

*Инновационный Евразийский
Университет*

**Слайд-лекции по дисциплине
«ИНФОРМАТИКА»**

Измерение информации

Разработала ст.преподаватель Айтуллина Б.А.

Измерение информации

Содержательный подход к измерению информации. Сообщение – информативный поток, который в процессе передачи информации поступает к приемнику. Сообщение несет информацию для человека, если содержащиеся в нем сведения являются для него новыми и понятными. Информация - знания человека ? сообщение должно быть информативно. Если сообщение не информативно, то количество информации с точки зрения человека = 0. (Пример: вузовский учебник по высшей математике содержит знания, но они не доступны 1-класснику)

Измерение информации

Алфавитный подход к измерению информации не связывает кол-во информации с содержанием сообщения. Алфавитный подход - объективный подход к измерению информации. Он удобен при использовании технических средств работы с информацией, т.к. не зависит от содержания сообщения. Кол-во информации зависит от объема текста и мощности алфавита. Ограничений на max мощность алфавита нет, но есть достаточный алфавит мощностью 256 символов. Этот алфавит используется для представления текстов в компьютере. Поскольку $256=2^8$, то 1 символ несет в тексте 8 бит информации.

Измерение информации

Вероятностный подход к измерения информации. Все события происходят с различной вероятностью, но зависимость между вероятностью событий и количеством информации, полученной при совершении того или иного события можно выразить формулой которую в 1948 году предложил Шеннон.

Формула Шеннона

$$I = -\sum_{i=1}^N p_i \log_2 p_i$$

I - количество информации

p_i – вероятности отдельных событий

N – количество возможных событий

Количество информации достигает max значения, если события равновероятны, поэтому количество информации можно рассчитать по формуле

$$i = \log_2 N \text{ (бит)}$$

Единицы измерения

1 бит (*binary digit, двоичная цифра*) – это количество информации, которое мы получаем при выборе одного из двух возможных вариантов (вопрос: «Да» или «Нет»?)

1 байт (*byte*) = **8** бит

1 Кб (килобайт) = **1024** байта

1 Мб (мегабайт) = **1024** Кб

1 Гб (гигабайт) = **1024** Мб

1 Тб (терабайт) = **1024** Гб

1 Пб (петабайт) = **1024** Тб

2¹⁰

Информационный вес символа произвольного алфавита

Информационный вес символа алфавита i и мощность алфавита N связаны между собой соотношением: $N = 2^i$.

Информационный объём сообщения
Информационный объём сообщения (количество информации в сообщении), представленного символами естественного или формального языка, складывается из информационных весов составляющих его символов.

Информационный объём сообщения I равен произведению количества символов в сообщении K на информационный вес символа алфавита i ;

$$I = K * i.$$

Единицы измерения

Информационный объем текста складывается из информационных весов составляющих его символов.



Информационный объем текста:

$$1 \text{ бит} \times 25 \text{ символов} = 25 \text{ бит}$$

Информационный вес символа произвольного алфавита

Задача. Сообщение, записанное буквами 32-символьного алфавита, содержит 140 символов. Какое количество информации оно несёт?

Решение.

$$N = 32$$

$$K = 140$$

$I = ?$

$$I = K * i$$

$$N = 2^i$$

$$32 = 2^i, i = 5,$$

$$I = 140 * 5 = 700 \text{ (битов)}$$

Ответ: 700 битов.

Задача. Информационное сообщение объёмом 720 битов состоит из 180 символов. Какова мощность алфавита, с помощью которого записано это сообщение?

Решение.

$$I = 720$$

$$K = 180$$

$$N = ?$$

$$N = 2^i,$$
$$I = K * i,$$
$$i = I / K$$

$$i = 720 / 180 = 4 \text{ (бита)}$$
$$N = 2^4 = 16 \text{ (символов)}$$

Ответ: 16 символов.

Задача. Информационное сообщение объёмом 4 Кбайта состоит из 4096 символов. Каков информационный вес символа используемого алфавита? Сколько символов содержит алфавит, с помощью которого записано это сообщение?

Решение.

$$I = 4 \text{ Кб}$$

$$K = 4096$$

$$i - ?$$

$$N - ?$$

$$I = 4 \text{ Кб} = 4 * 1024 * 8 \text{ битов}$$

$$N = 2^i$$

$$I = K * i$$

$$i = I / K$$

$$i = 4 * 1024 * 8 / 4096 = 8 \text{ битов}$$

$$N = 2^8 = 256 \text{ символов}$$

Ответ: 8 битов, 256 символов.

Кодирование текстовой информации

Сначала применялась 7-битная кодировка, которая могла представить 128 символов. С распространением *IBM PC* международным стандартом стала таблица кодировки *ASCII* (*American Standard Code for Information Interchange*) – Американский стандартный код для

информационного обмена

Таблица кодировки *ASCII*

Позже она была расширена до 8 бит (256 символов). При этом первая половина (символы 0-127) были всегда одни и те же, соответствующие стандарту *ASCII*, а вторая половина таблицы (символы 128-255) менялась в зависимости от страны, где она использовалась.

символ	10- й код	2-й код	символ	10- й код	2-й код	символ	10-й код	2-й код	символ	10-й код	2-й код
	32	00100000	8	56	00111000	P	80	01010000	h	104	01101000
!	33	00100001	9	57	00111001	Q	81	01010001	i	105	01101001
"	34	00100010	:	58	00111010	R	82	01010010	j	106	01101010
#	35	00100011	;	59	00111011	S	83	01010011	k	107	01101011
\$	36	00100100	<	60	00111100	T	84	01010100	l	108	01101100
%	37	00100101	=	61	00111101	U	85	01010101	m	109	01101101
&	38	00100110	>	62	00111110	V	86	01010110	n	110	01101110
'	39	00100111	?	63	00111111	W	87	01010111	o	111	01101111
(40	00101000	@	64	01000000	X	88	01011000	p	112	01110000
)	41	00101001	A	65	01000001	Y	89	01011001	q	113	01110001
*	42	00101010	B	66	01000010	Z	90	01011010	r	114	01110010
+	43	00101011	C	67	01000011	[91	01011011	s	115	01110011
,	44	00101100	D	68	01000100	\	92	01011100	t	116	01110100
-	45	00101101	E	69	01000101]	93	01011101	u	117	01110101
.	46	00101110	F	70	01000110	^	94	01011110	v	118	01110110
/	47	00101111	G	71	01000111	_	95	01011111	w	119	01110111
0	48	00110000	H	72	01001000	`	96	01100000	x	120	01111000
1	49	00110001	I	73	01001001	a	97	01100001	y	121	01111001
2	50	00110010	J	74	01001010	b	98	01100010	z	122	01111010
3	51	00110011	K	75	01001011	c	99	01100011	{	123	01111011
4	52	00110100	L	76	01001100	d	100	01100100		124	01111100
5	53	00110101	M	77	01001101	e	101	01100101	}	125	01111101
6	54	00110110	N	78	01001110	f	102	01100110	~	126	01111110
7	55	00110111	O	79	01001111	g	103	01100111	□	127	01111111

символ	10-Б код	2-Б код	символ	10-Б код	2-Б код	символ	10-Б код	2-Б код	символ	10-Б код	2-Б код
Ъ	128	10000000		160	10100000	А	192	11000000	а	224	11100000
Ґ	129	10000001	Ү	161	10100001	Б	193	11000001	б	225	11100001
,	130	10000010	Ѹ	162	10100010	В	194	11000010	в	226	11100010
ѓ	131	10000011	Ј	163	10100011	Г	195	11000011	г	227	11100011
„	132	10000100	о	164	10100100	Д	196	11000100	д	228	11100100
…	133	10000101	Ґ	165	10100101	Е	197	11000101	е	229	11100101
†	134	10000110		166	10100110	Ж	198	11000110	ж	230	11100110
‡	135	10000111	§	167	10100111	З	199	11000111	з	231	11100111
€	136	10001000	Є	168	10101000	И	200	11001000	и	232	11101000
‰	137	10001001	©	169	10101001	Й	201	11001001	й	233	11101001
Љ	138	10001010	€	170	10101010	К	202	11001010	к	234	11101010
<	139	10001011	«	171	10101011	Л	203	11001011	л	235	11101011
Њ	140	10001100	¬	172	10101100	М	204	11001100	м	236	11101100
Ќ	141	10001101	-	173	10101101	Н	205	11001101	н	237	11101101
Ћ	142	10001110	®	174	10101110	О	206	11001110	о	238	11101110
Ќ	143	10001111	Ї	175	10101111	П	207	11001111	п	239	11101111
ђ	144	10010000	о	176	10110000	Р	208	11010000	р	240	11110000
‘	145	10010001	±	177	10110001	С	209	11010001	с	241	11110001
’	146	10010010	І	178	10110010	Т	210	11010010	т	242	11110010
“	147	10010011	і	179	10110011	У	211	11010011	у	243	11110011
”	148	10010100	г	180	10110100	Ф	212	11010100	ф	244	11110100
•	149	10010101	μ	181	10110101	Х	213	11010101	х	245	11110101
—	150	10010110	¶	182	10110110	Ц	214	11010110	ц	246	11110110
—	151	10010111	·	183	10110111	Ч	215	11010111	ч	247	11110111
□	152	10011000	ё	184	10111000	Ш	216	11011000	ш	248	11111000
™	153	10011001	№	185	10111001	Щ	217	11011001	щ	249	11111001
љ	154	10011010	€	186	10111010	Ъ	218	11011010	ъ	250	11111010
›	155	10011011	»	187	10111011	Ы	219	11011011	ы	251	11111011
њ	156	10011100	ј	188	10111100	Ь	220	11011100	ь	252	11111100
ќ	157	10011101	§	189	10111101	Э	221	11011101	э	253	11111101
ћ	158	10011110	ѕ	190	10111110	Ю	222	11011110	ю	254	11111110
џ	159	10011111	ї	191	10111111	Я	223	11011111	я	255	11111111

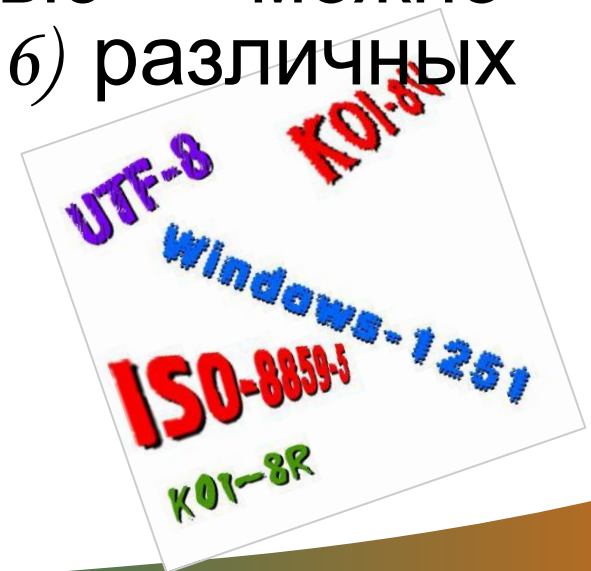
Таблица
расширенного
кода ASCII
Кодировка
Windows-1251
(CP1251)

Кодовые таблицы для русских

букв

В настоящее время существует 5 разных кодовых таблиц для русских букв (KOI8, CP 1251, CP 866, Mac, ISO).

Широкое распространение получил новый международный стандарт *Unicode*, который отводит на каждый символ два байта. С его помощью можно закодировать 65536 ($2^{16} = 65536$) различных символов.



Проблемы с кодировками

Проблемы с кодировками делятся на несколько типов. Первый тип - это отсутствие информации о кодировке.

ПРОБЛЕМЫ С КОДИРОВКАМИ

Проблемы второго типа - это когда кодировка в файле указана, но конечная программа такой кодировки не знает.

Проблемы с кодировками

Третий тип проблем, наоборот, связан с избытком информации о кодировках.

Это актуальная в настоящее время проблема (например, для веб-страниц).

Обратите внимание!

Цифры кодируются по стандарту *ASCII* в двух случаях – при вводе-выводе и когда они встречаются в тексте. Если цифры участвуют в вычислениях, то осуществляется их преобразование в другой двоичный код.



Возьмем число *57*.

При использовании в тексте каждая цифра будет представлена своим кодом в соответствии с таблицей *ASCII*. В двоичной системе это – *0011010100110111*.

При использовании в вычислениях, код этого числа будет получен по правилам перевода в двоичную систему и получим – *00111001*.

Компьютерные редакторы, в основном, работают с алфавитом размером 256 символов.

В этом случае легко подсчитать объем информации в тексте. Если 1 символ алфавита несет 1 байт информации, то надо просто сосчитать количество символов; полученное число даст информационный объем текста в байтах.

$$I = \mathcal{K} \times i, \text{ где}$$

I -информационный объем
сообщения

\mathcal{K} - количество символов в тексте

i - информационный вес одного
символа

$$2^i = \mathcal{N}$$

\mathcal{N} - мощность алфавита

Задачи: текст

Сколько места в памяти надо выделить для хранения предложения

Привет, друг!

- считаем все символы, включая знаки препинания (здесь **13** символов)
- если нет дополнительной информации, то считаем, что 1 символ занимает **1 байт**
- в кодировке UNICODE 1 символ занимает **2 байта**

Ответ: 13 байт или 104 бита

(в UNICODE: 26 байт или 208 бит)

Задачи: текст

Сколько места надо выделить для хранения 10 страниц книги, если на каждой странице помещаются 32 строки по 64 символа в каждой?

Решение:

- на 1 странице $32 \cdot 64 = 2048$ символов
- на 10 страницах $10 \cdot 2048 = 20480$ символов
- каждый символ занимает 1 байт

Ответ:

- 20480 байт или ...
- $20480 \cdot 8$ бит или ...
- $20480 : 1024$ Кб = 20 Кб

Задачи: кодирование

Объем сообщения, содержащего 1024 символов, составил 1/512 часть мегабайта. Какова мощность алфавита, с помощью которого записано сообщение?

Решение:

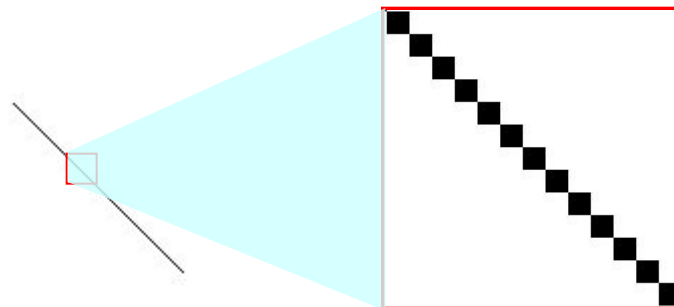
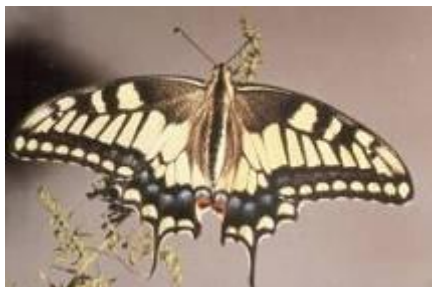
- объем сообщения в битах:
 $1024 \cdot 1024 \cdot 8 \text{ бит} / 512 = 16384 \text{ бит}$
- на 1 символ приходится
 $16384 / 1024 = 16 \text{ бит}$
- мощность алфавита $2^{16} = 65536$ символов

Ответ: 65536 символов (кодировка UNICODE)

Два типа кодирования рисунков

•растровое кодирование

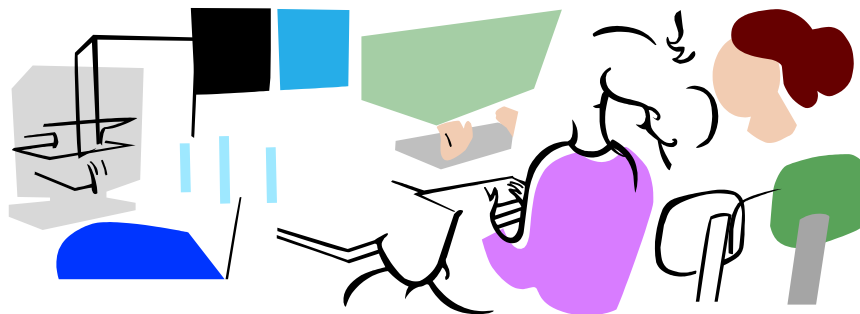
точечный рисунок, состоит из **пикселей**



фотографии, размытые изображения

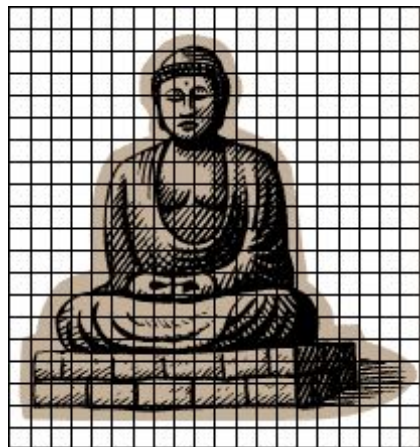
•векторное кодирование

рисунок, состоит из **отдельных геометрических фигур**



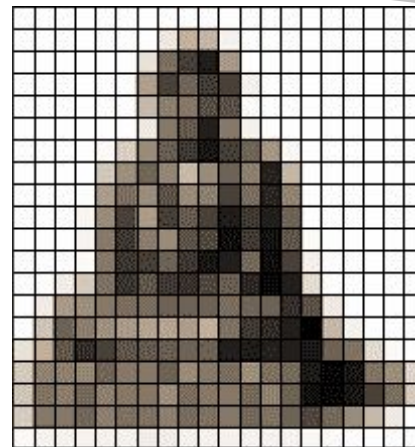
чертежи, схемы, карты

Растровое кодирование



Шаг 1. Дискретизация:
разбивка на *пиксели*.

Пиксель – это наименьший элемент рисунка, для которого можно независимо установить цвет.



Шаг 2. Для каждого пикселя определяется **единый цвет**.

Разрешение: число пикселей на дюйм, *pixels per inch (ppi)*
экран **96** ppi, печать **300-600** ppi, типография **1200** ppi

Растровое кодирование (*True Color*)

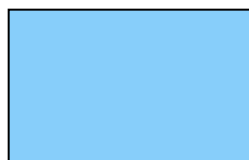
Шаг 3. От цвета – к числам: модель RGB

цвет = **R** + **G** + **B**

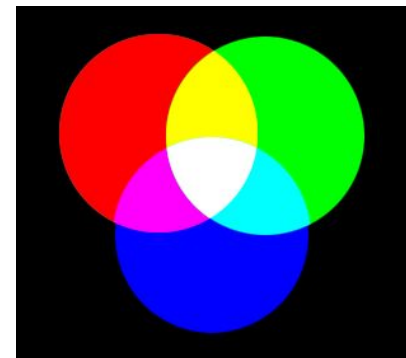
<i>red</i>	<i>green</i>	<i>blue</i>
красный	зеленый	синий
0..255	0..255	0..255



R = 218
G = 164
B = 32



R = 135
G = 206
B = 250



Шаг 4. Числа – в двоичную систему.



Сколько разных цветов можно кодировать?

$256 \cdot 256 \cdot 256 = 16\,777\,216$ (*True Color*)

Глубина
цвета



Сколько памяти нужно для хранения цвета 1 пикселя?

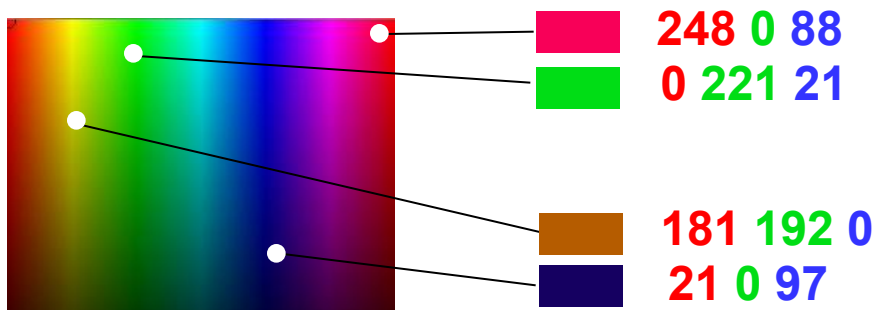
R: $256=2^8$ вариантов, нужно 8 бит = 1 байт

R G B: всего 3 байта

Растровое кодирование с палитрой

Шаг 1. Выбрать количество цветов: 2, 4, ... 256.

Шаг 2. Выбрать 256 цветов из палитры:



Шаг 3. Составить палитру (каждому цвету – номер 0..255)
палитра хранится в начале файла

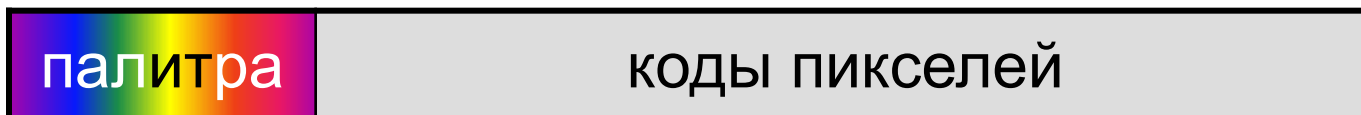
0	1		...	254	255
248 0 88	0 221 21		...	181 192 0	21 0 97

Шаг 4. Код пикселя = номеру его цвета в палитре

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

Растровое кодирование с палитрой

Файл с палитрой:



Один цвет в палитре: **3 байта (RGB)**

256 = 2⁸ цветов:

палитра 256·3 = 768 байт
рисунок 8 бит на пиксель

16 цветов:

палитра 16·3 = 48 байт
рисунок 4 бита на пиксель

2 цвета:

палитра 2·3 = 6 байт
рисунок 1 бит на пиксель

Глубина
цвета

Форматы файлов (растровые рисунки)

Формат	True Color	Палитра	Прозрачность
BMP			
JPG			
GIF			
PNG			

Растровые рисунки



- лучший способ для хранения **фотографий** и изображений без четких границ
- **спецэффекты** (тени, ореолы, и т.д.)



- есть **потеря информации**
- при изменении размеров рисунка он **искажается**
- **размер файла** не зависит от сложности рисунка

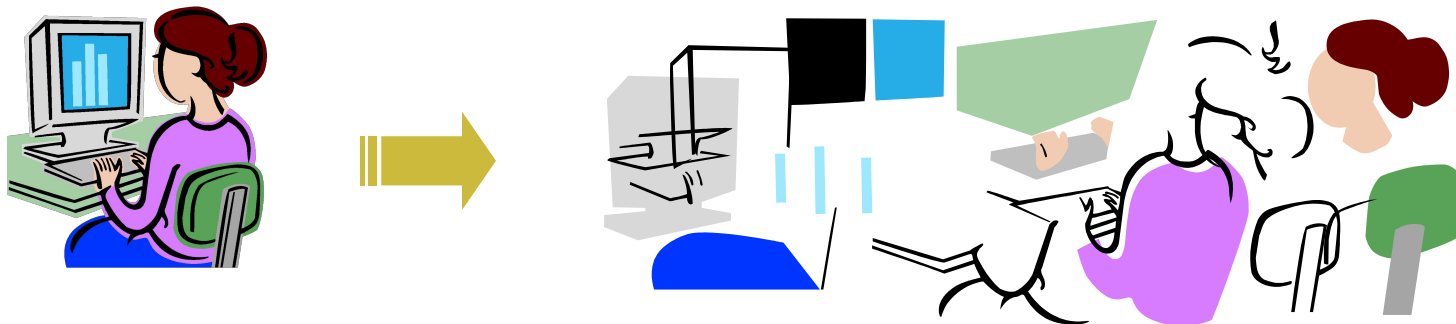
Векторные рисунки

Строятся из геометрических фигур:

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

Для каждой фигуры в памяти хранятся:

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



Форматы файлов:

- **WMF** (*Windows Metafile*)
- **AI** (*Adobe Illustrator*)
- **CDR** (*CorelDraw*)
- **FH** (*FreeHand*)

Векторные рисунки



- лучший способ для хранения **чертежей, схем, карт**;
- при кодировании **нет потери информации**;
- при изменении размера **нет искажений**;
- меньше **размер файла**, зависит от сложности рисунка;



- неэффективно использовать для **фотографий** и размытых изображений

Задачи: рисунок

Для хранения растрового рисунка размером 32x64 пикселя выделили 2 Кб памяти. Каково максимально возможное количество цветов в палитре?

Решение:

- общее число пикселей: $32 \cdot 64 = 2^5 \cdot 2^6 = 2^{11}$
- память
 $2 \text{ Кб} = 2 \cdot 2^{10} \text{ байта} = 2^{11} \text{ байта} = 2^{14} \text{ бита}$
- на 1 пиксель приходится
 $2^{14} : 2^{11} = 2^3 = 8 \text{ бит}$
- 8 бит \Rightarrow выбор 1 из 256 вариантов

Ответ: не более 256 цветов

Задачи: рисунок

Сколько места в памяти надо выделить для хранения 16-цветного рисунка размером 32 на 64 пикселя?

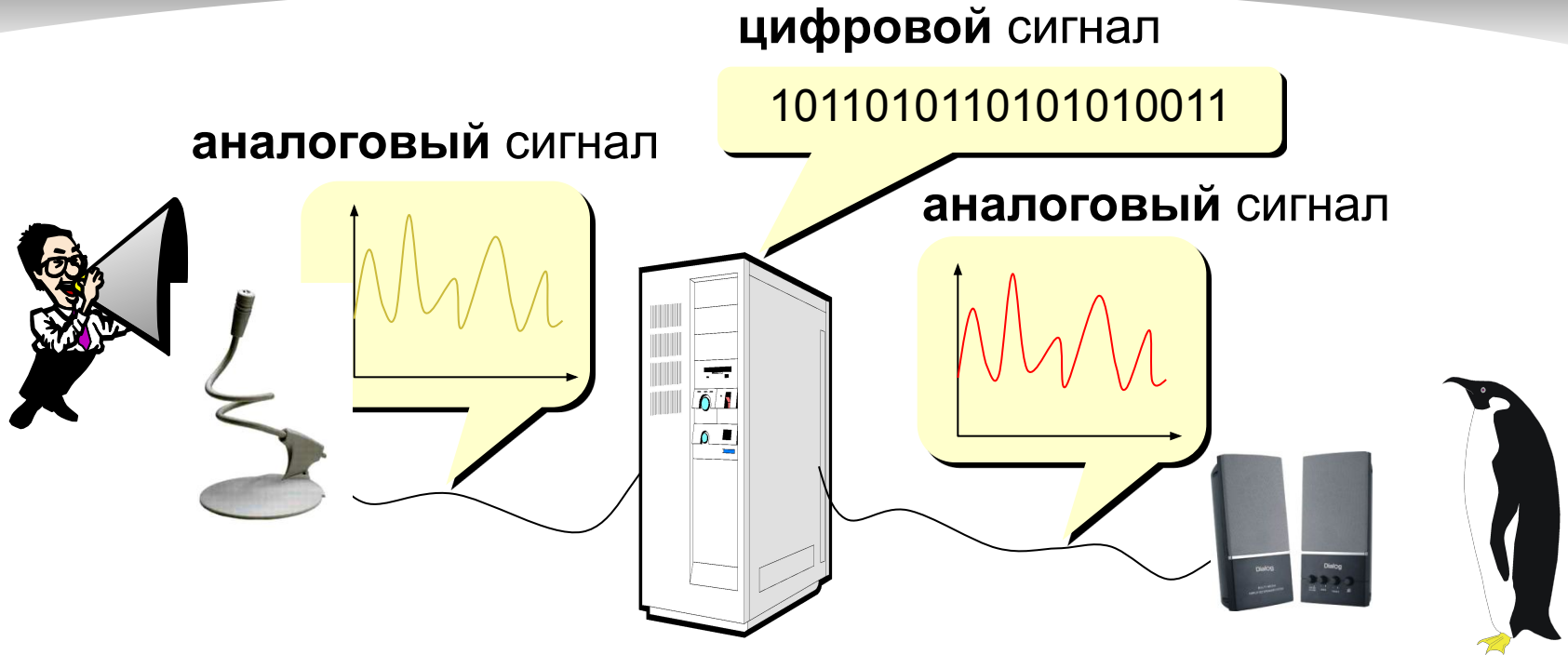
Решение:

- общее число пикселей: $32 \cdot 64 = 2048$
- при использовании 16 цветов на 1 пиксель отводится 4 бита (выбор 1 из 16 вариантов)

Ответ:

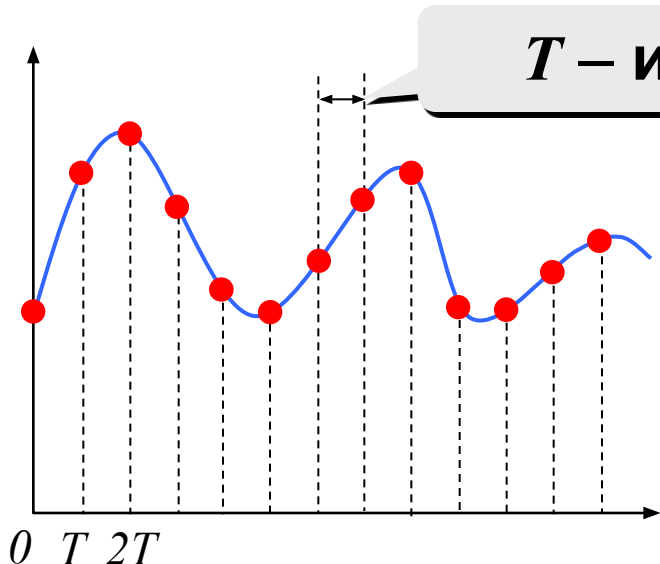
- $2048 \cdot 4 \text{ бита} = 8192 \text{ бита}$ или ...
- $2048 \cdot 4 : 8 \text{ байта} = 1024 \text{ байта}$ или ...
- $1024 : 1024 \text{ Кб} = 1 \text{ Кб}$

Оцифровка (перевод в цифровую форму)



Дискретизация по времени

хранятся только значения сигнала в моменты $0, T, 2T, \dots$

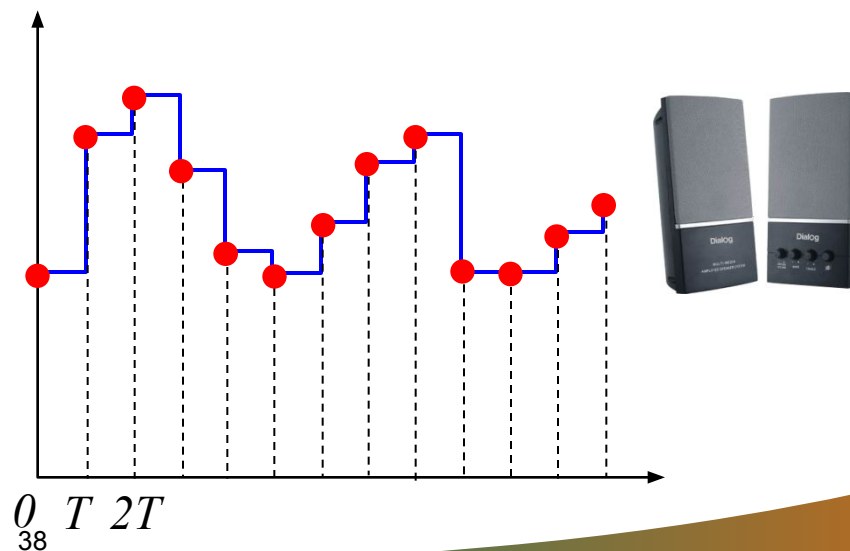


T – интервал дискретизации

Частота дискретизации: $f = \frac{1}{T}$
 $f = 8 \text{ кГц}, 11 \text{ кГц},$
 $22 \text{ кГц}, 44 \text{ кГц (CD)}$

22 кГц $T = \frac{1}{22000} \approx 0,00005 \text{ с}$

Человек слышит 16 Гц ... 20 кГц



Дискретизация по уровню



Сколько бит нужно, чтобы хранить число **0,7**?

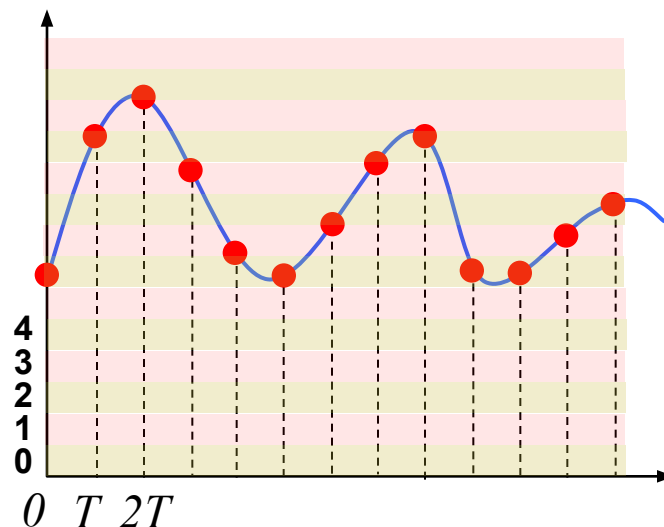
У всех точек в одной полосе одинаковый код!

8 бит = 256 уровней

16 бит = 65536 уровней

32 бита = 2^{32} уровней

64 бита = 2^{64} уровней



«Глубина» кодирования
(разрядность звуковой карты)



При оцифровке потерю информации дает дискретизация как по времени, так и по уровню!

Оцифровка – итог

 можно закодировать **любой звук** (в т.ч. ГОЛОС, СВИСТ, шорох, ...)

 • есть **потеря информации**
• большой **объем файлов**

 **Какие свойства цифрового звука определяют его качество?**

частота дискретизации 44 кГц,
глубина кодирования 16 бит:



88 Кб/с = 5,3 Мб/мин

Форматы файлов:

WAV (*Waveform audio format*), часто без сжатия (размер!)

MP3 (*MPEG-1 Audio Layer 3*, сжатие с потерями)

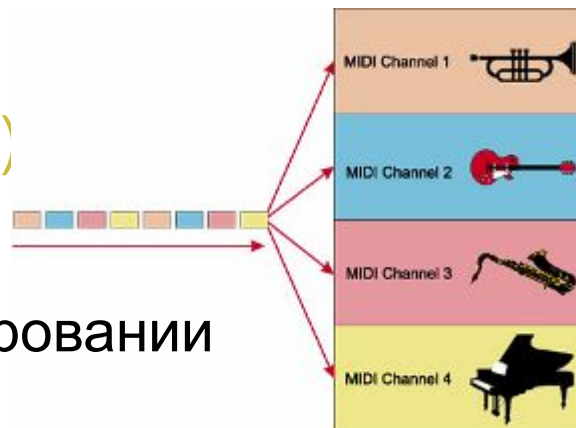
WMA (*Windows Media Audio*, потоковый звук, сжатие)

Инструментальное кодирование

MIDI (*Musical Instrument Digital Interface*), файлы *.MID

в файле:

- нота (высота, длительность)
- музыкальный инструмент
- параметры звука (громкость, тембр)
- может быть несколько каналов



- **нет потери информации** при кодировании инструментальной музыки
- маленький **размер файлов**



невозможно закодировать нестандартный звук, голос

MIDI-клавиатура:

