Кодирование информации

1. Язык и кодирование
2. Двоичное кодирование
3. Кодирование чисел и символов
4. Кодирование рисунков
5. Кодирование звука и видео

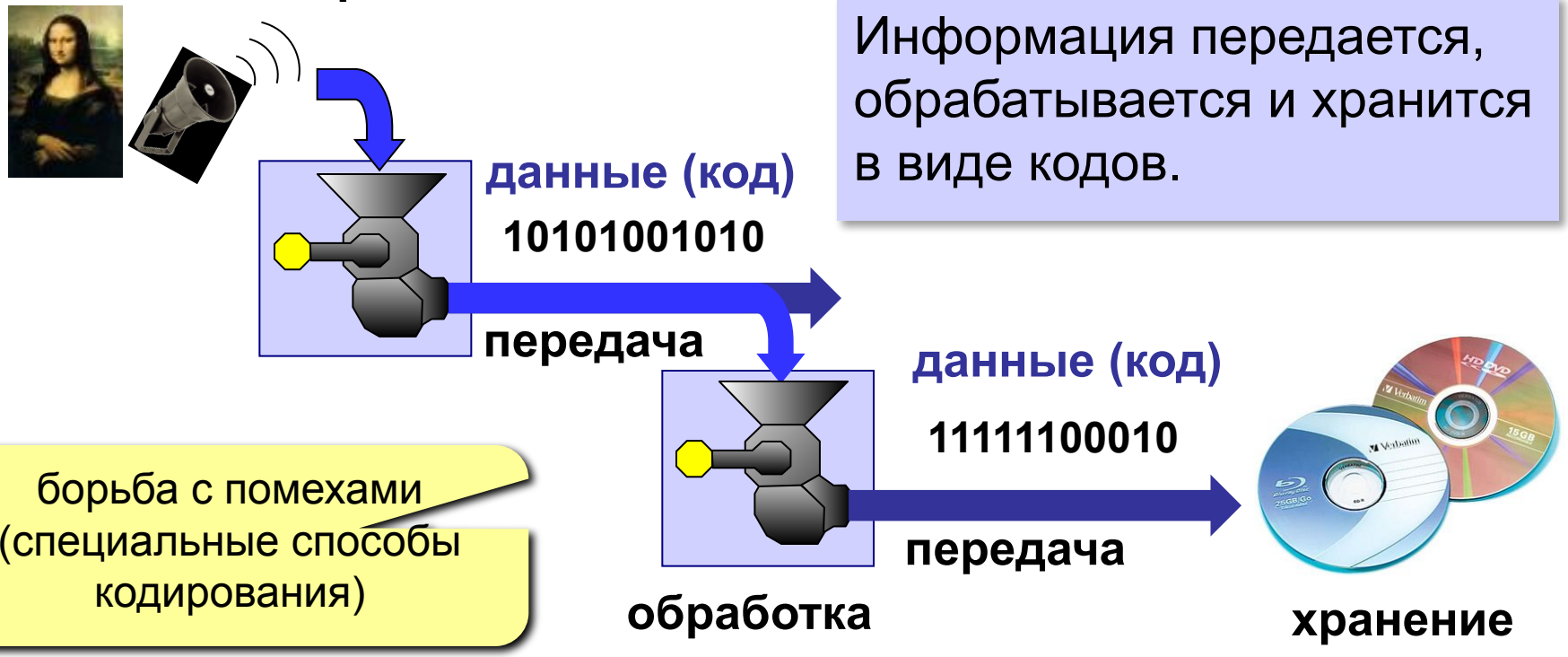
Кодирование информации

Тема 1. Язык и кодирование

Что такое кодирование?

Кодирование – это запись информации с помощью некоторой знаковой системы (языка).

Зачем кодируют информацию?



Информация передается, обрабатывается и хранится в виде кодов.

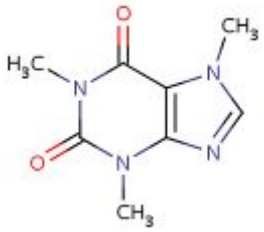
борьба с помехами (специальные способы кодирования)

Языки

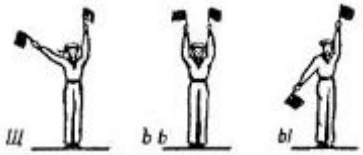
Язык – знаковая система, используемая для хранения и передачи информации.

- **естественные** (русский, английский, …)
  есть правила и исключения
- **формальные** (строгие правила)

$$E = mc^2$$



$$16 = 10_{16} = 20_8 = 10000_2$$



```
program qq;
begin
writeln("Привет!");
end.
```

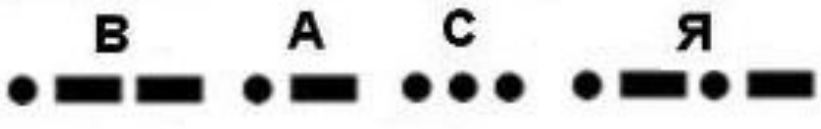
Грамматика – правила по которым из символов алфавита строятся слова.

Синтаксис – правила, по которым из слов строятся предложения.

Азбука Морзе

Задача 1. Закодируйте свое имя с помощью азбуки Морзе.

ВАСЯ



! Код неравномерный, нужен разделитель!

Кодовые таблицы

Задача 2. Закодируйте свое имя с помощью кодовой таблицы (*Windows-1251*):

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я

В А С Я

ВАСЯ **C2 C0 D1 DF**

Код равномерный, разделитель **НЕ** нужен!

Цели и способы кодирования

Текст:

- в России: ***Привет, Вася!***
- *Windows-1251*: *CFF0E8E2E52C20C2E0F1FF21*
- передача за рубеж (*транслит*): ***Privet, Vasya!***
- стенография: 
- шифрование: ***Рсйгжу-!Гбта”***

Числа:

- для вычислений: ***25***
- прописью: ***двадцать пять***
- римская система: ***XXV***

? Как зашифровано?

! Информация (смысл сообщения) может быть закодирована разными способами!

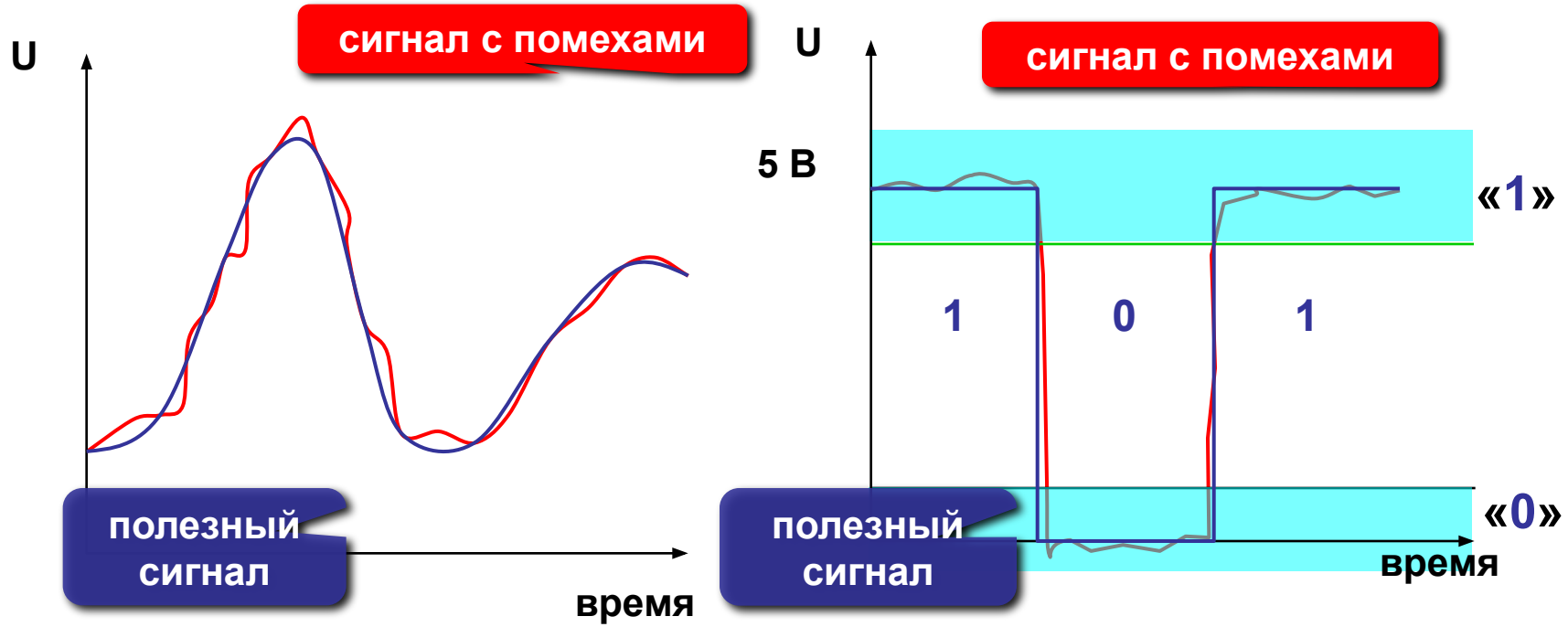
Кодирование информации

Тема 2. Двоичное кодирование

Двоичное кодирование

Двоичное кодирование – это кодирование всех видов информации с помощью двух знаков (обычно 0 и 1).

Передача электрических сигналов:



Двоичное кодирование

- в такой форме можно закодировать (почти) **все виды** информации
- нужны только устройства с **двумя состояниями**
- почти **нет ошибок** при передаче данных
- **компьютеру легче** обрабатывать данные

человеку сложно воспринимать двоичные коды

Можно ли использовать не «0» и «1», а другие символы, например, «А» и «Б»?

Декодирование

Декодирование – это восстановление сообщения из последовательности кодов.

М	А	Ы	Л	У	пробел
00	1	01	0	10	11

МАМА МЫЛА ЛАМУ → 00 1 00 1 11 00 01 0 1 11 0 1 00 10

Приняли сообщение:

00100111000101110100 10 → ???

ЛЛАЛЛАААЛЛЛАЛАААЛАЛЛАЛ

Не все коды допускают однозначное декодирование!

Почему?

Равномерные коды

Равномерные коды – все кодовые слова (коды отдельных букв) имеют одинаковую длину.

М	А	Ы	Л	У	пробел
000	001	010	011	100	101

МАМА МЫЛА ЛАМУ:
000 001 000 001 101 000 010 011 001 101 011 001 000 100

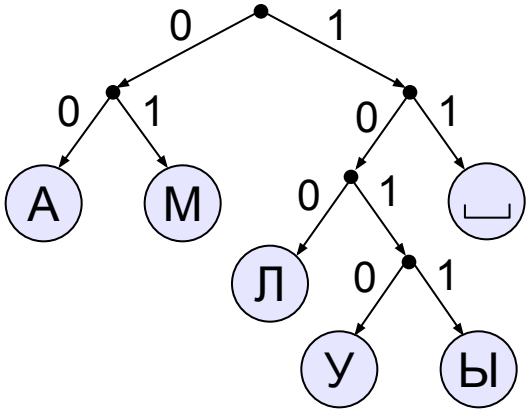
Равномерные коды позволяют однозначно декодировать сообщения!

сообщения получаются длинными

Неравномерные коды

кодовые слова имеют разную длину

М	А	Ы	Л	У	пробел
01	00	1011	100	1010	11



01	00	01	00	11	01	1011	100	00	11	100	00	1010
М	А	М	А	␣	М	Ы	Л	А	␣	Л	А	М	У

Префиксный код – ни одно кодовое слово не совпадает с началом другого кодового слова (*условие Фано*).

Любой префиксный код позволяет однозначно декодировать сообщения!

Постфиксные коды

Постфикс = окончание слова.

Постфиксный код – ни одно кодовое слово не совпадает с концом другого кодового слова *(«обратное» условие Фано)*.

М	А	Ы	Л	У	пробел
10	00	1101	001	0101	11

Любой постфиксный код позволяет однозначно декодировать сообщения (с конца)!

для декодирования нужно получить всё сообщение целиком

Задачи на построение кода

Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код:

А	Б	В	Г
1	000	001	?

Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?

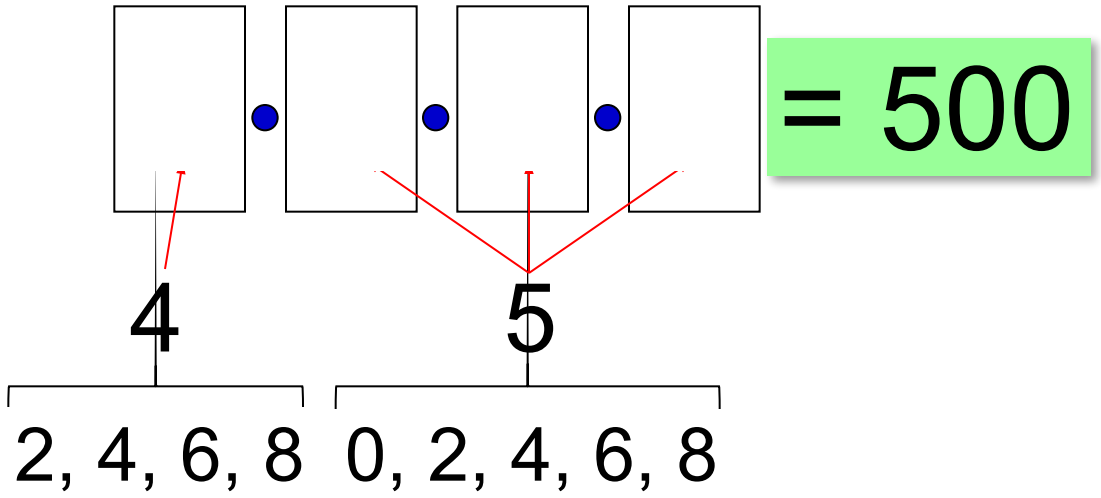
1) 00 2) 01 3) 11 4) 010

Решение:

1) для букв А-Б-В выполнятся условие Фано
2) при Г=00 условие Фано нарушится (пары Г-Б, Г-В)
3) **при Г=01 условие Фано выполняется**
4) при Г=11 условие Фано нарушится (пара А-Г)
5) при Г=010 условие Фано выполняется (но длиннее 01)

Комбинаторика

Задача 1. Сколько существует четырёхзначных чисел, составленных из чётных цифр?



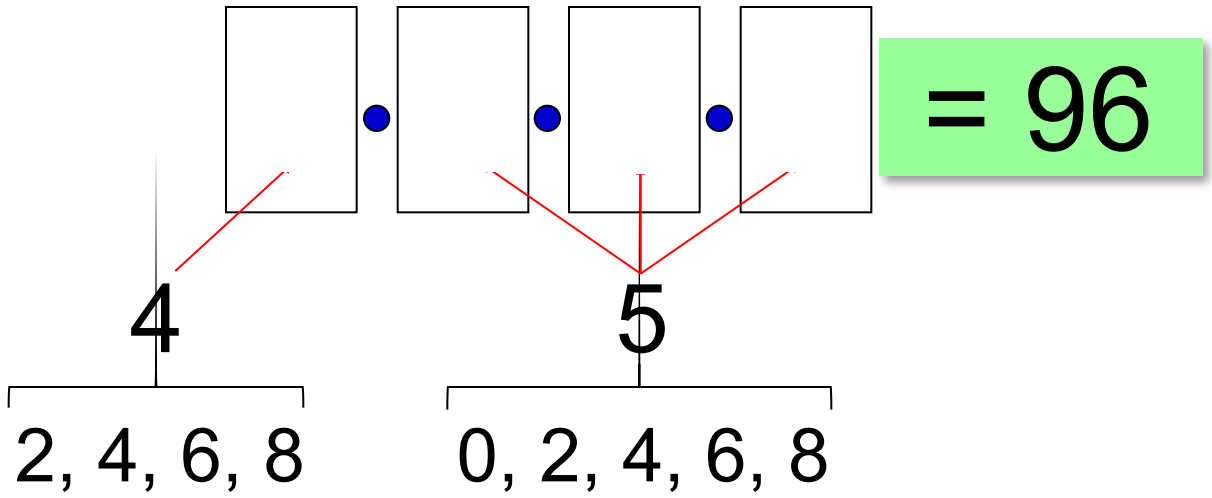
= 500

$$N = m_1 \cdot m_2 \cdot m_3 \cdot m_4$$

Правило умножения!

Комбинаторика

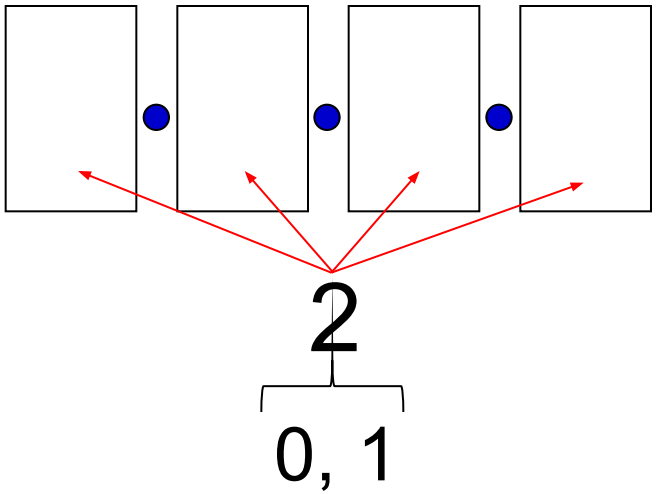
Задача 2. Сколько существует четырёхзначных чисел, составленных из чётных цифр, в которых **цифры не повторяются**?



одна цифра уже использована!

Комбинаторика

Задача 3. Сколько существует двоичных кодов длиной 4 бита?



$=2^4=16$

Правило умножения!

$$N = M \cdot M \cdot M \cdot M$$

$$N = M^L$$

длина сообщения

мощность алфавита

Комбинаторика

Задача 4. Сколько существует двоичных кодов длиной от 2 до 5 битов?

$$L = 2: \quad N_2 = 2^2 = 4 \qquad L = 3: \quad N_2 = 2^3 = 8$$
$$L = 4: \quad N_4 = 2^4 = 16 \qquad L = 5: \quad N_5 = 2^5 = 32$$

$$N = 4 + 8 + 16 + 32 = 60$$

$$N = N_2 + N_3 + N_4 + N_5$$

Правило сложения!

Комбинаторика

Задача 5. В некоторой стране живут 1000 человек. Правительство решило присвоить каждому собственный код, причем все коды должны быть одинаковой длины и состоять только из цифр 1, 2, 3 и 4. Определите наименьшую длину таких кодов.

$$N = 4^L \geq 1000$$

$L = 1$: $4^1 = 4 < 1000$

$L = 2$: $4^2 = 16 < 1000$

$L = 3$: $4^3 = 64 < 1000$

$L = 4$: $4^4 = 256 < 1000$

$L = 5$: $4^5 = 1024 > 1000$

Кодирование информации

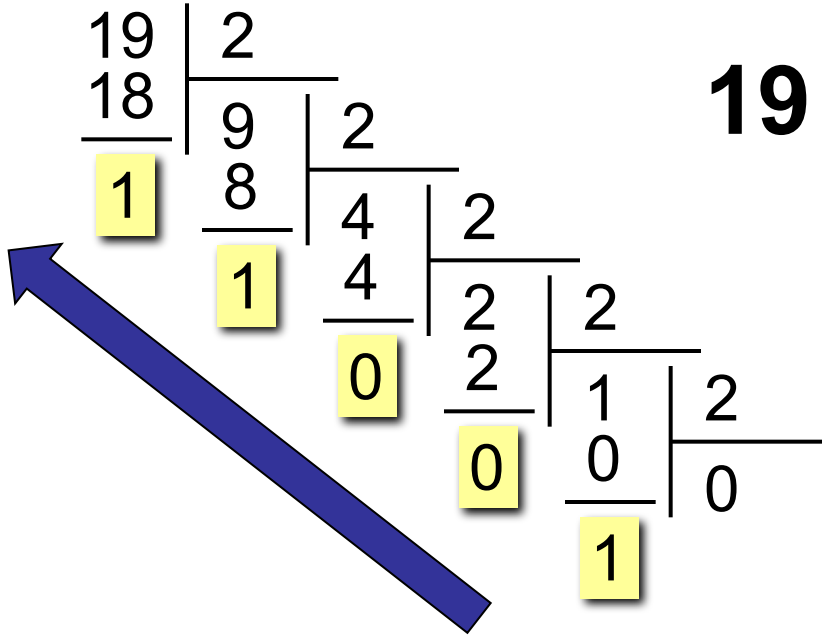
Тема 2. Кодирование чисел и символов

Кодирование чисел (двоичная система)

Алфавит: 0, 1

Основание (количество цифр): **2**

10 → 2



$$19 = 10011_2$$

система счисления

2 → 10

4 3 2 1 0 разряды

$$10011_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$= 16 + 2 + 1 = 19$$

Кодирование символов

Текстовый файл

- на экране (символы)
- в памяти – двоичные коды

1000001_2	1000010_2	1000011_2	1000100_2
65	66	67	68

В файле хранятся не изображения символов, а их числовые коды в двоичной системе!

А где же хранятся изображения?

Кодирование символов

1. **Сколько символов** надо использовать одновременно? **256** или 65536 (UNICODE)
2. **Сколько места** надо выделить **на символ**:

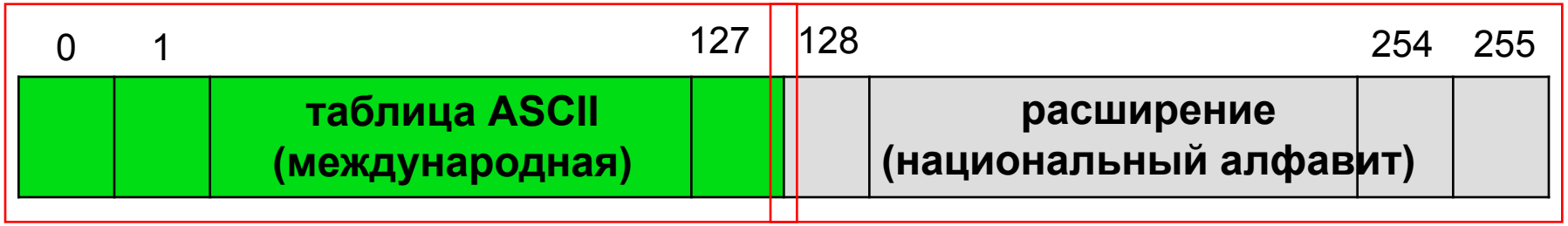
256 = 2^8 ⇒ **8 бит на символ**

3. Выбрать **256 любых символов** (или 65536) - **алфавит**.
4. Каждому символу – **уникальный код 0..255** (или 0..65535). Таблица символов:

коды	65	66	67	68		
…	A	B	C	D	…	

5. Коды – в **двоичную систему**.

8-битные кодировки (1 байт на символ)



ASCII = *American Standard Code for Information Interchange*

0-31 управляющие символы:

7 – звонок, 10 – новая строка, 13 – возврат каретки, 27 – Esc.

32 пробел

знаки препинания: `. , : ; ! ?`

специальные знаки: `+ - * / () {} []`

48-57 цифры **0..9**

65-90 заглавные латинские буквы **A-Z**

97-122 строчные латинские буквы **a-z**

Кодовая страница (расширенная таблица ASCII)

для русского языка:

CP-866 для системы *MS DOS*

CP-1251 для системы *Windows* (Интернет)

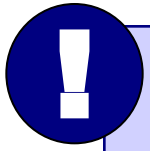
КОИ8-Р для системы *UNIX* (Интернет)

8-битные кодировки (1 байт на символ)

- 1 байт на символ – файлы небольшого размера!
- просто обрабатывать в программах

- нельзя использовать символы разных кодовых страниц одновременно (русские и французские буквы, и т.п.)
- неясно, в какой кодировке текст (перебор вариантов!)
- для каждой кодировки нужен свой шрифт (изображения символов)

Стандарт UNICODE



Идея: объединить все символы в одну таблицу!

- 110 182 символа (2012)
- каждому символу присвоен код
  кириллица:
  А – 0410_{16}, Б – 0411_{16}, …
  а – 0430_{16}, б – 0431_{16}, …
- коды $0..10FFFF_{16}$, всего 1 114 112

UNICODE в *Windows* (UTF-16)

- общеупотребительные символы
  0..65535 = 2^{16}-1 (0..FFFF$_{16}$)
- эти символы можно закодировать с помощью 16 бит
- кодировка UTF-16 (почти все символы по 16 бит)


можно одновременно использовать символы разных языков (Интернет)

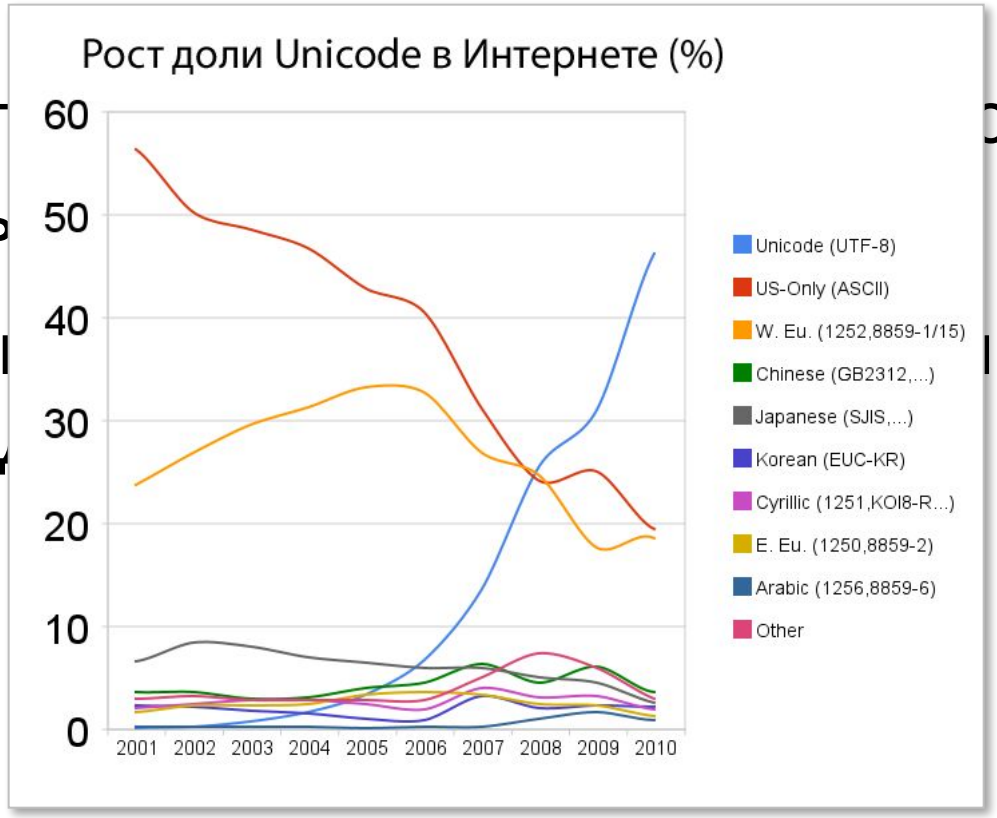

размер файла увеличивается

UNICODE в Linux (кодировка UTF-8)

- *символы ASCII – 1 байт на символ*
- остальные символы от 2 до 4 байт
- более 50% сайтов используют UTF-8

- текст
  (кодь

- пере
- заме



ов ASCII
в размере
вол

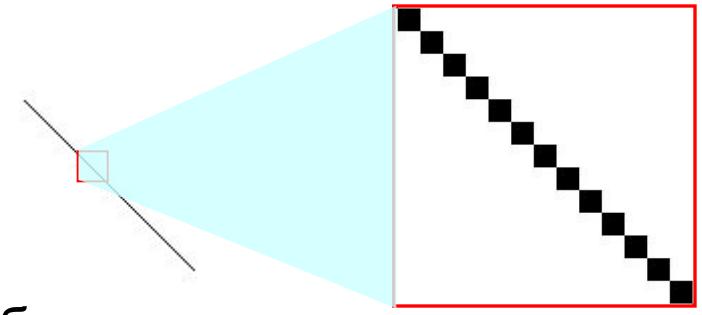
Кодирование информации

Тема 4. Кодирование рисунков

Два типа кодирования рисунков

- **растровое кодирование**

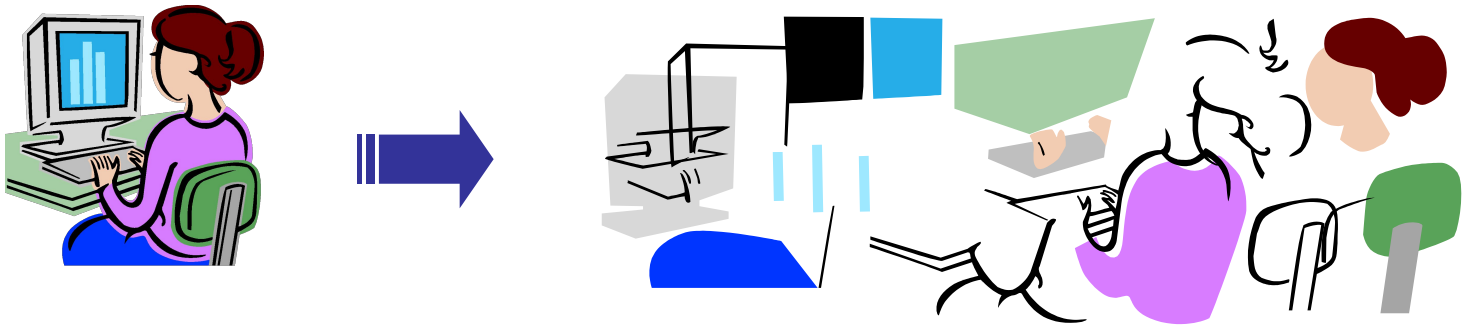
точечный рисунок, состоит из **пикселей**



фотографии, размытые изображения

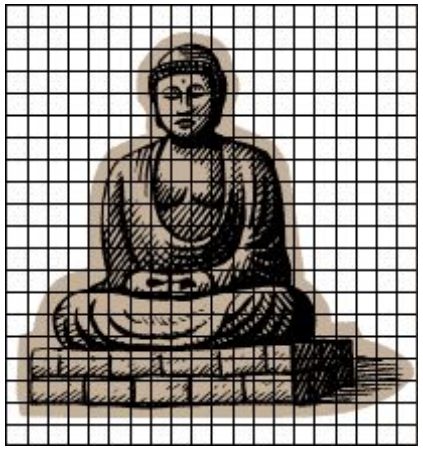
- **векторное кодирование**

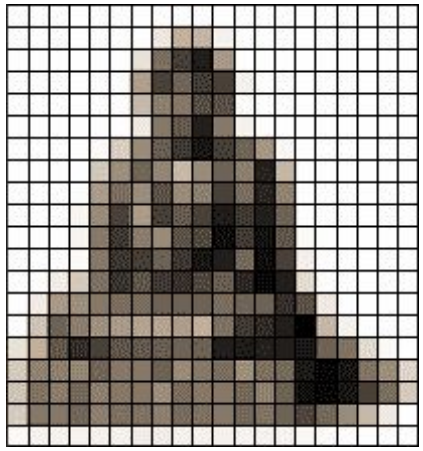
рисунок, состоит из **отдельных геометрических фигур**



чертежи, схемы, карты

Растровое кодирование





Шаг 1. Дискретизация: разбивка на *пиксели*.

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

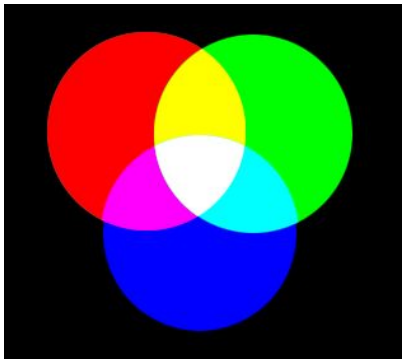
Шаг 2. Для каждого пикселя определяется **единый цвет**.

Есть потеря информации!
- **почему?**
- **как ее уменьшить?**

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**

	red красный 0..255	***green*** зеленый 0..255	***blue*** синий 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?

Глубина цвета

256·256·256 = **16 777 216 (*True Color*)**

Сколько памяти нужно для хранения цвета 1 пикселя?

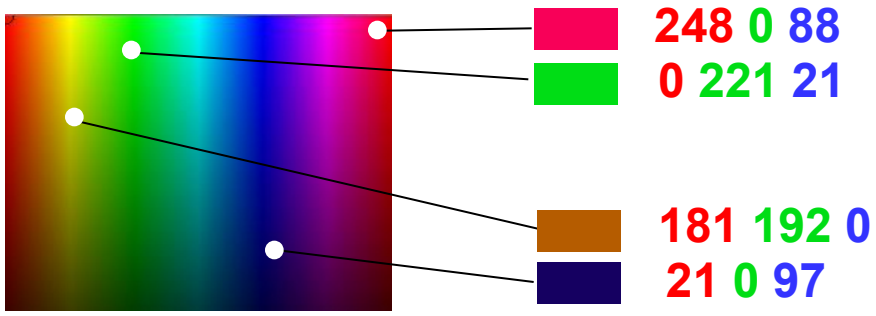
R: 256=2^8 вариантов, нужно 8 бит = 1 байт

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		45		65		254	255
248 0 88	0 221 21	…	181 192 0	…	21 0 97	…	161 12 20	19 23 90

Шаг 4. Код пикселя = номеру его цвета в палитре

65	1	45	14	…	12	23

Растровое кодирование с палитрой

Файл с палитрой:

Сколько занимает палитра и основная часть?

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

256 = 2^8 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:

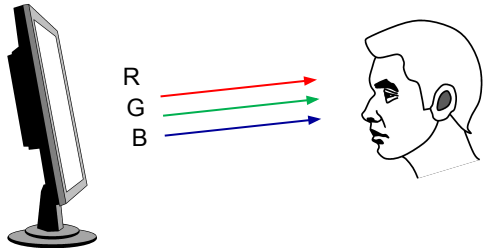
палитра	2·3 = 6 байт
рисунок	1 бит на пиксель

Глубина цвета

Форматы файлов (растровые рисунки)

Формат	True Color	Палитра	Прозрачность
BMP	✔	✔	⛔
JPG	✔	⛔	⛔
GIF	⛔	✔	✔
PNG	✔	✔	✔

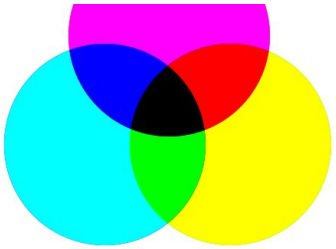
Кодирование цвета при печати



Белый – красный = голубой **C = Cyan**

Белый – зелёный = пурпурный **M = Magenta**

Белый – синий = желтый **Y = Yellow**



Модель CMY

C	M	Y	
0	0	0	
255	255	0	
255	0	255	
0	255	255	
255	255	255	

Модель CMYK: + *Key color*

Меньший расход краски и лучшее качество для чёрного и серого цветов.

Растровые рисунки

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

- есть **потеря информации** (почему?)
- при изменении размеров рисунка он **искажается**
- **размер файла** не зависит от сложности рисунка (а от чего зависит?)

Какие свойства цифрового рисунка определяют его качество?

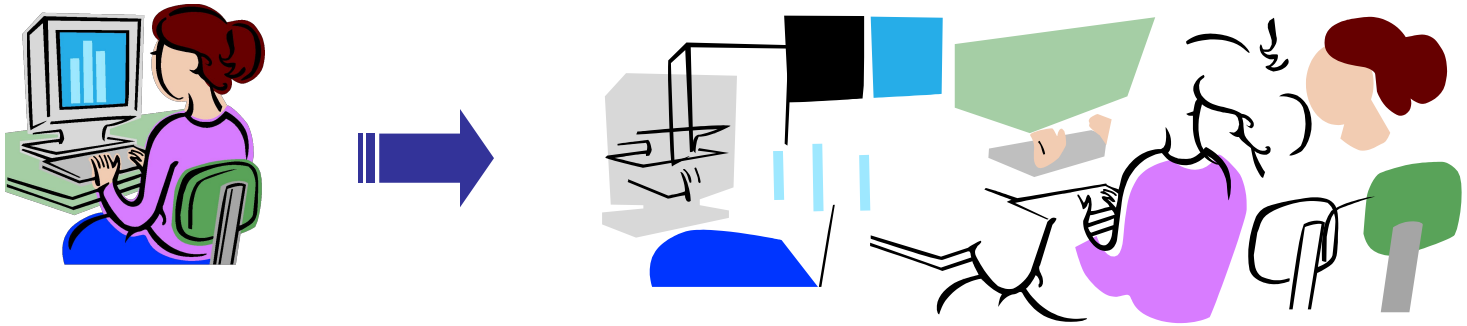
Векторные рисунки

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)*
- **CDR** *(CorelDraw)*
- **AI** *(Adobe Illustrator)*
- **SVG** *(Inkscape)*

для Web

Векторные рисунки

прямоугольник

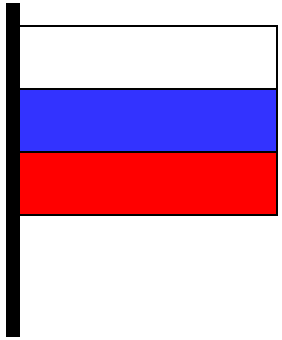
размеры

```
<svg>

<rect width="135" height="30"
      x="0" y="10"
      stroke-width="1" stroke="rgb(0,0,0)"
      fill="rgb(255,255,255)"/>

<rect width="135" height="30" x="0" y="40"
      stroke-width="1" stroke="rgb(0,0,0)"
      fill="rgb(0,0,255)"/>
<rect width="135" height="30" x="0" y="70"
      stroke-width="1" stroke="rgb(0,0,0)"
      fill="rgb(255,0,0)"/>
<line x1="0" y1="0"
      x2="0" y2="150"
      stroke-width="15" stroke="rgb(0,0,0)" />

</svg>
```

координаты

контур

заливка



Векторные рисунки

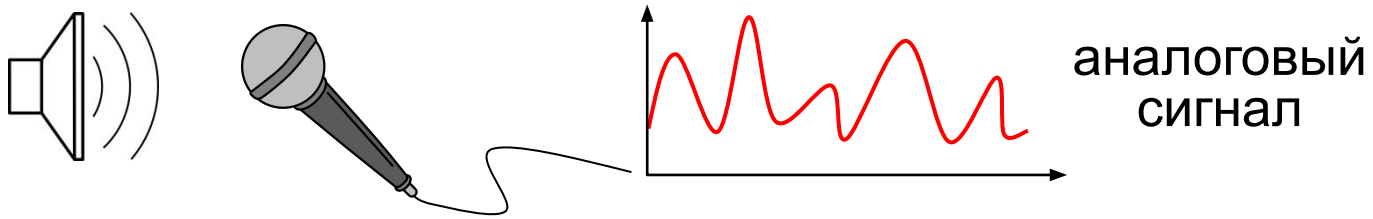
- лучший способ для хранения **чертежей, схем, карт;**
- при кодировании **нет потери информации**;
- при изменении размера **нет искажений**;
- меньше **размер файла**, зависит от сложности рисунка;

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

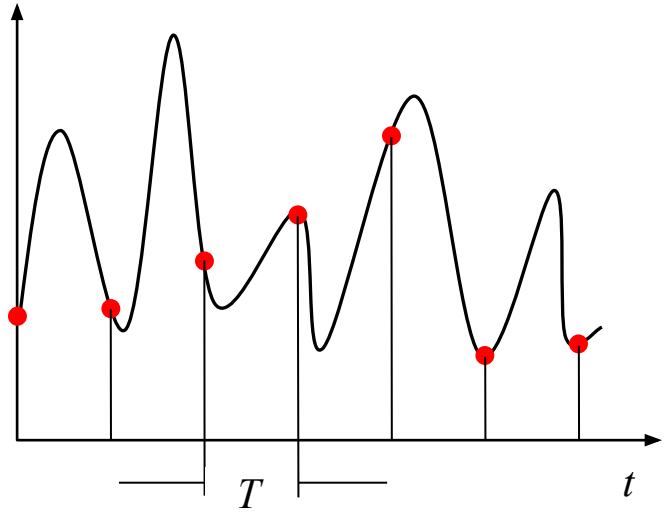
Кодирование информации

Тема 5. Кодирование звука и видео

Оцифровка звука



Оцифровка – это преобразование аналогового сигнала в цифровой код (дискретизация).



T – интервал дискретизации (с)

$f = \frac{1}{T}$ – частота дискретизации (Гц, кГц)

8 кГц – минимальная частота для распознавания речи

11 кГц, 22 кГц,

44,1 кГц – качество CD-дисков

48 кГц – фильмы на DVD

96 кГц, 192 кГц

Человек слышит

16 Гц … 20 кГц

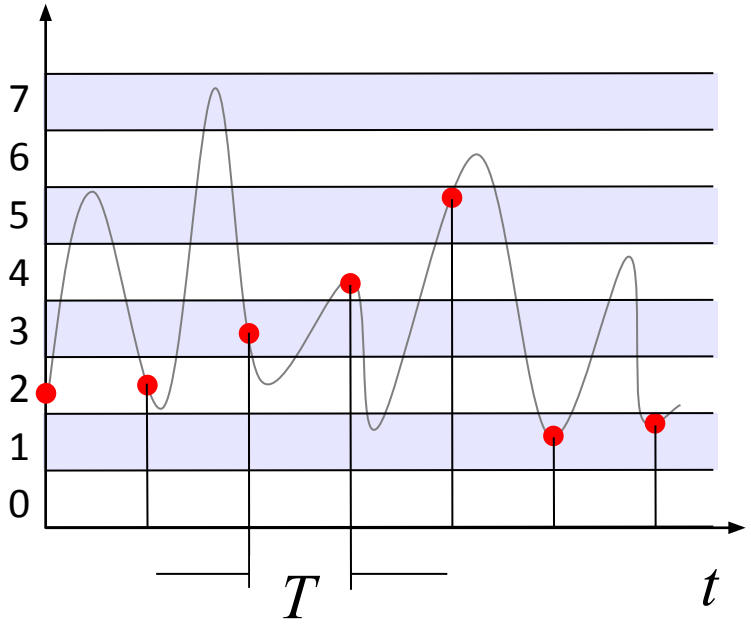
Оцифровка звука: квантование

Сколько битов нужно, чтобы записать число 0,6?

Квантование (дискретизация по уровню) – это представление числа в виде цифрового кода конечной длины.

АЦП = **А**налого-**Ц**ифровой **П**реобразователь

3-битное кодирование:



8 битов = 256 уровней

16 битов = 65536 уровней

24 бита = 2^{24} уровней

Разрядность кодирования — это число битов, используемое для хранения одного отсчёта.

Оцифровка звука

Задача. Определите информационный объем данных, полученных при оцифровке звука длительностью 1 минута с частотой 44 кГц с помощью 16-битной звуковой карты. Запись выполнена в режиме «стерео».

За 1 сек *каждый канал* записывает 44000 значений,
каждое занимает 16 битов = 2 байта
всего 44000 · 2 байта = 88000 байтов

С учётом «стерео»
всего 88000 · 2 = 176000 байтов

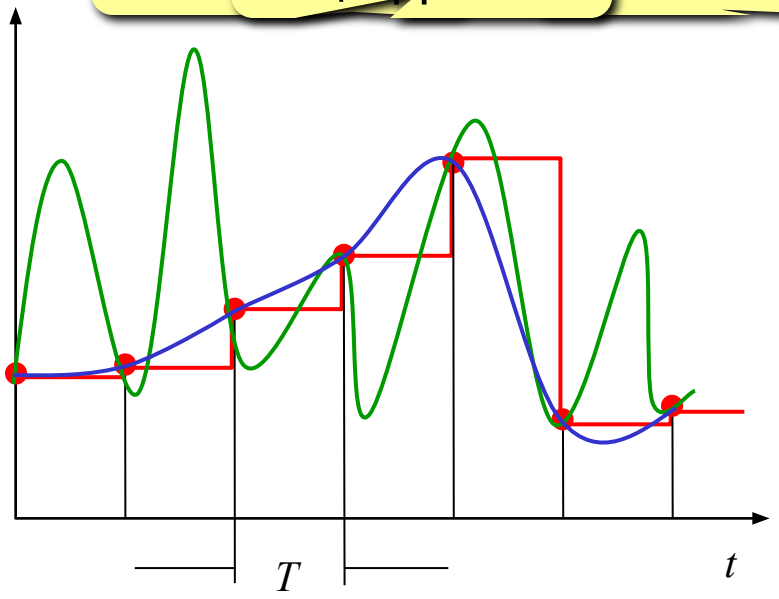
За 1 минуту
176000 · 60 = 1056000 байтов
≈ 10313 Кбайт ≈ 10 Мбайт

Оцифровка звука

Как восстановить сигнал?

ЦАП = **Ц**ифро-**А**налоговый **П**реобразователь

п… сгла… было до оцифровки … без …паживания



аналоговые устройства!

Какой улучшить качество? — уменьшать T

Что при этом ухудшится? — ↑ размер файла

Оцифровка – итог

можно закодировать **любой звук** (в т.ч. голос, свист, шорох, …)

- есть **потеря информации**
- большой **объем файлов**

Какие свойства оцифрованного звука определяют качество звучания?

Форматы файлов:

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

MP3 (*MPEG-1 Audio Layer 3*, сжатие с учётом восприятия человеком)

AAC (*Advanced Audio Coding*, 48 каналов, сжатие)

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

OGG (*Ogg Vorbis,* открытый формат, сжатие)

Инструментальное кодирование

MIDI (*Musical Instrument Digital Interface* — цифровой интерфейс музыкальных инструментов).

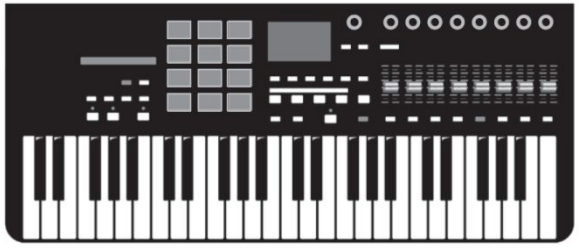
в файле **`.mid`**:
- нота (высота, длительность)
- музыкальный инструмент
- параметры звука (громкость, тембр)
- до 1024 каналов

128 мелодических и 47 ударных

в памяти звуковой карты:
- образцы звуков (волновые таблицы)

программа для звуковой карты!

MIDI-клавиатура:



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

невозможно закодировать нестандартный звук, голос

Трекерная музыка

В файле (модуле):

- образцы звуков (*сэмплы*)
- нотная запись, трек *(track)* – дорожка
- музыкальный инструмент
- до 32 каналов

Форматы файлов:

MOD разработан для компьютеров *Amiga*

S3M оцифрованные каналы + синтезированный звук, 99 инструментов

XM, STM, …

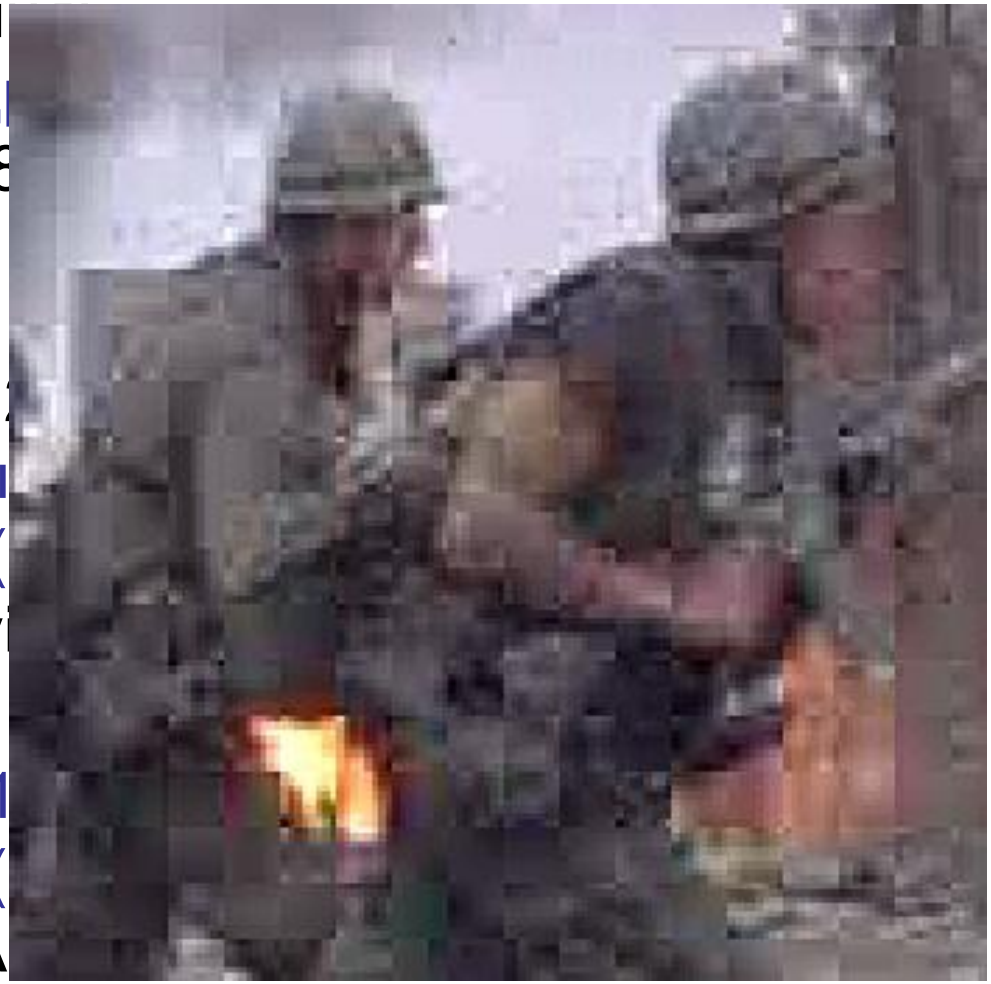
Использование: демосцены (важен размер файла)

Кодирование видео

Видео = изображения + звук Синхронность!

изображен
- ≥ 25 кад
- **PAL**: 768
  - за 1
  - за 1
- **HDTV**: 1
- исходны
- сжатие (
- DivX, Xvi

звук:
- 48 кГц, 1
- сжатие (
- MP3, AA



Форматы видеофайлов

AVI – *Audio Video Interleave* – чередующиеся звук и видео; контейнер – могут использоваться разные *кодеки*

MPEG – *Motion Picture Expert Group*

WMV – *Windows Media Video,* формат фирмы *Microsoft*

MP4 – *MPEG-4*, сжатое видео и звук

MOV – *Quick Time Movie*, формат фирмы *Apple*

WebM – открытый формат, поддерживается браузерами

Конец фильма

ПОЛЯКОВ Константин Юрьевич
д.т.н., учитель информатики высшей категории,
ГОУ СОШ № 163, г. Санкт-Петербург
kpolyakov@mail.ru