

ПОДПРОГРАМИ–ПРОЦЕДУРИ В ПАСКАЛ

Христина Михайлова Перфанова

Всеки преподавател по информатика си поставя за цел изграждането на умения за правилно и точно писане на програми, за приучаване на учениците към структурност в програмирането.

В разработката се предлага вариант за изучаване на подпрограмите - процедури в 2 учебни часа, като се изясняват основните понятия и особеностите при използването на формални и фактически параметри, на локални и глобални величини. Подбрани са подходящи задачи за работа в клас и самостоятелна работа, които допринасят за изграждане на умения за структуриране, ато прогресивен подход в програмирането.

Една от основните цели, които си поставя всеки преподавател при изучаването на езика Паскал е свързана с изграждането на умения у учениците за точно и прецизно описание на програмите, с приучаването им към структурност в програмирането, а това изисква задълбочено изучаване на подпрограмите.

Целта на моята разработка е да предложи вариант за изучаване на подпрограмите–процедури в рамките на 2 учебни часа. Първият час отделям за въвеждане, описание и извикване на процедурите и видове параметри. Целите, които си поставям с изучаването на този урок са: изграждане на умение за правилно описание на процедура и начина на извикването и в главната програма; запознаване с видовете параметри и тяхното действие след обръщението към нея.

Урокът е комбиниран, а дефинирането на новите понятия и затвърждаването им постигам с подходящо подбрани примери. В хода на урока изяснявам, че подпрограмата е логически завършена, относително самостоятелна програмна част, която има свое име и може да бъде извикана за изпълнение произволен брой пъти от различни места в главната програма.

Посочвам, че процедурата е именована група от оператори в програмата, реализираща определена част от общата задача.

Общият вид на описанието на една процедура е :

```
Procedure< име >(формални параметри);  
  <декларативна част>  
begin  
  <изпълнителна част>  
end;
```

Особено внимание обръщам на това че, описанието на една процедура започва със служебното име Procedure, след което следва нейното име и в малки скоби формалните параметри. Името на процедурата е идентификатор.

Параметрите са данните, които се предават на процедурата, преди да започне нейното изпълнение, или се получават в резултат от работата и.

Декларативната и изпълнителната част на процедурата не се различават по структура от частите в главната програма.

Изяснявам, че извикването на процедурата става чрез нейното име и списък на фактическите параметри. Коментирам с учениците различията между фактическите и формалните параметри и посочвам, че в езика Паскал се допускат два вида параметри: параметри-стойности и параметри-променливи.

Параметри стойности са входни параметри, като всеки представлява променлива, дефинирана и позната само в процедурата, но за разлика от обикновените променливи, те получават началната си стойност извън процедурата.

При обръщение към процедурата всеки формален параметър стойност може да бъде заместен с фактически параметър, който е константа, променлива или израз, имащ типа на съответния формален, т.е. извършва се “заместване” на формалните с фактическите параметри.

Трите стъпки на протичане на този процес са:

1. Пресмятат се стойностите на фактическите параметри; 2. Определя се памет за съответните формални параметри; 3. Записват се стойностите, определени в точка 1, в полетата на паметта, определена от т.2

Параметри променливи са входно-изходни параметри, с които могат да се “внесат” данни и да се “изнесат” резултати. За да се различават от параметрите – стойности те се декларират в списък, започващ със запазената дума VAR. Съответствието между формалните и фактическите параметри при обръщението към процедурата трябва да бъде пълно, т.е.: съответствие по брой, място и тип.

За да обясня как точно се осъществява обмена на информация между главната програма и подпрограмата чрез понятията формални и фактически параметри; параметри-стойности и параметри – променливи разглеждам следния пример:

Да се състави процедура, която намира средноаритметичното на две реални числа-а и b и се използва за намиране на средноаритметичното на всеки две реални числа.

```
Program sredno;
Var: x,y z:real; (*x,y- числа,въведени от потребителя,z-средноаритметично на x и y*)
procedure SR_aritm (a,b,z:real);
begin
    c:=(A+B)/2
end;
begin (*главна програма*)
    readln(x,y);
    SR_aritm(x,y,z);
    writeln ('z =',z:8:2)
end.
```

Посочвам, че извикването на процедурата за изпълнение в този случай не дава очаквания резултат - в променливата z не се съдържа средноаритметичното на числата x и y, тъй като: параметрите на процедурата A,B,C са обявени като параметри стойности, то в ОП се отделят клетки за тях; в клетките отделени за формалните параметри A,B се записват стойностите на съответните фактически параметри x и y;

В тялото на процедурата се изчислява стойността на C, но тази стойност не се предава към клетка Z, тъй като C е параметър стойност. Поради това при извеждането на съдържанието на Z върху екрана ще се извежда не резултатът от изчисленията извършени от процедурата, а случайна стойност.

Изяснявам, че ако променим описанието на процедурата SR_aritm, като обявим C за параметър променлива, т.е:

```
Procedure SR_aritm1 (A,B:real; var C:real);  
Begin  
    C:=(A+B)/2  
End;
```

при извикване на процедурата за изпълнение за A и B се отделят клетки от ОП, а за параметъра C не се отделя, тъй като той ще използва директно съдържанието на клетката, заета от съответния му фактически параметър – Z. Казва се, че на C се предава адресът на Z. Изчислява се стойността на C и тя става автоматично и стойност на Z. И сега вече при извеждането на съдържанието на клетка Z върху екрана ще бъде изведен действителния резултат, т.е. ако чрез механизма на параметрите трябва да “върнем” резултат от процедурата към главната програма, то това трябва да стане само чрез параметрите-променливи. Поставям за решаване следните задачи:

1. Да се опише процедура за намиране на корените на квадратното уравнение от вида $ax^2+bx+c = 0$, ако те са имагинерни да се считат за равни на 0.
2. Да се състави процедура за извеждане елементите на масив A с 10 елемента.

След коментара на решенията, където посочвам особеностите поставям задачи за самостоятелна работа:

1. Да се състави процедура за въвеждане на елементите на масив A с десет елемента..
2. Да се състави процедура за лицето на триъгълник по Хероновата формула с дадени три страни на триъгълника a,b,c.

В началото на следващия час обсъждам с учениците решенията им и обявявам темата на новия урок: ”Локални и глобални променливи”. Област на действие на променливите”, като си поставям следните цели: затвърдяване знанията за описание и извикване на процедура и взаимодействието между формални и фактически параметри; запознаване с понятията локална и глобална променлива и областта на действието им. При коментара на задачите обръщам внимание, че обмена на информация между главна програма и подпрограмата може да се осъществи чрез списъка на параметрите. Освен чрез тях това може да стане и чрез използването на локални и нелокални променливи.

Променливи, които се дефинират и използват само в дадена процедура, се наричат локални променливи. Те играят самостоятелна роля в процедурата. Ако програмните части имат само локални променливи, те не биха могли да обменят информация и цялата програма би представлявала съвкупност от отделни програмни части изпълнявани в определен ред, но не използващи резултатите на изпълнените преди това програмни части.

Един от начините за обмен на информация между програмните части е посредством използването на глобални променливи. За локални променливи могат да се избират произволни имена. Едно и също име на променлива може да се използва в различни програмни части. Анализирам примерите:

1. Program A;	2. Program B;
Var x: integer;	Var x: integer;
Procedure P1;	Procedure P2;
	Var x: integer;

<pre> Begin X:=1; (* X-глобална *) End; Begin X:=0; P1; Writeln(x) End. Резултатът е 1.</pre>	<pre> Begin X:=1; (*X –локална*) End; Begin X:=0; P2; Writeln(x) End. Резултатът е 0.</pre>
---	---

В програмата А променливата X е глобална за процедурата P1, поради което след обръщението към P1 променливата X променя стойността си от 0 на 1.

В програмата В променливата X е локална за процедурата P2, поради което промяната над променливата X в P2 не оказва влияние върху променливата X, дефинирана в програмата В.

Обръщам внимание на това, че взаимодействието между програмните части чрез глобални променливи има сериозни недостатъци. Един от тях е силната зависимост на процедурата от програмната част, чийто променливи използва. Един начин да се отстрани тази слабост е да се повиши относителната независимост на процедурата спрямо останалите е вместо глобални променливи да се използват параметри т.е:

<pre> 3. Program C; Var L,n:integer; Procedure PRINTC: Var I:integer; Begin For I:=1 to n do Write(' '); End; Begin For L :=1 to 5 do Begin N:=I; PRINTC: Writeln; End End.</pre>	<pre> 4. Program D; Var L:integer; Procedure PRINTD(n:integer); Var I:integer; Begin For I:=1 to n do Write(' '); End; Begin; For L:=1 to n do Begin PRINT(L): Writeln; End; End.</pre>
---	---

Коментирам, че в програмата С, N е глобална за процедурата PRINTC, а в програмата D, N е формален параметър. Различен е начинът на обръщение в двата случая. Във втория случай L играе и ролята на фактически параметър за процедурата PRINTD. Обръщението към тези процедури ще се извърши 5 пъти, но процедурата PRINTD има завършен вид и е напълно независима от останалата част на програмата .

Поставям за решаване следните задачи:

1. Да се използва процедура за въвеждане и извеждане елементите на едномерен масив А от 10 реални елемента и процедура за намиране на максималния елемент на масива и описанията им да се включат в програма и да се изпълнят.

Чрез нея затвърдявам знанията за мястото на описанието и мястото на извикването на една процедура; начина на задаването на двата вида формални параметри и взаимодействието между тях.

2. Да се състави програма за изчисляване стойността на функцията:

$$Z = \frac{(t - y_1)(t - y_2)}{(v - x_1)(v - x_2)}$$

където x_1 и x_2 са корени на уравнението $tx^2 - x + 1 = 0$, а y_1 и y_2 са корени на уравнението $2y^2 + y - v = 0$, като се използва процедура

3. Да се състави програма, която изчислява лицето на изпъкнал четириъгълник ABCD, зададен с дължините на страните си AB, BC, CD и DA и диагонала AC.

В заключение мога да направя следните изводи: 1. Всяка процедура има заглавен оператор, последван от името на процедурата и от списък от параметри; 2. Процедурите могат да връщат повече от една стойност и нямат тип; 3. Всеки програмен обект има област на действие, в която е познат и достъпен; 4. Ако една програмна част съдържа в блока си друга програмна част, тогава втората е вложена в първата; 5. Програмните обекти, декларирани в дадена програмна част са локални за нея;

ЛИТЕРАТУРА

1. Азълов П., Програмиране и алгоритмични езици (Azalov P., Programirane i algorit-michni ezici), 1990г.;
2. Тодорова М., Програмиране на Паскал, (Todorova M., Programirane na Paskal), 1995 г.;
3. Сандалски М., Информатика (Sandalski M., Informatika);
4. Чобанова Д., Турбо Паскал за начинаещи-I и II част (Chobanova D., Turbo Paskal za nachinaesti I, II chast)

гр. Смолян, ул. "Деспот Слав" №10, тел. 0301/2-29-26