Лабораторная работа №07 по дисциплине
"Вычислительная техника и программирование"

ТЕМА: Подпрограммы. Процедуры и функции.
---- Процедуры и функции для работы со строками.
Создание процедур и функций.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПОК

1. В Вашей папке создайте папку с именем ВТП_лр07, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.

2. При выполнении задания создавайте в папке ВТП_лр07 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.

КОНСТАНТЫ И ИХ ИСПОЛЬЗОВАНИЕ

4. Именованные константы могут быть использованы в программе вместо значений этих констант. Раздел описания (объявления) констант может находиться в программе в том же месте, где и раздел описания (объявления) переменных.
Примеры объявления констант:
```
Const w1='WORD'; otl='5'; e=2.71828;
```
Имя константы отделяется от выражения **знаком равенства**.
Тип константы автоматически распознается на основании типа выражения справа от **знака равенства**.
Именованные константы могут быть использованы при описании (объявлении) массивов, например:
```
Const N=20;
Var z: Array [1..N] of Real;
```

5. В разделе описания констант могут быть описаны (объявлены) также переменные, которым требуется присвоить значения до начала работы программы.
Описание такой переменной отличается от описания именованной константы наличием типа, например:
```
Const ww: real = 10.5;
```

ПОДПРОГРАММЫ

6. 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, водящих 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

7. 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;
   ```

НЕКОТОРЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ СО СТРОКАМИ

8. Для реализации действий со строками и фрагментами строк предусмотрены стандартные процедуры и функции.

ВНИМАНИЕ! Фрагмент строки часто называют подстрокой.

ВНИМАНИЕ! Фраза "функция возвращает" практически всегда используется вместо фразы "результатом выполнения функции является".

Процедура или функция с именем и примером перечня аргументов	**Описание**
Функция Concat(s1,s2,...,sN)	Возвращает строку, являющуюся слиянием строк s1,s2,...,sN. Действие функции аналогично операции конкатенации.
Функция Copy(s,start,len)	Возвращает подстроку длиной len, начинающуюся с позиции start строки s.
Процедура Delete(s,start,len)	Удаляет из строки s, начиная с позиции start, подстроку длиной len.
Процедура Insert(subs,s,start)	Вставляет подстроку subs в строку s, начиная с позиции start.
Функция Length(s)	Возвращает фактическую длину строки.
Функция Pos(subs,s)	Ищет вхождение подстроки subs в строку s и возвращает номер первого символа subs в s или нуль, если subs не содержится в s.
Процедура Str(x,s)	Преобразует числовое значение x в строку s, при этом для x может быть задан формат, как в процедурах Write и Writeln, например Str(x:8:2,s).
Процедура Val(s,x,errcode)	Преобразует строку s, содержащую символьное представление числа, в значение числовой переменной x. В случае успешного преобразования значение переменной errcode равно нулю, при обнаружении ошибки в символьном представлении числа значение errcode будет содержать номер позиции первого ошибочного символа, а значение x не определено.

Функция IntToStr(n)	Преобразует целочисленное значение n в строку
Функция StrToInt(s)	Преобразует строку s в целое число
Функция FloatToStr(x)	Преобразует вещественное значение x в строку
Функция StrToFloat(s)	Преобразует строку s в вещественное число
Функция AnsiUpperCase(s)	Возвращает строку s, преобразованную к верхнему регистру.
Функция AnsiLowerCase(s)	Возвращает строку s, преобразованную к нижнему регистру.
Функция Trim(s)	Возвращает строку s с удаленными из нее пробелами и управляющими символами в начале и в конце строки
Функция TrimLeft(s)	Возвращает строку s с удаленными из нее пробелами и управляющими символами в начале строки
Функция TrimRight(s)	Возвращает строку s с удаленными из нее пробелами и управляющими символами в конце строки

Примеры выражений с использованием функций для работы со строками:

```
'Summ '+ FloatToStr(x)
20+StrToInt(s)
```

СОЗДАНИЕ ПРОЦЕДУР И ФУНКЦИЙ

---

9. 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**;

10. 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.
```

11. 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.
12. 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.
```

СОЗДАНИЕ ПРОГРАММ

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

ПРОГРАММА 7-1 и ПРОГРАММА 7-2

13. Создайте консольное приложение, при выполнении которого происходит то же, что и при выполнении ПРОГРАММЫ 5-1 без поиска минимального и максимального значений и их индексов, но в объявлении массивов использованы идентификаторы именованных констант для указания значений границ индексов массивов, и, кроме того, идентификаторы этих же именованных констант должны быть использованы в выражении для вычисления среднего значения элементов массива.

14. Создайте консольное приложение, при выполнении которого происходит то же, что и при выполнении ПРОГРАММЫ 5-2, но в объявлении массивов использованы идентификаторы именованных констант для указания значений границ индексов массивов.

ПРОГРАММА 7-3

15. Создайте консольное приложение, при выполнении которого происходит ввод на английском языке в отдельных строках фамилии, имени и отчества, начинающихся с больших букв:
Familia
Imia
Otchestvo
и затем происходит формирование и вывод строки следующего вида:
Familia I.O.

ПРОГРАММА 7-4 и ПРОГРАММА 7-5

16. Создайте консольное приложение, при выполнении которого происходит:
а) ввод трех вещественных чисел X, Y и Z;
б) вызов процедуры, описанной в этой же программе и вычисляющей значение
$S=X^2+Y^2+Z^2$
б) вычисление с использованием вызова этой процедуры и печать в отдельной строке результата вычислений по формуле

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

17. Создайте консольное приложение, при выполнении которого происходит:
а) ввод трех вещественных чисел X, Y и Z;
б) 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 $S(X,Y,Z) = X^2+Y^2+Z^2$

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ
---

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