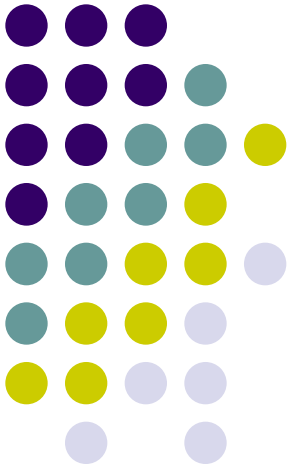
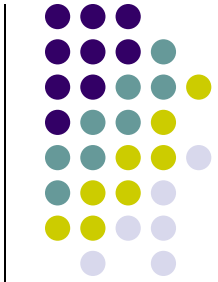
Подпрограммы в языке Object Pascal

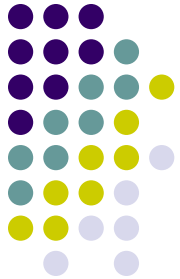


План



1. Понятие «подпрограмма»
2. Описание функций в ЯП Object Pascal
3. Описание процедур в в ЯП Object Pascal
4. Параметры подпрограмм в ЯП Object Pascal
5. Область действия описаний

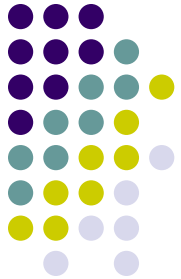
1. Понятие «подпрограмма»



Подпрограмма – обособленная именованная часть программы со своим собственным контекстом имен.

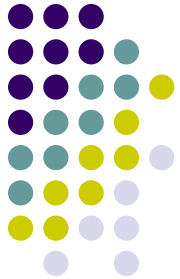
Средство структурирования программ

2. Описание функций



Функции – группа операторов, в результате выполнения которой вычисляется одно значение, присваиваемое имени функции

Функция включает:



- Заголовок
- Раздел описаний (констант, переменных, типов, процедур, функций), являющихся локальными по отношению к описываемой функции;
- Тело функции

Структура описания функции

Function F (q1:T1; q2:t2;…) : T;

<Раздел описания локальных переменных и подпрограмм>;

Begin

<операторы тела функции>;

F:=<вычисленное значение>;

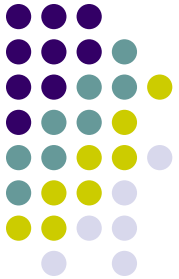
End;

Параметры:

Локальные Формальные Фактические

```
Programm Pr4;
Var a: longint;
….
Function Factor (k: integer): longint;
  Var i: integer; p: longint;
begin
   p:=1;
   for i:=1 to k do p:=p*i;
   Factor:=p;
end;
…
WriteIn( Factor(a) );
…
```

Обращение к функции:

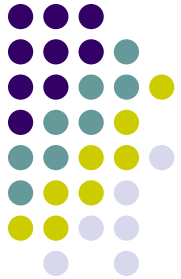


c:=F(b1, b2,…,bn)

Writeln(F(b1, b2,…,bn))

A:=X* F(b1, b2,…,bn)

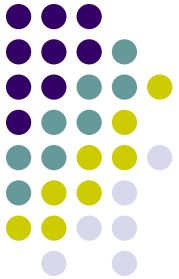
Result



- Имя функции – это не переменная. Оно может фигурировать только в левой части оператора присваивания и не может входить в выражения
- **Result** – это переменная

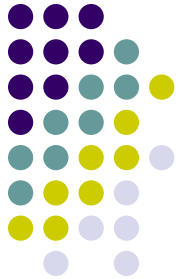
```
Function Max (x, y: real): Real;
Begin
   If x>y Then Result: =x
                Else
                     Result: =y;
End;
```

3. Описание процедур



Процедуры используются в тех случаях, когда в подпрограмме необходимо получить несколько результатов

Процедура включает



- Заголовок
- Раздел описаний (констант, переменных, типов, процедур, функций), являющихся локальными по отношению к описываемой процедуре;
- Тело процедуры

Структура описания процедуры

Procedure H (q1:t1; **var** q2:t2;…);

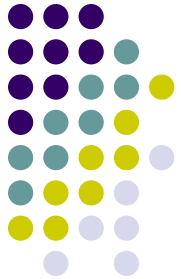
<Раздел описания локальных переменных, типов, констант, подпрограмм>;

Begin

<операторы тела процедуры>;

End;

Обращение к процедуре

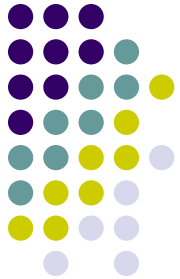


…

H(b1, b2, …, bn);

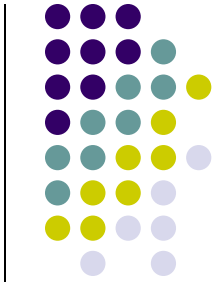
…

Параметры

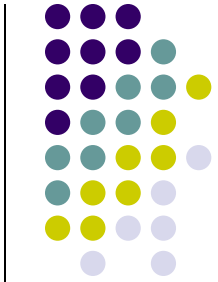


- Локальные и глобальные
- Формальные и фактические
- Параметры-значения и параметры-переменные

Задача: вычислить F=n!-m!

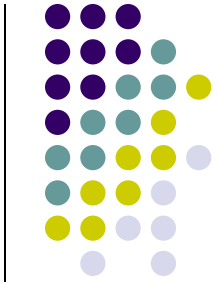


```
Program Pr5;
 Var n,m: integer; f: longint;
Function Factor(k: integer;) : longint;
 var i: integer; p: longint;
begin
 p:=1; for i:=1 to k do p:=p*i;
 Result:=p;
end;
```

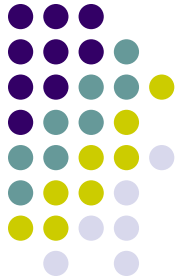


```
Begin
 writeln('input n, m');
 readln(n,m);
 if n>m then f:=factor(n)-factor(m)
 else f:=factor(m)-factor(n);
 writeln('f = ', f);
End.
```

Задача: провести сортировку трех чисел

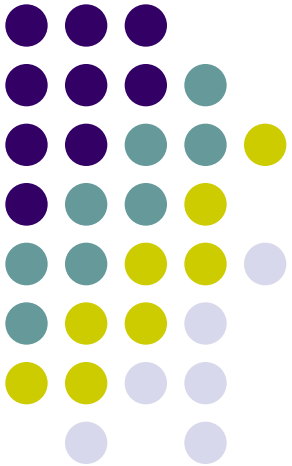


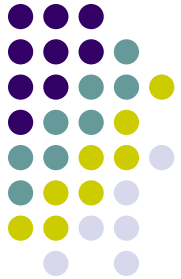
```
Program Pr6;
 Var x,y,z: integer;
procedure Sort2(var a,b : integer);
 Var c: integer;
 begin
        if a>b then
        begin
                c:=a;
                a:=b;
                b:=c
        end;
 end;
```



```
Begin
 writeln ('Sortirovka 3 numbers');
 writeln ('Your 3 numbers, please:');
 readln (x,y,z);
 sort2 (x,y);
 sort2 (y,z);
 sort2 (x,y);
 writeln ('Result of sort is  ', x, y, z);
 readln;
end.
```

4. Различные варианты передачи параметров в функции и процедуры

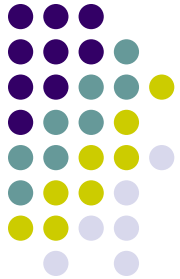




При организации процедур и функций различают несколько способов передачи параметров в подпрограмму:

1) Передача по значению
2) Передача параметров как переменных
3) Передача параметров как констант
4) Передача параметров как выходных параметров

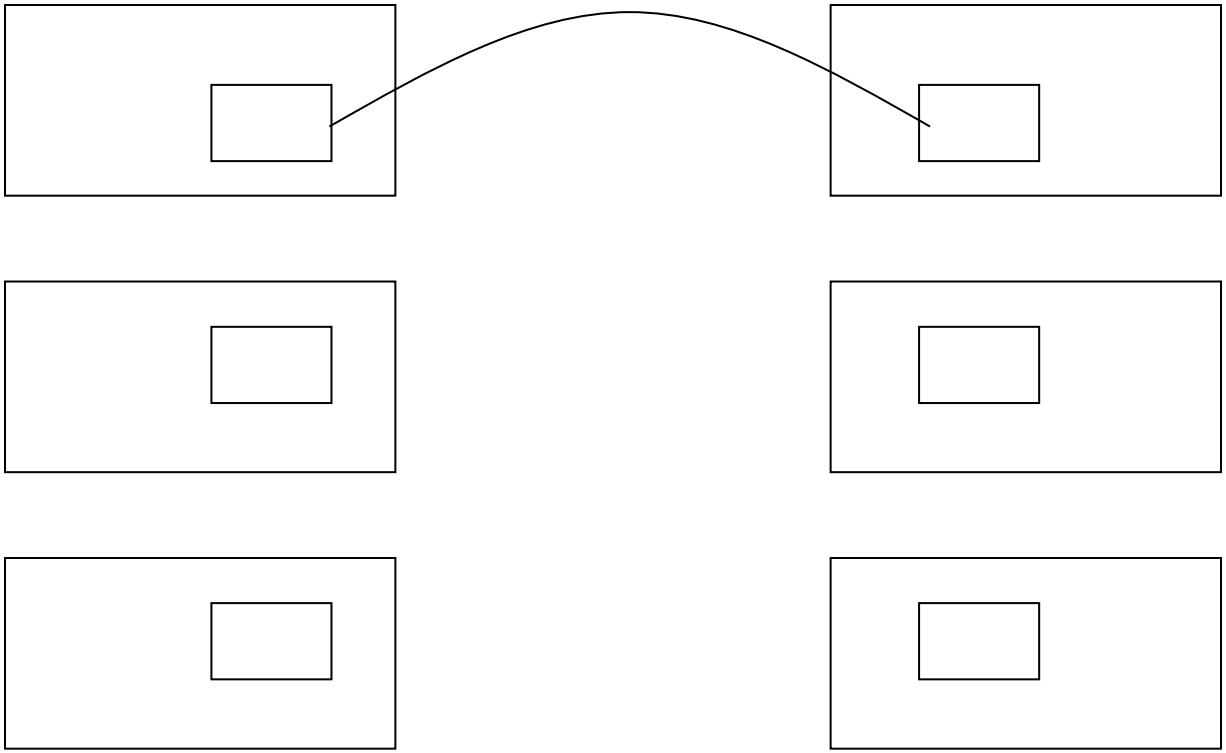
1. Передача по значению.



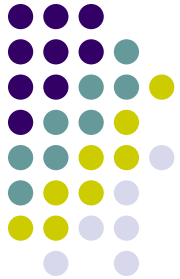
- **Параметры значения** определяют исходные данные для работы процедур и функций. В списке формальных параметров они описываются в следующем виде:
- (q1:T1; q2:T2) или (q1, q2:T)
- При вызове подпрограммы фактический параметр, может быть любым выражением, результат вычисления которого принадлежит тому же типу, что и формальный параметр.
- **Procedure myFun (a:integer; b,c:real);**

- При обращении к подпрограмме выражение вычисляется. В памяти создаются временные переменные с именами q1 и q2, и в них копируется результат вычислений.
- Изменение значений переменных q1 и q2 в теле подпрограммы никак не влияет на значения переменных из основной программы.
- При выходе из подпрограммы переменные q1 и q2 уничтожаются.

передача по значению

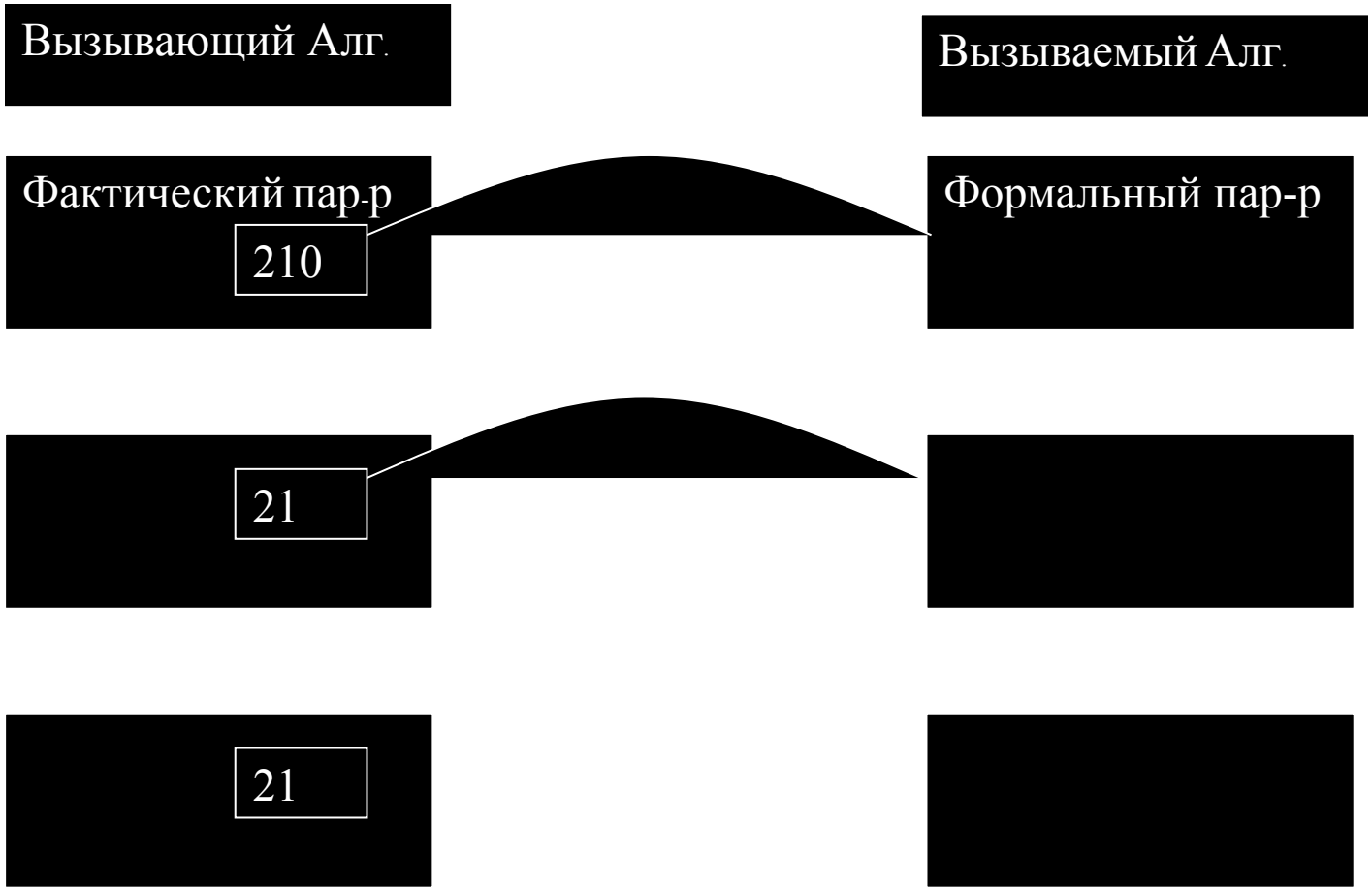


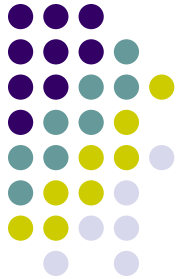
2) Передача параметров как переменных



- Параметры переменные позволяют передавать в основную программу измененные значения.
- Параметры переменные в списке формальных параметров описываются с помощью указания перед ними ключевого слова var. Например:
- (var q1, q2:t1; var q3:t2) ;
- Procedure KvUravn(a,b,c:real; var x1,x2:real)

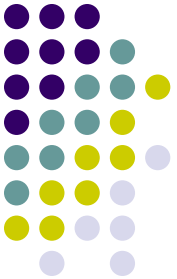
передача по ссылке





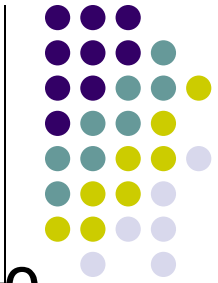
- При обращении к подпрограмме фактические параметры должны быть обязательно переменными, того же типа, что и формальные параметры.
- В подпрограмму передается адрес переменной (ссылка) и операторы процедуры непосредственно используют данную переменную.
- В данном случае любые изменения параметра q1, произведенные в подпрограмме, в действительности относятся к той переменной, которая будет указана при вызове подпрограммы на месте q1.

3) Передача параметров как констант



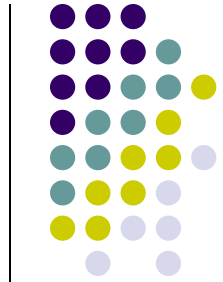
- Передача параметра как константы осуществляется заданием перед его описанием ключевого слова const.
- Например:
- (const q1:t1);
- Параметры константы по механизму передачи похожи на параметры значения, но их значения нельзя изменять в теле подпрограммы.

4) Передача параметров как выходных параметров



- Выходные параметры описываются с помощью зарезервированного слова out.
- Например: (out q1:t1);
- Выходные параметры по механизму передачи похожи на параметры переменные, но при этом в подпрограмму не передается начальное значение этого параметра, т.е. память, занимаемая соответствующей переменной, указанной при вызове подпрограмме на месте q1, будет очищена при вызове подпрограммы.

5. Область действия описаний



- Областью действия описания любого программного объекта (переменной, типа, константы и т.д.) является тот блок, в котором расположено это описание.
- Если данный блок вложен в другой (подпрограмма), то присутствующие в нем описания являются локальными. Они действуют только в пределах внутреннего блока.
- Например:

- PROGRAM Prog;
- Var V1: integer;
- Procedure A;
- Var V2: integer;
- Begin … end;
- Procedure B;
- Var V3: integer;
- Procedure B1;
- Var V4: integer;
- Begin … end;
- Begin … end;
- Var V5: integer;
- BEGIN … END.