

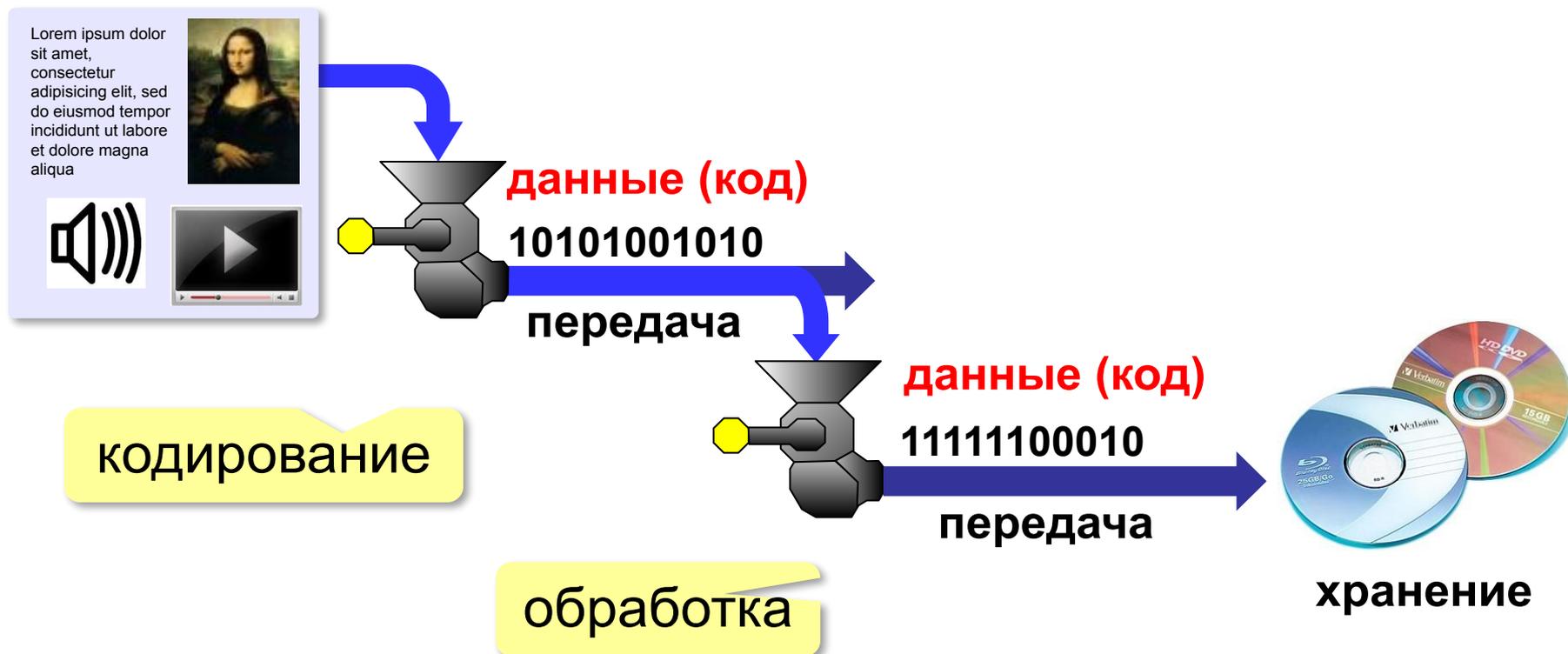
Кодирование информации

§ 16. Кодирование графической информации

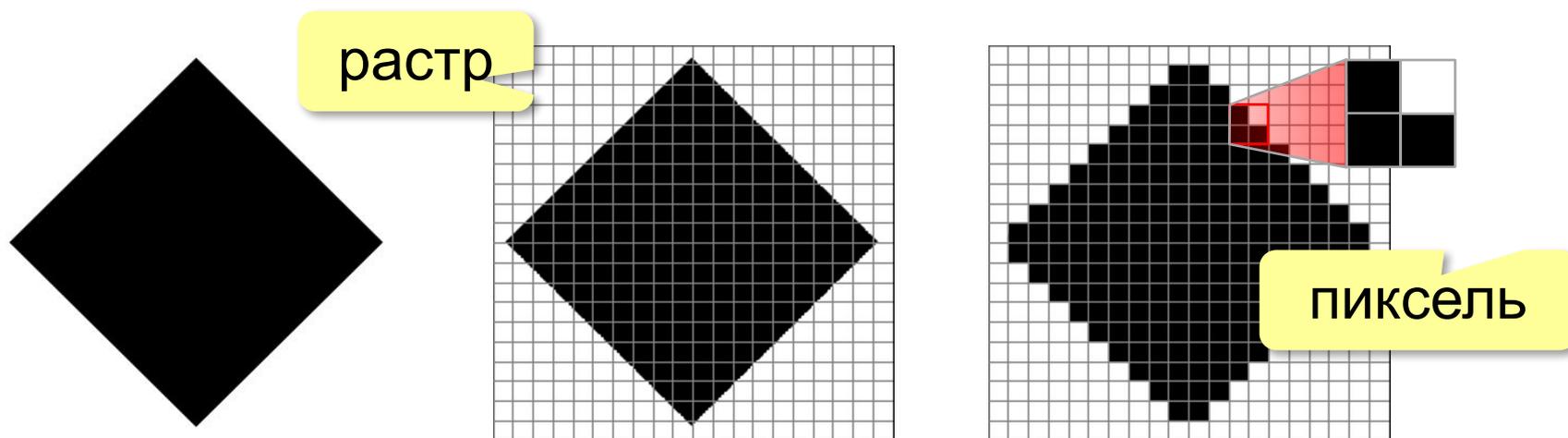
Зачем кодировать информацию?

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

В компьютерах используется двоичный код:



Растровое кодирование



дискретизация

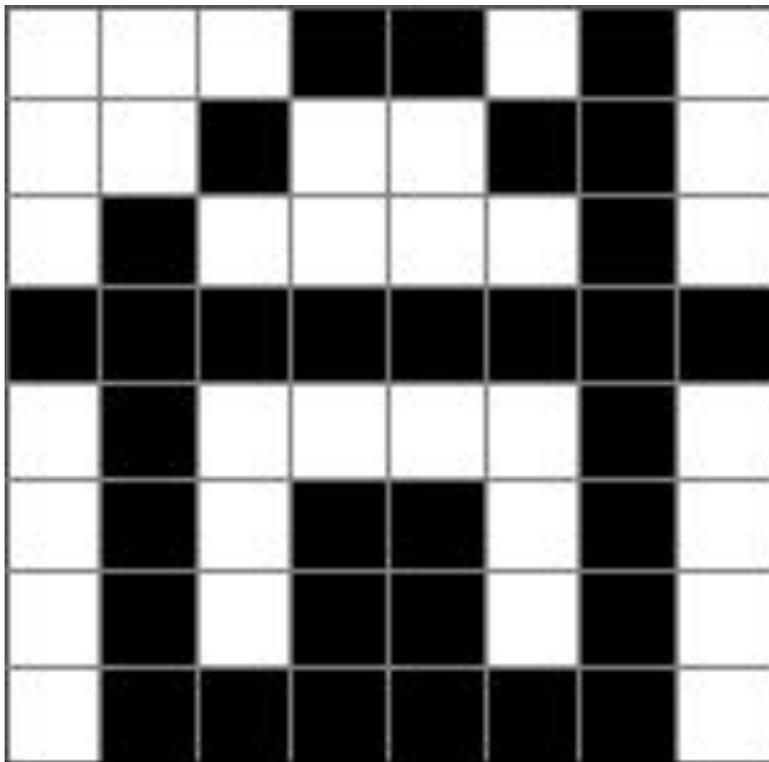


Рисунок искажается!

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

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

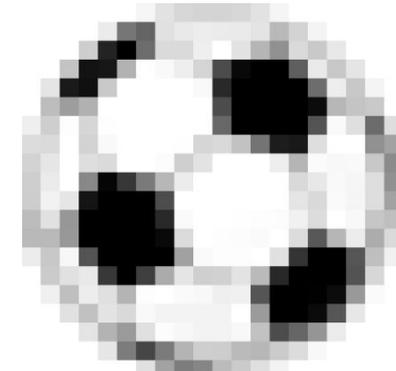


96 ppi

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?

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

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

Цветовая модель RGB

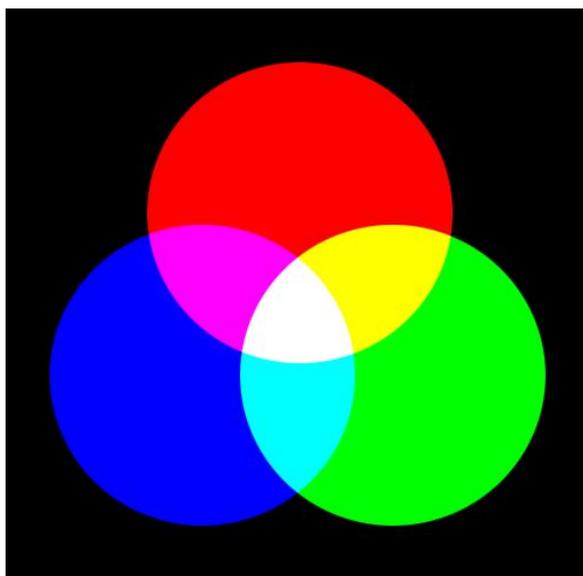
Д. Максвелл, 1860

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?

$256 \cdot 256 \cdot 256 = \mathbf{16\ 777\ 216}$ (*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



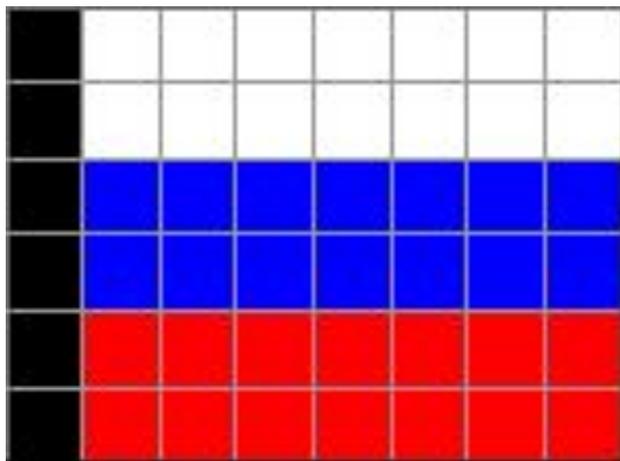
Как уменьшить размер файла?

- 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.

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



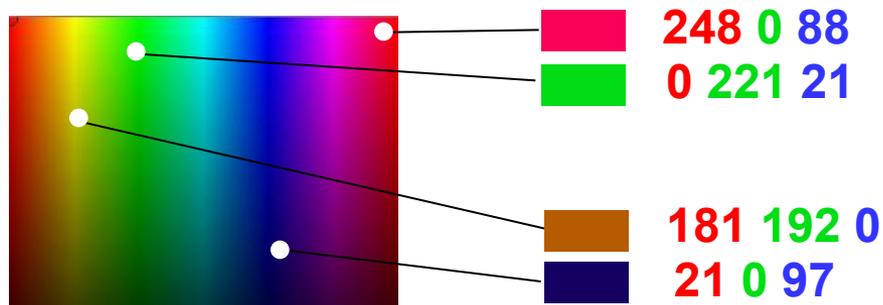
Сколько занимает палитра?

$3 \cdot 4 = 12$
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

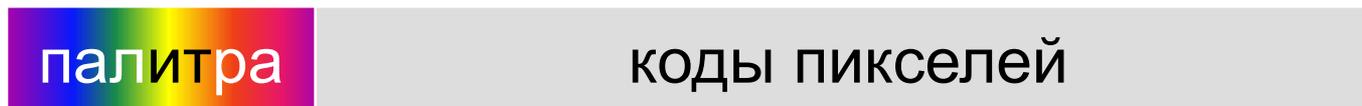
0	1	...	254	255
248 0 88	0 221 21	...	181 192 0	21 0 97

Шаг 4. Код пикселя = номеру его цвета в палитре

2	45	65	14	...	12	23
---	----	----	----	-----	----	----

Кодирование с палитрой

Файл с палитрой:

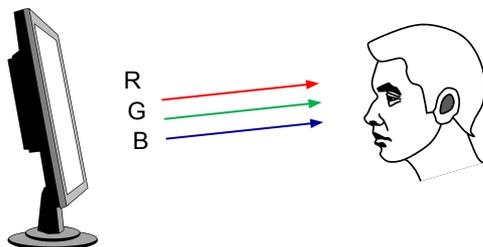


Количество цветов	Размер палитры (байтов)	Глубина цвета (битов на пиксель)
2	6	1
4	12	2
16	48	4
256	768	8

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

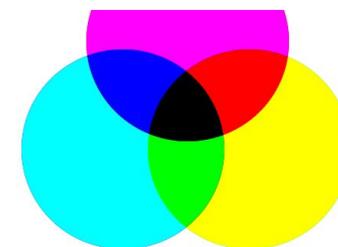
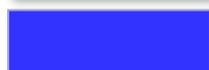
= пурпурный

M = Magenta

Белый – синий

= желтый

Y = Yellow



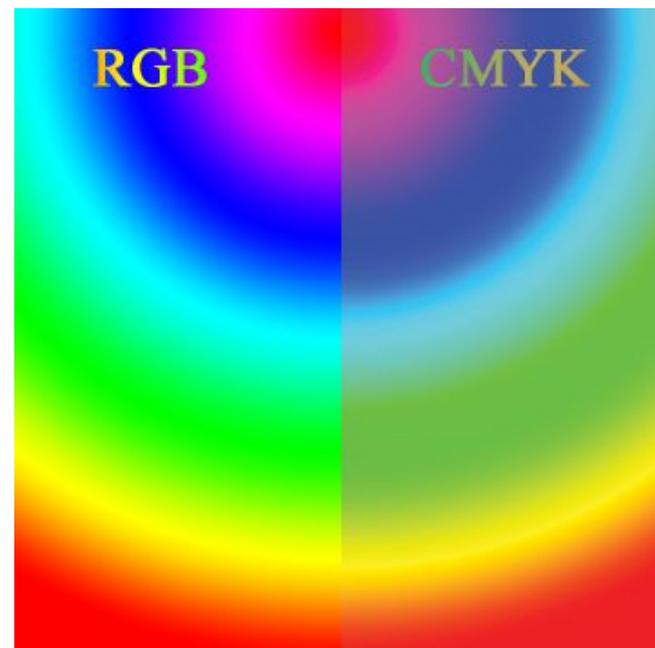
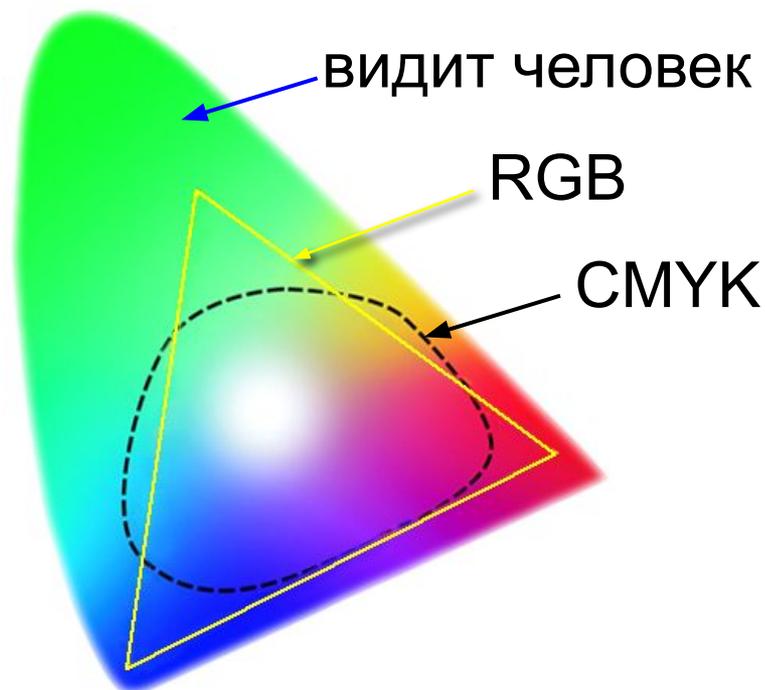
Модель СМУ

Модель СМУК: + **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)

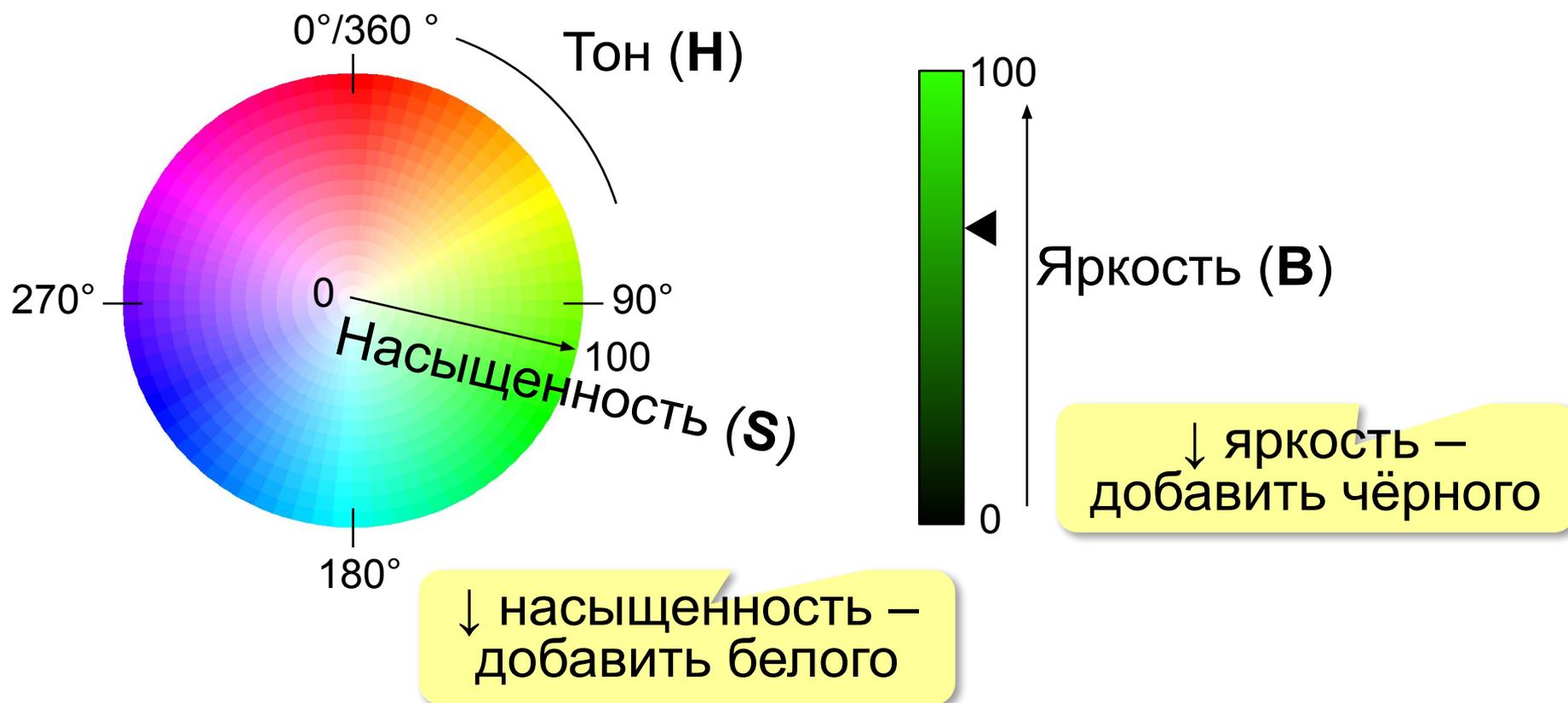
CMYK(65,0,100,0)
 → **RGB(104,175,35)**

Цветовая модель HSB (HSV)

HSB = *Hue* (тон, оттенок)

Saturation (насыщенность)

Brightness (яркость) или *Value* (величина)



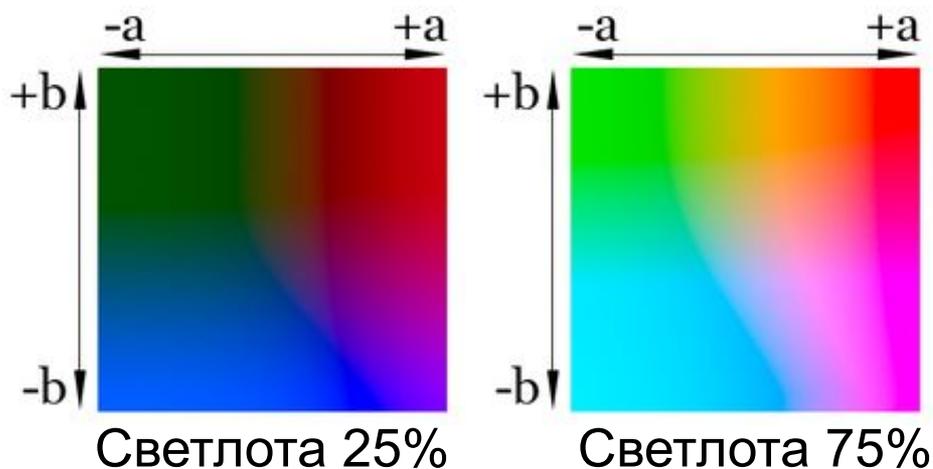
Цветовая модель Lab

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

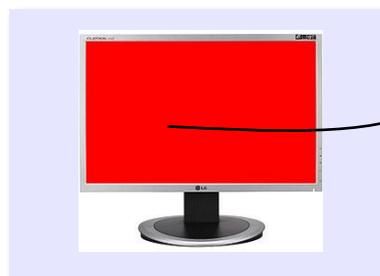


Какой цвет увидим?

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?)

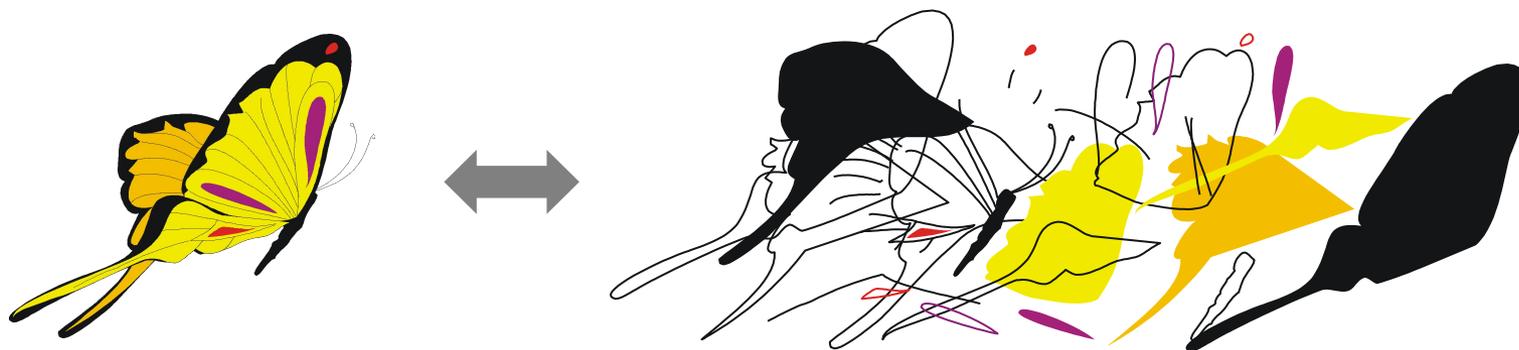
Векторное кодирование

Рисунки из геометрических фигур:

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

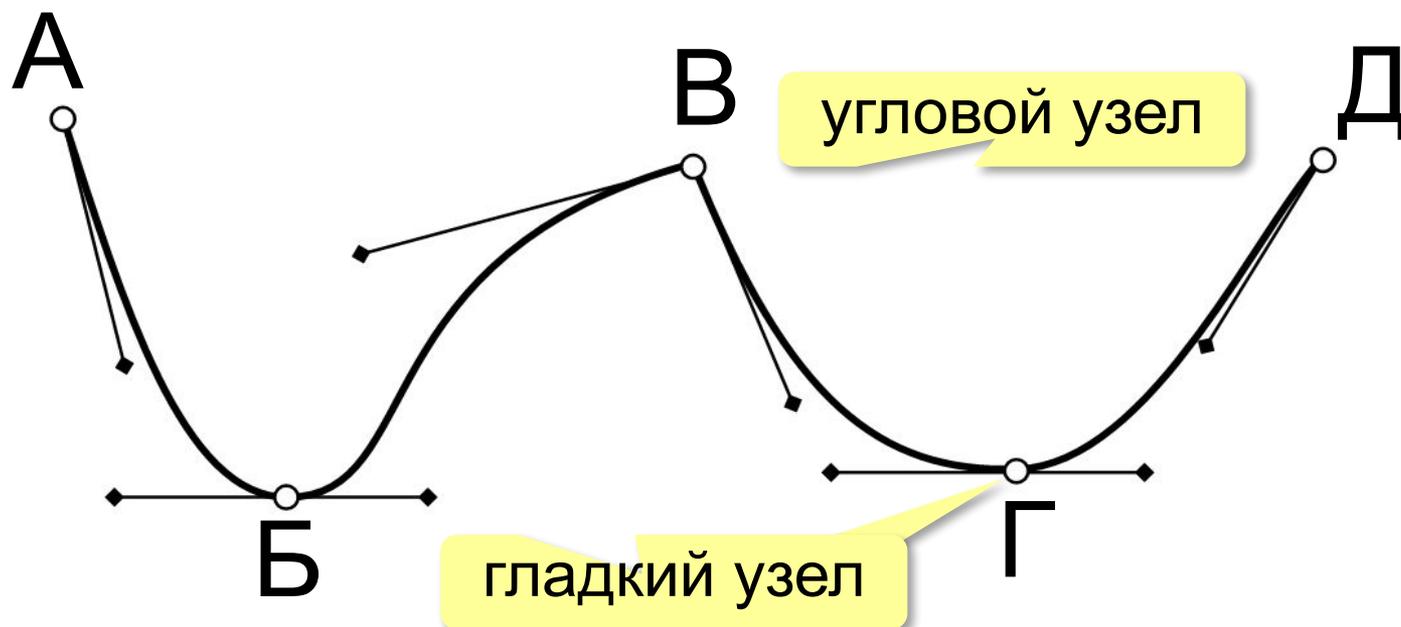
Для каждой фигуры в памяти хранятся:

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



Векторное кодирование

Кривые Безье:

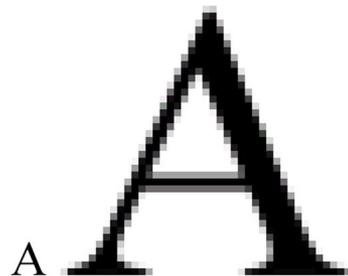


Хранятся координаты узлов и концов «рычагов»
(3 точки для каждого узла, кривые 3-го порядка).

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>

