

Язык программирования Pascal

Ветвление

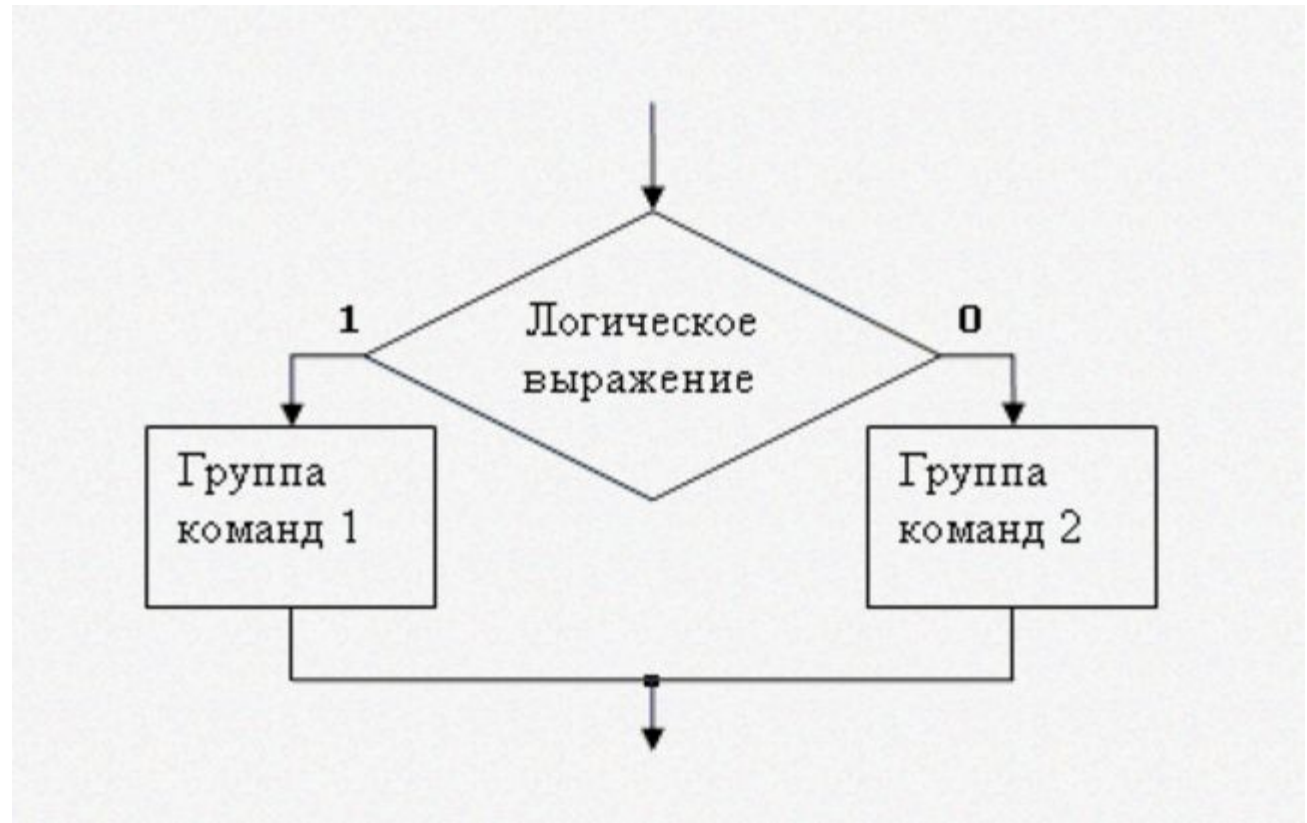
Цель урока

Научиться программировать на языке Паскаль алгоритмы ветвления.

1. Что такое алгоритм?
2. Какие типы алгоритмов Вы знаете?
3. Что такое линейный алгоритм?
4. Что такое ветвление в алгоритме?
5. Какие Виды ветвлений Вы знаете?

Ветвление

В **полном ветвлении** предусмотрено два варианта действий исполнителя в зависимости от значения логического выражения (условия). *Если условие истинно, то выполняться будет только первая ветвь, иначе только вторая ветвь.*



Условный оператор if...then...else полная форма

Синтаксис:

```
if <логическое выражение>  
  then <оператор1>  
  else <оператор2>;
```

Семантика:

1. Вычисляется значение <логического выражения>
2. Если <логическое выражение> истинно (TRUE), то выполняется оператор1, иначе выполняется оператор2.



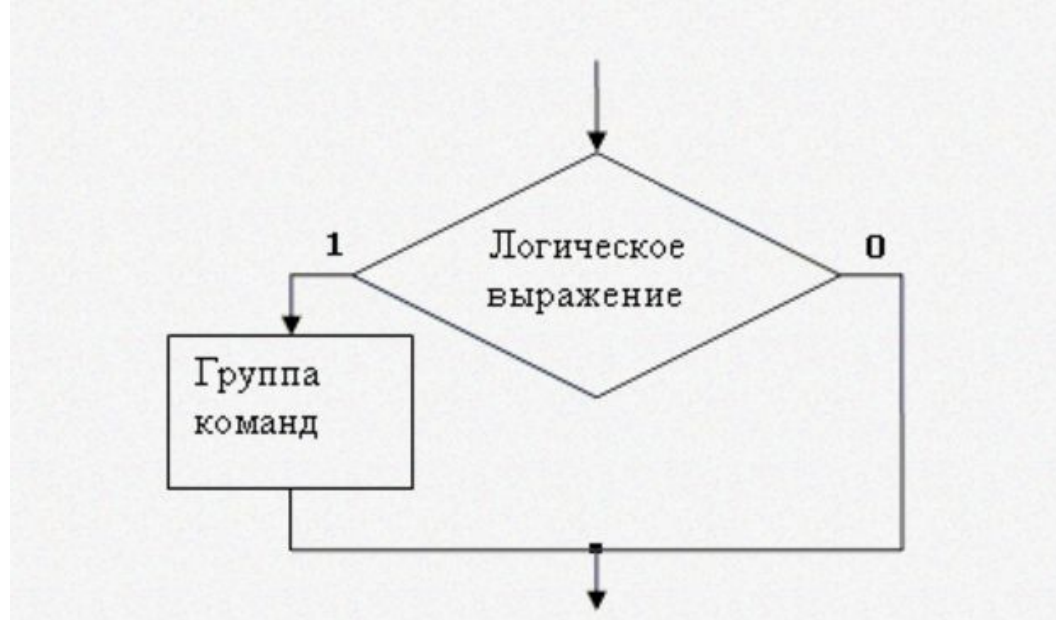
Пример:

If2. Дано целое число **N**.

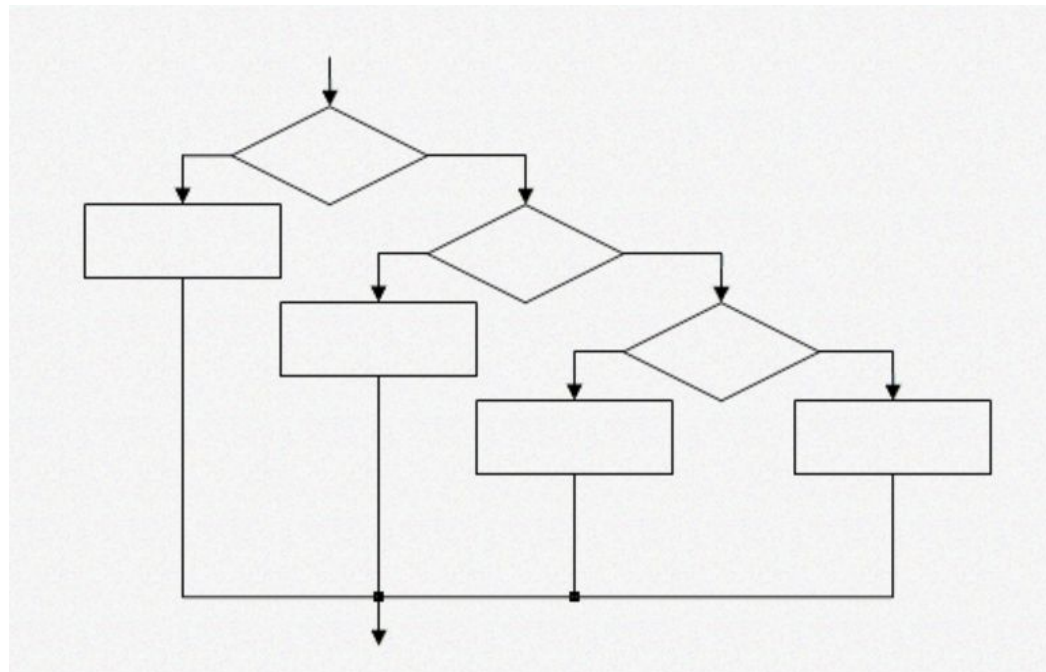
Если оно положительное, то прибавить к нему 1; если отрицательно вычесть из него 2. Вывести полученное число.

```
program if_2;  
var n: integer;  
begin  
  write ('введите целое число n=');  
  readln (n);  
  if n>0  
    then n:=n+1  
    else n:=n-2;  
  writeln ('n=',n);  
end.
```

Вторая ветвь может быть пустой. Такая структура называется **неполным ветвлением** или **обходом**.



Из нескольких ветвлений можно сконструировать структуру «**выбор**» (множественное ветвление), которая будет выбирать не из двух, а из большего количества вариантов действий исполнителя, зависящих от нескольких условий. Существенно, что выполняется только одна ветвь - в такой структуре важное значение приобретает порядок следования условий: если выполняются несколько условий, то сработает только одно из них - первое сверху.



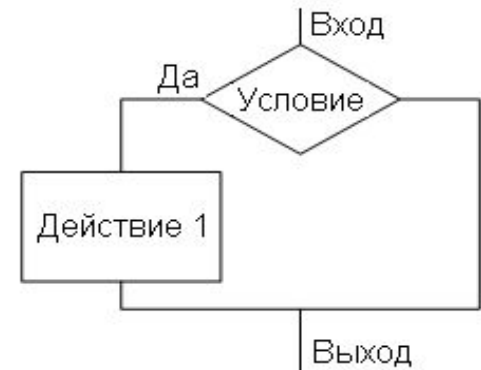
Условный оператор if...then... неполное ветвление(обход)

Синтаксис:

if <логическое выражение> then <оператор1>;

Семантика:

1. Вычисляется значение <логического выражения>
2. Если <логическое выражение> оказывается истинным (TRUE), то выполняется оператор1, иначе переход к следующему оператору.



Пример:

**И6. Даны два числа.
Вывести большее из них**

```
program a_max_b;  
var a,b,max : real;  
begin  
  writeln ('введите числа a и b');  
  readln (a,b);  
  writeln ('a=',a,'b=',b);  
  max:=b;  
  if a>b then max:=a;  
  writeln ('max=',max);  
end.
```

- **ЛОГИЧЕСКОЕ ВЫРАЖЕНИЕ** - ЭТО выражение, которое принимает значение TRUE(истина) или FALSE(ложь), и включает в себя арифметические выражения, связанные операциями отношения и логическими операциями.
- **ЛОГИЧЕСКИЕ ОПЕРАЦИИ:**
- **NOT** - НЕ; **AND** - И; **OR** - ИЛИ

ОПЕРАЦИИ ОТНОШЕНИЯ:

> больше ($a > 5$)

>= больше или равно ($a >= b$)

< меньше ($x < 6$)

<= меньше или равно ($y <= 0$)

= равно ($r = 10$)

<> неравно ($a <> d$)

- возвращают значение типа **boolean** (true или false)

Выражение отношения – два выражения, связанных операцией отношения.

Семантика:

- Вычисляются значения выражений;
- Проверяется истинность отношения

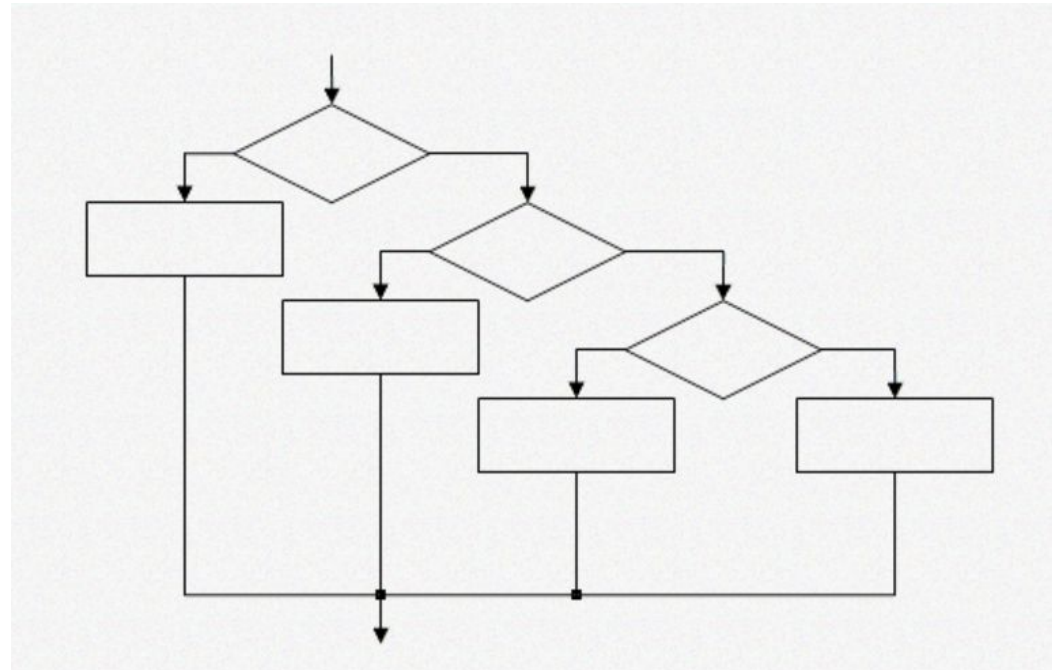
Например:

a:=1; b:=2;c:=3;

2*a>=b+c

соответствует $2 \geq 5$ значение **FALSE**

Из нескольких ветвлений можно сконструировать структуру «**выбор**» (**множественное ветвление**), которая будет выбирать не из двух, а из большего количества вариантов действий исполнителя, зависящих от нескольких условий. Существенно, что выполняется только одна ветвь - в такой структуре важное значение приобретает порядок следования условий: если выполняются несколько условий, то сработает только одно из них - первое сверху.



Условный оператор

- Если в ветвях необходимо выполнить несколько операторов то используется **составной оператор (блок - begin...end)**.

Пример

If9. Даны две переменные **A, B**. Поместить в **A** меньшее из значений, а в **B** — большее.

Вывести новые значения переменных **A** и **B**.

```
program aib;  
var a,b,v:real;  
begin  
  writeln ('введите числа a и b');  
  readln(a,b);  
  writeln ('a=',a,' b=',b);  
  if a>b then begin v:=a; a:=b; b:=v; end;  
  writeln ('a=',a,' b=',b);  
end.
```



Домашняя работа

Решение задач на ветвление

- Дано целое число. Если оно положительное, то удвоить его; если отрицательное, то вычесть из него 2; если нулевое, то заменить его на 10. Вывести полученное число .