

Лабораторная работа №04 по дисциплине "Программирование"

ТЕМА: Условный оператор (инструкция).

- Логические выражения в условных операторах (инструкциях).
- Полная и сокращенная форма условного оператора (инструкции).
- Составные операторы (инструкции).
- Метки. Оператор (инструкция) перехода.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПОК

1. В Вашей папке создайте папку с именем П_лр04, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.
2. При выполнении задания создавайте в папке П_лр04 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составляемой программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составляемой программой).
3. ПРИ ВЫПОЛНЕНИИ СЛЕДУЮЩИХ ЛАБОРАТОРНЫХ РАБОТ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ

4. Логическое выражение может состоять из выражений, над которыми выполняется одна из следующих операций сравнения:
 - = равно;
 - < меньше;
 - > больше;
 - <= меньше или равно;
 - >= больше или равно;
 - <> не равно.Операция сравнения производится после вычисления значений сравниваемых выражений. Результатом операции является значение True (Истина), если отношение, устанавливаемое операцией сравнения, выполняется, или значение False (Ложь), если это отношение не выполняется.
5. Примеры логических выражений:
 - x>y
 - a<=b
 - (x+1)<>abs(y)

ОПЕРАТОРЫ (ИНСТРУКЦИИ) ПЕРЕДАЧИ УПРАВЛЕНИЯ

6. ОПЕРАТОР ПЕРЕХОДА
goto <Метка>;
предназначен для изменения обычного порядка выполнения операторов программы, когда после выполнения некоторого оператора требуется передать управление не следующему по порядку, а другому оператору.
Оператор, которому передается управление, должен быть помечен меткой, стоящей перед оператором и отделенной от него двоеточием.
Метка может быть идентификатором или целым числом от 0 до 9999.
Меткой может быть снабжен и пустой оператор, представляющий собой точку с запятой.
Все метки должны быть предварительно объявлены в разделе объявления меток.

Раздел объявления меток находится в той же области, что и раздел объявления переменных.

7. Примеры использования ОПЕРАТОРА ПЕРЕХОДА:

```
...
Label m1,m2,100;
...
goto m1;
...
m1:<Оператор>;
...
m2:<Оператор>;
...
goto m2;
...
goto 100;
...
100:<Оператор>;
...
```

8. УСЛОВНЫЙ ОПЕРАТОР

```
if <Условие> then <Оператор1> else <Оператор2>;
```

Условие представляет собой логическое выражение (см.п.4).

УСЛОВНЫЙ ОПЕРАТОР работает следующим образом:

если Условие имеет значение True то выполняется Оператор1, иначе Оператор2. Такая форма УСЛОВНОГО ОПЕРАТОРА иногда называется его ПОЛНОЙ ФОРМОЙ.

Допускается использование СОКРАЩЕННОЙ ФОРМЫ УСЛОВНОГО ОПЕРАТОРА (без **else**):

```
if <Условие> then <Оператор>;
```

В этом случае, если Условие имеет значение False, то выполняется оператор, следующий за УСЛОВНЫМ ОПЕРАТОРОМ.

9. Примеры УСЛОВНЫХ ОПЕРАТОРОВ:

```
if x>0 then x:=x+1 else x:=0;
if abs(a)<=1 then b:=1;
```

10. Для организации разветвлений по 3 и более направлениям можно использовать несколько УСЛОВНЫХ ОПЕРАТОРОВ, вложенных друг в друга, например:

```
if x>0 then
    if x>=1 then x:=x+1 else x:=0;
```

СОСТАВНОЙ ОПЕРАТОР (СОСТАВНАЯ ИНСТРУКЦИЯ)

11. СОСТАВНОЙ ОПЕРАТОР

```
begin <Оператор1>; ...; <ОператорN>; end;
```

представляет собой группу из произвольного числа операторов, отделенных друг от друга ";", ограниченную операторными скобками "begin" и "end".

Независимо от количества входящих в него операторов, СОСТАВНОЙ ОПЕРАТОР воспринимается как единое целое и может располагаться в любом месте программы, где допускается наличие оператора.

12. Пример СОСТАВНОГО ОПЕРАТОРА:

```
begin
    n:=n+1;
    x:=x/n;
    s:=s+x;
end
```

13. СОСТАВНЫЕ ОПЕРАТОРЫ могут вкладываться друг в друга с неограниченной глубиной вложенности, например:

```
begin
  ...
  begin
    ...
    begin
      ...
      end;
    ...
  end;
  ...
end
```

СОЗДАНИЕ ПРОГРАММ

ПРОГРАММА 4-1

14. Создайте консольное приложение, предназначенное для получения решения квадратного уравнения вида $ax^2+bx+c=0$.

Приложение должно использовать ПОЛНУЮ ФОРМУ УСЛОВНОГО ОПЕРАТОРА и выполнять следующие действия:

- ввод значений вещественных коэффициентов a , b и c ;
- если значение дискриминанта $D=b^2-4ac \geq 0$, то вычисление значений корней квадратного уравнения по формулам

$$x_1 = \frac{-b-D^{1/2}}{2a}, \quad x_2 = \frac{-b+D^{1/2}}{2a} \quad \text{и вывод результатов;}$$

- если значение дискриминанта $D=b^2-4ac < 0$ (что говорит об отсутствии у уравнения действительных корней), то вывод текста " $D < 0$ ".
- в строках текста программы с операторами должны также находиться комментарии на русском языке с кратким описанием действий, выполняемыми соответствующими операторами.

ПРОГРАММА 4-2

15. Создайте консольное приложение, выполняющее те же действия, что и приложение из п.14, но использующее СОКРАЩЕННУЮ ФОРМУ УСЛОВНОГО ОПЕРАТОРА и ОПЕРАТОР ПЕРЕХОДА с МЕТКОЙ (в тексте программы также должны быть комментарии).

ПРОГРАММА 4-3

16. Создайте консольное приложение, выполняющее те же действия, что и приложение из п.14, но работающее таким образом, чтобы оно после вывода результатов снова переходило бы на ввод значений коэффициентов квадратного уравнения.

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

17. Продемонстрируйте тексты созданных программ, а также их выполнение в среде программирования и независимо от нее.
