



Nyelvi alapfogalmak

2. előadás





Nyelvi alapfogalmak



Típus

típus=értékhalmoz+műveletek (lehet statikus és dinamikus)

- típus nélküli (az azonosítóihoz nem rendelünk típust) – értékhalmoz ellenőrzés csak végrehajtáskor,
- típusos (futási időben kell a típus hozzárendelés) – lehetséges a típusparaméterezés változóval is, lehetséges a változóktól függő kezdőérték-adás is; lehetséges a típus megváltoztatás is;
- erősen típusos nyelv (fordítási időben kell a típus hozzárendelés)





Nyelvi alapfogalmak



Típus

- statikus típus
- dinamikus típus
- típus hozzárendelés:
 - deklarációban,
 - névhez kötötten (pl. BASIC: A\$ - szöveg típusú lesz),
 - első értékadáskor.





Nyelvi alapfogalmak



Típus

- altípus: az eredetivel ekvivalens típus, annak valamilyen része, műveleteit örökli (nem csak intervallum)

Pascal példa:

```
type Intervallum=1..90;
```

Ada példa:

```
subtype Tizedes is delta 0.1;
```





Nyelvi alapfogalmak



Típus

- altípus: az eredetivel ekvivalens típus, annak valamilyen része, műveleteit örökli (nem csak intervallum, nem csak elemi típusra)

Ada példa:

```
subtype Rendszer is String(1..6);
```

```
subtype Csalados is ember
```

```
(Csaladiallapot => nő, férjezett);
```





Nyelvi alapfogalmak



Típus

- származtatott típus: az eredetivel nem ekvivalens típus, lehet annak része, de az egész értékhalmoz is, műveleteit örökölheti, de van nyelv, ahol csak részben

Pascal példa:

```
type hossz=real;
```

Ada példa:

```
type darab is new Natural;
```





Nyelvi alapfogalmak



Típus

- névtelen típus (struktúrája van, neve nincs)

Pascal példa:

```
var x=array[1..100] of real;
```

Ilyenkor az `x` típusának nincs neve. Az ilyen változók semmilyen más típusúval sem kompatibilisek, ha névszerinti típusmegfeleltetés van.





Nyelvi alapfogalmak



Típus

Mikor tekinthetünk két típust ekvivalensnek? (Azaz mikor végezhető el az $A:=B$ értékadás?)

- típuskompatibilitás (név szerint, struktúra szerint)
 - két típus azonos, ha a nevük azonos,
 - két típus azonos, ha a szerkezetük azonos.
- kivételek:
 - minden típus ekvivalens az őstípusával
 - `valós := egész` értékadás sokszor lehetséges
 - `szöveg := karakter` értékadás sokszor lehetséges





Nyelvi alapfogalmak



Példa:

```
Type Komplex=Record re,im: Real End;  
      Pont    =Record x,y: Real End;  
Var   C: Komplex;  
      P: Pont;  
  
...  
C:=D;
```

Kérdés: szabályos-e az értékadás?





Nyelvi alapfogalmak



Típus

Mi történik, ha egy kifejezésben különböző típusú tényezők szerepelnek?

➤ típuskényszerítés (automatikus konverzió)

Példa:

```
Var I, J: Integer;
```

```
    X, Y: Real;
```

```
    ...
```

```
X := Y * I / J;
```

```
X := I / J * Y;
```





Nyelvi alapfogalmak



Típus

- paraméteres típus (paraméter lehet: konstans, típus – a kód másolása, vagy valódi paraméterezés), feltételezett paraméterértékek, feltételezett típus-kezdőértékek

Elan példa (típussal paraméterezés):

```
type binfa(elemtípus)=struct(  
    elem: elemtípus,  
    bal, jobb: binfa(elemtípus));  
binfa(int) var a;
```





Nyelvi alapfogalmak



Típus

- paraméteres típus (paraméter lehet: konstans, típus – a kód másolása, vagy valódi paraméterezés), feltételezett paraméterértékek, feltételezett típus-kezdőértékek

Ada példa (konstanssal paraméterezés):

```
type Szoveg is Array (Natural range <>)
  of Character;
s: Szoveg(5);
```





Nyelvi alapfogalmak



Típus

- átlátszatlan típus (csak a neve látszik kifelé)

Ada példa:

```
package Verem is
    type Veremtípus is private;
private
    type Veremtípus is record
        M: Integer;
        V: Array (1..100) of Elemtípus;
    end record;
    ...
end Verem;
```





Nyelvi alapfogalmak



Paraméter megfeleltetés

- sorrend szerint
`procedure valami (a, b) ;`
...
`valami (1, 2) ;`
- név szerint
`procedure valami (a, b) ;`
...
`valami (b=1, a=2) ;`





Nyelvi alapfogalmak



Paraméter feltételezett értéke (változó számú paraméter)

Ada példa:

```
function Hatvány(X: real; Kitevő: integer:=2)
    return real is
```

...

```
Y:=Hatvány(5); // Y:=52
```

...

```
Z:=Hatvány(Y, 3); // Z:=Y3
```

➤ típus nélküli paraméter





Nyelvi alapfogalmak



Paraméterátadás

Érték szerint: a hívott megkapja az aktuális paraméter értékét egy, a hívott futási idejére lefoglalt memóriaterületen.

Pascal példa:

```
Procedure Név (I: Integer; X: Tömb);
```

Jellemzők:

- A paraméter értéke nem változhat meg (bár a később felszabaduló memóriaterületre írhatunk)
- Az aktuális paraméter lehet: konstans, változó, kifejezés.
- Paraméterátadáskor mindent lemásolunk, a tömböket is.





Nyelvi alapfogalmak



Paraméterátadás

Cím szerint: a hívott a paraméter memóriabeli címét kapja meg.

Pascal példa:

```
Procedure Név (var I: Integer; var X: Tömb);
```

Jellemzők:

- A paraméter értéke lekérdezhető és módosítható is.
- Az aktuális paraméter csak változó lehet.





Nyelvi alapfogalmak



Paraméterátadás

Név szerint: a hívott a paraméter memóriabeli címét kiszámító függvény címét kapja meg.

Jellemzők:

- A paraméter értéke lekérdezhető és módosítható is.
- Az aktuális paraméter csak változó lehet.
- Ha az eljárás a paraméterre többször hivatkozik, akkor a címét többször is kiszámítja (ha egyszer sem, akkor persze egyszer sem számítja ki).





Nyelvi alapfogalmak



Paraméterátadás

ALGOL60 példa, név szerint:

```
real procedure össze(i,x); integer i; real x;  
  begin real s;  
    s:=0;  
    for i:=1 step 1 until n do s:=s+x;  
    össze:=s;  
  end;  
y:=össze(i,A[i]);
```





Nyelvi alapfogalmak



Példa a név-, és a címszerinti paraméterátadás problémáira

Névszerinti paraméter esetén:

```
Procedure Csere(X,Y: Valami);  
    Var Z: Valami;
```

```
Begin
```

```
    Z:=X; X:=Y; Y:=Z;
```

```
End;
```

```
...
```

```
i:=1; A[1]:=5; Csere(i,A[i]);
```

Hatása $X=i$, $Y=A[i]$ helyettesítéssel: $i=5$, $A[5]=1$.





Nyelvi alapfogalmak



Példa a név-, és a címszerinti paraméterátadás problémáira

Címszerinti paraméter esetén:

```
Procedure Csere (X, Y: Valami)
```

```
Begin
```

```
    X := X + Y; Y := X - Y; X := X - Y;
```

```
End;
```

```
...
```

```
i := 1; A[1] := 5; Csere (A[i], A[i]);
```

Hatása X és Y azonos memóriacíme miatt: $A[1]=0$.





Nyelvi alapfogalmak



Más elv szerint: bemenő, kimenő, vagy be- és kimenő:

Ada példa:

```
Procedure Push(in out V: Verem;  
              in X: Elemtípus) is ...
```

Más elv szerint: konstans vagy változó:

Pascal példa:

```
Procedure Push(var V: Verem;  
              const X: Elemtípus);
```





Nyelvi alapfogalmak

2. előadás vége