Планирование урока информатики в 11 классе

Информатика и информация.

Информация – это сведения об окружающем мире и процессах, в нем происходящих.

Кодирование и измерение информации.

Информация – это сведения об окружающем мире и процессах, в нем происходящих.

Информатика – это наука методах и способах сбора, хранения, обработки передачи информации с помощью средств ВТ.

Единицы измерения информации:

1 бит=1разряд(0 или1);

1 байт=8 разрядов;

1 Кбайт=1024 байта

1 Мбайт=1024 Кбайта

1 Гбайт=1024 Мбайта

1 Тбайт=1024 Гбайта

Кодирование – это представление символов одного алфавита символами другого.

Для кодирования, используется двоичная система счисления, в которой только два символа: 0 и 1. Причем,1- означает наличие сигнала, 0-его отсутствие. Один двоичный сигнал получил название – бит.

Измерение информации. Содержательный и алфавитный подход.

Пример 2.5. Какова мощность алфавита, с помощью которого записано сообщение, содержащее 2048 символов, если его объем составляет 1,25 Кбайта.

С помощью программы VersaVerter или арифметически перевести информационный объем сообщения в биты:

I = 10 240 бит

Определить количество бит, приходящееся на один символ:

10 240 бит : 2 048 = 5 бит

По формуле определить количество символов в алфавите:

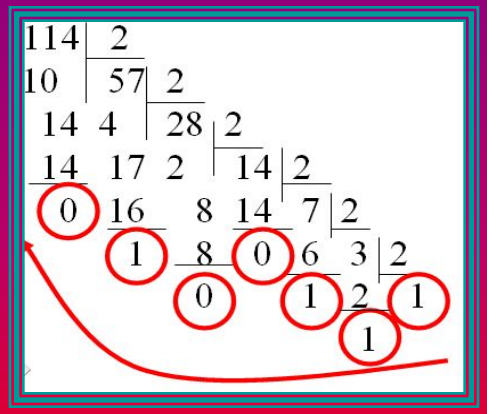
$N = 2^i = 2^5 = 32$

Системы счисления. Перевод чисел.

Непозиционные системы счисления – это система, в которой вводится ряд символов для представления основных чисел, а остальные числа – результат их сложения и вычитания (римская система счисления).

Позиционная система счисления – это такая система, в которой любое число представляется в виде последовательности цифр, количественное значение которых зависит от их места (позиции) в числе (десятичная система счисления).

Позиционная система счисления – это такая система, в которой любое число представляется в виде последовательности цифр, количественное значение которых зависит от их места (позиции) в числе (десятичная система счисления).



Десятичная	Двоичная	Десятичная	Двоичная	Десятичная	Двоичная
0	0	5	101	10	1001
1	1	6	110	11	1011
2	10	7	111	12	1100
3	11	8	1000	13	1101
4	100	9	1001	14	1110

$$1110010=1\times2^6+1\times2^5+1\times2^4+0\times2^3+0\times2^2+1\times2^1+0\times2^0=\quad 114$$

Арифметические операции.

Сложение	Вычитание	Умножение
0 + 0 = 0	0 – 0 = 0	0 * 0 = 0
0 + 1 = 1	1 0 – 0 = 1	1 * 0 = 0
1 + 0 = 0	1 – 0 = 0	0 * 1 = 0
1 + 1 = 10	1 – 1 = 0	1 * 1 = 1

Сложение	Вычитание	Умножение
0 + 0 = 0	0 – 0 = 0	0 * 0 = 0
0 + 1 = 1	1 0 – 0 = 1	1 * 0 = 0
1 + 0 = 0	1 – 0 = 0	0 * 1 = 0
1 + 1 = 10	1 – 1 = 0	1 * 1 = 1

$$\begin{array}{r} 110_2 \\ +\ 11_2 \\ \hline 1001_2 \end{array}$$

$$\begin{array}{r} 110_2 \\ \times\ 11_2 \\ \hline 110 \\ 110 \\ \hline 10010_2 \end{array}$$

$$\begin{array}{r|l} 110_2 & 11_2 \\ \hline 11 & 10_2 \\ \hline 0 & \end{array}$$

Контрольная работа: "Системы счисления".

1. Перевести из десятичной системы счисления в двоичную, восьмеричную, шестнадцатеричную целое число :

368

2. Перевести из десятичной системы счисления в двоичную, восьмеричную, шестнадцатеричную дробное число :

0.373

3. Перевести число шестнадцатеричное в двоичную систему счисления: BFC

4. Перевести двоичное число в шестнадцатеричную систему счисления:

10111110101, 110011

5. Решить выражение и записать ответ в двоичной системе счисления:

270(8) +9A (16) - 111000,011(2)

Представление числовой информации. Фиксированная точка.

Задача.

Выполнить действия над машинными кодами чисел:

С фиксированной точкой. Формат 16 двоичных разрядов.

Дано: A=190; B=250

Найти: C1=A + B; C2=A – B.

Решение:

A(10) = 190; A(16)=BE=10111110(2)

B(10) = 250; B(16)=FA=11111010(2)

C1 = A+B

A= 0 000000010111110

+B= 0 000000011111010

C1= 0 000000110111000

C2 = A – B

A = 0 0000000010111110 (прямой код)

- B = 1 111111100000110 (дополнительный)

C2 = 1 111111111000100

Проверка:

C1=110111000(2)

C1(16) = 1B8 = 1*16*16+11*16+8*1 = 440(10)

Проверка:

C2 = - 111100 = - BC= - 3*16 +12*1 = - 60 (10)

Ответ:

C1 = 0 000000110111000

C2 = 1 000000000111100

Представление числовой информации. Плавающая точка.

Дано:

A = 12,75; B = 250

Найти: C3 = A + B, C4 = A – B

Формат – 32 двоичных разряда со смещенным порядком.

A(10) = 12,75 = A(16) = C.C;

B(10) = 250 = B(16) = FA

Нормализация мантисс

mA = 0.CC; pxA = 40 + 1 = 41

mB = 0.FA; pxB = 40 + 2 = 42

Выравнивание характеристик:

Δp = pxA – pxB = -1

$m^*A = mA * 16^{-1} = 0.0CC$; pxA = 41+ 1 = 42

C3 = A + B;

mA = 00 0CC000 pxA = 42

mB = 00 FA0000 pxB = 42

mC3 = 01 06C000 pxC = 42

Нормализация мантиссы результата

mxC3 = 00 106C00;

pxC3 = 42 + 1 = 43

Проверка

C3(16) = 106,C = (C3) = 262,75

C3 = 0 10000110001000001101100000000000

C4 = A – B

mA = 00 0CC000 pxA = 42

mB = 10 06000 pxB = 42

mC3 = 10 12C000 pxC = 42

Нормализация мантиссы результата:

mC4 = 10 ED4000 pxC4 = 42

Проверка:

C4 = - ED.4 = (C4) = - (14 * 16 + 13 * 1 + 4/16) = - 237, 25

C4 = 1 10000101110110101000000000000000

Представление графической и звуковой информации.

Представление графической и звуковой информации.

Кодирование графической информации.

Задача.

Определить объем видеопамяти компьютера, который необходим для реализации графического режима монитора High Color с разрешающей способностью 1024x768 точек и палитрой из 65536 цветов.

Глубина цвета составляет:

I = log(2)65536 = 16 бит.

Количество точек изображения равно: 1024x768 = 786432

Требуемый объем видеопамяти равен:

10 бит x786432 = 12582912 бит = 1,2 Мбайта

Кодирование звуковой информации

Задача

Оцените информационный объем высококачественного стереоаудиофайла длительностью звучания 1 минута, если «глубина» кодирования 16 бит, а частота дискретизации 48 кГц.

Информационный объем звукового файла длительностью в 1 секунду равно:

16 бит * 48000 * 2 = 1536000 бит = 187,5 Кбайт.

Информационный объем звукового файла длительностью в 1 минуту равен:

187,5Кбайт/с * 60 с + 11 Мбайт

Контрольная работа по теме: "Представление информации в компьютере".

B(10) = - 128; B(16)=- 80(16)= - 10000000(2)

A(2) = 1 000000110000011 –прямой код (-A) = 1 111111001111101

A(2) = 1 111111001111100 –обратный код (-B) = 1 111111110000000

A92) = 1 111111001111101 – дополн. код X = 1 111110111111101 –доп. код

Задача.

Выполнить действия над машинными кодами чисел:

С фиксированной точкой. Формат 16 двоичных разрядов.

Дано: A= - 387; B= - 128

Найти: C1=A + B;

Решение:

X = A+B

X = (-A) + (- B)

A(10) = - 387; A(16)=- 183(16)= - 110000011(2)

B(10) = - 128; B(16)=- 80(16)= - 10000000(2)

A(2) = 1 000000110000011 –прямой код (-A) = 1 111111001111101

A(2) = 1 111111001111100 –обратный код (-B) = 1 111111110000000

A92) = 1 111111001111101 – дополн. код X = 1 111110111111101 –доп. код

B(2) = 1 000000010000000 –прямой код X = 1 000001000000010 – обр.код

B(2) = 1 111111101111111 –обратный код X = 1 000001000000011 – пр.код

B(2) = 1 111111110000000 – дополн. Код X = - 203(16) = - (2*16*16+0*16+3*1) =

= - (256*2+3) = - (512+3)+ - 515

B(2) = 1 000000010000000 –прямой код X = 1 000001000000010 – обр.код

B(2) = 1 111111101111111 –обратный код X = 1 000001000000011 – пр.код

B(2) = 1 111111110000000 – дополн. Код X = - 203(16) = - (2*16*16+0*16+3*1) =

= - (256*2+3) = - (512+3)+ - 515

Пример. Для формулы $A\&(B \vee \quad \& \quad)$ построить таблицу истинности алгебраически и с использованием электронных таблиц.

Количество логических переменных 3, следовательно, количество строк в таблице истинности должно быть $2^3 = 8$.

Количество логических операций в формуле 5, следовательно количество столбцов в таблице истинности должно быть 3 + 5 = 8.

Алгебра логики. Логические выражения. Таблицы истинности.

Пример. Для формулы $A\&(B \vee \quad \& \quad)$ построить таблицу истинности алгебраически и с использованием электронных таблиц.

Количество логических переменных 3, следовательно, количество строк в таблице истинности должно быть $2^3 = 8$.

Количество логических операций в формуле 5, следовательно количество столбцов в таблице истинности должно быть 3 + 5 = 8.

A	B	C	$\overline{B}$	$\overline{C}$	$\overline{B}$ $\overline{C}$	$B \vee (\overline{B} \ \& \ \overline{C})$	$A\&(B \vee \overline{B} \ \& \ \overline{C})$
0	0	0	1	1	1	1	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	1	0
0	1	1	0	0	0	1	0
1	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0
1	1	0	0	1	0	1	1
1	1	1	0	0	0	1	1

Логические функции

Логической (булевой) ***функцией*** называют функцию ***F(X1, X2, ..., Xn)***, аргументы которой ***X1, X2, ..., Xn*** (независимые переменные) и сама функция (зависимая переменная) принимают значения 0 или 1.

Логической (булевой) ***функцией*** называют функцию ***F(X1, X2, ..., Xn)***, аргументы которой ***X1, X2, ..., Xn*** (независимые переменные) и сама функция (зависимая переменная) принимают значения 0 или 1.

Существует 16 различных логических функций от двух переменных.

Логические функции двух переменных

Аргументы		Логические функции															
A	*B*	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Логические законы

1. Закон двойного отрицания:

$$A = \overline{\overline{A}}.$$

Двойное отрицание исключает отрицание.

2. Переместительный (коммутативный) закон:

— для логического сложения:

$$A \vee B = B \vee A;$$

— для логического умножения:

$$A\&B = B\&A.$$

Результат операции над высказываниями не зависит от того, в каком порядке берутся эти высказывания.

В обычной алгебре $a + b = b + a$, $a \times b = b \times a$.

3. Сочетательный (ассоциативный) закон:

— для логического сложения:

$$(A \vee B) \vee C = A \vee (B \vee C);$$

— для логического умножения:

$$(A\&B)\&C = A\&(B\&C).$$

При одинаковых знаках скобки можно ставить произвольно или вообще опускать.

В обычной алгебре:

$$(a + b) + c = a + (b + c) = a + b + c,$$

$$a \times (b \times c) = a \times (b \times c) = a \times b \times c.$$

4. Распределительный (дистрибутивный) закон:

— для логического сложения:

$$(A \vee B)\&C = (A\&C) \vee (B\&C);$$

— для логического умножения:

$$(A\&B) \vee C = (A \vee C)\&(B \vee C).$$

Определяет правило выноса общего высказывания за скобку.

В обычной алгебре:

$$(a + b) \times c = a \times c + b \times c.$$

5. Закон общей инверсии (законы де Моргана):

— для логического сложения

$$\overline{A \vee B} = \overline{A}\ \&\ \overline{B};$$

— для логического умножения:

$$\overline{A\&B} = \overline{A} \vee \overline{B}$$

6. Закон идемпотентности (от латинских слов idem — тот же самый и potens —сильный; дословно — равносильный):

— для логического сложения:

$$A \vee A = A;$$

— для логического умножения:

$$A\&A = A.$$

Логические законы

7. *Законы исключения констант:*

— для логического сложения:

$$A \vee 1 = 1, \quad A \vee 0 = A;$$

— для логического умножения:

$$A \& 1 = A, \quad A \& 0 = 0.$$

8. *Закон противоречия:*

$$A \& \overline{A} = 0.$$

Невозможно, чтобы противоречащие высказывания были одновременно истинными.

9. *Закон исключения третьего*:

$$A \vee \overline{A} = 1.$$

Из двух противоречащих высказываний об одном и том же предмете одно всегда истинно, а второе — ложно, третьего не дано.

10. *Закон поглощения*:

— для логического сложения:

$$A \vee (A \& B) = A;$$

— для логического умножения:

$$A \& (A \vee B) = A.$$

11. *Закон исключения (склеивания)*:

— для логического сложения:

$$(A \& B) \vee (\overline{A} \& B) = B;$$

12. *Закон контрапозиции (правило перевертывания):*

$$(A \Leftrightarrow B) = (B \Leftrightarrow A).$$

Справедливость приведенных законов можно доказать табличным способом: выписать все наборы значений A и B, вычислить на них значения левой и правой частей доказываемого выражения и убедиться, что результирующие столбцы совпадут.

Пример 3.11. Найдите X, если $\overline{X \vee A} \vee \overline{X \vee \overline{A}} = B$.

Для преобразования левой части равенства последовательно воспользуемся законом де Моргана для логического сложения и законом двойного отрицания:

$$(\overline{X} \& \overline{A}) \vee (\overline{X} \& A)$$

Согласно распределительному закону для логического сложения:

$$\overline{X} \& (\overline{A} \vee A)$$

Согласно закону исключения третьего и закона исключения констант:

$$\overline{X} \& 1 = \overline{X}$$

Полученную левую часть приравняем правой:

$$\overline{X} = B$$

Окончательно получим, что $X = \overline{B}$.

Принцип работы логических устройств.

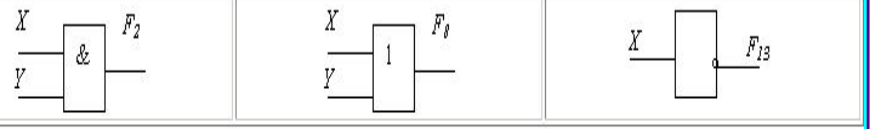
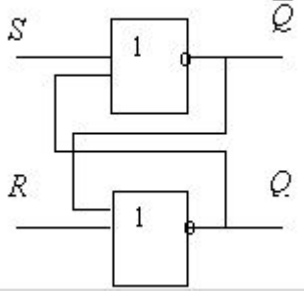
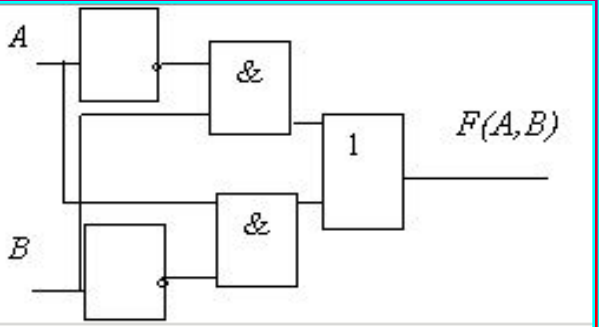


Рис. 3.1. Коньюнктор, дизьюнктор и инвертор

2. Логическая схема RS-триггера



Пример 3.13. По заданной логической функции $F(A, B) = B\& \overline{A} \vee \overline{B}\ \&A$ построить логическую схему.



Триггер. Для хранения информации в оперативной памяти компьютера, а также во внутренних регистрах процессора используются триггеры. Триггер может находиться а одном из двух устойчивых состояний, что позволяет запоминать, хранить и считывать 1 бит информации.

Самый простой триггер — RS-триггер. Он состоит из двух элементов ИЛИ-НЕ (смотри задание 3.34), входы и выходы которых соединены кольцом: выход первого соединен со входом второго и выход второго – со входом первого. Триггер имеет два входа S (от англ. set – установка) и R (от англ. reset – сброс) и два выхода Q (прямой) и $\overline{Q}$ (инверсный).

Контрольная работа: "Алгебра логики".

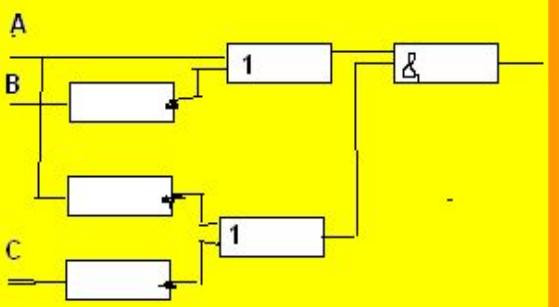
1. **Упростить выражение:**

$$\overline{\overline{A \wedge B} \vee (C \wedge B)}$$

2. **Постройте таблицу истинности для логической функции:**

$$F = (\overline{A} + B) * C \rightarrow (A * B)$$

3. **Дана логическая схема:**



Напишите по ней логическую функцию и составьте таблицу истинности.

Алгоритм. Алгоритмические конструкции. Свойства алгоритмов.

Алгоритм: понятное и точное предписание исполнителю выполнить порядок действий, направленных на решение конкретной задачи или достижение поставленной цели

Алгоритм состоит из законченных действий, называемых командами

Команды выполняются одна за другой

Исполнитель:

живое существо или технический объект, выполняющий команды алгоритма

Свойства алгоритмов

Дискретность – каждая команда должнабыть выполнена прежде, чем исполнитель перейдет к выполнению следующей

Понятность – каждая команда должна в ходить в СКИ

Точность (определенность) – команда должна пониматься исполнителем однозначно

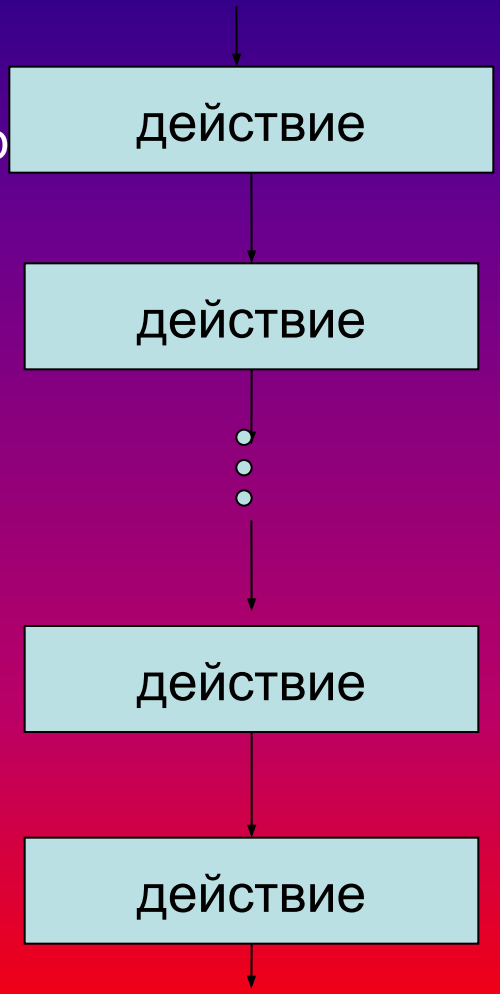
Результативность – выполнение всех команд алгоритма должно привести к решению конкретной задачи за конечное число шагов

Массовость – по одному и тому же алгоритму можно решать однотипные задачи

Основные алгоритмические конструкции

Следование (линейный тип алгоритма):

Все команды алгоритма следуют последовательно др за другом.



Вычислить площадь треугольника по формуле Герона.

```
program ABS (input, output);
var  s, a, b, c, p: real;
begin
writeln (‘a, b, c’);
readln (a, b, c);
p:=(a+b+c)/2;
s:=sqrt (p*(p-a)*(p-b)*(p-c));
writeln;
writeln (‘s=‘, s);
readln
end.
```

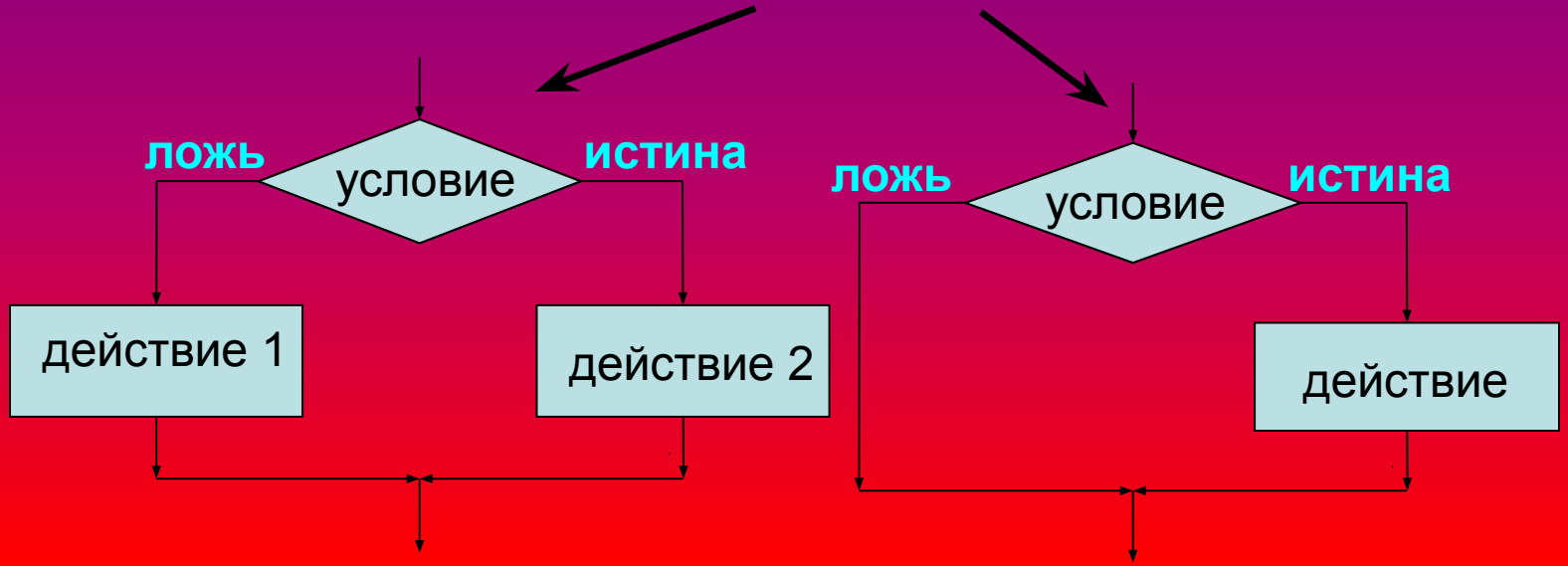
Основные алгоритмические конструкции

Ветвление (условный тип алгоритма):

Выбор действия зависит от выполнения некоторого условия.

Условие – выражение, которое может принимать значение либо истина, либо ложь.

Ветвления бывают полные и неполные



Примеры программ для конструкции ветвления

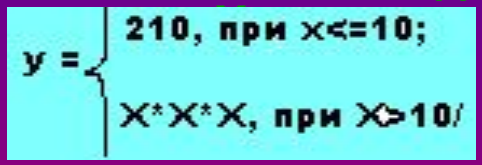
Краткая форма оператора

Выбрать наименьшее значение из трех различных чисел.

```
program ABS (input, output);
var   a, b, c, min: real;
begin
writeln (‘a, b, c’);
readln (a, b, c);
min:=a;
If b<min then min:=b;
 If c<min then min:=c;
writeln (‘наименьшее число‘, min );
readln
end.
```

Полная форма оператора

Составить программу для вычисления Y по заданному




```
program ABS (input, output);
var  x,y: real;
begin
writeln (‘введите x=’);
readln (x);
If x<10 then y:=210  else y:=x*x*x;
writeln (‘при x=‘,x, ‘y=‘, y);
readln
end.
```

Основные алгоритмические конструкции

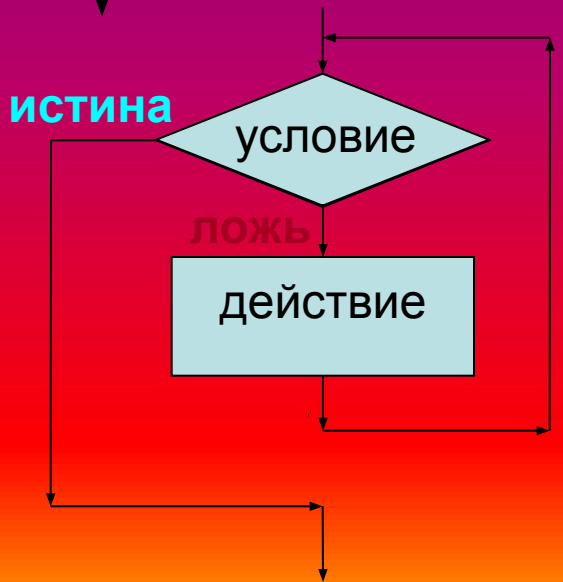
Основные алгоритмические конструкции

Повторение (циклический тип алгоритма)

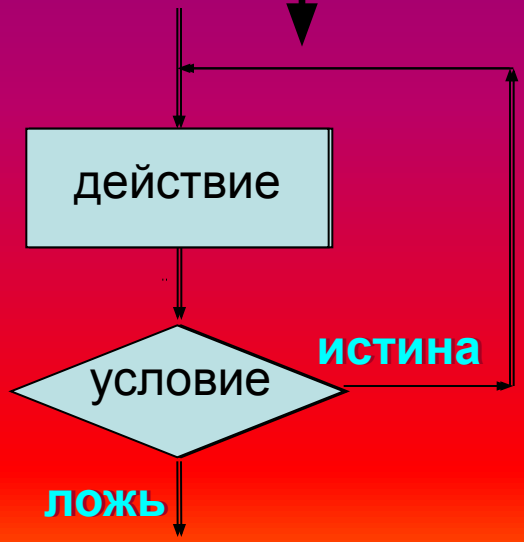
В алгоритме есть повторяющиеся действия.

Циклы бывают

с **предусловием**
(условие стоит перед повторением действий)



с **постусловием**
(условие стоит после повторения действий)



Конструкция повторение.

Операторы цикла.

Оператор цикла с постусловием

```
program zadacha (input, output);
var  a, b, c, n, i : integer;
begin
writeln (‘Введите N=’);
readln (n);
a:=1;
b:=1;
writeln (a,b);
i:=3;
repeat
c:=a+b;
writeln (c);
a:=b;
b:=c;
i:=i+1;
until i >n;
end.
```

Оператор цикла с постусловием

```
program zadacha (input, output);
var  a, b, c, n, i : integer;
begin
writeln (‘Введите N=’);
readln (n);
a:=1;
b:=1;
writeln (a,b);
i:=3;
repeat
c:=a+b;
writeln (c);
a:=b;
b:=c;
i:=i+1;
until i >n;
end.
```

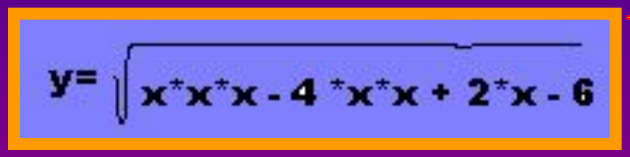
Оператор цикла с предусловием

```
program zadacha (input, output);
var  s,  i : integer;
begin
S:=0;
i:=101;
while  i<=301;
 begin
s:=s+1;
i:=i+2;
end;
writeln (‘сумма кубов нечетных
чисел от 101 до 301=‘, s);
readln;
end.
```

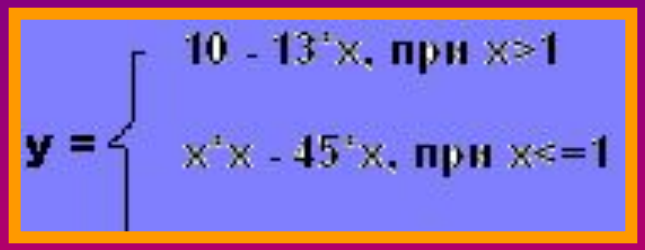
Контрольная работа

по теме:

"Алгоритмические конструкции".

1. Вычислить: 

2. Написать программу, вычисляющую значение Y:



3. Написать программу, вычисляющую значение S:

S = 1+2 +3 + …+ N !

N = 5!

Символьные выражения

Работа со строками.

Тип данных STRING (Строка)

Переменная типа STRING состоит из цепочки символов, т.е. элементов типа CHAR.

Описание строковых переменных

<имя переменной> : STRING [<количество символов>];

где <имя переменной> - идентификатор переменной

STRING – тип переменной (строковый)

[<количество символов>] – количество символов в строке.

Стандартные процедуры и функции для строк

LENGTH [<имя переменной>] – определяет фактическую длину текстовой строки, хранящейся в указанной переменной, включая все пробелы.

Пример:

```
var
word:string;
begin
writeln( (‘введите слово – ‘);
readln (word);
writeln(‘ слово состоит из  ‘ length (word), ‘ букв’);
end.
```

- С использованием процедур ввода ***Read, Readln.***

Одномерный массив

- С использованием процедур ввода ***Read, Readln.***

```
program zapoln2;
const n=5;
var i:integer;
a:array[1..100]of integer;
begin

      for i:=1 to n do begin
                       write('Vvedite a[',i,'] element ');readln(a[i]);
                       end;
      for i:=1 to n do begin
                        write (a[i]:3)
                        end;
      writeln;
      for i:=1 to n do begin
                       write('a[',i,']=');
                       writeln (a[i]:3)
                       end;

end.
```

Двумерный массив

Двумерный массив

блок заполнения массива с клавиатуры

```
for i:=l to N do
for j:=l to N do begin
write ('A[‘,i,’,’,j,’]=’);
readln (a[i,j]);
writeln
end;
```

блок заполнения массива с клавиатуры

```
for i:=l to N do
for j:=l to N do begin
write ('A[‘,i,’,’,j,’]=’);
readln (a[i,j]);
writeln
end;
```

блок заполнения массива случайным образом

```
randomize;
for i:=l to N do
for j:=l to N do A[i,j]:= random(50);
```

блок заполнения массива по правилу

```
for i:=l to N do
for j:=l to N do A[i,j]:= <правило>;
```

Рассмотрим следующую задачу: сформировать матрицу вида (рис 3.4.):

```
program zadacha;
var a: array [1..5, 1..5] of integer;
i,j,n: integer;
begin
n=5;
```

Контрольная работа по теме: "Массивы"

Нахождение суммы

- **Дан массив А(10). Найти сумму значений элементов массива, больших некоторой величины t**

```
Program zadacha (input, output);
Var
a: array [1…10] of real;
i: integer;
t, s: real;
begin
writeln (‘ввести значение переменной t’);
readln (t);
s:=0;
for i:=1  to  10 do begin
writeln (‘вв.знач.’, i, ‘эл-та’);
Readln (a [i]);
If a[i] >t then s:s+a[i]
End;
Writeln (‘сумма элементов =‘,s);
end.
```

2/ Дан массив А(10), заполненный датчиком случайных чисел.

```
program zadacha (input, output);
var
a: array (1..30) of integer;
t , min, max: integer;
begin
randomize;
for I :=1  to 30 do begin
a[i] :=random (50);
writeln (‘значение’ , i ,’эл.массива =‘, a [i] );
end;
min:=a(1);
max:=a(1);
for i:=1 to   30 do begin
If a (i) < min then min :=a [i];
If a(1) >max then max: =a [i];
end;
writeln( ‘max = ‘, max , ‘ min = ‘, min);
end/.
```

Моделирование физических процессов

Информационное моделирование – это творческий процесс. Не существует универсального рецепта построения моделей, пригодного на все случаи жизни, но можно выделить основные этапы и закономерности, характерные для создания самых разных моделей.

Первый этап – постановка задачи. Прежде всего следует уяснить цель моделирования. Исходя из цели моделирования, определяется вид и форма представления информационной модели, а также степень детализации и формализации модели. В соответствии с целью моделирования заранее определяются границы применимости создаваемой модели. На этом этапе также необходимо выбрать инструментарий, который будет использоваться при моделировании (например, компьютерную программу).

Моделирование физических процессов

Следующий этап – собственно моделирование, построение модели. На этом этапе важно правильно выявить составляющие систему объекты, их свойства и взаимоотношения и представить всю эту информацию в уже выбранной форме. Создаваемую модель необходимо периодически подвергать критическому анализу, чтобы своевременно выявлять избыточность, противоречивость и несоответствие целям моделирования.

Третий этап – оценка качества модели, заключающаяся в проверке соответствия модели целям моделирования. Такая проверка может производиться путем логических рассуждений, а также экспериментов, в том числе и компьютерных. При этом могут быть уточнены границы применимости модели. В случае выявления несоответствия модели целям моделирования она подлежит частичной или полной переделке.

Четвертый, и последний этап – эксплуатация модели, ее применение для решения практических задач в соответствии с целями моделирования.

Программирование на языке Visual Basic

- Проект 5.11. «Логический калькулятор». В языке Visual Basic основные логические операции могут быть реализованы с помощью логических операторов And (логическое умножение), Or (логическое сложение), Not (логическое отрицание), Xor (исключающее Or, которое принимает логическое значение True, тогда и только тогда, когда лишь один из аргументов имеет значение True) и Eqv (операция эквалентности, которая принимает логическое значение True, когда оба аргумента имеют значения True или False).
- Логические операторы могут оперировать с логическими аргументами True (логическая единица) и False (логический нуль), а также с логическими переменными типа Boolean.
- Разработаем проект «Логический калькулятор», который позволит определять истинность логических выражений.

Проект «Логический калькулятор».

1. Разместить на форме три текстовых поля, два `txtA` и `txtB` (для ввода логических аргументов) и `txtC` (для вывода значения логических функций).
2. Разместить на форме пять кнопок `cmdAnd`, `cmdOr`, `cmdNot`, `cmdXor` и `cmdEqv` для реализации соответствующих событийных процедур.
3. Программный код событийной процедуры `cmdAnd_Click()`, реализующий операцию логического сложения будет следующий:

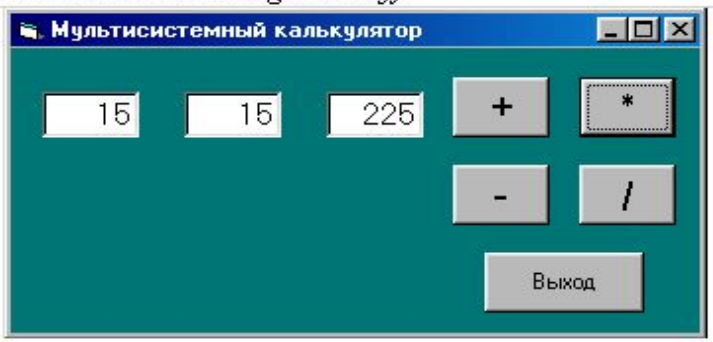
```
Private Sub cmdAnd_Click()
blnA = txtA.Text
blnB = txtB.Text
blnC = blnA And blnB
txtC.Text = blnC
End Sub
```

4. Программные коды событийных процедур, реализующих другие логические функции, записываются аналогично.
5. Запустить проект. Ввести в текстовые поля значения логических аргументов и щелкнуть по любой кнопке логической операции. В третьем окне появится результат.

Программирование на языке Visual Basic

Проект «Мультисистемный калькулятор».

1. Создать новый проект. Разместить на форме три текстовых поля и пять кнопок. Присвоить им имена: `txt1Dec, txt2Dec, txt3Dec, cmdPlus, cmdMinus, cmdUmn, cmdDelen, cmdExit`.
2. Для каждой из кнопок ввести программные коды событийных процедур.
3. Установить для свойства `Alignment` текстовых полей значение *Right Justufy*.
4. Запустить проект на выполнение. Ввести числа в два первых текстовых поля. Щелкнуть по любой кнопке арифметических операций и соответствующая она будет выполнена.



5. Добавить на форму три текстовых поля `txt1Oct, txt2Oct` и `txt3Oct`.
6. Поместить на форму кнопку `cmdDecOct` и ввести программный код событийной процедуры `cmdDecOct_Click()`.
7. Поместить на форму две метки с надписями *Десятичная система* и *Восьмеричная система*.
8. Запустить проект на выполнение. Ввести числа в два первых текстовых поля десятичной системы. Щелкнуть по кнопке «+», а затем *Dec-Oct*. Будет выполнено сложение десятичных чисел и их перевод в восьмеричную систему.

Программирование на языке Visual Basic

Перевод целых положительных чисел из восьмеричной системы счисления в десятичную с помощью функции Val. Для перевода чисел из восьмеричной системы счисления в десятичную воспользуемся функцией Val, аргументом которой являются восьмеричные числа в строковой форме, а значением десятичные числа.

Введенные в восьмеричные поля в строковой форме числа переведем сначала в восьмеричное представление добавлением приставки "&O" с помощью операции конкатенации строк. Полученное восьмеричное число в строковой форме переведем в десятичную числовую с помощью функции Val. Создадим кнопку cmdOctDec и запишем для нее событийную процедуру перевода чисел:

```
Private Sub cmdOctDec_Click()
txt1Dec.Text = Val("&O" + txt1Oct.Text)
txt2Dec.Text = Val("&O" + txt2Oct.Text)
txt3Dec.Text = Val("&O" + txt3Oct.Text)
End Sub
```

9	Поместить на форму кнопку cmdOctDec и ввести для нее программный код событийной процедуры перевода чисел.
10	Запустить проект на выполнение. Ввести числа в два первых текстовых поля восьмеричной системы. Щелкнуть по кнопке *Oct-Dec*. Будет выполнен их перевод в десятичную систему.
10	Выполнить арифметическую операцию в десятичной системе, щелкнув, например, по кнопке «-». Перевести результат в восьмеричную систему, щелкнув по кнопке *Dec-Oct*.

Социальная информатика

Защита доступа к компьютеру. Для предотвращения несанкционированного доступа к данным, хранящимся на компьютере, используются пароли. Компьютер разрешает доступ к своим ресурсам только тем пользователям, которые зарегистрированы и ввели правильный пароль. Каждому конкретному пользователю может быть разрешен доступ только к определенным информационным ресурсам. При этом может производиться регистрация всех попыток несанкционированного доступа.

Защита данных на дисках. Каждый диск, папка и файл локального компьютера, а также компьютера, подключенного к локальной сети, может быть защищен от несанкционированного доступа. Для этого могут быть установлены определенные права доступа (полный только чтение, по паролю), причем права могут быть различными для различных пользователей.

Защита информации в Интернете. Если компьютер подключен к Интернету, то в принципе любой пользователь, также подключенный к Интернету, может получить доступ к информационным ресурсам этого компьютера. Если сервер имеет соединение с Интернетом и одновременно служит сервером локальной сети, то возможно несанкционированное проникновение из Интернета в локальную сеть. Для того чтобы этого не происходило, устанавливается программный или аппаратный барьер между Интернетом и Интранетом с помощью брандмауэра.

Правовая охрана

Правовая охрана

Правовая охрана программы для ЭВМ и баз данных впервые в полном объеме введена в Российской Федерации Законом РФ «О правовой охране программ для электронных вычислительных машин и баз данных», который вступил в силу в 1992 году. Предоставляемая настоящим законом правовая охрана распространяется на все виды программы для ЭВМ, которые могут быть выражены на любом языке и в любой форме, включая исходный текст на языке программирования и машинный код.

Для признания и осуществления авторского права на программы для ЭВМ не требуется ее регистрация в какой – либо организации. Авторское право на программы для ЭВМ возникает автоматически при их создании.

Для оповещения о своих правах разработчик программы может, начиная с первого выпуска в свет программы, использовать знак охраны авторского права, состоящий из трех элементов:

- Буквы С в окружности или круглых скобках;
- Наименования (имени) правообладателя;
- Года первого выпуска программы в свет.

Итоговый тест за основную школу

1. ??? За минимальную единицу измерения количества информации принят ...
   - 1 бод
   - 1 пиксель
   - 1 байт
   - 1 бит

2. ??? В рулетке общее количество лунок равно 32. Какое количество информации мы получаем в зрительном сообщения об остановке шарика в одной из лунок?
   - 8 бит
   - 5 бит
   - 2 бит
   - 1 бит

3. ??? Черно-белое (без градаций серого) растровое графическое изображение имеет размер 10*10 точек. Какой объем памяти займет это изображение?
   - 100 бит
   - 100 байт
   - 10 Кбайт
   - 1000 бит

4. ??? Как записывается десятичное число 11_{10} в двоичной системе счисления?
   - 1111
   - 1101
   - 1011
   - 1001

5. ??? Определить истинность составного высказывания: "(2 x 2 = 4 или 3 x 3 = 10) и (2 x 2 = 5 или 3 x 3 = 9)"
   - ложно
   - истинно
   - не ложно и не истинно
   - не истинно

6. ??? В целях сохранения информации жесткие магнитные диски необходимо оберегать от ...
   - пониженной температуры
   - перепадов атмосферного давления
   - света
   - ударов при установке

7. ??? Файл - это ...
   - единица измерения информации
   - программа или данные на диске, имеющие имя
   - программа в оперативной памяти
   - текст, распечатанный на принтере

8. ??? Операционная система – это ...
   - программа, обеспечивающая управление базами данных
   - антивирусная программа
   - программа, управляющая работой компьютера
   - система программирования

9. ??? Может ли присутствовать компьютерный вирус на чистой дискете (на дискете отсутствуют файлы)?

- *нет*
- *да, в области данных*
- *да, в области каталога*
- *да, в загрузочном секторе дискеты*

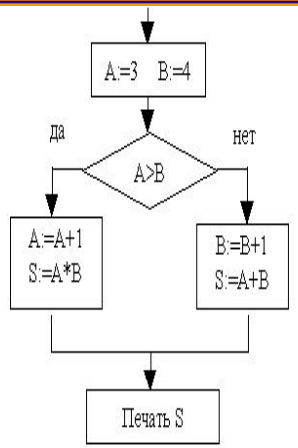
10. ??? Какие пары объектов находятся в отношении "объект - модель"?

- *компьютер - данные*
- *компьютер - его функциональная схема*
- *компьютер - программа*
- *компьютер - алгоритм*

11. ??? Какая модель компьютера является формальной (полученной в результате формализации)?

- *техническое описание компьютера*
- *фотография компьютера*
- *логическая схема компьютера*
- *рисунок компьютера*

12. ??? Фрагмент алгоритма изображен в виде блок-схемы. Определите, какое значение переменной S будет напечатано в результате выполнения алгоритма.



- *12*
- *3*
- *4*
- *8*

13. ??? Какое значение переменной S будет напечатано после выполнения фрагмента программы на Бейсике?

```
S=1
For N = 1 To 3
S=S*N
Next N
Print S
```

- *2*
- *3*
- *4*
- *6*

15. ??? Что будет меняться при представлении символа кириллицы на экране монитора в различных кодировках (Windows, MS-DOS, КОИ8-Р и т.д.)?

- *гарнитура шрифта*
- *размер символа*
- *символ*
- *начертание символа*

16. ??? В растровом графическом редакторе изображение формируется из ...

- *линий*
- *окружностей*
- *прямоугольников*
- *пикселей*

17. ??? Какой вид примет содержащая абсолютную и относительную ссылку формула, записанная в ячейке C1, после ее копирования в ячейку C2?

- *=$A$1*B2*
- *=$A$1*B1*
- *=$A$2*B1*
- *=$A$2*B2*

18. ??? В какой последовательности расположатся записи в базе данных после сортировки по возрастанию в поле *Память*?

- *1, 2, 3, 4*
- *4, 3, 2, 1*
- *4, 1, 2, 3*
- *2, 3, 4, 1*

19. ??? Браузер является ...

- *сетевым вирусом*
- *средством просмотра Web-страниц*
- *языком разметки Web-страниц*
- *транслятором языка программирования*

20. ??? Программы, которые можно бесплатно использовать и копировать, обозначаются компьютерным термином ...

- *hardware*
- *shareware*
- *freeware*
- *software*