

14 Eljárások (alprogramok), függvények

- eljárások (**procedure**)
- változók hatásköre
- függvények (**function**)

14.1 Eljárások (procedure)

A programozás során gyakran előfordulhat, hogy valamilyen programrészt a programunkban többször végrehajtunk. Például a tömböknél kiírtuk a tömböt, változtattunk rajta valamit majd ismét kiírtuk a képernyőre. Ilyenkor egyszerűbb a tömb kiírására egy eljárást (alprogramot) írni és kiírásnál csak ezt az eljárást meghívni. A programunk ebben az esetben így nézhet ki:

```

program Pelda34;
uses crt;
const n=10;
var i:integer;
    a:array[1..n] of integer;

procedure kiir;
begin
  write('A tömb elemei: ');
  for i:=1 to n do write(a[i], ' ');
  writeln;
end;

begin
  clrscr;
  { tömb kigeneralasa: }
  for i:=1 to n do a[i]:=random(99)+1;
  { tömb kiirasa alprogrammal: }
  kiir;
  { tömb megváltoztatasa, pl. mindegyik elemet megszorozzuk 2-vel: }
  for i:=1 to n do a[i]:=2*a[i];
  { tömb kiirasa ismet alprogrammal: }
  kiir;
  readln;
end.

```

Az eljárást (alprogramot) mindig a **procedure** szóval kezdjük írni még a főprogram kezdete előtt, a változók deklarálása után. A **procedure** szó után megadjuk az eljárás nevét, majd a következő sortól írhatjuk az eljárást **begin** és **end**; közé. Az eljárást a főprogramból egyszerűen az eljárás nevével hívhatjuk meg. Egy programba írhatunk több eljárást is, pl. egyet a tömb generalására, egyet a kiírására, stb.

Az eljárásunknak átadhatunk a főprogramból valamilyen értékeket is paraméterek segítségével. Például, ha szeretnénk készíteni egy olyan eljárást, amely kiír valamilyen szöveget úgy, hogy minden betűjét a szövegnek megváltoztatja nagy betűre, akkor a kiírandó szöveget átadhatjuk az eljárás paraméterében. Ekkor a programunk így nézhet ki:

```

program Pelda35a;
uses crt;
var s:string;

procedure nagybetukkel(m:string);
var i:integer;
begin
  for i:=1 to length(m) do m[i]:=upcase(m[i]);
  writeln(m);
end;

begin
  clrscr;
  nagybetukkel('Kerek egy szoveget: ');
  readln(s);
  nagybetukkel(s);
  writeln('S erteke a program vegen: ',s);
  readln;
end.

```

Ez program kiírja nagy betűkkel, hogy **KEREK EGY SZOVEGET:** ", majd bekér egy mondatot és ezt kiírja nagy betűkkel. Végül még kiírja az **s** értékét (eredeti szöveget). Az eljárásban paraméterként szereplő **m** változót csak az eljárásán belül használhatjuk, máshol nem érhető el. Hasonlóan az eljárásban deklarált **i** változót is csak az eljárásán belül használhatjuk - ezek úgynevezett lokális (helyi) változók.

Az eljárás az **s** változó értékét nem változtatja meg, csak a helyi **m** változóét. Tehát az eljárás lefutása után az **s** változóban marad az eredeti, kisbetűs szöveg (ez kiíródik a főprogram végén). Az ilyen **m** változót nevezzük formális paraméternek.

Ha valami végett mégis olyan eljárást szeretnénk írni, amely az **s** változó értékét is megváltoztatja (tehát azt szeretnénk, hogy az eljárás lefutása után az **s** változóban is a nagybetűs szöveg legyen - ugyanaz mint az eljárásán belül az **m** változóban), akkor az eljárásunk paraméterének megadásánál az **m** változó előtt használhatjuk a **var** utasítást. Az ilyen paramétert nevezzük valódi paraméternek.

```

program Pelda35b;
uses crt;
var s:string;

procedure nagybetukkel(var m:string);
var i:integer;
begin
  for i:=1 to length(m) do m[i]:=upcase(m[i]);
  writeln(m);
end;

begin
  clrscr;
  s:='Kerek egy szoveget: ';
  nagybetukkel(s);
  readln(s);
  nagybetukkel(s);
  writeln('S erteke a program vegen: ',s);
  readln;
end.

```

Eljárásunkban használhatunk több változót is, melyek közül lehet némelyik formális paraméter, némelyik valódi paraméter. Például egy eljárást megadhatunk ilyen paraméterezéssel is:

```
procedure szamol(s:string; a,b:integer; var c:integer);
```

Ebben az esetben csak a **c** valódi paraméter, tehát a főprogramban az eljárás lefutása után csak a **c** helyén megadott változó értéke változik meg.

14.2 Változók hatásköre

A programozás során használhatunk **globális változókat** és **lokális (helyi) változókat**.

A **globális változók** azok a változók, melyet a programunkban bárhol elérhetünk, bárhol használhatjuk - a főprogramba is és az eljárásban is. A globális változókat a programunk elején a **var** szó után soroljuk fel, ahogy eddig is tettük.

A **lokális (helyi) változók** azok a változók, melyek a főprogramban nem érhetők el, csak az eljárásban. Tehát ezeket a változókat csak az adott eljárásban használhatjuk, a főprogramban nem. Ezeket szintén a **var** utasítás segítségével adjuk meg, de nem a programunk elején, hanem abban az eljárásban, melyben ezeket használni szeretnénk.

Ha ugyanolyan nevű változót deklarálunk globális és lokális változóként is, akkor az eljárásban csak a lokális (helyi) változóval tudunk dolgozni, az eljáráson kívül pedig a globális változóval. Az eljárás nem fogja megváltoztatni a globális változó értékét. Példaként vizsgáljuk meg és próbáljuk ki ezt a programot:

```
program Pelda36;
uses crt;
var v:integer;

procedure alprg;
var v:integer;
begin
v:=888;
writeln('Lokalis V nevu valtozo erteke: ',v);
end;

begin
clrscr;
v:=555;
writeln('Globalis V nevu valtozo erteke: ',v);
alprg;
writeln('Globalis V nevu valtozo erteke: ',v);
readln;
end.
```

A program kiírja először az 555-t, majd a 888-at végül ismét az 555-t.

14.3 Függvények (function)

A Pascalban vannak előre definiált függvények, ilyen például az **abs()**, **sin()**, **cos()**, **upper()**, **length()**, stb. Készíthetünk mi is saját függvényt. Ezt hasonlóan kell megírunk mint a saját eljárásunkat, a különbség csak abban van, hogy a **procedure** szó helyett a **function**-t használjuk, és meg kell adnunk hogy milyen típust adjon vissza az általunk készített függvény. Továbbá a függvényünkön belül (általában a végén) egy ilyen típusú értéket kell adunk a függvénynek (függvény nevének).

Példaként készítsünk egy függvényt, amely megszámolja, hogy egy megadott szövegben mennyi szóköz van és ezt a számot adja vissza függvényértékként:

```
program Pelda37;
uses crt;
var s:string;

function helyekszama(x:string):integer;
var i,h:integer;
begin
  h:=0;
  for i:=1 to length(x) do
    if x[i]=' ' then h:=h+1;
  helyekszama:=h;
end;

begin
  clrscr;
  write('Írj be egy mondatot: ');
  readln(s);
  writeln('A mondatban ',helyekszama(s),' szóköz van. ');
  readln;
end.
```