

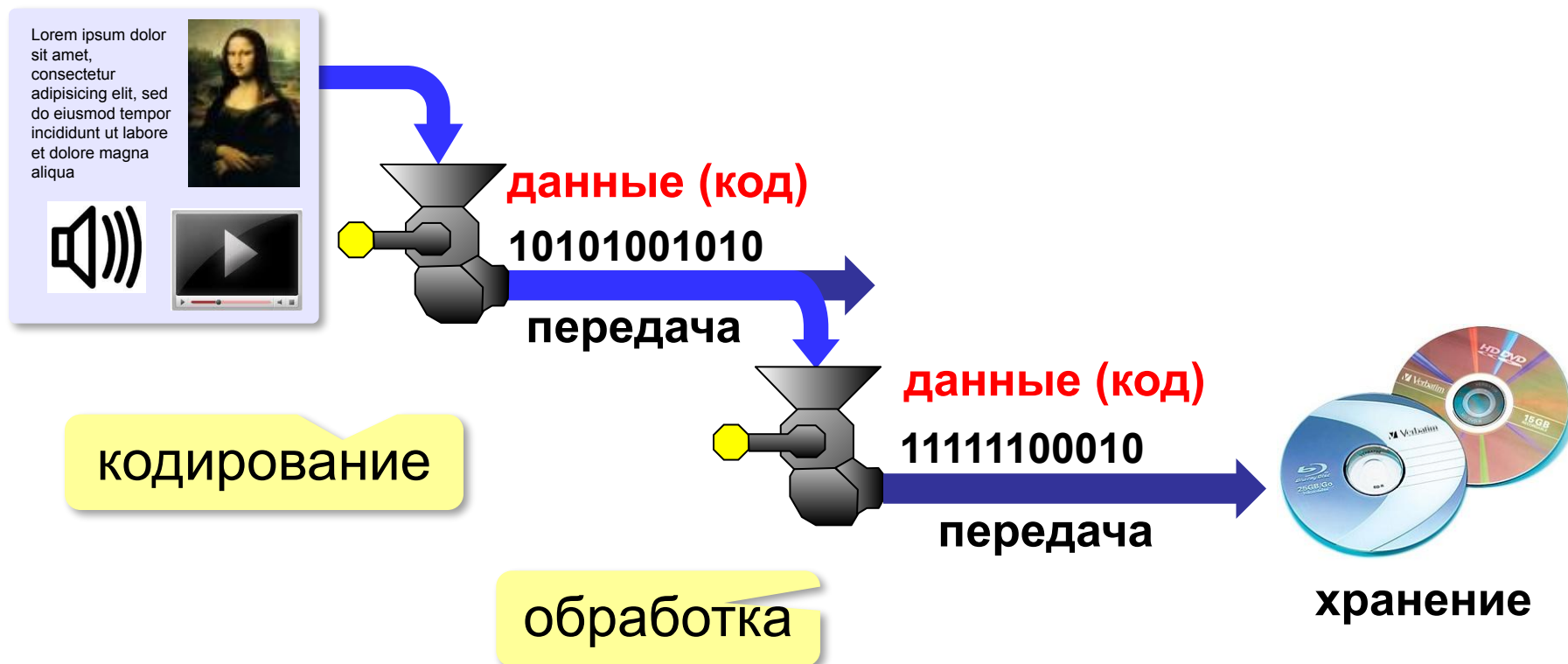
Кодирование информации

- § 13. Кодирование текстов
- § 14. Кодирование графической информации
- § 15. 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:

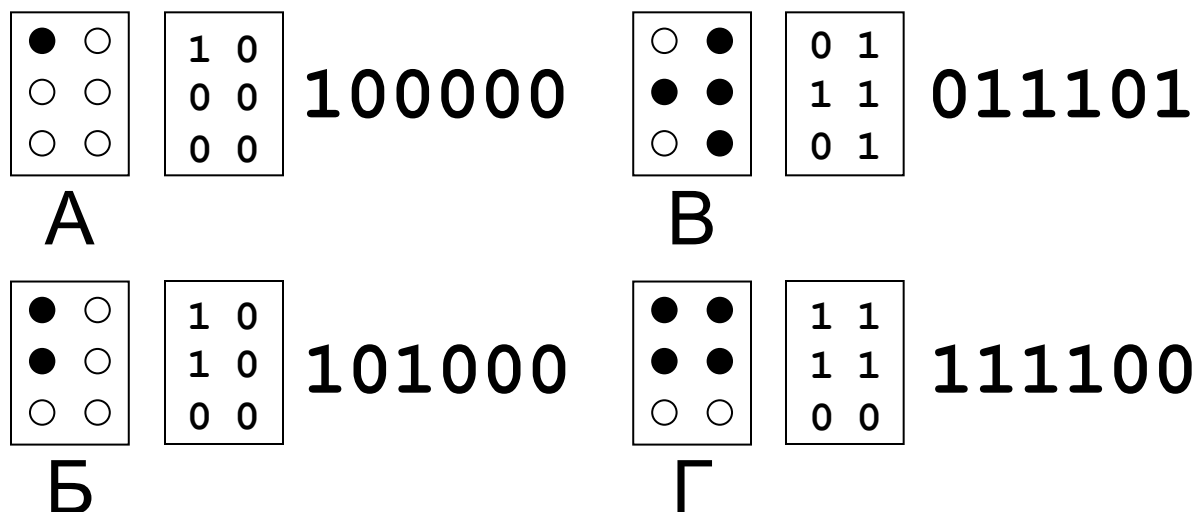


Кодирование информации

§ 13. Кодирование текстов

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**

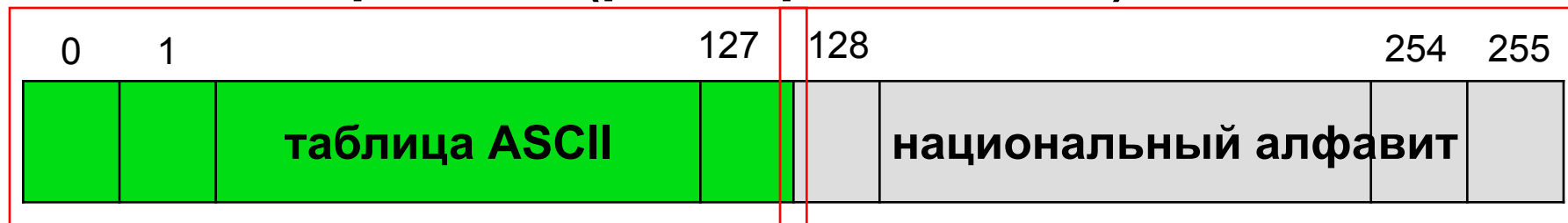
97-122 строчные латинские буквы **a-z**



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

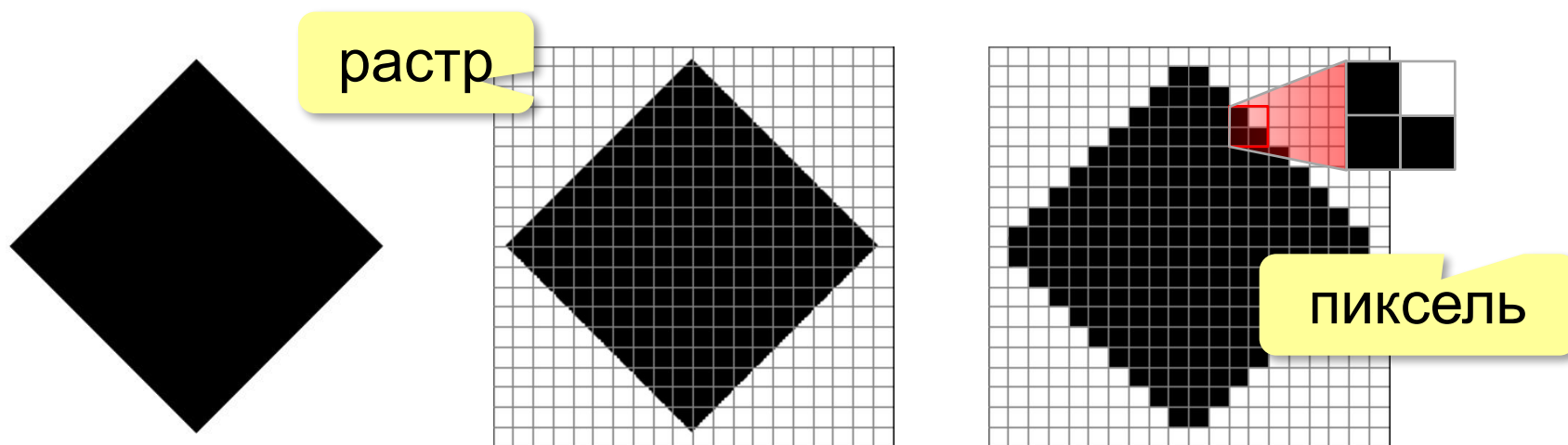


2010 г. – 50% сайтов использовали UTF-8!

Кодирование информации

§ 14. Кодирование графической информации

Растровое кодирование



дискретизация

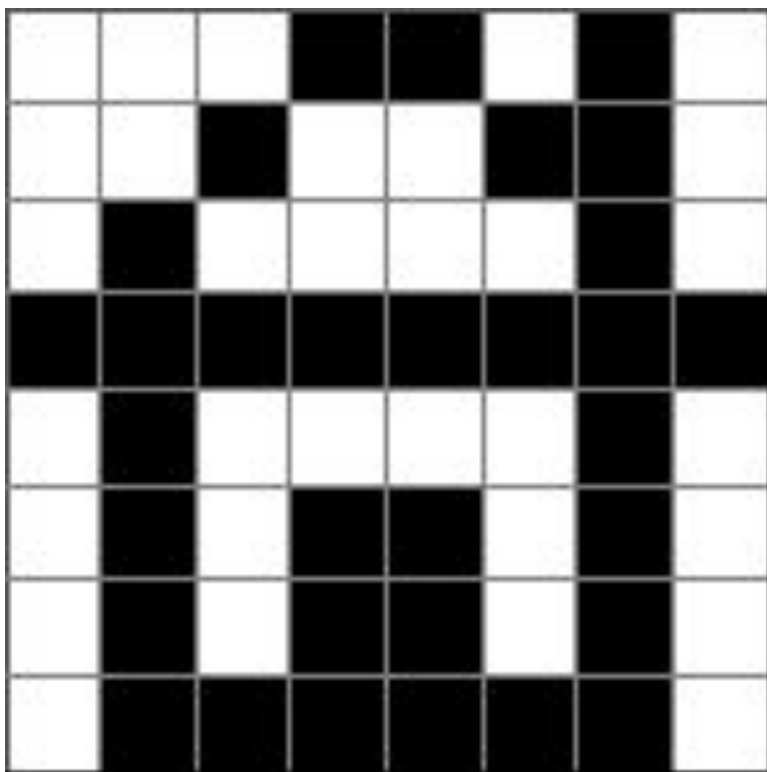


Рисунок искажается!

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

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

Растровое кодирование

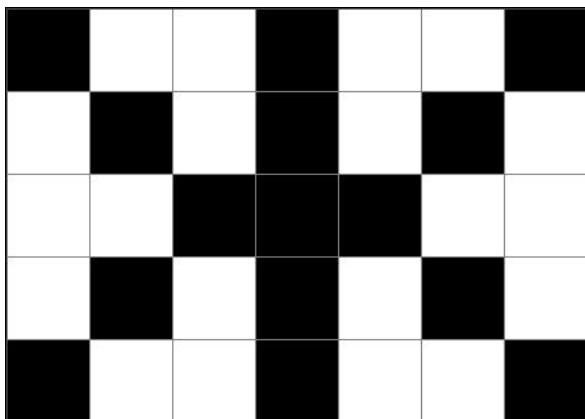


0	0	0	1	1	0	1	0	1A
0	0	1	0	0	1	1	0	26
0	1	0	0	0	0	1	0	42
1	1	1	1	1	1	1	1	FF
0	1	0	0	0	0	1	0	42
0	1	0	1	1	0	1	0	5A
0	1	0	1	1	0	1	0	5A
0	1	1	1	1	1	1	0	7E

1A2642FF425A5A7E₁₆

Задача

Закодируйте рисунок с помощью шестнадцатеричного кода:



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

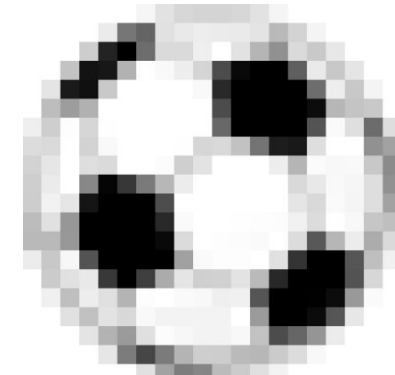


96 ppi

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**?

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

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

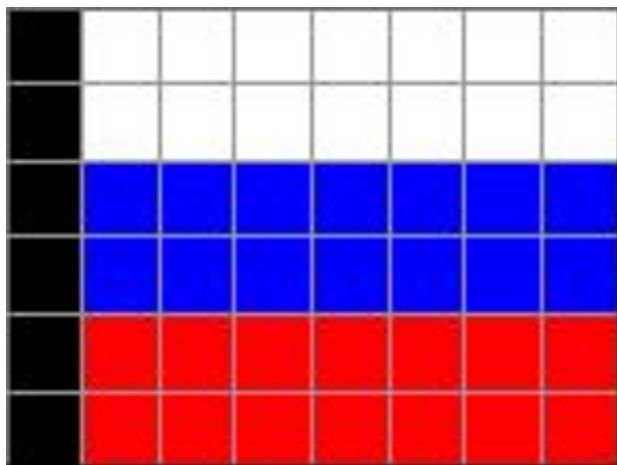
Разрешение

Задача 2. Закодированный рисунок имеет размеры 5760×3840 пикселей и разрешение 600 ppi . Какой размер будет у изображения, отпечатанного на принтере?

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

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

Кодирование цвета



00	11	11	11	11	11	11	11
00	11	11	11	11	11	11	11
00	01	01	01	01	01	01	01
00	01	01	01	01	01	01	01
00	10	10	10	10	10	10	10
00	10	10	10	10	10	10	10



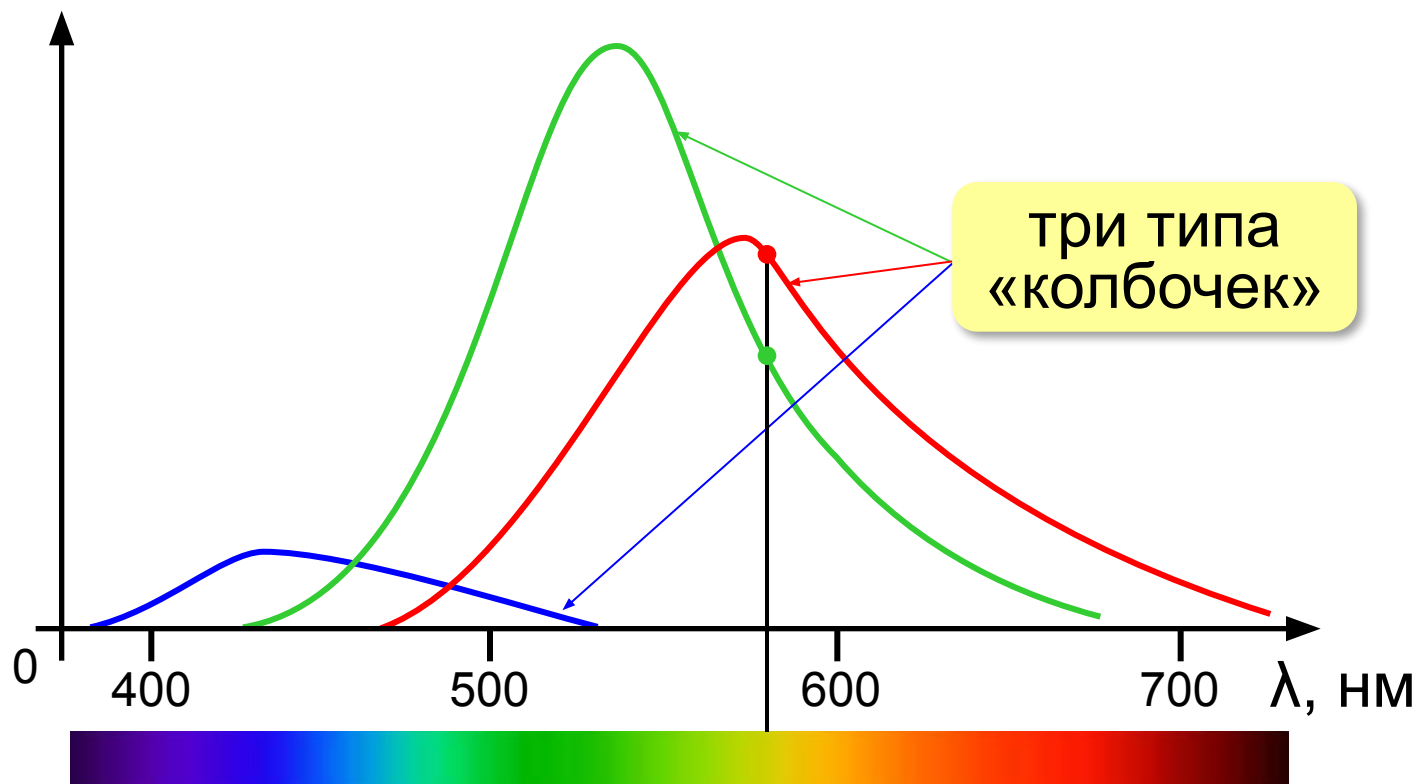
Как выводить на монитор цвет с кодом 00?



Как закодировать цвет в виде чисел?

Теория цвета Юнга-Гельмгольца

чувствительность



Свет любой длины волны можно заменить на красный, зелёный и синий лучи!

Цветовая модель RGB

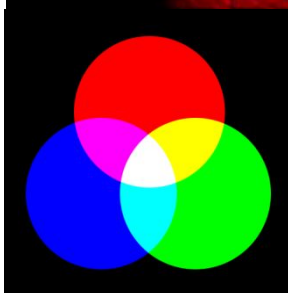
Д. Максвелл, 1860

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)

■ (255, 255, 0)

■ (0, 0, 255)

■ (100, 0, 0)



Сколько разных цветов можно кодировать?

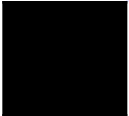






$256 \cdot 256 \cdot 256 = 16\ 777\ 216$ (*True Color*, «истинный цвет»)



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

Цветовая модель RGB

(**255**, **255**, **0**) → #**FF****FF****00**

	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

Задачи

Постройте шестнадцатеричные коды:

RGB (100, 200, 200) →

RGB (30, 50, 200) →

RGB (60, 180, 20) →

RGB (220, 150, 30) →

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



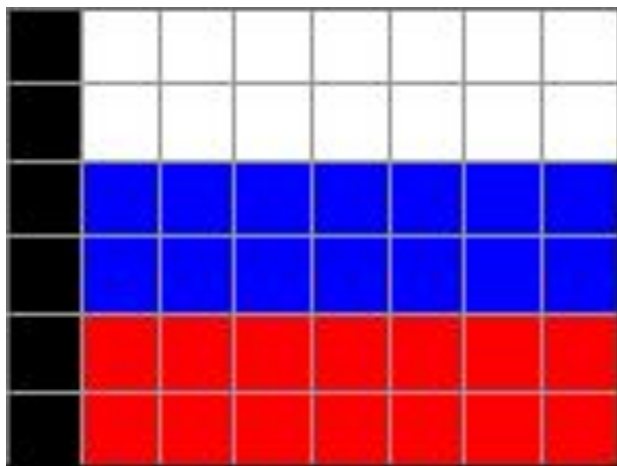
Как уменьшить размер файла?

- 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.

Кодирование с палитрой



00	11	11	11	11	11	11	11
00	11	11	11	11	11	11	11
00	01	01	01	01	01	01	01
00	01	01	01	01	01	01	01
00	10	10	10	10	10	10	10
00	10	10	10	10	10	10	10

Палитра:

0	0	0	0	0	255	255	0	0	255	255	255
цвет 00_2			цвет 01_2			цвет 10_2			цвет 11_2		



Какая глубина цвета?

2 бита на пиксель



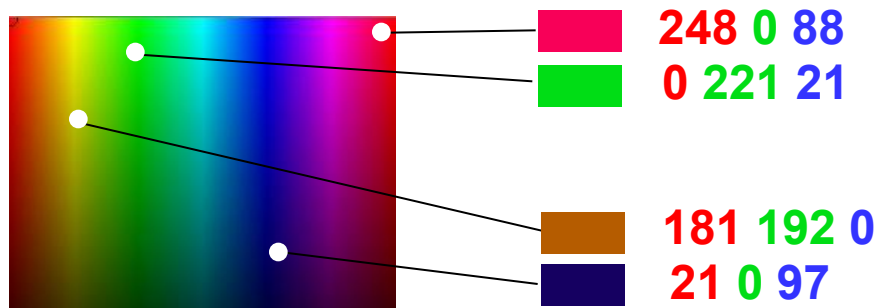
Сколько занимает палитра?

$3 \cdot 4 = 12$
байтов

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

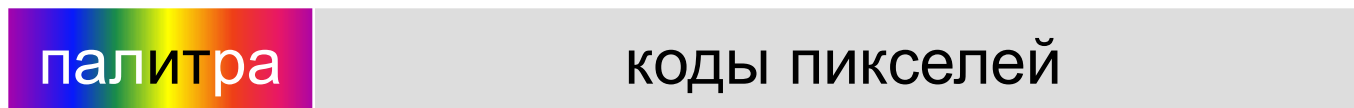
0	1	45	254	255
248 0 88	0 221 21	148 35 115	181 192 0	21 0 97

Шаг 4. Код пикселя = номеру его цвета в палитре

1	254	45	14	...	12	23
---	-----	----	----	-----	----	----

Кодирование с палитрой

Файл с палитрой:



Количество цветов	Размер палитры (байтов)	Глубина цвета (битов на пиксель)
2	6	1
4	12	2
16	48	4
256	768	8

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?

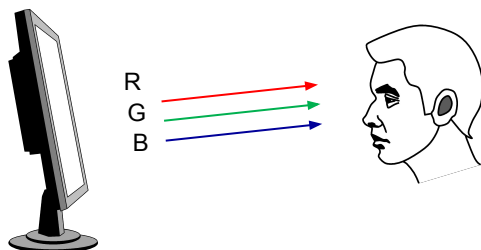
Задача 3. Сколько байт будет занимать код рисунка размером 40×50 пикселей в режиме истинного цвета? при кодировании с палитрой 256 цветов?

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
BMP				
JPG				
GIF				
PNG				

Кодирование цвета при печати (СМУК)



Белый – красный

= голубой

C = Cyan

Белый – зелёный

= пурпурный

M = Magenta

Белый – синий

= желтый

Y = Yellow

C	M	Y
---	---	---

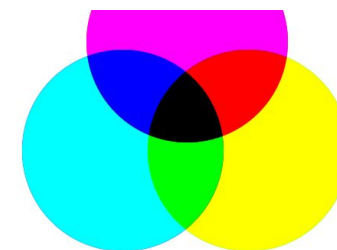
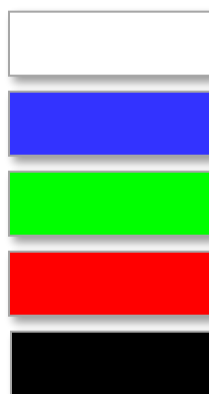
0	0	0
---	---	---

255	255	0
-----	-----	---

255	0	255
-----	---	-----

0	255	255
---	-----	-----

255	255	255
-----	-----	-----



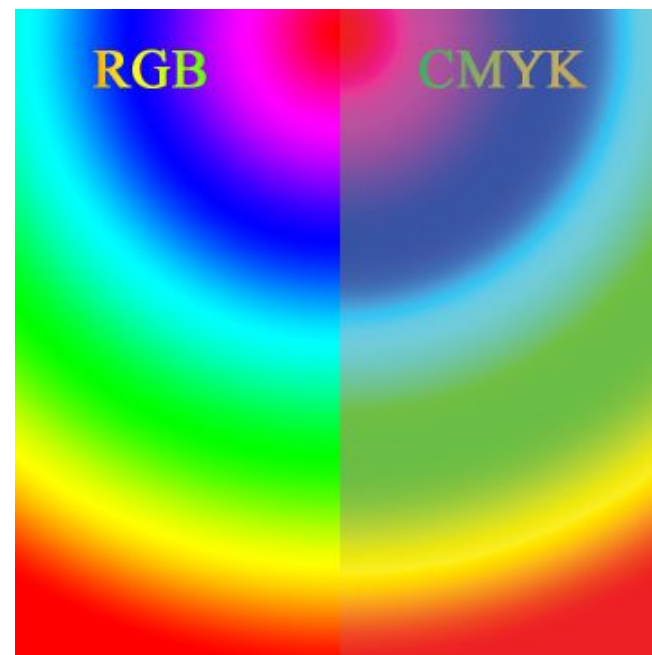
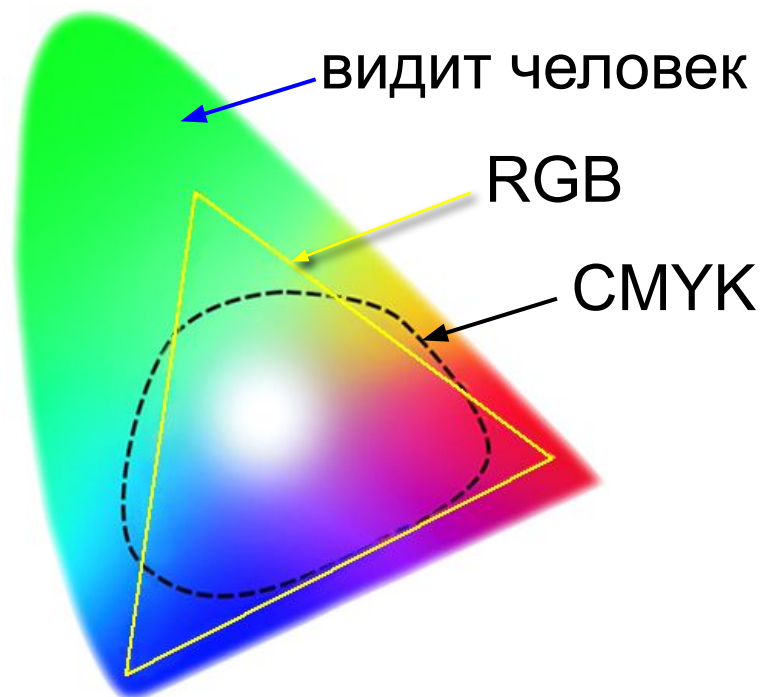
Модель CMY

Модель CMYK: + **Key color**



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

RGB и CMYK



- не все цвета, которые показывает монитор (RGB), можно напечатать (CMYK)
- при переводе кода цвета из RGB в CMYK цвет искажается

RGB(0,255,0)

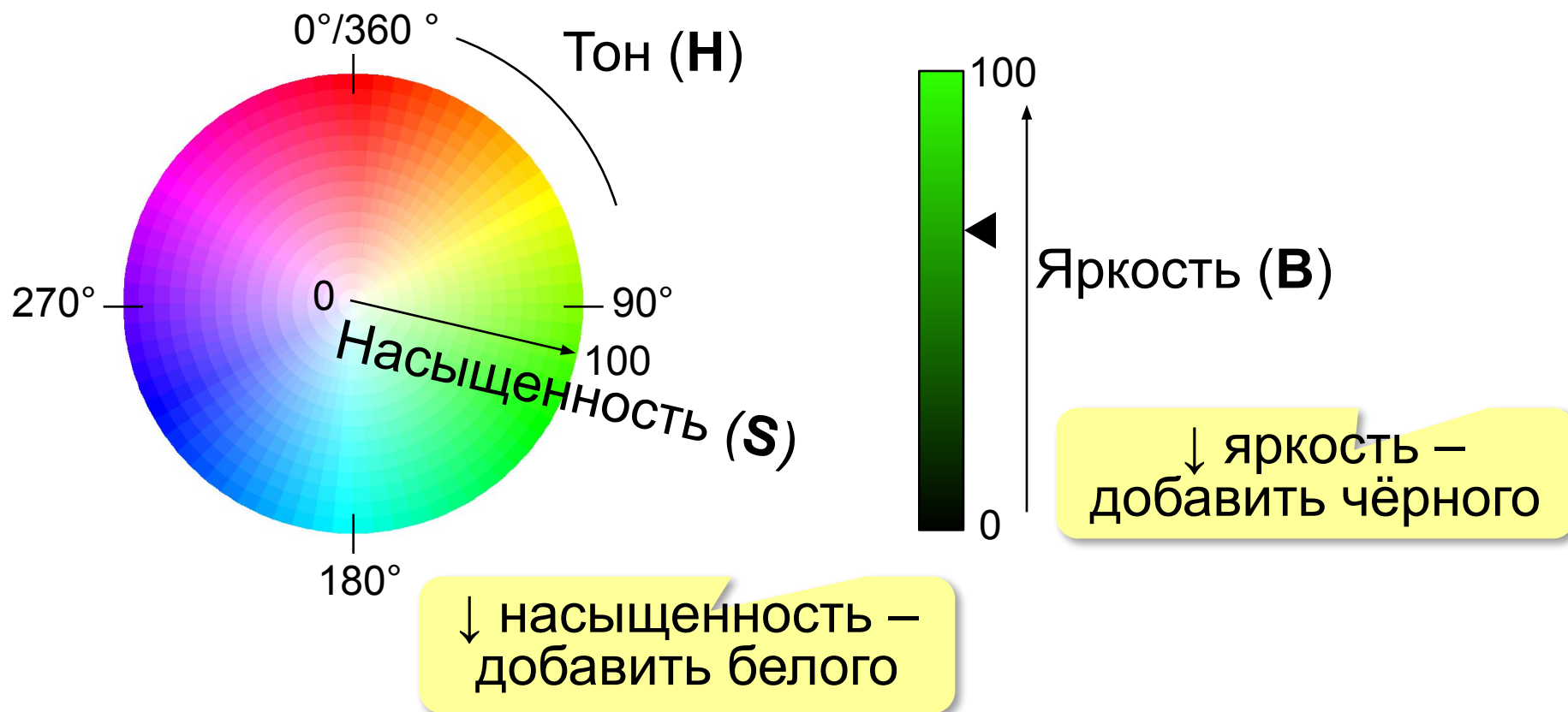
CMYK(65,0,100,0)
 → **RGB(104,175,35)**

Цветовая модель HSB (HSV)

HSB = *Hue* (тон, оттенок)

Saturation (насыщенность)

Brightness (яркость) или *Value* (величина)



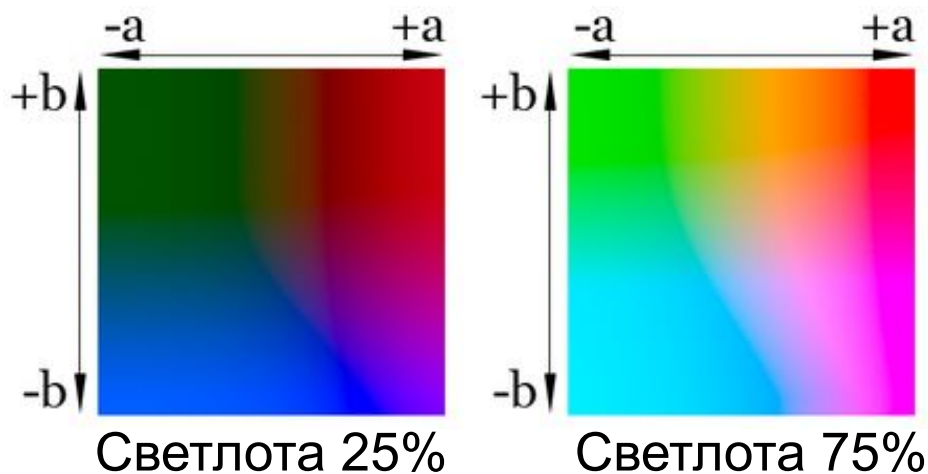
Цветовая модель Lab

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

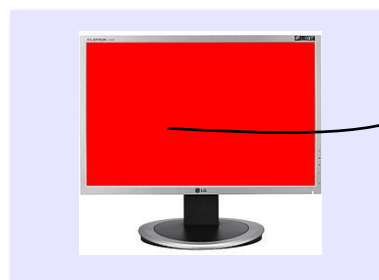


Какой цвет увидим?

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)

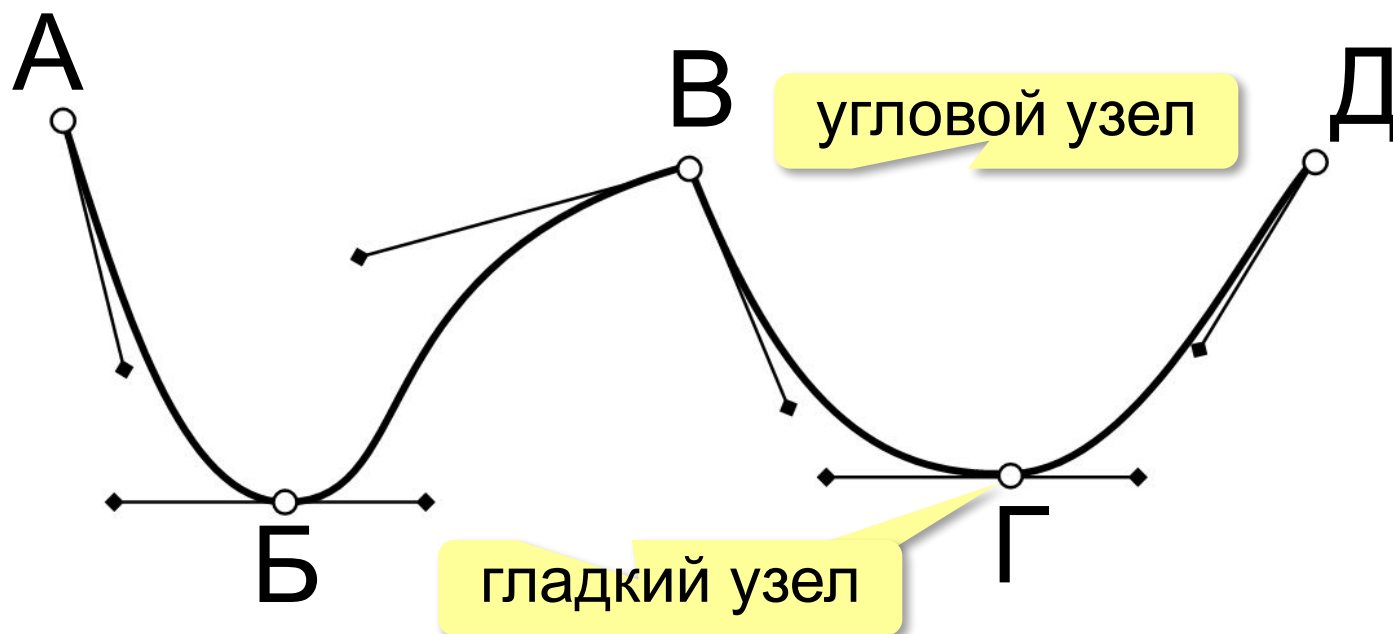
Для каждой фигуры в памяти хранятся:

- 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

Векторные рисунки: SVG

<svg>

прямоугольник

размеры

```
<rect width="135" height="30"  
  x="0" y="10"  
  stroke-width="1" stroke="rgb(0,0,0)"  
  fill="rgb(255,255,255)" />
```

координаты

контур

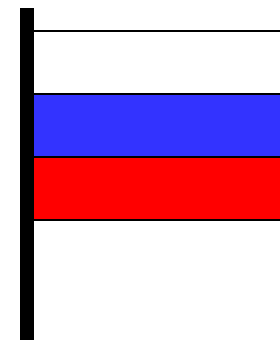
заливка

```
<rect width="135" height="30" x="0" y="40"  
  stroke-width="1" stroke="rgb(0,0,0)"  
  fill="rgb(0,0,255)" />
```

```
<rect width="135" height="30" x="0" y="70"  
  stroke-width="1" stroke="rgb(0,0,0)"  
  fill="rgb(255,0,0)" />
```

```
<line x1="0" y1="0"  
  x2="0" y2="150"  
  stroke-width="15" stroke="rgb(0,0,0)" />
```

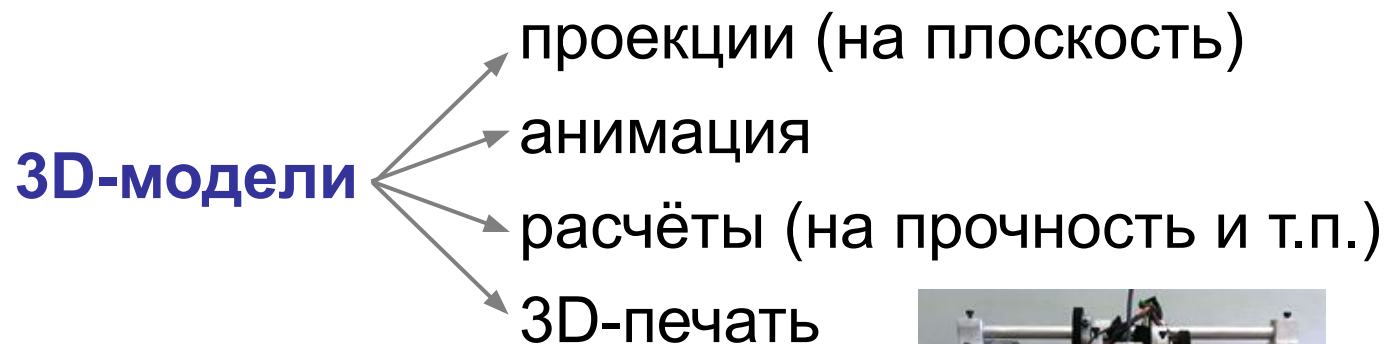
</svg>



3D-графика

Трёхмерная графика (3D-графика) – это раздел компьютерной графики, который занимается созданием моделей и изображений *трёхмерных* объектов.

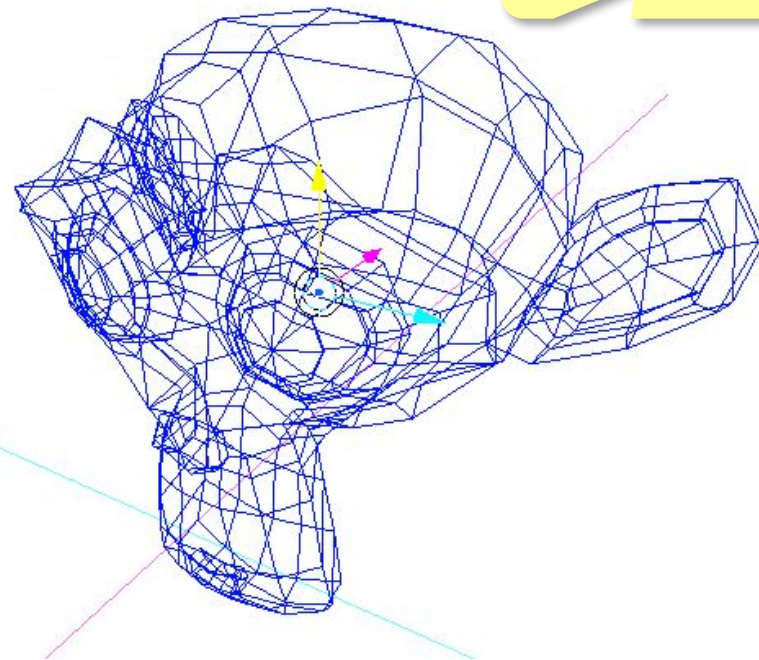
3D-модели: каждая точка имеет 3 координаты



Построение каркаса (рёбер)

узлы
(вершины)

рёбра

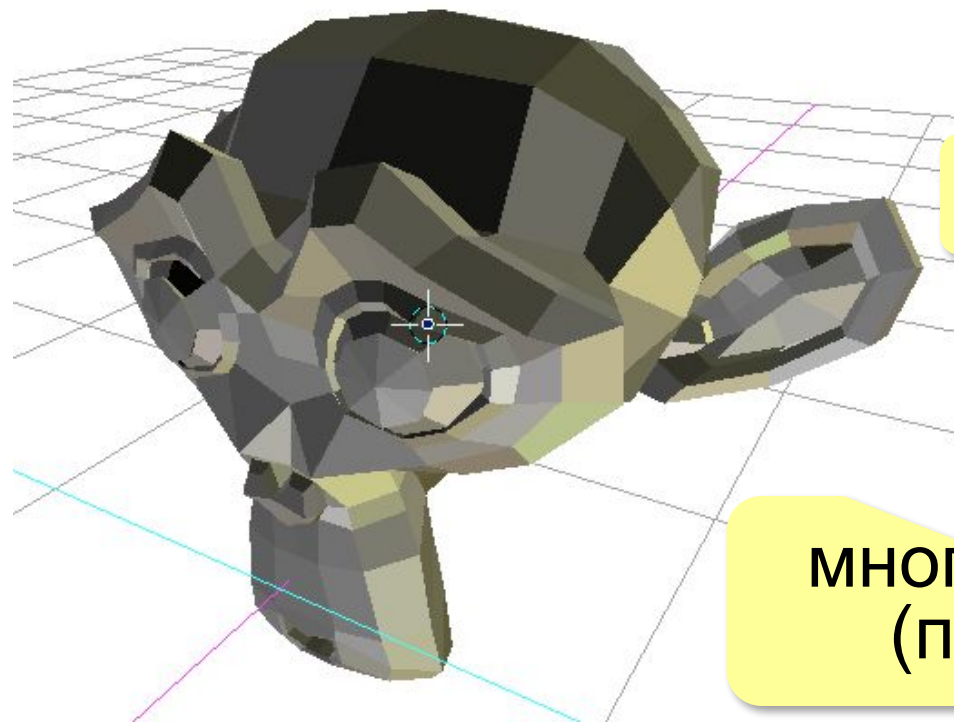


Хранятся координаты точек (x, y, z) !



~~Раск~~ривая или векторная?

Поверхность

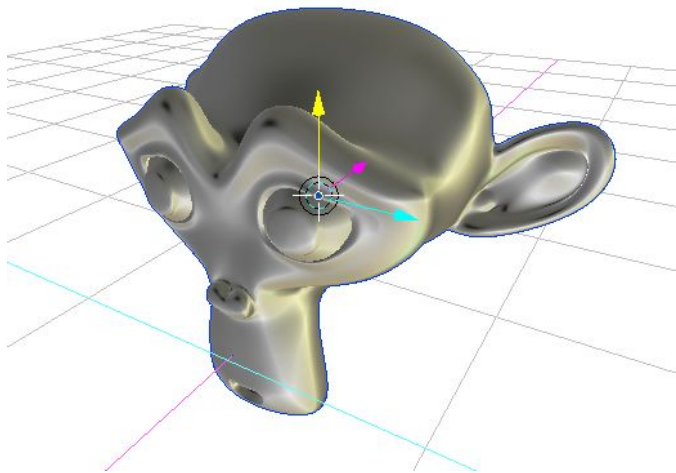


треугольники

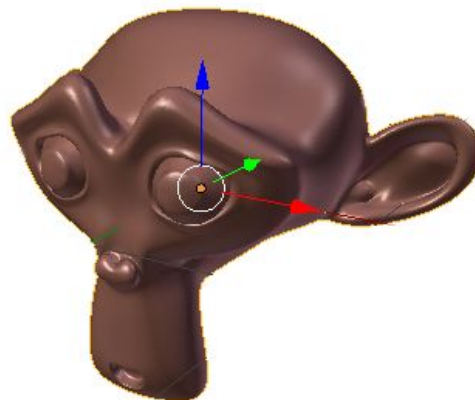
многоугольники
(полигоны)

Завершение модели

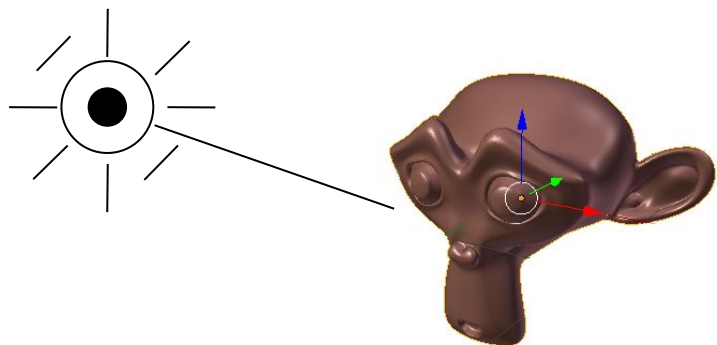
сглаживание



материал

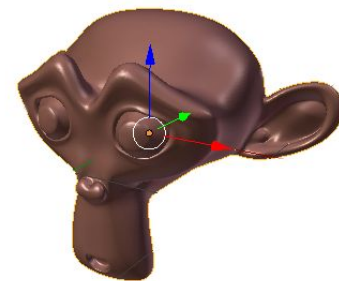
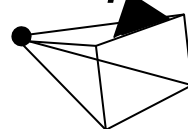


установка света



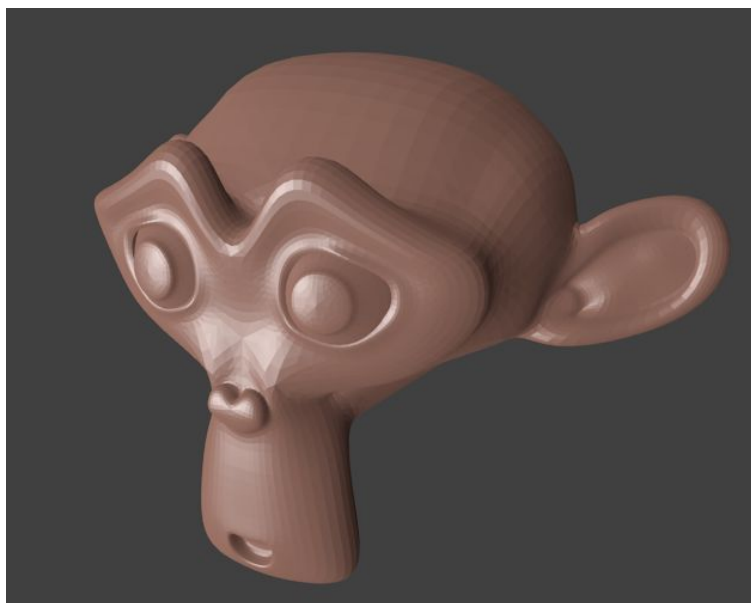
установка камеры

камера



Результат

рендеринг



3D-печать

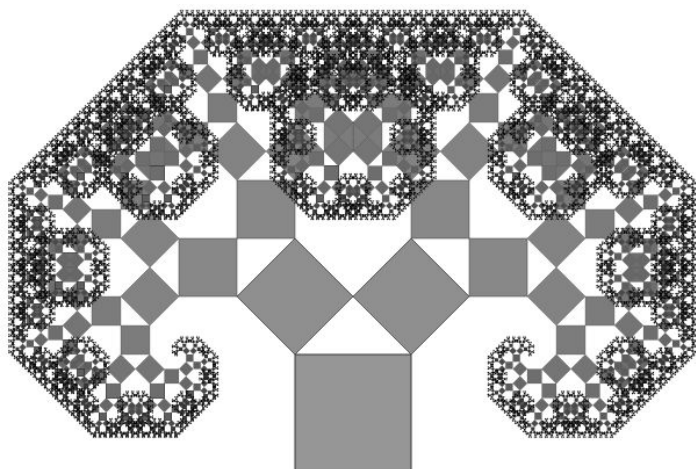


Рендеринг (визуализация) — построение двухмерного изображения по 3D-модели.

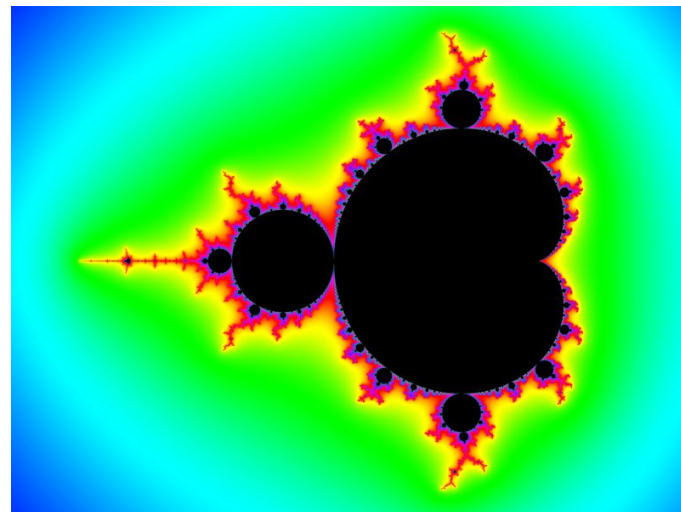
Фрактальная графика

Фрактал — это фигура, обладающая *самоподобием*: основная фигура состоит из нескольких таких же, только меньшего размера.

Пифагорово дерево



Множество Мандельброта

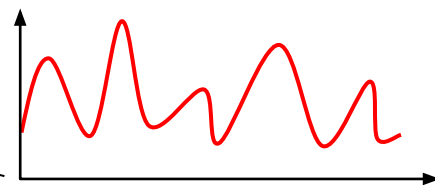
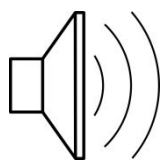


Задаётся математической формулой + алгоритмом построения.

Кодирование информации

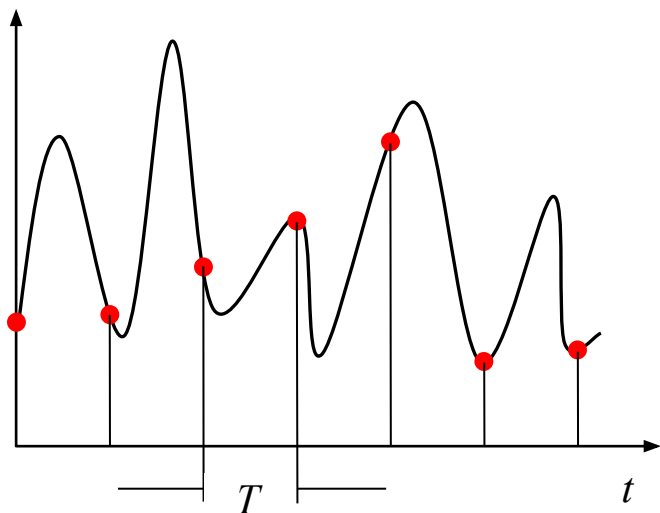
§ 15. Кодирование звуковой и видеоинформации

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

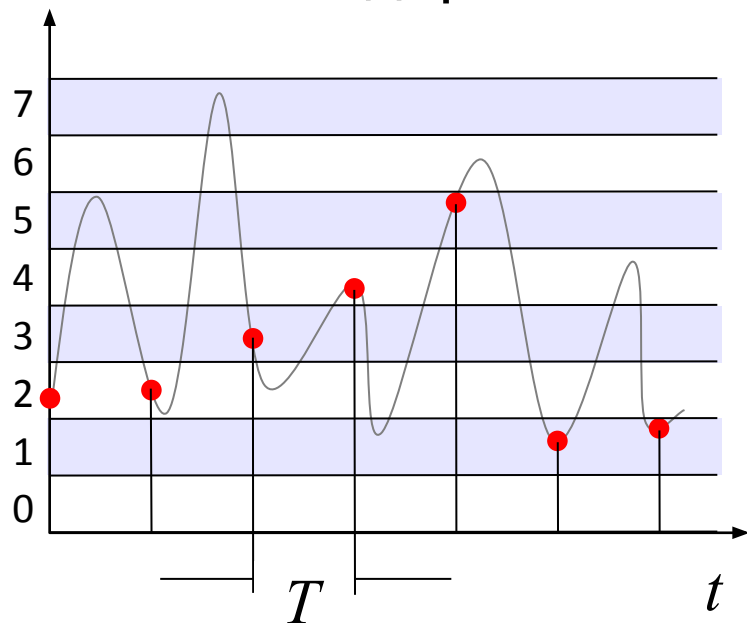


Сколько битов нужно, чтобы записать число 0,6?

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**

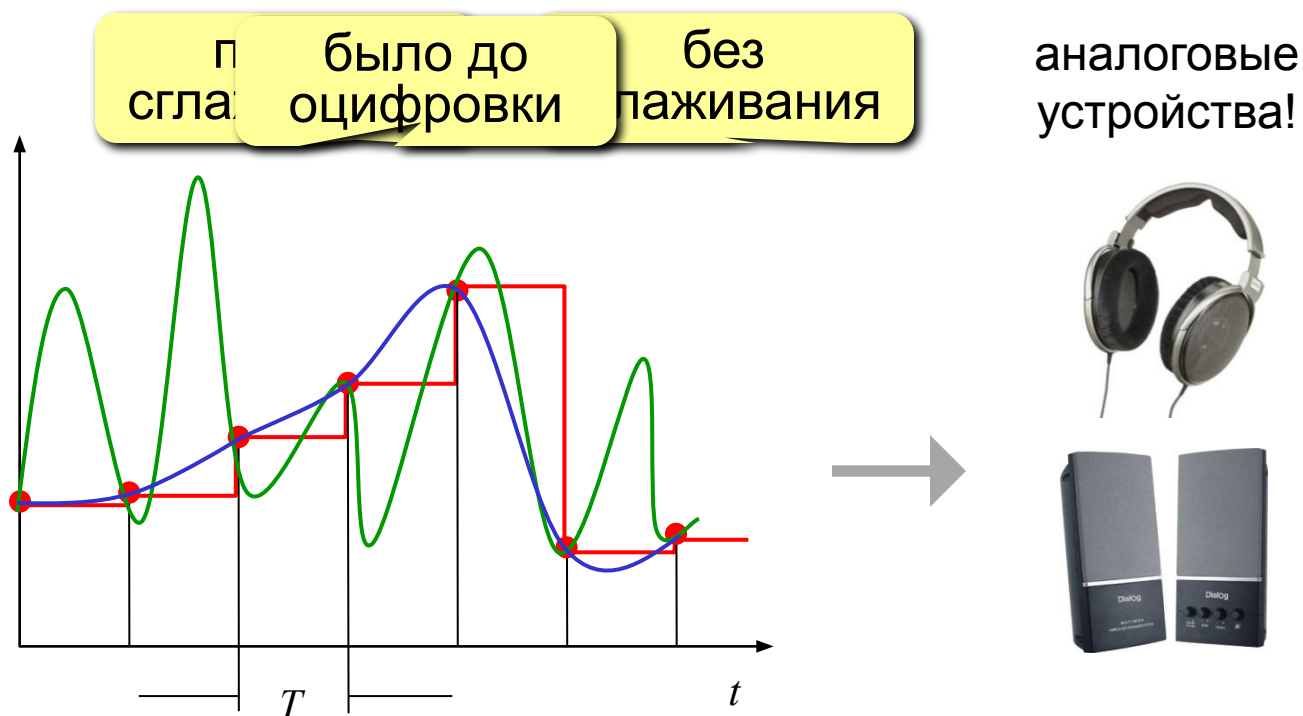
С учётом «стерео»
всего **$88000 \cdot 2 = 176000$ байтов**

За 1 минуту
 $176000 \cdot 60 = 10560000$ байтов
 ≈ 10313 Кбайт ≈ 10 Мбайт

Оцифровка звука

Как восстановить сигнал?

ЦАП = Цифро-Аналоговый Преобразователь



? Как 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

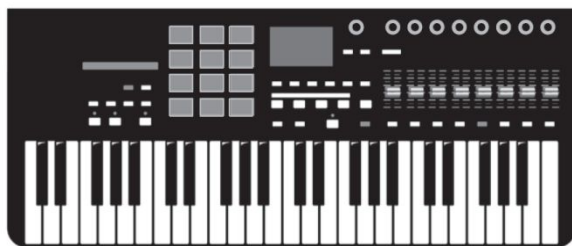
128 мелодических
и 47 ударных

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!

изображения:

- ≥ 25 кад
- **PAL:** 720
за 1
за 1
- **HDTV:** 1
- **4K:** 4096
- исходны
- сжатие (
- DivX, Xv

звук:

- 48 кГц, 1
- сжатие (



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

Конец фильма

ПОЛЯКОВ Константин Юрьевич

д.т.н., учитель информатики

ГБОУ СОШ № 163, г. Санкт-Петербург

kpolyakov@mail.ru

ЕРЕМИН Евгений Александрович

к.ф.-м.н., доцент кафедры мультимедийной

дидактики и ИТО ПГГПУ, г. Пермь

eremin@pspu.ac.ru

Источники иллюстраций

1. <http://drupal.org>
2. [http:// www.cyberphysics.co.uk](http://www.cyberphysics.co.uk)
3. <http://epson.su>
4. <http://www8.hp.com>
5. <https://www.khanacademy.org>
6. <http://head-fi.org>
7. <http://en.wikipedia.org>
8. <http://ru.wikipedia.org>
9. авторские материалы