

**Преподаватель: Дариенко Татьяна Викторовна**

### **Практическая работа №4, 5**

**Тема:** Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеоинформации.

**Цель:** изучить способы представления текстовой, графической, звуковой информации и видеоинформации; научиться записывать числа в различных системах счисления, выполнять арифметические действия с ними.

#### **Краткие теоретические сведения**

#### **Представление информации в ЭВМ**

Обработка информации в ЭВМ основана на обмене электрическими сигналами между различными устройствами машины. Эти сигналы возникают в определенной последовательности. Признак наличия сигнала можно обозначить цифрой 1, признак отсутствия – цифрой 0. Таким образом, в ЭВМ реализуются два устойчивых состояния. С помощью определенных наборов цифр 0 и 1 можно закодировать любую информацию. Каждый такой набор нулей и единиц называется двоичным кодом. Количество информации, кодируемое двоичной цифрой – 0 или 1 – называется битом. Бит является наименьшей единицей измерения количества информации. На практике чаще приходится работать с байтом – основной единицей измерения объема данных.

Двоичная система счисления намного старше электронных машин, она издавна была предметом пристального внимания ученых. Особенно сильным это увлечение было с конца 16 до 19 века. Официальное рождение двоичной системы счисления связано с именем Г.В.Лейбница, опубликовавшего в 1703 г. статью, в которой он рассмотрел правила выполнения арифметических действий над двоичными числами. Знаменитый Лейбниц считал двоичную систему счисления простой, удобной, красивой. Даже по его просьбе была выбита медаль в честь этой «диадической» системы (так называли тогда двоичную систему счисления).

Двоичная система счисления наиболее проста и удобна для автоматизации. Наличие в системе всего лишь двух символов упрощает их преобразование, что позволило создать надежно работающие технические устройства, которые могут со 100 процентной надежностью сохранять и распознавать не более двух различных состояний (цифр):

- электромагнитные реле (замкнуто/разомкнуто), широко использовались в конструкциях первых ЭВМ;
- участок поверхности магнитного носителя информации (намагничен/ размагничен);
- участок поверхности лазерного диска (отражает/не отражает).

Утверждение двоичной арифметики в качестве общепринятой при конструкции ЭВМ с программным управлением состоялось под влиянием работы Джон фон Неймана о проекте первой ЭВМ с хранимой в памяти программой.

Компьютеры используют двоичную систему так как:

- для её реализации нужны технические устройства с двумя устойчивыми состояниями,
- представление информации с помощью только двух состояний надежно и помехоустойчиво,
- двоичная арифметика намного проще десятичной.

Двоичная система, удобная для компьютера, для человека неудобна из-за её громоздкости и непривычной записи. Для того чтобы понимать слово компьютера, разработаны восьмеричная и шестнадцатеричная системы счисления. Числа в этих системах требуют в 3-4 раза меньше разрядов, чем в двоичной системе.

Из любой системы счисления можно перейти к двоичному коду. Почти все ЭВМ используют либо непосредственно двоичную систему счисления, либо двоичное кодирование какой-либо другой системы счисления.

## Системы счисления

Понятие «число» является ключевым как для математики, так и для информатики. Люди всегда считали и записывали числа. Число изображалось с помощью одного или нескольких символов.

**Система счисления** — это способ изображения чисел и соответствующие ему правила действия над числами. Разнообразные системы счисления, которые существовали раньше и которые используются в наше время, можно разделить на непозиционные и позиционные. Знаки, используемые при записи чисел, называются цифрами.

В **непозиционных системах счисления** значение цифры не зависит от положения в числе.

Примером непозиционной системы счисления является римская система (римские цифры). В римской системе в качестве цифр используются латинские буквы:

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

В **позиционных системах счисления** величина, обозначаемая цифрой в записи числа, зависит от ее позиции. Количество используемых цифр называется основанием позиционной системы счисления.

**Десятичная система.** Система счисления, применяемая в современной математике, является позиционной. Данная система является для нас наиболее привычной. В ней используются цифры от 0 до 9 для записи чисел. Они также носят название арабских. В зависимости от положения цифры в числе, она может обозначать разные разряды – единицы, десятки, сотни, тысячи или миллионы. Позиционный характер этой системы легко понять на примере любого многозначного числа. Например, в числе 333 первая тройка означает три сотни, вторая — три десятка, третья — три единицы. Причина, по которой десятичная система счисления стала общепринятой, вовсе не математическая. Десять пальцев рук – вот аппарат для счёта, которым человек пользуется с доисторических времен.

**Двоичная система.** Одна из основных систем счисления в информатике – двоичная. Ее простота позволяет компьютеру производить громоздкие вычисления в несколько раз быстрее, нежели в десятичной системе. Для записи чисел используется лишь две цифры – 0 и 1. При этом, в зависимости от положения 0 или 1 в числе, его значение будет меняться. Изначально именно с помощью двоичного кода компьютеры получали всю необходимую информацию. При этом, единица означала наличие сигнала, передаваемого с помощью напряжения, а ноль – его отсутствие.

**Восьмеричная система.** Еще одна известная компьютерная система счисления, в которой применяются цифры от 0 до 7. Применялась в основном в тех областях знаний, которые связаны с цифровыми устройствами. Но в последнее время она употребляется значительно реже, так как на смену ей пришла шестнадцатеричная система счисления.

**Шестнадцатеричная система.** В последнее время все большую популярность приобретает в программировании и информатике система счисления шестнадцатеричная. В ней используются не только цифры от 0 до 9, но и ряд латинских букв – A, B, C, D, E, F. При этом, каждая из букв имеет свое значение, так A=10, B=11, C=12 и так далее.

### *Дискретное представление информации. Основы кодирования*

**Теория кодирования** – это раздел теории информации, связанный с задачами кодирования и декодирования сообщений, поступающих к потребителям и посылаемых из источников информации.

Теория кодирования близка к древнейшему искусству тайнописи – криптографии. В середине XIX в. изобретение телефона и искрового телеграфа поставило перед учеными и инженерами проблему создания новой теории кодирования. Первой ориентированной на технику системой кодирования оказалась азбука Морзе, в которой принято троичное

кодирование (точка, тире, пауза).

Цифровые ЭВМ (персональные компьютеры относятся к классу цифровых) используют двоичное кодирование любой информации. В основном это объясняется тем, что построить техническое устройство, безошибочно различающее 2 разных состояния сигнала, технически оказалось проще, чем то, которое бы безошибочно различало 5 или 10 различных состояний.

К недостаткам двоичного кодирования относят очень длинные записи двоичных кодов, что затрудняет работу с ними

**Кодирование чисел.** Вопрос о кодировании чисел возникает по той причине, что в машину нельзя вводить числа в том виде, в котором они изображаются человеком на бумаге. Нужно кодировать знак числа - один бит отводится при этом для представления знака числа (нулем кодируется знак "плюс", единицей - "минус"), а для кодирования действительных чисел существует специальный формат чисел с плавающей запятой.

**Кодирование текста.** Для кодирования букв и других символов, используемых в печатных документах, необходимо закрепить за каждым символом числовой номер – код. В англоязычных странах используются 26 прописных и 26 строчных букв (A ... Z, a ... z), 9 знаков препинания (., : ! " ; ? ( ) ), пробел, 10 цифр, 5 знаков арифметических действий (+, -, \*, /, ^) и специальные символы (№, %, \_, #, \$, &, >, <, |, \) – всего чуть больше 100 символов.

Суть кодирования заключается в том, что каждому символу ставят в соответствие двоичный код от 00000000 до 11111111 или соответствующий ему десятичный код от 0 до 255. Для кодирования текстов используются различные таблицы перекодировки. Важно, чтобы при кодировании и декодировании одного и того же текста использовалась одна и та же таблица. Таблица перекодировки - таблица, содержащая упорядоченный некоторым образом перечень кодируемых символов, в соответствии с которой происходит преобразование символа в его двоичный код и обратно. Наиболее популярные таблицы перекодировки: ДКОИ-8, ASCII, CP1251, Unicode. В настоящее время для кодировки русских букв используют пять различных кодовых таблиц (КОИ - 8, CP1251, CP866, Mac, ISO), причем тексты, закодированные при помощи одной таблицы не будут правильно отображаться в другой кодировке.

**Кодирование графической информации.** В видеопамяти находится двоичная информация об изображении, выводимом на экран. Почти все создаваемые, обрабатываемые или просматриваемые с помощью компьютера изображения можно разделить на две большие части – растровую и векторную графику. Растровые изображения представляют собой однослойную сетку точек, называемых пикселями (pixel, от англ. picture element). Код пикселя содержит информации о его цвете. В противоположность растровой графике векторное изображение многослойно. Каждый элемент векторного изображения – линия. Каждый элемент векторного изображения является объектом, который описывается с помощью математических уравнений. Сложные объекты (ломанные линии, различные геометрические фигуры) представляются в виде совокупности элементарных графических объектов.

**Кодирование звука.** На компьютере работать со звуковыми файлами начали в 90-х годах. В основе цифрового кодирования звука лежит – процесс преобразования колебаний воздуха в колебания электрического тока и последующая дискретизация аналогового электрического сигнала. Кодирование и воспроизведение звуковой информации осуществляется с помощью специальных программ (редактор звукозаписи).

#### ***Аналоговый и дискретный способ кодирования***

Человек способен воспринимать и хранить информацию в форме образов (зрительных, звуковых, осязательных, вкусовых и обонятельных). Зрительные образы могут быть сохранены в виде изображений (рисунков, фотографий и так далее), а звуковые — зафиксированы на пластинках, магнитных лентах, лазерных дисках и так далее.

Информация может быть представлена в аналоговой или дискретной форме. При

аналоговом представлении физическая величина принимает бесконечное множество значений, причем ее значения изменяются непрерывно. При дискретном представлении физическая величина принимает конечное множество значений, причем ее величина изменяется скачкообразно.

Примером аналогового представления графической информации может служить, например, живописное полотно, цвет которого изменяется непрерывно, а дискретного–изображение, напечатанное с помощью струйного принтера и состоящее из отдельных точек разного цвета.

Примером аналогового хранения звуковой информации является виниловая пластинка (звуковая дорожка изменяет свою форму непрерывно), а дискретного – аудиокомпакт-диск (звуковая дорожка которого содержит участки с различной отражающей способностью).

Преобразование графической и звуковой информации из аналоговой формы в дискретную производится путем дискретизации, то есть разбиения непрерывного графического изображения и непрерывного (аналогового) звукового сигнала на отдельные элементы. В процессе дискретизации производится кодирование, то есть присвоение каждому элементу конкретного значения в форме кода.

*Дискретизация* – это преобразование непрерывных изображений и звука в набор дискретных значений в форме кодов.

### ***Кодирование изображений***

Создавать и хранить графические объекты в компьютере можно двумя способами – как *растровое* или как *векторное* изображение. Для каждого типа изображений используется свой способ кодирования.

#### ***Кодирование растровых изображений***

Растровое изображение представляет собой совокупность точек (пикселей) разных цветов. Пиксель – минимальный участок изображения, цвет которого можно задать независимым образом.

В процессе кодирования изображения производится его пространственная дискретизация. Пространственную дискретизацию изображения можно сравнить с построением изображения из мозаики (большого количества маленьких разноцветных стекол). Изображение разбивается на отдельные маленькие фрагменты (точки), причем каждому фрагменту присваивается значение его цвета, то есть код цвета (красный, зеленый, синий и так далее).

Для черно-белого изображения информационный объем одной точки равен одному биту (либо черная, либо белая – либо 1, либо 0). Для 4 цветов необходимо – 2 бита, для 8 цветов – 3 бита, для 16 цветов – 4 бита, для 256 цветов – 8 бит (1 байт).

Качество изображения зависит от количества точек (чем меньше размер точки и, соответственно, больше их количество, тем лучше качество) и количества используемых цветов (чем больше цветов, тем качественнее кодируется изображение).

Для представления цвета в виде числового кода используются две обратных друг другу цветовые модели: RGB или CMYK. Модель RGB используется в телевизорах, мониторах, проекторах, сканерах, цифровых фотоаппаратах... Основные цвета в этой модели: красный (Red), зеленый (Green), синий (Blue). Цветовая модель CMYK используется в полиграфии при формировании изображений, предназначенных для печати на бумаге.

Если кодировать цвет одной точки изображения тремя битами (по одному биту на каждый цвет RGB), то мы получим все восемь различных цветов.

<b>R</b>	<b>G</b>	<b>B</b>	<b>Цвет</b>
1	1	1	Белый
1	1	0	Желтый
1	0	1	Пурпурный
1	0	0	Красный
0	1	1	Голубой

0	1	0	Зеленый
0	0	1	Синий
0	0	0	Черный

На практике же, для сохранения информации о цвете каждой точки цветного изображения в модели RGB обычно отводится 3 байта (то есть 24 бита) - по 1 байту (то есть по 8 бит) под значение цвета каждой составляющей. Таким образом, каждая RGB-составляющая может принимать значение в диапазоне от 0 до 255 (всего  $2^8=256$  значений), а каждая точка изображения, при такой системе кодирования может быть окрашена в один из 16 777 216 цветов. Такой набор цветов принято называть True Color (правдивые цвета), потому что человеческий глаз все равно не в состоянии различить большего разнообразия.

Для того чтобы на экране монитора формировалось изображение, информация о каждой точке (код цвета точки) должна храниться в видеопамати компьютера. Рассчитаем необходимый объем видеопамати для одного из графических режимов. В современных компьютерах разрешение экрана обычно составляет 1280x1024 точек. Т.е. всего  $1280 * 1024 = 1310720$  точек. При глубине цвета 32 бита на точку необходимый объем видеопамати:  $32 * 1310720 = 41943040$  бит = 5242880 байт = 5120 Кб = 5 Мб.

Растровые изображения очень чувствительны к масштабированию (увеличению или уменьшению). При уменьшении растрового изображения несколько соседних точек преобразуются в одну, поэтому теряется различимость мелких деталей изображения. При увеличении изображения увеличивается размер каждой точки и появляется ступенчатый эффект, который можно увидеть невооруженным глазом.

#### *Кодирование векторных изображений*

Векторное изображение представляет собой совокупность графических примитивов (точка, отрезок, эллипс...). Каждый примитив описывается математическими формулами. Кодирование зависит от прикладной среды.

Достоинством векторной графики является то, что файлы, хранящие векторные графические изображения, имеют сравнительно небольшой объем.

Важно также, что векторные графические изображения могут быть увеличены или уменьшены без потери качества.

#### *Двоичное кодирование звука*

Использование компьютера для обработки звука началось позднее, нежели чисел, текстов и графики.

Звук – волна с непрерывно изменяющейся амплитудой и частотой. Чем больше амплитуда, тем он громче для человека, чем больше частота, тем выше тон.

Звуковые сигналы в окружающем нас мире необычайно разнообразны. Сложные непрерывные сигналы можно с достаточной точностью представлять в виде суммы некоторого числа простейших синусоидальных колебаний. Причем каждое слагаемое, то есть каждая синусоида, может быть точно задана некоторым набором числовых параметров – амплитуды, фазы и частоты, которые можно рассматривать как код звука в некоторый момент времени.

В процессе кодирования звукового сигнала производится его временная дискретизация – непрерывная волна разбивается на отдельные маленькие временные участки и для каждого такого участка устанавливается определенная величина амплитуды.

Таким образом непрерывная зависимость амплитуды сигнала от времени заменяется на дискретную последовательность уровней громкости.

Каждому уровню громкости присваивается его код. Чем большее количество уровней громкости будет выделено в процессе кодирования, тем большее количество информации будет нести значение каждого уровня и тем более качественным будет звучание.

Качество двоичного кодирования звука определяется глубиной кодирования и частотой дискретизации.

*Частота дискретизации* – количество измерений уровня сигнала в единицу времени.

Количество уровней громкости определяет глубину кодирования. Современные звуковые карты обеспечивают 16-битную глубину кодирования звука. При этом количество уровней громкости равно  $N = 2^{16} = 65536$ .

### ***Представление видеoinформации***

В последнее время компьютер все чаще используется для работы с видеoinформацией. Простейшей такой работой является просмотр кинофильмов и видеоклипов. Следует четко представлять, что обработка видеoinформации требует очень высокого быстродействия компьютерной системы.

Что представляет собой фильм с точки зрения информатики? Прежде всего, это сочетание звуковой и графической информации. Кроме того, для создания на экране эффекта движения используется дискретная по своей сути технология быстрой смены статических картинок. Исследования показали, что если за одну секунду сменяется более 10-12 кадров, то человеческий глаз воспринимает изменения на них как непрерывные.

Казалось бы, если проблемы кодирования статической графики и звука решены, то сохранить видеоизображение уже не составит труда. Но это только на первый взгляд, поскольку, как показывает разобранный выше пример, при использовании традиционных методов сохранения информации электронная версия фильма получится слишком большой. Достаточно очевидное усовершенствование состоит в том, чтобы первый кадр запомнить целиком (в литературе его принято называть ключевым), а в следующих сохранять лишь отличия от начального кадра (разностные кадры).

Существует множество различных форматов представления видеоданных.

В среде Windows, например, уже более 10 лет (начиная с версии 3.1) применяется формат Video for Windows, базирующийся на универсальных файлах с расширением AVI (Audio Video Interleave – чередование аудио и видео). Более универсальным является мультимедийный формат Quick Time, первоначально возникший на компьютерах Apple.

### **Содержание работы**

**Задание №1.** Выполнить задания по переводу чисел из одной системы счисления в другую.

#### **Порядок работы:**

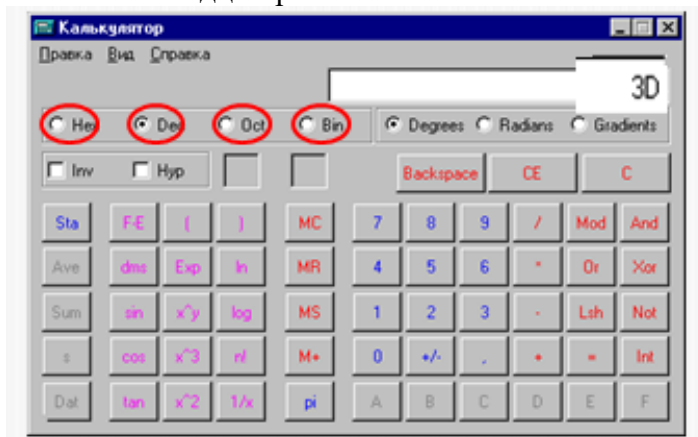
1. Запустить программу *Калькулятор* командой главного меню [*Пуск – Программы – Стандартные – Калькулятор*].
2. В меню *Вид* установить режим *Инженерный*.
3. Изменяя положение переключателя в положение Bin, Oct, Dec, Hex, наблюдать за изменением состояния «кнопок» Калькулятора.

Bin – двоичная система счисления;

Oct – восьмеричная система счисления;

Dec – десятичная система счисления;

Hex – шестнадцатеричная система счисления.



4. Убедиться, что переключатель стоит в положении Dec. Ввести число соответствующее текущему году. Перевести переключатель последовательно в положение Bin, Oct, Hex. Результаты перевода числа записать в тетрадь, используя подстрочный индекс (для обозначения основания системы счисления):

Например:  $2007_{10} = 1111010111_2 = 3727_8 = 7D7_{16}$

5. Аналогично перевести в 2-ую, 8-ую, 16-ую системы счисления следующие числа: **текущий год, дата рождения, месяц рождения и год рождения**. Результаты перевода чисел записать в тетрадь по образцу.

**Задание №2.** Выполнить арифметические вычисления в различных системах счисления.

**Порядок работы:**

1. Запустить программу *Калькулятор*.  
 2. Перевести переключатель в положение Bin. Выполнить сложение, вычитание, умножение и деление чисел  $1010_2$  и  $10_2$ . Результат записать в тетрадь.

$$1010_2 + 10_2 = \qquad \qquad \qquad 1010_2 * 10_2 =$$

$$1010_2 - 10_2 = \qquad \qquad \qquad 1010_2 : 10_2 =$$

3. Перевести переключатель в положение Oct. Выполнить аналогичные операции для чисел  $124_8$  и  $14_8$ .

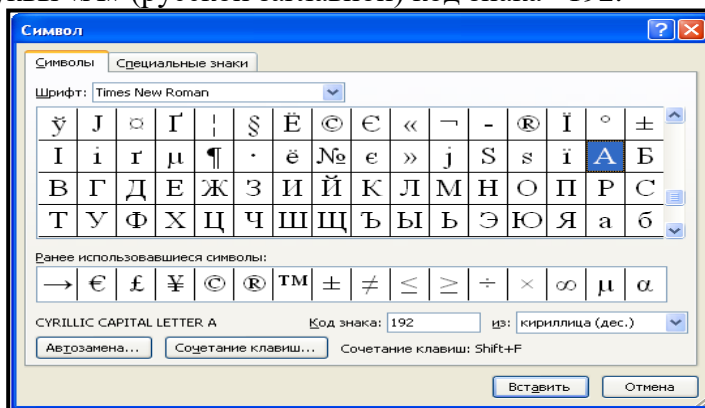
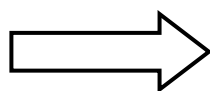
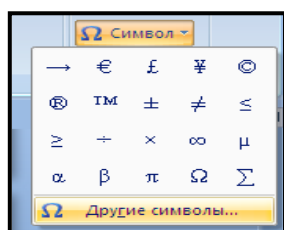
4. Перевести переключатель в положение Hex. Выполнить аналогичные операции для чисел  $3A8_{16}$  и  $6F_{16}$ .

**Задание №3.** Используя таблицу символов, записать последовательность десятичных числовых кодов в кодировке Windows для названия учебного заведения, своих ФИО.

**Порядок работы:**

1. Начертить в тетради таблицу по образцу (см. ниже), заполнить первую строку буквами (различая заглавные и строчные буквы).

2. Заполнить вторую строку десятичными числовыми кодами, открыв редактор MS Word, используя таблицу символов. Таблица символов отображается с помощью команды: вкладка **Вставка**→**Символ**→**Другие символы**. В поле **Шрифт** выбрать **Times New Roman**, в поле **из** выбрать **кириллица**. Например, для буквы «А» (русской заглавной) код знака– 192.



**Образец:**

С	Г	У	В	Т
209	195	211	194	210

Т	а	т	ь	я	н	а
210	224	242	252	255	237	224

В	и	к	т	о	р	о	в	н	а
194	232	234	242	238	240	238	226	237	224

**Задание №4.** Используя стандартную программу *Блокнот*, определить, какая фраза в кодировке Windows задана последовательностью числовых кодов, продолжить код.

### Порядок работы:

1. Запустить **Блокнот**. С помощью дополнительной цифровой клавиатуры при нажатой клавише ALT ввести код, отпустить клавишу **ALT**. В документе появится соответствующий символ.

0255	0243	0247	0243	0241	0252	0226	0237	0234	0240	0243	

0237	0224	0239	0229	0240	0226	0238	0236	0234	0243	0240	0241	0229	

Заполнить верхнюю строку названием специальности:


**Задание №5.** Заполнить пропуски числами, зная что:

- 1 байт=8бит
- 1 Кбайт=1 024 байт
- 1 Мбайт=1 024 Кбайт
- 1 Гбайт=1 024 Мбайт
- 1 Тбайт=1 024 Гбайт

129 Кбайт	=	байт	=	бит
243 Мбайт	=	байт	=	бит
571 Гбайт	=	байт	=	бит

**Задание №6.** Решить задачу: Текстовый документ, состоящий из 3072 символов, хранился в 8-битной кодировке КОИ-8. Этот документ был преобразован в 16-битную кодировку Unicode. Укажите, какое дополнительное количество байт потребуется для хранения документа.

**Задание №7.** Решить задачу: Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 128×128 пикселей при условии, что в изображении могут использоваться 256 различных цветов?

**Задание №8\*.** Решить задачу: Производилась двухканальная (стерео) звукозапись с частотой дискретизации 48 кГц и 24-битным разрешением. В результате был получен файл размером 5625 Мбайт, сжатие данных не производилось. Определите, сколько времени (в минутах) производилась запись.

### Контрольные вопросы

1. Как представлена информация, которую обрабатывает компьютер?
2. Что такое бит?
3. Что такое байт?
4. Перечислите единицы измерения информации.
5. Что такое система счисления?
6. Что такое непозиционная система счисления?
7. Что такое позиционная система счисления?
8. Сколько символов в десятичной системе счисления?
9. Сколько символов в двоичной системе счисления?
10. Сколько символов в восьмеричной системе счисления?
11. Сколько символов в шестнадцатеричной системе счисления?
12. Что изучает теория кодирования?
13. Что такое дискретизация?
14. Приведите примеры аналогового представления информации.
15. Приведите примеры дискретного представления информации.



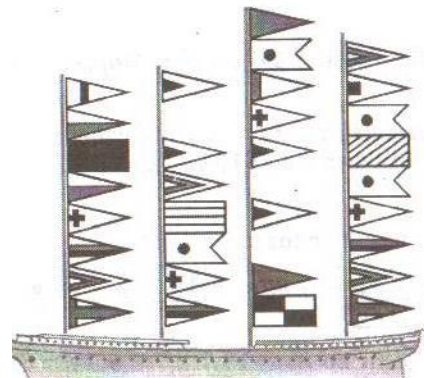
16. Что такое пиксель?
17. Чем отличается растровое изображение от векторного?
18. Что такое звук?
19. От чего зависит качество двоичного кодирования звука?
20. Что такое частота дискретизации?

**Дополнительно:**

**Задание №1.** С помощью флажковой азбуки расшифруйте следующее сообщение.

Дана кодовая таблица флажковой азбуки

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		



**Задание №2.** Расшифруйте следующие слова и определите правило кодирования:  
**ЕРАВШН, УМЫЗАК, АШНРРИ, РКДЕТИ.**