

Уважаемые студенты групп!

Вашему вниманию представлена лекция на тему «ОРГАНИЗАЦИЯ ПРОГРАММЫ, СОДЕРЖАЩИЕ ПОДПРОГРАММЫ, ПРОЦЕДУРЫ - ФУНКЦИИ»

Задание

1. Прочитать внимательно лекцию.
 2. Законспектировать лекцию в рабочую тетрадь не менее 3-5 страницы рукописного текста. В конспекте лекции обязательно должно быть приведены примеры.
 3. Решить приведенные в лекции в контрольных вопросах задачи.
 4. Дата предоставления фотоотчета лекции до 07.04.2023.
- С уважением Ганзенко Ирина Владимировна
- !!! Если возникнут вопросы обращаться по телефону, 0721134803 (вацап), +79591134803 (телеграмм)

disobuch.ganzenko2020@mail.ru

ОРГАНИЗАЦИЯ ПРОГРАММЫ, СОДЕРЖАЩИЕ ПОДПРОГРАММЫ, ПРОЦЕДУРЫ - ФУНКЦИИ

План

- 1 Теоретические положения
2. Область действия имен
3. Пример оформления программы содержащий процедуру:
- 4 Контрольные вопросы

1 Теоретические положения

Известно, что функции пользователя, как и процедуры, представляют собой относительно самостоятельные фрагменты Pascal-программы, оформленные особым образом, и имеют свои имена. Упоминание этого имени в тексте программы называется вызовом функции пользователя.

Разница между функцией и процедурой заключается в том, что функция возвращает **результат**. Результатом выполнения операторов, образующих тело функции, всегда является единственным значением того или иного типа, поэтому обращение к функции можно использовать в арифметических выражениях наряду с переменными и константами. То есть вызов функции пользователя выполняется простой записью его имени в арифметическом выражении.

Функция пользователя, как и процедура, представляет собой инструмент, с помощью которого любая Pascal-программа может быть

разделена на ряд независимых друг от друга частей. Такое разбиение необходимо по двум причинам. Во-первых, это средство экономии памяти - каждая функция пользователя существует в Р-программы в единственном экземпляре, в то время как обращаться к ней можно многократно с разных точек программы. Во время вызова функции пользователя активизируется последовательность операторов, которые ее повторяют, а с помощью переданных ей параметров модифицируется ее алгоритм.

Процесс разделения Р-программы - это творческий процесс, который проходит в несколько этапов. Сначала проделывают наиболее общий, генеральный алгоритм, не останавливаясь на технических деталях его реализации. Такая разработка алгоритма подается в виде последовательности относительно крупных частей программы (блоков), которые реализуют более или менее самостоятельные смысловые части алгоритма. Блоки, в свою очередь, могут разбиваться на более мелкие под блоки, а те, в свою очередь, - на блоки нижнего уровня и тому подобное.

Процесс последовательного структурирования Р-программы продолжается до тех пор, пока блоки, реализующих алгоритм, не станут простыми и легко программируемыми.

Преимущество такого нисходящего программирования очевидна: блоки повышают надежность Р-программы (каждый блок можно реализовать независимо от других и автономно его тестировать), облегчают ее понимание и позволяют вести разработку Р-программы коллективом программистов.

Приведем основные особенности функции пользователя.

1) Она имеет только один результат выполнения (но может иметь несколько входных параметров).

2) Результат сказывается именем функции и передается в Р-программу, которая ее вызвала.

Функция пользователя имеет вид:

Function [имя] (формальные входные параметры; var выходные параметры): тип;

Раздел описаний (начинается ключевым словом "var")
BEGIN

Раздел операторов

Вызывается функция ее именем при наличии фактических параметров. При этом вызов функции пользователя можно делать непосредственно в арифметическом выражении.

2. Область действия имен

Имена, объявленные в разделе описания основной программы, действуют в разделе операторов основной программы и в любой подпрограмме (процедуре или функции). Эти имена называются

глобальными. Имена, объявленные в подпрограмме, в том числе и формальные параметры, действуют только в этой подпрограмме и в любой объявленной, в ней или процедуре функции. Такие имена называются локальными. Они недоступны для операторов основной программы. Формальные параметры представляют собой список переменных с указанием их типа.

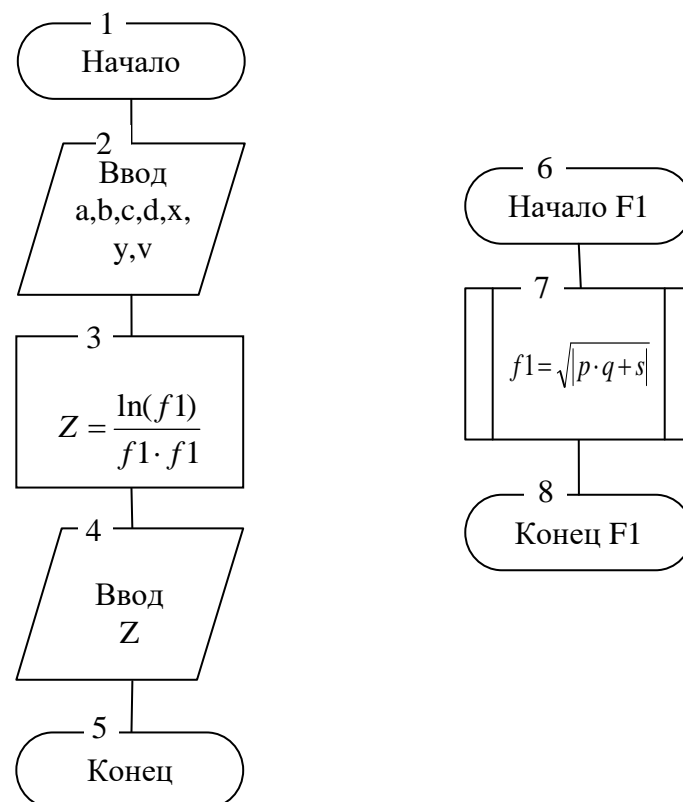
Для выделения выходных параметров перед ними ставится слово **Var**.

3. Пример оформления программы содержащий процедуру:

Составить блок-схему алгоритма, программу, содержащую программу и подпрограмму (функцию) для вычисления значения функции:

$$Z = \frac{\ln \sqrt{(a+c)v - (b+d)}}{\sqrt{ax+b}\sqrt{cy+d}}$$

Составим блок-схему алгоритма:



{Подпрограмма пользователя FUNCTION}

```
Program myProgram;
```

```
var
```

```
  a, b, c, d, x, y, v, z: real;
```

```
Function F1 (p, q, s: real): real;
begin
  F1 = Sqrt (Abs (p * q + s));
end;
```

```
Begin
  writeln ( 'Введите значения переменных a, b, c, d, x, y, v');
  read (a, b, c, d, x, y, v)
  z = Ln (F1 (a + c, v - (b + d))) / (F1 (a, x, b) * F1 (c, y, d));
  writeln ( 'Результат');
  writeln ( 'Значение функции Z =', z: 6: 2)
End.
```

Реакція ЕОМ:

Введите значения переменных a, b, c, d, x, y, v
 2.25 1.78 4.44 3.35 14.6 16.4 8.17
 результат
 Значение функции Z = 0.04

4 Контрольные вопросы

1. Что называют функцией?
2. Что такое заголовок функции?
3. Что включает в себя тело функции?
4. Что такое параметр (ы) функции?
5. Какие могут быть параметры в функции?
6. Объясните: Function A1 (a, b, c: real; var f: real): string;
7. Где программист должен размещать текст функции, приведите примеры.
8. Что такое формальные и фактические параметры?
9. Сколько входных и выходных параметров может содержать функция?
10. Для чего используют выходные параметры в функции?
11. Какое главное назначение функции?
12. Какие виды подпрограмм существуют в языке Pascal?
13. Какое соответствие должно существовать между формальными и фактическими параметрами?
14. Что это означает: Z = PK1 (0.98, 7.53, a, b) ?
15. Каким ключевым словом заканчивается функция?
16. Может ли тело процедуры включать еще одну функцию?
17. Каким образом обратиться к функции?
18. С каким ключевым словом должна начинаться функция?
19. ли обязательным является раздел описания переменных в функции?
20. Можно ли при вызове функции, во входных параметрах передавать константы, объясните?