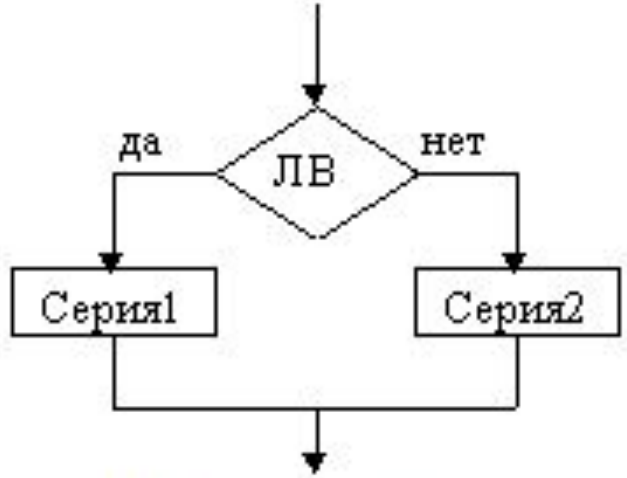
Алгоритмы и решение задач.

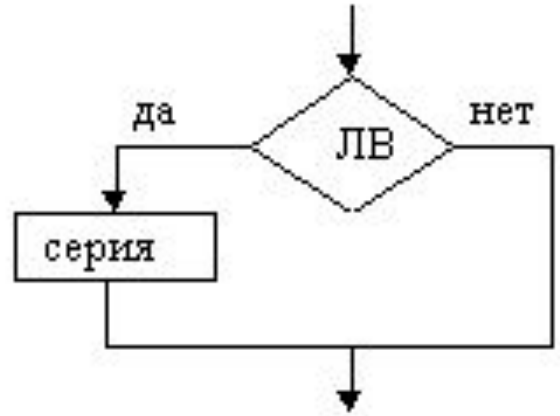
Программирование на языке Паскаль

Урок 1.

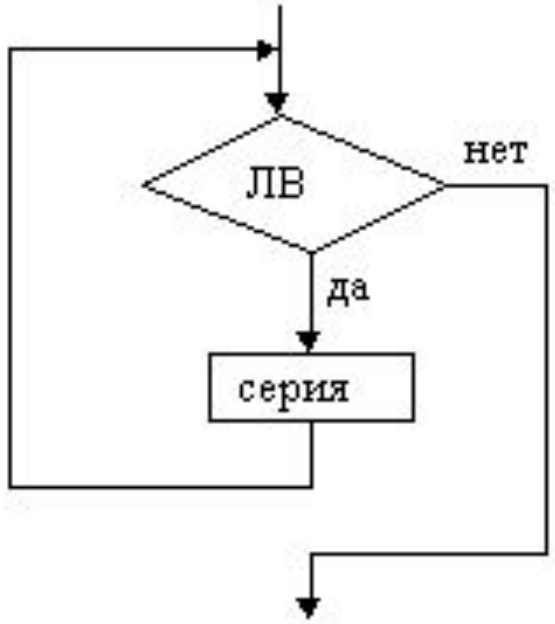
Структура "следование"



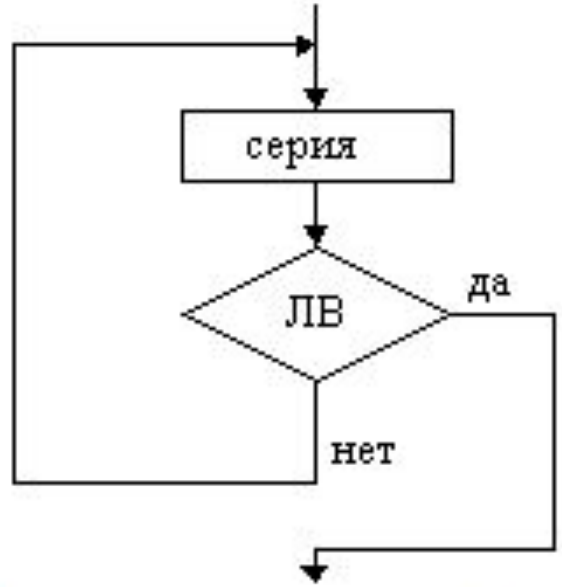
Полная развилка



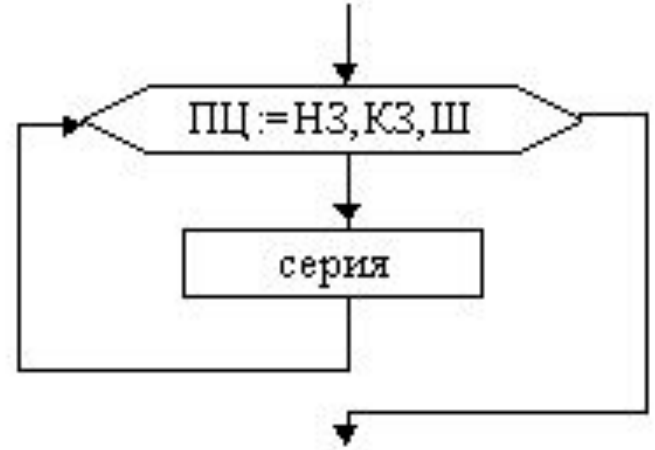
Неполная развилка



Цикл с предусловием (цикл ПОКА)



Цикл с постусловием (цикл ДО)

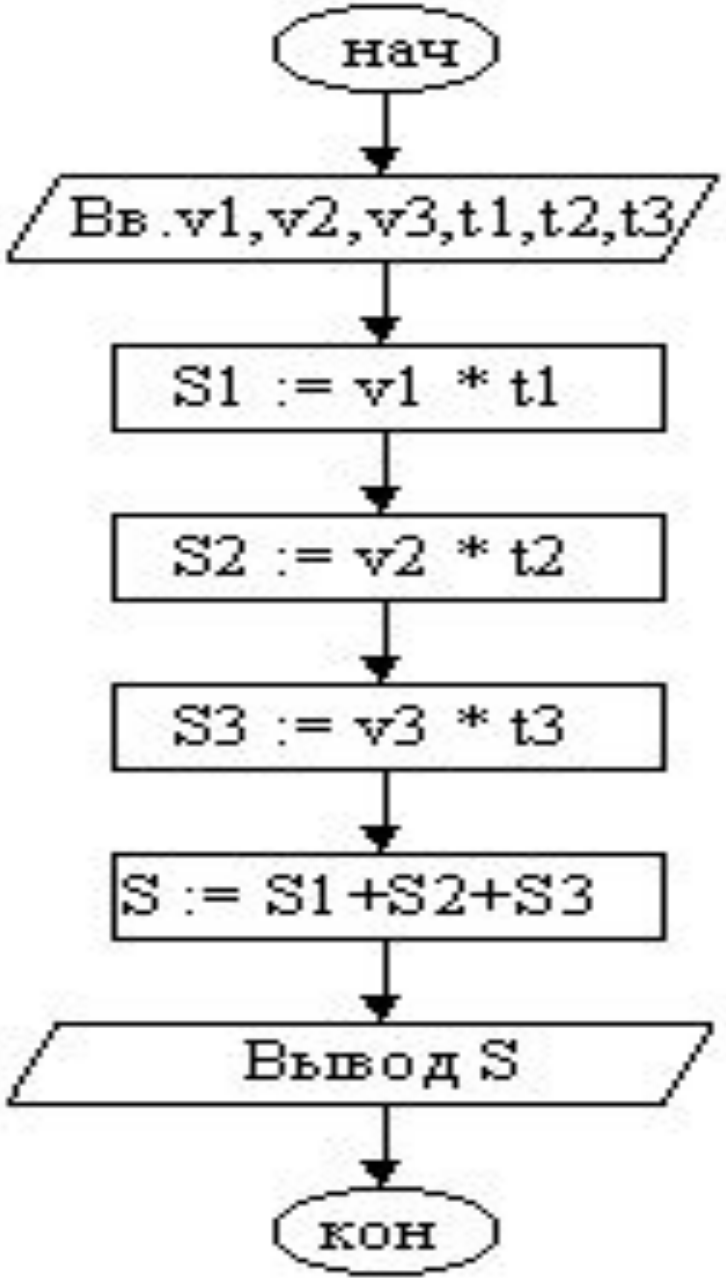


Цикл с параметром

Линейные алгоритмы

Линейные алгоритмы - это такие алгоритмы, когда действия выполняются в одну линию друг за другом.

Пример 1. Пешеход шел по пересеченной местности. Его скорость движения по равнине v1 км/ч, в гору — v2 км/ч и под гору — v3 км/ч. Время движения соответственно t1, t2 и t3 ч. Какой путь прошел пешеход?



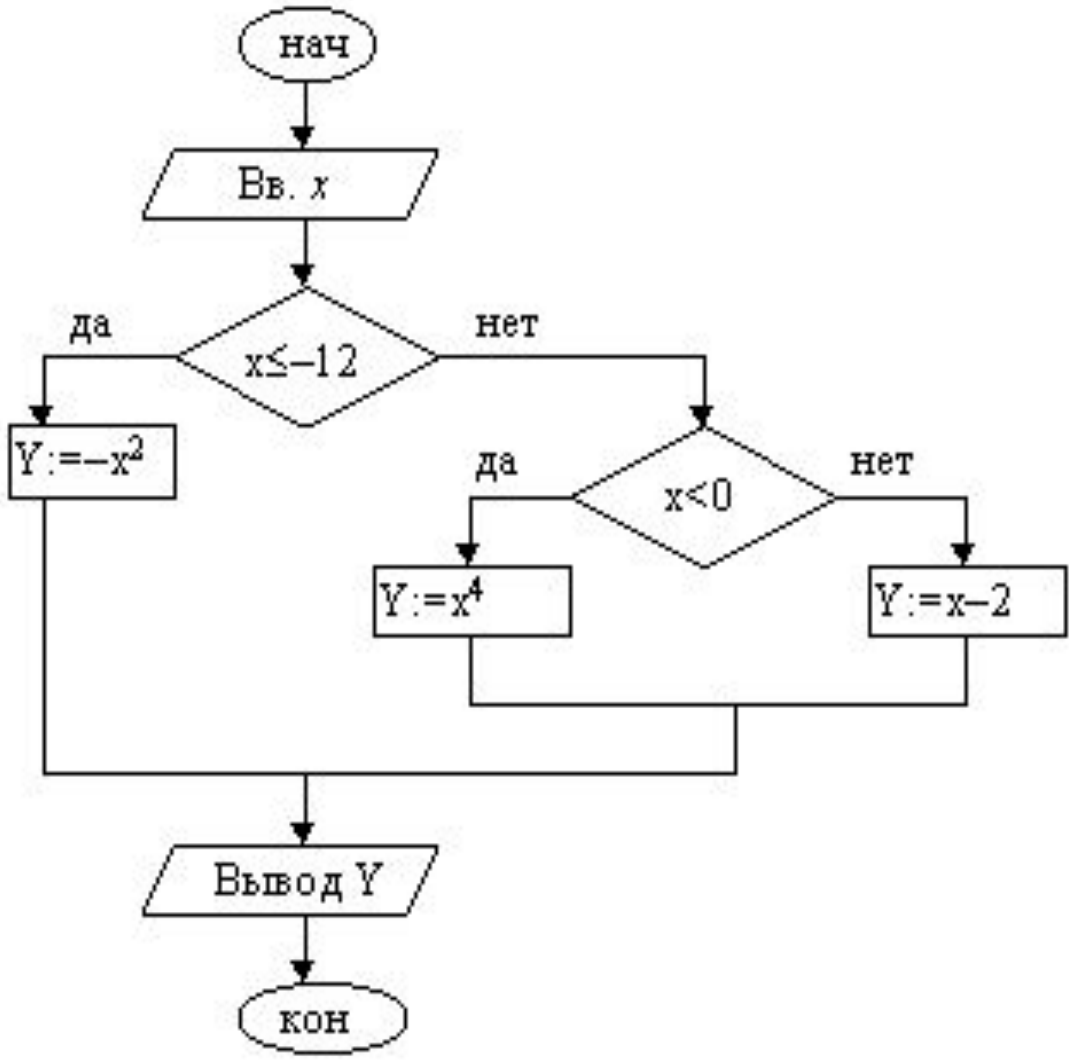
1. Ввести v1, v2, v3, t1, t2, t3.
2. S1 := v1 * t1.
3. S2 := v2 * t2.
4. S3 := v3 * t3.
5. S := S1 + S2 + S3.
6. Вывести значение S.
7. Конец.

Развилка

Достаточно часто то или иное действие должно быть выполнено в зависимости от значения логического выражения, выступающего в качестве условия. В таких случаях используется развилка.

Пример 1. Вычислить значение функции

$$y = \begin{cases} -x^2 \text{ при } x \le -12, \\ x^4 \text{ при } -12 < x < 0, \\ x - 2 \text{ при } x \ge 0. \end{cases}$$



1. Ввести x.
2. Если x <= –12, то y:=–x^2
3. Если x<0, то y:=x^4
4. y := x–2
5. Вывести y
6. Конец

Пример 2. Дано натуральное число *n*. Если число нечётное и его удвоение не приведет к выходу за 32767 (двухбайтовое целое число со знаком), удвоить его, иначе — оставить без изменения.

Чтобы удовлетворить условию удвоения, число n должно быть нечетным и меньше 16384.

Проверка

Немного теории

В паскале за целочисленное деление отвечает **оператор div** .

z := x div y

x - число , которое будем делить на **y** (делимое)

y - число , на которое будем делить число x (делитель)

z - результат целочисленного деления (целочисленное частное)

Таким образом, вот такая запись (55 / 6) нацело = 9 в результате использования **оператора div** будет выглядеть так

z := 55 div 6

Запомните ! При использовании оператора div дробная часть будет отброшена!

При делении с остатком потребуется **оператор mod**

z := x mod y

x - число , которое будем делить на **y** (делимое)

y - число , на которое будем делить число x (делитель)

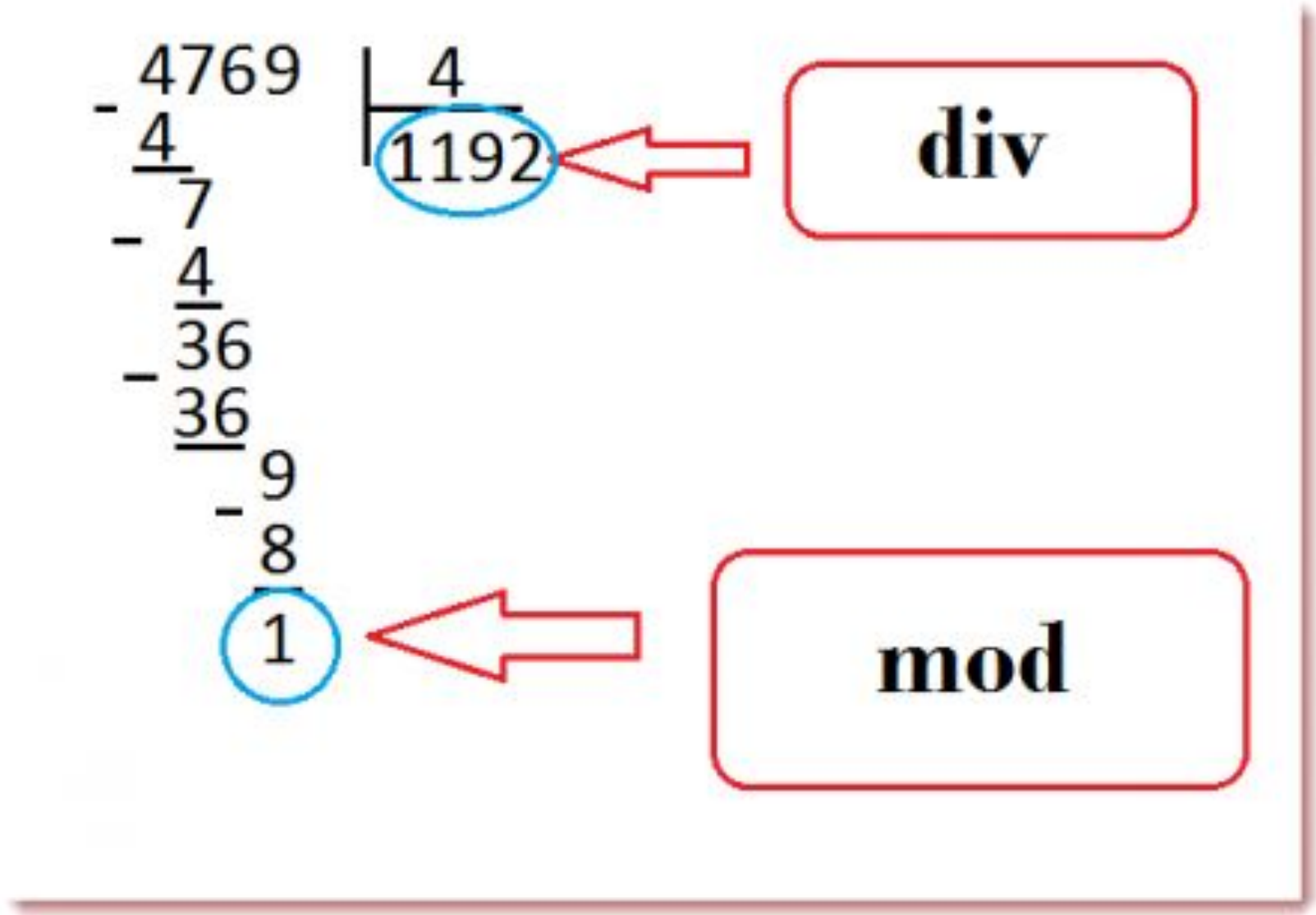
z - остаток

Например (40 / 6) с остатком = 4 с оператором mod будет такой

z := 55 mod 6

И как результат получим z=1 .

Запомните ! При использовании оператора mod целая часть будет отброшена!

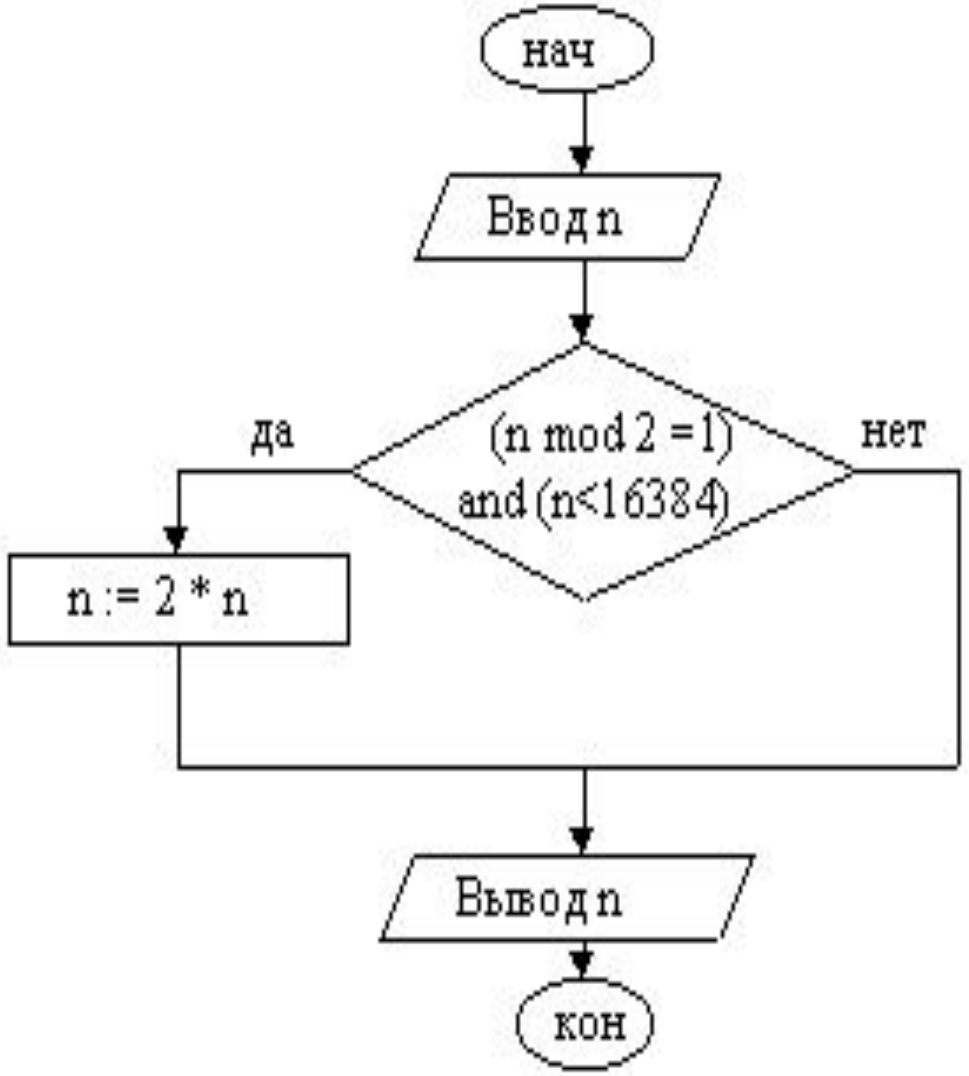

4769 | 4
-4 1192
7
-4
4
-36
36
9
-8
1
div
mod

Кстати **оператор mod** часто используют , для определения кратности чисел.

Если число кратное , то у него остаток равен 0

```
if  v mod m = 0 then …
```

Вернуться к задаче

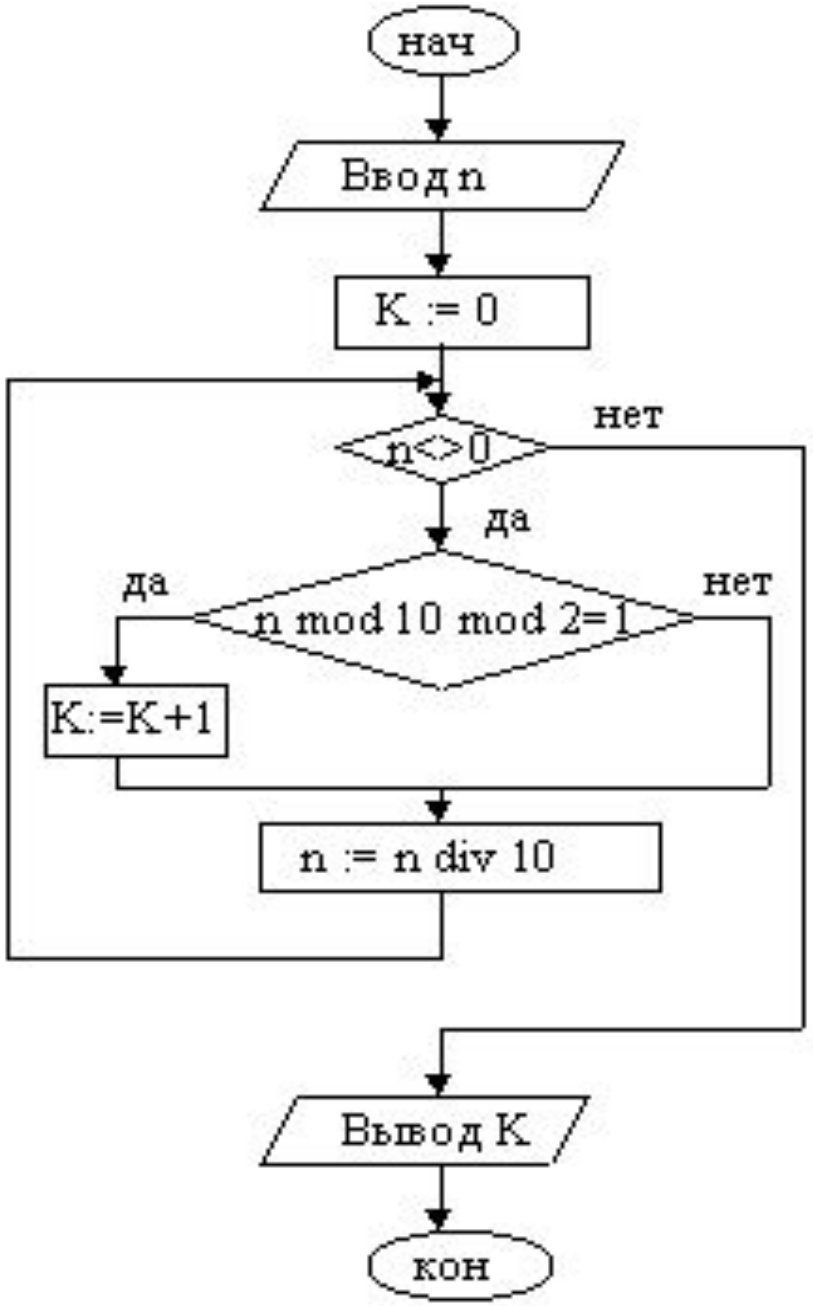


1. Ввести число *n*
2. Если число *n* нечетное и меньше 16384,
   то *n* := *n* * 2
3. Вывод *n*
4. Конец

Циклы

Если какие-либо операторы необходимо выполнить несколько раз, то их не переписывают каждый раз заново, а организуют цикл.

Пример 1. Подсчитать количество нечетных цифр в записи натурального числа n.



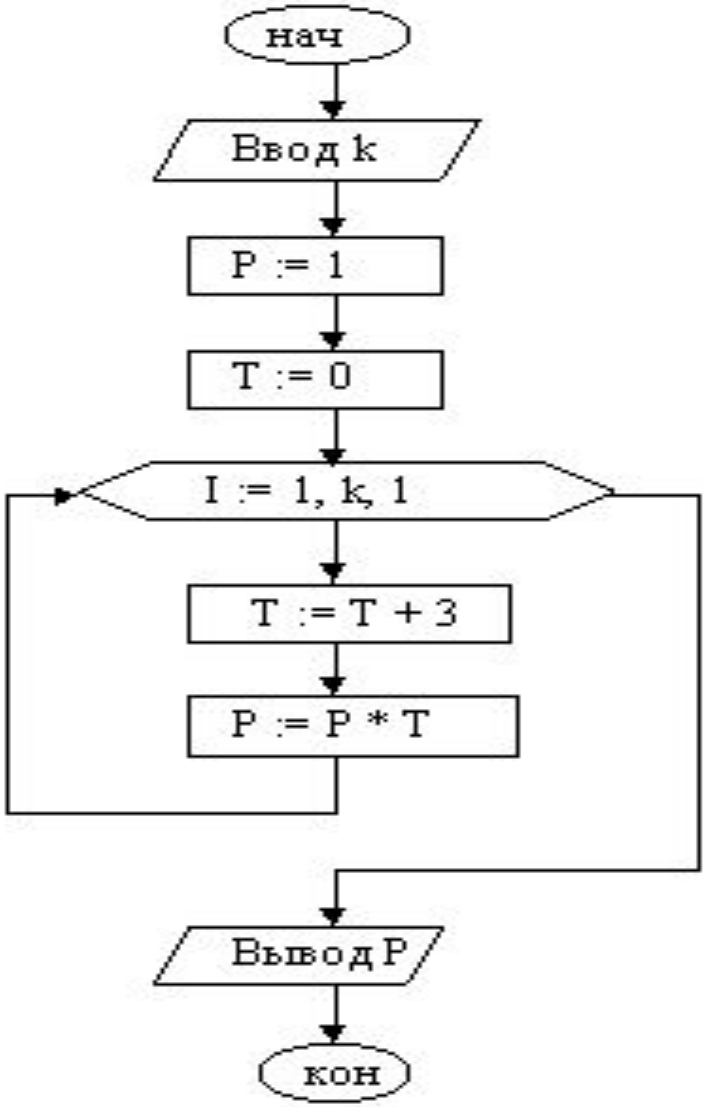
1. Ввести число n
2. K := 0 {подготавливаем счётчик}
3. Если n = 0, переход к п. 7
4. Если n mod 10 mod 2 = 1, то K := K +1
5. n := n div 10
6. Переход к п. 3
7. Вывод K
8. Конец

Задача 1:
Найти произведение первых *k* натуральных чисел, кратных трём.

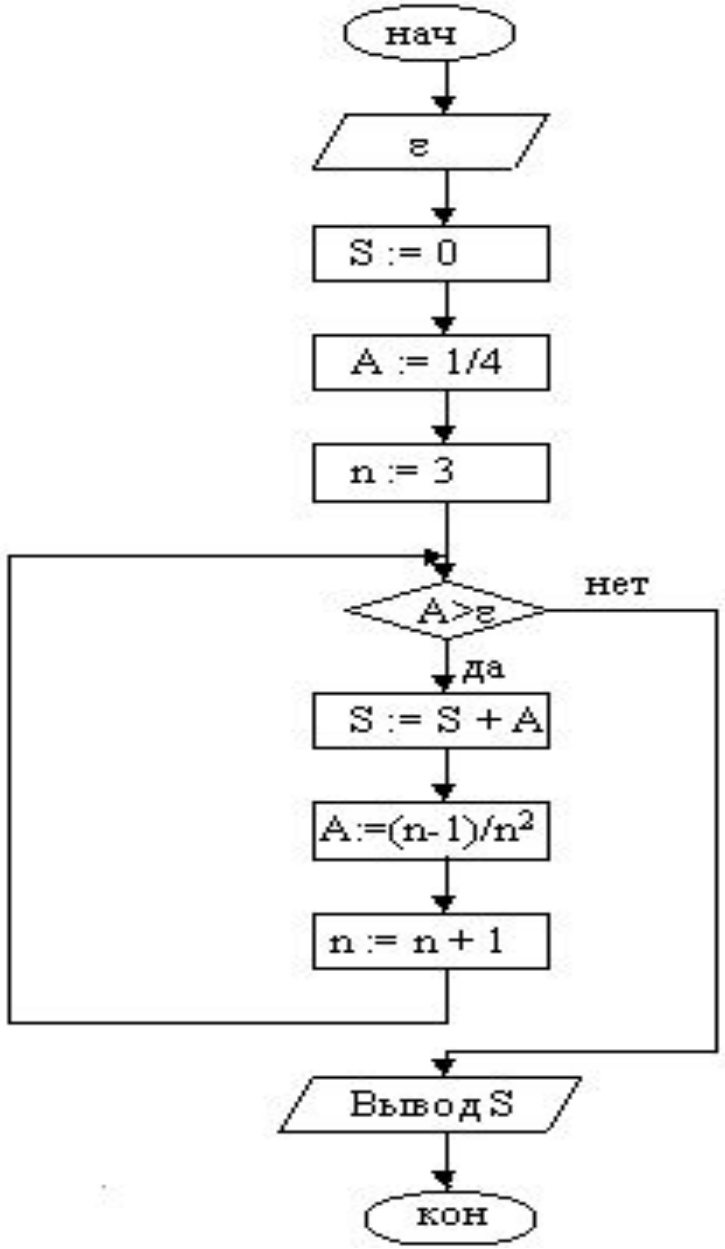
Задача 2:
Дана последовательность, общий член которой определяется формулой:

$$a_n = \frac{n-1}{n^2}.$$

Вычислить при $n>2$ сумму тех ее членов, которые больше заданного числа e.



1. Ввод k
2. P := 1 {здесь накапливаем произведение}
3. T := 0 {здесь будут числа, кратные 3}
4. I := 1
5. Если I > k, переход к п. 10
6. T := T + 3
7. P := P * T
8. I := I + 1
9. Перейти к п. 5
10. Вывод P
11. Конец



1. Ввести e
2. S := 0
3. A := 1/4
4. n := 3
5. Сравнить A с e. Если A>=e, переход к п. 10
6. S := S + A
7. A := (n-1)/(n*n)
8. n := n + 1
9. Переход к п. 5
10. Вывод S
11. Конец

Основный операторы языка программирования Pascal.

Программирование на языке Паскаль

Урок 2.

Оператор	Назначение
Program <имя программы>;	Начало программы
Var <список однотипных переменных>: <тип>;	Описание переменных
integer	Целый тип
real	Вещественный тип
boolean	Логический тип
char	Символьный тип
string	Строковый тип
Const <имя константы> =<значение>	Описание констант
Begin <операторы> end.	Операторные скобки
Read (<переменные>); readln (<переменные>)	Ввод данных

Оператор	Назначение
<переменная>:=<выражение>;	Присваивание значений
Write (<список вывода>); writeln (<список вывода>);	Вывод результатов
If <условие> then <оператор 1> [else<оператор 2>];	Ветвление
While <условие> do <оператор>;	Цикл с предусловием
Repeat <операторы> until <условие>	Цикл с постусловием
For <переменная>:=<н.з.> to <к.з.> do <оператор>;	Цикл с параметром
{любой текст в фигурных скобках}	Комментарии

Название	Служебное слово	Диапазон	Объем памяти
Короткое целое без знака	BYTE	0…255	1 байт, без знака
Короткое целое со знаком	SHORTINT	-128…127	1 байт, со знаком
Целое без знака	WORD	0…65535	2 байта, без знака
Целое со знаком	INTEGER	-32768…32767	2 байта, со знаком
Длинное целое со знаком	LONGINT	-2147483648… 214783647	4 байта, со знаком

Целочисленные типы

Операции с величинами целого типа: сложение (+), вычитание (-), умножение (*), нахождение целой части деления (div), нахождение остатка от деления (mod).

Внимание! Переменной целого типа нельзя присваивать результат операции вещественного деления (/).

ТИП	ЗНАЧЕНИЯ	ОПЕРАЦИИ	ПРЕДСТАВЛЕНИЕ
Вещественный (real)	От 2,9*10-39 до 1,7 *10 +38	Арифметические операции (+,-,*,/), операции отношений	Формат с плавающей точкой (6 байт)
single	1.5*10E – 45..3.4*10E38		4 байта
double	5.0*10E – 324..1.7*10E308		8 байт

ТИП	ЗНАЧЕНИЯ	ОПЕРАЦИИ	ПРЕДСТАВЛЕНИЕ
Логический (boolean)	True , false	Логические операции, операции отношений	1 –True, 0- false (1 бит)

ТИП	ЗНАЧЕНИЯ	ОПЕРАЦИИ	ПРЕДСТАВЛЕНИЕ
Символьный тип (char)	Символы компьютерного алфавита	Операции отношений	Коды таблицы символьной кодировки(1 байт)

char -переменная «символ»,
Например, если нужно узнать 5 букву в слове "слон" и поместить ее в переменную типа char то будет ошибка.
Код таблицы символьной кодировки состоит из 256 символов, и каждый имеет свой номер. Чтобы узнать например номер буквы "п" нужно ввести команду ord ('п');
Чтобы узнать чей номер 115 в этом словаре, нужно набрать команду chr(115); которая нам вернет букву, которая в этом "букваре" под номером 115.

ТИП	ЗНАЧЕНИЯ	ОПЕРАЦИИ	ПРЕДСТАВЛЕНИЕ
Строковый тип (string)			

var s: **string**[n]; var s: **string**;

n - максимально возможная длина строки - целое число в диапазоне 1..255. Если этот параметр опущен, то по умолчанию он принимается равным 255.

Строковые константы записываются как последовательности символов, ограниченные апострофами. Пустой символ обозначается двумя подряд стоящими апострофами. Если апостроф входит в строку как литера, то при записи он удваивается.

```
program Modul1_1;                                   {Заголовок программы.}

 var                                                {Начало раздела описания переменных.}

  a, b, c: integer;                                 {Описание трех переменных целого типа.}

begin                                               {Начало тела программы.}

 writeln ('Введите два целых числа =>  ');          {Вывод на экран запроса входных данных. Данный
                                                    оператор writeln выведет на экран фразу заключенную в
                                                    апострофы.}

 readln (a, b);                                     {Ввод значений переменных a и b.}

 c:=a+b;                                            {Переменной c присваивается значение суммы
                                                    переменных a и b.}

 writeln ('Сумма введенных чисел равна ', c);       {Вывод на экран полученной суммы..}

 writeln ('Нажмите Enter');                         {Вывод на экран сообщения.}

 readln;                                            {Ожидание нажатия клавиши Enter.}
```

Экспериментальные задачи

Измените программу Modul1_1 так, чтобы она находила произведение двух целых чисел. Запустите измененную программу со следующими входными данными: a=3456, b=789. Выполнение вашей программы или прервется сообщением Run-time error, или вы получите неправдоподобный (отрицательный) результат. Найдите экспериментальным путем тот интервал значений переменных a и b, когда результат умножения будет правильным. Объясните, почему это так.

Тип результата должен соответствовать типу операндов, а если операнды относятся к различным целым типам, то выдается ошибка. Диапазон значений типа integer от -32768 до +32767. В данном случае результат вычисления будет верным если вводить числа от -181 до +181.

Измените в программе Modul1_1 оператор c:=a+b на c:=a-(a div b)*b. Какое действие производит операция div над переменными целого типа?

Измените вывод результата так, чтобы текстовая строка отражала результат вашего исследования.

```
var a,b,c: integer;
begin
writeln ('введите два целых числа');
readln (a,b);
c:= a- (a div b)*b;
writeln ('Остаток от деления числа a на b равен',c);
readln;
end.
```

Добавьте к программе предыдущего задания (№2) переменную с именем d, оператор присваивания d:=a mod b; и оператор вывода на экран writeln ('????', d);. Какое действие выполняется операцией mod?
Замените знаки вопроса пояснениями его работы.

```
var a,b,d: integer;
begin
writeln ('введите два целых числа');
readln (a,b);
d:= a mod b;
writeln ('Остаток от деления a на
b',d);
readln;
end.
```

Целый тип данных.

Программирование на языке Паскаль

Урок 3.

```
a:=19 div 4; {a=4}
a:=19 div -4; {a=-4}
a:=-19 div 4; {a=-4}
a:=-19 div -4; {a=4}
```

```
a:=19 mod 4; {a=3}
a:=19 mod -4; {a=3}
a:=-19 mod 4; {a=-3}
a:=-19 mod -4; {a=-3}
```

Задача 1.

Дано целое трехзначное число *n*. Составить программу, находящую сумму его цифр.

Разберем алгоритм решения поставленной задачи.

Пусть дано число $n=572$

Разложим $n=5\times100+7\times10+2=5\times10^2+7\times10^1+2$.

Для того, чтобы найти цифру единиц данного числа *n* (в нашем случае - это 2), необходимо найти остаток от деления данного числа *n* на 10.

Для нахождения цифры десятков необходимо разделить данное число нацело на 10 (в нашем случае в результате получим 57), а затем найдем остаток от деления найденного числа на 10 (57 mod 10=7).

Для нахождения цифры сотен необходимо разделить нацело на 100 данное число *n*.

Для нахождения требуемого результата сложим все найденные цифры.

```
program Modul1_2;
var
n, one, dec, hun, s: integer;
begin
writeln ('Введите трехзначное число:');
readln ( n );
one:=n mod 10; {находим цифру единиц}
dec:=(n div 10) mod 10; {находим цифру десятков}
hun:=n div 100; {находим цифру сотен}
s:=one+dec+hun; {находим результат - сумму цифр трехзначного
числа}
writeln ('Сумма цифр данного числа равна: ', s);
writeln ('Нажмите Enter');
readln;
end.
```

Пройти
тест

Задача 1. Дано натуральное трехзначное число N. Определить произведение его цифр .

Практикум линейные алгоритмы.

Программирование на языке Паскаль

Урок 4.

Оператор ветвления.

Программирование на языке Паскаль

Урок 5.

Процедуры ввода-вывода и оператор присваивания позволяют писать только *линейные* программы, в которых все команды выполняются последовательно, одна за другой. Но очень часто возникает необходимость выполнять различные команды в зависимости от выполнения какого-то условия (например, надо найти наибольшее из двух чисел). Для таких целей существует оператор ветвления (или условный оператор):

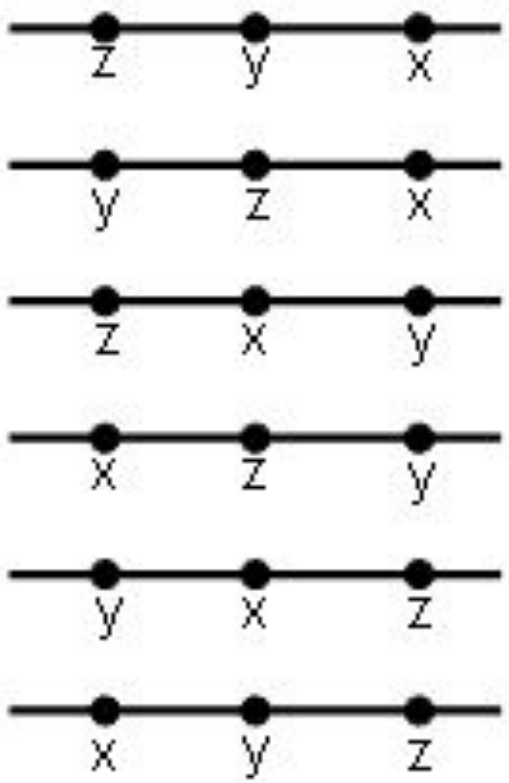
```
If <условие>
then <оператор 1>
else <оператор 2>;
```

Примечание: Обратите внимание, что точка с запятой (разделитель операторов) стоит после <оператор 2>, то есть после ВСЕЙ конструкции ветвления.

Экспериментальная задача:

Найти наибольшее из трех чисел *x*, *y* и *z*. Предположим, что нет равенств, то есть все числа различны.

Возможны шесть различных случаев, они приведены на рисунке.



Программа определения значения наибольшего из трех чисел имеет вид:

```
program ex4;
var x, y, z: integer;
begin
writeln ('Введите три числа через пробел');
readln (x, y, z);
if (x>y) and (x>z)
then writeln (x)
else
if (y>x) and (y>z)
then writeln ( y )
else writeln (z);
readln;
end.
```

Предложите иную версию решения данной задачи.

```
var
   x, y, z, max: integer;

begin
   write ('Введите три числа: ');
   readln (x, y, z);
   if x >= y then
      max := x
   else
      max := y;
   if z > max then
      max := z;
   writeln ('Максимальное из них: ', max);
readln
end.
```

Оператор ветвления. Практикум по решению задач

Программирование на языке Паскаль

Урок 6.

Оператор выбора

Программирование на языке Паскаль

Урок 7.

Задача 1. Даны действительные числа *x*, *y*. Если *x* и *y* отрицательны, то каждое значение заменить модулем; если отрицательно только одно из них, то оба значения увеличить на 0,5; если оба значения неотрицательны и ни одно из них не принадлежит отрезку [0,5; 2,0], то оба значения уменьшить в 10 раз; в остальных случаях *x* и *y* оставить без изменения.

(10 минут)

```
Program Usl;
 Var X, Y : Real;
 Begin
  Write('Введите два действительных числа '); ReadLn(X, Y);
  If (X < 0) AND (Y < 0) THEN
  Begin
    X = ABS(X);
    Y = ABS(Y)
  End
                ELSE
                 IF (X < 0) OR (Y < 0) THEN
                  Begin
                   X = X + 0.5;
                   Y = Y + 0.5
                  End
                          ELSE
                           IF NOT (((X >= 0.5) AND (X <= 2))
                                 OR ((Y >= 0.5) AND (Y <= 2)))
                               THEN
                                Begin
                                      X = X / 10;
                                      Y = Y / 10
                                End;
 WriteLn('Результат:'); WriteLn('X= ', X:10:6); WriteLn('Y= ', Y:10:6);  END.
```

Оператор выбора CASE

Формат: *CASE [ключ_выбора] OF*
[константа_выбора_1]:[оператор_1];
[константа_выбора_2]:[оператор_2];
…
[константа_выбора_N]:[оператор_N];
ELSE [оператор];
End;

Здесь **case, of, else, end** – зарезервированные слова (случай, из, иначе, конец);

Составить программу задачи, моделирующей работу светофора. При вводе символа первой буквы цветов светофора, программа должна выводить сообщение о соответствующем цвете и действиях.

```
Program Svetofor;
  Var
    cvet:char;
Begin
  WriteLn('введите символ цвета');
  Read(cvet);
  Case cvet of
     'з': WriteLn('Зеленый цвет, движение разрешено');
     'ж': WriteLn('Желтый цвет, внимание');
     'к': WriteLn('Красный цвет, движение запрещено');
     Else WriteLn('Светофор не работает');
  End;
```

Задача 1. В старо японском календаре был принят двенадцатилетний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, петуха, собаки и свиньи. Написать программу, которая позволяет ввести номер года и печатает его название по старо японскому календарю. Справка: 1996 г. — год крысы — начало очередного цикла.

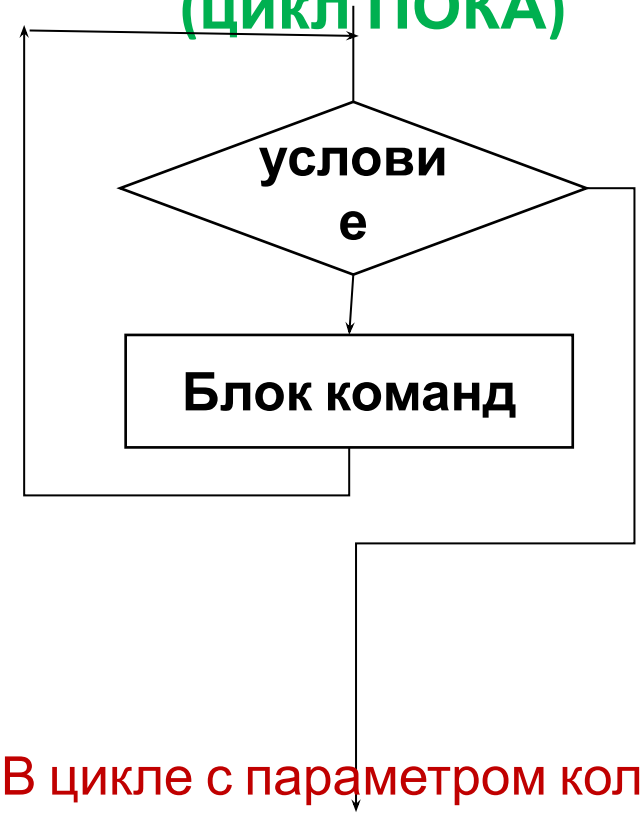
```
Program Goroskop;
Var Year : Integer;
Begin
 Write('Введите год ');
  ReadLn(Year);
   CASE  Year MOD 12 OF
    0 : WriteLn ('Год Обезьяны');
    1 : WriteLn ('Год Петуха');
    2 : WriteLn ('Год Собаки');
    3 : WriteLn ('Год Свиньи');
    4 : WriteLn ('Год Крысы');
    5 : WriteLn ('Год Коровы');
    6 : WriteLn ('Год Тигра');
    7 : WriteLn ('Год Зайца');
    8 : WriteLn  ('Год Дракона');
    9 : WriteLn ('Год Змеи');
   10 : WriteLn ('Год Лошади');
   11 : WriteLn ('Год Овцы')
  END;
END.
```

Циклы: трассировка алгоритма

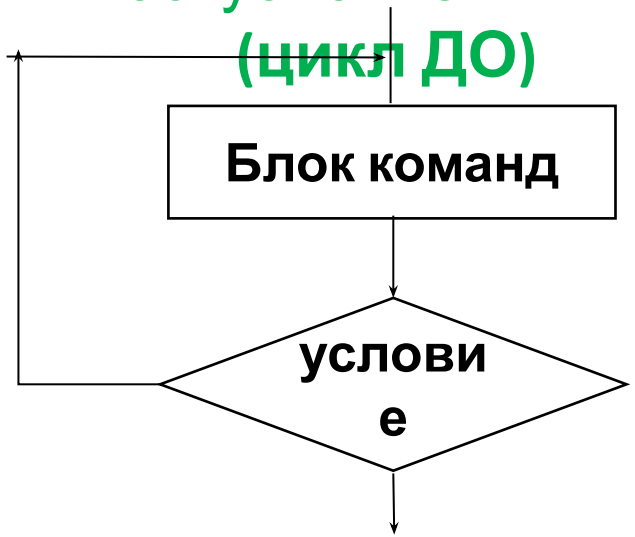
Программирование на языке Паскаль

Урок 8.

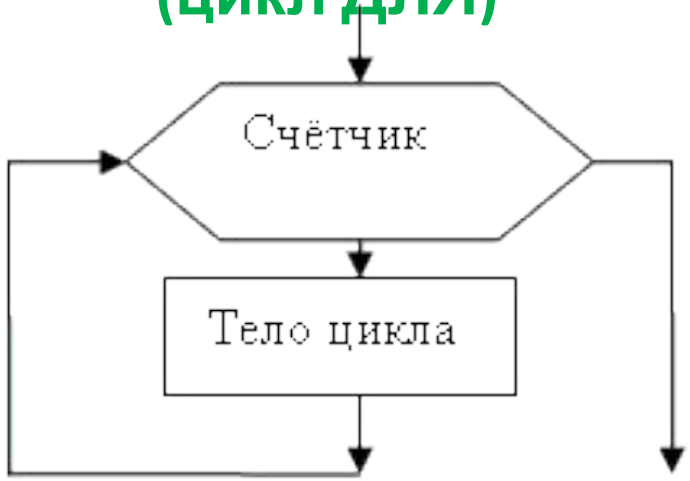
Цикл с предусловием **(цикл ПОКА)**



Цикл с постусловием **(цикл ДО)**



Цикл с параметром **(цикл ДЛЯ)**



В цикле с параметром количество выполнений тел цикла задано жестко и однозначно.
В циклах ПОКА и ДО количество выполнений тела цикла заранее не известно и определяется ситуацией. Поэтому необходимо проследить, чтобы во время выполнения циклов ПОКА и ДО происходило какое то изменение переменных, задействованных в условии цикла. Несоблюдение этого условия приведет к «зацикливанию» программы.

Конструкция циклов

Цикл с предусловием **(цикл ПОКА)**

While <условие>
Begin
<блок команд>
end;

Или

Do while <условие> do
<команда>;

Цикл с постусловием **(цикл ДО)**

Repeat < команды>
until < условие>;

Цикл с параметром **(цикл ДЛЯ)**

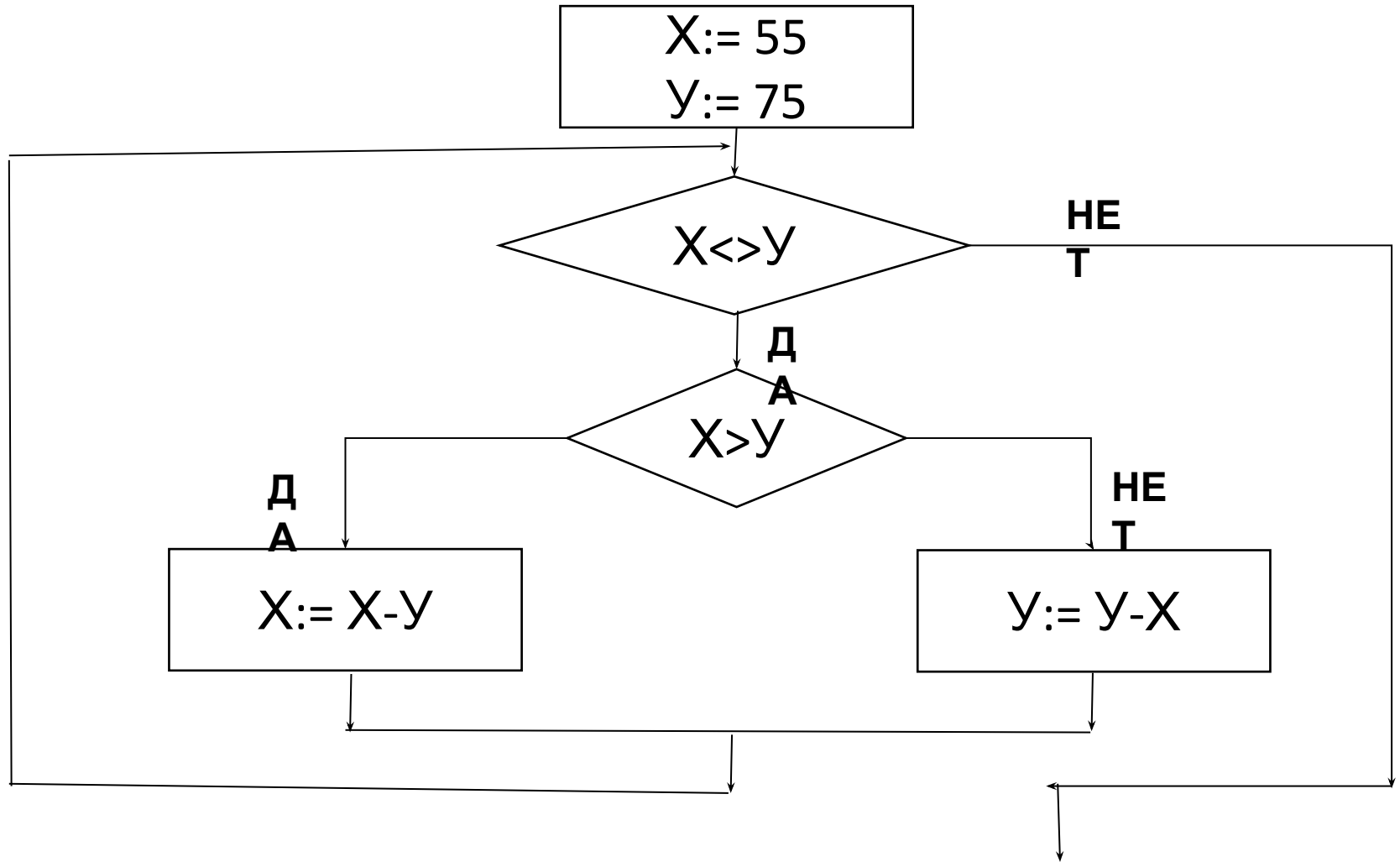
For <i> := <in> To <ik> do
Begin
<блок команд>
End;

ИЛИ

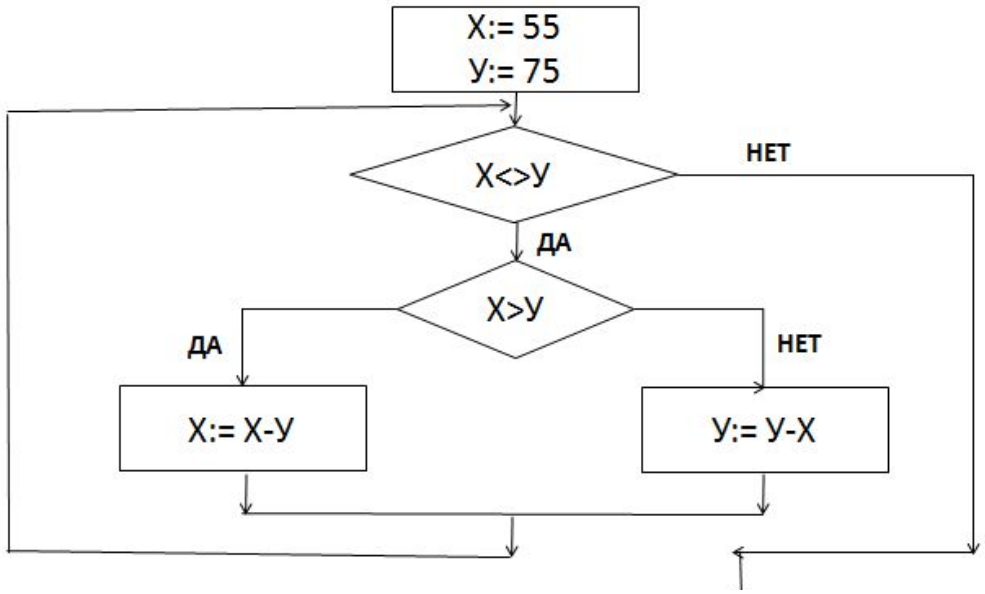
For <i> := <in> To <ik> do
< команда>;

Разбор типовых задач

Определите значение целочисленной переменной X после выполнения следующего фрагмента программы



МЕТОД ТРАССИРОВКИ

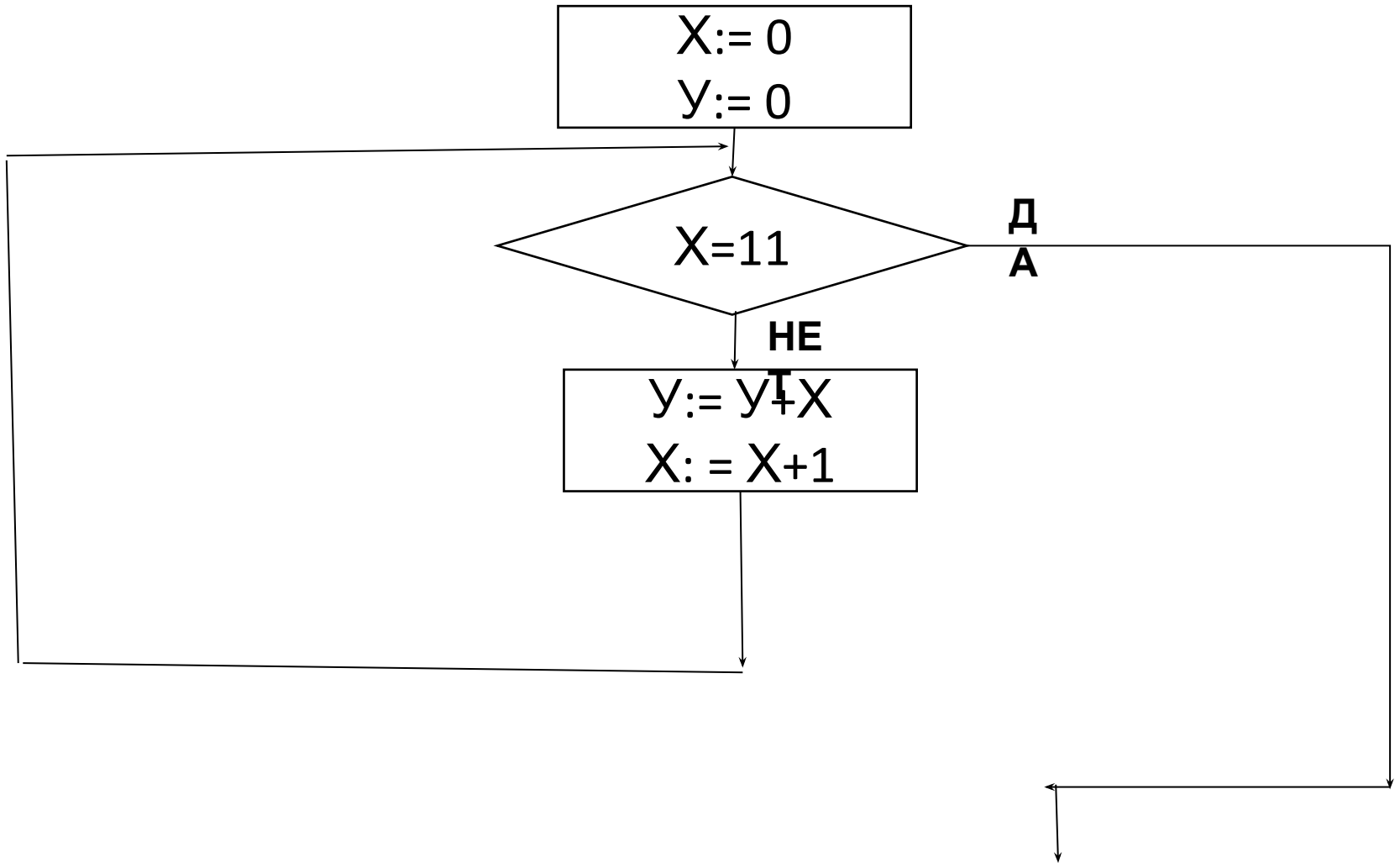


Операторы в блоке	X	У
X:= 55 У:= 75	55	75
X<>У *Условие выполняется- Ветвь «ДА»*	55	75
X>У *Условие не выполняется- Ветвь «НЕТ»*	55	75
У:= У-X	55	20
X<>У *Условие выполняется- Ветвь «ДА»*	55	20
X>У *Условие выполняется- Ветвь «ДА»*	55	20
X:= X-У	35	20
И т.д.		
Ответ	5	

5

Разбор типовых задач

Определите значение переменной У после выполнения фрагмента алгоритма:



Ответ У = 55

Разбор типовых задач

Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
Var k, s : integer;
Begin
S:=0;
K:= 0;
While s<1024 do
Begin
S:= s+10;
K:= k+1;
End;
Writeln (k);
Readln;
end.
```

Ответ У = 103