



МОУ «Новоархангельская СОШ»

# Графическое представление информации.

Выполнила :ученица 11 класса  
Коптян Ирина.

Руководитель: учитель информатики  
Скульбеда Н.И.

2006 год

# Цель:

- рассказать о графическом представлении информации;
- показать, как кодируется графическая информация в компьютере;
- научить вычислять объем видеопамяти;
- научить вычислять цветность компьютера.

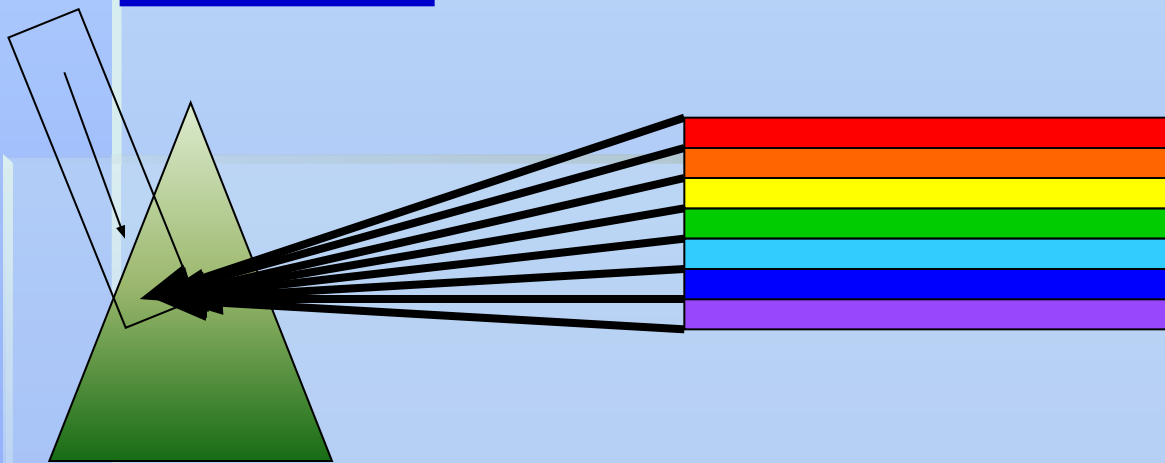
# СОДЕРЖАНИЕ

- Почему трава зеленая.
- Как мы видим цветные изображения.
- Цветное изображение монитора.
- Кодирование цвета.
- Вычисление цветности изображения.
- Вычисление размера видеопамяти.

# Почему трава зеленая

- Известно, что белый цвет представляет собой смесь цветов, Это легко увидеть, если пропустить луч света через стеклянную призму, Так как разные цвета имеют разные углы преломления, то мы увидим все составляющие белого цвета по отдельности, Условно эти цвета разбивают на семь групп («цвета радуги»)

Белый цвет

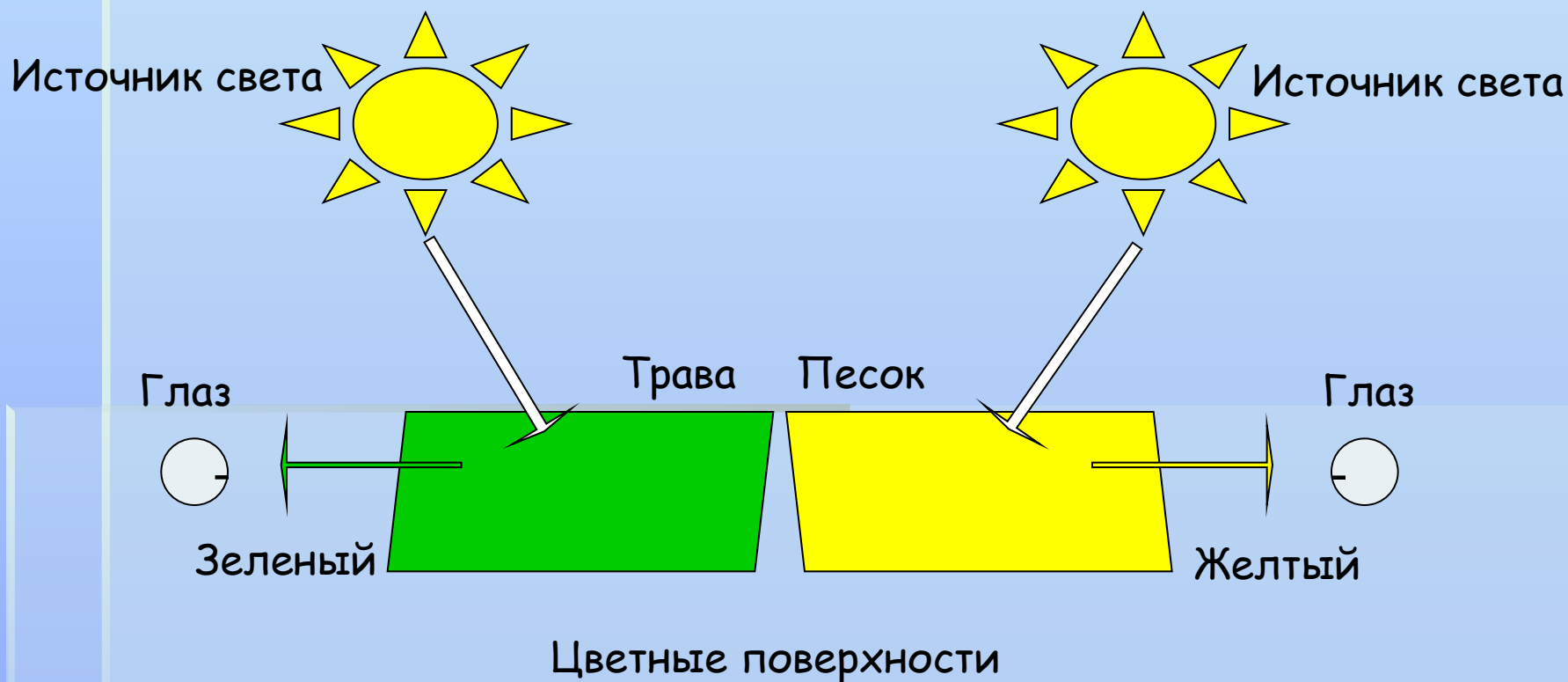


# Белый цвет ≠ 7 цветов



- В семь групп собраны все оттенки красного, оранжевого, зелёного, голубого, синего и фиолетового цветов.
- На рисунке каждая группа условно изображена одним «чистым» цветом.

В солнечном свете мы видим траву зеленой, потому что она отражает зеленый цвет и он попадает в наши глаза, а остальные цвета- поглощает. А песок выглядит желтым ,потому что он отражает только желтый цвет, а остальные поглощает .



**Мы видим свет, когда он попадает на светочувствительные клетки глаза (сетчатку).**

Нервные  
клетки

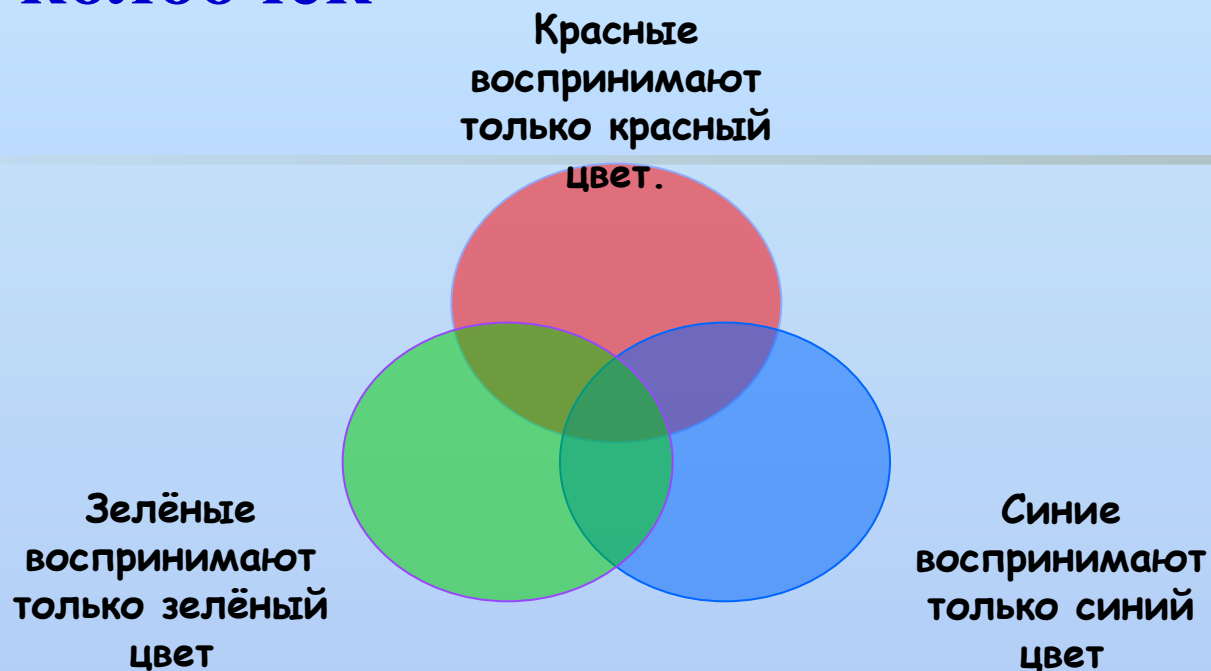
Отвечают  
за цветное  
зрение

Колбочки

Палочки

Отвечают за  
черно-белое  
вечернее и  
ночное зрение

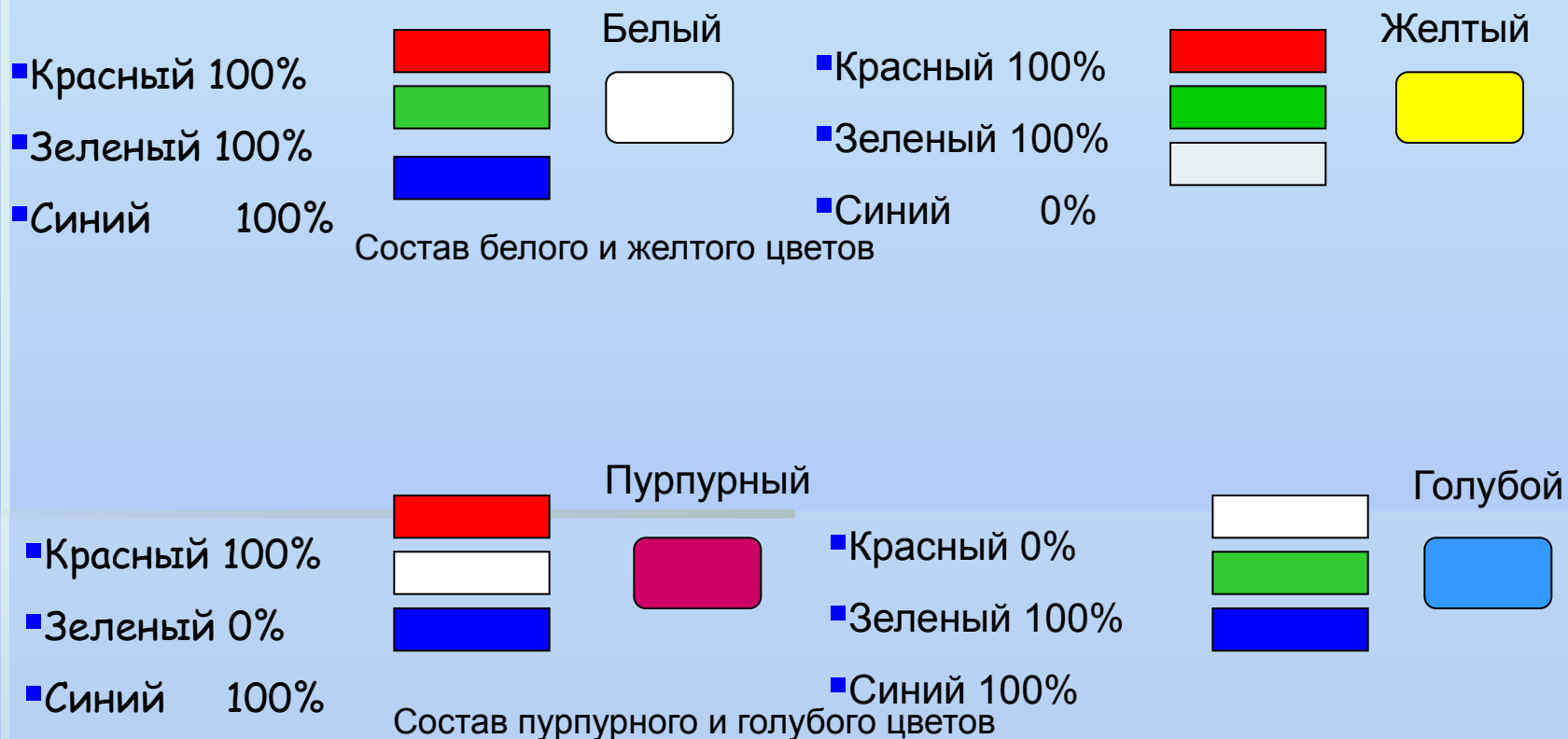
# Состав колбочек



Информация от колбочек поступает в зрительный нерв, где суммируется, и человек видит цвет, как смесь красной, зелёной и синей составляющей.

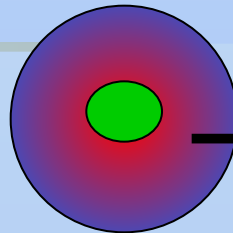
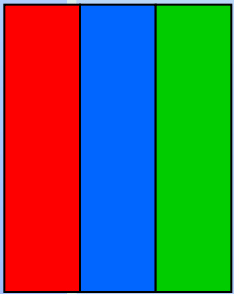


Какой именно оттенок мы видим, зависит от пропорций, в которых смешиваются основные цвета. Смешивая в равных количествах красную, зеленую и синюю краску, получаем белый цвет

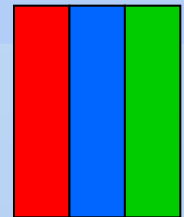


# Цветное изображение на экране монитора также получается смешиванием красок.

Каждый пиксель состоит из трех крупинок люминофора — красного, зеленого и синего цветов.



=















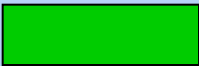

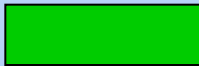

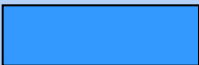



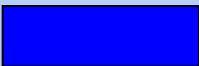


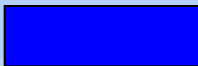







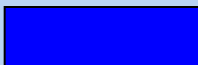
Механизм цветопередачи монитора практически повторяет устройство человеческого глаза

- Система кодирования цвета при помощи трех составляющих- **красной**, **зелёной** и **синей**- носит название RGB.



Число возможных цветов на экране компьютера зависит как от физических характеристик самого монитора, так и от количества памяти, расположенной на видеокарте. Монитор можно настроить на разные режимы работы.

Пусть каждый из трех образующих цветов либо участвует в образовании цвета, либо нет. Тогда для кодирования интенсивности красного, зеленого или синего цветов достаточно двух значений: 0- цвета нет, 1- цвет есть. При таком кодировании получается палитра из 8 цветов.

Цвет	R(красный)	G(зеленый)	B(синий)	Код
				000
				001
				010
				011
				100
				101
				110
				111

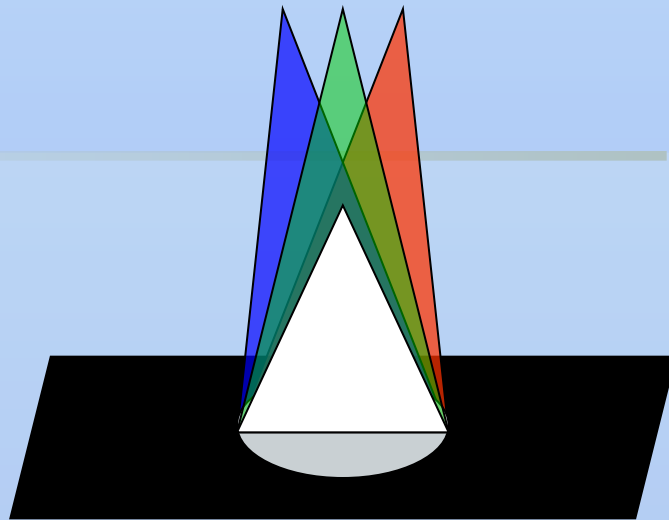
Каждая RGB-компонента принимает два значения

# Вычисление цветности монитора

Цветное пятно получается наложением RGB-лучей трех прожекторов.

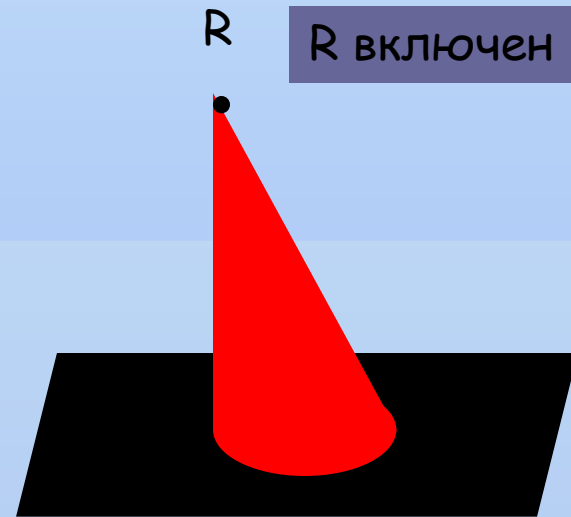
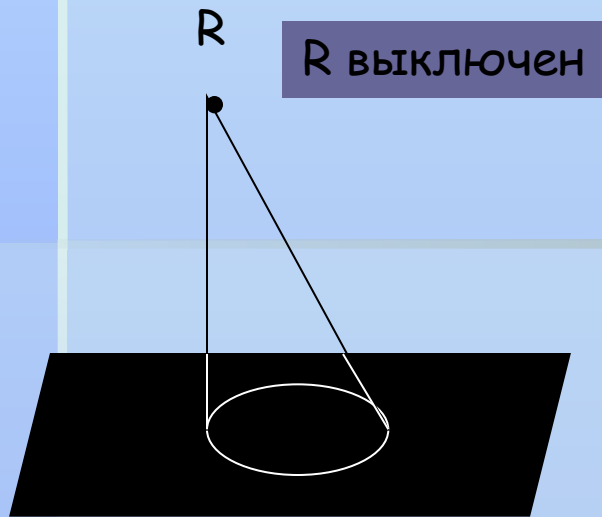
Пусть каждый прожектор имеет два состояния: выключен и включен.

Сколько цветов на экране смогут создать такие прожекторы?



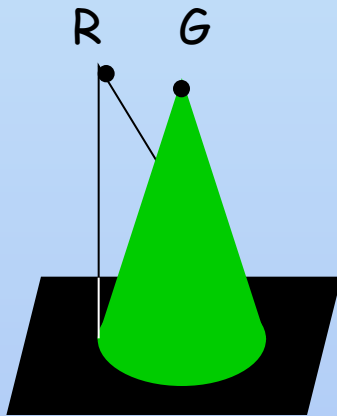
Пусть каждый прожектор имеет два состояния: выключен и включен.  
Сколько цветов на экране смогут создать такие прожекторы?

Если прожектор один, то он создает два цвета (один из них черный).

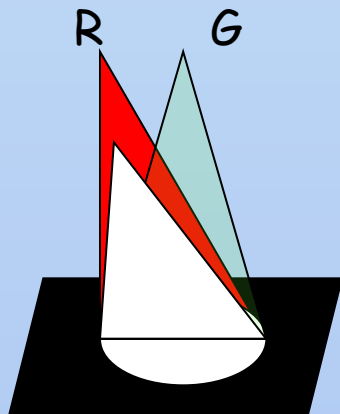
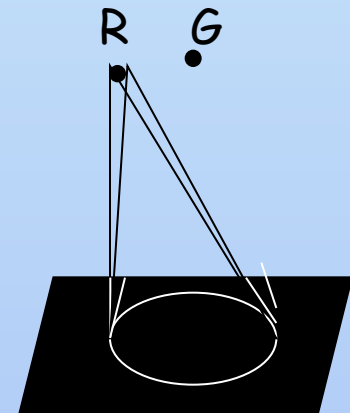


Пусть теперь прожекторов два. В каждом состоянии первого второй может быть выключен или включен.

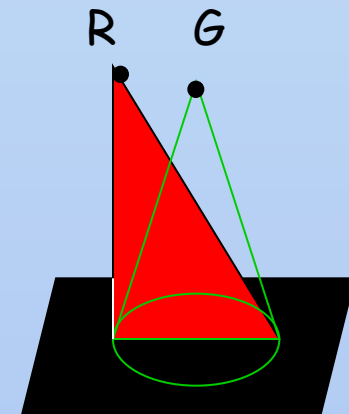
Получается, что двумя прожекторами можно получить четыре цвета: каждое из двух состояний первого прожектора «умножается» на два состояния второго.



R выключен



R включен



# Вычисление цветности.

Цветность вычисляется перемножением трех чисел, задающих количество возможных состояний для каждого прожектора.

Пусть число вариантов интенсивности каждой RGB-компоненты равно  $k$ . Получаем универсальную формулу для вычисления цветности  $C$ :

$$C = k * k * k = k^3$$



# Вычисление размера видеопамяти

Сначала определим, сколько бит потребуется для кодирования  $k$  состояний одной RGB-компоненты.

Пусть  $k=2$  (прожектор выключен, прожектор включен). Для кодирования этих состояний прожектора достаточно одного бита.

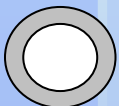
## Два состояния прожектора

$k=2$	<u>Состояние</u>	<u>Двоичный код</u>
-------	------------------	---------------------

	0	0
--	---	---

	1	1
--	---	---

Для 2-х состояний  
достаточно одного  
двоичного разряда



# Три состояния прожектора





Для  $k=3$  потребуется 2 бита.

$k=3$	<u>Состояние</u>	<u>Двоичный код</u>
	0	00
	1	01
	2	10

Для 3-х состояний достаточно 2-х двоичных разрядов

# Четыре состояния прожектора






Двух битов достаточно для кодирования и четырех состояний.

k=4	<u>Состояние</u>	<u>Двоичный код</u>
	0	00
	1	01
	2	10
	3	11

Для 4-х  
состояний  
достаточно 2-х  
двоичных  
разрядов

# Пять состояний прожектора

А вот для кодирования 5 состояний двух битов уже мало.

$k=5$	<u>Состояние</u>	<u>Двоичный код</u>
	0	000
	1	001
	2	010
	3	011
	4	100

Число битов, необходимых для кодирования  $k$  состояний, равно числу двоичных разрядов в двоичной записи числа  $(k-1)$ .

# Алгоритм вычисления размера видеопамяти

Пусть монитор работает в разрешении  $w \times h$ , и каждая RGB-компонента может быть в одном из  $k$  состояний.  
Определить  $v$ -размер необходимой видеопамяти.

1. Определим число бит для кодирования одной компоненты.  
Для этого запишем число  $k-1$  двоичным кодом и подсчитаем число получившихся двоичных разрядов  $b$ .
2. Определим число бит, необходимых для кодирования одного пикселя:  $p=3b$  (кодирование трех компонент).
3. Определим размер видеопамяти:  $V=pwh$



# Пример.

Подсчитаем необходимый размер видеопамати для разрешения  $640*480$ , если каждая RGB-компонента имеет 6 градаций интенсивности

Дано:  $k=6$ ,  $w=640$ ,  $h=480$

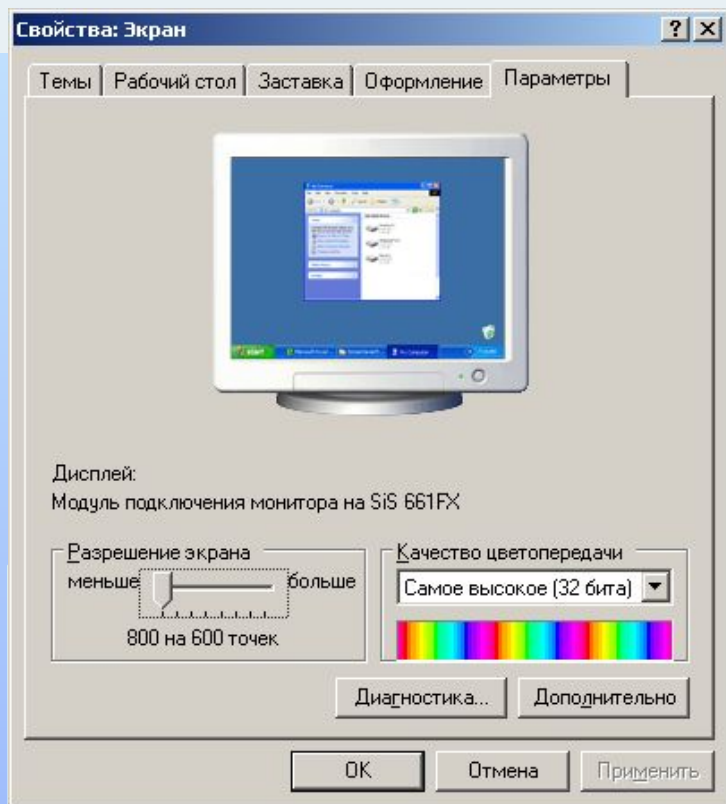
1. Определим число бит для кодирования одной компоненты. Для этого запишем число  $k-1=5$  двоичным кодом и подсчитаем получившееся число двоичных разрядов:  
 $5=101$ ,  $b=3$

2. Определим число бит, необходимых для кодирования одного пикселя:  $p=3*3=9$  бит.

3. Определим размер видеопамати:  
 $V=9*640*480=2\ 764\ 800$  бит =  $345\ 600$  байт



# Режимы работы монитора



Как правило, монитор может работать в разных цветовых режимах. Обычный режим для современного компьютера - «True Color (24 бита)»

Английское выражение «True Color» переводится как «естественные цвета».

Каждый составляющий RGB-цвет кодируется в этой палитре 8 битами. Для трех цветов получается 24 бита на один пиксель

Восьмью битами можно кодировать числа от 0 до 255, то есть всего можно закодировать 256 значений.

Подсчитаем, сколько цветов вмещается в 24-битную палитру и сколько видеопамяти потребуется для хранения полного экрана монитора с разрешением 1024 x 768.

Число цветов определяется перемножением числа вариантов RGB-составляющих. Получается:

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

Объем памяти:

$$24*1024*768=18\ 874\ 368\ \text{бит.}$$

В килобайтах получается:

$$18\ 874\ 368 / 8 / 1024=2304\ \text{Кб,}$$

что составляет 2,25 Мб



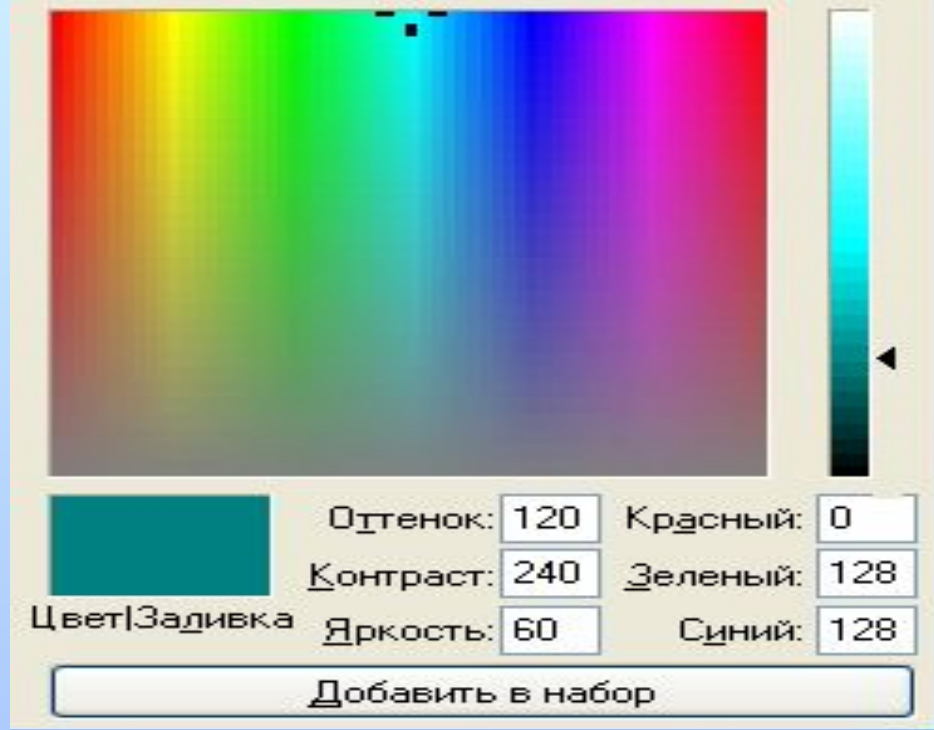


# КОНСТРУИРОВАНИЕ ЦВЕТА

RGB-модель, основанная на физических свойствах света, не слишком удобна для практического цветового конструирования. Поэтому все графические редакторы снабжаются перцепционной (ориентированной на восприятие) системой светового кодирования HSB. В этой системе в качестве цветовых координат выступают:

- ❑ Цветовой тон или оттенок (Hue)
- ❑ Насыщенность или контрастность (Saturation)
- ❑ Яркость (Brightness)

показано окно HSB-конструирования цвета из редактора Paint.

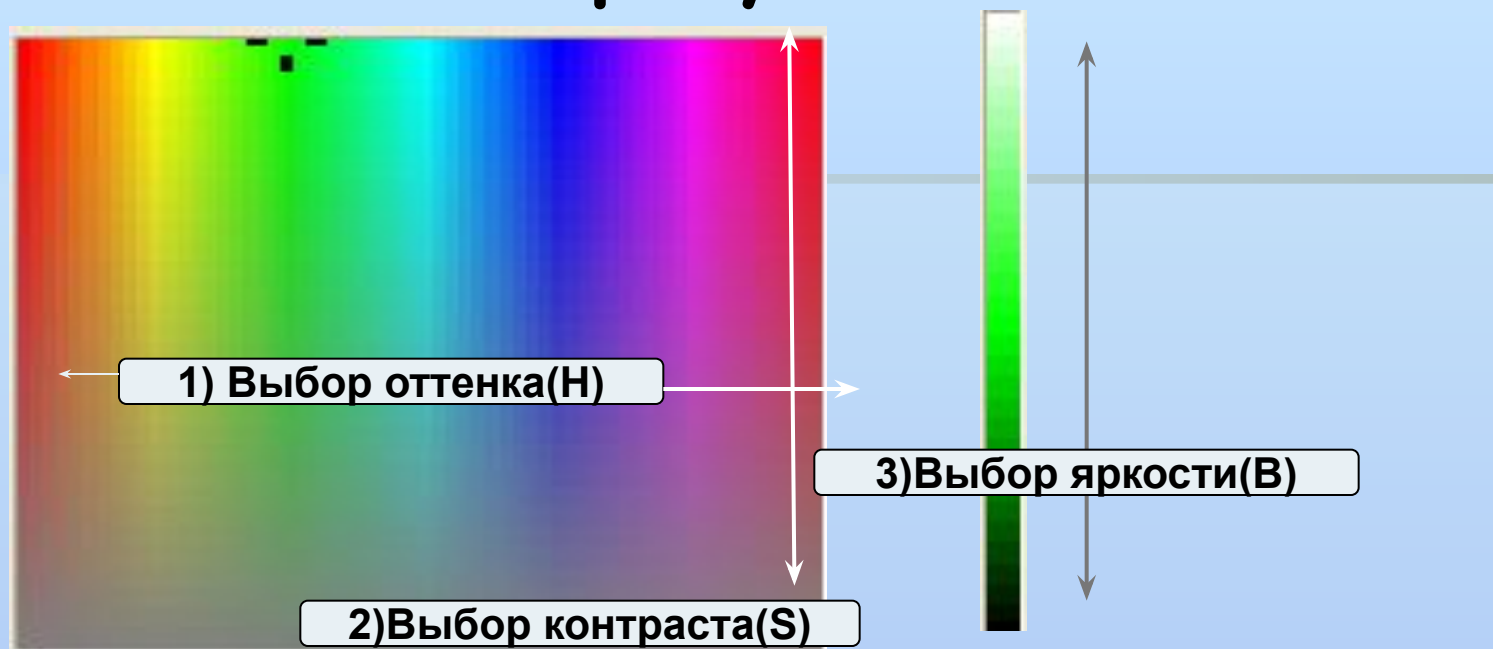


Перемещение движка цвета по горизонтали меняет оттенок(H), по вертикали- контрастность(S). Перемещение треугольного движка (по отдельной вертикальной линейке) меняет яркость(V).

Оттенок(тон) – это цвет на радуге. Контрастность (насыщенность)-это содержание в цвете серой примеси. Цвет максимальной насыщенности не содержит серого вообще, а при нулевой насыщенности-все цвета серые.

Яркость-это интенсивность, с которой излучается цвет. При максимальной яркости все цвета превращаются в белый цвет, при нулевой - в черный.

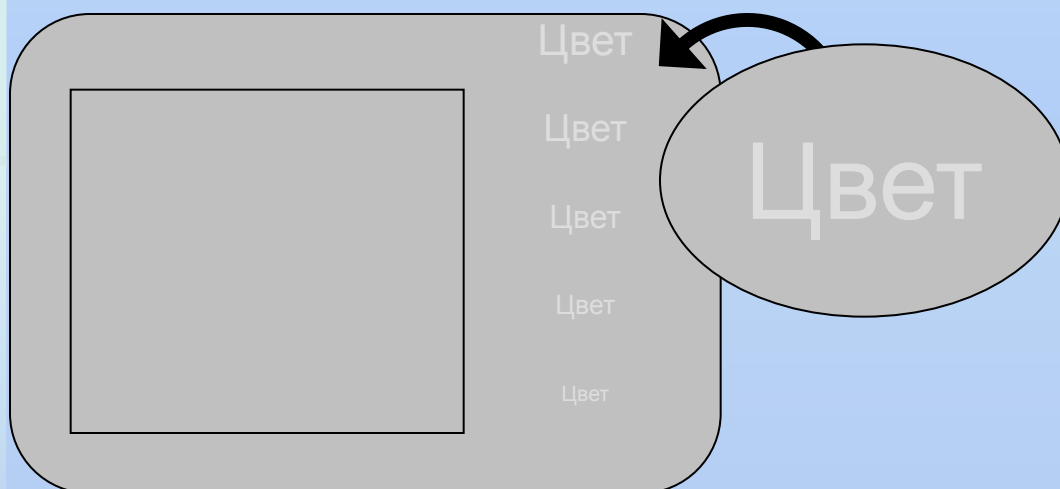
# Алгоритм подбора цвета показан на рисунке



Сначала выбирается цвет на радуге(слева направо), потом устанавливается его контрастность(сверху вниз), а затем отдельным движком задается яркость.

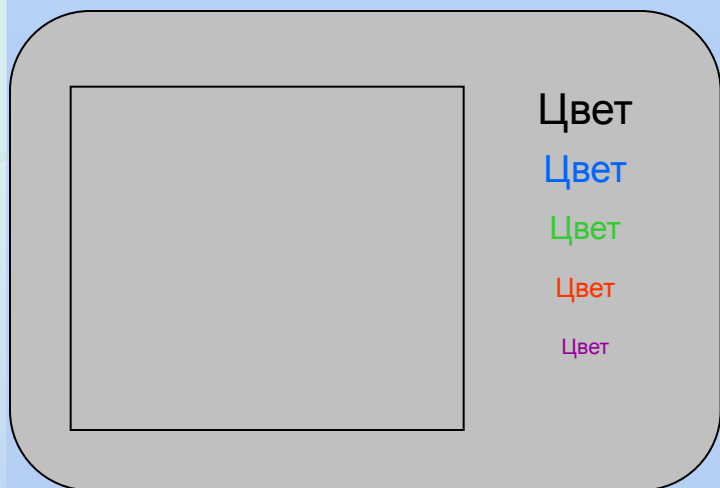
# Рекомендации по работе с ЦВЕТОМ

Человеческий глаз устроен таким образом, что одни и те же цвета могут восприниматься по-разному. При уменьшении площади, которую занимает цвет, оттенки становятся малоразличимыми, и цвета визуально «теряют» насыщенность. Это особенно хорошо заметно на цветных текстах

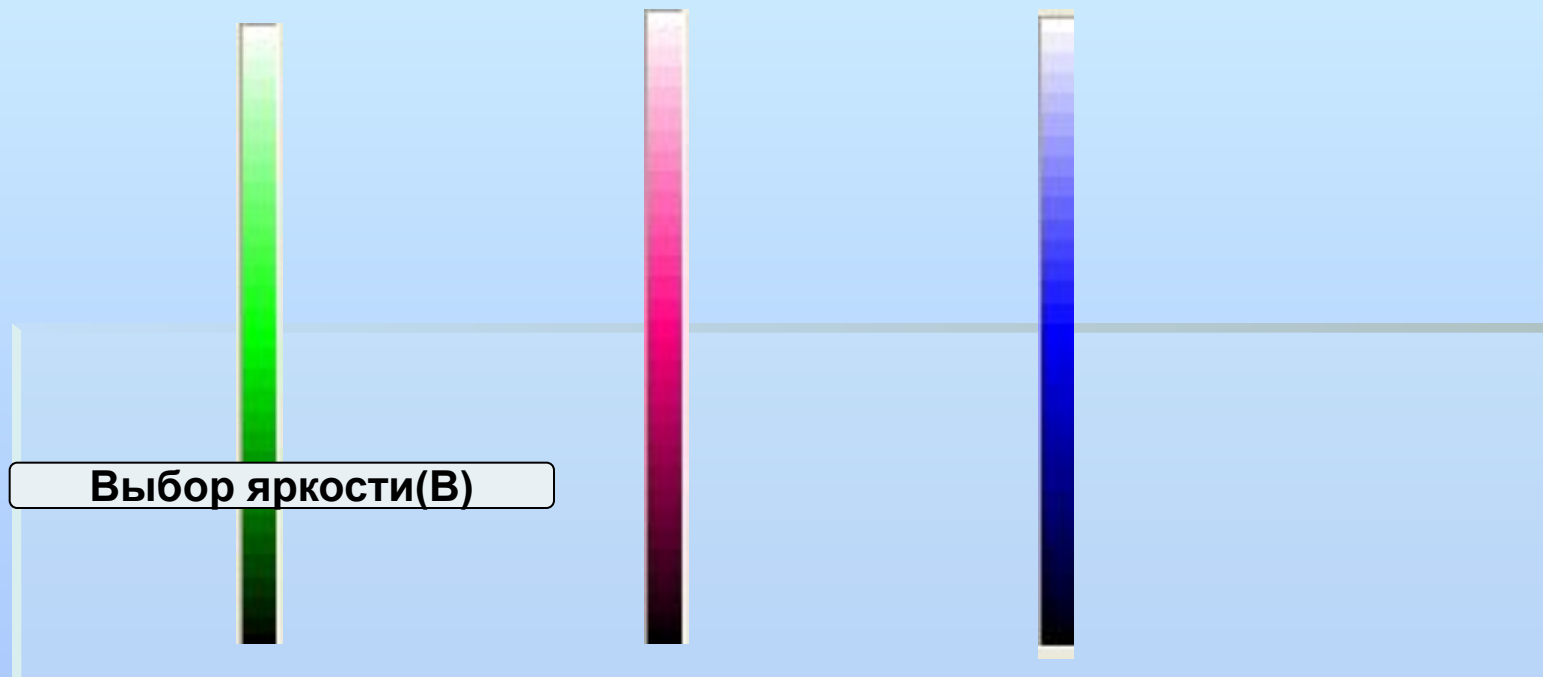


**Буквы того же цвета, что и прямоугольник**

Практическая рекомендация: для объектов с маленькими закрашенными площадями нужно выбирать яркие цвета (6 основных цветов на радуге плюс белый и черный цвета), а для сложных цветов со слаборасшищенным тоном - использовать большие площади



**Буквы ярких цветов**



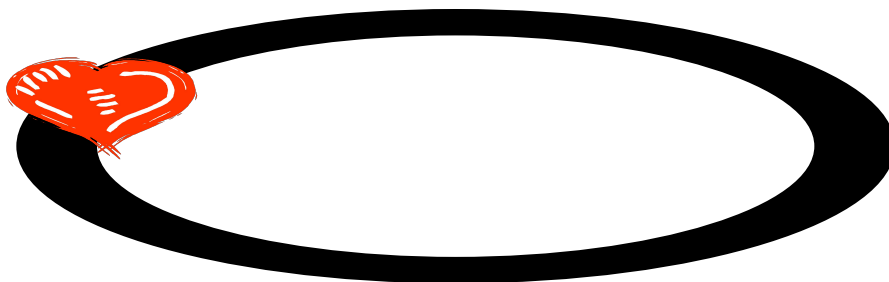
Выбор яркости(В)

Черный и белый цвета являются особыми цветами. Их нет в радуге, а в то же время любой цвет превращается в черный при уменьшении яркости до нуля и в белый при увеличении яркости до максимального значения. Эти цвета наиболее часто используются в дизайне, так как они хорошо сочетаются с большинством других цветов.

# Сочетаемость цветов

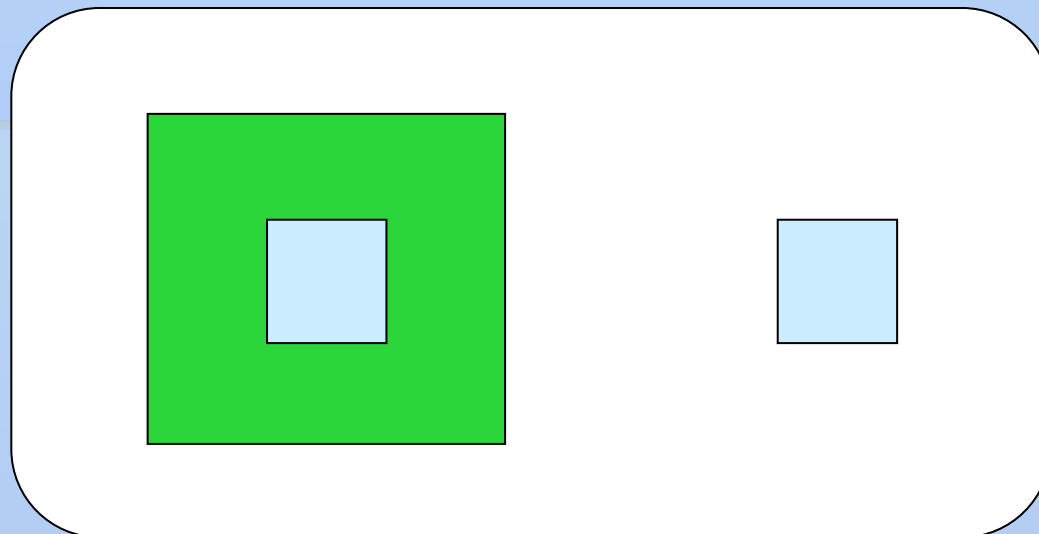
Если вы чувствуете себя не совсем уверенно в цветовом пространстве - возьмите за основу черный и белый цвета, добавьте к ним красный - эта палитра вас не подведет

Черный + белый + красный = отличная сочетаемость



# Визуальное изменение цвета

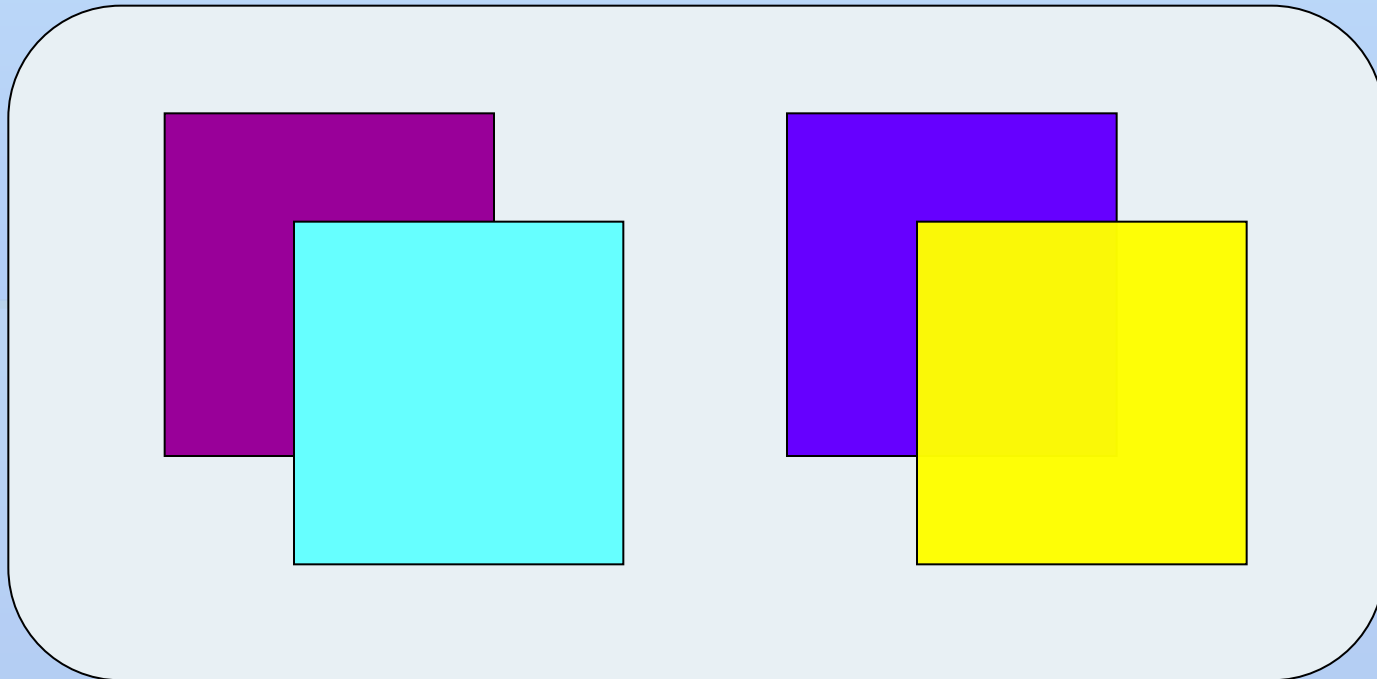
Цвета расположенные рядом, выглядят несколько иначе, чем взятые по отдельности, кроме того, вид цвета зависит от площади, которую он занимает. Например, маленький объект тускло-синего цвета внутри большого яркого-зеленого приобретает зеленый оттенок.





# Изменение цвета на границах областей

На границах цвета меняют свой вид. Темный цвет рядом со светлым становится еще темнее, а светлый – ярче.



# Литература.

- А.А.Дуванов «Основы web-дизайна и школьного «сайтостроительства»  
Газета «Информатика», 2005 г, №21
- А.Г.Гейн «Информатика» , 2001г