A silver metal spiral binding is visible on the left side of the page, with the wire looping through a series of holes.

Понятие подпрограмм. Подпрограммы в языке Паскаль.

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

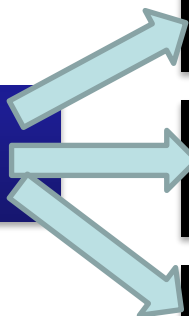
Подпрограмма – повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы.

ГЛАВНАЯ

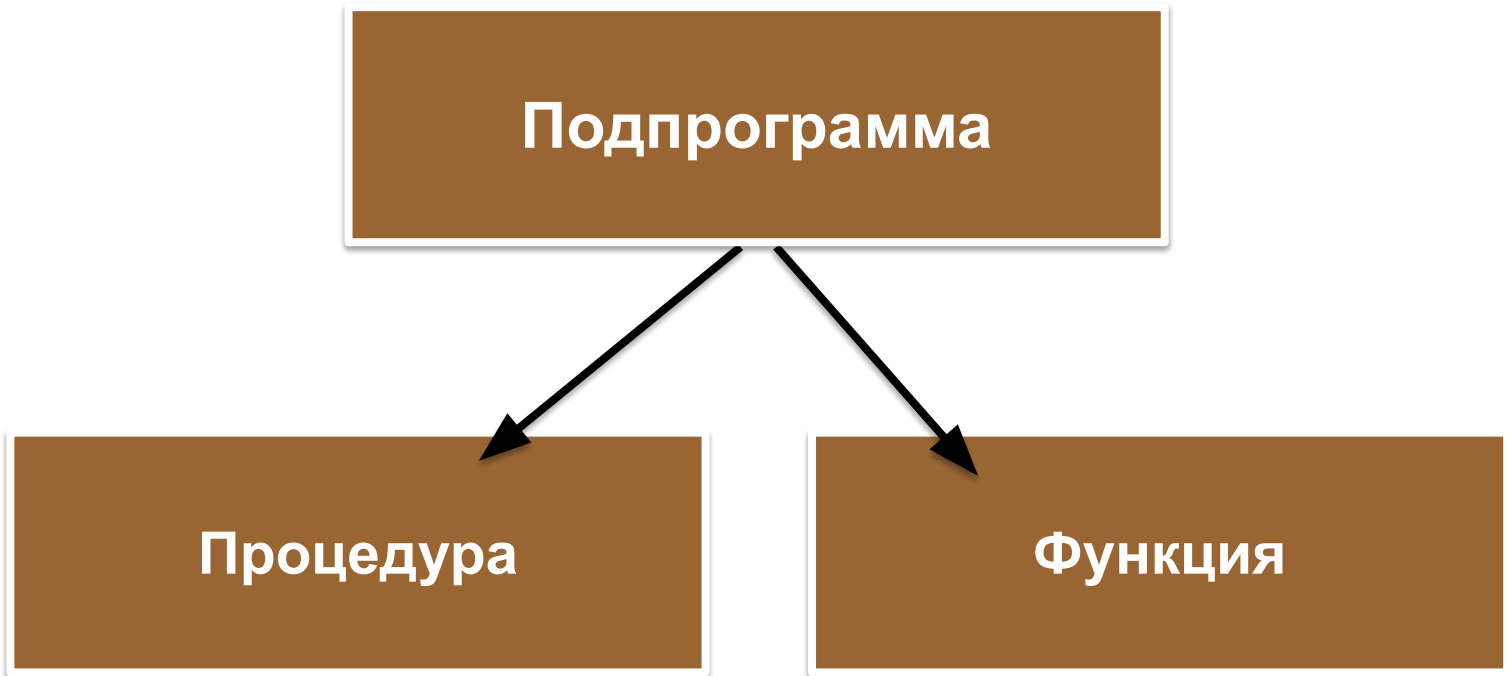
Подпрограмма 1

Подпрограмма 2

Подпрограмма 3



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



Подпрограммы

ПРОЦЕДУРЫ

(PROCEDURE)

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.

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.

Пример

Составить программу сложения 2-х чисел

Program pr;

фактические

формальные

Var a,b,s:real;

Procedure **summa**(x,y:real; var s:real);

Begin

S:=x+y;

End;

параметры-значений,
входные, при выходе
не сохраняются

параметры-переменные,
выходные, при выходе
сохраняются

Begin

Writeln('сложить 2 числа');

Summa(a,b,s);

Writeln('s=',s);

End.

Формальные и фактические
параметры должны совпадать
по 3 признакам:

✓ количеству

✓ типу

✓ порядку следования

Procedure Ploshad3(a, b, c : real; var s: real);

Ploshad3(AB, BC, AC, S1);



Вывести на экран 5 строк примерно такого текста:

Привет, Катя! Поздравляю тебя с новым годом!

Привет, Таня! Поздравляю тебя с новым годом!

Привет, Вася! Поздравляю тебя с новым годом!

Привет, Коля! Поздравляю тебя с новым годом!

Привет, Боря! Поздравляю тебя с новым годом!

В программе использовать процедуру для вывода текста.

```
program primer;  
  var s: string;
```

```
  procedure ok;  
    begin  
      writeln ('Привет', s, '! Поздравляю тебя с новым годом! ');  
    end;
```

```
begin  
  s:='Катя';  
  ok;  
  s:='Таня';  
  ok;  
  s:='Вася';  
  ok;  
  s:='Коля';  
  ok;  
  s:='Боря';  
  ok;  
end.
```

Подпрограммы

ФУНКЦИИ

(FUNCTION)

Функции

Функция - подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем функции.

Перечень формальных параметров и их типов

```
function <имя_функции> (<описание входных данных>):  
<тип_функции>;
```

begin

<операторы>;

<имя_функции> := <результат>

end;

Тип результата

! В блоке функции обязательно должен присутствовать оператор **<имя_функции>:=<результат>**.

Для вызова функции её имя со списком параметров может быть в любом месте программы(в выражении, в условиях, в операторе **write**).

Функция поиска максимального из 2-х

```
program n_20;  
  var a, b, c, d, f: integer;
```

} Описание
переменных

```
Function max(x,y:integer):integer;  
begin  
  if x>y then max:=x  
    else max:=y;  
end;
```

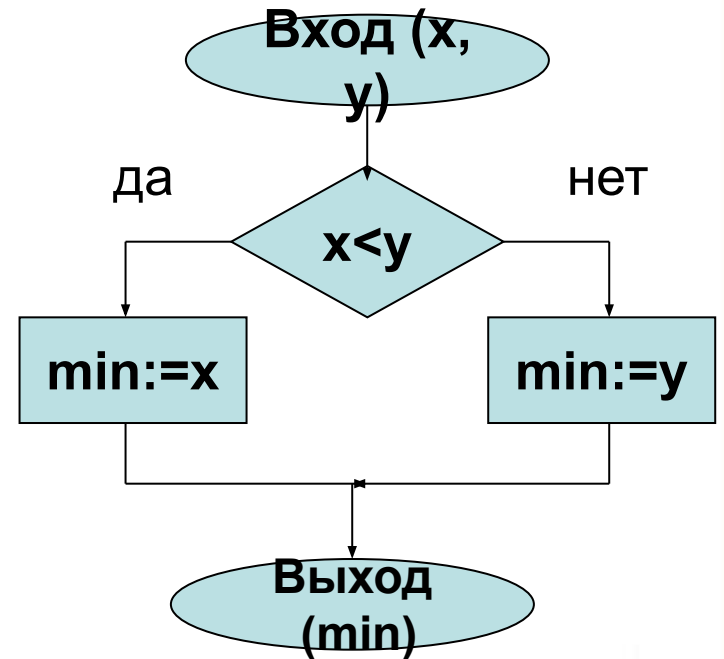
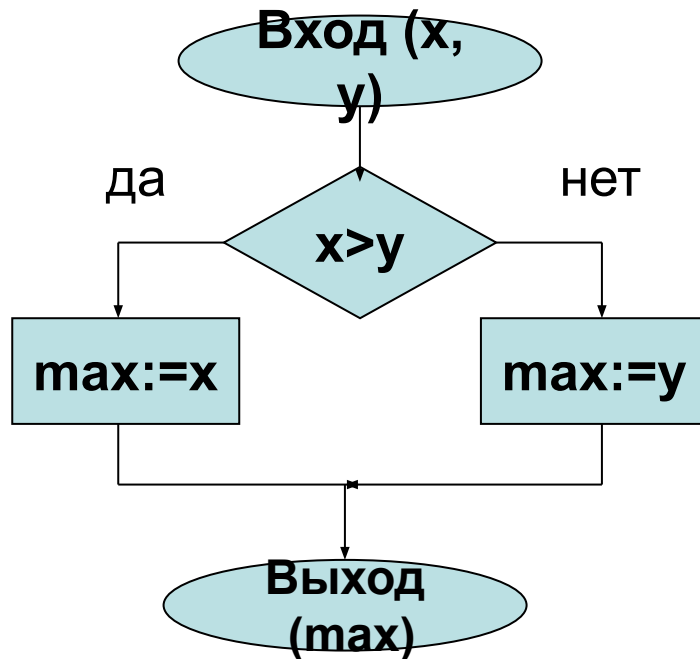
} Раздел
описания
Под-
программы

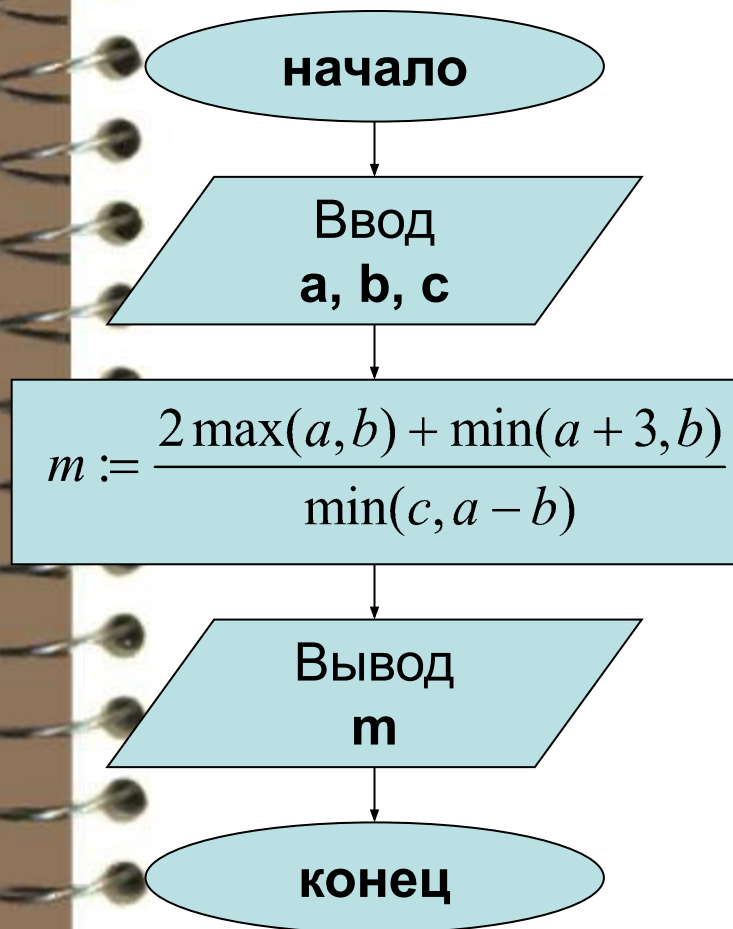
```
begin  
  readln (a, b, c, d);  
  f:=max(max(a, b),max(c, d));  
  writeln ('f=', f);  
end.
```

} Раздел
операторов
главной
программы

Найти $m := \frac{2 \max(a, b) + \min(a + 3, b)}{\min(c, a - b)}$, используя подпрограммы нахождения большего и меньшего из двух чисел.

Вспомогательные алгоритмы





program pr2;

var a, b, c, m: real;

function max(x, y: real): real;

begin

if x>y then max:=x else max:=y

end;

function min(x, y: real): real;

begin

if x<y then min:=x else min:=y

end;

begin

writeln('a='); readln (a);

writeln('b='); readln (b);

writeln('c='); readln (c);

m:=(2*max(a, b)+min(a+3,b))/min(c,a-b);

writeln ('max= ', m);

end.

Вычисление функции

$$\frac{\sqrt{6}+6}{2} + \frac{\sqrt{13}+13}{2} + \frac{\sqrt{21}+21}{2}$$

```
program f1;  
uses Crt;  
var n,x,s: real;  
    i:integer;
```

Ф
У
Н
К
Ц
И
Я

```
function fun(a:real):real;  
var y:real;  
begin  
y:=sqrt(a)+a;  
fun:=y;  
end;
```

```
begin  
s:=0;  
for i:=1 to 3 do begin  
write('n='); readln(n);  
x:=fun(n)/2;  
s:=s+x;  
end;  
writeln('s=',s:3:5);  
end.
```



Требуется найти разность факториалов $F=M!-K!$

PROGRAM 4;

VAR F,M,K:1INTEGER;

FUNCTION **FACT**(N:INTEGER):INTEGER;

VAR P, I:INTEGER;

BEGIN

P:=1;

FOR I =2 TO N DO

P=P*I;

FACT:=P;

END;

BEGIN

READ(M,K);

F:=**FACT**(M) - **FACT**(K);

WRITE (F)

END.

M=4

K=3

ФУНКЦИИ

Отличительные особенности: функция имеет результат выполнения (но может иметь только один входных параметров); результат обозначается именем функции и передается в основную программу. несколько

FUNCTION имя (параметры) : тип данных;

Раздел описаний

Begin

Раздел операторов ;

имя:=

end;

Пример функции:

Найти разность факториалов $F:=m!-k!$

```
Function FACT(n:integer):integer;
```

```
  Var p,i:integer;
```

```
Begin
```

```
  P:=1;
```

```
  For i:=1 to n do
```

```
    P:=p*i;
```

```
  FACT:=p;
```

```
End;
```

Вызов функции из программы:

$F:=FACT(m)-FACT(k);$



[вернуться](#)

Опорный конспект

Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью *подпрограмм*.

Подпрограмма

```
graph TD; A[Подпрограмма] --> B[Процедура]; A --> C[Функция]; B --> D[Подпрограмма, имеющая произвольное количество входных и выходных данных.]; C --> E[Подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем функции.];
```

Процедура

Подпрограмма, имеющая произвольное количество входных и выходных данных.

Функция

Подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем функции.

В языке Паскаль подпрограммы реализуются в виде **процедур и функций.**

процедура	функция
предназначена	предназначена
для выполнения законченной последовательности действий	для выполнения законченной последовательности действий,
	<u>результатом</u> которой является один
	параметр
заголовок	заголовок
<code>procedure <ИМЯ>(<список формальных параметров>);</code>	<code>function<ИМЯ > (<список формальных параметров>): <тип ></code>
обращение к процедуре	обращение к функции
<code><ИМЯ> (< список фактических параметров >);</code>	<code><ИМЯ> (< список фактических параметров >);</code>
осуществляется в любом месте программы	функция записывается только в <u>выражениях</u>
	(т.е. справа от знака :=)

A spiral-bound notebook with a white page and a brown cover. The spiral binding is on the left side. The text is written in a bold, yellow, sans-serif font.

Практическая работа

Подпрограммы

ФУНКЦИИ

(FUNCTION)

Вариант 1

Напишите, что печатает следующая программа.

```
Program P1;
Var a,b,c, d: integer;
Function func (x,y,z;integer):integer;
  Var W: integer;
  Begin
    W:=x+y+z;
    If y<z+1 then W:=W+1;
    Func:=W;
  End;
Begin
  A:=5;
  B:=4;
  C:=2;
  D:= func (a, b, c);
  Writeln ('func от ',A, ' , ',B,'и', C,'даст
результат',D);
  A:=func(b,c,d);
  Writeln ('func от ',B, ' , ',C,'и', D,'даст
результат',D);
End.
```


Вариант 2

Напишите, что печатает следующая программа.

```
Program P2;
Var d,e,f,g,h,i,j,k:integer;
Funcion Sub1(x,y:integer):integer;
  Var x,y:integer;
  Begin
    If x+y=3 then Sub1:=3 else Sub1:=y-x;
  End;
Funcion Sub2(p,q:integer):integer;
  Var R,S:integer;
  Begin
    R:=Sub1(p,q);
    S:=Sub1(q,p);
    Sub2:=R+S;
  End;
Begin
  g:=4; h:=7; i:=9; j:=5; k:=2; d:=1; e:=3;
  f:=Sub1(g,h);
  f:=Sub2(i,j);
  f:=sub1(g,k);
  writeln(g,' ',h,' ',i,' ',j,' ',k,' ',d,' ',e,' ',f);
end.
```

A silver metal spiral binding is visible on the left side of the page, consisting of a series of loops that hold the paper in place.

Подпрограммы

ПРОЦЕДУРЫ

(PROCEDURE)

Что выведется на экран?

Program z2;

Procedure horline (Len: integer; s: char);

Var i: integer;

Begin

For i:=1 **to** Len **do** write(s);

Writeln;

end;

begin

Horline (10,'-');

Horline (20,'*');

Horline (30,'#');

End.