX, Xv**

звук:

- 48 кГц, 1
- **сжатие (**
- **MP3, AA**



# Форматы видеофайлов

---

- AVI** – *Audio Video Interleave* – чередующиеся звук и видео; контейнер – могут использоваться разные кодеки
- MPEG** – *Motion Picture Expert Group*
- WMV** – *Windows Media Video*, формат фирмы *Microsoft*
- MP4** – *MPEG-4*, сжатое видео и звук
- MOV** – *Quick Time Movie*, формат фирмы *Apple*
- WebM** – открытый формат, поддерживается браузерами