

Раздел 2. Информация и информационные процессы.

1.1. Подходы к понятию информации и измерению информации. Информационные объекты различных видов. Универсальность дискретного (цифрового) представления информации. Представление информации в двоичной системе счисления.

Слово «**информация**» происходит от латинского слова *informatio*, что в переводе означает сведение, разъяснение, ознакомление.

Можно выделить следующие подходы к определению информации:

❖ традиционный (обыденный) - используется в информатике: **Информация** – это сведения, знания, сообщения о положении дел, которые человек воспринимает из окружающего мира с помощью органов чувств (зрения, слуха, вкуса, обоняния, осязания).

❖ вероятностный - используется в теории об информации: **Информация** – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределённости и неполноты знаний.

Для человека: **Информация** – это знания, которые он получает из различных источников с помощью органов чувств.

Вся информация, которую обрабатывает компьютер, представлена **двоичным кодом** с помощью двух цифр – **0** и **1**. Эти два символа 0 и 1 принято называть **битами** (от англ. **binary digit** – двоичный знак).

Бит – наименьшая единица измерения объема информации.

Название	Условное обозначение	Соотношение
Байт	Байт	1 байт = 2^3 бит = 8 бит
Килобит	Кбит	1 Кбит = 2^{10} бит = 1024 бит
КилоБайт	Кб	1 Кб = 2^{10} байт = 1024 байт
МегаБайт	Мб	1 Мб = 2^{10} Кб = 1024 Кб
ГигаБайт	Гб	1 Гб = 2^{10} Мб = 1024 Мб
ТераБайт	Тб	1 Тб = 2^{10} Гб = 1024 Гб

ИЗМЕРЕНИЕ ИНФОРМАЦИИ

В информатике используются различные подходы к измерению информации:

Содержательный подход к измерению информации.

Сообщение, уменьшающее неопределенность знаний человека в два раза, несет для него **1 бит** информации.

Количество информации, заключенное в сообщении, определяется по формуле Хартли: где **N** – количество равновероятных событий;

I – количество информации (бит), заключенное в сообщении об одном из событий.

$$I = \log_2 N$$

$$N = 2^I$$

Алфавитный (технический) подход к измерению информации основан на подсчете числа символов в сообщении.

Если допустить, что все символы алфавита встречаются в тексте с одинаковой частотой, то количество информации, заключенное в **сообщении** вычисляется по формуле:

$$I_c = i * K$$

I_c – информационный объем сообщения,

K – количество символов,

$$N = 2^i$$

N – мощность алфавита (количество символов),

i - информационный объем 1 символа.

Раздел 2. Информация и информационные процессы.

Информационный объект – обобщающее понятие, описывающее различные виды объектов; это предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств.

Простые информационные объекты: звук, изображение, текст, число. **Комплексные (структурированные) информационные объекты:** элемент, база данных, таблица, гипертекст, гипермедиа.

Кодирование – это операция преобразования знаков или групп знаков одной знаковой системы в знаки или группы знаков другой знаковой системы.

Декодирование – расшифровка кодированных знаков, преобразование кода символа в его изображение

Двоичное кодирование – кодирование информации в виде 0 и 1.

Способы кодирования и декодирования информации в компьютере, в первую очередь, зависят от вида информации, а именно, что должно кодироваться:

- числа
- символьная информация (буквы, цифры, знаки)
- графические изображения
- звук

ДВОИЧНОЕ КОДИРОВАНИЕ ЧИСЕЛ

Система счисления – это совокупность правил для обозначения и наименования чисел.

Непозиционной называется такая система счисления, в которой количественный эквивалент каждой цифры не зависит от ее положения (места, позиции) в записи числа.

Основанием системы счисления называется количество знаков или символов, используемых для изображения числа в данной системе счисления.

Наименование системы счисления соответствует ее основанию (например, десятичной называется система счисления так потому, что ее основание равно 10, т.е. используется десять цифр).

Система счисления называется **позиционной**, если значение цифры зависит от ее места (позиции) в записи числа.

Для кодирования чисел используются специальные правила перевода чисел из десятичной системы счисления в двоичную и наоборот.

Системы счисления, используемые в компьютерах

Двоичная система счисления. Для записи чисел используются только две цифры – 0 и 1. Выбор двоичной системы объясняется тем, что электронные элементы, из которых строятся ЭВМ, могут находиться только в двух хорошо различимых состояниях. По существу эти элементы представляют собой выключатели. Как известно выключатель либо включен, либо выключен. Третьего не дано. Одно из состояний обозначается цифрой 1, другое – 0. Благодаря таким особенностям двоичная система стала стандартом при построении ЭВМ.

Восьмеричная система счисления. Для записи чисел используется восемь чисел 0,1,2,3,4,5,6,7.

Шестнадцатеричная система счисления. Для записи чисел в шестнадцатеричной системе необходимо располагать шестнадцатью символами, используемыми как цифры. В качестве первых десяти используются те же, что и в десятичной системе. Для обозначения остальных шести цифр (в

Раздел 2. Информация и информационные процессы.

десятичной они соответствуют числам 10,11,12,13,14,15) используются буквы латинского алфавита – А, В, С, D, E, F.

Римская система счисления - непозиционная система счисления, в которой для записи чисел используются буквы латинского алфавита:

1 - I, 5 - V, 10 - X, 50 - L, 100 - C, 500 - D и 1000 - M.

Для правильной записи больших чисел римскими цифрами необходимо *сначала записать число тысяч, затем сотен, затем десятков и, наконец, единиц.* Натуральные числа записываются при помощи повторения этих цифр. При этом, *если большая цифра стоит перед меньшей, то они добавляются (принцип сложения)*, если же меньшая – перед большей, то меньшая вычитается из большей (**принцип вычитания**). Последнее правило применяется только во избежание четырехкратного повторения одной цифры. Например, I, X, C ставятся соответственно перед X, C, M для обозначения 9, 90, 900 или перед V, L, D для обозначения 4, 40, 400.

Например, VI = 5 + 1 = 6, IV = 5 - 1 = 4 (вместо III); XIX = 10 + 10 - 1 = 19 (вместо XVIII), XL = 50 - 10 = 40 (вместо XXXX).

Правила перевода чисел из одной системы счисления в другую:

1. Для перевода двоичного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 2, и вычислить по правилам десятичной арифметики:

$$X_2 = A_n \cdot 2^{n-1} + A_{n-1} \cdot 2^{n-2} + A_{n-2} \cdot 2^{n-3} + \dots + A_2 \cdot 2^1 + A_1 \cdot 2^0$$

Пример . Число 11101000_2 перевести в десятичную систему счисления.

$$11101000_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 232_{10}$$

2. Для перевода восьмеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 8, и вычислить по правилам десятичной арифметики:

$$X_8 = A_n \cdot 8^{n-1} + A_{n-1} \cdot 8^{n-2} + A_{n-2} \cdot 8^{n-3} + \dots + A_2 \cdot 8^1 + A_1 \cdot 8^0$$

Пример . Число 75013_8 перевести в десятичную систему счисления.

$$75013_8 = 7 \cdot 8^4 + 5 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 = 31243_{10}$$

Раздел 2. Информация и информационные процессы.

3. Для перевода шестнадцатеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 16, и вычислить по правилам десятичной арифметики:

$$X_{16} = A_n \cdot 16^{n-1} + A_{n-1} \cdot 16^{n-2} + A_{n-2} \cdot 16^{n-3} + \dots + A_2 \cdot 16^1 + A_1 \cdot 16^0$$

Пример. Число FDA_{16} перевести в десятичную систему счисления.

$$FDA_{16} = 15 \cdot 16^3 + 13 \cdot 16^2 + 10 \cdot 16^1 + 1 \cdot 16^0 = 64929_{10}$$

4. Для перевода десятичного числа в двоичную систему его необходимо последовательно делить на 2 до тех пор, пока не останется остаток, меньший или равный 1. Число в двоичной системе записывается как последовательность последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 22_{10} перевести в двоичную систему счисления.

$$22_{10} = 10110_2$$

22	2
22	11
0	10
1	5
1	4
0	2
1	2
0	2
1	1

5. Для перевода десятичного числа в восьмеричную систему его необходимо последовательно делить на 8 до тех пор, пока не останется остаток, меньший или равный 7. Число в восьмеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 571_{10} перевести в восьмеричную систему счисления.

$$571_{10} = 1073_8$$

571	8
56	71
11	64
8	8
3	7
0	8
1	0

6. Для перевода десятичного числа в шестнадцатеричную систему его необходимо последовательно делить на 16 до тех пор, пока не останется остаток, меньший или равный 15. Число в шестнадцатеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 7467_{10} перевести в шестнадцатеричную систему счисления.

$$7467_{10} = 1D2B_{16}$$

7467	16
7456	466
11	464
2	29
13	16

Раздел 2. Информация и информационные процессы.

7. Чтобы перевести число из двоичной системы в восьмеричную, его нужно разбить на триады (тройки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую триаду нулями, и каждую триаду заменить соответствующей восьмеричной цифрой.

Пример. Число 1001011_2 перевести в восьмеричную систему счисления.

$$001\ 001\ 011_2 = 113_8$$

8. Чтобы перевести число из двоичной системы в шестнадцатеричную, его нужно разбить на тетрады (четверки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую тетраду нулями, и каждую тетраду заменить соответствующей восьмеричной цифрой.

Пример. Число 1011100011_2 перевести в шестнадцатеричную систему счисления.

$$0010\ 1110\ 0011_2 = 2E3_{16}$$

9. Для перевода восьмеричного числа в двоичную необходимо каждую цифру заменить эквивалентной ей двоичной триадой.

Пример. Число 531_8 перевести в двоичную систему счисления.

$$531_8 = 101011001_2$$

10. Для перевода шестнадцатеричного числа в двоичную необходимо каждую цифру заменить эквивалентной ей двоичной тетрадой.

Пример. Число $EE8_{16}$ перевести в двоичную систему счисления.

$$EE8_{16} = 111011101000_2$$

11. При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно, необходим промежуточный перевод чисел в двоичную систему.

Пример 1. Число FEA_{16} перевести в восьмеричную систему счисления.

$$FEA_{16} = 111111101010_2$$

$$111\ 111\ 101\ 010_2 = 7752_8$$

Пример 2. Число 6635_8 перевести в шестнадцатеричную систему счисления.

$$6635_8 = 110110011101_2$$

$$1101\ 1001\ 1101_2 = D9D_{16}$$

ДВОИЧНОЕ КОДИРОВАНИЕ ТЕКСТА

Для обработки текстовой информации на компьютере необходимо представить ее в двоичной знаковой системе. Каждому знаку необходимо поставить в соответствие уникальный 8-битовый двоичный код, значение которого находится в интервале от 00000000 до 11111111 (в десятичном коде от 0 до 255).

Присвоение знаку конкретного двоичного кода – это вопрос соглашения, которое фиксируется в кодовой таблице. В настоящий момент существует пять кодировок кириллицы: КОИ-8, CP1251, CP866, ISO, Mac. Для преобразования текстовых документов из одной кодировки в другую

Раздел 2. Информация и информационные процессы.

существуют программы, которые называются Конверторы. Важно помнить, что тексты, созданные в одной кодировке, не будут правильно отображаться в другой.

На **1 символ** отводится **1 байт** (8 бит), всего можно закодировать $2^8 = 256$ символов.

С 1997 года появился новый международный стандарт **Unicode**, который отводит для кодировки одного символа **2 байта** (16 бит), и можно закодировать 65536 различных символов (Unicode включает в себя все существующие, вымершие и искусственно созданные алфавиты мира, множество математических, музыкальных, химических и прочих символов). Такого количества символов оказалось достаточно, чтобы закодировать не только русский и латинский алфавиты, цифры, знаки и математические символы, но и греческий, арабский, иврит и другие алфавиты.

ДВОИЧНОЕ КОДИРОВАНИЕ ГРАФИКИ

Пространственная дискретизация – перевод графического изображения из аналоговой формы в цифровой компьютерный формат путем разбиения изображения на отдельные маленькие фрагменты (точки) где каждому элементу присваивается код цвета.

Пиксель – минимальный участок изображения на экране, заданный определенным цветом.

Растровое изображение формируется из отдельных точек - пикселей, каждая из которых может иметь свой цвет. **Двоичный код изображения**, выводимого на экран храниться в видеопамяти. **Кодирование рисунка растровой графики** напоминает – мозаику из квадратов, имеющих определенный цвет.

Качество кодирования изображения зависит от:

- 1) размера точки (чем меньше её размер, тем больше количество точек в изображении);
- 2) количества цветов (чем большее количество возможных состояний точки, тем качественнее изображение). Палитра цветов – совокупность используемого набора цвета.

Качество растрового изображения зависит от:

- 1) разрешающей способности монитора – количество точек по вертикали и горизонтали;
- 2) используемой палитры цветов (16, 256, 65536 цветов);
- 3) глубины цвета – количество бит для кодирования цвета точки.

Для хранения **черно-белого** изображения используется **1 бит**.

Цветные изображения формируются в соответствии с двоичным кодом цвета, который хранится в видеопамяти. Цветные изображения имеют различную глубину цвета. Цветное изображение на экране формируется за счет смешивания трех базовых цветов – красного, зеленого и синего. Для получения богатой палитры базовым цветам могут быть заданы различные интенсивности.

Векторная графика основным элементом имеет линию, как прямую, так и кривую. В векторной графике в памяти хранятся параметры линии, которые не зависят от ее длины. **Линия** – элементарный объект векторной графики.

Так как основой векторной графики является линия, то ее называют *объектно-ориентированной*. Каждый раз при выводе линии на экран происходит вычисление координат и цветности этих точек, поэтому векторную графику иначе называют *аналитической или вычисляемой*.

У каждого элементарного объекта векторного изображения кодируется его положение через координаты точек и длину радиуса, тип линии, толщина и цвет.

Фрактальная графика является вычисляемой, но информация об объектах в памяти не хранится, а хранятся только математические формулы. Меняя значения коэффициентов в этих формулах, можно получать совершенно другое изображение. **Фрактал** — это объект, отдельные элементарные части которого повторяют (наследуют) свойства своих «родительских» структур.

Раздел 2. Информация и информационные процессы. ДВОИЧНОЕ КОДИРОВАНИЕ ЗВУКА

В аналоговой форме звук представляет собой волну с непрерывно меняющейся амплитудой и частотой. На компьютере работать со звуковыми файлами начали с начала 90-х годов. В основе кодирования звука с использованием ПК 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). Качество воспроизведения закодированного звука зависит от – частоты дискретизации и её разрешения (глубины кодирования звука - количество уровней).

Временная дискретизация – способ преобразования звука в цифровую форму путем разбиения звуковой волны на отдельные маленькие временные участки, где амплитуды этих участков квантуются (им присваивается определенное значение).

Это производится с помощью аналого-цифрового преобразователя, размещенного на звуковой плате. Таким образом, непрерывная зависимость амплитуды сигнала от времени заменяется дискретной последовательностью уровней громкости. Современные 16-битные звуковые карты кодируют 65536 различных уровней громкости или 16-битную глубину звука (каждому значению амплитуды сигнала звука присваивается 16-битный код).

Качество кодирования звука зависит от:

- 1) глубины кодирования звука - количество уровней звука;
- 2) частоты дискретизации – количество изменений уровня сигнала в единицу времени (как правило, за 1 сек).

$$N = 2^i$$

N – количество различных уровней сигнала

i – глубина кодирования звука

Информационный объем звуковой информации равен:

$$I = i * k * t,$$

где i – глубина звука (бит),

k – частота вещания (качество звука) (Гц) (48 кГц – аудио CD),

t – время звучания (сек).

ПРЕДСТАВЛЕНИЕ ВИДЕОИНФОРМАЦИИ

В последнее время компьютер все чаще используется для работы с видеоинформацией. Простейшей такой работой является просмотр кинофильмов и видеоклипов. Следует четко представлять, что обработка видеоинформации 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.

2.2. Основные информационные процессы и их реализация с помощью компьютеров: обработка информации.

Компьютер (англ. computer — вычислитель) представляет собой программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами.

Существует два основных класса компьютеров:

Раздел 2. Информация и информационные процессы.

- а) цифровые компьютеры, обрабатывающие данные в виде двоичных кодов;
- б) аналоговые компьютеры, обрабатывающие непрерывно меняющиеся физические величины (электрическое напряжение, время и т.д.), которые являются аналогами вычисляемых величин.

В настоящее время подавляющее большинство компьютеров являются цифровыми.

Основу компьютеров образует аппаратура (HardWare), построенная, в основном, с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (SoftWare) — заранее заданных, четко определённых последовательностей арифметических, логических и других операций.

Любая компьютерная программа представляет собой последовательность отдельных команд.

Команда — это описание операции, которую должен выполнить компьютер. Как правило, у команды есть свой код (условное обозначение), сходные данные (операнды) и результат.

Разнообразие современных компьютеров очень велико. Но их структуры основаны на общих логических принципах, позволяющих выделить в любом компьютере следующие главные устройства:

- а) память (запоминающее устройство, ЗУ), состоящую из перенумерованных ячеек;
- б) процессор, включающий в себя устройство управления (УУ) и арифметико-логическое устройство (АЛУ);
- в) устройство ввода;
- г) устройство вывода.

Эти устройства соединены каналами связи, по которым передается информация.

Арифметические и логические основы компьютеров.

Логический элемент компьютера — это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы **И**, **ИЛИ**, **НЕ**, **И—НЕ**, **ИЛИ—НЕ** и другие.

С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера.

Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем.

Работу логических элементов описывают с помощью **таблиц истинности**.

Логика — это наука о формах и способах мышления. Это учение о способах рассуждений и доказательств.

Мышление всегда существует через понятия, высказывания и умозаключения.

Понятие — это форма мышления, которая выделяет существенные признаки предмета или класса предметов, позволяющих отличать их от других.

Высказывание — это форма мышления, в которой что-либо утверждается или отрицается о реальных предметах, их свойствах и отношениях между ними. Высказывание может быть **истинно**, или **ложно**.

Умозаключение — это форма мышления, с помощью которой из одного или нескольких высказываний (посылок) может быть получено новое высказывание (вывод).

Алгебра — это наука об общих операциях, аналогичных сложению и умножению, которые выполняются не только над числами, но и над другими математическими объектами, в том числе и над высказываниями. Такая алгебра называется **алгеброй логики**. Алгебра логики отвлекается от

Раздел 2. Информация и информационные процессы.

смысловой содержательности высказываний и принимает во внимание только истинность или ложность высказывания.

Логическая переменная – это простое высказывание, содержащее только одну мысль. Ее символическое обозначение – латинская буква (например, А, В, Х, У и т.д.) Значением логической переменной могут быть только константы ИСТИНА и ЛОЖЬ (1 и 0).

Логическая функция – это составное высказывание, которое содержит несколько простых мыслей, соединенных между собой с помощью логических операций. Ее символическое обозначение – F (А, В,...).

Логическая операция – это логическое действие.

Логическая операция НЕ – операция отрицание (дополнение или *инверсия*, обозначают NOT (-). Если А – истинно, то \bar{A} – ложно и наоборот. Результат отрицания всегда противоположен значению аргумента.

Таблица истинности:

A	\bar{A}
0	1
1	0

Логическая операция И - *конъюнкция*, или логическим умножением. Операция И (обозначается «И», «and», «&», $A \cdot B$) имеет результат «истина» только в том случае, если оба ее операнда истинны.

Таблица истинности $F = A \wedge B$:

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Логическая операция ИЛИ (обозначается «ИЛИ», «or», $A + B$, $A \vee B$) называется *дизъюнкцией* или логическим сложением и дает «истину», если значение «истина» имеет хотя бы один из операндов. Разумеется, в случае, когда справедливы оба аргумента одновременно, результат по-прежнему истинный.

Таблица истинности $F = A \vee B$:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Логическое следование: импликация – связывает два простых логических выражения, из которых первое является условием (А), а второе (В) – следствием из этого условия. Результатом импликации является ЛОЖЬ только тогда, когда условие А истинно, а следствие В ложно. Обозначается символом "следовательно" и выражается словами ЕСЛИ ... , ТО ...

Таблица истинности $F = A \rightarrow B$:

A	B	F
0	0	1
0	1	1
1	0	0
1	1	1

Логическая равнозначность: эквивалентность – определяет результат сравнения двух простых логических выражений А и В. Результатом эквивалентности является новое логическое

Раздел 2. Информация и информационные процессы.

выражение, которое будет истинным тогда и только тогда, когда оба исходных выражения одновременно истинны или ложны. Обозначается символом "эквивалентности".

Таблица истинности $F = A \leftrightarrow B$:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Порядок выполнения логических операций в сложном логическом выражении: 1. инверсия →

2. Конъюнкция → 3. Дизъюнкция → 4. Импликация → 5. Эквивалентность. Для изменения указанного порядка выполнения операций используются круглые скобки.

Алгоритмы и способы их описания

Алгоритм - это четкое описание последовательности действий, которые необходимо выполнить для решения поставленной задачи.

Программа - это алгоритм, записанный на языке программирования.

Языком программирования называется специальный язык, понятный для компьютера.

Программирование - это процесс создания отладки и тестирования программ.

Создание любой программы начинается с разработки алгоритма. Именно четкое описание последовательности действий позволяет мысленно представить будущую программу. Построив алгоритм, программист мыслит четко, последовательно, однозначно - так, как и будет впоследствии мыслить компьютер.

Для алгоритма характерны следующие свойства:

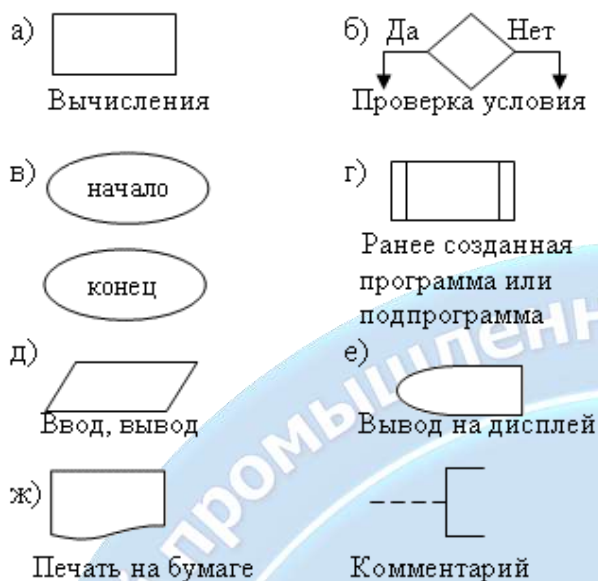
1. **Дискретность** - алгоритм должен быть представлен как последовательное выполнение простых шагов.
2. **Шагом** называется каждое действие алгоритма.
3. **Определенность** - каждое действие алгоритма должно быть четким и однозначным.
4. **Результативность** - алгоритм должен приводить к решению задачи за определенное число шагов.
5. **Массовость** - алгоритм составляется в общем виде, т.е. он должен быть применим к ряду задач, различающимся исходными данными.

Способы записи алгоритма

1. **Формальный** - запись алгоритма словесно, на естественном языке.
2. **Графический** - изображение алгоритма в виде блок-схемы.

Раздел 2. Информация и информационные процессы.

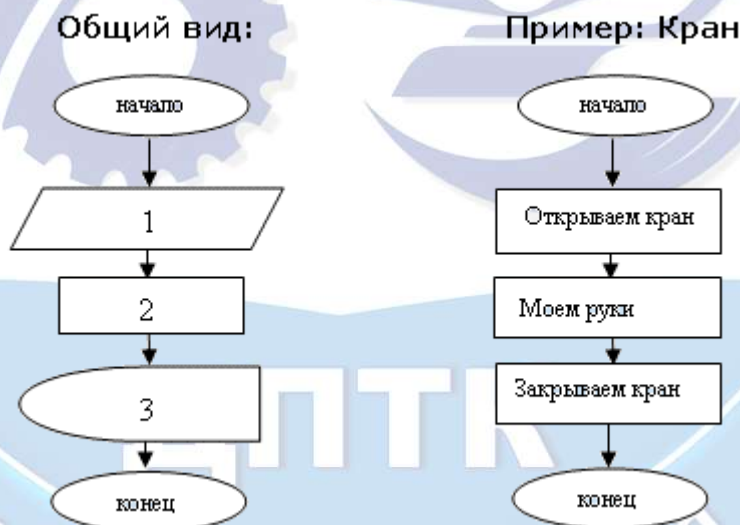
Действия алгоритма изображаются следующими геометрическими фигурами:



Виды алгоритмов

В зависимости от поставленной задачи и последовательности выполняемых шагов различают следующие виды алгоритмов:

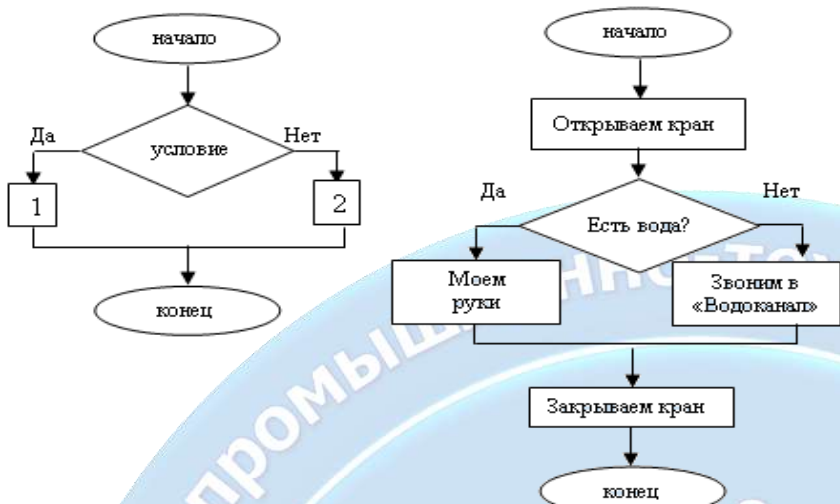
- 1. Линейный** - шаги алгоритма следуют один за другим не повторяясь, действия происходят только в одной заранее намеченной последовательности.



Блоки алгоритма 1, 2, 3 выполняются именно в такой последовательности, после чего алгоритм достигает цели и заканчивается.

Раздел 2. Информация и информационные процессы.

2. **Алгоритм с ветвлением** - в зависимости от выполнения или невыполнения условия, исполняется **Общий вид:** либо одна, **Пример:** либо другая ветвь алгоритма

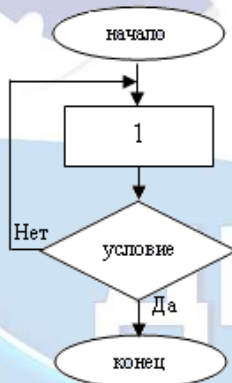


В данном алгоритме проверяется выполнение условия, и если оно выполняется, то есть на вопрос можно ответить "Да", исполняется блок алгоритма 1 (одно или несколько действий), а если не выполняется - ответ на вопрос отрицательный, то исполняется блок 2.

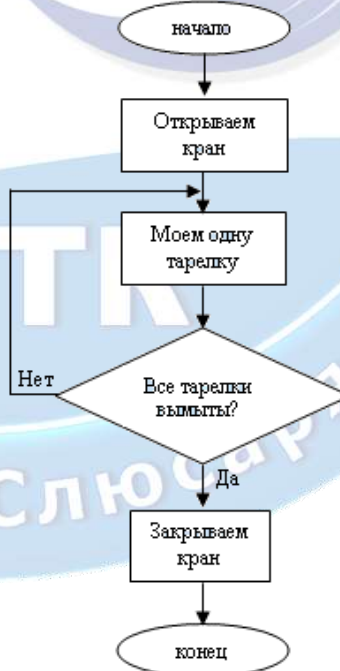
Примечания: одно из блоков: 1 или 2 может не быть вовсе. Тогда в одном из случаев будут выполняться какие-либо действия, а в другом - ничего не будет выполняться.

3. **Циклический** - блоки алгоритма выполняются до тех пор, пока не будет выполнено определенное условие.

Общий вид:



Пример: Кран



Блок алгоритма 1 будет выполняться один или несколько раз до тех пор, пока не выполнится условие.

Раздел 2. Информация и информационные процессы.

Алгоритм выполняется так: выполняется блок 1, проверяется условие, если оно не выполняется, то блок 1 выполняется снова и условие проверяется заново. При выполнении условия алгоритм заканчивается.

Этапы создания программы:

1. **Постановка задачи** - составление точного и понятного словесного описания того, как должна работать будущая программа, что должен делать пользователь в процессе ее работы.
2. **Разработка интерфейса** (интерфейс - способ общения) - создание экранной формы (окна программы).
3. **Составление алгоритма.**
4. **Программирование** - создание программного кода на языке программирования.
5. **Отладка программы** - устранение ошибок.
6. **Тестирование программы** - проверка правильности ее работы.
7. **Создание документации, помощи.**

Основные понятия Visual Basic

Visual Basic – один из первых языков программирования, поддерживающих *событийно-управляемое* программирование. Основной смысл событийно-управляемого программирования следующий: вместо скрупулезного описания каждого шага Вы лишь указываете, как реагировать на различные события (действия пользователя): выбор команды, щелчок в окне, перемещение мыши и т.д. Вы создаете не одну большую программу, а приложение, состоящее из набора взаимодействующих микропрограмм (процедур), управляемых пользователем.

Visual Basic – это среда разработки приложений под Windows, в которую включено все, что необходимо для создания, модификации, тестирования, корректирования и компиляции ваших программ.

Слово *Visual* «визуальный» (наглядный) означает способ разработки пользовательского интерфейса программы.

Интерфейс – совокупность средств, обеспечивающих физическое или логическое взаимодействие устройств и программ вычислительной системы. Здесь мы вкладываем в термин разработка интерфейса более узкий смысл – изобретение и создание такого способа ввода и вывода информации, который был бы максимально прост, удобен и приятен пользователю.

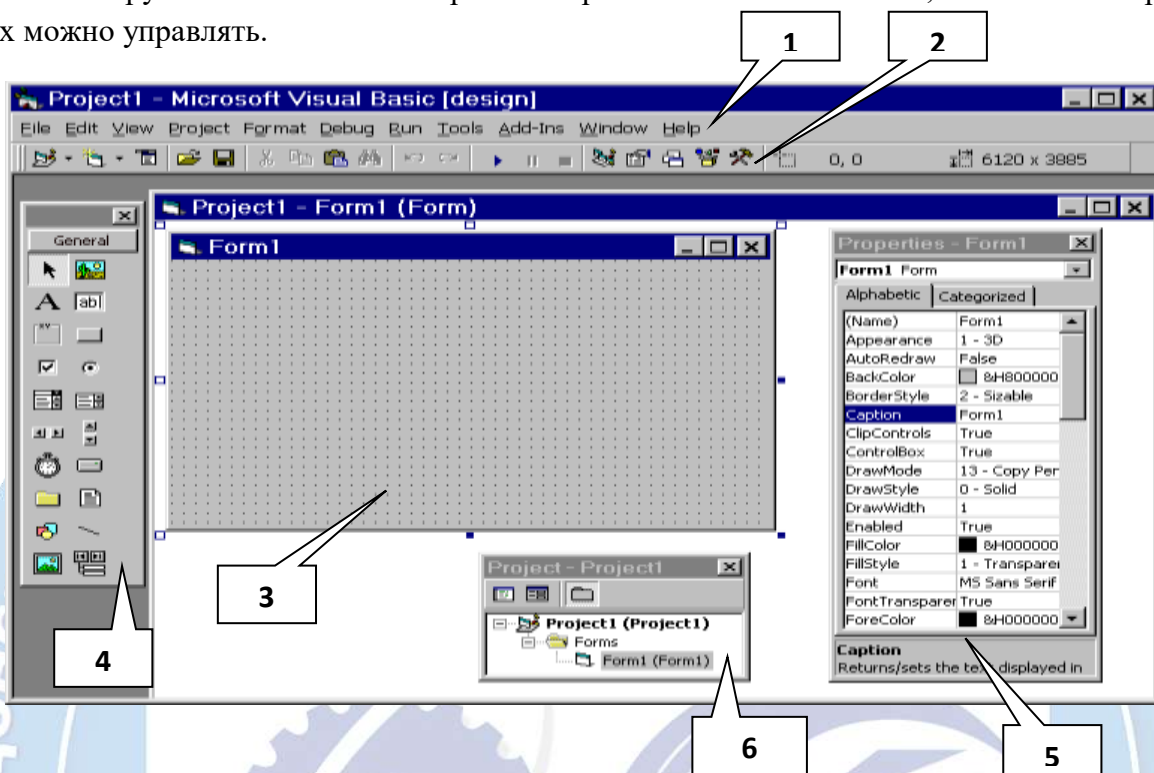
Этапы создания Windows-приложений

1. **Постановка задачи** – составление по возможности точного и понятного словесного описания того, как должно работать будущее приложение, - что должен делать пользователь в процессе его работы.
2. **Разработка интерфейса** – создание экранной формы (окна приложения) со всеми находящимися на этой форме объектами и свойствами этих объектов
3. **Программирование** – определение того, какие события будут происходить в процессе работы приложения, составление алгоритмов процедур для этих событий и написание программных кодов этих процедур.
4. **Отладка** – устранение логических ошибок в процедурах и достижение того, чтобы приложение удовлетворительно работало в среде проектирования.
5. **Сохранение проекта и компиляция** (создание исполняемого приложения).

Раздел 2. Информация и информационные процессы.

Экран проектирования в системе Visual Basic

После загрузки Visual Basic на экране отображается несколько окон, положением и размерами которых можно управлять.



Экран содержит окна:

1. Строка меню

Меню содержит команды, используемые при работе Visual Basic. Кроме стандартных меню **File, Edit, View, Window, Help**, здесь расположены меню, обеспечивающие доступ к функциям программирования, например, **Project, Format, Debug**

2. Панель инструментов (Toolbars Standard)

Предоставляет быстрый доступ к наиболее часто используемым командам среды программирования

3. Форма (Form) – окно будущего приложения

4. Панель элементов управления (Toolbox)

Панель обеспечивает проектировщика набором инструментов, необходимых во время разработки для размещения элементов управления на форме

5. Окно свойств (Properties)

Перечисляет установленные свойства для выбранной формы или элемента управления.

6. Окно проводника проекта (Project Explorer)

Представляет список форм и модулей текущего проекта. Проект – это набор файлов, используемых для построения приложения

Объекты управления и их свойства

Объектом называется некая сущность, которая, во-первых, четко проявляет свое поведение, а во-вторых, является представителем некоторого класса подобных себе объектов.

Каждый объект имеет:

Раздел 2. Информация и информационные процессы.

- ◆ *свойства*
- ◆ *методы*
- ◆ *события*

Свойства - это показатели, характеризующие объект.

Методы - это действия, которые можно произвести с объектом.

События - это действия, которые происходят с объектом.

Классом объектов называется общее описание объектов, для которых характерно наличие множества общих свойств и общих действий. Примером класса может служить класс Командная кнопка – общее описание кнопок в окнах приложений. Эти кнопки могут быть всех цветов и размеров, но должны иметь множество общих свойств и других характеристик, например, событий, которые для этих объектов одинаковы.

Форма – это окно будущего приложения. Форма обладает свойствами, определяющими ее внешний вид, методами, определяющими ее поведение, и событиями, которые определяют ее взаимодействие с пользователем.

Элементы управления – это объекты, содержащиеся внутри объекта - *Форма*. Каждый тип элемента управления имеет свой собственный набор свойств, методов и событий, что делает его пригодным для определенной цели.

Свойства в Visual Basic - количественно измеряемый атрибут объекта (элемента управления или формы). Значения свойств можно задать на стадии проектирования в окне свойств. Активизировать окно свойств, можно по-разному:

- в строке меню **View / Properties Window**
- клавиша **F4**
- кнопка на панели инструментов 

Установка значений свойств объектов

Значения свойств объектов можно менять двумя способами:

– *при проектировании:*

В каждый момент проектирования только один объект является выделенным (активным). Он окружен рамкой из восьми маркеров. В окне свойств отображается список свойств именно активного объекта. Новое значение свойства вводится в окне свойств.

– *при выполнении приложения:*

в программный код включается команда, имеющая следующий общий вид

ИмяОбъекта.Свойство=Значение

Получение значений свойств

Получают значение свойств тогда, когда хотят определить состояние объекта до выполнения каких-либо действий из кода. Общий вид команды следующий:

Переменная=ИмяОбъекта.Свойство

Раздел 2. Информация и информационные процессы.

Основные свойства объектов управления

Свойство	Назначение
Name	Имя объекта
Caption	Заголовок
Visible	Видимость
BorderStyle	Стиль границ
FontBold	Полужирный шрифт
FontItalic	Курсив
FontName	<i>Тип шрифта</i>
FontSize	Размер шрифта
FontUnderline	Шрифт подчеркнутый
Enabled	Доступ
Left	Координата по горизонтали
Top	Координата по вертикали
Height	Высота объекта
Width	Ширина объекта
BackColor	Цвет фона
ForeColor	Цвет шрифта
BorderColor	Цвет границ
FillStyle	Стиль заполнения
MousePointer	Вид курсора при наведении на объект







Свойства, используемые для управления формой

Свойство	Назначение
MinButton MaxButton	Наличие кнопки минимизации окна Наличие кнопки максимизации окна
KeyPreview	Определяет, вызываются ли процедуры обработки события клавиатуры формы перед событиями клавиатуры элементов управления
Left Top	Определяют местоположение формы по отношению к левому верхнему углу экрана монитора
Icon	Устанавливает отображаемый при сворачивании формы значок
WindowState	Состояние окна после загрузки приложения
Auto Redraw	Определяет возможность автоматического перерисования
ClipControls	Определяет необходимость перерисовки всего объекта или появляющейся части
ControlBox	Определяет наличие кнопки системного меню на форме
DrawWidth	Определяет ширину рисуемой линии (точки)
ScaleHeight, ScaleWidth	Определяет число единиц измерения по вертикали и горизонтали
ScaleLeft, ScaleTop	Определяет координаты верхнего левого угла








Основные объекты управления и их специфичные свойства:

Пиктограмма / Назначение	Специфические свойства
--------------------------	------------------------





Раздел 2. Информация и информационные процессы.

	<p><i>Командная кнопка (CommandButton)</i></p>	<p><i>Default</i> при значении True командная кнопка определена как кнопка по умолчанию, т.е. при нажатии Enter она будет нажата.</p> <p><i>Cancel</i> определяет как кнопку отмены по умолчанию т.е. при нажатии Esc она будет нажата.</p> <p><i>Style</i> стиль, принимает два значения: <i>стандартный и графический</i></p> <p>Если стиль Graphical, то можно менять свойства:</p> <p><i>Picture</i> картинка</p> <p><i>DownPicture</i>- картинка внизу</p> <p><i>DisabledPicture</i>- картинка если у кнопки нет доступа</p>
	<p><i>Текстовое окно (TextBox)</i> – экранная область, в которое можно вводить текст</p>	<p><i>MaxLength</i> максимальная длина, если значение нуль, то можно вводить любое кол-во символов</p> <p><i>Multiline</i> значение False запрещает ввод более одной строки, значение True – разрешает ввод нескольких строк после нажатия Enter</p> <p><i>ScrollBars</i> наличие (1, 2, 3) или отсутствие (0) линеек прокруток в текстовом поле</p> <p><i>Text</i> текст, отображаемый в поле.</p> <p><i>Locked</i> блокировка редактирования</p> <p>Следующие свойства доступны в режиме выполнения</p> <p><i>SelStart</i> число, указывающее место вставки в строке текста</p> <p><i>SelLength</i> количество выделяемых символов</p> <p><i>SelText</i> определяет выделенный текст</p>
	<p><i>Метка (Label)</i> – применяется для отображения текста, который пользователь не может редактировать</p>	<p><i>Alignment</i> выравнивание</p> <p><i>AutoSize</i> автоподстройка размера. При значении True размер метки подгоняется под размер текста, заданный свойством Caption. Если значение False метка сохраняет размер, установленный при проектировании</p> <p><i>WordWrap</i> перенос слов</p> <p><i>BorderStyle</i> стиль границ</p>
	<p><i>Переключатель (OptionButton)</i> для организации выбора из нескольких возможностей. Выбор одного сбрасывает все другие переключатели.</p>	
	<p><i>Флажок (Check Box)</i> - для организации выбора типа да/нет. Работают независимо друг от друга, пользователь может установить любое их число одновременно.</p>	
	<p><i>Рамка (Frame)</i> для объединения объектов в группы</p>	

Раздел 2. Информация и информационные процессы.

 	<p><i>Линейки прокрутки</i> (Scroll bar) горизонтальная и вертикальная действуют совершенно одинаково. Эти объекты позволяют узнавать о позиции движка (scrollbox), кроме того контролировать диапазон действия линейки прокрутки и дискретность перемещения движка</p>	<p><i>LargeChange</i> определяет величину, которая добавляется или вычитается из значения Value при щелчке внутри линейки прокрутки</p> <p><i>Max</i> число, определяющее крайнюю правую или нижнюю позицию</p> <p><i>Min</i> число, определяющее крайнюю левую или верхнюю позицию</p> <p><i>SmallChange</i> -определяет величину, которая добавляется или вычитается из значения Value при щелчке на одной из стрелок на концах линейки прокрутки</p> <p><i>Value</i> число, которое отражает текущую позицию движка на линейке</p>
	<p><i>Таймер (Timer)</i> – это объект, способный инициировать события через регулярные промежутки времени</p>	<p><i>Interval</i> число (от 0 до 65535), определяющее интервал времени в мс между двумя событиями. Интервал, равный нулю, отключает таймер</p>
	<p><i>Линия (Line)</i> –для вычерчивания линий на поверхности формы. Не поддерживает никаких событий.</p>	<p><i>X1, Y1</i> координаты левого края линии</p> <p><i>X2, Y2</i> координаты правого края линии</p> <p><i>BorderWidth</i> толщина линии</p> <p><i>BorderStyle</i> стиль линии</p>
	<p><i>Список (ListBox)</i>-предоставляет список возможных вариантов выбора, позволяет ограничить ввод элементами списка</p>	
	<p><i>Комбинированный список (ComboBox)</i> Совмещает возможности списка и текстового окна, содержит редактируемое поле</p>	
	<p><i>Окно рисунка (PictureBox)</i>- для размещения графической информации в определенных участках формы. Требуют больше памяти и времени на обработку, больше подходят для динамических объектов. Может выполнять функции контейнера для других элементов управления.</p>	<p><i>Picture</i> позволяет выводить растровую картинку (.bmp), либо значок (.icon)</p> <p><i>AutoSize</i> автоподстройка размера</p>

Раздел 2. Информация и информационные процессы.

	<p><i>Изображение (Image)</i> – для размещения графической информации в определенных участках формы. Удобно использовать в статической среде (не предполагается изменение)</p>	<p><i>Picture</i> позволяет выводить растровую картинку (.bmp), метафайл, файлы JPEG или GIF либо значок (.icon)</p> <p><i>Stretch</i> Растягивать. Если значение True картинка подгоняется под размер элемента управления.</p>
	<p><i>Контур или фигура (Shape)</i> для вычерчивания контуров в виде прямоугольника, окружности, овала, квадрата, прямоугольника, квадрата с закругленными углами</p>	<p><i>Shape</i> тип контура</p> <p><i>FillStyle</i> стиль заполнения</p> <p><i>BorderStyle</i> стиль границ контура</p> <p><i>BorderWidth</i> толщина контура</p>
	<p><i>Список файлов (FileListBox)</i> позволяет узнать, какие есть файлы на дисках системы и выбрать один из них</p>	<p><i>Pattern</i> определение шаблона для списка файлов</p> <p>Следующие свойства определяют тип отображаемых файлов</p> <p><i>Archive</i> архивный</p> <p><i>System</i> системный</p> <p><i>Hidden</i> скрытый</p> <p><i>ReadOnly</i> только для чтения</p>
	<p><i>Список каталогов (DirListBox)</i> позволяет узнать, какие есть каталоги на дисках системы и выбрать один из них</p>	<p><i>Path</i> позволяет установить или получить текущий каталог</p>

Наименование объектов Visual Basic

При изменении имени (*Name*) объектов Visual Basic рекомендуется использовать следующую простую схему:

- начинать название с трехбуквенного префикса;
- использовать только буквы, цифры и знак подчеркивания (_);
- использовать не более 40 символов.

Создатели Visual Basic рекомендуют начинать название с трехбуквенного префикса в соответствии с типом объекта. Например, у вас может быть командная кнопка с названием **cmdCancel** и форма **frmMain**. Рекомендуемые префиксы перечислены в табл.

Объект	Рекомендуемый префикс
Форма	Frm
Флажок	Chk
Комбинированное окно	Cbo
Командная кнопка	Cmd
Окно данных	Dat
Список каталогов	Dir
Список дисков	Dsk
Рамка	Fra

Раздел 2. Информация и информационные процессы.

Сетка	Grd
Горизонтальная линейка прокрутки	Hsb
Изображение	Img
Метка	Lbl
Линия	Lin
Список	Lst
Меню	Mnu
Переключатель	Opt
Окно рисунка	Pic
Фигура	Shp
Текстовое окно	Txt
Таймер	Tmr
Вертикальная линейка прокрутки	Vsb

События

Событием называется характеристика класса объектов, описывающая внешнее воздействие, на которое реагирует объект этого класса во время работы приложения.

Программы на Visual Basic управляются событиями, другими словами – действия пользователя вызывают выполнение различных процедур. Работает это примерно так: программа ждет, пока пользователь не сделает что-либо, т.е. пока не произойдет событие; затем программа реагирует на это событие, запуская соответствующую процедуру или процедуры, затем программа снова терпеливо ждет следующего события.

События, возникающие при работе с мышью:

Событие	Описание	Параметры событий
Click	Щелчок	
DbtClick	Двойной щелчок	
MouseDown	Кнопка мыши нажата	
MouseUp	Нажатая кнопка мыши отпущена	
MouseMove	Мышь перемещается из своей текущей позиции	
DragDrop	Завершение перетаскивания	Source – ссылка на объект, который был перемещен, X, Y – позиция курсора
DragOver	«Буксируемый» объект попадает в область другого объекта	Stale – принимает значения 0 – область занята, 1 – область свободна

События, возникающие при работе с клавиатурой:

Событие	Описание	Параметры событий
KeyPress	Нажата клавиша, соответствующая символу ASCII	KeyAscii - значение ASCII-кода нажатой клавиши
KeyDown	Нажата любая клавиша на клавиатуре	
KeyUp	Отпущена любая клавиша	

Раздел 2. Информация и информационные процессы.

События, связанные с фокусом:

Объект, *имеющий фокус*, может получать вводимую пользователем информацию с помощью мыши и клавиатуры.

Событие	Описание
GotFocus	Получении фокуса
LostFocus	Потеря фокуса

События, специфичные для объектов:

Событие	Описание	Для какого объекта характерны
Load UnLoad	Загрузка Закрыть форму	Форма
Resize	Изменение размера	Форма, Картинка
Change	Изменение	Текстовое окно, Полосы прокрутки, Список каталогов Комбинированный список,
Scroll	Прокрутка	Полосы прокрутки, Список, Комбинированный список, Список файлов, Список каталогов
Timer	Истечение интервала времени	Таймер

Окно кода процедуры обработки события

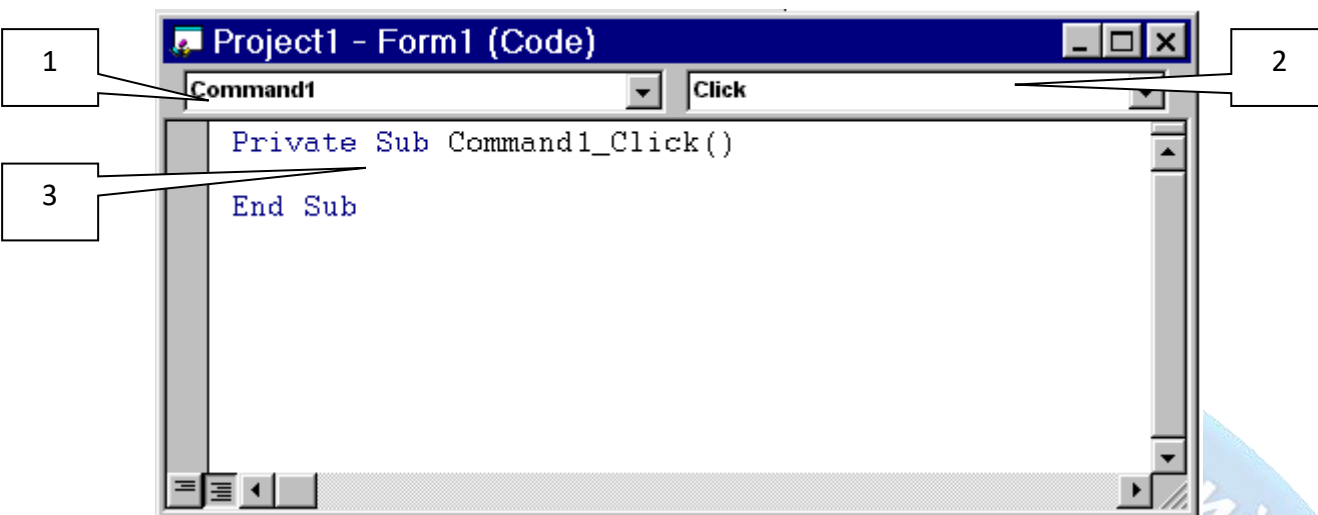
Любой объект можно связать с набором процедур, исполняемых в строго определенные моменты. Процедура (*procedure*) – это группа операторов языка. Исполняется процедура – исполняются ее операторы. Так или иначе, весь составленный вами исполняемый код обязательно помещается в какую-нибудь процедуру. Процедура, присвоенная объекту, связана с определенным событием и поэтому называется *процедурой обработки события*. Важно отметить, что с одним объектом могут быть связаны несколько событий.

Имя процедуры обработки события для элемента управления составляется из имени элемента управления (Name), знака подчеркивания (_) и имени события.

Для открытия окна кода процедуры существует три способа:

1. двойной щелчок по объекту
2. клавиша **F7**
3. в меню выбрать **View / Code**

Раздел 2. Информация и информационные процессы.



1. список объектов формы
2. список событий объекта
3. процедура

Процедуры по умолчанию не делают ничего; они состоят лишь из объявления процедуры (*Sub*) и оператора, помечающего конец процедуры (*End Sub*)

Программный код вводится между строками **Private Sub** и **End Sub**

Понятие метода в Visual Basic

Visual Basic предоставляет большое количество встроенных процедур и функций (функции – это особый класс процедур, которые возвращают значения аргументов). Эти процедуры и функции описаны в справочной системе или документации на систему программирования.

Процедуры и функции уже не одно десятилетие являются неотъемлемой частью любого языка программирования. Лишь сравнительно недавно, когда появились языки, поддерживающие объектно-ориентированное программирование, были введены особые типы процедур и функций, названные **методами**. Методы «работают» как процедуры и функции, но принадлежат конкретным объектам, так же как и свойства. Методы объекта определяют те задачи, которые может выполнить данный объект.

Чтобы вызвать метод, надо указать имя объекта и через точку имя метода.

Некоторые методы предусмотрены практически для всех типов объектов, другие же более специфичны. В таблице приводится список основных методов, элементы управления, для которых они определены, и краткое описание действия.

Название метода	Элементы управления, для которых используются методы	Описание действия
AddItem	ListBox, ComboBox	Добавление элемента в список
Circle	Form, PictureBox	Рисование дуги, эллипса или окружности
Clear	ListBox, ComboBox	Удаление всех элементов списка
Cls	Form, PictureBox	Очистка от графических элементов или текста
Drag	Все, кроме Line, Menu,	Перемещение объекта

Раздел 2. Информация и информационные процессы.

	Shape, Timer	
Hide	Form	Спрятать форму с экрана без ее выгрузки
Line	Form, PictureBox	Рисование линии или прямоугольника
Move	Все, кроме Timer и Menu	Перемещение элемента
Point	Form, PictureBox	RGB (red-green-blue) цвет точки
PopupMenu	Form	Вывод всплывающего меню в заданной точке формы
Print	Form, PictureBox.	Печать строки на объекте
Pset	Form, PictureBox.	Точка на объекте
Refresh	Все	Немедленная перерисовка на экране
RemoveItem	ListBox, ComboBox	Удаление элемента списка или ячеек в сетке
Scale	Form, PictureBox.	Координаты объекта
SetFocus	CheckBox, ComboBox, CommandButton, FileListBox, Form, HscrollBar, ListBox, OptionButton, PictureBox, TextBox, VScrollBar	Установка курсора
Show	Form	Вывод формы на экран
ZOrder	Все	Расположение на переднем или заднем плане

Проект приложения

Программный проект – это совокупность частей, составляющих будущее Windows-приложен. Любой проект должен состоять из одной или нескольких *экранных форм* и одного или нескольких *программных модулей*.

Структура проекта

При создании приложения в Visual Basic программист работает с проектами. Проект (*project*) - это набор файлов, используемых для построения приложения. В этот набор входят:

Файл формы(для каждой формы) – содержит описание формы и ее элементов управления, включая установленные значения свойств, а также используемые в форме процедуры обработки событий. Файл формы имеет расширение **.FRM**

Стандартные модули хранятся в файлах с расширением .BAS (они могут использоваться при решении не одной, а нескольких задач и в данном пособии не рассматриваются, как и некоторые другие)


Файл проекта – содержит сведения о размещении всех других файлов проекта, имеет расширение **.VBP**.

Все файлы проекта могут быть объединены в один исполняемый файл (с расширением **.EXE**).


Запуск проекта

Для запуска проекта используйте один из приведенных способов:

Раздел 2. Информация и информационные процессы.

- с помощью меню выбрать **Run / Start**
- нажать клавишу **F5**
- нажать кнопку запуска на панели инструментов 


Сохранение проекта

Для сохранения следует выполнить **File / Save Project** или нажать кнопку  на панели инструментов. Visual Basic сначала предложит сохранить форму, а затем проект.

В появившемся диалоговом окне :

- установите диск и каталог назначения;
- измените имя файла формы;
- нажмите кнопку **Сохранить**
- установите диск и каталог назначения;
- измените имя файла проекта;
- нажмите кнопку **Сохранить**

Открытие проекта

Для открытия существующего проекта следует выполнить **File / Open Project**  или нажать на кнопке панели инструментов. В появившемся диалоговом окне:

- установите диск и каталог назначения;
- выберите имя файла проекта;
- нажмите **Открыть**

Данные. Выражения. Функции

Данные в Visual Basic

Любая программа выполняет обработку данных. Данные могут быть постоянными величинами (константы) и переменными величинами. Переменные и постоянные величины должны быть отнесены к какому-либо типу данных. В Visual Basic существует набор стандартных типов данных, кроме того можно объявить свой собственный тип данных.

Типы данных

Типом данных называется способ хранения и представления данных в компьютерной системе. Стандартные типы данных, поддерживаемые языком Visual Basic, приведены в таблице.

Тип данных	Диапазон значений	Объем занимаемой памяти
Integer (целое)	от -32768 до 32767	2 байта
Long (длинное целое)	от -2147483648 до 2147483647	4 байта
Byte (короткое целое число)	от 0 до 255	1 байт
Boolean (булево)	False, True	2 байта
Single (десятичное обычной точности)	от 1,41 E-45 до 3,4 E+38 (по модулю)	4 байта
Double (десятичное двойной точности)	от 4,95 E-324 до 1,79 E+308 (по модулю)	8 байт
String (строка)	Строка любых символов	1 байт на каждый символ

Раздел 2. Информация и информационные процессы.

Object (объект)	Ссылка на объект	4 байта
Variant (произвольный)		16 байт (числ. тип) 22 байта +длина строки (строковый тип)
Date (дата)	Значения даты и времени диапазон от 01 янв. 100 г до 31 дек. 9999г.	8 байт
Currency Десятичное протяженное	Значением является число до 15 знаков до десят.точки до 4 знаков после точки.	8 байт

Константы

Константы – это такие величины, которые в процессе выполнения программы не изменяют своих значений. Visual Basic имеет целый ряд внутренних (системных) констант, которые имеют префикс **vb**. В качестве примера - *Таблица констант дней недели*

Имя константы	Значение
VbSunday	1
VbMonday	2
VbTuesday	3
VbWednesday	4
VbThursday	5
VbFriday	6
VbSaturday	7

Константы, определяющие цвет, смотрите в приложении *Учебного пособия часть I*

Константа **VbTab** делает промежутки при выводе информации.

Подробно константы системы Visual Basic перечислены в библиотеках объектов VB и VBA в окне **Object Browser** (просмотр объектов)

Переменные

Переменные величины при выполнении программы могут свои значения изменять. Переменная имеет имя. Правила формирования имени переменной:

- имя начинается с буквы; остальные символы – буквы и цифры, можно использовать знак подчеркивания «_»;
- имя не должно содержать точки;
- длина имени не более 255 символов;
- имя должно быть уникальным в пределах данной процедуры;
- выбирать имя лучше так, чтобы оно несло информацию о назначении переменной;
- имя не должно быть *ключевым словом* Visual Basic.

Visual Basic не требует обязательного объявления переменной перед ее использованием. Если переменная не объявлена, Visual Basic использует тип данных, заданный по умолчанию - Variant. Этот тип может содержать любую информацию. Частое использование этого типа для хранения информации имеет два существенных недостатка – влечет лишнюю трату ресурсов памяти и может привести к непредсказуемому поведению заданных по умолчанию значений. Поэтому все же лучше объявлять переменные перед их использованием.

Объявление типа переменной означает команду установить границы значений переменной, которые определяются ее типом. Команда записывается с помощью программного кода. Существует несколько способов объявления типа переменной.

Раздел 2. Информация и информационные процессы.

1 способ. Объявление типа с помощью суффикса

К имени переменной приписывается один из следующих суффиксов:

Суффикс	%	&	!	#	\$	@
Тип	Integer	Long	Single	Double	String	Currency

2 способ. Объявление типа с помощью оператора описания типа

Dim *ИмяПеременной* **As** *ТипПеременной*

Dim, **As**- это ключевые слова Visual Basic.

При объявлении переменных следует учитывать следующее: переменная, объявленная в процедуре, инициализируется каждый раз, когда происходит вызов этой процедуры. Чтобы переменная сохраняла свое старое значение при повторном обращении к процедуре, при ее описании надо использовать ключевое слово **Static** (статическая):

Static *ИмяПеременной* **As** *ТипПеременной*

Область видимости переменных

Переменные могут быть объявлены в модулях, формах или процедурах.

- Переменная, объявленная в процедуре, доступна для использования только в пределах этой процедуры (*локальная*), даже если она была объявлена как **Static**;
- Переменная, объявленная в **General Declarations** (раздел общих объявлений) формы доступна в пределах всей формы.
- Область видимости переменных, объявленных на уровне модуля, зависит от ключевых слов, использованных при объявлении.
- Если переменная объявлена посредством **Dim**, область видимости – модуль.
- Если переменная объявлена посредством ключевого слова **Public**, область ее видимости – вся программа (переменная - *глобальная*).

Присвоение значения переменной

Для присвоения переменной некоторого значения используется *оператор присвоения*.

Общий вид оператора присвоения:

[Let] *ИмяПеременной* = *ЗначениеПеременной*

Ключевое слово **Let** в квадратных скобках может отсутствовать.

ЗначениеПеременной – это либо константа или выражение, значение которого должно быть вычислено. В том и другом случае это значение должно находиться в диапазоне возможных значений для данного типа переменной.

Выражения и функции

Арифметическое выражение

Арифметическое выражение – это последовательность чисел, констант, переменных, числовых функций, которые соединены между собой знаками арифметических действий (см. табл.).

Раздел 2. Информация и информационные процессы.

Операция	Описание операции
$A \wedge B$	Возведение A в степень B
$\neg A$	Перемена знака A
$A * B$	Умножение A на B
A / B	Деление A на B
$A \setminus B$	Целая часть от деления A на B
$A \bmod B$	Остаток от деления A на B
$A + B$	Сложение A с B
$A - B$	Вычитание B из A

Логическое выражение

Логическое выражение – это последовательность символьных, числовых констант и переменных, математических и строковых функций, которые соединены между собой знаками сравнения и знаками логических операций (см. табл.). Логическое выражение принимает только два значения: истина (True) и ложь (False).

Знак сравнения	Действие
=	Равно
<	Меньше
>	Больше
<=	Меньше или равно
>=	Больше или равно
<>	Не равно

Знак логической операции	Действие
And	Логическое «И»
Eqv	Эквивалентность
Imp	Импликация
Or	Логическое «ИЛИ»
Xor	Исключающее «ИЛИ»
Not	Логическое «НЕ»

Строковое выражение

Строковое выражение – это последовательность символьных констант, переменных, символьных функций и знака присоединения «+» (конкатенации)

Функция – это процедура, которая выполняет и возвращает значение. Общий вид обращения к функции:

ИмяФункции (СписокАргументовФункции)

ИмяФункции - это либо имя уже имеющейся в языке (встроенной) функции, либо имя функции, определяемой программистом.

АргументФункции – это либо константа, либо переменная, либо выражение.

Встроенные математические функции

Функция	Значение функции
$Abs (x)$	Абсолютная величина числа x
$Cint (x)$	Целое число, ближайшее к числу x
$Fix (x)$	Целое число, равное числу x без дробной части
$Int (x)$	Наибольшее целое число, не превышающее x
$Sqr (x)$	Квадратный корень из x

Раздел 2. Информация и информационные процессы.

<i>Sin</i> (x)	Синус
<i>Cos</i> (x)	Косинус
<i>Tan</i> (x)	Тангенс
<i>Atn</i> (x)	Арктангенс
<i>Rnd</i>	Псевдослучайное число от 0 до 1

Финансовые функции

Функция	Значение функции
<i>Pmt</i> (проц.ставка, число выплат, кредит)	Периодическая выплата
<i>PV</i> (проц.ставка, число выплат, выплата)	Кредит
<i>Rate</i> (число выплат, выплата, кредит)	Процентная ставка, отнесенная на период выплаты

Функции обработки строк

Функция	Значение функции
<i>InStr</i> (строка, подстрока)	Позиция подстроки в строке
<i>Lcase</i> (строка)	Строка со строчными буквами
<i>Left</i> (строка, длина)	Левая подстрока данной длины
<i>Len</i> (строка)	Длина строки
<i>Mid</i> (строка, позиция, длина)	Подстрока данной длины, начиная с данной позиции
<i>Right</i> (строка, длина)	Правая подстрока данной длины
<i>Trim</i> (строка)	Строка без первых и последних пробелов
<i>Ucase</i> (строка)	Строка с прописными буквами

Функции даты и времени

Функция	Значение функции
<i>Time</i>	Текущее системное время
<i>Timer</i>	Число прошедших с полуночи секунд
<i>Now</i>	Системная дата и время
<i>Date</i>	Системная дата
<i>Month</i> (дата)	Номер месяца
<i>Year</i> (дата)	Год
<i>Day</i>	День
<i>WeekDay</i> (дата)	Номер дня недели (воскресенье – 1 и т.д)

Системные функции

Функция	Значение функции
<i>InputBox</i> (Приглашение[, Заголовок] [, НачЗначение] [,X] [,Y]) Предназначена для ввода данных пользователем через системное окно X,Y – координаты левого верхнего угла окна	Введенный пользователем текст
<i>MsgBox</i> (Сообщение [, Опция, Заголовок]) Предназначена для вывода сообщения через системное окно	Целое число, определяющее какая из кнопок <i>Окна сообщения</i> была нажата

Раздел 2. Информация и информационные процессы.

<i>Опция</i> определяет набор кнопок и вид пиктограммы в окне сообщения		1 (VbOk) Ок
0 Ок		2 (VbCancel) Отмена
1 Ок, Отмена		3 (VbAbort) Стоп
2 Стоп, Повтор, Пропустить		4 (VbRetry) Повтор
3 Да, Нет, Отмена		5 (VbIgnore) Пропустить
4 Да, Нет		6 (VbYes) Да
5 Повтор, Отмена		7 (VbNo) Нет
16 критическое сообщение		
32 вопрос		
48 предупреждение		
64 информация		

Функции преобразования типов

Функция	Значение функции
<i>Asc</i> (строка)	Код ASCII первого символа строки
<i>Chr</i> (код ASCII)	Символ, соответствующий коду
<i>CInt</i> (строка цифр или число)	Целое число
<i>Val</i> (строка)	Число
<i>Str</i> (число)	Строка
<i>CDate</i> (выражение)	Выражение типа Date
<i>CBool</i> (выражение)	Выражение типа Boolean
<i>CByte</i> (выражение)	Выражение типа Byte
<i>CCur</i> (выражение)	Выражение типа Currency
<i>CDbl</i> (выражение)	Выражение типа Double
<i>CInt</i> (выражение)	Выражение типа Integer
<i>CLng</i> (выражение)	Выражение типа Long

Функции для работы с графикой

Функция	Значение функции
<i>LoadPicture</i> (строка-путь к графическому файлу)	Содержимое графического файла
<i>QBColor</i> (число от 0 до 15)	Цвет
<i>RGB</i> (N1, N2, N3) N1 – интенсивность красного цвета (от 0 до 255) N2 - интенсивность зеленого цвета (от 0 до 255) N3 - интенсивность синего цвета (от 0 до 255)	Цвет

2.3 Основные информационные процессы и их реализация с помощью компьютеров: хранение, поиск и передача информации: хранение, поиск и передача информации.

Информация, закодированная с помощью естественных и формальных языков, а также информация в форме зрительных и звуковых образов хранится в памяти человека. Однако для долговременного хранения информации, ее накопления и передачи из поколения в поколение используются носители информации.

Материальная природа носителей информации может быть различной:

Раздел 2. Информация и информационные процессы.

- а) молекулы ДНК, которые хранят генетическую информацию;
- б) бумага, на которой хранятся тексты и изображения;
- в) магнитная лента, на которой хранится звуковая информация;
- г) фото- и киноплёнки, на которых хранится графическая информация;
- д) микросхемы памяти, магнитные и лазерные диски, на которых хранятся программы и данные в компьютере, и так далее.

По оценкам специалистов, объем информации, фиксируемой на различных носителях, превышает один эксабайт в год. Примерно 80% всей этой информации хранится в цифровой форме на магнитных и оптических носителях и только 20% - на аналоговых носителях (бумага, магнитные ленты, фото- и киноплёнки).

Большое значение имеет надежность и долговременность хранения информации. Большую устойчивость к возможным повреждениям имеют молекулы ДНК, так как существует механизм обнаружения повреждений их структуры (мутаций) и самовосстановления. Надежность (устойчивость к повреждениям) достаточно высока у аналоговых носителей, повреждение которых приводит к потере информации только на поврежденном участке. Поврежденная часть фотографии не лишает возможности видеть оставшуюся часть, повреждение участка магнитной ленты приводит лишь к временному пропаданию звука и так далее. Цифровые носители гораздо более чувствительны к повреждениям, даже потеря одного бита данных на магнитном или оптическом диске может привести к невозможности считать файл, то есть к потере большого объема данных. Именно поэтому необходимо соблюдать правила эксплуатации и хранения цифровых носителей информации.

Наиболее долговременным носителем информации является молекула ДНК, которая в течение десятков тысяч лет (человек) и миллионов лет (некоторые живые организмы), сохраняет генетическую информацию данного вида.

Аналоговые носители способны сохранять информацию в течение тысяч лет (египетские папирусы и шумерские глиняные таблички), сотен лет (бумага) и десятков лет (магнитные ленты, фото- и киноплёнки). **Цифровые носители** появились сравнительно недавно и поэтому об их долговременности можно судить только по оценкам специалистов. По экспертным оценкам, при правильном хранении оптические носители способны хранить информацию сотни лет, а магнитные - десятки лет.

Носители информации характеризуются информационной емкостью, то есть количеством информации, которое они могут хранить. Наиболее информационно емкими являются молекулы ДНК, которые имеют очень малый размер и плотно упакованы. Это позволяет хранить огромное количество информации (до 10^{21} битов в 1 см^3), что дает возможность организму развиваться из одной-единственной клетки, содержащей всю необходимую генетическую информацию.

Современные микросхемы памяти позволяют хранить в 1 см^3 до 10^{10} битов информации, однако это в 100 миллиардов раз меньше, чем в ДНК. Можно сказать, что современные технологии пока существенно проигрывают биологической эволюции. Однако если сравнивать информационную емкость традиционных носителей информации (книг) и современных компьютерных носителей, то прогресс очевиден:

- а) Лист формата А4 с текстом (набран на компьютере шрифтом 12-го кегля с одинарным интервалом) - около 3500 символов

Раздел 2. Информация и информационные процессы.

- б) Страница учебника - 2000 символов
- в) Гибкий магнитный диск – 1,44 Мб
- г) Оптический диск CD-R(W) – 700 Мб
- д) Оптический диск DVD – 4,2 Гб
- е) Флэш-накопитель - несколько Гб
- ж) Жесткий магнитный диск – сотни Гб

Таким образом, на дискете может храниться 2-3 книги, а на жестком магнитном диске или DVD - целая библиотека, включающая десятки тысяч книг.

Достоинство внутренней памяти – быстрота воспроизведения информации, а **недостаток**- со временем часть информации забывается. Достоинство внешней памяти- большие объемы информации хранятся долго, а недостаток- для доступа к определенной информации требуется время (например, чтобы подготовить реферат по предмету необходимо найти, проанализировать и выбрать подходящий материал).

Созданную или полученную каким-либо образом информацию хранят в течение определённого времени, в течение которого её временно или долговременно содержат на различных носителях электронных данных. Если информация представляет интерес для её создателей или правообладателей, то им приходится создавать электронные архивы.

Архив информации. *Электронный архив* - это файл, содержащий один или несколько файлов в сжатой или несжатой форме и информацию, связанную с этими файлами (имя файла, дата и время последней редакции и т.п.). Электронные архивы позволяют в любой момент времени извлекать из них необходимые данные для дальнейшего их использования в различных ситуациях (например, для обновления или восстановления утраченных данных). Такие архивы называют страховочными копиями. Их используют в случае утраты или порчи основной машиночитаемой информации, а также для длительного её хранения в месте, которое защищено от вредных воздействий и несанкционированного доступа.

Как правило, компьютерными архивами информации являются электронные каталоги, базы и банки данных, а также коллекции любых видов электронной информации. Для обеспечения надёжности хранения и защиты данных рекомендуют создавать по 2–3 архивные копии последних редакций файлов. В случае необходимости осуществляется разархивирование данных.

Разархивирование - это процесс точного восстановления электронной информации, ранее сжатой и хранящейся в файле-архиве.

Для создания архивных файлов и разархивирования используют специальные программы-архиваторы: WinRAR, 7-Zip File Manager.

Основные возможности архиваторов:

- а) просмотр содержания архива и файлов, содержащихся в архиве;
- б) распаковка архива или отдельных файлов архива;
- в) создание простого архива файлов (файлов и папок) в виде файла с расширением, определяющим используемую программу-архиватор;
- г) создание самораспаковывающегося архива файлов (файлов и папок) в виде файла с пусковым расширением EXE;

Раздел 2. Информация и информационные процессы.

- д) создание многотомного архива файлов (файлов и папок) в виде группы файлов- томов заданного размера (раньше - в размер дискеты).

Вопросы и задания:

1. Сколько байт в 2,6 Гбайт?
2. Что больше, 36 Кбайт или 0,000037 Гбайт? Ответ обоснуйте.
3. Какие достоинства и недостатки имеют аналоговые и цифровые носители информации?
4. Чем отличаются позиционные системы счисления от непозиционных?
5. Какие параметры участвуют в кодировании звуковой информации?
6. Какие логические операции вы знаете? Как они обозначаются?
7. Соедините правильные определения или обозначения:

1. Логика	1. $A \rightarrow B$
2. Высказывание	2. Логическое сложение
3. Алгебра логики	3. Наука о формах и способах мышления
4. Логическая константа	4. Логическое отрицание
5. Дизъюнкция	5. ИСТИНА и ЛОЖЬ
6. Инверсия	6. $A \leftrightarrow B$
7. Конъюнкция	7. &
8. Импликация	8. Наука об операциях над высказываниями
9. Эквивалентность	9. Повествовательное предложение, в котором что-либо утверждается или отрицается.

8. Перечислите типовые алгоритмические конструкции и объясните их назначение.
9. Каковы характерные особенности алгоритмических языков программирования? Объектно-ориентированных языков программирования?
10. Ответьте на вопросы по Visual Basic:
 - а) Какими способами можно запустить VB?
 - б) Для чего используется окно Проводника проектов?
 - в) Как открыть окно Свойств?
 - г) Как сохранить проект?
 - д) Как переводятся на русский язык слова Label, Caption, Command, Form.
 - е) С помощью какой команды главного меню можно установить Панель инструментов?
 - ж) Что такое форма?
 - з) Что означает свойство Caption у формы?
 - и) Что означает свойство BackColor у формы?
 - к) Что означает свойство ForeColor у формы?
 - л) Как запустить программу на выполнение?
 - м) Как создать элемент управления на форме?
 - н) Как удалить элемент управления?
 - о) Для чего используется элемент управления TextBox?
 - п) Что означает свойство Font?
 - р) Что означает свойство Alignment?

Проектные задания:

1. Напишите реферат «Двоичное кодирование и компьютер». Рассмотрите вопрос о видах позиционных систем счисления – почему именно позиционные системы нашли широкое применение в компьютерной техники? Опишите, почему именно двоичная система счисления

Раздел 2. Информация и информационные процессы.

нашла широкое применение в компьютерной технике. Обоснуйте пары понятий бит и триггер, байт и регистр. (для технического профиля).

2. Подготовьте реферат по истории развития языков программирования «Языки программирования: время, открытия, люди». (для всех профессий)

Список использованной литературы и интернет - ресурсов:

1. Информатика и ИКТ: учебник для нач. и сред.проф.образования / М.С.Цветкова, Л.С.Великович.- М.: Издательский центр «Академия», 2013.
2. <http://informatika.sch880.ru/p18aa1.html> - измерение информации.
3. http://life-prog.ru/1_28809_informatsionnie-ob-ekti-razlichnih-vidov.html - информационные объекты.
4. <http://www.ido.rudn.ru/nfpk/inf/inf1.html> - информатика, информационные объекты разных видов.
5. http://informatika-spo.org.ru/doc/kurs1/new_program/konspekt_lection_3.pdf - универсальность дискретного (цифрового) представления информации.
6. <http://www.studfiles.ru/preview/3600795/> - принципы обработки информации компьютером, арифметические и логические основы работы компьютера.
7. <http://www.studfiles.ru/preview/2535719/> - арифметические и логические основы работы компьютера.
8. <https://sites.google.com/site/okotsitomsk/informatika/2-2-osnovnye-informacionnye-processy-i-ih-realizacia-s-pomосу-komputerov> - основные информационные процессы и их реализация с помощью компьютеров.
9. http://esate.ru/article/cg/fraktalnaya_grafika/ - фрактальная графика.
10. <http://smartappliance.ru/graphic-and-video/fractal.php> - фрактальная графика.
11. <http://www.studfiles.ru/preview/5627319/> - кодирование звуковой информации.
12. <http://www.5byte.ru/9/0009.php> - кодирование и обработка звуковой информации.
13. http://studopedia.su/6_11212_kodirovanie-videoinformatsii.html - кодирование звуковой информации.