

Лабораторная работа №13 по дисциплине "Программирование"

ТЕМА: Подпрограммы. Процедуры и функции.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПОК

1. В Вашей папке создайте папку с именем П_лр13, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.
2. При выполнении задания создавайте в папке П_лр13 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.

ПОДПРОГРАММЫ

4. Подпрограмма – это группа операторов, логически законченная и специальным образом оформленная.
Подпрограмма описывается (объявляется) один раз, а обращаться к ней (вызывать ее) можно по ее имени неограниченное число раз из различных частей программы там, где требуется получить результаты работы подпрограммы.
Внутри подпрограммы можно описывать другие подпрограммы, к которым можно обращаться только из той подпрограммы, внутри которой они описаны.
При вызове подпрограммы указываются ее имя и список аргументов (фактических параметров), передаваемых подпрограмме для выполнения операторов, входящих в состав этой подпрограммы. Одна и та же подпрограмма может обрабатывать различные данные, переданные ей в качестве аргументов.
Существует много стандартных подпрограмм (например, `abs`, `sin`, `readln`, `write` и проч.), которые можно вызывать без предварительного описания.
Кроме того, программист может создавать собственные подпрограммы, которые называются пользовательскими.

ПРОЦЕДУРЫ И ФУНКЦИИ

5. Подпрограммы делятся на процедуры и функции, основное различие между которыми состоит в том, что результатом выполнения функции является некоторое значение, присвоенное ее имени, и это имя можно использовать как операнд выражения.
Примеры вызовов стандартных процедуры и функции:
`Writeln(a,b,c);`
`y:=sin(x)+1;`

СОЗДАНИЕ ПРОЦЕДУР И ФУНКЦИЙ

6. Описание (объявление) процедуры располагается в программе там же, где располагаются описания (объявления) переменных, констант, меток. Описание процедуры состоит из заголовка и блока (тела процедуры). Заголовок имеет формат
- ```
Procedure <Имя> [(Формальные параметры)];
```
- и состоит из ключевого слова **procedure**, имени процедуры и необязательного списка формальных параметров в круглых скобках с указанием типа каждого параметра, например:

```
Procedure x2y2(var s:real;x,y:real);
```

В списке формальных параметров после указателя типа ставится точка с запятой, если же тип указан сразу для нескольких параметров, то их идентификаторы отделяются друг от друга запятыми. Блок процедуры начинается с **begin** и заканчивается **end**;

7. Для обращения к процедуре (для ее вызова) используется оператор вызова процедуры, состоящий из имени процедуры и списка разделенных запятыми аргументов в круглых скобках, при этом каждому формальному параметру, указанному в заголовке процедуры, должен соответствовать аргумент того же типа.

Если в описании процедуры формальный параметр указан с описателем **var**, то при обращении к процедуре в качестве фактического параметра передается адрес аргумента в памяти компьютера. Значение по такому адресу может быть изменено при выполнении процедуры и по такому адресу может быть помещен, например, результат выполнения процедуры. В качестве такого аргумента может быть переменная и не может быть выражение или константа.

Если в описании процедуры формальный параметр указан без описателя **var**, то при обращении к процедуре в качестве фактического параметра передается значение аргумента в памяти компьютера. Это значение не может быть изменено при выполнении процедуры. В качестве такого аргумента может быть переменная, выражение или константа.

Пример программы с описанием и вызовом процедуры:

```
program PP;

{$APPTYPE CONSOLE}

uses
 SysUtils;
var w,z,t,u:real;
Procedure x2y2(var s:real;x,y:real); //Описание процедуры
begin
 s:=x*x+y*y;
end;
begin
 readln(t,u);
 x2y2(z,t,u); //Вызов процедуры
 w:=sin(t/z)+cos(u/z);
 writeln(w);
 readln;
end.
```

8. Описание (объявление) функции располагается в программе там же, где располагаются описания (объявления) переменных, меток, констант, описания процедур.
- Описание функции состоит из заголовка и блока (тела функции). Заголовок имеет формат

```
Function <Имя> [(Формальные параметры)]:<Тип результата>;
```

и состоит из ключевого слова **function**, имени функции, необязательного

списка формальных параметров в круглых скобках с указанием типа каждого параметра и типа возвращаемого функцией значения.

Блок функции начинается с **begin** и заканчивается **end**;

В блоке функции должен быть хотя бы один оператор присваивания с именем функции в левой части этого оператора. Последний выполненный из таких операторов и определяет значение, возвращаемое функцией. В этих операторах допускается использование вместо имени функции переменной *Result*, причем в отличие от имени функции переменную *Result* можно использовать в выражениях блока функции. Использование этой переменной дает возможность получить в любой момент доступ к текущему значению функции.

9. Для обращения к функции (для ее вызова) используется ее имя со списком разделенных запятыми аргументов в круглых скобках, при этом каждому формальному параметру, указанному в заголовке функции, должен соответствовать аргумент того же типа. В отличие от процедуры функция может входить в выражения в качестве операнда. При обращении к функции передача адреса или значения аргумента происходит так же, как и при обращении к процедуре.

Пример программы с описанием и вызовом функции:

**program** F;

```
{$APPTYPE CONSOLE}
```

**uses**

```
 SysUtils;
```

**var** w,t,u:real;

**Function** x2y2(x,y:real):real; //Описание функции

**begin**

```
 x2y2:=x*x+y*y;
```

**end**;

**begin**

```
 readln(t,u);
```

```
 w:=sin(t/x2y2(t,u))+cos(u/x2y2(t,u)); //Вызовы функции
```

```
 writeln(w);
```

```
 readln;
```

**end.**

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ НЕКОТОРЫХ МАТЕМАТИЧЕСКИХ ФУНКЦИЙ

10. Некоторые математические функции описаны только в модуле *Math* и без подключения этого модуля не могут быть использованы. Для подключения модуля *Math* его имя должно быть добавлено в операторе **uses**, например,

```
uses SysUtils, Math;
```

Примерами математических функций, использование которых требует подключения указанного модуля, являются функции *IntPower* и *Power*.

Функция *IntPower* предназначена для возведения вещественного числа в целую степень (первый аргумент – возводимое число и второй аргумент – степень, могут принимать любые значения, но нулевое значение не может возводиться в отрицательную степень).

Функция *Power* предназначена для возведения вещественного числа в вещественную степень (первый аргумент – возводимое число и второй аргумент – степень, могут принимать любые значения, но нулевое значение не может возводиться в отрицательную степень и отрицательное значение не может возводиться в нецелую степень).

## СОЗДАНИЕ ПРОГРАММ

-----

ВНИМАНИЕ! Все создаваемые здесь программы должны быть "зациклены", то есть после вывода результатов следует переходить снова на ввод данных.

### ПРОГРАММА 13-1 и ПРОГРАММА 13-2

-----

11. Создайте консольное приложение, при выполнении которого происходит:

- ввод трех вещественных чисел X, Y и Z;
- вызов процедуры, описанной в этой же программе и вычисляющей значение

$$S=X^2+Y^2+Z^2$$

- вычисление с использованием вызова этой процедуры и печать в отдельной строке результата вычислений по формуле

$$D = \frac{e^{-S}+e^S}{S} \cdot (\sin^2(X/S^{0,5})+\sin^2(Y/S^{0,5})+\sin^2(Z/S^{0,5})).$$

12. Создайте консольное приложение, при выполнении которого происходит:

- ввод трех вещественных чисел X, Y и Z;
- вычисление и печать в отдельной строке результата вычислений по формуле

$$D = \frac{e^{-S(X,Y,Z)}+e^{S(X,Y,Z)}}{S(X,Y,Z)} \cdot (\sin^2(X/S^{0,5}(X,Y,Z))+\sin^2(Y/S^{0,5}(X,Y,Z))+\sin^2(Z/S^{0,5}(X,Y,Z))).$$

с использованием вместо S вызова функции, описанной в этой же программе и вычисляющей значение  $S(X,Y,Z) = X^2+Y^2+Z^2$

### ПРОГРАММА 13-3

-----

13. Создайте консольное приложение (с подключенным модулем Math), при выполнении которого происходит:

- ввод данных;
- выполнение возведения вещественного числа X в вещественную степень Y (значения числа и степени должны быть введены) с использованием математической функции Power(X,Y) - см. п.10;
- вывод результата или сообщения о причине невозможности возведения в степень (для определения целого или нецелого значения степени можно использовать функцию преобразования Int - см., например, файл П\_лр02)).

## ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

-----

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

---