

---

# Лекция 5

## Хранение информации в памяти компьютера (продолжение)

---





---

# План урока:

1. Представление числовой информации в памяти ЭВМ
2. Представление текстовой информации в памяти ЭВМ
3. Представление графической информации в памяти ЭВМ



---

## 2. Представление текстовой информации в памяти ЭВМ

---





**Таблица кодировки** — таблица, содержащая упорядоченный перечень кодируемых символов, в соответствии с которой происходит преобразование символа в его двоичный код и обратно.

Наиболее популярные таблицы кодировки: **ASCII** (N=256) и **Unicode** (N=65535)

Минимальная длина кода (i) для кодирования любого знака из N- символьного алфавита определяется по формуле

$$2^i = N$$



## ASCII-код 1960г

### Стандартная часть

от 00 до 7F ,

- от 00 до 1F – управляющие коды, отданы производителям аппаратных средств
- от 20 (пробел) до 7F – английский алфавит, знаки препинания, цифры, арифметические операции

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	.	р	А	Р	а		Л	л	р	Ё
1	␣	␠	!	1	A	Q	a	q	Б	С	б		Т	т	с	ё
2	␣	␡	!"	2	B	R	b	r	В	Т	в		У	у	т	ё
3	␣	␢	!"#\$	3	C	S	c	s	Г	У	г		Ф	ф	у	ё
4	␣	␣	!"#\$%	4	D	T	d	t	Д	Ф	д		Х	х	ф	ё
5	␣	␤	!"#\$%&	5	E	U	e	u	Е	Х	е		Ц	ц	х	ё
6	␣	␥	!"#\$%&'	6	F	V	f	v	Ж	Ц	ж		Ч	ч	ц	ё
7	␣	␦	!"#\$%&'&	7	G	W	g	w	З	Ч	з		Ш	ш	ч	ё
8	␣	␧	!"#\$%&'&()	8	H	X	h	x	И	Ш	и		Щ	щ	ш	ё
9	␣	␨	!"#\$%&'&()&	9	I	Y	i	y	Й	Щ	й		Ъ	ъ	щ	ё
A	␣	␩	!"#\$%&'&()*	A	J	Z	j	z	К	Ъ	к		Ы	ы	ъ	ё
B	␣	␪	!"#\$%&'&()*&	B	K	[	k	{	П	Ы	п		Ь	ь	ы	ё
C	␣	␫	!"#\$%&'&()*&	C	L	\	l		М	Ь	м		Э	э	ь	ё
D	␣	␬	!"#\$%&'&()*&	D	M	^	m	~	Н	Э	н		Ю	ю	э	ё
E	␣	␭	!"#\$%&'&()*&	E	N	␣	n	␣	О	Ю	о		Я	я	ю	ё
F	␣	␮	!"#\$%&'&()*&	F	O	␣	o	␣	␣	␣	␣		␣	␣	␣	␣

### Расширенная часть

от 80 до FF

Символы национальных алфавитов, символы псевдографики, некоторые научные символы



## Кодировка КОИ-8

128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ

## Кодировка Windows-1251

128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Á	à	,	è	”	…	†	‡	€	‰	É	<	й	Й	ó	ú
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
â	·	’	“	”	•	–	—	è	™	é	>	ò	й	ó	ú
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
nbsp	û	Ы	Э	и	ы	і	ѕ	Ё	Ө	Ю	«	¬	shy	Ө	Я
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
•	±	Ы	Э	’	µ	¶	•	ё	№	ю	»	э	ю	я	я
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

# Unicode (1993г)

Снимает ограничение однобайтовых таблиц – невозможность одновременного использования нескольких языков

В Unicode на кодирование символов отводится 32 бит (сколько символов может быть закодировано в этой таблице?)

- 0-127 совпадают с ASCII,
- далее алфавиты современных языков (до 65536)

В современных компьютерах и операционных системах используется укороченная, 16-битовая версия Unicode



---

## Кодирование и декодирование с помощью таблицы ASCII

- 1. Зашифруйте текст «Принтер», используя таблицу ASCII-кодов.*
  - 2. Дешифруйте текст «42 69 6E 61 72 79», используя таблицу ASCII-кодов.*
  - 3. Используя объемный подход, посчитайте количество информации в сообщении: «Теоретические основы информатики».*  
*(Символы закодированы с помощью таблицы ASCII)*
-

---

# 3. Представление графической информации

---



# Общие подходы к представлению в компьютере информации естественного происхождения

Для преобразования «естественной» информации в дискретную форму ее подвергают *дискретизации* и *квантованию*



# Дискретизация

- **Дискретизацией** (англ. *discretisation*) называют процедуру устранения временной и/или пространственной непрерывности естественных сигналов, являющихся носителями информации



# Пространственная дискретизация

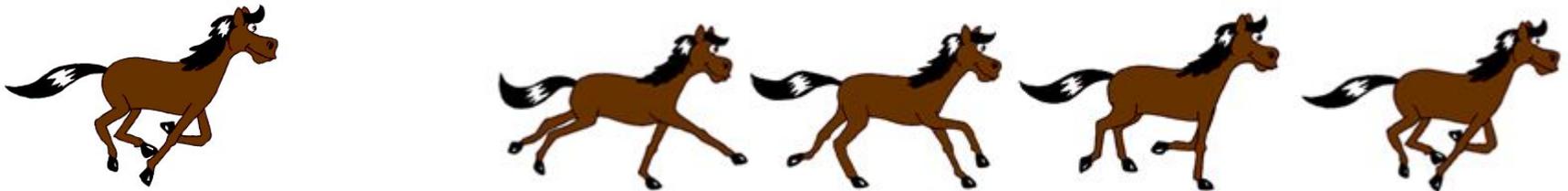
- При *пространственной дискретизации* изображения его разбивают на небольшие области, в пределах которых характеристики изображения считают неизменными.



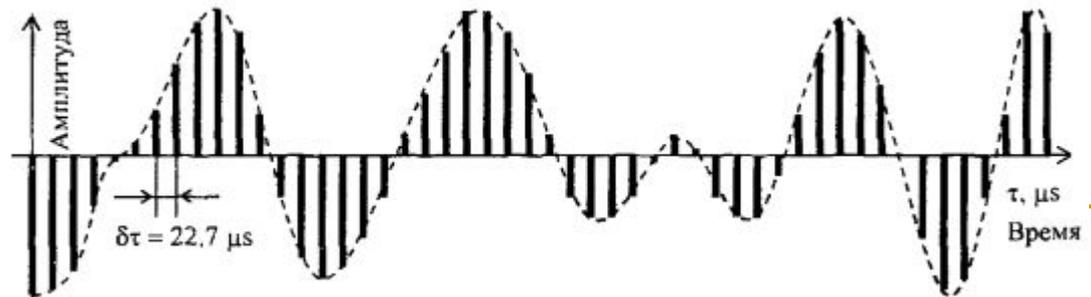
# Временная дискретизация

- При *временной дискретизации* время разбивают на небольшие интервалы, в пределах которых характеристики природных сигналов, как и в пространственном случае, считают неизменными

- **Пример.** Временная дискретизация движущегося изображения



- **Пример.** Временная дискретизация звукового сигнала



---

# Квантование

- **Квантованием** (англ. quantisation) называют процедуру преобразования непрерывного диапазона всех возможных входных значений измеряемой величины в дискретный набор выходных значений



# Пример. Пример квантования цветовых оттенков серого цвета

	100%
	83%
	67%
	50%
	33%
	17%
	0%

1) Диапазон возможных значений измеряемой величины разбивается на несколько поддиапазонов.

2) При измерении определяется поддиапазон, в который попадает значение, и в компьютере сохраняется только **номер поддиапазона**.

Пусть яркость серого оттенка составляет 70%. Это значение попадает в поддиапазон 4(67% - 83%), поэтому в компьютере этот оттенок серого будет закодирован числом **4**.



- Графическое изображение – совокупность световых сигналов на плоскости
- Можно измерять (выражать в числах) местоположение, цветовой оттенок и яркость каждого сигнала
- Каждая характеристика может принимать любое из бесконечного множества значений в некотором диапазоне – все их невозможно представить в компьютере
- Поэтому все измеряемые непрерывные характеристики подвергают **дискретизации и квантованию**



---

**Дискретизация** – выделение в непрерывном объекте **конечного числа элементов**, информация о которых будет сохранена в компьютере.

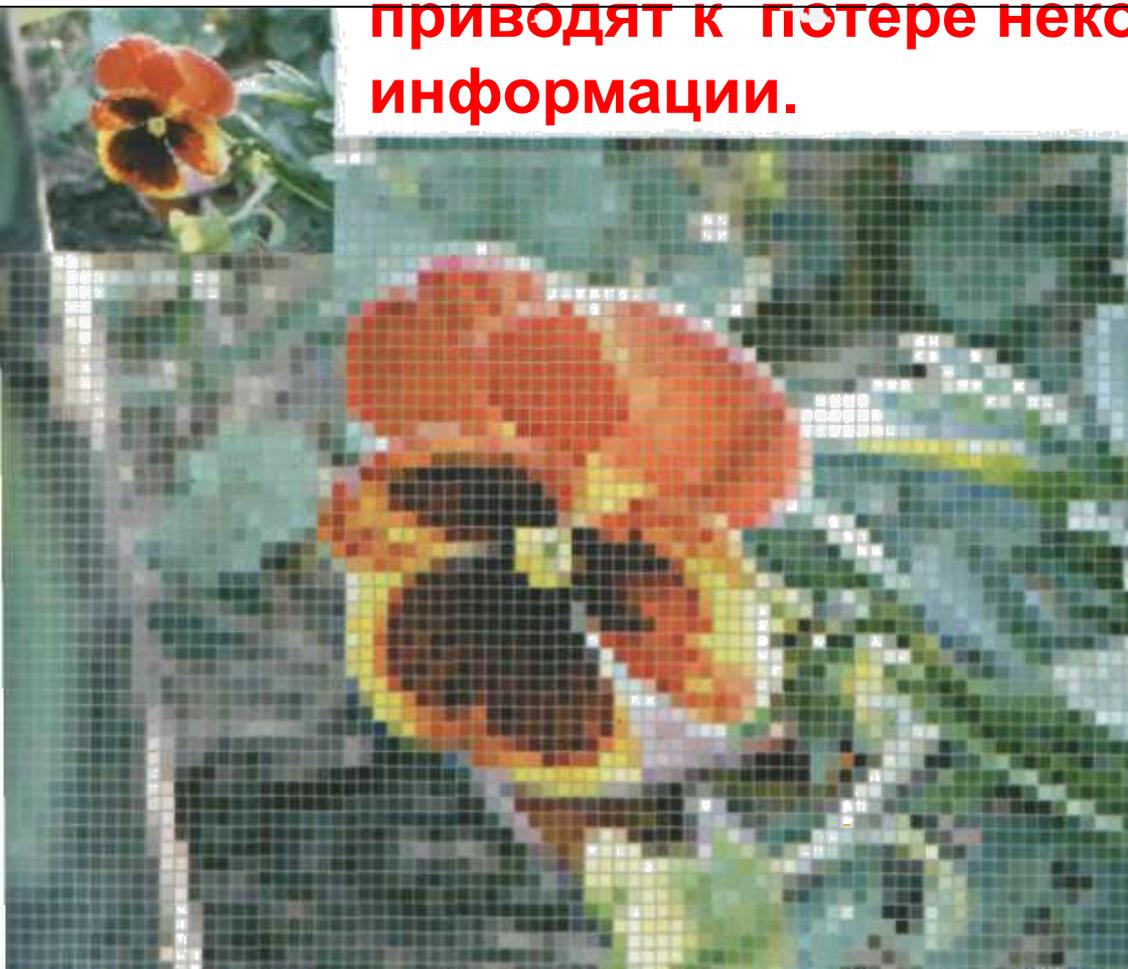
Информация об остальных элементах будет *утеряна!*

**Квантование** - формирование **чисел**, характеризующих эти элементы



# Выводы

**Дискретизация и квантование всегда приводят к потере некоторой доли информации.**



Компьютерное изображение живописного произведения , цифровая запись музыкального произведения всегда отличаются от оригиналов в худшую сторону

---

# Пространственная дискретизация изображений



# Векторное и растровое представление графической информации



- Растровое представление можно охарактеризовать как *поточечное представление* изображения
- Векторное — как *структурное представление* изображения

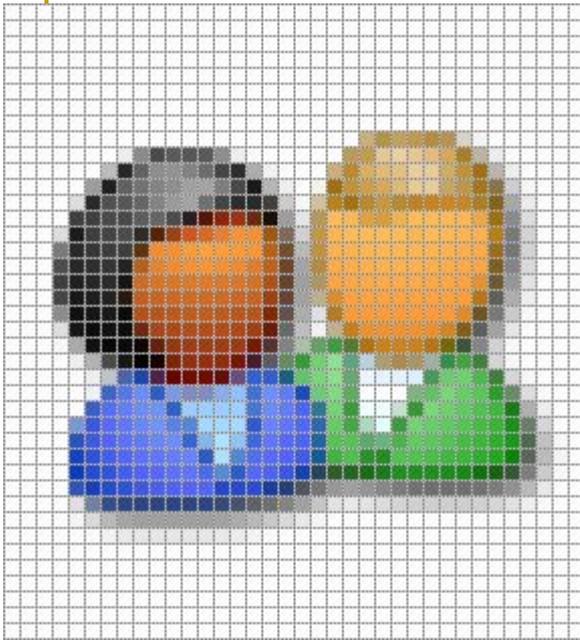
# Растровое представление

При **растровой пространственной дискретизации** графической информации на изображение накладывается сетка (**растр**), каждая ячейка которой (**пиксель**) рассматривается как далее неделимый фрагмент, определяемый набором атрибутов: координатами, формой, размером и цветом.

Процедура разбиения изображения на пиксели называется **растеризацией**, или







**Размер сетки растра** , задаваемый в виде  **$M*N$** ,  
где **M** - число пикселей по горизонтали,  
**N** – число пикселей по вертикали называется **разрешающей способностью** (или **графическим разрешением**) экрана.

**Стандартные значения графического разрешения экрана :**

**640\*480**

**800\*600**

**1024\*768**

**1280\*1024**

**1600\*1200**

**Видеопамять** – оперативная память, хранящая видеоинформацию во время ее воспроизведения в изображение на экране ( может делиться на **страницы**).

**Страница** – раздел видеопамяти, вмещающий информацию об одном образе экрана (одной картинке)

**Графический файл** – файл, хранящий информацию о графическом изображении.

# Векторное представление

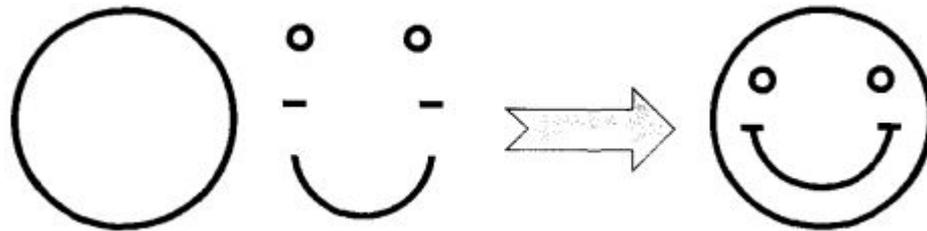
- **Векторная пространственная дискретизация** - это разбиение изображения на набор графических примитивов (*кривая, прямая, окружность и др.*)

Векторное представление изображения соответствует математическому описанию объектов с использованием системы координат



# Векторное представление

- Изображение строится при помощи заранее определенного набора стандартных геометрических фигур



---

# Квантование цвета графических изображений



- 
- 
- **Квантованию** в случае растрового и векторного представления графической информации **подвергается цвет** точек изображения или графических примитивов



# Математические модели цвета

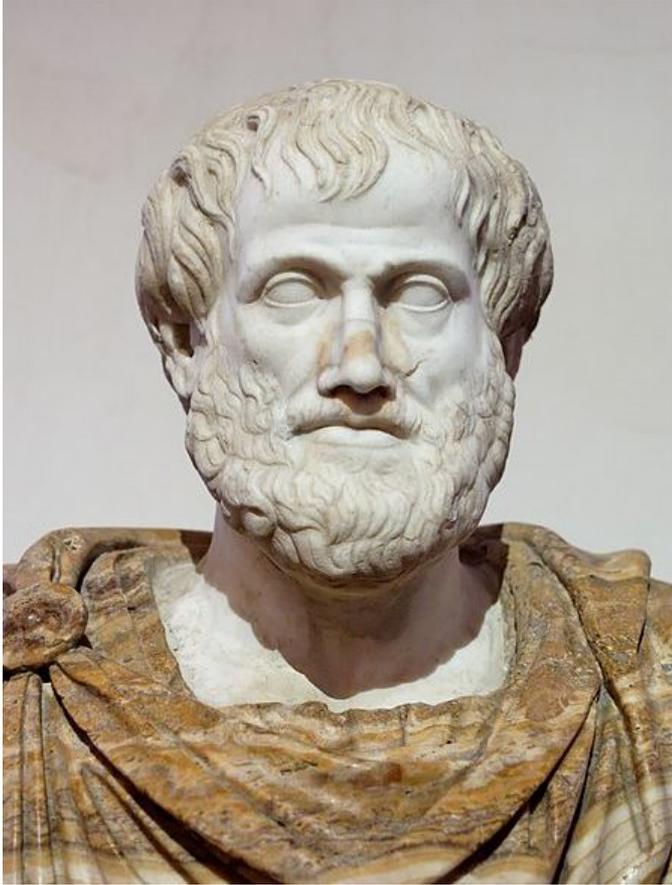


Квантование (кодирование) цвета базируется на **математическом описании цвета**, которое, в свою очередь, опирается на тот факт, что цвета можно *измерять и сравнивать*



Научная дисциплина, изучающая вопросы измерения цветовых характеристик, называется *метрологией цвета*, или **колориметрией**

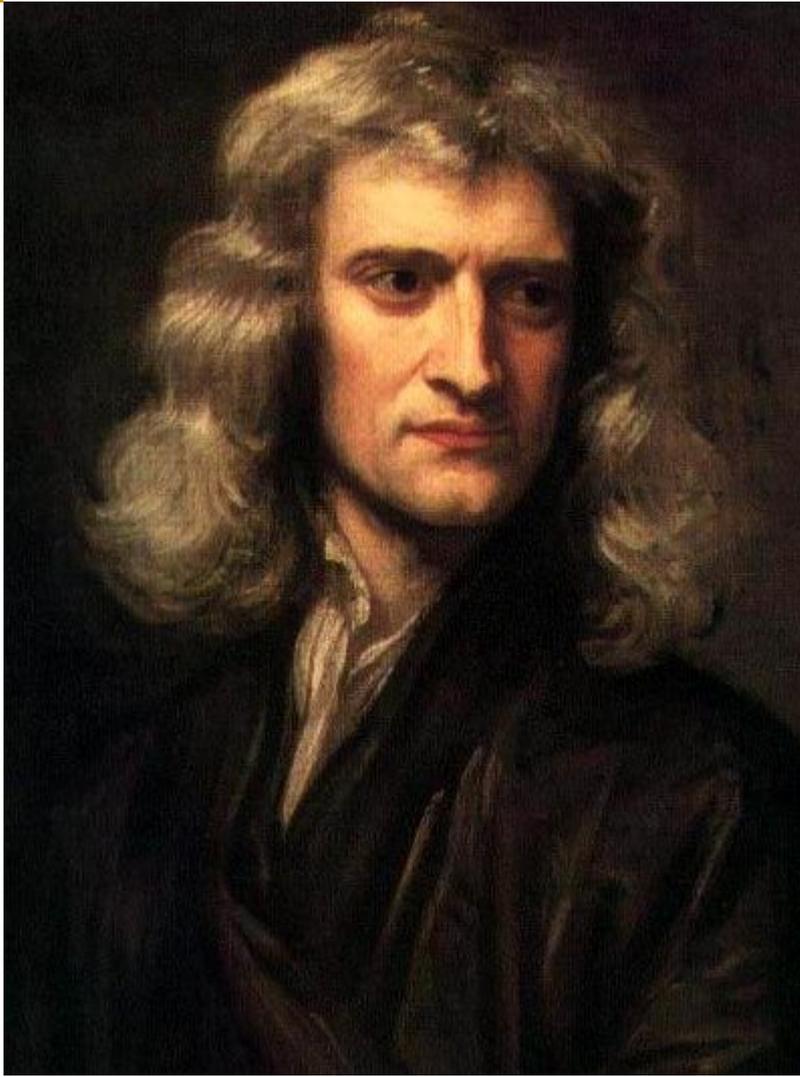
# Объяснение процесса цветовосприятия



Теория Аристотеля  
(до сер. XVII века) -  
все цвета образуются  
при подмешивании  
черного цвета к  
белому

Аристотель  
Род. 384 год до н. э.

# Объяснение процесса цветовосприятия

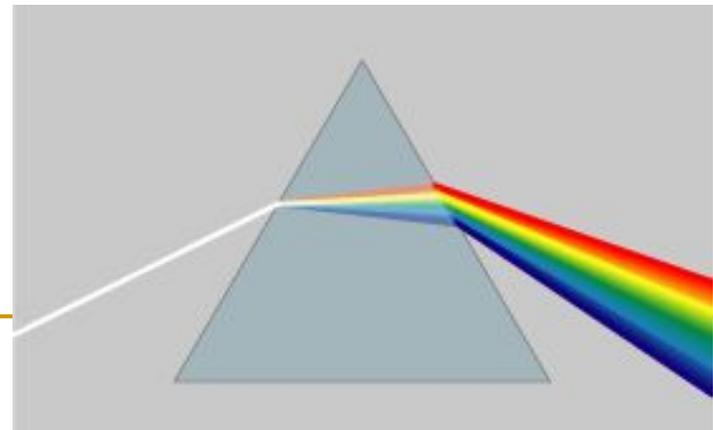


Исаак Ньютон (1642 — 1727)

Заложил основы современной теории цветов

Показал, что белый свет не первичен.

Выделил с помощью призмы в спектре белого света семь наиболее заметных спектральных цветов и назвал их *основными* (далее не разложимыми)— красный, оранжевый, желтый, зеленый, голубой, синий и фиолетовый.



# Объяснение процесса цветовосприятия



М.В. Ломоносов  
1711-1765

В 1756 г. сформулировал **трехкомпонентную теорию восприятия цветов:**

в глазу имеются три вида приемников лучистой энергии (колбочек), воспринимающих красную (длинноволновую), желтую (средневолновую) и голубую (коротковолновую) части видимого спектра.

Для получения любого цвета достаточно использовать всего три основные краски



---

## Подобные гипотезы были выдвинуты

- в Англии Томасом Юнгом в 1807 г.
- в Германии Гемгольцем в 1852 г.

в этих гипотезах за основные цвета были приняты **красный, зеленый и синий**.

- Наши ощущения — результат смешения (суммирования) в различных пропорциях **этих цветов**
-

# Физиологическое обоснование трех компонентной модели цвета

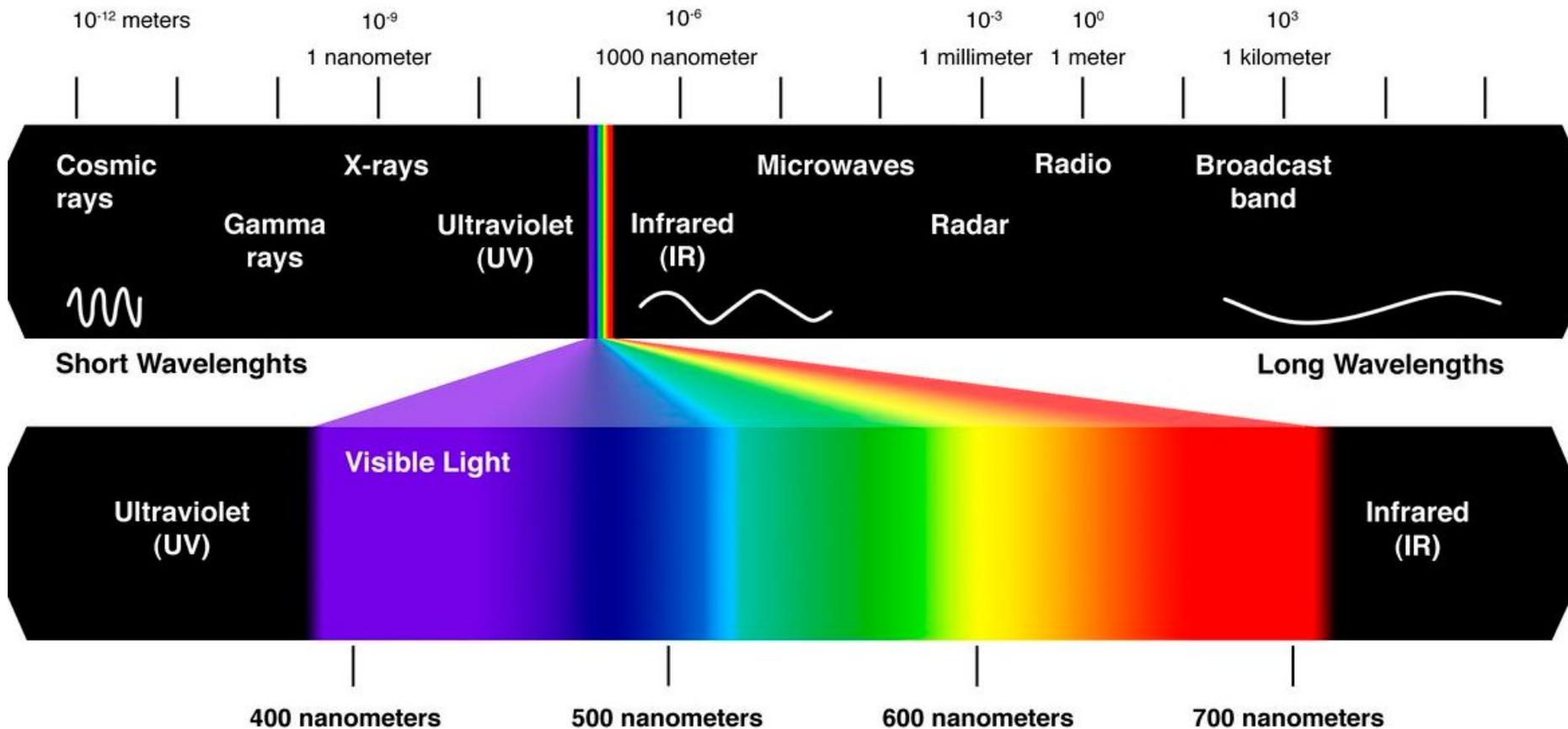
Человеческий глаз содержит четыре типа зрительных рецепторов: «палочки» (рецепторы интенсивности) и три типа «колбочек» (рецепторы цветовых оттенков).

Колбочки каждого типа чувствительны к свету в своем узком диапазоне длин волн.

Тип колбочек	Диапазон длин волн	Максимум чувствительности
Красные	от 760 до 550 нм	~610 нм
Зеленые	от 650 до 450 нм	~550 нм
Синие	от 550 до 380 нм	~450 нм
Общий спектр видимого света	от 760 до 380 нм	555 нм (дневное зрение) 510 нм (ночное зрение)

10 миллионов цветов

# Видимый спектр



# Математический аппарат для трехкомпонентной теории цвета (законы Грассмана)



Герман Грассман  
1809-1877

**Закон трехмерности** : с помощью трех выбранных линейно независимых цветов можно однозначно выразить любой цвет.

*(Цвета считаются линейно независимыми, если никакой из них нельзя получить путем смешения остальных)*

**Закон непрерывности** : при непрерывном изменении излучения цвет смеси также меняется непрерывно.

*(К любому цвету можно подобрать бесконечно близкий цвет)*

**Закон аддитивности** : все цвета равноправны, разложение цветов можно выполнять по любым независимым цветам.



## Законы Грассмана устанавливают общие свойства математических моделей цвета:

- Любому цвету можно однозначным образом поставить в соответствие некоторую точку трехмерного пространства
- Абсолютно черному цвету всегда соответствует точка  $(0, 0, 0)$
- Цвета можно рассматривать как точки или векторы в трехмерном цветовом пространстве



# Цветовые модели



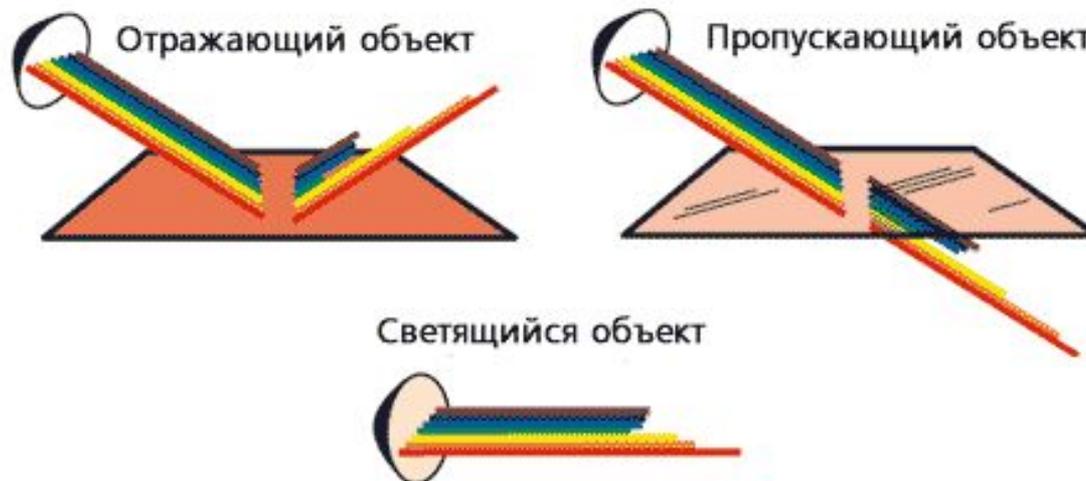
Каждая цветовая модель задает в трехмерном цветовом пространстве некоторую систему координат, в которой основные цвета модели играют роль базисных векторов.

- **RGB** (*Red-Green-Blue*, красный-зеленый-синий).
- **CMYK** (*Cyan-Magenta-Yellow-black*, голубой-пурпурный-желтый-черный) .
- **HSB** (*Hue-Saturation-Brightness*, цветовой оттенок-насыщенность-яркость).



# Обратные цветовые модели RGB и CMYK

- Все объекты окружающего мира можно разделить на:
- излучающие (светящиеся: солнце, лампа, монитор),
  - отражающие излучение (бумага)
  - пропускающие (стекло).



# Цветовая модель RGB – описание цвета излучающего объекта



Модель *RGB* используется в телевизорах, мониторах, проекторах, сканерах, цифровых фотоаппаратах...

*Основные* цвета в этой модели: *красный (Red)*, *зеленый (Green)*, *синий (Blue)*.

# Формирование цвета в модели RGB

Модель RGB является **аддитивной** (суммарной), что означает, что цвета в этой модели *добавляются* к черному цвету.

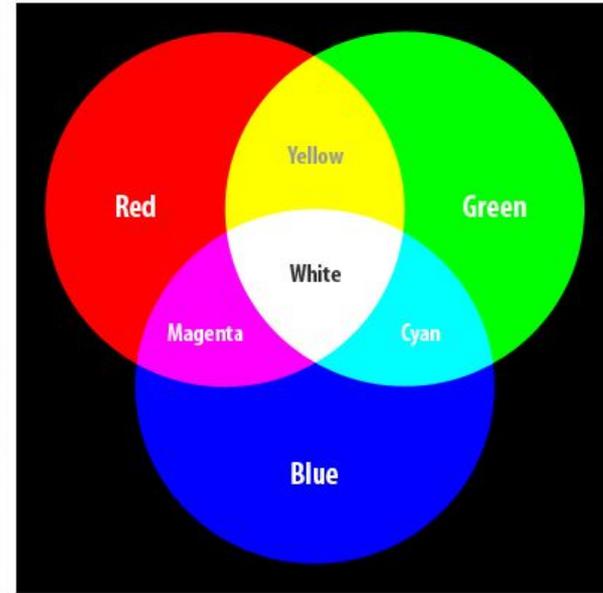
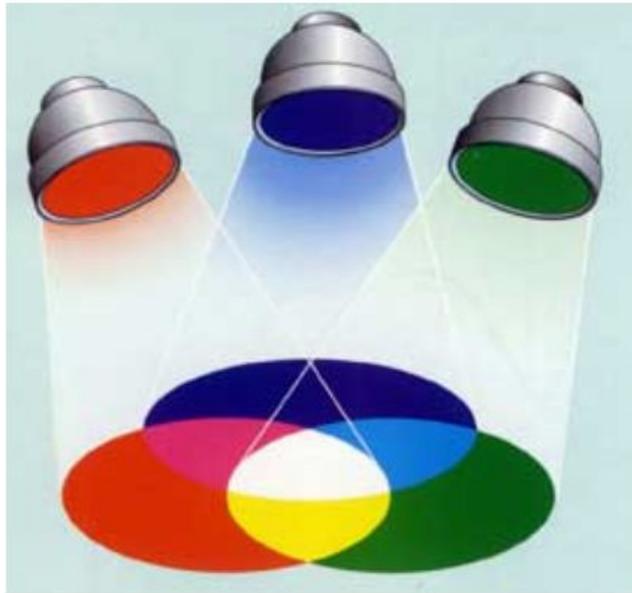


Пиксель монитора *излучает свет* с помощью трех субэлементов: *красного (Red)*, *зеленого (Green)*, *синего (Blue)*.

Чтобы создать на экране **основной** цвет, надо включить (*добавить* к черному цвету) субэлемент определенного типа

Для получения **составного** цвета надо дополнительно включить (т. е. *добавить*) субэлементы другого типа

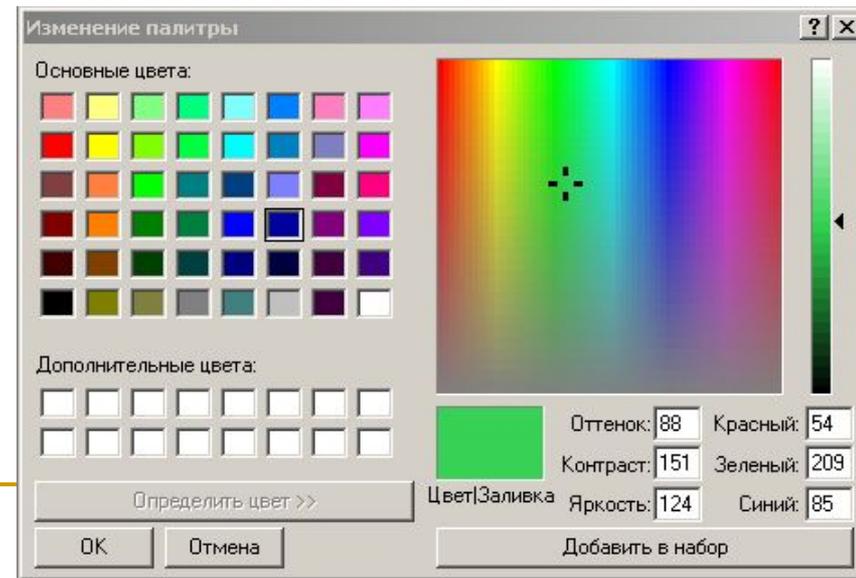
# Цветовая модель RGB



- $R+G=Y$ ;
- $G+B=C$ ;
- $B+R=M$ .

Парное сочетание основных цветов в равных долях дает дополнительные цвета: желтый (*Yellow*), голубой (*Cyan*) и пурпурный (*Magenta*).

Сумма всех трех основных цветов в равных долях дает белый (*White*) цвет:



# Пространство цветов модели RGB

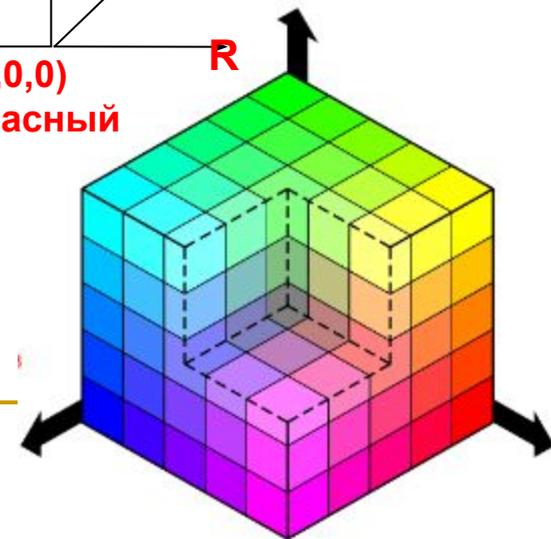
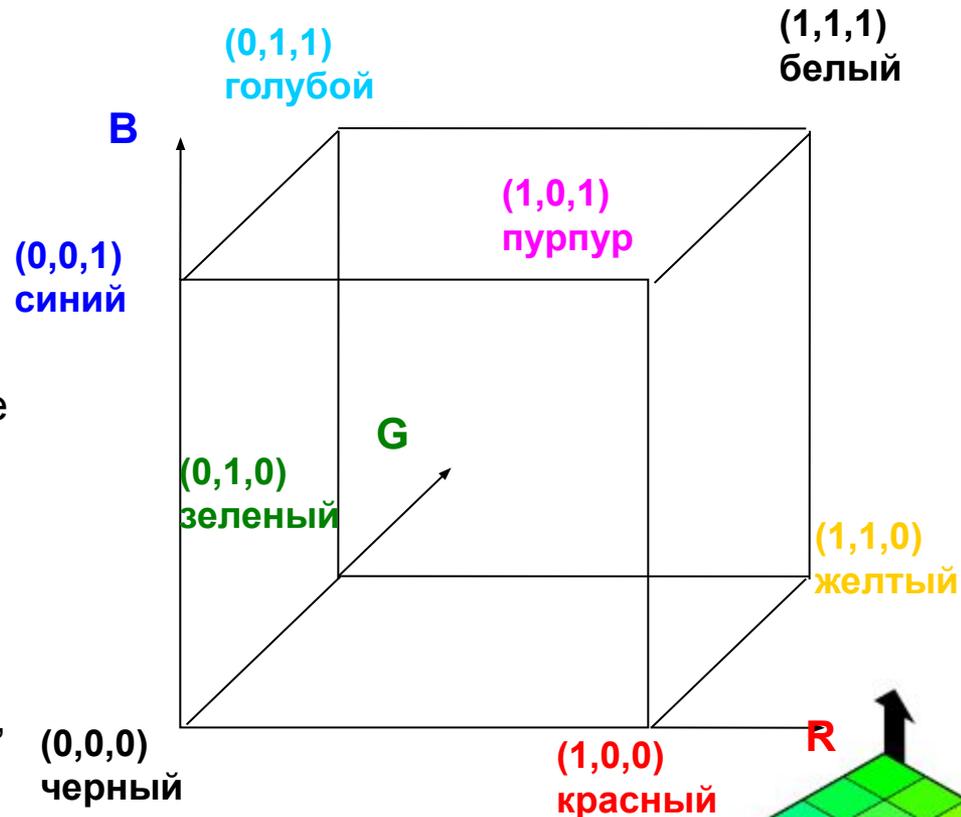
Любая точка куба  $(r, g, b)$  определяет какой-то цвет;

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 СМУК – описание цвета отражающего объекта



Цветовая модель *СМУК* используется в полиграфии при формировании изображений, предназначенных для печати на бумаге.

Основными цветами в ней являются: голубой (*Cyan*), пурпурный (*Magenta*), желтый (*Yellow*)



# Формирование цвета в модели СМУК

Модель СМУК является **субтрактивной** (вычитательной), что означает, что цвета в этой модели получаются *вычитанием* цветов RGB из белого цвета.



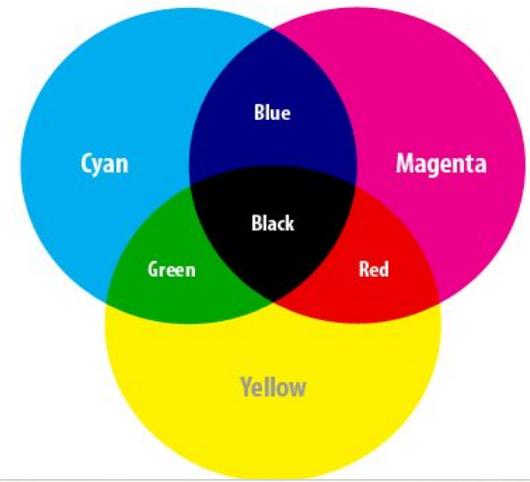
Бумага *отражает* падающий свет, который обычно является «белым»

Краски на бумаге поглощают лучи определенного цвета, а остальные отражают.

Видимый нами цвет краски определяется теми лучами, которые *не были поглощены*.

Таким образом, краски *вычитают*, или *ослабляют* цвета в отражаемом потоке света.

# CMYK

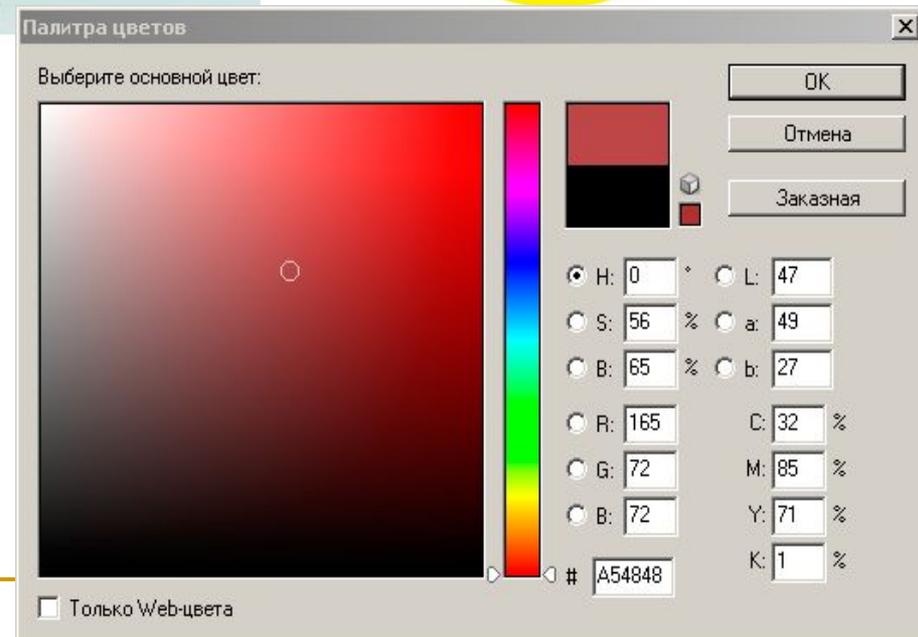


$C=W-R;$   
 $M=W-G;$   
 $Y=W-B$

Парное сочетание в равных долях цветов модели CMY дает цвета модели RGB

В теории, сумма  $C+M+Y=K$ , т.е. дает *черный (black)* цвет,

на практике это затруднительно, поэтому в модели CMYK к триаде CMY добавляют *черный* цвет K



# Пространство цветов модели СМУК

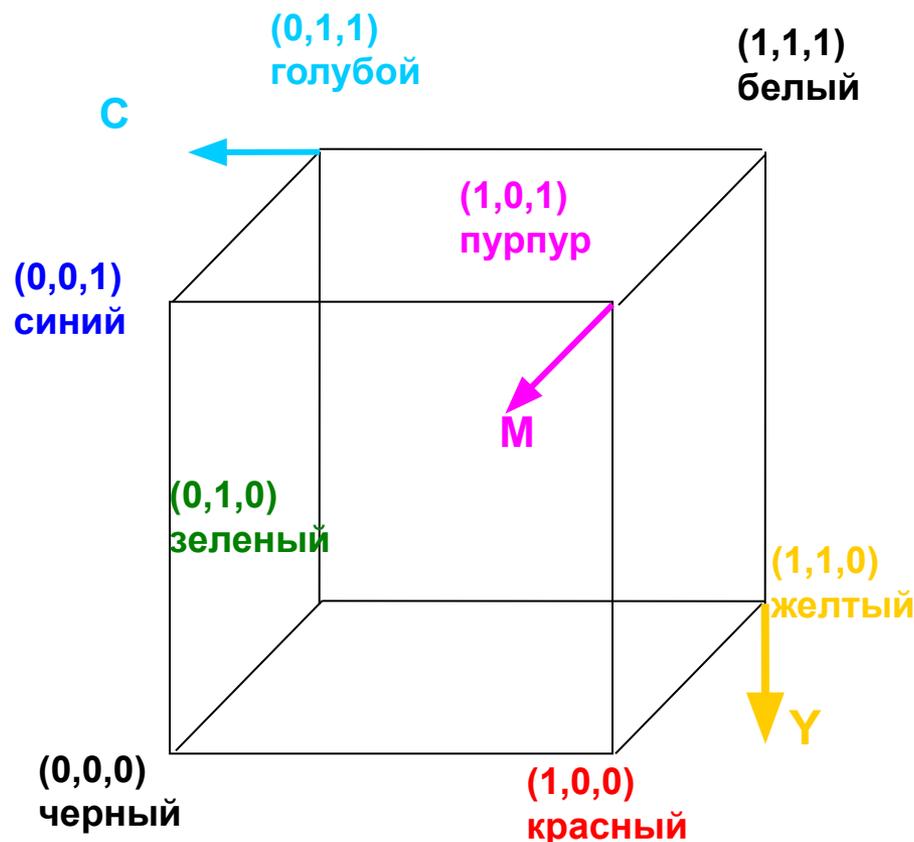
Любая точка куба (C, M, Y)  
определяет какой-то цвет;

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, наибольшее  
значение определяет долю серого  
оттенка, а разность значений – тон  
и долю насыщенного цветового  
оттенка.



# Цветовая модель HSB



Основой для модели HSB (Hue, Saturation, Brightness), она же HSL (Hue, Saturation, Lightness) - являются 3 параметра:

- Тон (Hue),
- Насыщенность (Saturation),
- Яркость (Brightness)



---

# Цветовая модель HSB

(Hue-Saturation-Brightness)

(цветовой тон-насыщенность-яркость)

**Чистый цветовой тон** – один из цветов спектрального разложения цвета.

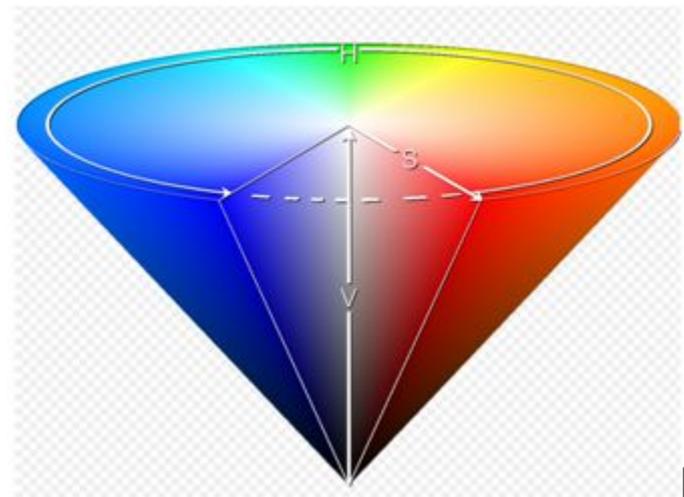
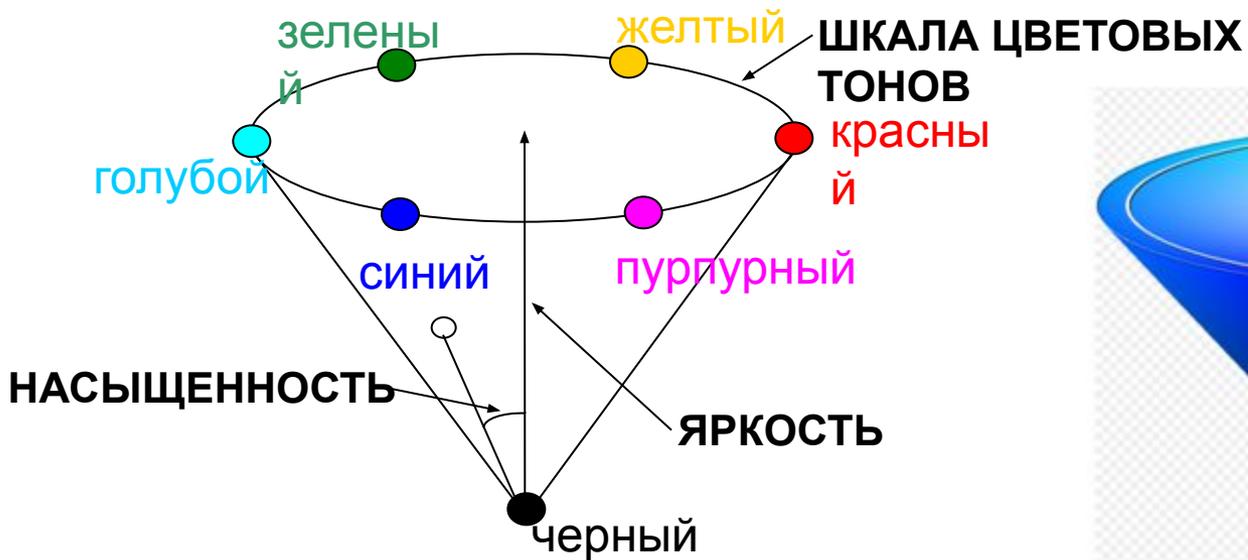
**Цветовой оттенок** – смесь чистого цветового тона с серым цветом.

**Насыщенность цвета** – доля чистого тона в цветовой смеси.

**Яркость** характеризуется общей светлостью смешиваемых цветов

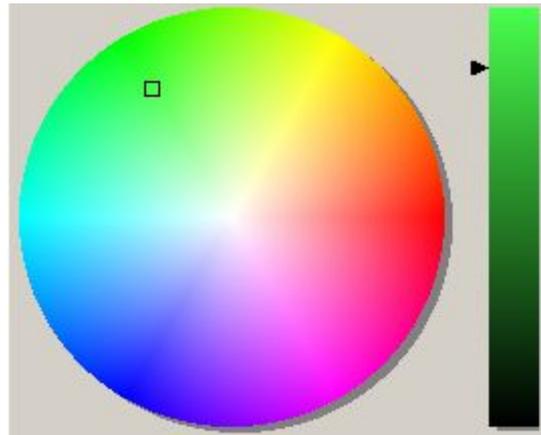


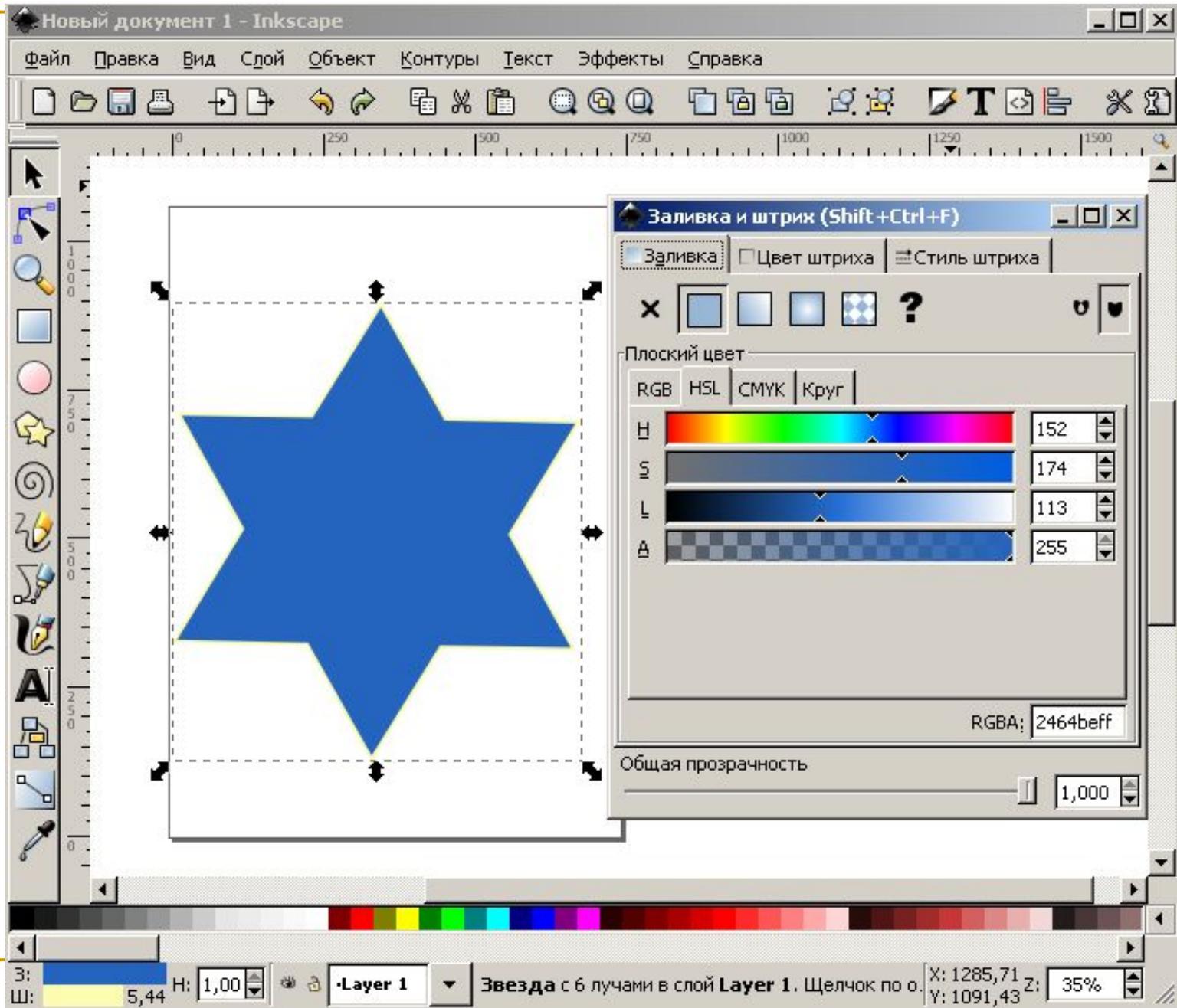
# Цветовое пространство модели HSB



# Способы «разворачивания» трёхмерного пространства HSV на двумерный экран компьютера

- Цветовой круг - эта визуализация состоит из цветового круга (то есть, поперечного сечения цилиндра) и движка яркости (высоты цилиндра).



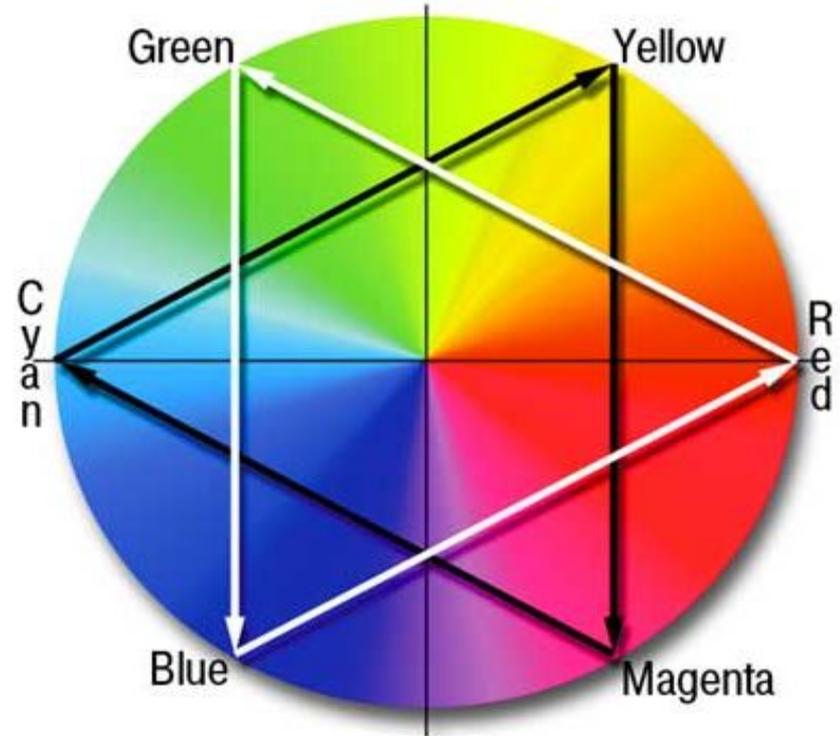


# Цветовое пространство

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



---

**ВЫВОДЫ:** Модель HSB позволяет закодировать практически все цвета, воспринимаемые человеком.

Модели RGB и CMYK описывают возможности компьютерных устройств по воспроизведению цвета.

Некоторые цвета в принципе не могут быть воспроизведены на компьютере



---

# Квантование (кодирование) цветового пространства



# Квантование цветового пространства

- Квантование достигается переводом вещественных чисел из интервала  $[0; 1)$  в интервал целых чисел от 0 до  $N - 1$  путем умножения на целое число  $N$ , с последующим округлением.
- Интервал  $[0; 1)$  каждой цветовой оси разбивается на  $N$  равных подинтервалов
- После квантования каждый цвет представляется *триадой* целых неотрицательных чисел

$$(k_r, k_g, k_b), \quad 0 \leq k_i < N$$

$N$  – количество цветов



# Глубина цвета

- Число  $N$  выбирают степенью 2

$$N = 2^i$$

- Величину  $i$  называют **глубиной цвета**

- $i$  – количество бит, которым кодируется 1 пиксель (сумма бит, которыми кодируются каждая компонента цвета)

$$i = i_r + i_g + i_b$$

# Глубина цвета

Количество бит, используемых для кодирования цвета одной точки называется *глубиной цвета* (битовой глубиной, цветовым разрешением)

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.



---

*Глубина цвета, к (бит)*

*Количество отображаемых цветов, N*

**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)**

$$2^{24} = 16\,777\,216$$

---

Пример: для кодирования одной компоненты цвета используется 1 бит

1 бит на каждый компонент RGB  $2^3=8$	<i>R</i>	<i>G</i>	<i>B</i>	<i>Цвет</i>
	1	1	1	W (white / белый)
	1	1	0	Y (yellow / желтый)
	1	0	1	M (magenta / пурпурный)
	1	0	0	R (red / красный)
	0	1	1	C (cyan / голубой)
	0	1	0	G (green / зеленый)
	0	0	1	B (blue / синий)
	0	0	0	K (black / черный)

$$i = 1+1+1 = 3 \text{ бит} \quad N = 2^3 = 8 \text{ цветов}$$



Пример: для кодирования компонент R и B используется 5 бит, компоненты G – 6 бит (стандарт HighColor )

На глубину красного и синего цвета отводится 5 бит, на глубину зеленого — 6 бит	<i>R</i>	<i>G</i>	<i>B</i>	<i>Некоторые цвета</i>
$2^{16}=65536$	11111	11111	111111	W (white / белый)
	11111	11111	000000	Y (yellow / желтый)
	11111	00000	111111	M (magenta / пурпурный)
	11111	00000	000000	R (red / красный)
	00000	11111	111111	C (cyan / голубой)
	00000	11111	000000	G (green / зеленый)
	00000	00000	111111	B (blue / синий)
	00000	00000	000000	K (black / черный)

$$i = 5+5+6 = 16 \text{ бит} \quad N = 2^{16} = 65536 \text{ цветов}$$



Пример: для кодирования одной компоненты цвета используется 1 байт (стандарт TrueColor)

1 байт на каждый компонент RGB	<i>R</i>	<i>G</i>	<i>B</i>	Некоторые цвета
$2^{24}=16\ 777\ 216$	11111111	11111111	11111111	W (white / белый)
	11111111	11111111	00000000	Y (yellow / желтый)
	11111111	00000000	11111111	M (magenta / пурпурный)
	11111111	00000000	00000000	R (red / красный)
	00000000	11111111	11111111	C (cyan / голубой)
	00000000	11111111	00000000	G (green / зеленый)
	00000000	00000000	11111111	B (blue / синий)
	00000000	00000000	00000000	K (black / черный)

$$i = 8+8+8 = 24 \text{ бит} \quad N = 2^{24} = 16777216 \text{ цветов}$$

---

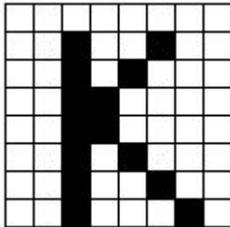
# Квантование цветового пространства в модели HSB

- В ОС Windows каждая из HSB-характеристик описывается одним байтом



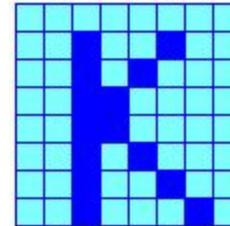
# Битовая карта изображения

Битовая карта является двоичным кодом изображения, хранится в *видеопамяти* компьютера, считывается *видеопроцессором* (не реже 60 раз в секунду – частота обновления экрана) и отображается на экран.



Битовая карта черно-белого изображения будет выглядеть так:

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



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**

**Информационный объем  
изображения**

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

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

# Несжатое полноцветное растровое изображение занимает значительное место в памяти компьютера.

При масштабировании (изменении размеров изображения)  
качество растрового изображения значительно ухудшается



При увеличении проявляется  
«пикселизованность» - контуры  
становятся ступенчатыми  
(добавляются лишние точки с цветом  
соседней точки)

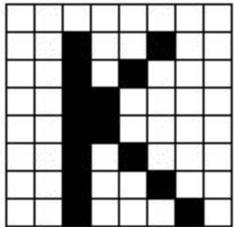
При уменьшении  
безвозвратно теряется  
часть информации  
(уменьшается количество  
точек)



# Векторная графика

В векторном представлении графическое изображение на экране формируется из объектов – линий, прямоугольников, окружностей, дуг, закрасок – которые называются **графическими примитивами**.

В этом случае графическая информация – это данные, однозначно определяющие все графические примитивы, составляющие рисунок (координаты, толщину линий, цвет).



Векторное изображение занимает значительно меньше места, легко масштабируется, при этом качество изображения не ухудшается.

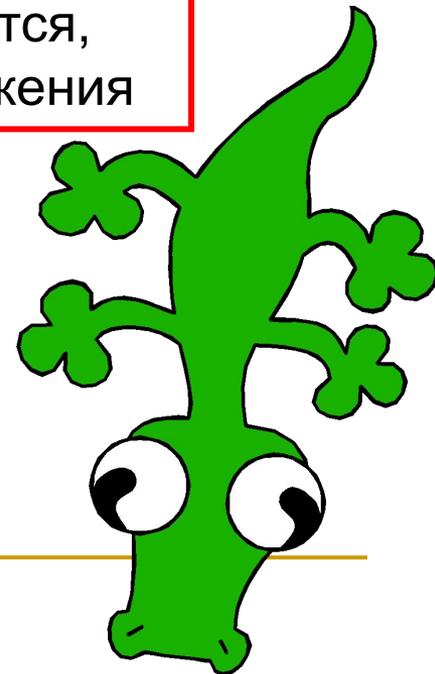
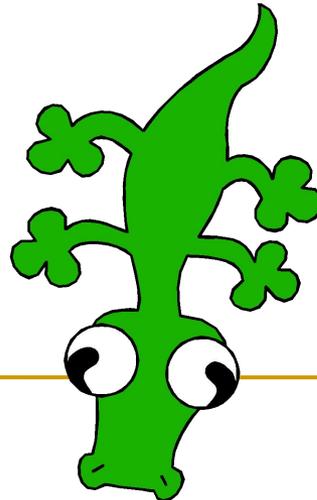
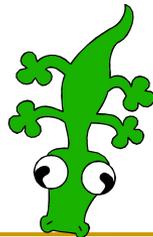
## Векторные команды:

Цвет черный

Линия 3, 2, 3, 8

Линия 4, 4, 6, 2

Линия 4, 5, 7, 8



# Ответьте письменно на вопросы

- 1) Изображение разбивают на графические примитивы. Что это за процедура: дискретизация или квантование? Какое изображение будет сформировано после этой процедуры: растровое или векторное?
- 2) Сколько базовых цветов нужно использовать, чтобы воспроизвести любой из видимых цветов?
- 3) Какую цветовую модель нужно выбрать в графическом редакторе, если вы создаете изображение, которое будет демонстрироваться только с экрана компьютера?
- 4) Какая цветовая модель позволяет закодировать все видимые цвета спектра, в том числе те, которые не представимы на компьютере?
- 5) Сколько цветов можно использовать в изображении, если для кодирования одного цвета в памяти компьютера используется

# Форматы графических файлов.

**Windows Bit MaP** (расширение файлов - **.bmp**) – формат операционной системы Windows для хранения растровых изображений; поддерживается всеми Windows-приложениями.

**TIFF (Tagged Image File Format)** (расширение файлов - **.tif**) – предназначен для хранения растровых изображений высокого качества в широком цветовом диапазоне; поддерживается большинством графических, издательских и дизайнерских программ;

**GIF (Graphic Interchange Format)** (расширение файлов - **.gif**) – стандартизирован в 1987 г. как средство хранения изображений с фиксированным (256) количеством цветов. Из-за ограниченных цветовых возможностей применяется исключительно в электронных публикациях. Благодаря компактности файлов широко используется для размещения графических изображений на Web-страницах в Интернете

**JPEG (Joint Photographic Experts Group)** (расширение файлов - **.jpg**) –обеспечивает хранение растровых графических изображений в более компактной форме на основе использования эффективного алгоритма сжатия. Применяемые методы сжатия основаны на удалении «избыточной» информации. Позволяет регулировать соотношение между степенью сжатия файла и качеством изображения.

**PSD (PhotoShop Document)** (расширение файлов - **.psd**) – собственный формат графического редактора Adobe Photoshop, один из наиболее мощных по возможностям хранения растровой графической информации.

**PDF (Portable Document Format)** (расширение файлов - **.pdf**) – разработан фирмой Adobe для хранения изображений документов (например, страниц книг, журналов и др.); является аппаратно-независимым (вывод изображений допустим на любых устройствах). Мощный алгоритм сжатия со средствами управления итоговым разрешением изображения обеспечивает компактность файлов при высоком качестве иллюстраций.

**WMF (Windows MetaFile)** (расширение файлов - **.wmf**) – формат операционной системы Windows для хранения векторных изображений; поддерживается всеми Windows-приложениями. Однако отсутствие средств для работы со стандартизированными цветовыми палитрами, принятыми в полиграфии, и другие недостатки ограничивают его применение.

---

## Задание

Заполните таблицу, вычислив информационный объем графической информации при различных соотношениях графического и цветового разрешения.

Разрешение	16 цветов	256 цветов	65536 цветов	16777216 цветов
640*480				
800*600				
1024*768				
1280*1024				

Закодируйте монохромный рисунок с помощью двоичного алфавита в соответствии с матричным принципом.

*Имеем матрицу 6x9, всего 54 бита. Закрашенной клетке поставим в соответствие 1, незакрашенной – 0.*

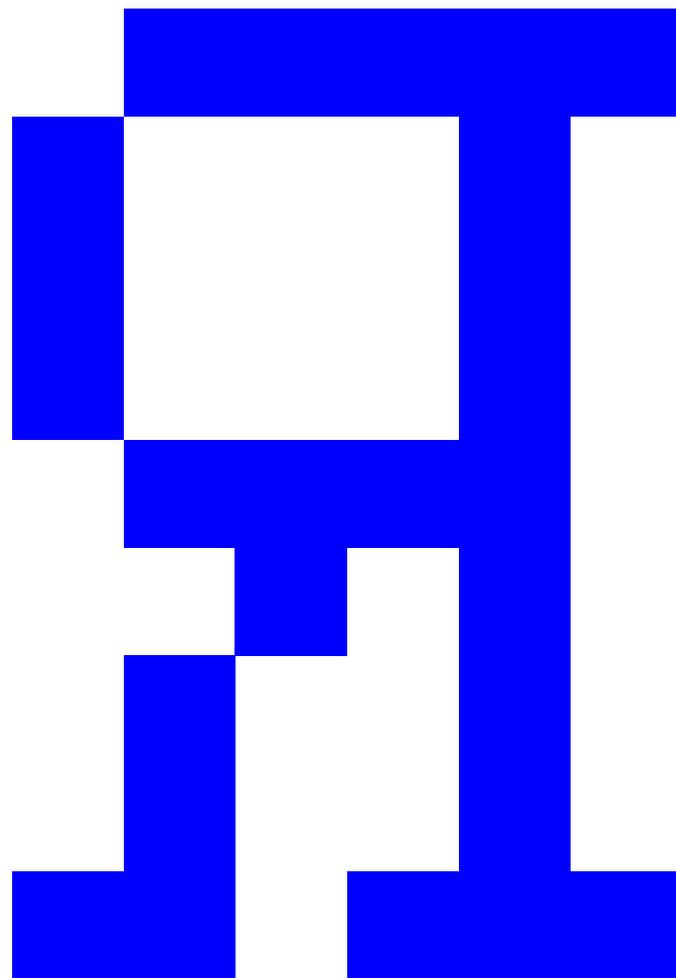
Получим: **011111 100010**  
**100010 100010 011110**  
**001010 010010 010010**

**Сравните:**

**код буквы «я» в КОИ8 - 11110001**

**ВЫВОД:** Отсканированная страница текста занимает места в памяти больше, чем та же страница после распознавания текста (перевода рисунка в текстовый формат)

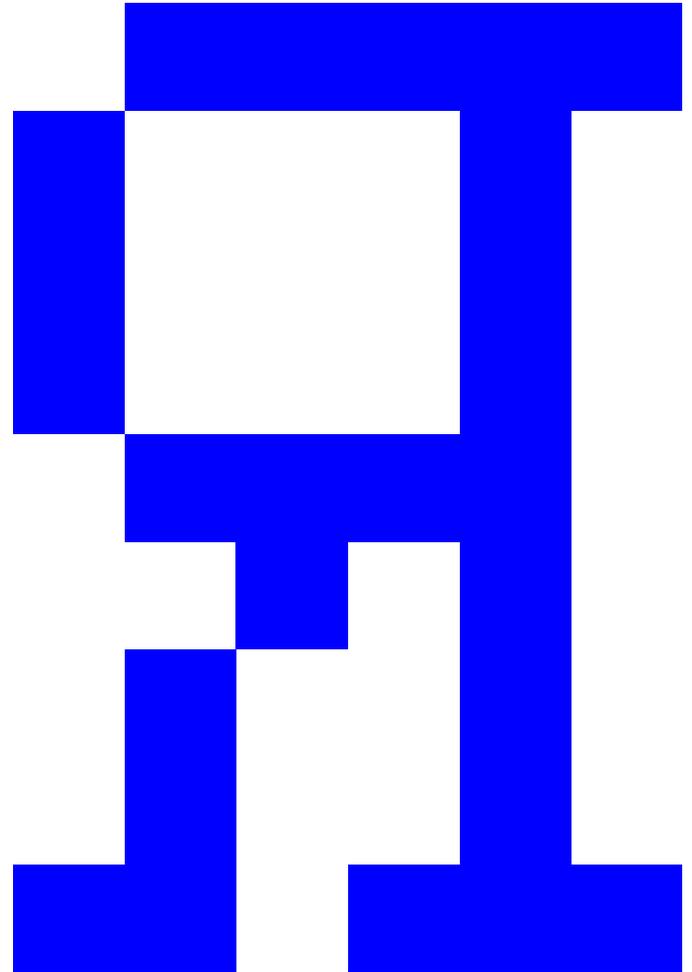
## Задача 1.



Сколько места в памяти будет занимать тот же рисунок, если сохранить его в формате как

- А) 256-цветный рисунок;
- В) в режиме HighColor;
- С) в режиме True Color?

## Задача 2.



### Решение.

Рисунок разбит на  $6 \cdot 9 = 54$  пикселя.

А)  $256 = 2^8$ , т.е. код каждого пикселя передается 8 битами.  **$I = 54 \cdot 1 = 54$  байта**

В) HighColor: 1 пиксель передается 16 битами (2 байта).  **$I = 54 \cdot 2 = 108$  байтов**

С) TrueColor: цвет пикселя передается 24 битами (3 байта).  **$I = 54 \cdot 3 = 162$  байта** (т.е. в 24 раза больше, чем монохромный)

**ВЫВОД:** Монохромный рисунок нужно сохранять именно как монохромный

## Задача 3.

Какой объем видеопамати необходим для хранения четырех страниц изображения, при условии, что разрешающая способность дисплея равна 640X480 точек, а используемых цветов 32?

### Решение.

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

2)  $I=640*480*5*4=6144000$  бит = 750

Кбайт

**Ответ: 750 Кбайт**

## Задача 4.

265-цветный рисунок содержит 1 Кбайт информации. Из скольких точек он состоит?

### Решение.

- 1)  $N=2^i$ ,  $256=2^i$ ,  $i=8$  бит – информационный объем одной точки;
- 2) 1 Кбайт =  $1024*8$  бит = 8192 бит - объем изображения;
- 3)  $8192:8=1024$  точек – на изображении

**Ответ: 1024 точки**

## Задача 5.

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

### Решение.

1)  $1024 \cong 10 * 10 * 10$  – по 10 бит приходится на каждую из трех составляющих (красную, зеленую, синюю);

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

**Ответ: 30 бит**

## Задача 6.

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 7.

Видеопамять имеет объем, в котором может храниться 8-цветное изображение размером 640X350 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
- 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**

# Задание 1

- Для хранения растрового изображения размером  $128 \times 128$  пикселей отвели 4 кбайта памяти. Каково максимально возможное число цветов в палитре изображения.



## Задание 2

- Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 640\*480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами



## Задание 3

- Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером  $64 \times 64$  пикселя, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно



## Задание 4

- Рассчитайте объем видеопамяти, необходимой для хранения графического изображения, занимающего весь экран монитора с разрешением 640 x 480 и количеством отображаемых цветов, равным 65 536.



## Задание 5

- Подсчитайте объем информации, передаваемой от видеоадаптера к монитору в видеорежиме 1024 x 768 пикселей с глубиной цвета 24 бита и частотой обновления экрана 85 Гц.



## Задание 6

- Вы хотите работать с разрешением 1600 x 1200 пикселей, используя 16777216 цветов. В магазине продаются видеокарты с памятью 512 Кбайт, 2 Мбайта, 4 Мбайта и 64 Мбайта. Какие из них можно купить для вашей работы?

