

ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ

1. Какие типы алгоритмических структур вам известны?
 - *Следование, ветвление и циклические алгоритмы.*
2. На прошлых уроках мы с вами разбирали ...
 - *Программирование линейных алгоритмов*

Написать программу,
считающую
сумму 3 товаров

```
Program sum;  
var a,b,c,s:real;  
begin  
  cls;  
  writeln ('Программа подсчитывает стоимость 3 товаров в рублях');  
  write ('Цена товара №1=');  
  read (a);  
  write ('Цена товара №2=');  
  read (b);  
  write ('Цена товара №3=');  
  read (c);  
  s:=a+b+c;  
  writeln ('Общая стоимость товаров = ',s, ' рублей');  
end.
```

- Дано целое число. Если оно является положительным, то прибавить к нему 1. Вывести полученное число.



**Скажи мне – и я забуду,
Покажи мне – и я
запомню,
Вовлеки меня – и я
пойму.**

*Программирование
разветвляющихся
алгоритмов*

8

класс

РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ

Разветвляющийся алгоритм – это алгоритм в котором выполняется либо одна, либо другая группа действий в зависимости от истинности или ложности условия .



Ключевые слова

- **условный оператор**
- **неполный условный оператор**



Общий вид условного оператора

Полная форма условного оператора:

if <условие> **then** <оператор_1> **else** <оператор_2>

Неполная форма условного оператора:

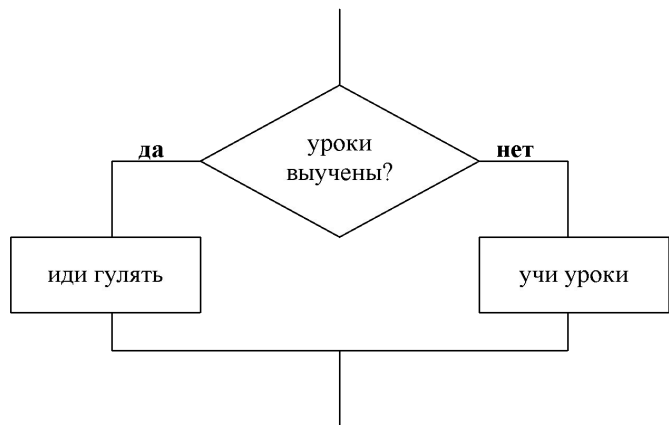
if <условие> **then** <оператор>



Перед **else** знак «;» не ставится.

Полная форма

Если <условие>, то <действие 1>, иначе <действие 2>



Пример: Если уроки выучены то иди гулять иначе седи учи.

Неполная форма

Если <условие>, то <действия>



Пример: Если низко ласточки летают, то будет дождь.



Условия - логические выражения

```
graph TD; A[Условия - логические выражения] --> B[Простые]; A --> C[Сложные]; B --> D[Операции отношений]; C --> E[Логические операции];
```

Простые

**Операции
отношений**

Сложные

**Логические
операции**

УСЛОВИЕ (СРАВНЕНИЕ ВЫРАЖЕНИЙ)

простое

операции отношений

= равно

< меньше

> больше

<= меньше или равно

>= больше или равно

<> не равно

Примеры: 1) if (n>0) then n:=n+2;

2) if (n<0) **and** (n<>-3) then n:=n+1;

составное

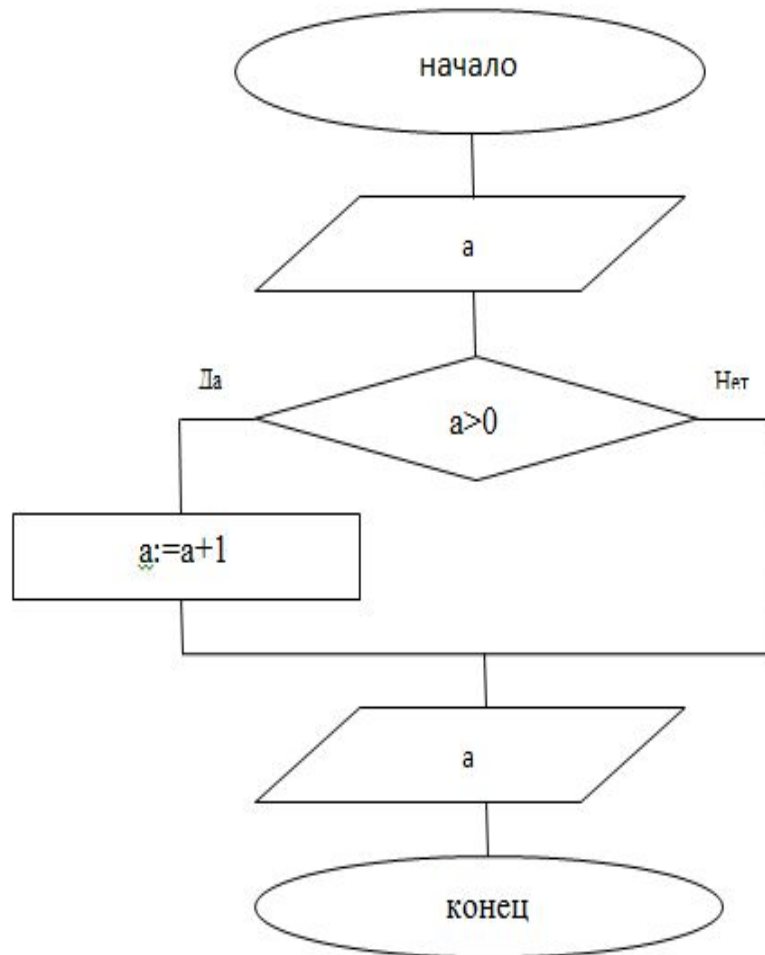
логические операции

not не

and и

or или

Неполный условный оператор



Programzadacha;

Vara :integer;

Begin

Writeln('введите целое число a');

readln(a);

if a > 0 then a := a + 1;

Writeln('Полученное число ', a);

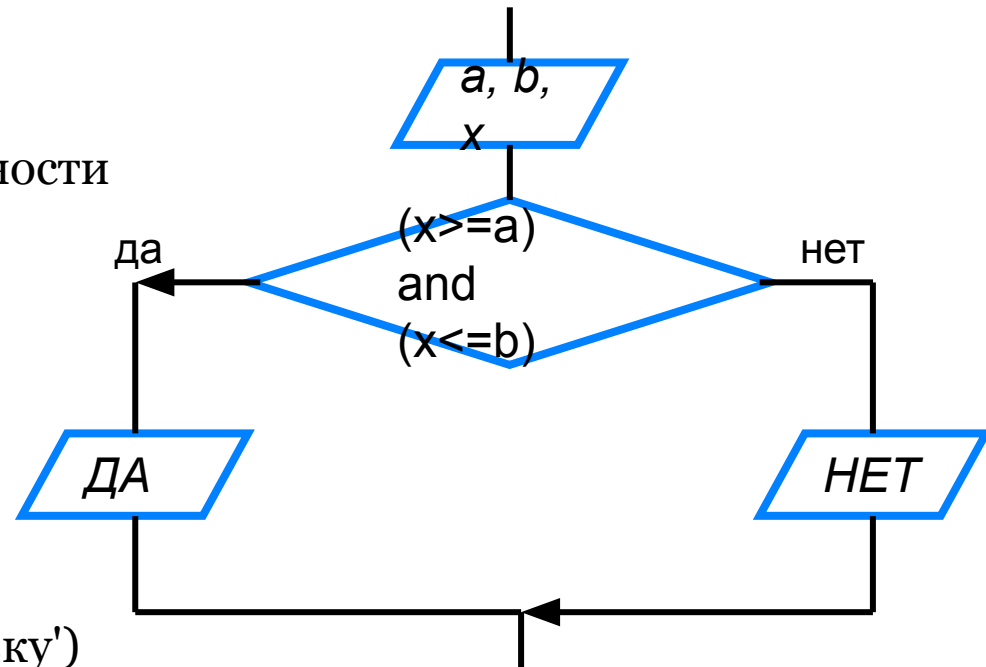
end.

Пример.

Запишем на языке Паскаль алгоритм определения принадлежности точки x отрезку $[a,b]$. Если точка x принадлежит данному отрезку, то выводится ответ ДА, в противном случае – НЕТ.

Условный оператор

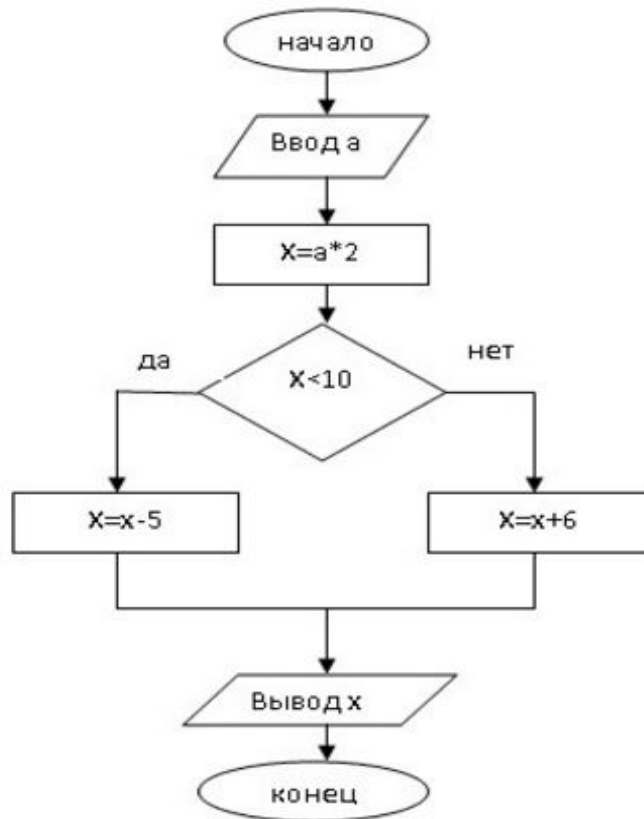
```
program n_9;  
  var x, a, b: real;  
begin  
  writeln ('Определение принадлежности  
           точки отрезку');  
  write ('Введите a, b>>');  
  readln (a, b);  
  write ('Введите x>>');  
  readln (x);  
  if (x>=a) and (x<=b) then  
    writeln ('Точка принадлежит отрезку')  
  else writeln ('Точка не принадлежит отрезку')  
end.
```



Домашнее задание

Задача 1. Ввести два числа. Если их сумма больше 100, то сумму уменьшить в 2 раза, в противном случае увеличить в 2 раза.

Задача 2. На рисунке ниже представлен алгоритм разветвленной структуры, представленной в виде блок-схемы. Запишите данный алгоритм на языке Паскаль.



Спасибо за внимание!