

**Уважаемые студенты групп!**

**Вашему вниманию представлена лекция на тему «ОРГАНИЗАЦИЯ ПРОГРАММ, СОДЕРЖАЩИХ ПОДПРОГРАММЫ (процедуры)»**

**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 03.04.2023.

С уважением Ганзенко Ирина Владимировна

!!! Если возникнут вопросы обращаться по телефону 0721134803 (вацап), +79591134803 (телеграмм)

[disobuch.ganzenko2020@mail.ru](mailto:disobuch.ganzenko2020@mail.ru)

**ОРГАНИЗАЦИЯ ПРОГРАММ, СОДЕРЖАЩИХ ПОДПРОГРАММЫ (процедуры)**

**План**

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

**1 Теоретические положения**

В практике программирования часто приходится делать многократно одинаковые вычисления, но при различных входных данных. Чтобы исключить повторение одинаковых записей и сделать программу проще и понятнее, можно выделить эти повторяющиеся вычисления в самостоятельную часть программы, которая может быть использована многократно по мере необходимости. Автономная часть программы, реализующей определенный алгоритм, и допускает обращение к ней с разных частей общей программы, называется *подпрограммой*.

Подпрограммы оформляются в виде замкнутых частей программы, имеющие четко обозначены вход и выход. Подпрограммы позволяют более полно реализовать в программах метод структурного программирования.

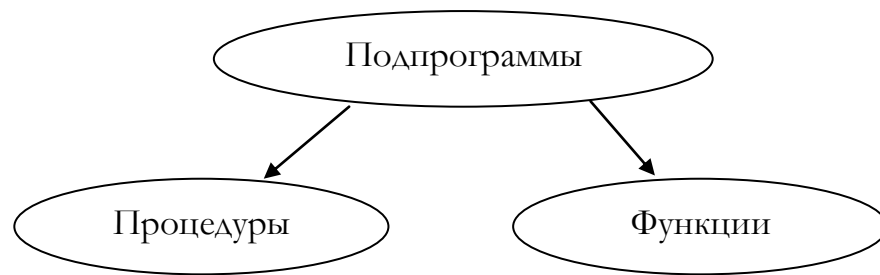


Рисунок 1 - Классификация подпрограмм

В языке Турбо-Паскаль выделяют два вида подпрограмм: процедуры (Procedure) и функции (Function). Любая программа может содержать несколько процедур и функций. Процедуры и функции объявляются в разделе описания вслед за разделом переменных.

Любая процедура оформляется аналогично программе, то есть содержит заголовок, раздел описаний и раздел операторов:

**PROCEDURE** [имя процедуры] (формальные входные параметры; var выходные параметры)  
 Раздел описаний (начинается ключевым словом "var")  
**BEGIN**  
 раздел операторов  
**END;**

В заголовке указывается служебное слово **PROCEDURE**, далее следует имя процедуры и параметры, которые называются *формальными*. Допускается использование процедур *без параметров*.

Раздел описаний процедуры, подобно основной программе, состоит из разделов меток, констант, типов, переменных и, в свою очередь, процедур и функций.

Раздел операторов заключается в скобки BEGIN ... END, причем после END, в отличие от основной программы, ставится точка с запятой. Выполнение программы начинается с операторов основной программы. Как только появляется необходимость выполнить действие процедуры, она вызывается по ее имени и начинает выполняться. Данные из основной программы передаются в процедуру (*входные параметры*). После выполнения процедуры ее результаты (*выходные параметры*) передаются в основную программу. Затем продолжает выполняться основная программа с оператора, следующего за вызовом процедуры.

Program SUNIGOT;

Раздел описания глобальных элементов

Procedure P1;

Раздел описания процедуры P1

Begin

Раздел описания операторов P1

End;

Procedure P2;

Раздел описания процедуры P2

Begin

Раздел описания операторов P2

End;

BEGIN

Раздел операторов основной программы

END.

## 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**.*

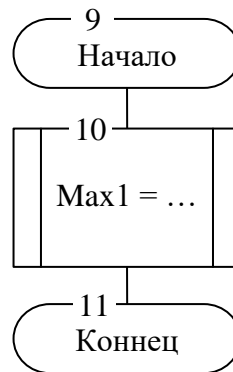
## 3. Пример оформления программы содержащий процедуру:

Нарисовать блок-схему алгоритма, программу, содержащую программу и подпрограмму (процедуру) для вычисления значения функции.

Даны действительные числа  $t, s$ . вычислить

$$D = \max(t, -2s, 1.17) + \max(2.2, t, s - t), \text{ где } t=1.3, s=0.8$$

**Составим блок-схему алгоритма:**



{Подпрограмма пользователя PROCEDURE}

**Program** myProgram;

**var**

t,s,d,m : real;

**procedure** Max1(a,b,c : real; **var** f : real);

**begin**

f := a;

**if** b > a **then** f := b;

**if** c > a **then** f := c;

**end;**

**Begin**

**writeln**('Введіть значення змінних t,s');

```
read(t,s);  
Max1(t, -2*s, 1.17, m);  
d := m;  
Max1(2.2, t, s - t, m);  
d := d + m;  
writeln('Результат');  
writeln('d=',d:6:2);  
End.
```

### **Реакция ЭВМ:**

Введите значения переменных t, s 1.3 0.8 Результат d= 3.50
---

В процедуре: а, в и с - входные параметры, их значения передаются из основной программы в эту процедуру. Результатами процедуры (выходными параметрами) является максимальное значение, которое передается из процедуры в основную программу и там используется.

В общем случае процедуры могут содержать один или несколько входных и выходных параметров, или вообще не иметь их. Вызов процедуры в основной программе имеет форму:

Имя процедуры (фактические параметры).

Те из фактических параметров, которые передаются в процедуру, могут быть переменными, или константами, или выражениями.

## **4 Контрольные вопросы**

1. Что называют процедурой?
2. Что такое заголовок процедуры?
3. Что включает в себя тело процедуры?
4. Что такое параметр (ы) процедуры?
5. Какие могут быть параметры в процедуре?
6. Объясните: Procedure Max1 (a, b, c: real; var f: real)
7. Где программист должен размещать текст процедуры, приведите примеры.
8. Что такое формальные и фактические параметры?
9. Сколько входных и выходных параметров может содержать процедура?
10. Для чего используют выходные параметры в процедуре?
11. Какое главное назначение процедур?
12. Какие виды подпрограмм существуют в языке Pascal?

13. Какое соответствие должно существовать между формальными и фактическими параметрами?
14. Что это означает: РК1 (0.98,7.53, a, b) ?
15. Каким ключевым словом заканчивается процедура?
16. Может ли тело процедуры включать еще одну процедуру?
17. Каким образом обратиться к процедуре?
18. С какого ключевого слова должна начинаться процедура?
19. обязательным является раздел описания переменных в процедуре?
20. Можно ли при вызове процедуры, во входных параметрах передавать константы, объясните?