

Тема 9

Програмни модули в Pascal. Подпрограми – процедури и функции

1. Прости и съставни програми

За всички разгледани до сега примери съставените програми наричаме прости програми. Много често практическите задачи, за които трябва да се разработи програма, са много по-сложни и обемни. Добър подход за решаване на такива задачи е те да се разделят на подзадачи, всяка от които да е относително завършена и самостоятелна обработка над определени данни. В този случай може да се разработи програмен модул за всяка подзадача, като накрая всички модули се обединят в една обща програма. Програма, която е организирана по този начин наричаме съставна програма. Съставната програма се състои от няколко програмни единици. Тази програмна единица, която обединява всички останали се нарича главна програма, а програмните единици, които са включени в нея се наричат подпрограми. Подпрограмите могат да се изпълняват само като компоненти на главна програма, т.е. не могат да се изпълняват самостоятелно. Една подпрограма може да се изпълни многократно в рамките на една главна програма.

Подпрограмата (ПП) е логически завършена, относително самостоятелна програмна част, която има свое име и може да бъде извиквана за изпълнение чрез името си произволен брой пъти от различни места в главната програма.

Съставната програма може да има по-сложна структура. В главната програма може да има подпрограма(и) а също така е възможно подпрограмата да съдържа в себе си подпрограма. Например:



Идеята за структура на съставната програма главна програма – подпрограми е в основата на най-прогресивните подходи в програмирането – структурното програмиране.

Използването на подпрограми има следните предимства:

- По-прегледна, логически ясна и удобна за разчитане програма;
- Разработването на подпрограмите може да стане от различни програмисти;
- По-лесна настройка;
- По-ефективно използване на ОП (обектите в ПП заемат място в ОП само по време на изпълнението и);
- По-лесна модификация в програмата;
- Възможност за създаване на библиотеки от ПП.

В Pascal се използват два вида подпрограми: функция и процедура.

Когато в главната програма се включват ПП трябва ясно да се различават два момента:

1. Подпрограмите функция и процедура, които се съставят от програмиста, задължително се описват в декларативната част на главната програма. Описването на ПП не предизвиква нейното изпълнение.

2. ПП се извикват за изпълнение (обръщаме се към тях) в изпълнителната част на главната програма.

За описание и извикване на ПП се прилагат съответните правила в езика Pascal.

При работата с ПП се използва техниката на формалните и фактическите параметри. Чрез тези параметри се осъществява обменът на информация между главната програма и ПП. Трябва да се прави разлика между формалните параметри при описанието на ПП и фактическите параметри при извикването на ПП. Може да се направи аналогия с описанието на една формула в общ вид (с букви), напр.: $(a+b)/a$ и заместването и с конкретни стойности (числа), напр.: $(2+5)/2$. Чрез формалните параметри се описват в "общ вид" действията, които трябва да се изпълнят от ПП. Фактическите параметри са обекти, декларирани в главната програма и са действителните стойности, за които трябва да се изпълни ПП и действителните резултати от действието на ПП. В момента на извикване на (обръщане към) ПП се извършва свързване на фактическите с формалните параметри. На всеки фактически параметър от списъка на фактическите параметри при извикване на ПП се съпоставя съответния формален параметър в описанието на ПП. Това изисква задължително формалните параметри да съответстват по брой, тип и място на фактическите параметри.

2. Подпрограма функция

Подпрограма функция изчислява и връща в мястото на извикването им в главната програма един скаларен резултат.

В Pascal се използват т.нар. вградени функции. Често, обаче се налага програмиста да разработва свои функции, които наричаме функции, дефинирани от потребителя или подпрограми функции (ПП-Ф).

Деклариране (описание) на ПП-Ф

Общият вид на описание на ПП-Ф е следния

FUNCTION *име(формални параметри):тип на резултата;*

Декларативна част за ПП-Ф

BEGIN

Изпълнителна част на ПП-Ф

END;

където

име – идентификатор според правилата на Pascal

формални параметри – това са данните, които се предават на функцията непосредствено преди да започне нейното изпълнение или се получават в резултат на изпълнението. За ПП-Ф се препоръчват формални параметри от вида *параметри-стойности*. Чрез тях се получават входни стойности за работата на ПП. Описанието им е от вида: *идентификатор:тип*. Формалните параметри се разделят с точка и запетая (;).

Декларативна част за ПП-Ф- подчинява се на правилата за декларативна част на главната програма, но областта на действие на описаните величини е само в рамките на ПП.

Изпълнителна част на ПП-Ф- оператори на Pascal, чрез които се реализира алгоритъма на ПП. В изпълнителната част на функцията трябва да има най-малко един оператор за присвояване, чрез който на името на функцията да се присвоява изчислената в тази част стойност

Връщането на резултата от изпълнението на ПП-Ф става чрез нейното име. Това налага две неща:

1. Да се определи *тип на резултата*.
2. В изпълнителната част на функцията трябва да има най-малко един оператор за присвояване, чрез който на името на функцията да се присвоява изчислената в тази част стойност.

Обръщение към ПП-Ф

Обръщението към ПП-Ф не се извършва с отделен оператор, а като се включи като операнд в израз в главната програма името на функцията със зададени фактически параметри.

При обръщане към ПП-Ф формалните параметри в описанието се заместват със съответните им фактическите параметри, зададени при обръщането към ПП-Ф; със стойностите на фактическите параметри се изчислява стойността на функцията, чрез работата на ПП-Ф и резултатът се връща в главната програма като стойност на функцията в мястото, от където е извикана ПП-Ф.

Изисква се да има съответствие между типа на формалните и фактическите параметри.

Пример: Да се състави програма за намиране стойността на израза $c=(a-b)^{(k-1)} + (ab)^l$, където a, b са реални числа a, b , k, l – цели числа.

Да се използва ПП-Ф, която да повдига произволно реално число, различно от 0, в произволна цяла степен (x^n).

```
PROGRAM PPF;
```

```
VAR
```

```
  A, B, C: REAL;
```

```
  K, L: INTEGER;
```

```
FUNCTION STEPEN(X:REAL; N:INTEGER):REAL;
```

```
{ ПП за повдигане на произволно число x в цяла степен n }
```

```
  VAR
```

```
    I : INTEGER; { брояч на цикъл }
```

```
    V: REAL; { частично и крайно произведение }
```

```
  BEGIN
```

```
    V:=1;
```

```
    FOR I:= 1 TO ABS(N) DO V:=V*X;
```

```
    IF N<0 THEN V:= 1.0/V;
```

```
    STEPEN := V {на името на функцията се присвоява стойност}
```

```
  END; {край на описанието на функцията}
```

```
BEGIN {главна програма}
```

```
  READLN (A, B);
```

```
  RADLN(K, L);
```

```
  C:= STEPEN(A-B, K-L) + STEPEN(A*B, L);
```

```
{функцията се извиква в израз}
```

```
  WRITELN(' C=', C:9:3)
```

```
END.
```

3. Подпрограма процедура

Подпрограмата процедура изчислява и връща в главната програма 0, 1 или повече резултата.

Деклариране (описание) на ПП-П

Общият вид на описание на ПП-П е следния

```
PROCEDURE име{(формални параметри)};  
Декларативна част за ПП-П  
BEGIN  
Изпълнителна част на ПП-П  
END;
```

Големите скоби тук показват, че формалните параметри не са задължителни.

име – идентификатор според правилата на Pascal

формални параметри – това са данните, които се предават на функцията непосредствено преди да започне нейното изпълнение или се получават в резултат на изпълнението. Формалните параметри се разделят с точка и запетая (;). Съществуват два вида формални параметри:

параметри-стойности - чрез тях се получават входни стойности за работата на ПП. Описанието им е от вида: *идентификатор:тип*.

параметри променливи – те могат да носят както входни стойности за процедурата, така и да приемат изходните стойности от работата на подпрограмата. Описанието на тези параметри е следното: **VAR** *идентификатор:тип*.

Декларативна част за ПП-П- подчинява се на правилата за декларативна част на главната програма, но областта на действие на описаните величини е само в рамките на ПП.

Изпълнителна част на ПП-П- оператори на Pascal, чрез които се реализира алгоритъма на ПП. В изпълнителната част на функцията трябва да има най-малко един оператор за присвояване, чрез който на името на функцията да се присвоява изчислената в тази част стойност.

Всички величини, описани в декларативната част на главната програма могат да се използват в операторната част на ПП-П. Не се препоръчва да се декларират величини с еднакви имена в главната програма и ПП-П.

Обръщение към ПП-П

Извикването на ПП-П в изпълнителната част на главната програма се извършва със следния оператор:

```
име(фактически параметри);
```

където

име - името на процедурата

фактически параметри – те трябва да отговарят на формалните по брой, тип и място. Разделят се със запетая (.). За формален *параметър-стойност* съответния фактически параметър може да бъде константа, променлива, израз. За формален *параметър-променлива* съответния фактически параметър може да бъде само променлива.

При извикване на ПП-П се преминава през следните етапи:

- Формалните параметри се заместват със съответните им фактически параметри. За *параметър-стойност* се отделя място в ОП, където тази стойност се записва. За *параметър-променлива* към него се подава само адреса от ОП, където е записан съответният му фактически параметър;
- Изпълнява се тялото на ПП-П;

- В главната програма се връщат резултатите от работата на ПП-П чрез изходните параметри на ПП-П (*параметър-променлива*).

Пример: A(M) и B(N) са масиви, които съдържат целочислени стойности. Да се намери броя на нечетните елементи за всеки масив.

```

PROGRAM MASIVI;
CONST MN=20;
TYPE MASIV=ARRAY[1..MN] OF INTEGER;
VAR A,B:MASIV;
M,N,BN:INTEGER;

{OPISANIE NA PODPROGRAMA PROCEDURA S IME VHOD}
PROCEDURE VHOD(VAR X:MASIV; VAR K:INTEGER);
VAR
I:INTEGER;
BEGIN
REPEAT
WRITE('BROJ ELEMENTI ');
READLN(K);
UNTIL (K>1) AND (K<=MN);
WRITELN('VAVEDI MASIV S ', K, ' ELEMENTA');
FOR I:=1 TO K DO READ (X[I]);
END;
{KRAJ NA OPISANIETO NA PODPROGRAMA PROCEDURA VHOD}

{OPISANIE NA PODPROGRAMA PROCEDURA S IME OBRAB}
PROCEDURE OBRAB(X:MASIV; K:INTEGER; VAR BR:INTEGER);
VAR
I:INTEGER;
BEGIN
BR:=0;
FOR I:=1 TO K DO
IF ODD(X[I]) THEN BR:=BR+1;
END;
{KRAJ NA OPISANIETO NA PODPROGRAM AOBRA B}

{GLAVNA PROGRAMA}
BEGIN
VHOD(A, M);
VHOD(B, N);
OBRAB(A, M, BN);
WRITELN('ZA MASIV A BROJ NECETNI ELEMENTI E ', BN);
OBRAB(B, N, BN);
WRITELN('ZA MASIV B BROJ NECETNI ELEMENTI E ', BN)
END.

```

Въпроси

1. Обяснете каква е разликата между проста и съствна програма.
2. Какво е подпрограма? Покажете примена структура на съставна програма.
3. Опишете предназначението, декларирането и обръщението към подпрограма функция.
4. Опишете предназначението, декларирането и обръщението към подпрограма процедура.