

ПОВТОРИМ

Структура программы на Паскале



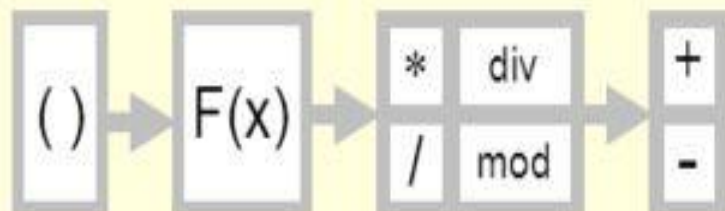
Арифметические операции

+	сложение
-	вычитание
*	умножение
/	деление
div	деление нацело
mod	Остаток от деления

Функции F (x)

ABS (x)	x
SQR (x)	x^2
SQRT (x)	\sqrt{x}
EXP (x)	e^x
LN (x)	$\ln x$
SIN (x)	$\sin x$
COS (x)	$\cos x$

Приоритеты выполнения операций



```
PROGRAM ИМЯ ПРОГРАММЫ ;
```

```
VAR список однотипных переменных : ТИП ;
```

```
BEGIN ОПЕРАТОРЫ END .
```

```
PROGRAM ИМЯ ПРОГРАММЫ ;
```

```
VAR список однотипных переменных : INTEGER ;
```

```
список однотипных переменных : REAL ;
```

```
BEGIN
```

```
READ ( список ввода ) ;
```

```
READLN ( список ввода ) ;
```

```
переменная := арифметическое выражение ;
```

```
WRITE ( список вывода ) ;
```

```
WRITELN ( список вывода )
```

```
END .
```

Структура Pascal-программы

```
program First;  
  const Pi = 3.14;  
  var r: real;  
      S,C: real;  
begin  
  write('Введите радиус окружности: ');  
  readln(r);  
  S := Pi*r*r;  
  C := 2*Pi*r;  
  writeln('Длина окружности равна C=',C);  
  writeln('Площадь круга равна S=',S);  
end.
```

} **Имя программы Заголовок**

} **Секция описаний**

} **Начало блока операторов**

} **Операторы (блок)**

} **Конец блока операторов**

Внимание!!! Операторы языка отделяются знаком ;

Внимание!!! Программа заканчивается end.

Зарезервированные слова **begin** и **end** служат **операторными скобками.**

ПРОГРАММЫ С УСЛОВНЫМ ОПЕРАТОРОМ

Разветвляющиеся алгоритмы

Особенность: действия исполнителя зависят от некоторых условий {если (*if*) ... то (*then*).....иначе (*else*) ...}.

Алгоритмы, в которых последовательность шагов зависит от выполнения некоторых условий, называются
разветвляющимися.

Условие - это выражение, стоящее после слова *if*



РАЗНОВИДНОСТЬ УСЛОВИЙ

Простые условия (отношения)

< <= > >= = <>



равно

не равно

Сложное (составное) условие – это условие, состоящее из нескольких простых условий (отношений), связанных с помощью логических операций:

- `not` – НЕ (отрицание, инверсия)
- `and` – И (одновременное выполнение условий)
- `or` – ИЛИ (выполнение хотя бы одного из условий)

Условный оператор

```
if <условие> then begin
    {что делать, если условие верно}
end
else begin
    {что делать, если условие неверно}
end;
```

Особенности:

- перед *else* **НЕ** ставится точка с запятой
- вторая часть (*else ...*) может отсутствовать (неполная форма)

Если используется оператор *else* – полная структура

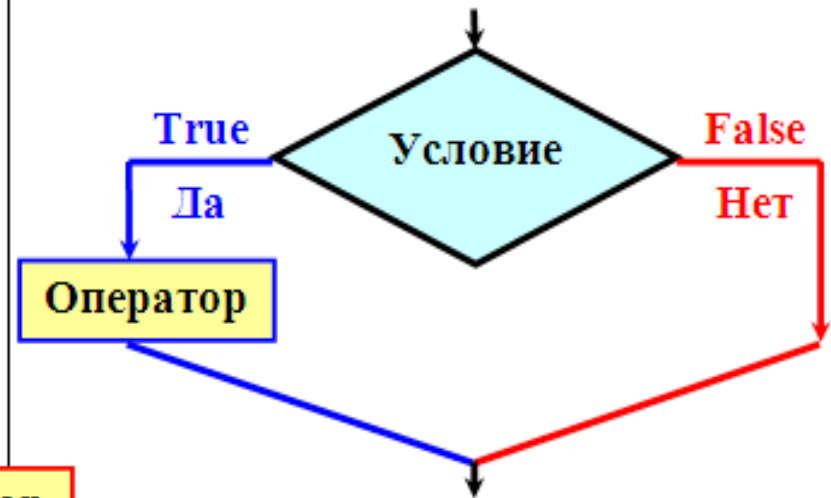
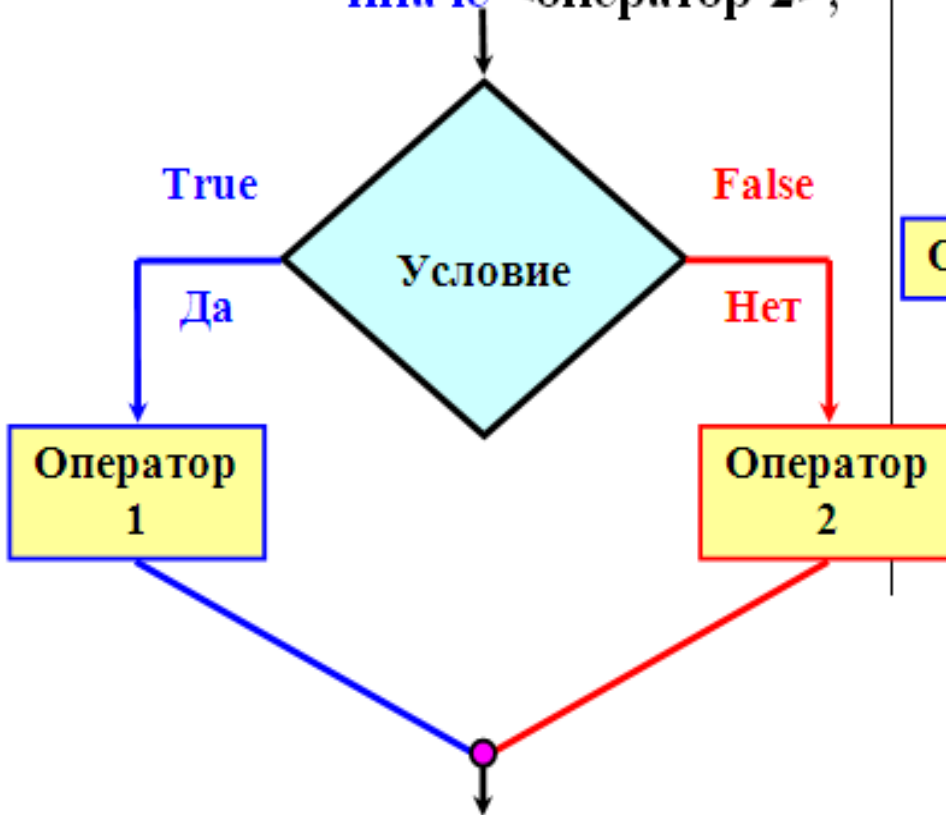


Условный оператор



If <условие> then <оператор 1>
 else <оператор 2>;
 если <условие> то <оператор 1>
 иначе <оператор 2>;

If <условие> then <оператор>;
 если <условие> то <оператор>;



Условия записываются, как правило, в виде логических выражений. Простой формой логического выражения является **операция отношения**.

В Паскале допускаются все виды отношений:

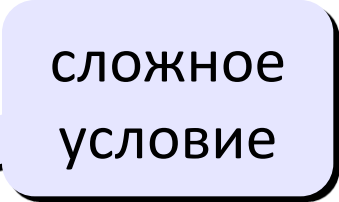
- > - больше;
- < - меньше;
- = - равно;
- <= - меньше или равно;
- >= - больше или равно;
- <> - не равно.

Пример программы

Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

Особенность: надо проверить, выполняются ли два условия одновременно.

```
program qq;  
var x: integer;  
begin  
  writeln('Введите возраст');  
  read ( x );  
  if (x >= 25) and (x <= 40) then  
    writeln ('Подходит')  
  else writeln ('Не подходит')  
end.
```

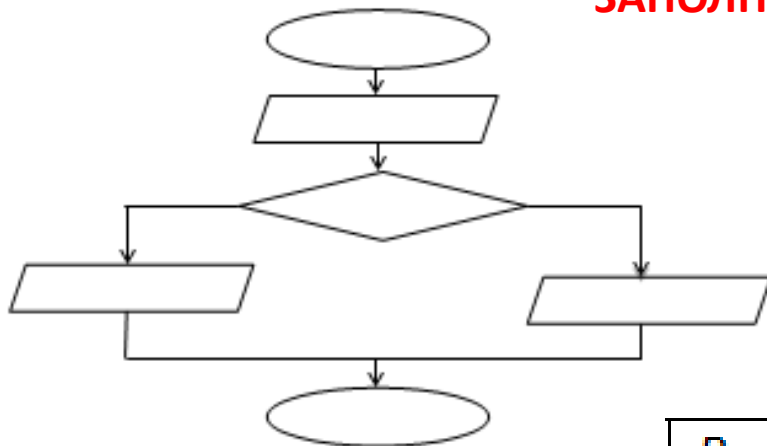


сложное
условие

СОСТАВИТЬ ПРОГРАММУ К ЗАДАЧЕ:

В баскетбольную секцию принимаются юноши в возрасте 13 лет и не ниже 165 см.
Составить программу зачисления юношей в секцию.

ЗАПОЛНИТЬ БЛОК-СХЕМУ К ЗАДАЧЕ!!!



!!! ПОДСКАЗКА. Заполните пропуски!!!

```
Program Z3;  
Var _____ : integer;  
Begin  
write ('_____'); readln (____);  
write ('_____'); readln (____);  
if (____) and (____) then  
writeln('_____')  
else writeln('_____');  
end.
```