

Подготовка к контрольной работе

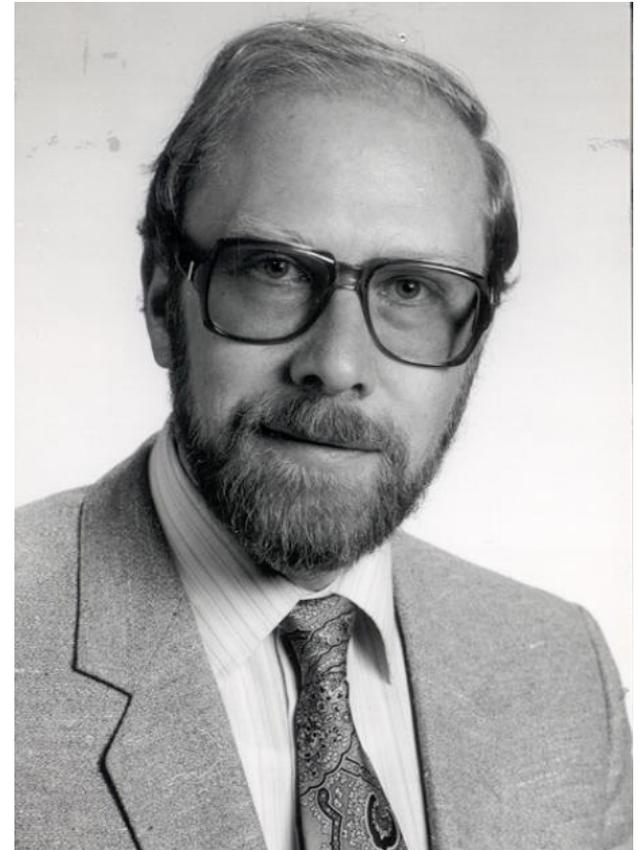


«НАЧАЛА ПРОГРАММИРОВАНИЯ»
8 КЛАСС

Pascal (Паскаль)

- ЯЗЫК
программирования
общего назначения.

Создан в 1970 году
специалистами
Технического
Университета в Цюрихе
под руководством профессора
Никлауса Вирта



Основные правила грамматики

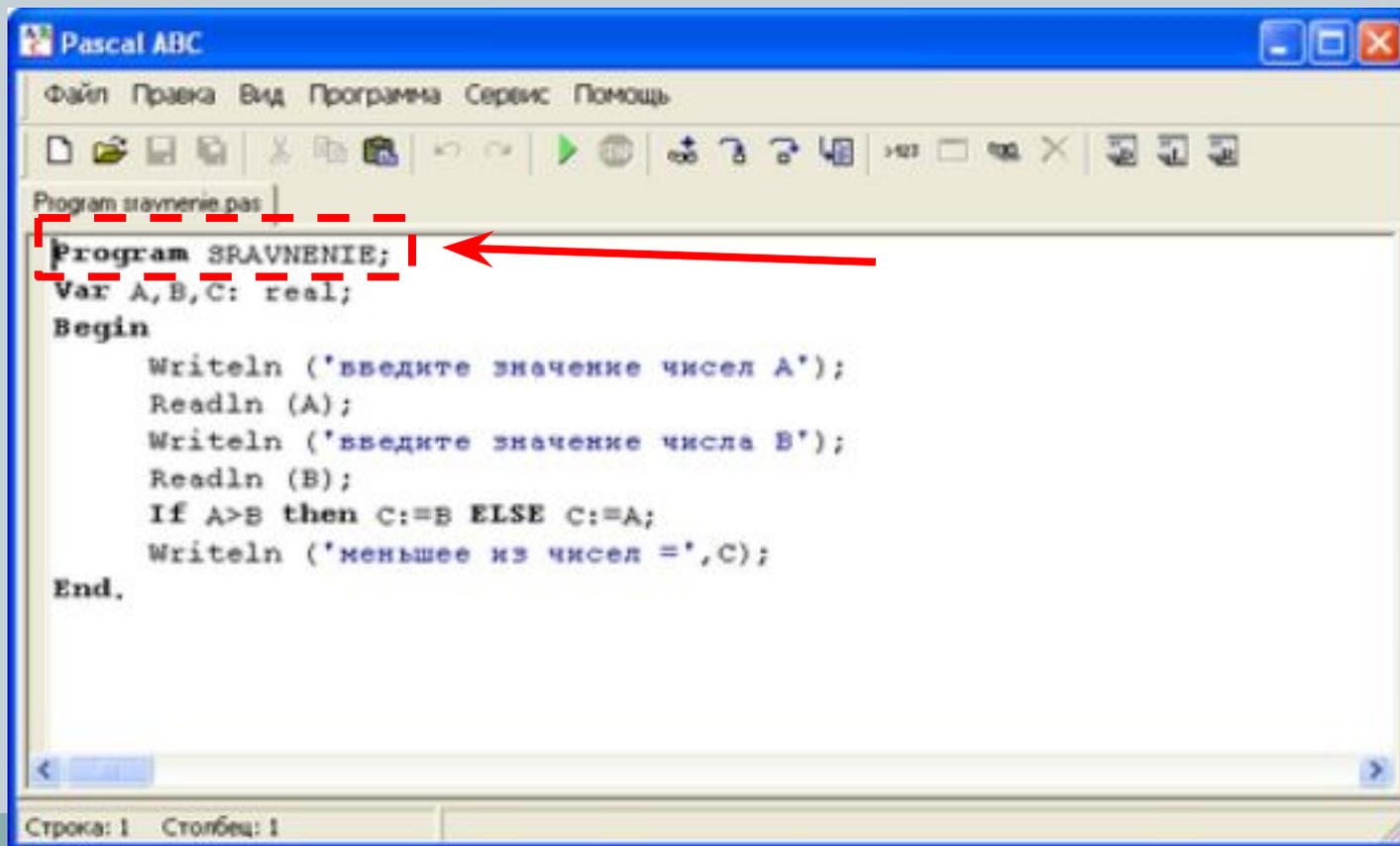


Текст программы состоит из трех составных частей:

1. заголовок,
2. описания,
3. операторы

Заголовок

В Turbo Pascal заголовок имеет декоративное значение и использовать его не обязательно.



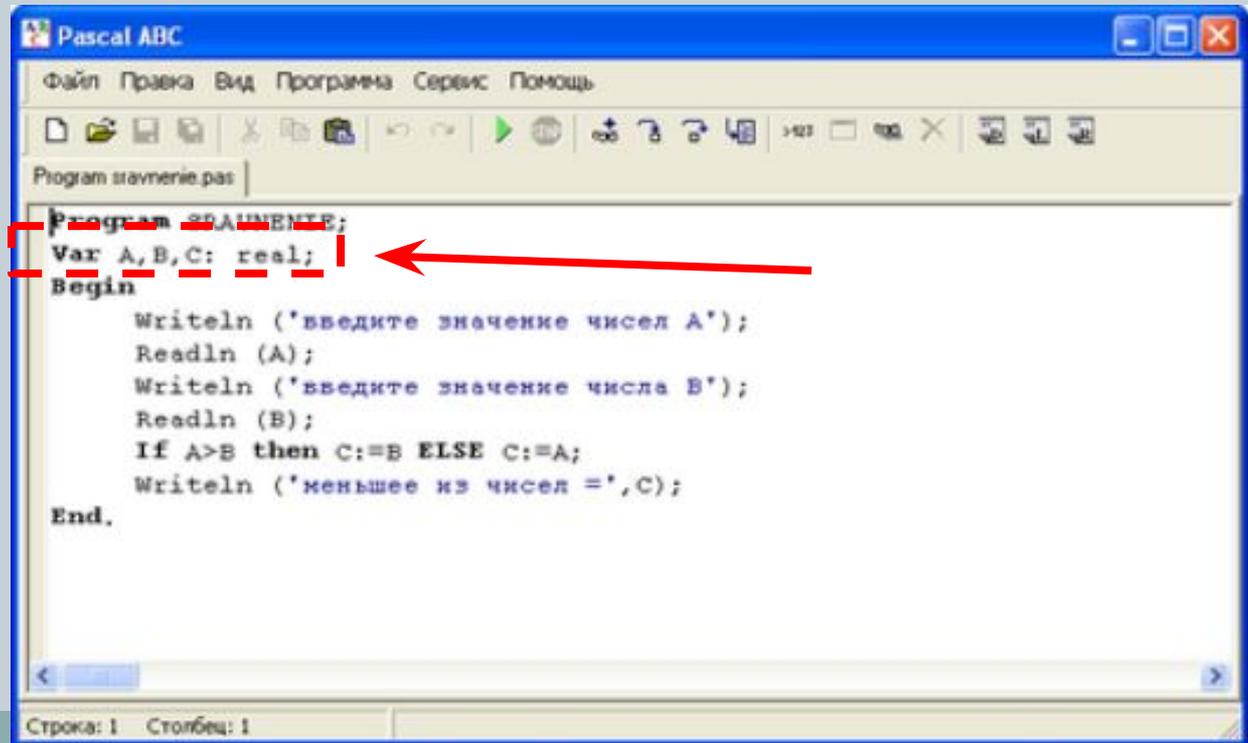
The screenshot shows the Turbo Pascal IDE window titled "Pascal ABC". The menu bar includes "Файл", "Правка", "Вид", "Программа", "Сервис", and "Помощь". The toolbar contains various icons for file operations and execution. The main text area displays the following Pascal code:

```
Program spravnenie.pas |
Program SPAVNIENIE;
Var A, B, C: real;
Begin
    Writeln ('введите значение чисел A');
    Readln (A);
    Writeln ('введите значение числа B');
    Readln (B);
    If A>B then C:=B ELSE C:=A;
    Writeln ('меньшее из чисел =', C);
End.
```

The line `Program SPAVNIENIE;` is highlighted with a red dashed box, and a red arrow points to it from the right. The status bar at the bottom indicates "Строка: 1" and "Столбец: 1".

Описание

это символьные конструкции для описания свойств объектов программы. Все объекты программы надо обязательно указать в этой части программы.

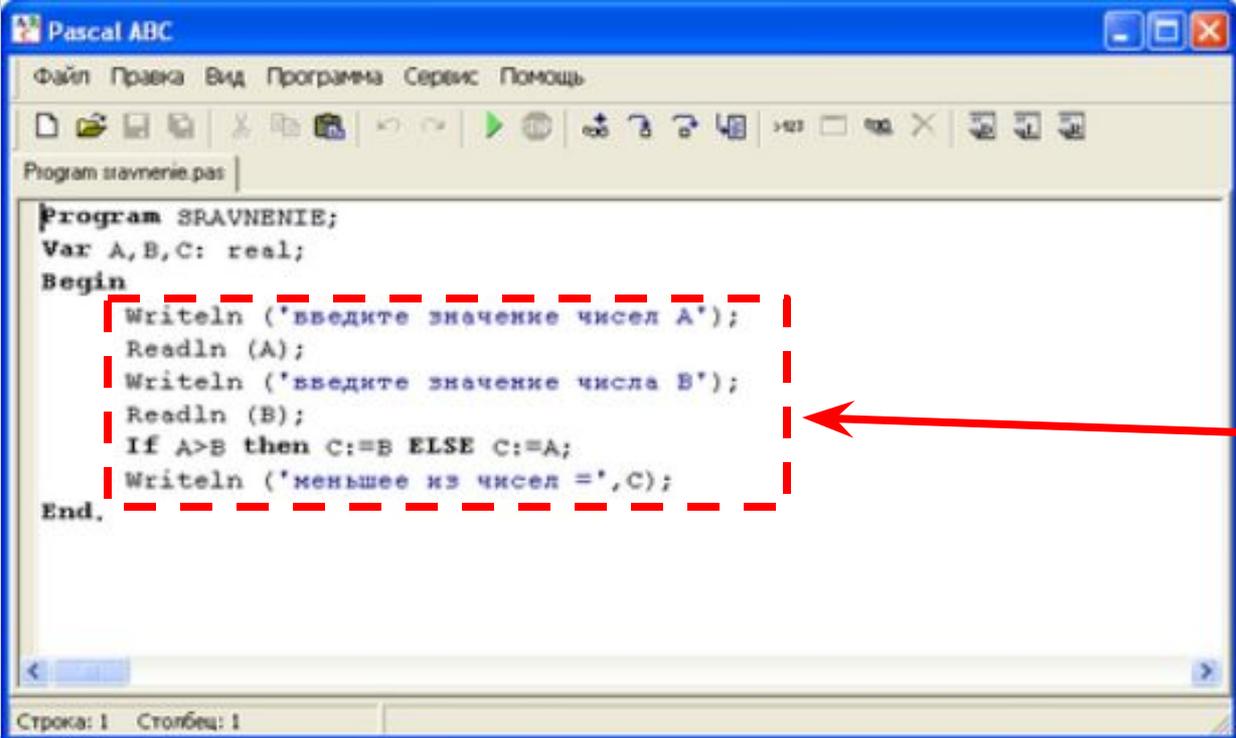


```
Program zravnenie;  
Var A,B,C: real;  
Begin  
    Writeln ('введите значение чисел A');  
    Readln (A);  
    Writeln ('введите значение числа B');  
    Readln (B);  
    If A>B then C:=B ELSE C:=A;  
    Writeln ('меньшее из чисел =',C);  
End.
```

The screenshot shows a window titled "Pascal ABC" with a menu bar (Файл, Правка, Вид, Программа, Сервис, Помощь) and a toolbar. The main text area contains the Pascal code shown above. A red dashed box highlights the first two lines of code, and a red arrow points to the line "Var A,B,C: real;". The status bar at the bottom indicates "Строка: 1 Столбец: 1".

Операторы

это символьные конструкции для описания действий над объектами программы. Операторы должны находиться между словами begin и end.



```
Program sraivnenie.pas  
  
Program SRAIVNENIE;  
Var A,B,C: real;  
Begin  
  Writeln ('введите значение чисел A');  
  Readln (A);  
  Writeln ('введите значение числа B');  
  Readln (B);  
  If A>B then C:=B ELSE C:=A;  
  Writeln ('меньшее из чисел =',C);  
End.
```

Строка: 1 Столбец: 1

Переменные



Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

● **Типы переменных:**

- integer { целая }
- real { вещественная }
- char { символьная }
- string { символьная строка }
- boolean { логическая }

Изменение значения переменной



Оператор – это команда языка программирования высокого уровня.

Оператор присваивания служит для изменения значения переменной.

```
program qq;  
var a, b: integer;  
begin  
    a := 5;  
    b := a + 2;  
    a := (a + 2) * (b - 3);  
end.
```

Оператор ввода



```
read ( a ) ;      { ввод значения  
                  переменной a }
```

```
read ( a , b ) ; { ввод значений  
                  переменных a и b }
```

Как вводить два числа?

через пробел:

25 30

через *Enter*:

25

30

Оператор вывода



```
write ( a );      { вывод значения  
                  переменной a }
```

```
writeln ( a );   { вывод значения  
                  переменной a и переход  
                  на новую строку }
```

```
writeln ( 'Привет!' ); { вывод текста }
```

```
writeln ( 'Ответ: ', c ); { вывод  
                          текста и значения переменной c }
```

```
writeln ( a, '+', b, '=', c );
```

Условный оператор

Полная форма условного оператора:

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

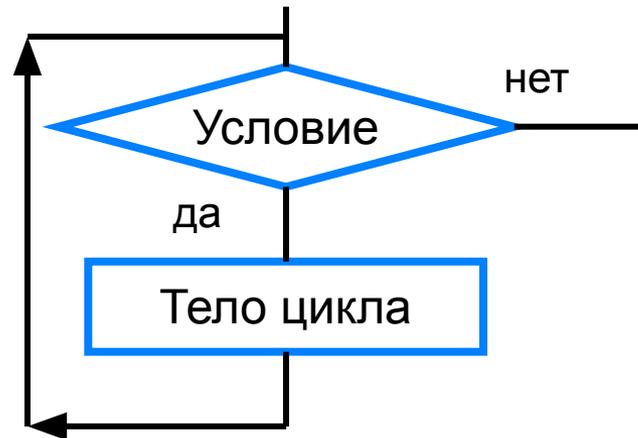
Неполная форма условного оператора:

if <условие> **then** <оператор>

! Перед **else** знак «;» не ставится.



Программирование циклов с заданным условием продолжения работы



Общий вид оператора:

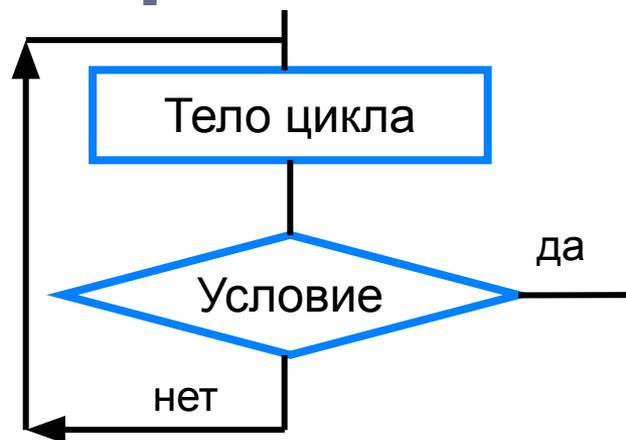
```
while <условие> do <оператор>
```

Здесь:

<условие> - логическое выражение;
пока оно истинно, выполняется тело цикла;

<оператор> - простой или составной оператор,
с помощью которого записано тело цикла.

Программирование циклов с заданным условием окончания работы



Общий вид оператора:

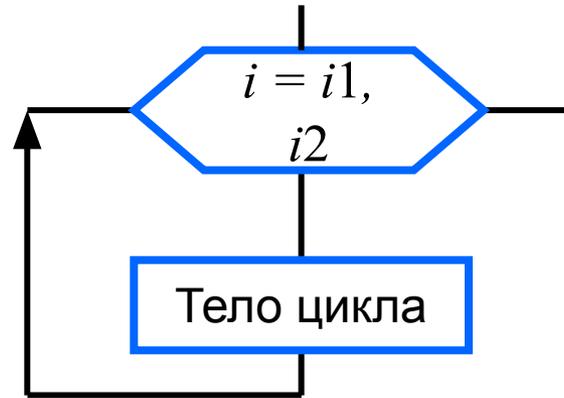
repeat <оператор1; оператор2; ...; > **until** <условие>

Здесь:

<оператор1>; <оператор2>; ... - операторы, образующие тело цикла;

<условие> - логическое выражение; если оно ложно, то выполняется тело цикла.

Программирование циклов с заданным числом повторений



Общий вид оператора:

```
for <параметр>:=<начальное_значение>  
to <конечное_значение> do <оператор>
```

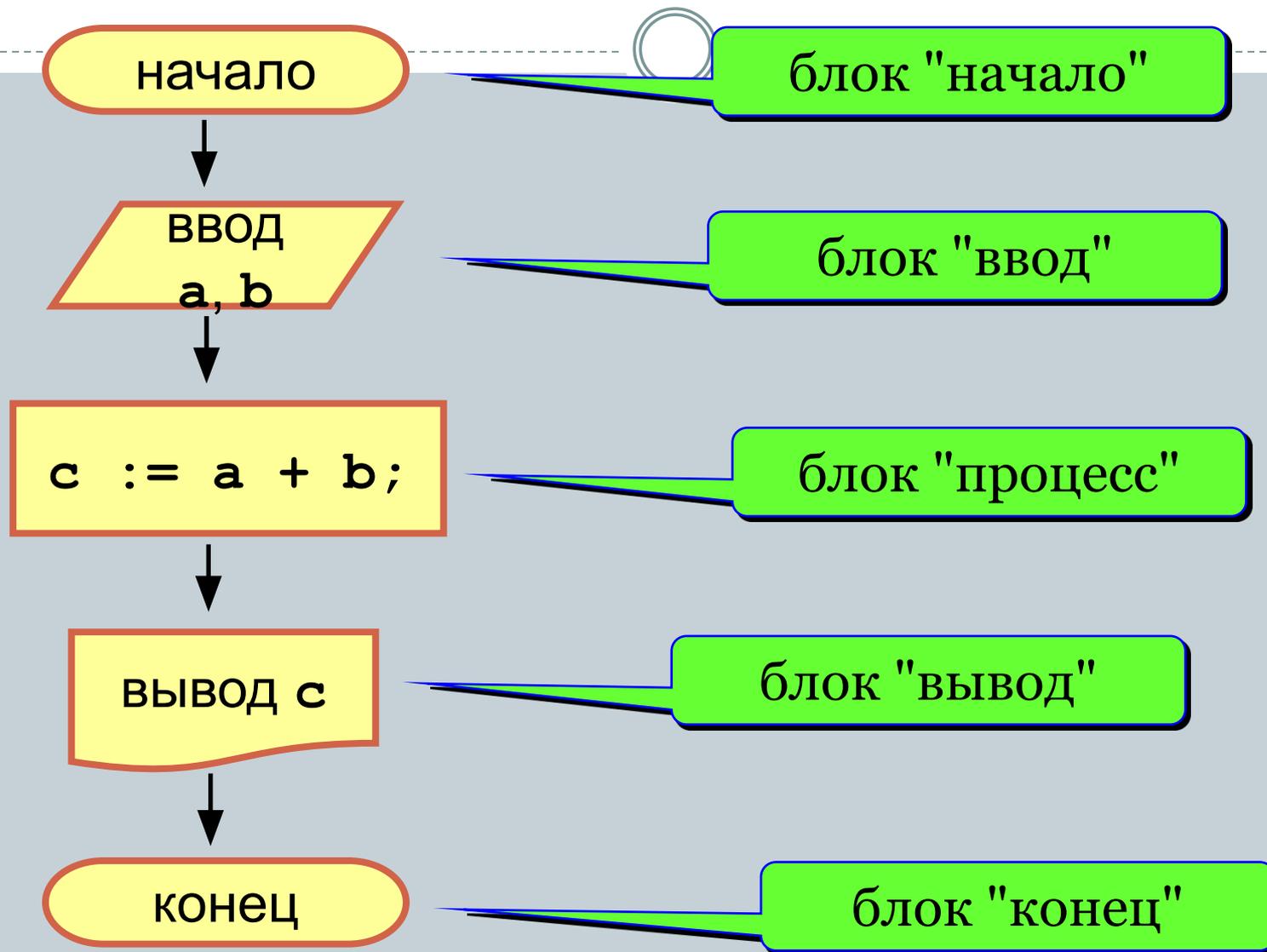
Здесь:

<параметр> - переменная целого типа;

После каждого выполнения тела цикла происходит увеличение на единицу параметра цикла; условие выхода из цикла - превышение параметром конечного значения.

<начальное значение> и <конечное значение> - выражения того же типа, что и параметр;
<оператор> - простой или составной оператор - тело цикла.

Блок-схема линейного алгоритма



Решение задач

