

Основные вопросы:

1. Подпрограммы.
2. Примеры использования стандартных процедур и функций в программах.
3. Пользовательские функции. Формат объявления пользовательской функции.
4. Пользовательские процедуры. Формат объявления пользовательской процедуры.
5. Составление программ с использованием пользовательских функций.
6. Составление программ с использованием пользовательских процедур.

Первый вопрос: Подпрограммы.

Подпрограммой называется участок программы, оформленный определенным образом, к которому можно обращаться из разных точек программы любое число раз. При этом подпрограмма может решать каждый раз одну и ту же задачу с разными значениями исходных данных.

ГЛАВНАЯ



Подпрограмма

Подпрограмма – повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы.



В языке Pascal механизм подпрограмм реализуется в виде **ПРОЦЕДУР** (PROCEDURE) и **ФУНКЦИЙ** (FUNCTION)



Основные термины и понятия

Подпрограммы – это программы в миниатюре. Применение подпрограмм дает возможность уменьшить число повторений одной и той же последовательности операторов.

пример

Структура подпрограммы аналогична структуре основной программы на языке Паскаль.

В программе **объявление подпрограмм** должно располагаться между разделами переменных и операторов.

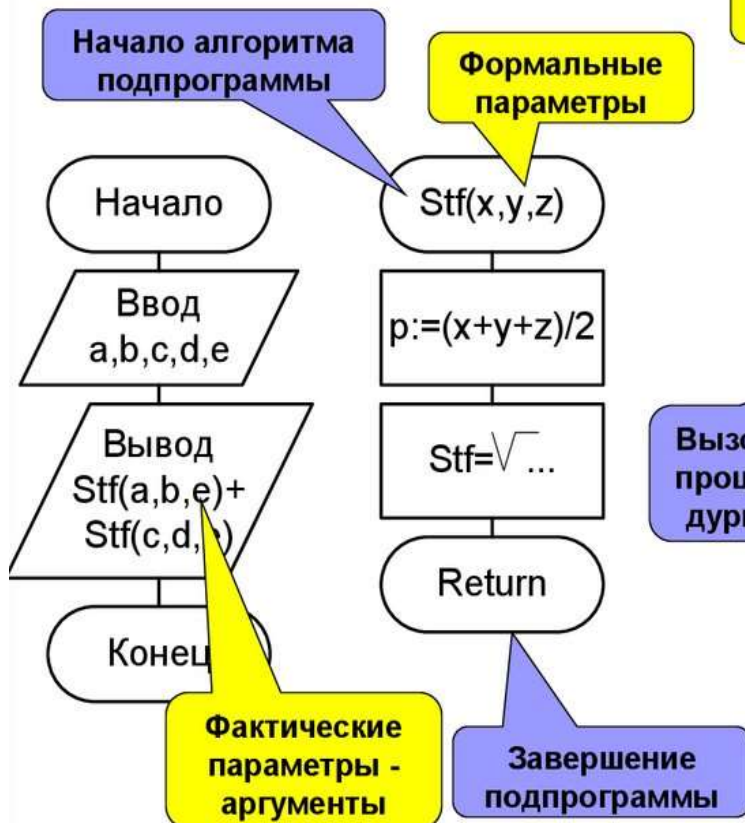
Подпрограммы подразделяются на **процедуры и функции**.

Каждая процедура и функция **определяется только один раз**, но **может использоваться многократно**.

Вызов подпрограммы осуществляется упоминанием имени подпрограммы.

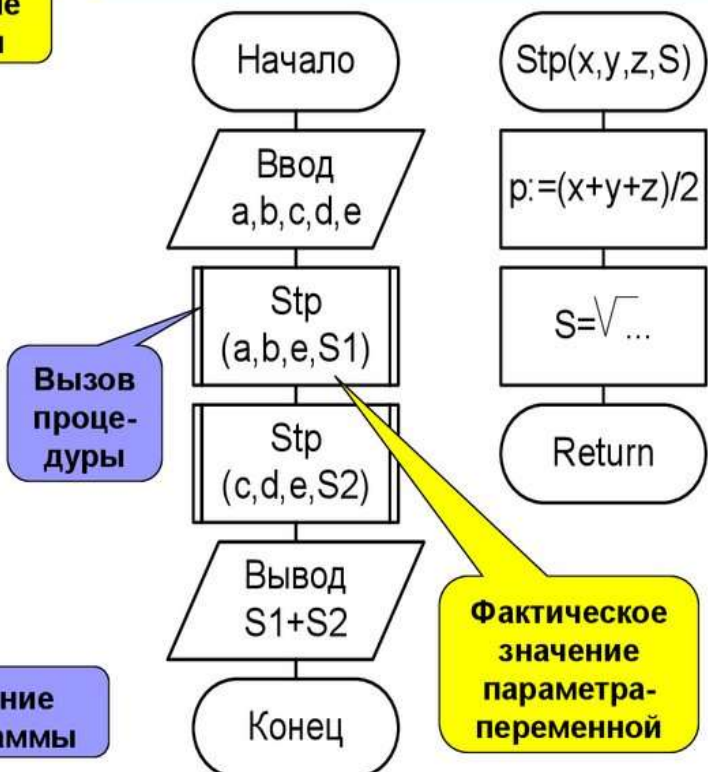
Схемы алгоритмов подпрограмм

Подпрограмма-функция



Подпрограмма-процедура

Формальный параметр-переменная в заголовке на схеме не выделяется



Второй вопрос: Примеры использования стандартных процедур и функций в программах.

Стандартные функции

В Pascal	В математике	Тип результата
abs(x)	x	целое, вещественное
sqr(x)	x ²	целое, вещественное
sgrt(x)	\sqrt{x}	вещественное
pi	$\pi=3.14$	вещественное
sin(x)	sin x, где x - в радианах	вещественное
cos(x)	cos x, где x - в радианах	вещественное
arctan(x)	arctn x, где x - в радианах	вещественное
int(x)	целая часть x	вещественное
frac(x)	дробная часть x	вещественное
trunc(x)	отсечение дробной части числа	целое
round(x)	округление до ближайшего целого	целое
A div B	деление A на B с отбрасыванием остатка	целое
A mod B	остаток от целого деления A на B	целое

Стандартные процедуры и функции

■ Поиск подстроки в строке (первое вхождение).

Поиск подстроки в строке осуществляется с помощью функции **pos**, **k:=pos(st1, st)**, где st1—подстрока, st—строка, k—позиция **первого** вхождения подстроки, либо 0, если подстрока отсутствует.

Пример:

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.	23.	24.	25.	
st:=	'И	в	а	н	о	в		А	л	е	к	с	а	н	д	р		г	.	М	о	с	к	в	а	'

Будем искать подстроку "г.Москва" в строке st,
k:=pos('г.Москва', st);

Ответ: k=18.

Внимание! Повторим еще раз:

1. Значение функции pos—переменная целого типа.
2. Функция pos ищет **первое** вхождение подстроки в строку.

Задача 1.

- Составить программу, вычисляющую сумму цифр трехзначного числа n .

n – трехзначное число (например 762) – исходные данные

a, b, c – цифры этого числа (вспомогательные переменные)

S – сумма цифр этого числа (результат)

```

Program chislo;
var n, a, b, c, s : integer;
begin
  writeln ('введите трехзначное число');
  readln (n);
  c:=n mod 10;
  a:=n div 100;
  b:=(n div 10) mod 10;
  s:=a+b+c;
  writeln ('сумма цифр числа равна ', s);
end.

```

Третий вопрос: Пользовательские функции. Формат объявления пользовательской функции.

Функция

Функция – подпрограмма, имеющая имя, которая объявляется в разделе описаний, вызывается упоминанием ее имени в теле основной программы.

Результатом работы функции является одно значение.

При вызове функции:

- ☞ работа основной программы прерывается
- ☞ вычисляется значение функции
- ☞ по окончании работы функции, в тело программы возвращается значение функции, при заданных параметрах

пример

Функции

Общая форма записи заголовка функции:

Function Имя (список формальных параметров): тип результата;

В теле функции обязательно должен присутствовать оператор присваивания, в левой части которой стоит имя функции.

Вызов и выполнение функции в теле программы осуществляются следующим образом :

переменная := Имя функции (список фактических параметров);

ИЛИ

write(' Значение функции равно – ',

Имя функции (список фактических параметров));

Четвертый вопрос: Пользовательские процедуры. Формат объявления пользовательской процедуры.

Процедуры

Процедура - подпрограмма, имеющая имя, которая объявляется в разделе описаний, вызывается упоминанием ее имени в теле основной программы. Результатом работы процедуры является выполнение ее операторов .

При вызове процедуры:

- ☞ работа основной программы прерывается
- ☞ выполняется тело процедуры
- ☞ по окончании процедуры, продолжается работа основной программы

Общий вид структуры подпрограммы

```

Program pr;
Var a,b:integer;
Procedure <имя> (формальные параметры);
VAR (описание локальных параметров, они
    могут быть или не быть)
begin
    тело процедуры
end;
    BEGIN
        <имя> (Фактические параметры)
    END.

```

Процедуры

Любая процедура может содержать несколько параметров и несколько результатов ее выполнения.

PROCEDURE имя (формальные параметры) ;

Раздел описаний

Begin

Раздел операторов

end;

Пример процедуры

Procedure express (a,b,c:real; var x,y:real) ;

var z:real;

Begin

z:=a+b+c;

x:=sqr(z) ;

y:=sqrt(z) ;

End;

Вызов процедуры из программы:
Express(25.4, 44.6, 30, x1, y1) ;
 {здесь a=25.4, b=44.6, c=30}

Результатом выполнения процедуры являются x и y, которые передают свои значения в x1, y1

(т.е. в основной программе будем иметь x1=10000, y1=10)

Параметры

- **Фактические** – располагаются в головном модуле при вызове процедуры.
- **Формальные** – описываются в заголовке процедуры, к ним относятся входные и выходные параметры.
- **Входные** – это и параметры значений, описываются через запятую с указанием типа. При выходе из процедуры – не сохраняются.
- **Выходные** – это и параметры переменные, описываются после VAR через запятую, с указанием типа. При выходе из процедуры – сохраняются.

Процедуры

Общая форма записи заголовка процедуры:

Procedure Имя (список формальных параметров);

пример

Вызов и выполнение процедуры в теле программы осуществляются при помощи оператора процедуры:

Имя процедуры (список фактических параметров);

Количество, порядок следования, тип каждого из формальных и соответствующих им фактических параметров должно быть одинаковым.

При переходе из тела программы к процедуре **формальному параметру присваивается значение фактического.**

Процедура может быть без параметров.

Процедура m_color, изменяет цвет текста

```

uses CRT; var i : integer ;

        procedure m_color ( c : word);
        { c- формальный параметр }
        begin clrscr ; TextColor (c) ;
        writeln( ' Press <ENTER> ' ) ;
        readln; end;

BEGIN
for i := 1 to 15 do m_color ( i );
{ i – фактический параметр}

        writeln ( ' НАЖМИ <ENTER> ' ) ;

        readln; END.

```

Пятый вопрос: Составление программ с использованием пользовательских функций.

Функции

Задача: найти наибольшее из 4-х чисел, используя подпрограмму нахождения наибольшего из 2-х чисел

```

program max;
uses crt;
var a,b,c,d,p,q,m: integer;
    function bid(x,y: real): real;
    begin
        if x>y then bid:=x else bid:=y
    end;
begin
clrscr;
write('введите 4 числа:');
readln(a,b,c,d);
p:=bid(a,b);
q:=bid(c,d);
m:=bid(p,q);
writeln('наибольшее из 4-х чисел';m);
readkey;
end.

```

Шестой вопрос: Составление программ с использованием пользовательских процедур.

Пример

Составить программу сложения 2-х чисел

```

Program pr;
Var a,b,s:real;
Procedure summa(x,y:real; var s:real);
Begin
S:=x+y;
End;
Begin
Writeln('сложить 2 числа');
Summa(a,b,s);
Writeln('s=',s);
End.
  
```

фактические формальные

параметры-значений, входные, при выходе не сохраняются параметры-переменные, выходные, при выходе сохраняются

Задание

Написать и отладить программы с использованием процедур:

1. Программу вычисления площади треугольника.
2. Программу вычисления площади квадрата.
3. Сгенерировать массив случайных чисел из N элементов. Вывести массив. Найти минимальный и максимальный элементы и их индексы, вывести. Ваша основная программа должна иметь вид:

```
Uses crt; BEGIN Clrscr;
```

```
Mass_min(200,200);
```

```
{ В процедуру Mass_min передаются два параметра – диапазон генерируемых чисел }
```

```
Vichod; {Процедура выхода}
```

```
END.
```