



Информатика Представление информации в компьютере



Содержание лекции





Представление чисел



Целые числа

Неотрицательные целые числа представляются в двоичной системе счисления.

Диапазон целых чисел, которые можно представить через n -битную последовательность составляет

- для чисел без знака $0 \dots 2^n - 1$
- для чисел со знаком $-2^{(n-1)} \dots 2^{(n-1)} - 1$

Например, 1 байт (8 бит) $0 \dots 255$ или $-128 \dots 127$

2 байта (16 бит) $0 \dots 65\,535$ или $-32\,768 \dots 32\,767$

4 байта (32 бита) $0 \dots 4\,294\,967\,296$ или $-2\,147\,483\,648 \dots 2\,147\,483\,647$



Отрицательные целые числа

- Для представления отрицательного числа используют дополнительный код.
- Дополнительный код заменяет операцию вычитания на операцию сложения.

Дополнительный код для отрицательного числа можно получить инвертированием его представления (первое дополнение) и прибавлением к полученной инверсии единицы (второе дополнение).

Это эквивалентно вычитанию абсолютной величины числа из нуля.



Пример. Отрицательные целые числа

Пример:

— 6_{10} представить в дополнительном коде, используя 4 байта.

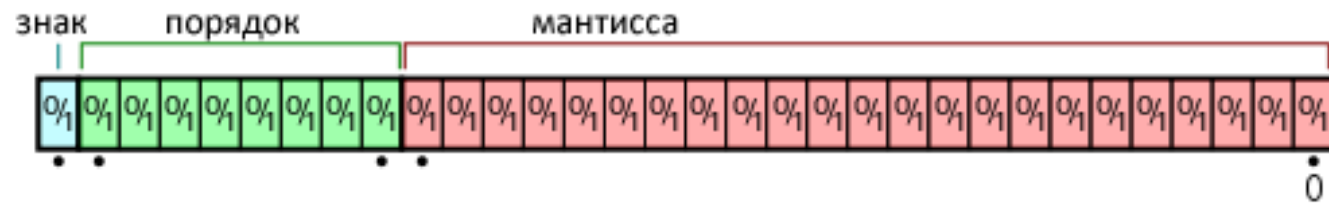
$$6_{10} = 0110_2$$

инверсия (1-е дополнение): 1001

2-е дополнение: 1010 (прибавили 1)



Идея представление числа с плавающей точкой состоит из набора отдельных разрядов, условно разделенных на знак, экспоненту порядок и мантиссу:





Числа с плавающей точкой



Представление определяет число по формуле $(-1)^s \cdot M \cdot B^E$, где s — знак, M — мантисса, B — основание, E — порядок.

Знак — это 0 или 1 (определяет знак числа, "+" или "-" соотв.)

Мантисса — это целое число фиксированной длины, которое представляет старшие разряды действительного числа.

Порядок — это показатель степени базы.

В качестве базы используется $B = 2$ или $B = 10$.

Представление чисел с базой 2 наиболее устойчивы к ошибкам округления.



Числа с плавающей точкой

Дробные числа преобразуются иными методами, а именно умножением. Умножению подвергаем только дробную часть до тех пор пока она не исчезнет.

Продemonстрируем преобразование числа $0,125_{10}$ и $0,1_{10}$ в двоичную систему счисления:

$$\begin{array}{r} \times \quad 0,125 \\ \hline \quad 2 \\ \hline \times \quad 0,250 \\ \hline \quad 2 \\ \hline \times \quad 0,500 \\ \hline \quad 2 \\ \hline 1,000 \end{array}$$

$$\begin{array}{r} \times \quad 0,1 \\ \hline \quad 2 \\ \hline \times \quad 0,2 \\ \hline \quad 2 \\ \hline \times \quad 0,4 \\ \hline \quad 2 \\ \hline \times \quad 0,8 \\ \hline \quad 2 \\ \hline \times \quad 1,6^* \\ \hline \quad 2 \\ \hline \times \quad 1,2 \\ \hline \quad 2 \\ \hline 0,4 \end{array}$$



Числа с плавающей точкой

- Разрядами двоичного числа будут целые части чисел, образующихся при умножении. Первый «ноль» в исходном числе это «ноль» в целой части, а остальные - разряды двоичного числа. Поэтому имеем $0,125_{10} \rightarrow 0,001_2$. Во втором случае наблюдается закливание расчетов, таким образом, расчеты дают двоичное число с бесконечной дробной частью (иррациональное число) которое записывается следующим образом $0,000110011(0011)...$
- Поскольку в памяти компьютера все числа могут быть только рациональными, то приходится полученное число округлять до заданной точности (необходимое число разрядов в ячейках памяти компьютера), что приводит к ошибке. Таким образом, не все дробные числа, вводимые в десятичной системе счисления, представляются абсолютно точно в памяти компьютера.



Пример. Числа с плавающей точкой

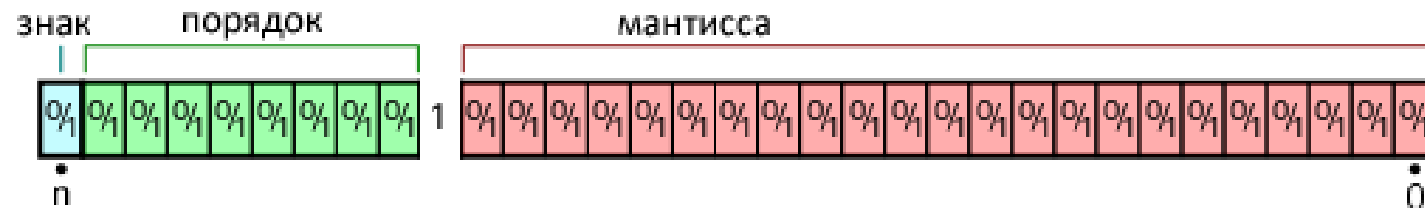
Число 1.5 с плавающей точкой представляют как $1.1e-2$ (для $B = 2$): Мантисса $110_2 = 6_{10}$. $1.5 = 6 \div 4 = 6 \times 2^{-2}$.

В таком представлении $s = 0$, $M = 110$, $E = -2$.

Недостаток: этот подход дает неоднозначность представления:

$$1.1e-2 = 0.11e-3 = 11e-1$$

Поэтому уже в первых реализациях чисел с плавающей точкой в компьютере использовали **нормализованное** представление, в котором первый бит мантиссы всегда положительный, а сама мантисса считалась числом от 1 до 2 (для $B = 2$):





Числа с плавающей точкой

Достоинства:

- устраняется неоднозначность представления
- экономит 1 бит

Недостаток: невозможно представить 0.

То есть в нормализованном виде число имеет вид $(-1)^s \cdot 1.M \cdot B^E$.



формат IEEE754-80

знак порядок (8 бит) мантисса (23 бит)

0 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 = 0.15625

31 30 23 22 0

Порядок $E = 01111100_2 = 124_{10}$.

Эффективный порядок равен $124_{10} - 127_{10} = -3_{10}$.

Мантисса $M = 1.01$ (первая единица неявная)

В результате число равно

$$F = 1.01_2 e-3 = (2 + 2^{-2}) \cdot 2^{-3} = 2^{-3} + 2^{-5} = 0.125 + 0.03125 = 0.15625$$

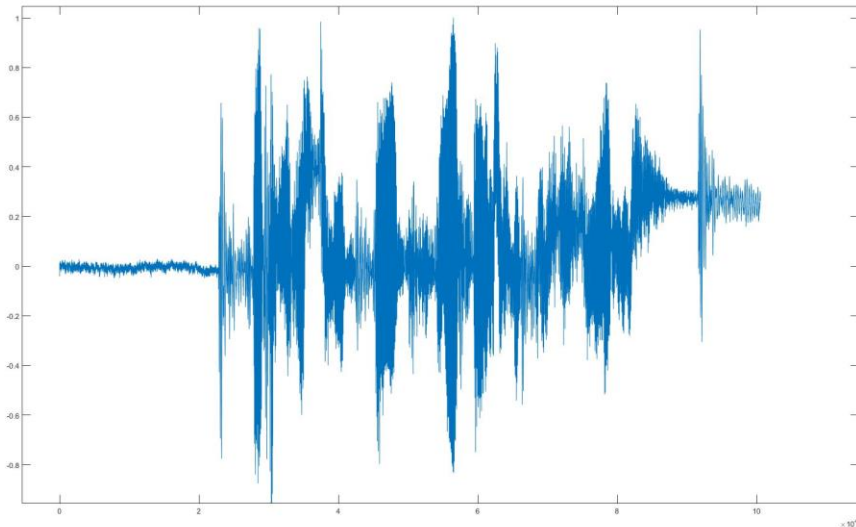


Представление изображений и звука



Сигнал

- Сигнал – скалярная функция от одного или нескольких аргументов



$s(t)$ – звук



$f(x,y)$ – изображение



Виды сигналов

- Аналоговые (непрерывные)
 - звук в воздухе или в проводе, идущем от микрофона
 - изображение (до ввода в компьютер)
 - запись показаний датчика
- Цифровые (дискретные)
 - звук в компьютере (одномерный массив чисел)
 - изображение в компьютере (двумерный массив чисел)
 - запись показаний датчика в компьютере (одномерный массив)



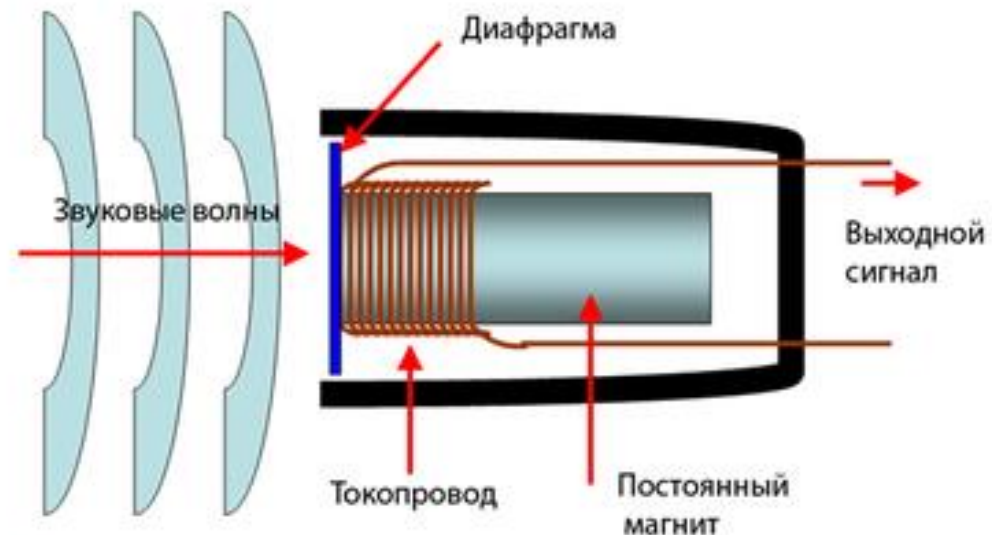
Оцифровка

- Цифровой сигнал — сигнал данных, у которого каждый из представляющих параметров описывается функцией дискретного времени и конечным множеством возможных значений
- Оцифр^овка (англ. digitization) — описание объекта, изображения или аудио- видеосигнала (в аналоговом виде) в виде набора дискретных цифровых замеров (выборок) этого сигнала/объекта, при помощи той или иной аппаратуры, т. е. перевод его в цифровой вид, пригодный для записи на электронные носители
- Кодирование информации — процесс преобразования сигнала из формы, удобной для непосредственного использования информации, в форму, удобную для передачи, хранения или автоматической переработки



Оцифровка звука

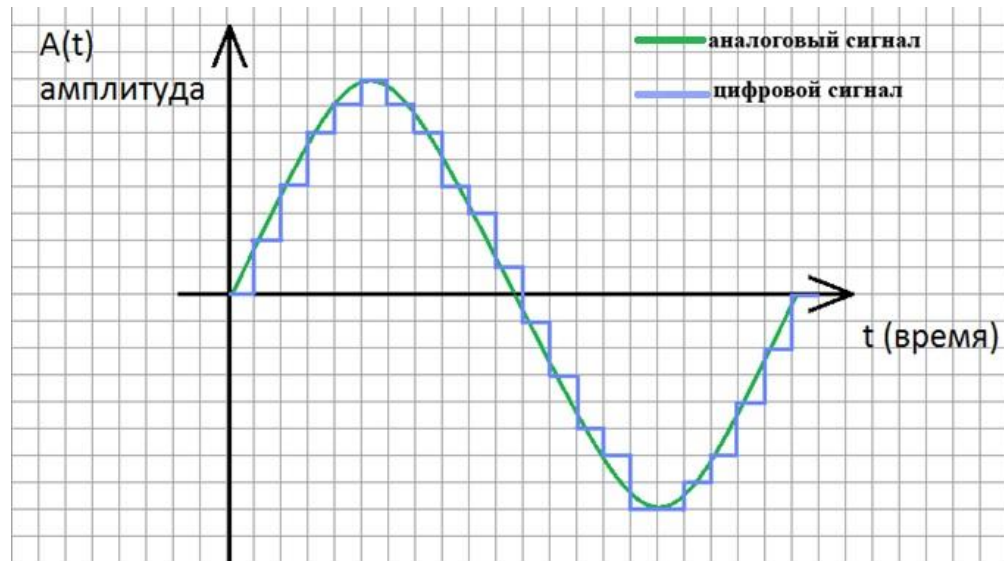
Давление звуковых колебаний воздуха, воды или твёрдого вещества действует на тонкую мембрану микрофона. В свою очередь, колебания мембраны возбуждают электрические колебания; в зависимости от типа микрофона для этого используются явление электромагнитной индукции, изменение ёмкости конденсаторов или пьезоэлектрический эффект





Аналого-цифровое преобразование

- Для преобразования электрического сигнала в цифровой используется временная дискретизация:
- Аналоговый сигнал разбивается на отдельные маленькие временные участки и для каждого такого участка устанавливается определенная величина амплитуды.





Частота дискретизации

Через равные промежутки времени мы измеряем уровень аналогового сигнала. Количество таких измерений за одну секунду называется частотой дискретизации.

Частота дискретизации измеряется в Герцах.

Примеры используемых частот дискретизации:

8 000 Гц	— телефон, достаточно для речи;
11 025 Гц	— для качественной передачи речи;
22 050 Гц	— радио;
44 100 Гц	— Audio CD;
48 000 Гц	— DVD, DAT;
96 000 Гц	— DVD-Audio (MLP 5.1);
192 000 Гц	— DVD-Audio (MLP 2.0);
2 822 400 Гц	— SACD Super audio CD 5.1



Глубина кодирования звука

- Глубина кодирования звука (разрядность преобразования) — это количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.
- Если известна глубина кодирования I в битах, то количество уровней громкости цифрового звука можно рассчитать по формуле $N = 2^I$.
- Как правило, используется глубина кодирования 8, 16 или 24 бита.

Пример.

При глубине кодирования 8 бит количество уровней громкости составляет

$$N = 2^8 = 256$$

В процессе кодирования каждому уровню громкости звука присваивается свой 8-битовый двоичный код.

Наименьшему уровню звука будет соответствовать код 00000000, а наибольшему — 11111111.



Качество цифрового звука

Для передачи многоканального звука каждый канал кодируется отдельно.

Очевидно, что чем выше частота дискретизации и чем больше глубина кодирования, тем качественнее будет цифровой звук.

Для общей оценки качества используют **битрейт** (bitrate) — количество бит, используемых для хранения одной секунды мультимедийного контента.

Для звука без компрессии (сжатия):

$$\text{битрейт} = \text{частота дискр.} \times \text{глубина код.} \times \text{число каналов}$$

Чаще всего битрейт звука и видео измеряют в килобитах в секунду.



Битрейт

В форматах потокового аудио (например, MP3), использующих сжатие с потерей качества, битрейт выражает степень сжатия потока и, тем самым, определяет размер канала, для которого сжат поток данных.

Существует три режима сжатия потоковых данных:

- CBR (англ. Constant bitrate) — с постоянным битрейтом;
- VBR (англ. Variable bitrate) — с переменным битрейтом;
- ABR (англ. Average bitrate) — с усреднённым битрейтом.

Наиболее часто используемые значения битрейта в MP3:

- 32 кбит/с — для речи
- 96 кбит/с — для передачи речи или звука низкого качества
- 128 или 160 кбит/с — начальный уровень кодирования музыки
- 192 кбит/с — приемлемое качество кодирования музыки
- 256 кбит/с — высокое качество кодирования музыки
- 320 кбит/с — наивысшее качество кодирования, поддерживаемое MP3



Аудиоформаты

Формат звукового файла определяет структуру и особенности представления звуковых данных.

- аудиоформаты без сжатия (например, WAV)
- аудиоформаты со сжатием без потерь (APE, FLAC)
- аудиоформаты со сжатием с потерями (MP3, Ogg)
- **WAV** (от англ. *waveform* — «в форме волны») — используется для хранения **несжатого звука** в импульсно-кодовой модуляции. Не налагает каких-либо ограничений на используемый алгоритм кодирования.
- **FLAC** (англ. *Free Lossless Audio Codec*) — популярный свободный кодек, предназначенный для **сжатия аудиоданных без потерь**.
- **MP3** (MPEG Layer 3) — кодек третьего уровня, разработанный командой MPEG (Moving Picture Expert Group). Использует **алгоритм сжатия с потерями**, разработанный для существенного уменьшения размера данных, необходимых для воспроизведения звука, близкого к оригинальному, но с ощутимыми потерями качества при прослушивании на качественной звуковой системе.



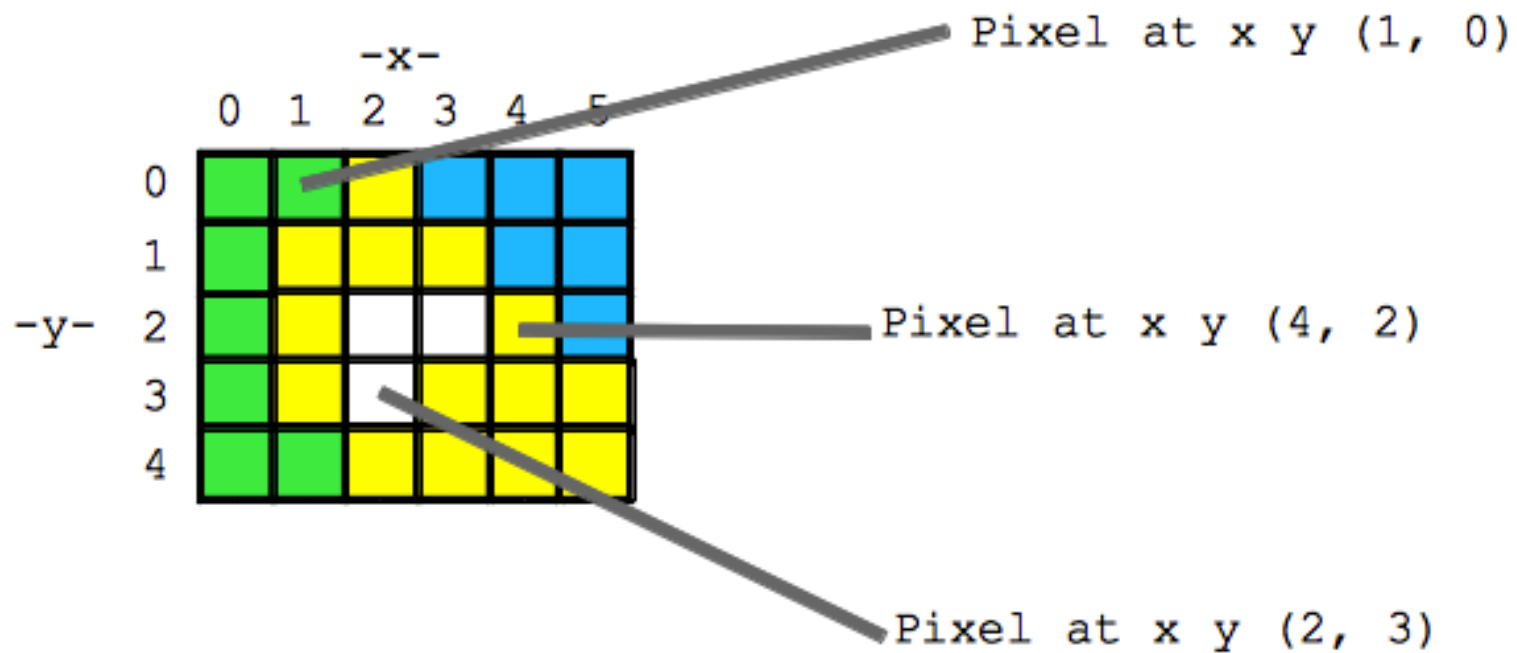
Представления изображений

- Растровые
- Векторные





Растровая графика





Цветовой диапазон и глубина цвета

- Цветовой диапазон – это максимальное число цветов (N) используемых при создании изображения
- Глубина цвета – это число бит (n), используемых для представления цвета каждого отдельного пикселя изображения, определяемое цветовым диапазоном

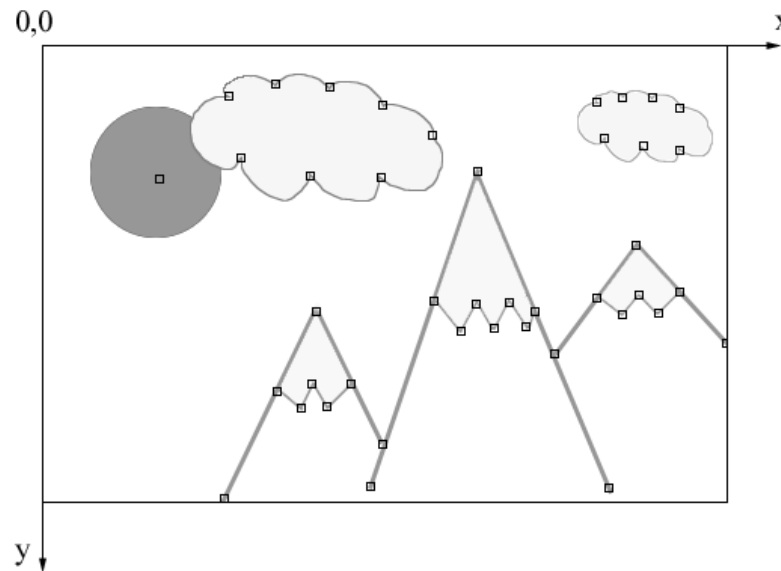
$$n = \log_2(N)$$

- Пример. 256 цветов – 1 байт (8 бит)



Векторная графика

Способ представления изображений в виде совокупности отдельных объектов (графических примитивов). Каждый примитив описывается математически относительно его узлов.



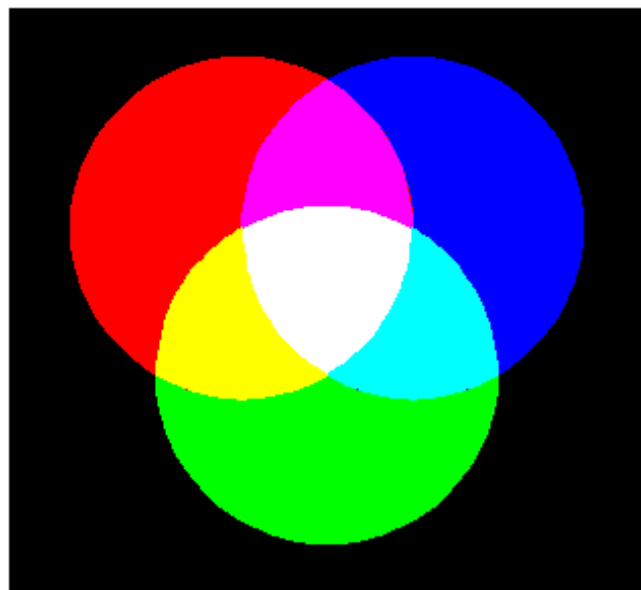


Цветовые модели

- аппаратно-зависимые цветовые пространства (RGB, CMYK)
- аппаратно-независимые цветовые пространства (CIE XYZ, sRGB)



RGB



0..255

24 бит

16,7 млн.

красный

зеленый

синий



Red

Green

Blue

RGB



(0, 0, 0)



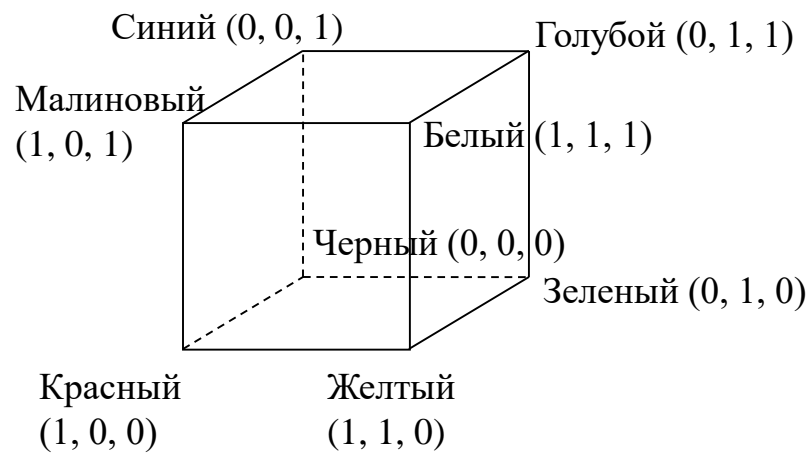
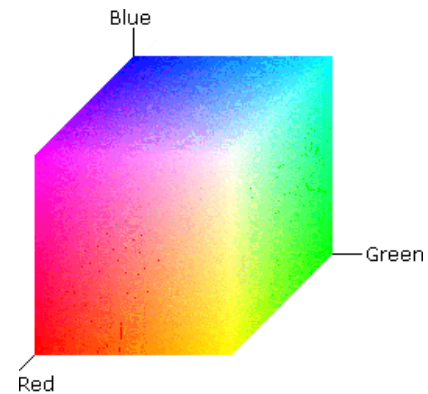
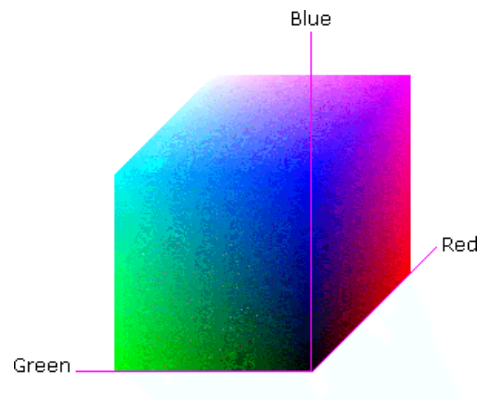
(255, 255, 255)



(100, 0, 100)

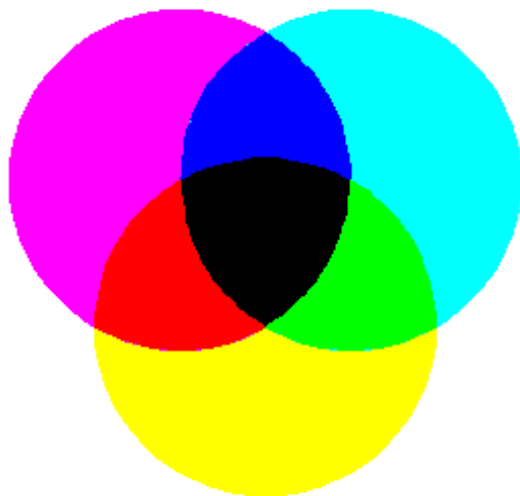


RGB





СМУК



0..255

24 бит

16,7 млн.

голубой

розовый

желтый



Cyan

Magenta

Yellow

СМУК



(0, 0, 0)



(255, 255, 255)



(100, 0, 100)

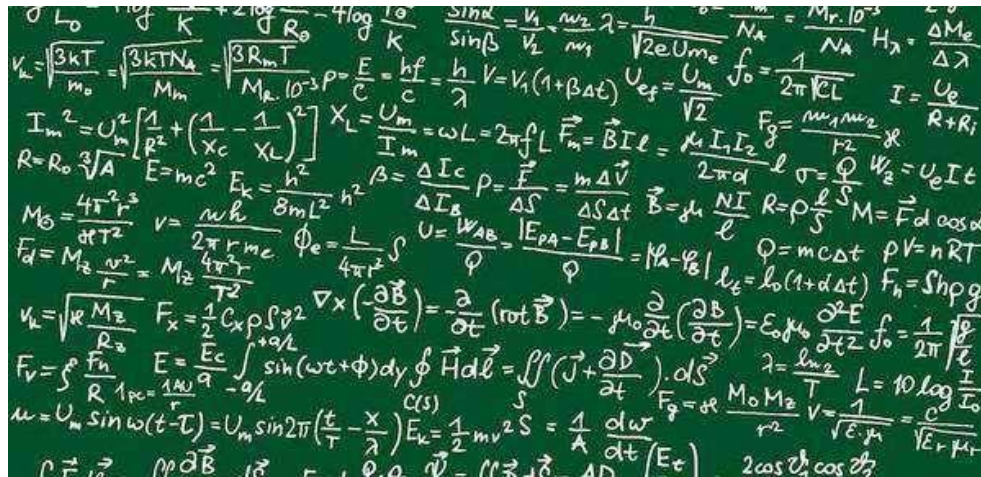


Представление текста

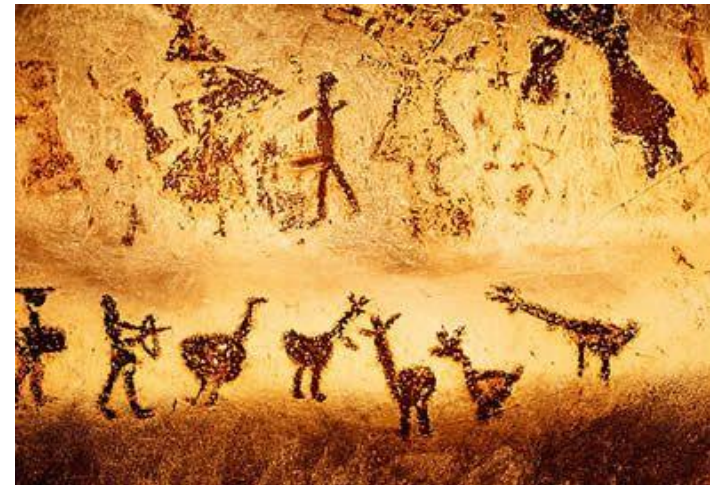


Значение текста

Лекция преподавателя:



Конспект студента:





Значение текста

- Текст представлен в виде набора символов
- Символы группируются в алфавиты
- В математике алфавит – множество символов
- Символ представляется в виде кодового значения из набора кодовых точек

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33



ASCII

- American Standard Code for Information Interchange
- 7 бит → 128 кодовых точек
- 95 знаков
- 0..31 и 127 – управляющие символы

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



Unicode

- Юникод поддерживает практически все существующие наборы символов. В стандарте представлены три различные формы кодирования символов Юникода: 8, 16 и 32-битными блоками. Соответственно, они носят название UTF-8, UTF-16 и UTF-32.
- Каждая из форм кодирования Юникода разработана с учётом недопустимости частичного наложения (значения ведущей, замыкающей и одиночной единицы хранения информации не совпадают и каждый символ имеет чётко определяемые границы).



<http://bit.ly/RUTINFORMCOURSE2019>



Есть вопросы?

miitqa@outlook.com