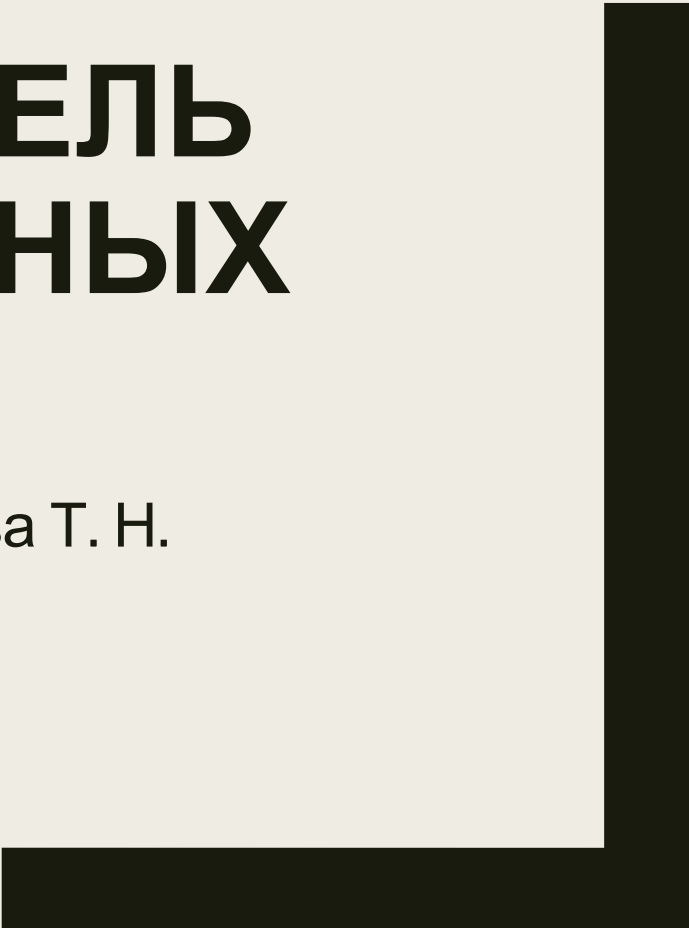


# **РАСТРОВАЯ МОДЕЛЬ ПРОСТРАНСТВЕННЫХ ДАННЫХ**

Разработал: д. с.-х. н., проф. Мыслыва Т. Н.



# РАССМАТРИВАЕМЫЕ ВОПРОСЫ

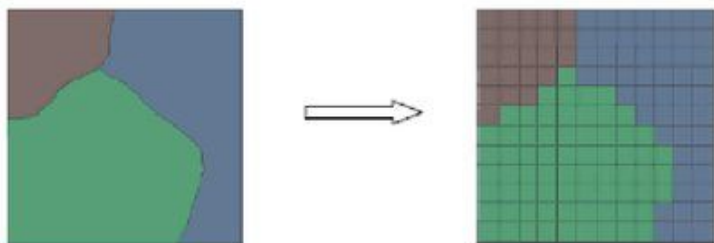
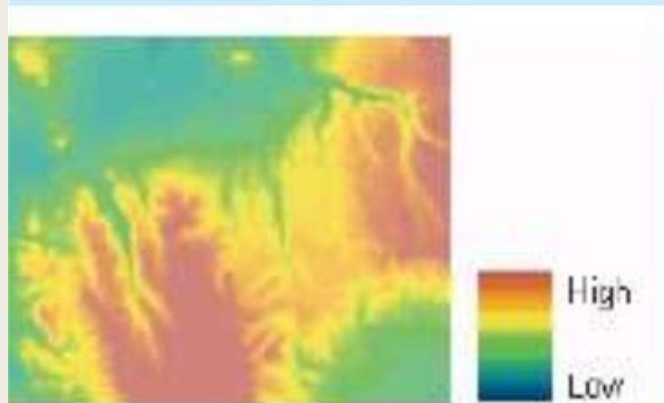
1. Понятие растровой модели и ее характеристики.
2. Понятие пиксела и воксела. Характеристики пиксела. Деление растрового изображения по фотометрическому содержанию и шкале уровней яркости.
3. Цветовые модели RGB, CMY, CMYK, HSV.
4. Алгоритмы сжатия растровой информации. Понятие о форматах файлов для хранения растровых изображений (tiff, bmp, jpeg, gif, psx).
5. Поддержка растровых данных в ГИС. Работа с растрами большой размерности

# **ПОНЯТИЕ РАСТРОВОЙ МОДЕЛИ И ЕЕ ХАРАКТЕРИСТИКИ**



# Модели данных в ГИС

Растровые



Пространственная информация представлена в виде таблицы, каждой ячейке которой соответствует заданный цвет.

Векторные

2D

3D

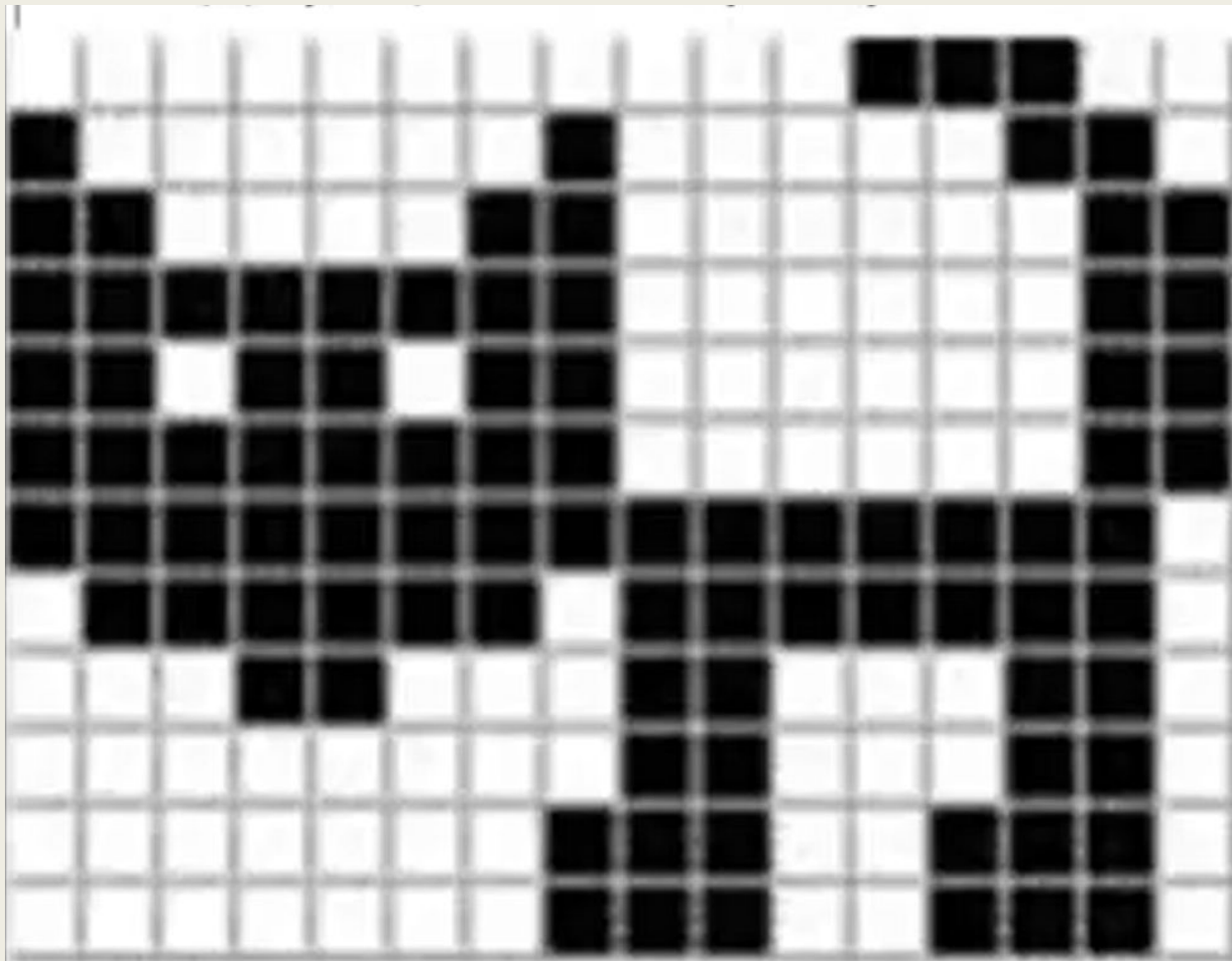


Объекты типа «полигон» создаются в результате сборки полигонов из дуг, образующих замкнутые контура.

# РАСТРОВОЕ КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

- Графическую информацию разбивают на крошечные фрагменты одинаковой величины (растровая модель) или простейшие геометрические объекты (векторная модель).
- Растровое изображение формируется из определенного количества строк, а каждая строка – из точек, которые принято называть пикселями.
- Графический объект, подлежащий отображению в цифровом виде, делится вертикальными и горизонтальными линиями на пиксели.
- Цвет каждого пикселя кодируется двоичным числом. Такой способ кодирования называется растровым.

# РАСТРОВОЕ КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ



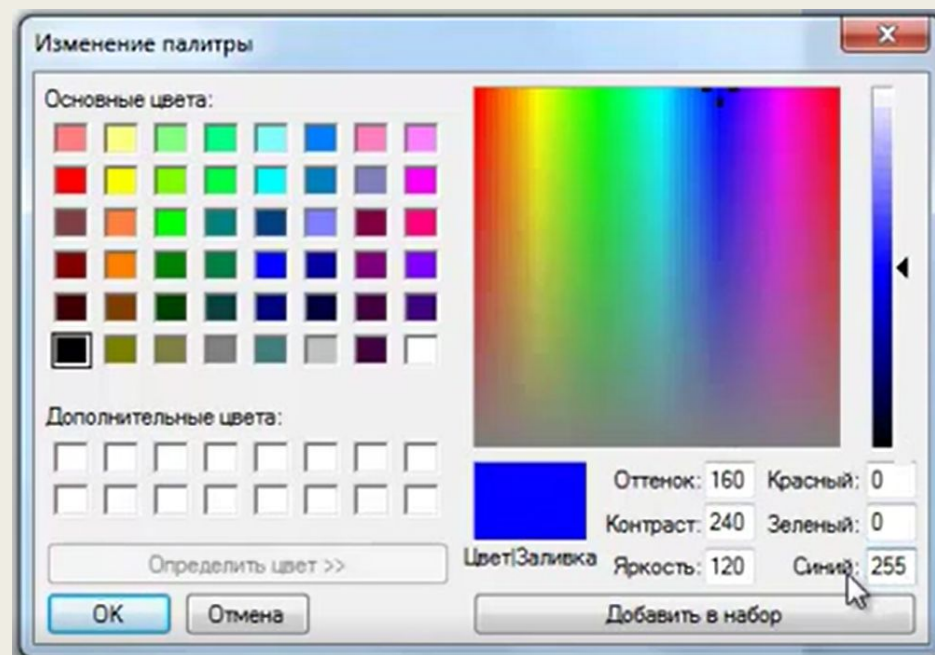
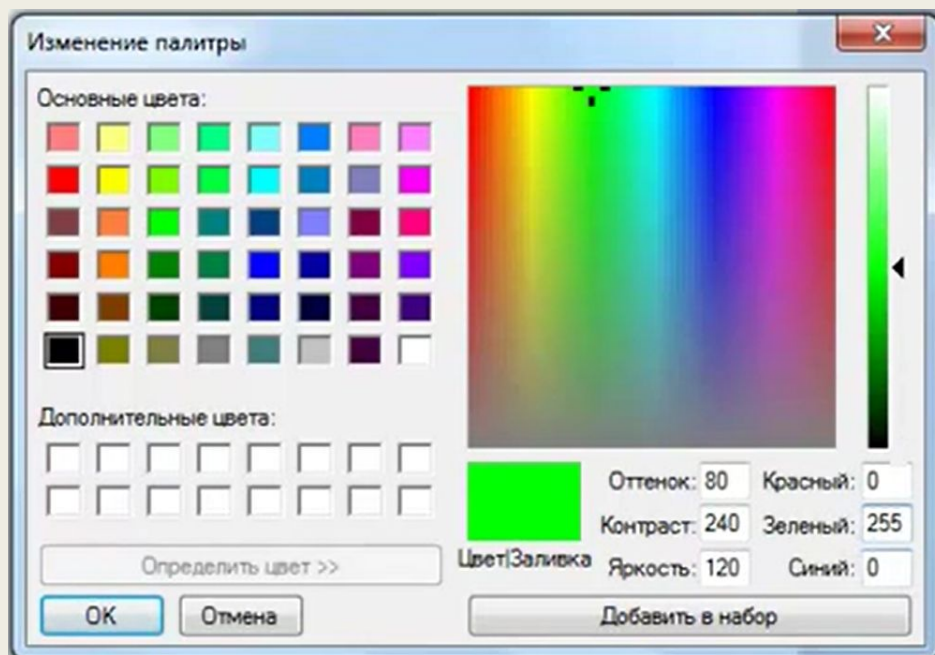
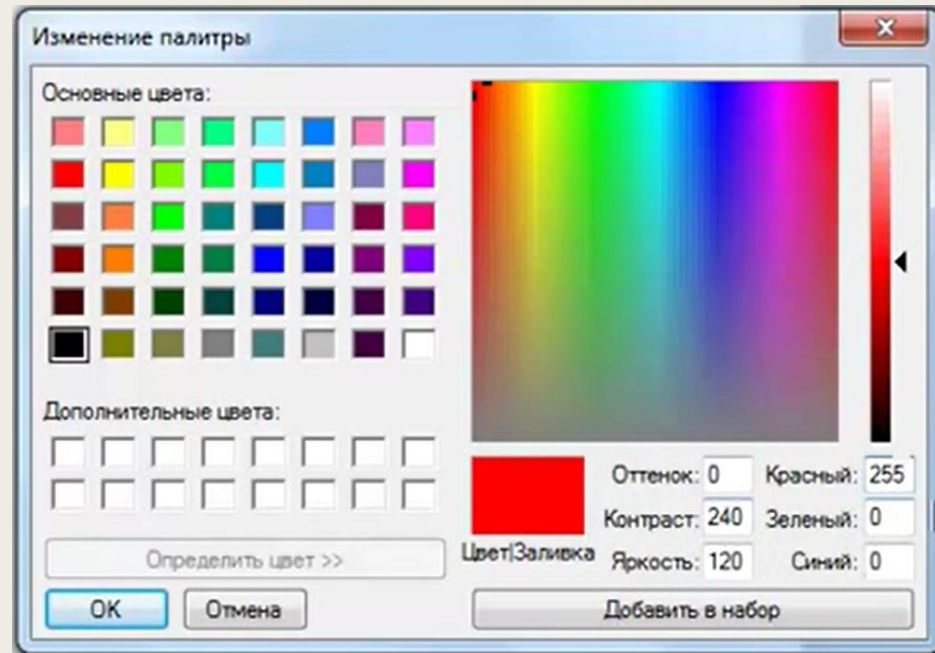
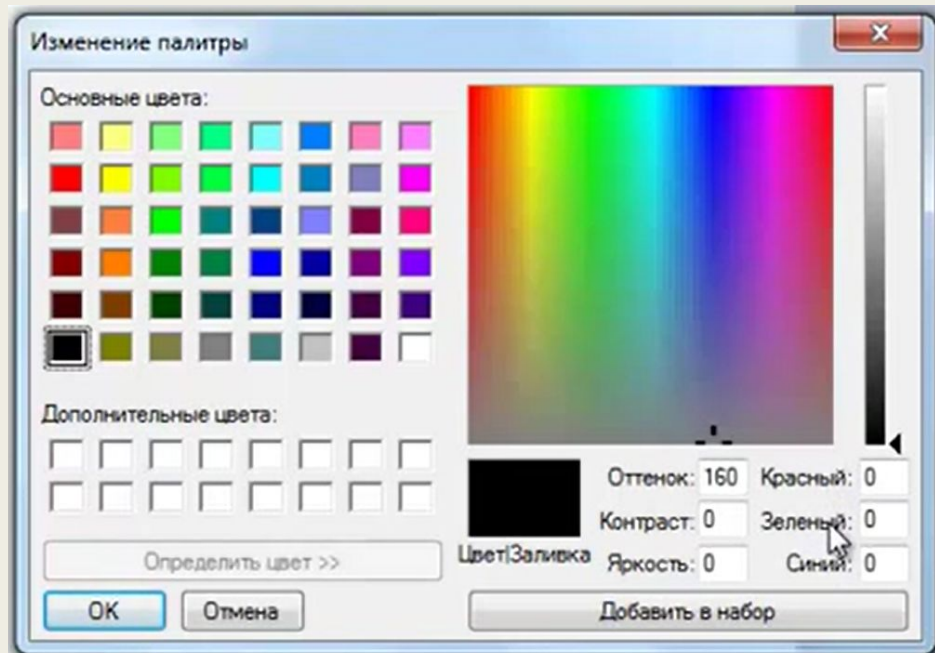
```
00000000000011100
1000000100000110
1100001100000011
1111111100000011
1101101100000011
1111111100000011
1111111111111110
0111111011111110
0001100011000110
0000000011000110
0000000111001110
0000000111001110
```

# РАСТРОВОЕ КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

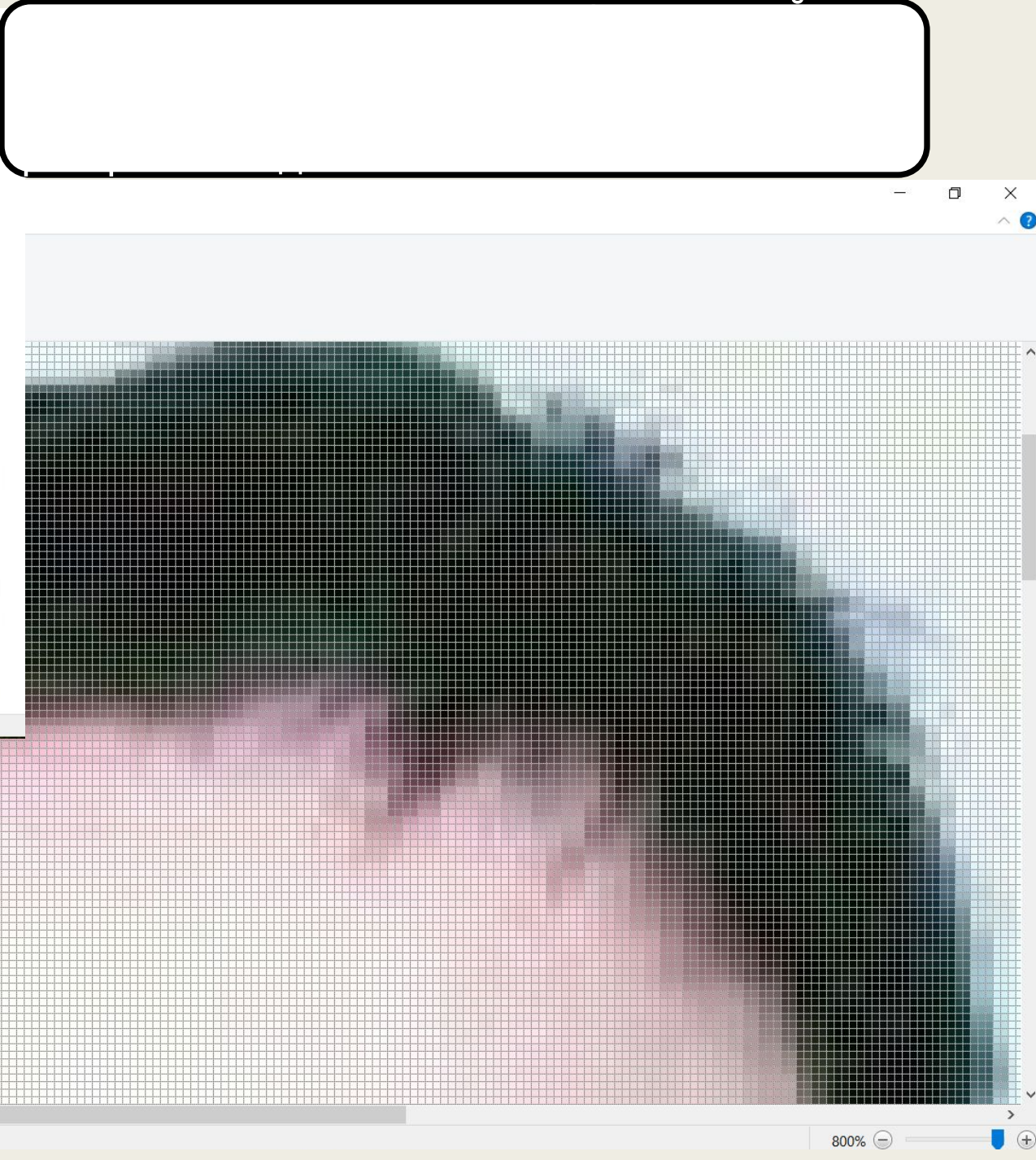
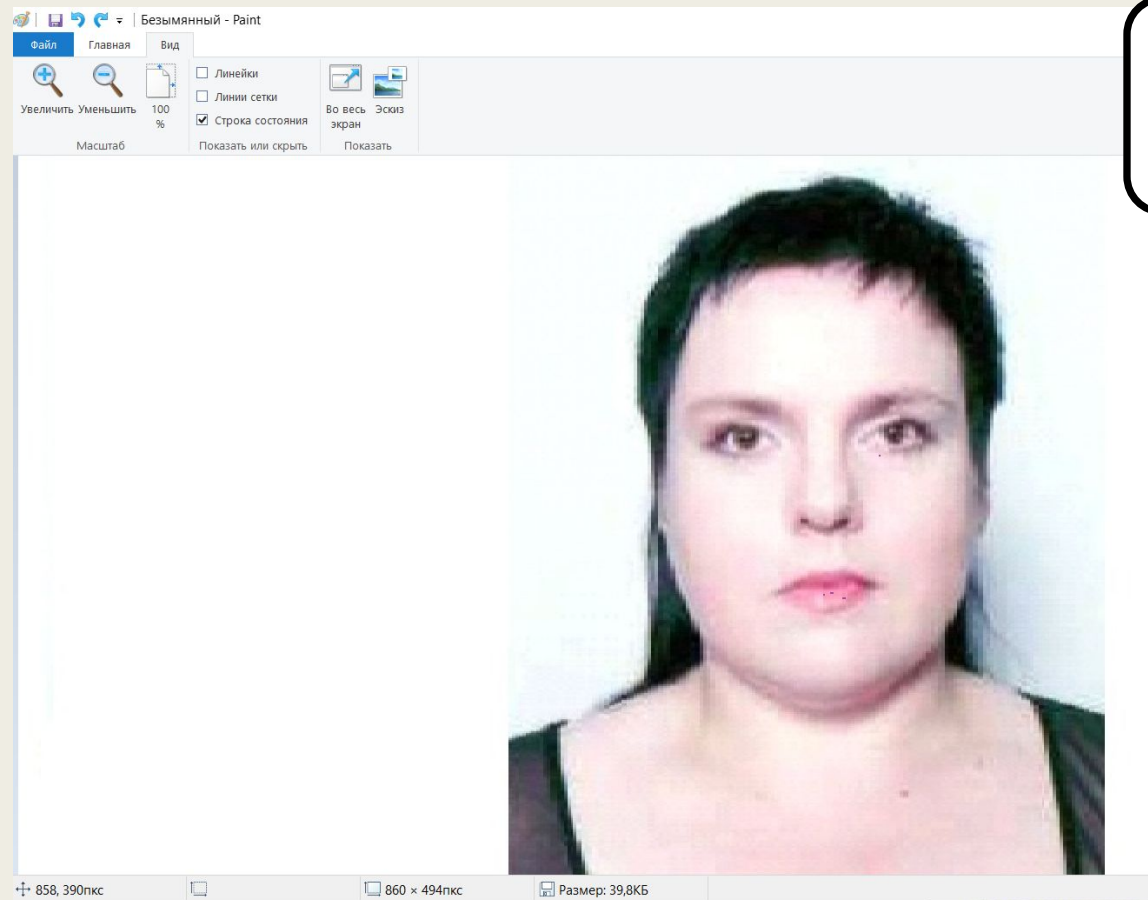
- Каждый пиксель на цветном экране – это совокупность трех точек красного, зеленого и синего цвета.
- **Каждый цвет кодируется** цепочкой из восьми единиц и нулей, то есть **8-ю битами**, а **каждый пиксель** – цепочкой из 24 единиц и нулей, то есть **24-мя битами**.
- Поскольку **каждый цвет позволяет закодировать 256 различных оттенков**, максимальное число цветов, которое можно отобразить с помощью трех основных цветов, составляет:

$$256 * 256 * 256 = 16\,777\,216$$











# ПОНЯТИЕ РАСТРОВОЙ МОДЕЛИ ПРОСТРАНСТВЕННЫХ ДАННЫХ

- **РАСТРОВАЯ МОДЕЛЬ ПРОСТРАНСТВЕННЫХ ДАННЫХ** - цифровое представление пространственных объектов в виде совокупности ячеек растра (пикселей) с присвоенными им значениями класса объекта.
- Растровая модель предполагает позиционирование объектов с указанием их положения в соответствующей растру прямоугольной матрице единообразно для всех типов пространственных объектов (точек, линий, полигонов и поверхностей).
- Данная модель разбивает всю изучаемую территорию на элементы регулярной сетки или ячейки, каждая из которых содержит только одно значение.
- **ОСНОВНОЕ НАЗНАЧЕНИЕ РАСТРОВЫХ МОДЕЛЕЙ** - непрерывное отображение поверхности. Если векторная модель дает информацию о том, где расположен тот или иной объект, то растровая - показывает, что расположено в той или иной точке территории.
- Является пространственно заполненной, поскольку каждое местоположение на изучаемой территории соответствует ячейке растра, иными словами - растровая модель оперирует элементарными местоположениями.

В общем виде любой растр можно записать в виде следующей

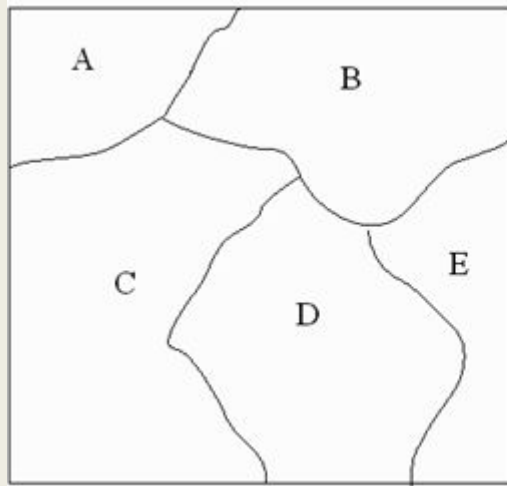
Color(0,0)	Color(1,0)	Color(2,0)	Color(3,0)		Color(n,0)
Color(0,1)	Color(1,1)	Color(2,1)	Color(3,1)		Color(n,1)
Color(0,2)	Color(1,2)	Color(2,2)	Color(3,2)		Color(n,2)
Color(0,m)	Color(1,m)	Color(2,m)	Color(3,m)		Color(n,m)

# Матрица, соответствующая фрагменту раstra

15	15	15	15	0	15	0	15
15	15	15	15	0	15	0	15
15	15	15	15	0	15	0	15
15	15	15	15	15	15	15	15
15	15	15	15	0	0	0	15
15	15	15	15	15	15	15	15
15	15	15	15	15	15	15	15
15	15	15	15	15	15	15	15

15(0,0) 15(1,0) 15(2,0) 15(3,0) 0(4,0) 15(5,0) 0(6,0) 15(7,0)  
 15(0,1) 15(1,1) 15(2,1) 15(3,1) 0(4,1) 15(5,1) 0(6,1) 15(7,1)  
 15(0,2) 15(1,2) 15(2,2) 15(3,2) 0(4,2) 15(5,2) 0(6,2) 15(7,2)  
 15(0,3) 15(1,3) 15(2,3) 15(3,3) 15(4,3) 15(5,3) 15(6,3) 15(7,3)  
 15(0,4) 15(1,4) 15(2,4) 15(3,4) 0(4,4) 0(5,4) 0(6,4) 15(7,4)  
 15(0,5) 15(1,5) 15(2,5) 15(3,5) 15(4,5) 15(5,5) 15(6,5) 15(7,5)  
 15(0,6) 15(1,6) 15(2,6) 15(3,6) 15(4,6) 15(5,6) 15(6,6) 15(7,6)  
 15(0,7) 15(1,7) 15(2,7) 15(3,7) 15(4,7) 15(5,7) 15(6,7) 15(7,7)

# ПОНЯТИЕ РАСТРОВОЙ МОДЕЛИ ПРОСТРАНСТВЕННЫХ ДАННЫХ



A	A	A	A	B	B	B	B	B	B
A	A	A	A	B	B	B	B	B	B
A	A	A	B	B	B	B	B	B	B
C	C	C	C	C	C	B	B	B	E
C	C	C	C	C	D	D	E	E	E
C	C	C	C	D	D	D	D	E	E
C	C	C	D	D	D	D	D	D	E
C	C	C	C	D	D	D	D	D	E
C	C	C	C	D	D	D	D	D	E
C	C	C	C	C	D	D	D	E	E



■ вырубки    ■ леса  
■ болота    ■ луга



● точки    ↑ дуги    ■ полигоны

# ХАРАКТЕРИСТИКИ РАСТРОВОЙ МОДЕЛИ ПРОСТРАНСТВЕННЫХ ДАННЫХ



# ХАРАКТЕРИСТИКИ РАСТРОВОЙ МОДЕЛИ ПРОСТРАНСТВЕННЫХ ДАННЫХ

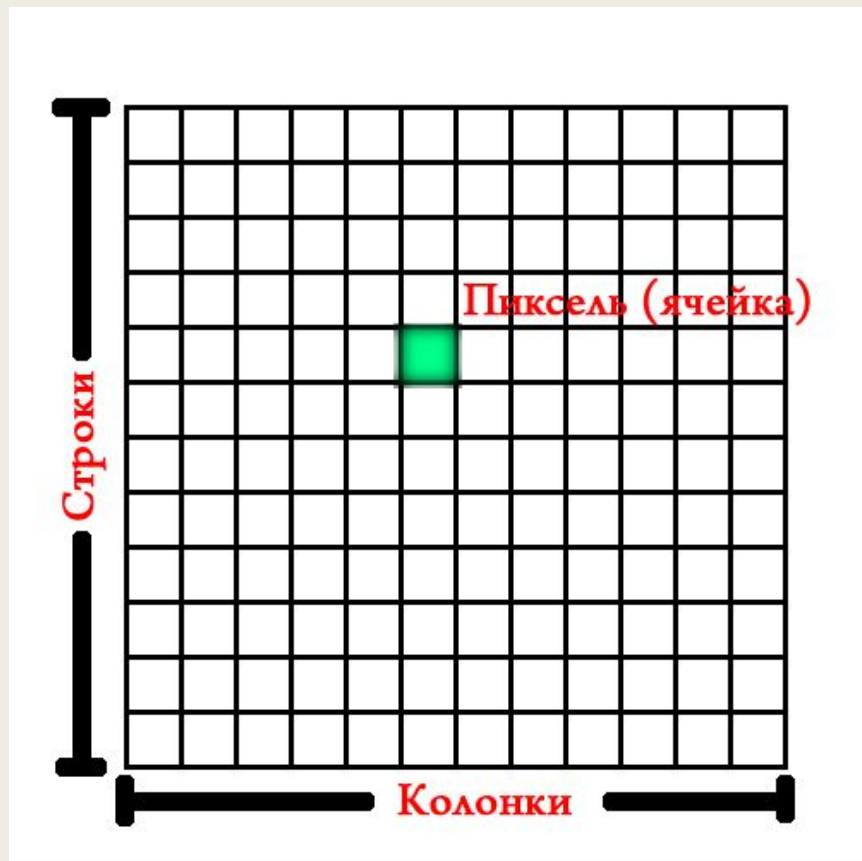


- Растр может содержать один из трех типов информации.
- Растр с тематическими данными описывает территорию качественно, то есть дает представление о том, каковы свойства поверхности в данной точке. Например, возможны следующие градации значений тематического растра для типов земель: пахотные, луговые, лесные.
- Спектральные данные дают количественную характеристику, демонстрируют какова величина одного свойства в данной точке. Примером могут служить содержание гумуса, имеющее определенное значение в каждой точке.
- И, наконец, фотографии, сканированные карты и другие графические данные, несущие только визуальную информацию.



**ПОНЯТИЕ ПИКСЕЛА И ВОКСЕЛА.  
ХАРАКТЕРИСТИКИ ПИКСЕЛА.  
ДЕЛЕНИЕ РАСТРОВОГО  
ИЗОБРАЖЕНИЯ ПО  
ФОТОМЕТРИЧЕСКОМУ  
СОДЕРЖАНИЮ И ШКАЛЕ УРОВНЕЙ  
ЯРКОСТИ**

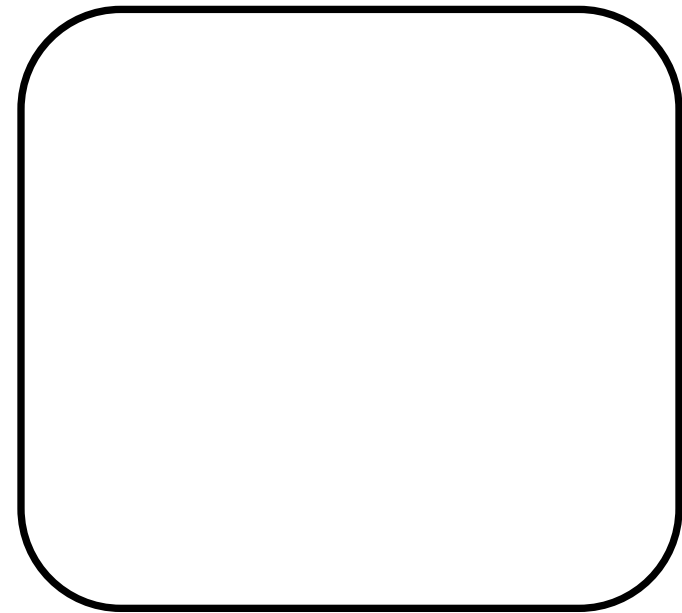
# ПОНЯТИЕ ПИКСЕЛА



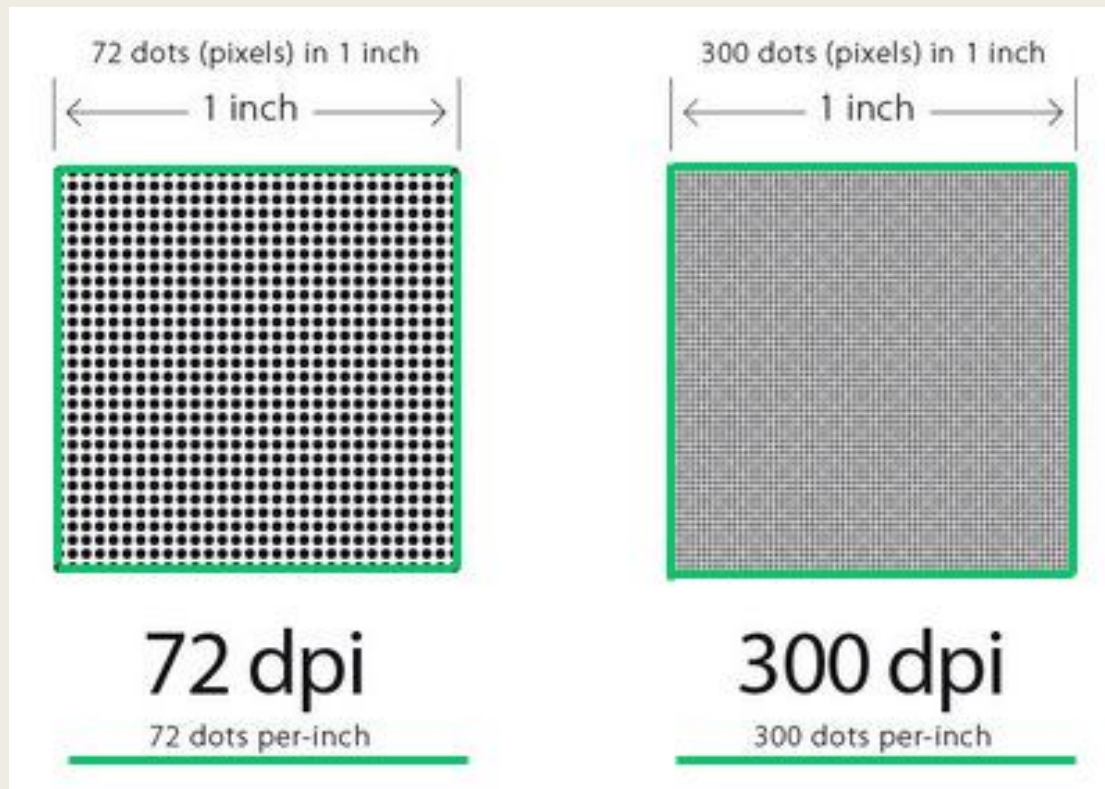
Первая публикация термина «пиксел» была сделана в журнале Wireless World в 1927 году, хотя он использовался ранее в различных патентах США, поданных еще в 1911 году.

# КРАТНЫЕ И ДОЛЬНЫЕ ЕДИНИЦЫ ПИКСЕЛА

Кратные				Дольные			
величина	название	обозначение		величина	название	обозначение	
$10^1$ пикс	декапиксель	дапикс	dapel	$10^{-1}$ пикс	деципиксель	дпикс	dpel
$10^2$ пикс	гектопиксель	гпикс	hpel	$10^{-2}$ пикс	сантипиксель	спикс	spel
$10^3$ пикс	килопиксель	кпикс	kpel	$10^{-3}$ пикс	миллипиксель	мпикс	mpel
$10^6$ пикс	мегапиксель	Мпикс	Mpel	$10^{-6}$ пикс	микропиксель	мкпикс	μpel
$10^9$ пикс	гигапиксель	Гпикс	Gpel	$10^{-9}$ пикс	нанопиксель	нпикс	npel
$10^{12}$ пикс	терапиксель	Тпикс	Tpel	$10^{-12}$ пикс	пикопиксель	ппикс	ppel
$10^{15}$ пикс	петапиксель	Ппикс	Ppel	$10^{-15}$ пикс	фемтопиксель	фпикс	fpel
$10^{18}$ пикс	эксапиксель	Эпикс	Epel	$10^{-18}$ пикс	аттопиксель	апикс	apel
$10^{21}$ пикс	зеттапиксель	Зпикс	Zpel	$10^{-21}$ пикс	зептопиксель	зпикс	zpel
$10^{24}$ пикс	иоттапиксель	Ипикс	Ypel	$10^{-24}$ пикс	иоктопиксель	ипикс	ypel



# ПОНЯТИЕ ПИКСЕЛА

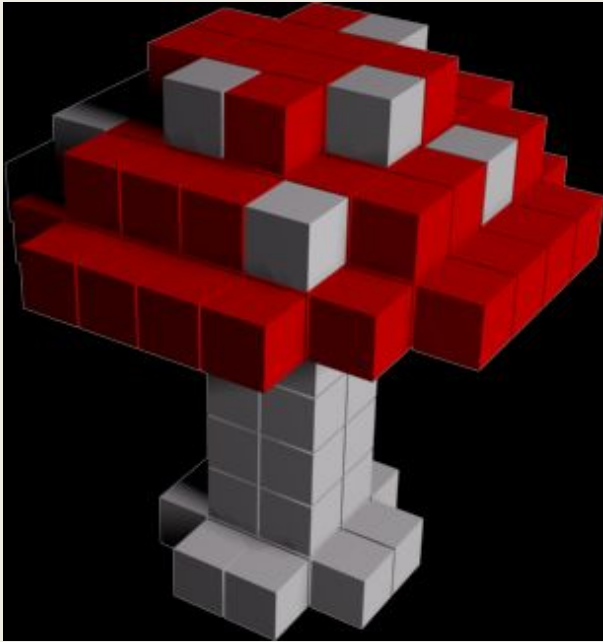


Соответственно, при разрешении 1/72 dpi размер пиксела составит 0,014 дюйма, а при разрешении 1/300 dpi – 0,0033 дюйма.

# ХАРАКТЕРИСТИКИ ПИКСЕЛА

- **ПЛОТНОСТЬ** (DPI) – количество пикселей в одном дюйме изображения.
- **РАЗМЕР** – величина, характеризующая величину пиксела в дюймах, определяемая, как частица от деления единицы на количество пикселей в одном дюйме изображения.
- **ГЛУБИНА ЦВЕТА** – длина двоичного кода, используемого для кодирования цвета пиксела, количество бит на пиксел.
- **МЕСТОПОЛОЖЕНИЕ** - определенные горизонтальные и вертикальные координаты внутри сетки. В большинстве основных программ редактирования изображений можно узнать координаты любого пиксела, поместив над ним инструмент Eyedropper (пипетка). Физические размеры сетки, определяемой общим количеством пикселей и разрешением, задают относительное положение пикселей.

# ПОНЯТИЕ ВОКСЕЛ



# ИСПОЛЬЗОВАНИЕ ВОКСЕЛОВ

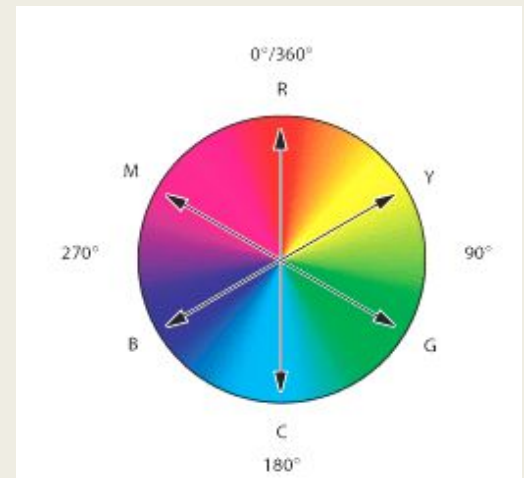
- Игры Command & Conquer: Tiberian Sun и Command & Conquer: Red Alert 2 от компании Westwood Studios использовали воксельные модели транспортных средств.
- Игра Outcast использовала воксели для прорисовки объектов.
- Игра Rising World - игра с процедурно-генерируемым миром, которая представляет собой песочницу (с различными режимами, включая выживание), построенная на воксельной графике. Благодаря этому можно полностью изменять ландшафт (копать пещеры, строить, наращивать горы и создавать водоемы).
- Игра Вангеры использовала большие многоуровневые непрерывные воксельные пространства с изменяемыми «живыми» ландшафтами.
- Воксельные объекты используются в шутерах Shadow Warrior и Blood, а также в переработанном Duke Nukem 3D High Resolution Pack.
- Master of Orion III использует воксельную графику для отображения космических битв и солнечных систем.
- В игровом движке CryEngine 2, который использовался в играх Crysis, Crysis Warhead и Crysis Wars, воксели использовались для построения пустых пространств под поверхностью ландшафта уровня.
- Игры Worms 3D и Worms 4: Mayhem использовали «поксели» (англ. poxel, образовано от voxel (воксел) + polygon (полигон)) для динамически разрушаемого трёхмерного ландшафта, подобного ландшафту в двухмерных версиях.
- Игра Minecraft использует подобие воксельной графики для создания случайно генерируемого ландшафта, но рендеринг производит с помощью полигонов.
- Игра Voxelstein 3D использует воксельный движок «Voxlap», благодаря которому вся геометрия игры построена на вокселях и полностью разрушаема.
- Игра Cube World использует воксельную графику для создания случайно генерируемого игрового мира, моделей игроков и NPC.

# Законы Грассмана

Квантование (кодирование) цвета базируется на *математическом описании цвета*, которое опирается на тот факт, что цвета можно *измерять и сравнивать*.

Цвета – это характеристики реальных объектов, а колориметрические законы Грассмана устанавливают общие свойства математических моделей света.

Любому цвету можно поставить в соответствие некоторую точку трехмерного пространства. Абсолютно черному телу всегда соответствует точка (0.0.0)





# ДЕЛЕНИЕ РАСТРОВОГО ИЗОБРАЖЕНИЯ ПО ФОТОМЕТРИЧЕСКОМУ


# ДЕЛЕНИЕ РАСТРОВОГО ИЗОБРАЖЕНИЯ ПО ФОТОМЕТРИЧЕСКОМУ

■ В полутонных изображениях пиксел полутонного изображения может кодироваться различным количеством бит, что определяет количество возможных полутонов (уровней яркости).

- Например:
- 2 бит — 4 полутона;
- 3 бит — 8 полутонов;
- 4 бит — 16 полутонов;
- 8 бит — 256 полутонов.
- При аэрофотосъёмке и космической съёмке на выходе могут получать полутонные изображения с глубиной цвета (количеством бит на пиксел) 16 или 32 бита.
- Полутонные изображения используются преимущественно для хранения материалов дистанционного зондирования.

# КОДИРОВАНИЕ ЦВЕТА ПИКСЕЛА



$$2^i = N$$

$N$  – количество цветов  
 $i$  – **глубина цвета**, длина двоичного кода, который используется для кодирования цвета пикселя

Яркость цветов			Цвет	Код
Красный	Зеленый	Синий		
0	0	0	черный	000
0	0	1	синий	001
0	1	0	зеленый	010
0	1	1	голубой	011
1	0	0	красный	100
1	0	1	пурпурный	101
1	1	0	желтый	110
1	1	1	белый	111



$$\begin{array}{l|l} N = 8 & 2^i = N \\ \hline i - ? & \end{array}$$

Ответ: 3 бит.

$$2^i = 8 \Rightarrow 2^i = 2^3 \Rightarrow i = 3(\text{бит})$$

Цветные изображения имеют широкий диапазон яркости от 16 цветов до 24 млн. и более.

# КОДИРОВАНИЕ ЦВЕТА ПИКСЕЛА



$$2^i = N$$

$N$  – количество цветов  
 $i$  – **глубина цвета**, длина двоичного кода, который используется для кодирования цвета пикселя

Яркость цветов			Цвет	Код
Красный	Зеленый	Синий		
0	0	0	черный	000
0	0	1	синий	001
0	1	0	зеленый	010
0	1	1	голубой	011
1	0	0	красный	100
1	0	1	пурпурный	101
1	1	0	желтый	110
1	1	1	белый	111

Глубина цвета	Количество цветов в палитре
<u>8</u>	$2^8 = 256$
<u>16</u>	$2^{16} = \underline{65\ 356}$
<u>24</u>	$2^{24} = 16\ 777\ 216$

# Битовая глубина (цветовое

От глубины цвета зависит количество отображаемых цветов, которое может быть вычислено по формуле:

$$N=2^k,$$

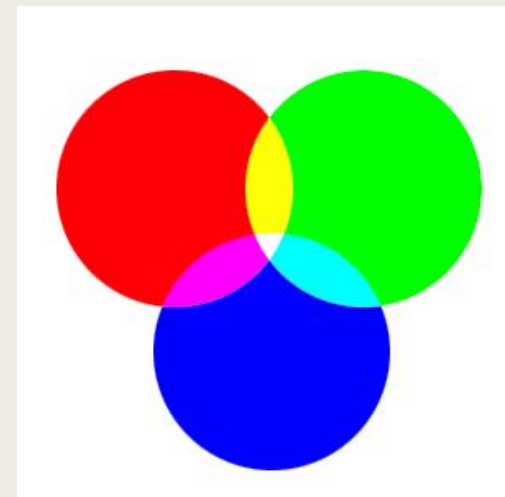
где  $N$  – количество отображаемых цветов,

$k$  – глубина цвета.

Наиболее распространенными значениями глубины цвета являются 4, 8, 16 или 24 бита на точку.

<i>Глубина цвета, <math>k</math> (бит)</i>	<i>Количество отображаемых цветов, <math>N</math></i>
1 (монохромная)	$2^1 = 2$
3	$2^3 = 8$
4	$2^4 = 16$
8	$2^8 = 256$
16 (High Color)	$2^{16} = 65\,536$
24 (True Color)	$2^{24} = 16\,777\,216$

# 8 основных цветовых комбинаций



Цвет	R	G	B	Цвет	R	G	B
Черный 	0	0	0	Красный 	1	0	0
Синий 	0	0	1	Фиолетовый 	1	0	1
Зеленый 	0	1	0	Коричневый 	1	1	0
Голубой 	0	1	1	Белый 	1	1	1

# Кодирование 16-цветной палитры

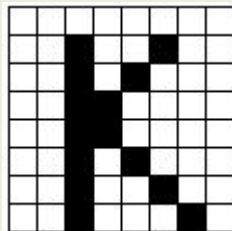
Цвет	Яркость	R	G	B		Цвет	Яркость	R	G	B	
Черный	0	0	0	0		Серый	1	0	0	0	
Синий	0	0	0	1		Светло-синий	1	0	0	1	
Зеленый	0	0	1	0		Светло-зеленый	1	0	1	0	
Голубой	0	0	1	1		Светло-голубой	1	0	1	1	
Красный	0	1	0	0		Светло-красный	1	1	0	0	
Фиолетовый	0	1	0	1		Светло-фиолетовый	1	1	0	1	
Коричневый	0	1	1	0		Желтый	1	1	1	0	
Белый	0	1	1	1		Ярко-белый	1	1	1	1	

# Кодирование основных цветов при глубине цвета 24 бит ( RGB)

Цвет	Интенсивность		
	R	G	B
Черный 	00000000	00000000	00000000
Синий 	00000000	00000000	11111111
Зеленый 	00000000	11111111	00000000
Голубой 	00000000	11111111	11111111
Красный 	11111111	00000000	00000000
Фиолетовый 	11111111	00000000	11111111
Коричневый 	11111111	11111111	00000000
Белый 	11111111	11111111	11111111



# Битовая карта изображения

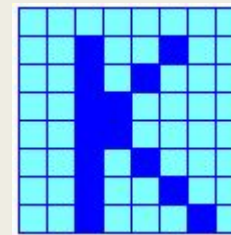


**Битовая карта черно-белого  
изображения будет выглядеть так:**

```
00000000
00100100
00101000
00110000
00110000
00101000
00100100
00100010
```

**Информационный объем изображения**

$$I=8*8*1(\text{бит})=64 \text{ бита}=8 \text{ байт}$$



**Битовая карта при трехбитном кодировании  
изображения будет выглядеть так:**

```
011 011 011 011 011 011 011 011
011 011 001 011 011 001 011 011
011 011 001 011 001 011 011 011
011 011 001 001 011 011 011 011
011 011 001 001 011 011 011 011
011 011 001 011 001 011 011 011
011 011 001 011 011 001 011 011
011 011 001 011 011 011 001 011
```


**Информационный объем изображения**

$$I=8*8*3(\text{бит})=192 \text{ бита}=24 \text{ байт}$$

# ДЕЛЕНИЕ РАСТРОВОГО ИЗОБРАЖЕНИЯ ПО


# **ЦВЕТОВЫЕ МОДЕЛИ RGB, CMY, CMYK, HSV**

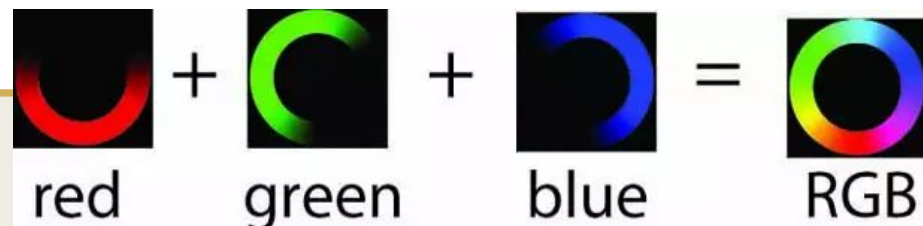
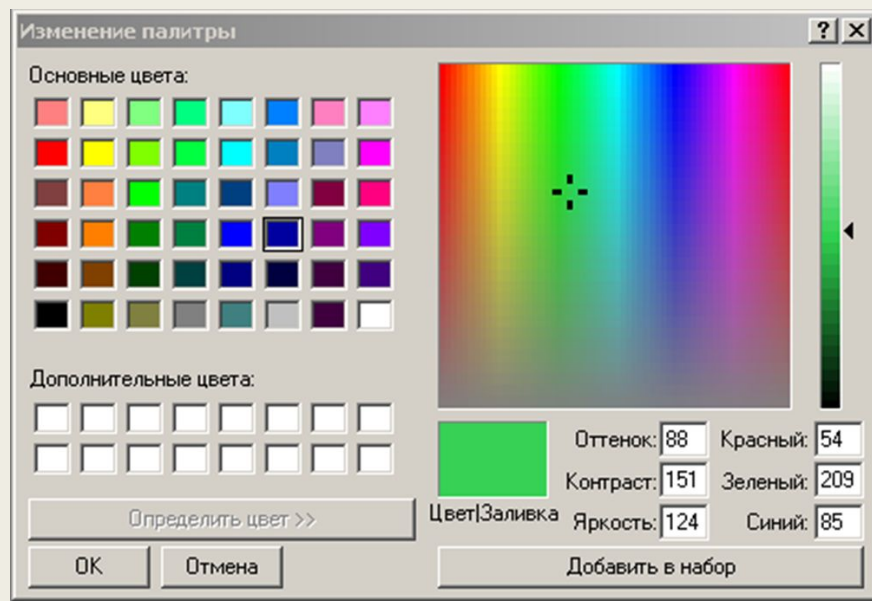
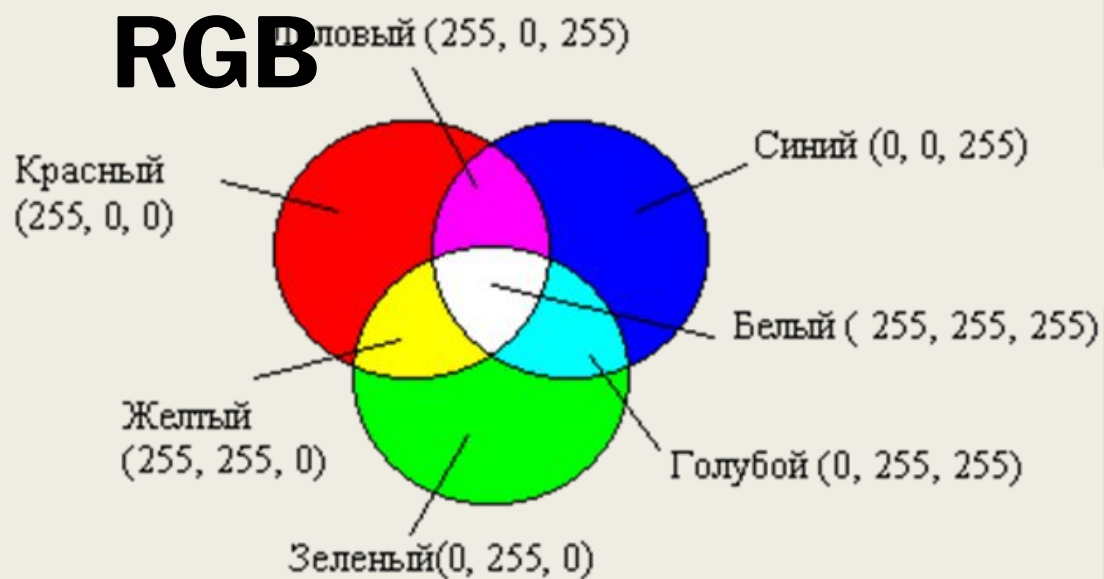
**Способ разделения цвета на составляющие компоненты называется ЦВЕТОВОЙ МОДЕЛЬЮ.**



**Цветовой тон** - качество цвета, соответствующее одному из спектральных цветов (красный, зеленый, синий и др.).

Для монохроматических цветов он выражается длиной волны соответствующего излучения. Из спектра можно выделить значительно больше цветовых тонов, чем существует их названий. Исключение составляют пурпурные цвета, отсутствующие в спектре. Они образуются из смеси крайних спектральных цветов - фиолетового и красного.

# Цветовая модель RGB



# Цветовая модель

Любая точка куба  $(r, g, b)$  определяет какой-то цвет;

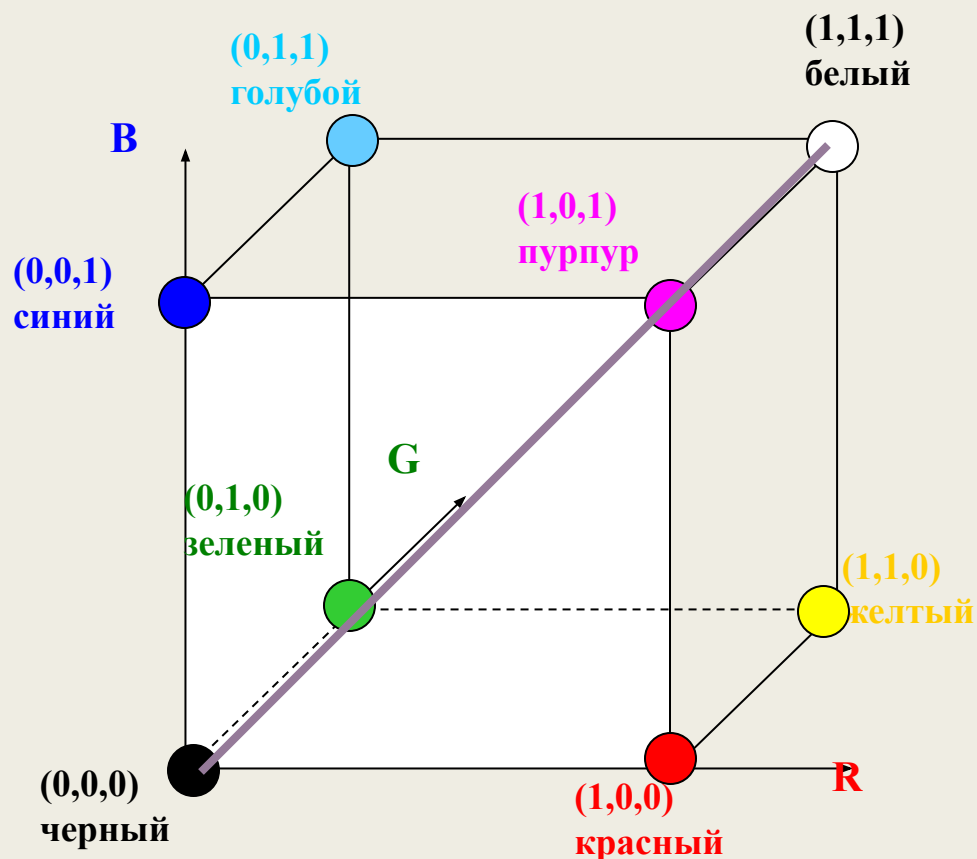
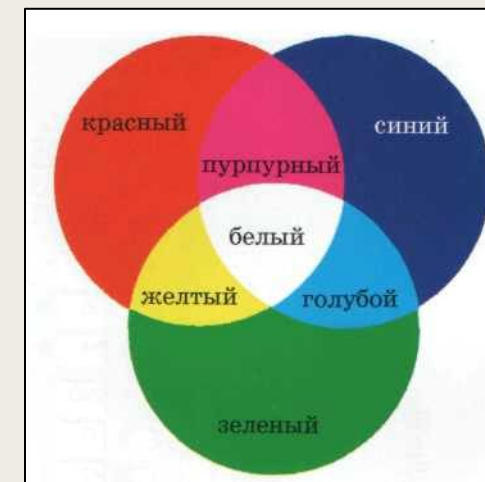
**RGB**

Линия  $(0,0,0) - (1,1,1)$  описывает все градации серого от черного до белого;

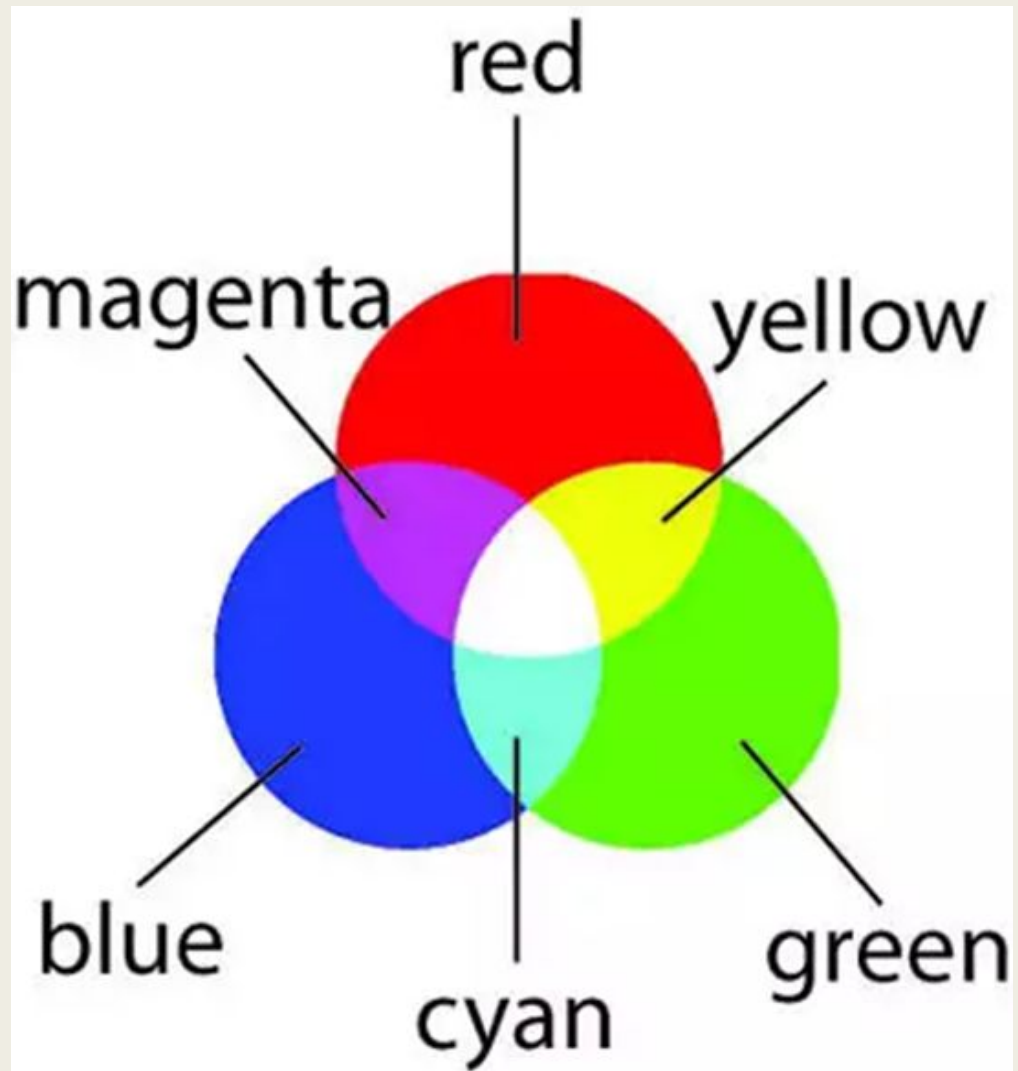
На гранях куба расположены самые насыщенные цвета;

Чем ближе точка к главной диагонали, тем менее насыщен соответствующий цвет;

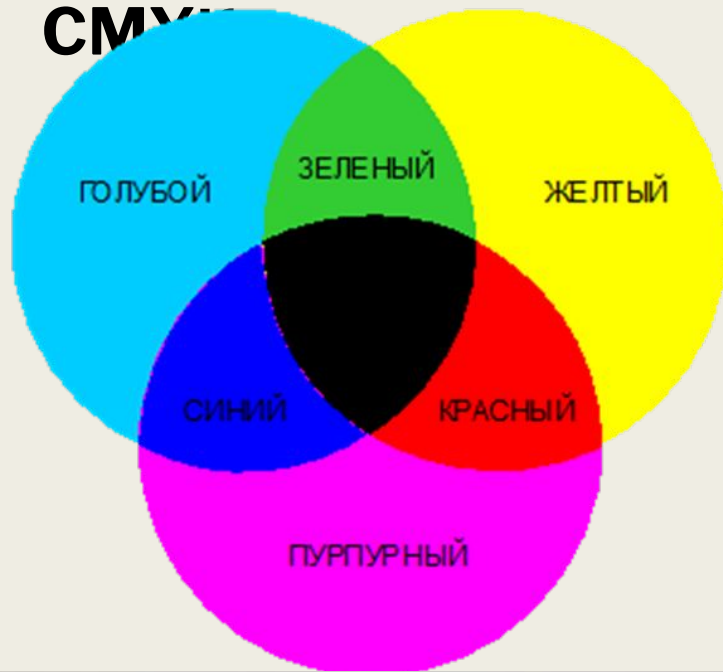
Если все три координаты точки  $(r, g, b)$  ненулевые, то цвет ненасыщенный, причем, наименьшее значение определяет долю серого оттенка, а разность значений – тон и долю насыщенного цветового оттенка.



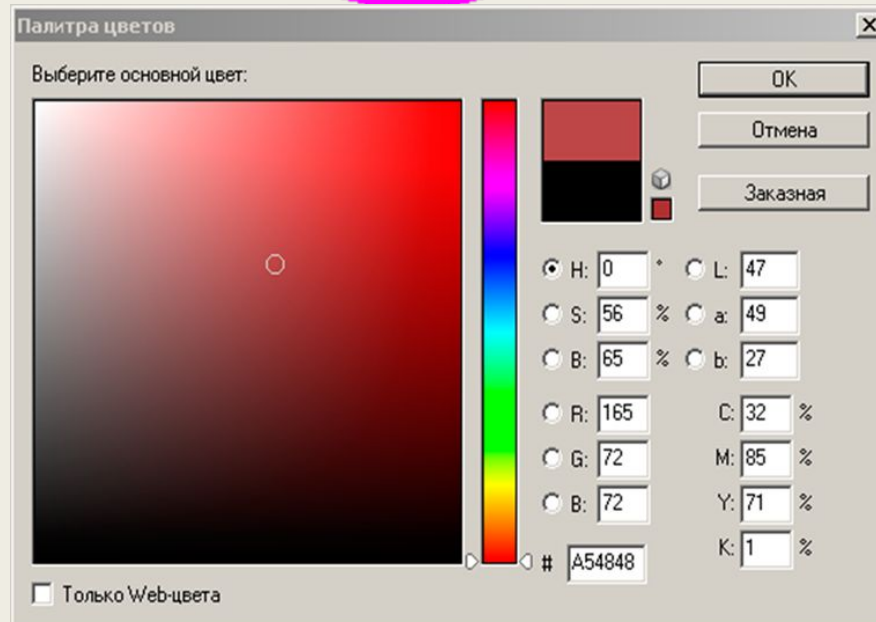
# Цветовая модель CMY



# Цветовая модель CMYK



Цветовыми составляющими этой модели являются цвета: голубой (Cyan), лиловый (Magenta), желтый (Yellow) и черный (Black). Эти цвета получаются в результате вычитания основных цветов модели RGB из белого цвета. Черный цвет задается отдельно.

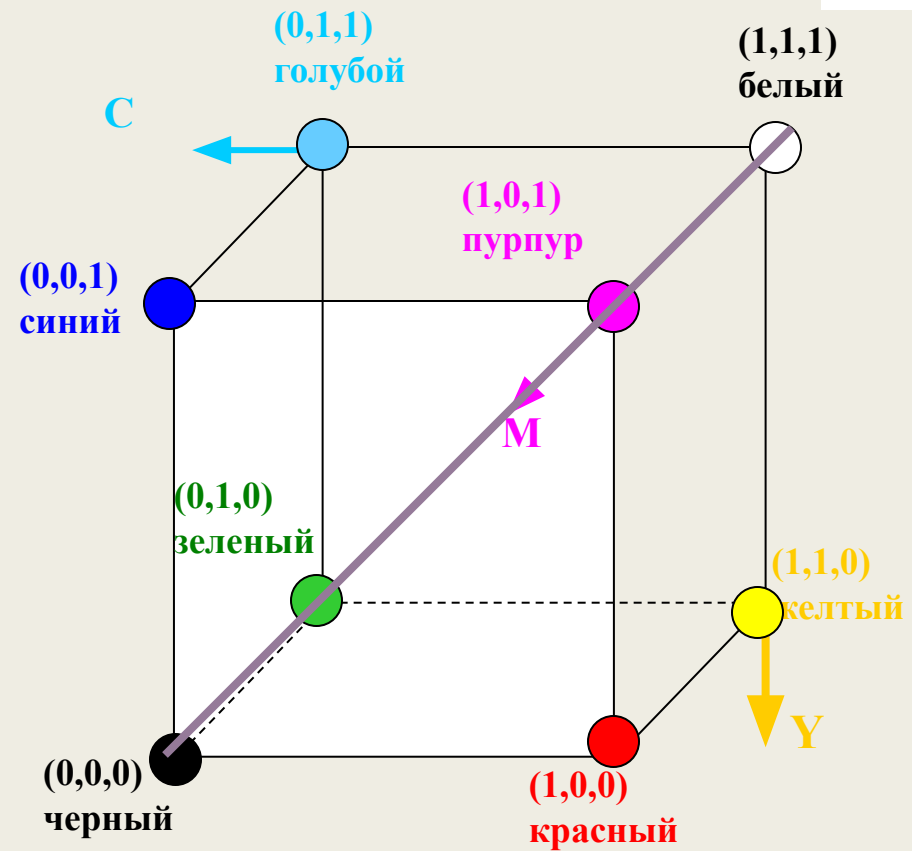
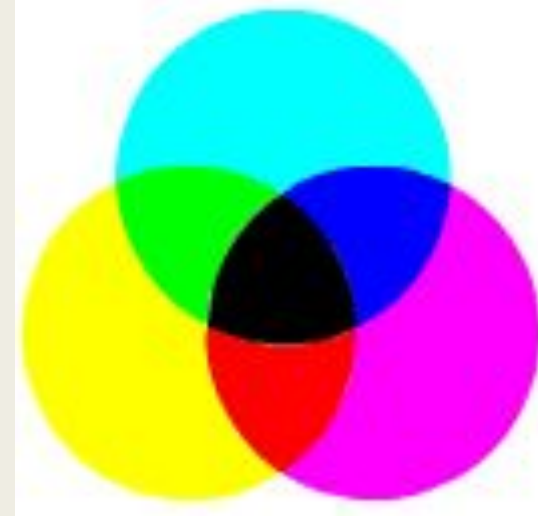


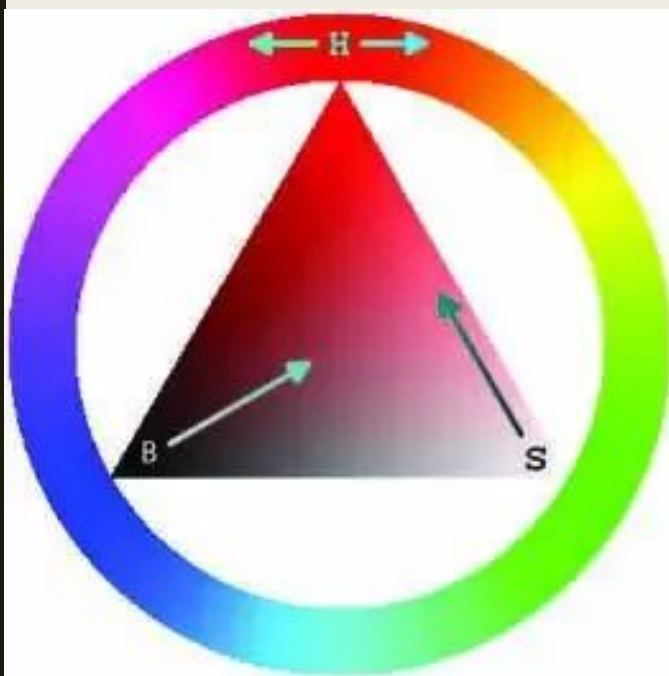


# Цветовая модель

CMYK

Diagram showing five empty rectangular boxes, each preceded by a circle, representing a sequence of color components or steps in a process.





## Цветовая модель HSB

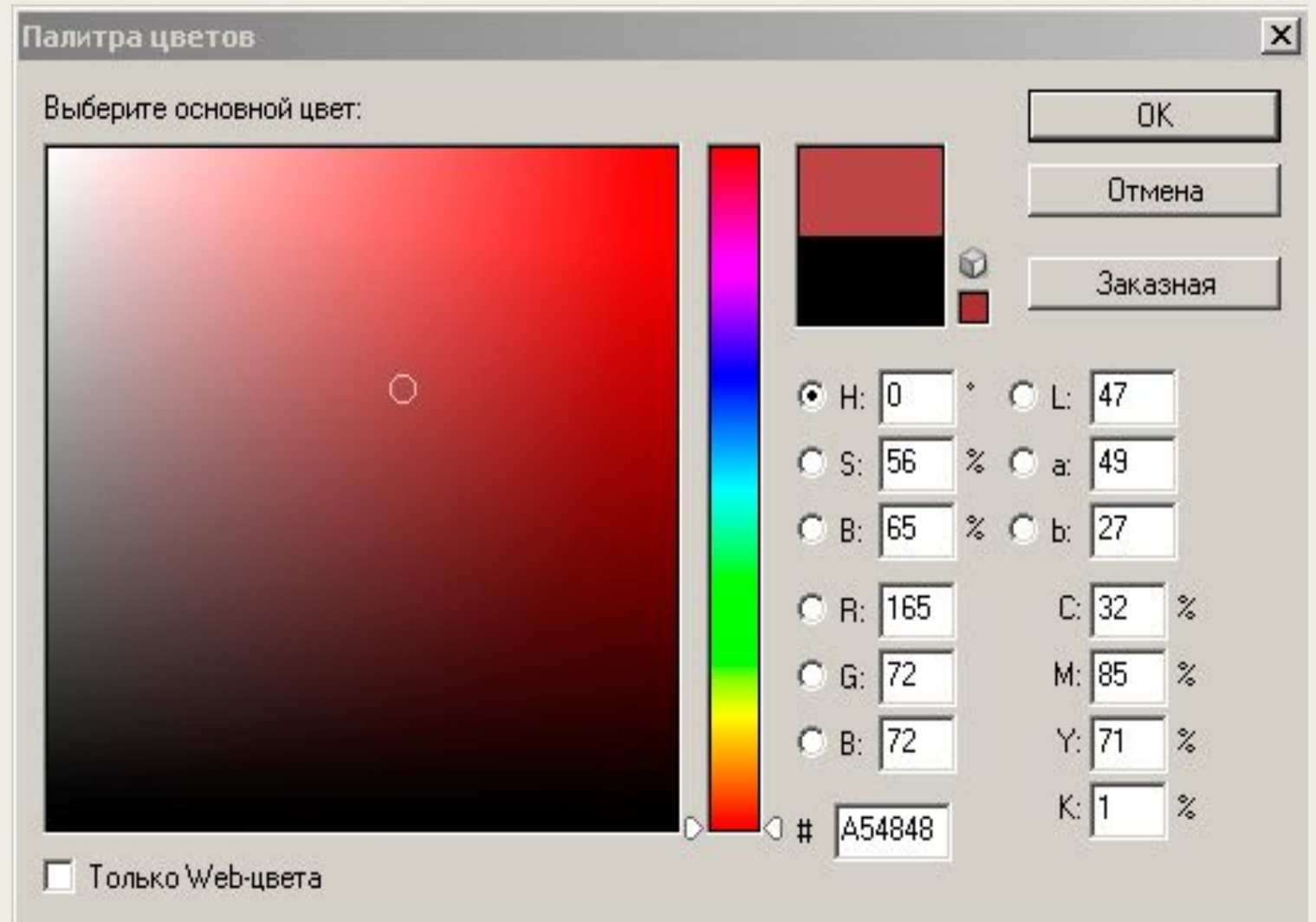
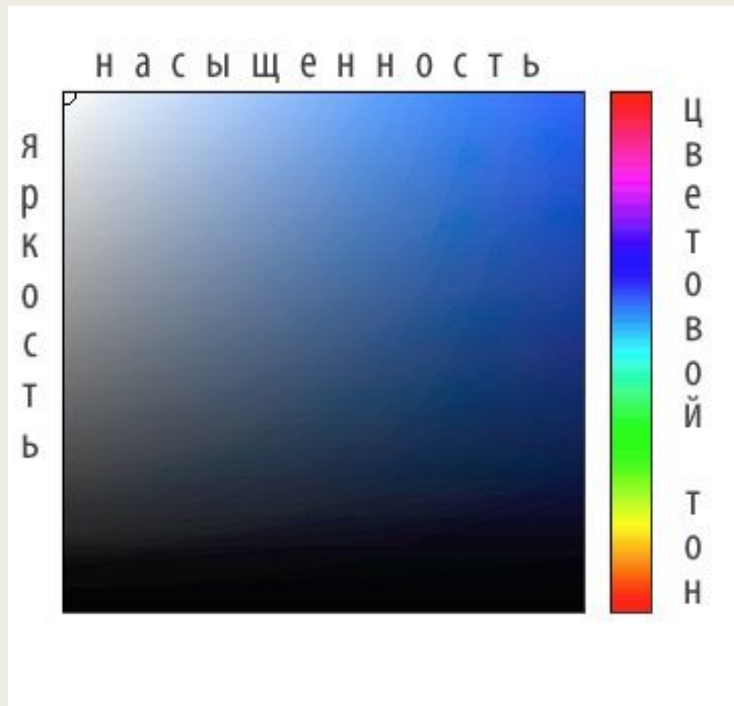
Параметр **Hue** определяется градусами от 0 до 360, исходя из цветов радужного кольца.

Параметр **Saturation** - процент добавления к основному цвету белой краски, имеет значение от 0% до 100%.

Параметр **Brightness** - процент добавления черной краски, также изменяется от 0% до 100%.

# Цветовая модель HSB

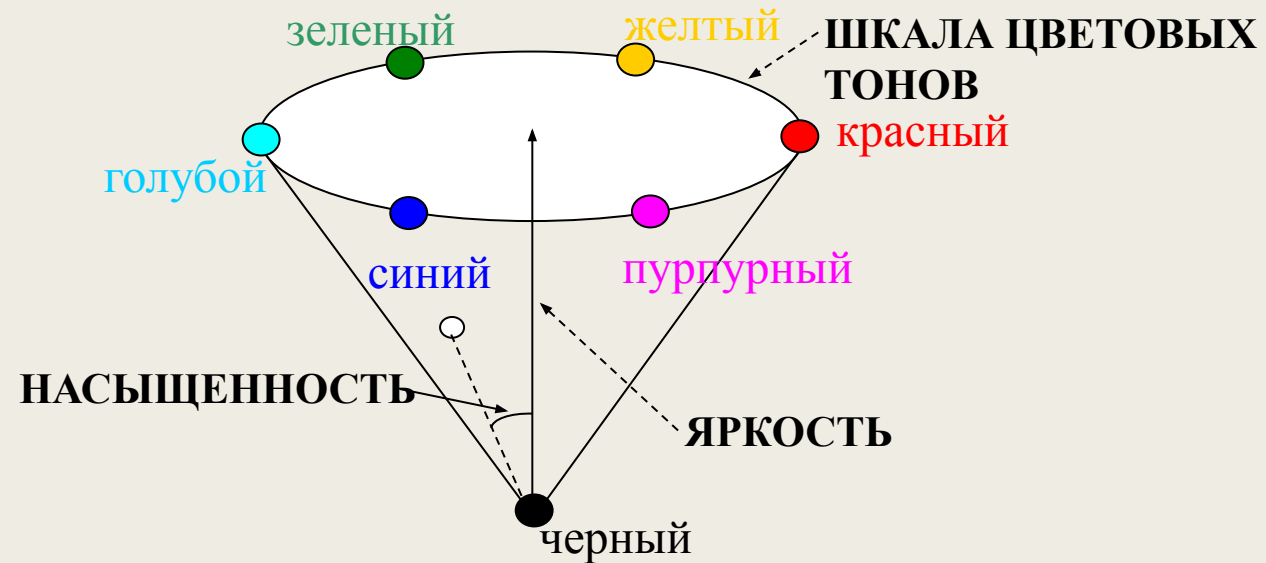
(Hue-Saturation-Brightness)  
(цветовой тон-насыщенность-яркость)



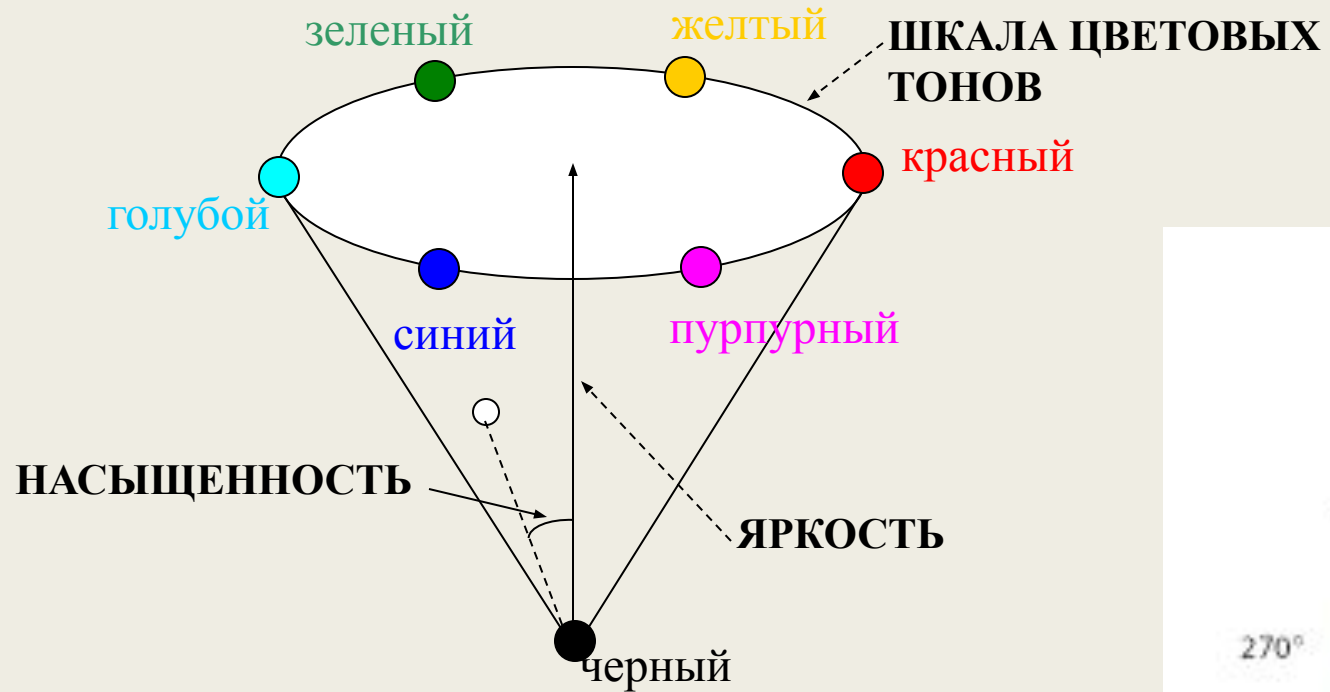
# Цветовая модель HSB

(Hue-Saturation-Brightness)

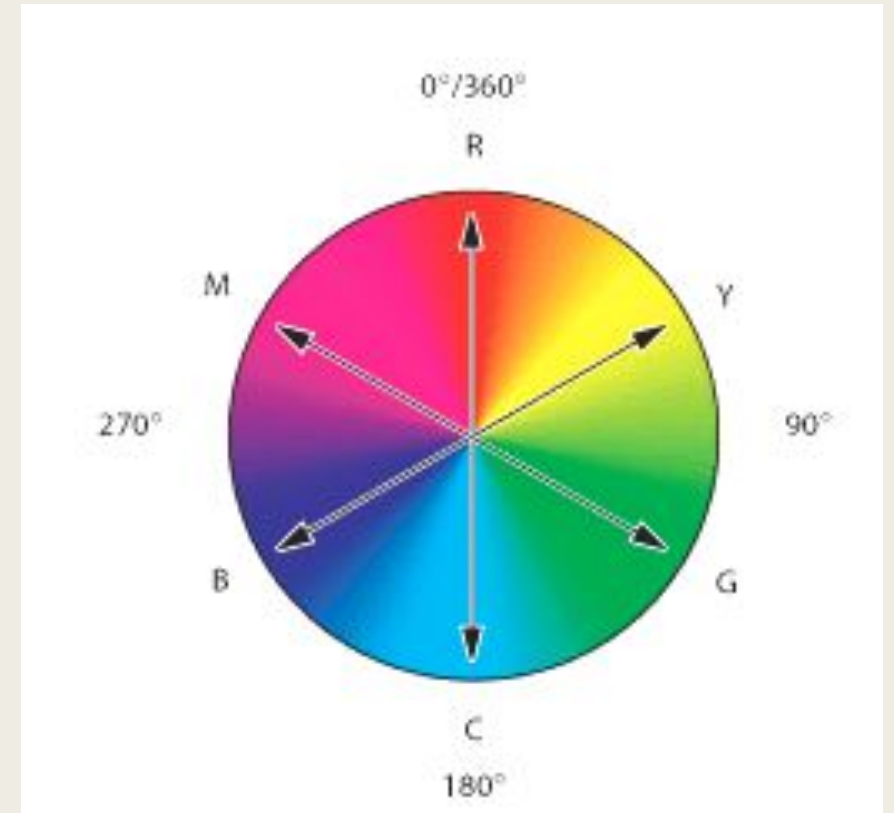
(цветовой тон-насыщенность-яркость)

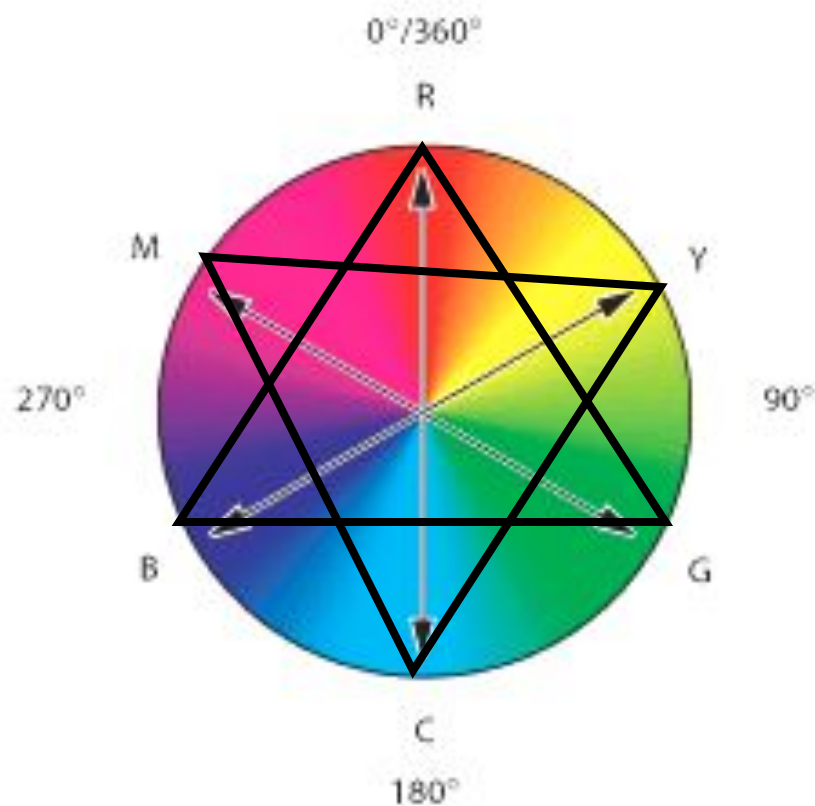
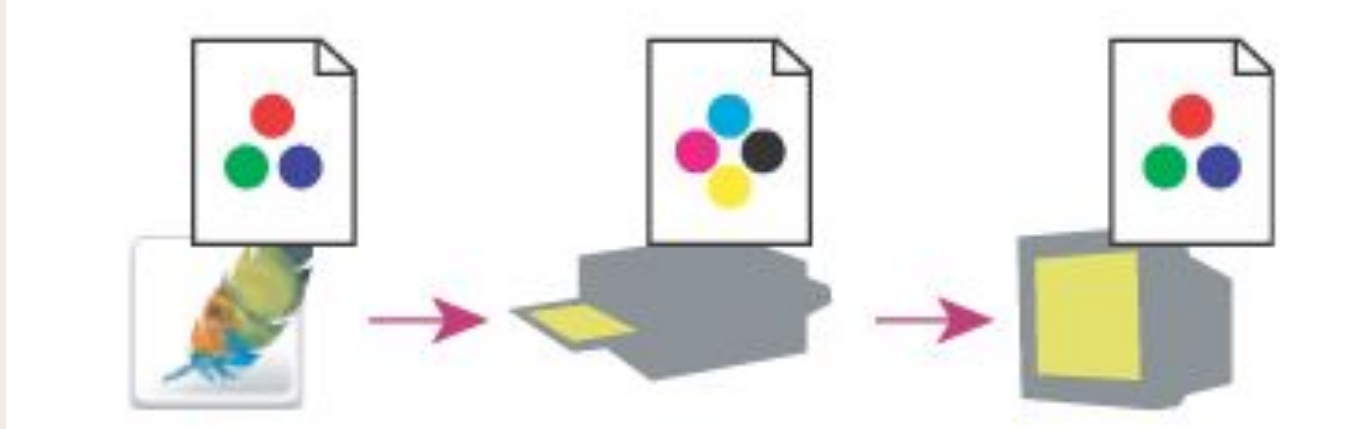


# Цветовое пространство модели HSB



Разрез конуса в плоскости  
постоянной яркости





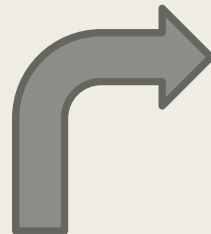
**В цветовом пространстве HSB хорошо видна связь между моделями RGB и CMYK: на цветовом круге основные цвета одной модели расположены точно напротив основных цветов другой модели;**

**Цвета модели RGB, которые не попадают в треугольник, полученный соединением основных цветов, в RGB модели будут непредставимы;**

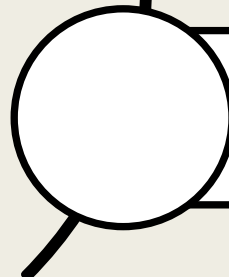
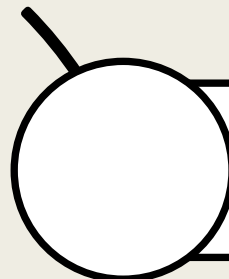
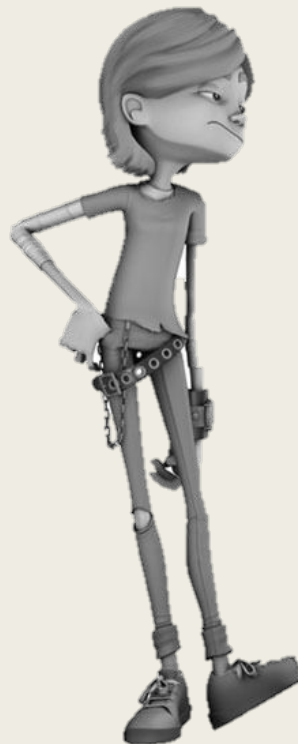
**Аналогично – для модели CMYK**

# **АЛГОРИТМЫ СЖАТИЯ РАСТРОВОЙ ИНФОРМАЦИИ. ПОНЯТИЕ О ФОРМАТАХ ФАЙЛОВ ДЛЯ ХРАНЕНИЯ РАСТРОВЫХ ИЗОБРАЖЕНИЙ (TIFF, BMP, JPEG, GIF, PSX)**

# ЗАЧЕМ НЕОБХОДИМО СЖАТИЕ?



И ПОСЛЕ  
СЖАТИЯ





# МЕТОДЫ СЖАТИЯ РАСТРОВЫХ ДАННЫХ

Методы сжатия без потерь дают более низкий коэффициент сжатия, но зато сохраняют точное значение пикселей исходного изображения.

- **СЖАТИЕ БЕЗ ПОТЕРЬ**


Методы сжатия с потерями дают более высокие коэффициенты сжатия, но не позволяют воспроизвести первоначальное изображение с точностью до пиксела.

- **СЖАТИЕ С ПОТЕРЯМИ**


# ЛЕКСОГРАФИЧЕСКИЙ КОД ИЛИ ГРУППОВОЕ КОЛИРОВАНИЕ

A	A	A	A
A	B	B	B
A	A	B	B
A	A	A	B

AAAA ABVB AABV AAAB 16 байт  
5A 3B 2A 2B 3A 1B 12 байт

Если с помощью группового кодирования сжать изображение, в котором встречается подряд 100 пикселей с нулевым значением и последовательность из 100 нулей закодировать парой чисел (100,0), то размер изображения сократится в 50 раз.

# Как работает алгоритм группового кодирования

- Если внимательно посмотреть на растровое изображение, то можно увидеть, что пиксели одного цвета часто оказываются рядом друг с другом. Если начать с левого верхнего угла изображения и исследовать пиксели каждой строки, выписывая последовательно их значения слева направо, то можно заметить, что картинка состоит из множества отрезков, в которых повторяется одно и то же число. **Количество пикселей в отрезке называют длиной отрезка.**
- Начиная с первой строки, программа группового кодирования просматривает значения пикселей слева направо и ищет отрезки повторяющихся пикселей. Всякий раз, когда встречаются три или более идущих подряд пикселей с одинаковым значением, программа заменяет их парой чисел: **первое число указывает длину отрезка, второе - значение пикселей.** Число, определяющее длину отрезка, называют **меткой отрезка.**
- Чтобы идентифицировать серии неповторяющихся значений пикселей, программа также вставляет метки, указывающие количество таких значений в серии. Зарезервированный бит необходим для того, чтобы можно было отличить метку отрезка, от метки серии неповторяющихся значений. Например, в 8-ми битах можно специфицировать последовательности длиной до 127 пикселей (максимальное число, представимое 7-ю битами); восьмой бит в каждой метке может отличать отрезок от серии неповторяющихся пикселей. Точно так же обрабатывается каждая строка пикселей, и отрезки одинаковых значений пикселей сжимаются во всем изображении.
- Графическая программа декодирует изображение, считывая сжатый файл и восстанавливая отрезки повторяющихся значений пикселей. Заметим, что восстановленное изображение полностью совпадает с оригиналом.



**до сжатия**

**AAAA BBAB ABAB ABVB ABAА = 20 бит**

**после сжатия**

**4A 2B 1A 1B 1A 1B 1A 1B 1A 3B 1A 1B 2A = 26  
бит**

# МЕТОД СЖАТИЯ РАСТРОВЫХ ДАННЫХ

**LZW**

- алгоритм Лемпеля — Зива — Велча (Lempel-Ziv-Welch, LZW) — универсальный алгоритм сжатия данных без потерь

- Используется в файлах формата TIFF, PDF, GIF, PostScript и других, а также отчасти во многих популярных программах сжатия данных (ZIP, ARJ, LHA).
- Процесс сжатия выглядит следующим образом: последовательно считываются символы входного потока и происходит проверка, существует ли в созданной таблице строк такая строка. Если такая строка существует, считывается следующий символ, а если строка не существует, то в поток заносится код для предыдущей найденной строки, строка заносится в таблицу, а поиск начинается снова.
- Для декодирования на вход подается только закодированный текст, поскольку алгоритм LZW может воссоздать соответствующую таблицу преобразования непосредственно по закодированному тексту. Алгоритм генерирует однозначно декодируемый код за счет того, что каждый раз, когда генерируется новый код, новая строка добавляется в таблицу строк. LZW постоянно проверяет, является ли строка уже известной, и, если так, выводит существующий код без генерации нового. Таким образом, каждая строка будет храниться в единственном экземпляре и иметь свой уникальный номер. Следовательно, при декодировании во время получения нового кода генерируется новая строка, а при получении уже известного, строка извлекается из словаря.

# МЕТОД СЖАТИЯ РАСТРОВЫХ ДАННЫХ

## АЛГОРИТМ КОДИРОВАНИЯ

- Начало.
- Шаг 1. Все возможные символы заносятся в словарь. Во входную фразу ХХ заносится первый символ сообщения.
- Шаг 2. Считать очередной символ YY из сообщения.
- Шаг 3. Если YY — это символ конца сообщения, то выдать код для ХХ, иначе:
- Если фраза ХУХУ уже имеется в словаре, то присвоить входной фразе значение ХУХУ и перейти к Шагу 2 ,
- Иначе выдать код для входной фразы ХХ, добавить ХУХУ в словарь и присвоить входной фразе значение YY. Перейти к Шагу 2.
- Конец.

## АЛГОРИТМ ДЕКОДИРОВАНИЯ

- Начало.
- Шаг 1. Все возможные символы заносятся в словарь. Во входную фразу ХХ заносится первый код декодируемого сообщения.
- Шаг 2. Считать очередной код YY из сообщения.
- Шаг 3. Если YY — это конец сообщения, то выдать символ, соответствующий коду ХХ, иначе:
- Если фразы под кодом ХУХУ нет в словаре, вывести фразу, соответствующую коду ХХ, а фразу с кодом ХУХУ занести в словарь.
- Иначе присвоить входной фразе код ХУХУ и перейти к Шагу 2 .
- Конец.

# ПРИМЕР LZW-СЖАТИЯ ДАННЫХ

- **Шаг 1:** Тогда, согласно изложенному выше алгоритму, мы добавим к изначально пустой строке aa и проверим, есть ли строка aa в таблице. Поскольку мы при инициализации занесли в таблицу все строки из одного символа, то строка aa есть в таблице.
- **Шаг 2:** Далее мы читаем следующий символ bb из входного потока и проверяем, есть ли строка abab в таблице. Такой строки в таблице пока нет.
- Добавляем в таблицу  $\langle 5 \rangle \langle 5 \rangle$  abab. В поток:  $\langle 0 \rangle \langle 0 \rangle$ ;
- **Шаг 3:** baba — нет. В таблицу:  $\langle 6 \rangle \langle 6 \rangle$  baba. В поток:  $\langle 1 \rangle \langle 1 \rangle$ ;
- **Шаг 4:** acac — нет. В таблицу:  $\langle 7 \rangle \langle 7 \rangle$  acac. В поток:  $\langle 0 \rangle \langle 0 \rangle$ ;
- **Шаг 5:** caca — нет. В таблицу:  $\langle 8 \rangle \langle 8 \rangle$  caca. В поток:  $\langle 2 \rangle \langle 2 \rangle$ ;
- **Шаг 6:** abab — есть в таблице; abaaba — нет. В таблицу:  $\langle 9 \rangle \langle 9 \rangle$  abaaba. В поток:  $\langle 5 \rangle \langle 5 \rangle$ ;
- **Шаг 7:** adad — нет. В таблицу:  $\langle 10 \rangle \langle 10 \rangle$  adad. В поток:  $\langle 0 \rangle \langle 0 \rangle$ ;
- **Шаг 8:** dada — нет. В таблицу:  $\langle 11 \rangle \langle 11 \rangle$  dada. В поток:  $\langle 3 \rangle \langle 3 \rangle$ ;
- **Шаг 9:** abaaba — есть в таблице; abacabac — нет. В таблицу:  $\langle 12 \rangle \langle 12 \rangle$  abacabac. В поток:  $\langle 9 \rangle \langle 9 \rangle$ ;
- **Шаг 10:** caca — есть в таблице; cabcab — нет. В таблицу:  $\langle 13 \rangle \langle 13 \rangle$  cabcab. В поток:  $\langle 8 \rangle \langle 8 \rangle$ ;
- **Шаг 11:** baba — есть в таблице; baebae — нет. В таблицу:  $\langle 14 \rangle \langle 14 \rangle$  baebae. В поток:  $\langle 6 \rangle \langle 6 \rangle$ ;
- **Шаг 12:** И, наконец последняя строка ee, за ней идет конец сообщения, поэтому мы просто выводим в поток  $\langle 4 \rangle \langle 4 \rangle$ .

Символ	Битовый код	Код
a	000	0
b	001	1
c	010	2
d	011	3
e	100	4

кущая строка	Текущий символ	Следующий символ	Вывод		Словарь
			Код	Биты	
ab	a	b	0	000	5: ab
ba	b	a	1	001	6: ba
ac	a	c	0	000	7: ac
ca	c	a	2	010	8: ca
ab	a	b	-	-	- -
aba	b	a	5	0101	9: aba
ad	a	d	0	0000	10: ad
da	d	a	3	0011	11: da
ab	a	b	-	-	- -
aba	b	a	-	-	- -
abac	a	c	9	1001	12: abac
ca	c	a	-	-	- -
cab	a	b	8	1000	13: cab
ba	b	a	-	-	- -
bae	a	e	6	0110	14: bae
e	e	-	4	0100	- -

# МЕТОД СЖАТИЯ РАСТРОВЫХ ДАННЫХ

## ■ JPEG

JPEG (произносится "джей-пег") - метод, сжимающий данные с потерями.

Метод получил свое название от аббревиатуры объединенной группы экспертов в области фотографии (Joint Photographic Expert Group - JPEG), которая его и разработала.

- **JPEG широко используется при сжатии статических изображений для их хранения на компакт-дисках. Основная идея метода - разделить информацию в изображении по уровню важности, и затем отбросить менее важную ее часть, уменьшая тем самым общий объем хранимых данных. Это достигается преобразованием матрицы цветовых значений в матрицу амплитуд, которые соответствуют определенным частотам разложения изображения.**
- **JPEG-сжатие отбрасывает часть высокочастотных компонент изображения, оставляя компоненты с низкими частотами.** Человеческий глаз менее чувствителен к высокочастотным вариациям цвета, поскольку общий вид изображения определяется низкими частотами. Значение пиксела, полученное при восстановлении изображения, несколько отличается от исходного значения, так как часть информации была потеряна, хотя обычно они очень близки.
- У метода JPEG есть очень интересная особенность: пользователь может задавать коэффициент качества. Высокий коэффициент качества позволяет сохранить больше деталей, но при этом уменьшается степень сжатия. При низком коэффициенте качества степень сжатия увеличивается, но изображение становится менее четким.



# Как работает алгоритм JPEG

- Прежде всего программа делит изображение на **блоки - матрицы размером 8x8 пикселей**. Поскольку при использовании метода JPEG время, затрачиваемое на сжатие изображения, пропорционально квадрату числа пикселей в блоке, обработка нескольких блоков меньшего размера делается значительно быстрее, чем обработка всего изображения целиком.
- К значениям пикселей применяется формула, названная **дискретным косинусоидальным преобразованием** (Discrete Cosine Transform - DCT). DCT переводит матрицу значений пикселей 8x8 в матрицу значений амплитуд такой же размерности, соответствующую определенным частотам синусоидальных колебаний. Левый верхний угол матрицы соответствует низким частотам, а правый нижний - высоким.
- Коэффициент качества, введенный пользователем, используется в простой формуле, которая генерирует значения элементов другой матрицы 8x8, названной **матрицей квантования**. Чем ниже коэффициент качества, тем большие значения будут иметь элементы матрицы.
- Каждое значение в матрице, получившееся после DCT-преобразования, делится на соответствующее значение из матрицы квантования, затем округляется до ближайшего целого числа. Так как большие числа находятся в правой нижней половине матрицы квантования, то основная часть высокочастотной информации изображения будет отброшена. Поэтому нижняя правая часть матрицы пикселей будет состоять в основном из нулей.
- Далее программа, двигаясь по матрице зигзагообразно, считывает элементы матрицы и кодирует их последовательно методами без потерь. Заметим, что сжатие существенно зависит от нулей в правой нижней половине матрицы. **Чем ниже коэффициент качества, тем больше нулей в матрице и, следовательно, тем выше степень сжатия.**
- Декодирование JPEG-изображения начинается с шага обратного кодированию без потерь, в результате чего восстанавливается матрица квантования пикселей.
- Значения из матрицы пикселей умножаются на значения из матрицы квантования, чтобы восстановить, насколько это возможно, матрицу, которая была вычислена на шаге применения DCT. На этапе квантования была потеряна некоторая часть информации, поэтому числа в матрице будут близки к первоначальным, но не будет абсолютного совпадения.
- Обратная к DCT формула (IDCT) применяется к матрице для восстановления значений пикселей исходного изображения. Еще раз отметим, что полученные цвета не будут полностью соответствовать первоначальному из-за потери информации на шаге квантования. Восстановленное изображение, при сравнении с оригиналом, будет выглядеть несколько размытым и обесцвеченным.

# ФОРМАТ ФАЙЛА ИЛИ ДАННЫХ


- **BMP** - Bit Map — битовая карта. В этом формате первоначально использовалось простейшее кодирование — по пикселям (самое неэкономное), которые обходились последовательно по строкам, начиная с нижнего левого угла графического изображения.
- Файлы этого формата входили в первые версии Windows. В этом формате использовалось только 256 цветов, т.е. пиксел представлялся только одним байтом.
- В дальнейшем формат стал использоваться и для сохранения полноцветных изображений. Формат BMP — один из стандартных форматов растровой графики.

# ФОРМАТ BMP



## Характеристики формата:

- **Поддержка сжатия:** есть возможность использования сжатия без потерь (Run Length Encoding, RLE)
- **Поддержка прозрачности:** отсутствует
- **Поддержка анимации:** отсутствует
- Поддержка чересстрочного отображения: отсутствует
- **Количество цветов:** модель RGB, глубина цвета 24 бита
- **Совместимость с другими платформами:** только Windows

- **TIFF (Taged Image File Format)** — стандартный формат в топографической графике и издательских системах. Файлы в формате TIFF обеспечивают лучшее качество печати. Из-за большого размера, данный формат не применяется при создании Web-сайтов и публикации в Интернет.
- Формат TIFF относится к числу наиболее универсальных и распространенных форматов растровой графики. Он создавался в качестве межплатформенного универсального формата для цветных изображений. Работа с ним поддерживается почти всеми программами для работы с точечной графикой. Может хранить графику в монохромном виде, в RGB и CMYK цветовых представлениях.
- Формат включает в себя внутреннюю компрессию. Он имеет открытую архитектуру — предусмотрена возможность объявления в заголовке сведений о типе изображения, т.е. его версии могут быть использованы в дальнейшем для представления новых разработок. В формате сохраняется и сопроводительная информация передаваемых изображений (подписи и пр.). Изображения в формате TIFF хранятся в файлах с расширением .tif.

# ФОРМАТ TIFF



## Характеристики формата:

- **Поддержка сжатия:** поддерживает возможность применения различных алгоритмов сжатия (в зависимости от самого сохраняемого изображения)
- **Поддержка прозрачности:** отсутствует
- **Поддержка анимации:** отсутствует
- **Поддержка черезстрочного отображения:** отсутствует
- Количество цветов: 8, 16, 32 и 64 бит на канал (модели Lab, RGB и CMYK)
- **Совместимость с другими платформами:** все

- **GIF (Graphic Interchange Format)** — формат обмена графическими данными, который служит для записи и хранения растровых графических изображений. Этот формат отличается от других форматов растровой графики тем, что он долгое время поддерживается в Интернете. Использует индексированные цвета (ограниченный набор цветов). Это один из самых распространенных форматов картинок, распространяемых в Интернет и применяемых при создании Web-сайтов.
- Изображения в GIF формате хранятся в файлах с расширением .gif. К преимуществам GIF изображения относится то, что вид изображения не зависит от браузера и платформы. Лучше всего отображаются рисунки, чертежи и изображения с небольшим количеством однородных цветов, прозрачные изображения и анимационные последовательности (эта очень известная особенность данного формата графики). В GIF-изображениях используется сжатие без потери информации.

# ФОРМАТ GIF



## Характеристики формата:

- **Поддержка сжатия:** без потерь (Lempel-Ziv-Welch, LZW)
- **Поддержка прозрачности:** Да
- **Поддержка анимации:** Да (отличительная особенность данного формата)
- **Поддержка чересстрочного отображения:** Да (interlaced)
- **Количество цветов:** индексированные цвета (256 цветов)

- **JPEG (Joint Photographic Expert Group)** — предназначен для хранения изображений со сжатием.
- JPEG практически является стандартом для хранения полноцветных изображений. Формат JPEG был создан для того, чтобы избавиться от ограничений, которые налагались на изображения, созданные в GIF-формате.
- Формат широко используется в документах HTML и для передачи данных по сети. Сохраняет параметры графики в цветовом представлении RGB. Изображения в JPEG формате хранятся в файлах с расширением .jpg.
- Программы, работающие с JPEG, используют алгоритмы сжатия с потерей информации, они исключают из изображения те данные, которые считаются несущественными. Перед применением алгоритма сжатия изображения делится на прямоугольные области. При сжатии есть риск получить нечеткое, размытое изображение с искажением деталей.

# ФОРМАТ JPEG



## Характеристики формата:

- **Поддержка сжатия:** сжатие с потерями
- **Поддержка прозрачности:** отсутствует
- **Поддержка анимации:** отсутствует
- Поддержка черезстрочного отображения: да (Progressive JPEG)
- **Количество цветов:** True Color (модели RGB и CMYK)



# ФОРМАТ JPEG2000

- Данный формат был разработан для замены JPEG.
- При сохранении изображения с одинаковым уровнем сжатия изображения, сохраненные в формате JPEG 2000, получаются более четкими и занимают меньше места на диске.
- В этом формате решена проблема с появлением дефектов JPEG, которые появились при сохранении с большим коэффициентом сжатия.
- Поддержка формата реализована не во всех браузерах, что сильно мешает его распространению.
- Обеспечивает хранение информации о цветовых пространствах, метаданных и информации для согласованного доступа в сетевых приложениях, взаимодействующих с помощью протокола JPEG Part 9 JPIP.

Используется для хранения файлов с 3D-видео.



# ФОРМАТ PNG

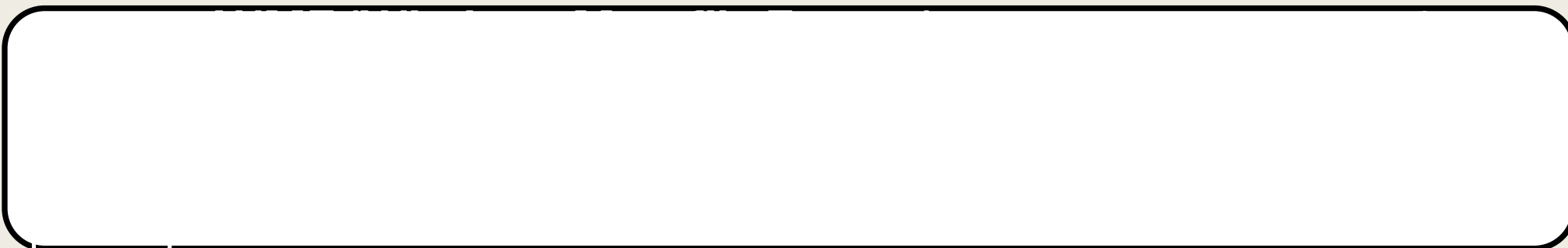


- **PNG (portable network graphics)** — формат хранения растровой графики, использующий сжатие без потерь. PNG — это свободный формат (в отличие от GIF), поэтому получил широкое распространение.
- Очень мощный и широко применяемый формат в Интернете и других областях компьютерной графики.

## Характеристики формата:

- **Поддержка сжатия:** сжатие без потерь (Deflate)
- **Поддержка прозрачности:** Да
- **Поддержка анимации:** Да (APNG)
- **Поддержка черезстрочного отображения:** Да (two-dimensional interlacing)
- **Количество цветов:** Deep Color
- **Совместимость с другими платформами:**

# ФОРМАТЫ WMF, PSD, CDR

An empty white rounded rectangular box with a black border, intended for content.An empty white rounded rectangular box with a black border, intended for content.An empty white rounded rectangular box with a black border, intended for content.

# СВОЙСТВА ФОРМАТОВ РАСТРОВОЙ ГРАФИКИ

Формат	Макс. число бит/пиксел	Макс. число цветов	Макс. размер изображения, пиксел	Методы сжатия	Кодирование нескольких изображений
BMP	24	16 777 216	65535 x 65535	RLE	-
GIF	8	256	65535 x 65535	LZW	+
JPEG	24	16 777 216	65535 x 65535	JPEG	-
PCX	24	16 777 216	65535 x 65535	RLE	-
TIFF	24	16 777 216	4 294 967 295	LZW, RLE и другие	+

# **ПОДДЕРЖКА РАСТРОВЫХ ДАННЫХ В ГИС. РАБОТА С РАСТРАМИ БОЛЬШОЙ РАЗМЕРНОСТИ**

# ПОДДЕРЖКА РАСТРОВЫХ ДАННЫХ

## Примеры таких задач:

- Прямое чтение большинства форматов растровых данных
- Визуализация многоканальных изображений
- Привязка изображений, в том числе полуавтоматическая
- Классификация изображений
- Хранение изображений в корпоративных базах данных
- Хранение изображений в мозаиках и каталогах растров
- Обработка изображений “на лету”
- Конвертация растровых данных в векторные
- Публикация изображений в виде сервисов
- Возможность публикации сервисов с обработкой данных по запросу

Растровые данные хранятся одним из

следующих способов

# Отображение растровых данных в GIS

- Растровые изображения хранятся в цифровом виде и для их просмотра необходимо выбрать определенный способ визуализации.
- GIS поддерживают как двухмерную, так и трехмерную визуализацию растровых данных.
- Пользователь может добавлять изображения в различных комбинациях каналов, улучшать читаемость изображений за счет изменения параметров растяжки гистограммы, регулировать прозрачность слоев.

## Хранение растровых данных в GIS

- GIS поддерживает прямое чтение самых распространенных форматов растровых данных.
- Для более эффективной работы с растрами, пользователь может загрузить их в базу геоданных GIS или в корпоративную базу данных.

# Обработка растровых данных в GIS

- Обработка растровых данных может заключаться как в простой географической привязке, так и в более сложных преобразованиях, например в расчете мозаики, улучшении пространственного разрешения снимков, вычислении индекса NDVI, тематической классификации данных.
- Результаты обработки изображения обычно сохраняются в виде нового файла. Альтернативой данному подходу является обработка данных “на лету”, при которых необходимые вычисления проводятся непосредственно в момент обращения пользователя к данным. Такой подход позволяет экономить дисковое пространство и ускоряет обработку больших объемов данных.

## Публикация растровых данных в GIS

- GIS предоставляют широкие возможности публикации изображений в локальной сети или Internet. Опубликованные изображения становятся доступными как пользователям GIS, имеющим доступ к сети, так и клиентам, использующим веб-приложения. Веб-приложения могут быть созданы на основе отдельных API для Sliverlight, JavaScript и Flex.
- Также сервисы изображений могут быть использованы любыми настольными и веб-приложениями, поддерживающими открытые стандарты WMS и WCS



# РАБОТА С РАСТРАМИ БОЛЬШОЙ РАЗМЕРНОСТИ

- Видео: Работа с изображениями в ArcGIS  
<https://www.youtube.com/watch?v=0ZkwlySjFPw>

# ЗАДАЧА 1

Какой объем видеопамати (I) необходим для хранения четырех страниц изображения при условии, что разрешающая способность дисплея равна 640 X 480 точек, а количество используемых цветов  $N=32$ ?

**Решение.**

1)  $N=2^i$ ,  $32=2^i$ ,  $i=5$  бит – глубина цвета

2)  $I = 640 \text{ точек} * 480 \text{ точек} * 5 \text{ бит} * 4 \text{ страницы} = 6144000 \text{ бит}$   
 $= 750 \text{ Кбайт}$  (1 Кбайт = 1024 байта = 8192 бит)

**Ответ: 750 Кбайт**

# ЗАДАЧА 2

256-цветный рисунок содержит 1 Кбайт информации (I). Из скольких точек он состоит?

**Решение.**

- 1)  $N=2^i$ ,  $256=2^i$ ,  $i=8$  бит — глубина цвета или информационный объем одной точки;
- 2) 1 Кбайт =  $1024 * 8$  бит = 8192 бит - объем изображения;
- 3)  $8192 \text{ бит} : 8 \text{ бит} = 1024$  точек — на изображении

**Ответ: 1024 точки**

# ЗАДАЧА 3

На экране монитора необходимо получить 1024 оттенка серого цвета. Какой должна быть глубина цвета?

**Решение.**

1)  $1024 \sim 10 * 10 * 10$  — по 10 бит приходится на каждую из трех составляющих модели RGB (красную, зеленую, синюю);

2)  $10 * 3 = 30$  бит - глубина цвета;

**Ответ: 30 бит**

# ЗАДАЧА 4

После преобразования графического изображения количество цветов уменьшилось с 256 до 32. Во сколько раз уменьшился объем занимаемой памяти?

**Решение.**

1)  $N_1 = 2^{i_1}$ ,  $256 = 2^8$ ,  $i_1 = 8$  бит — информационный объем одной точки 1-го изображения;

2)  $N_2 = 2^{i_2}$ ,  $32 = 2^5$ ,  $i_2 = 5$  бит — информационный объем одной точки 2-го изображения;

3)  $i_1 / i_2 = 8 / 5 = 1,6$  раза

**Ответ: 1,6 раза**

# ЗАДАЧА 5

Видеопамять имеет объем, в котором может храниться 8-цветное изображение размером 640 X 350 точек. Какого размера изображение можно хранить в том же объеме видеопамяти, если использовать 512-цветную палитру?

## Решение.

- 1)  $N_1 = 2^i$ ,  $8 = 2^i$ ,  $i_1 = 3$  бита – глубина цвета 1-го изображения;
- 2)  $640 * 350 * 3 = 672000$  бит – объем видеопамяти для хранения 1-го изображения;
- 3)  $N_2 = 2^i$ ,  $512 = 2^i$ ,  $i_2 = 9$  бит - информационный объем одной точки 2-го изображения;
- 3)  $672000 / 9 = 74667$  точек – размер 2-го изображения

**Ответ: 74667 точек**