



Programozási nyelvek a
közoktatásban
7. 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 értékalmaz 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 – a típusfogalom fejlődése



➤ Gépi szint:

- regiszterek,
- byte-ok, 2- vagy 4-byte-os egységek;
- verem, veremkezelő utasítások,
- feltételregiszter
- azonosító = memória kezdőcím,
- kódolással tisztában kell lenni (kettes komplement, lebegőpontos valós, szöveg, logikai érték),
- nincs konstans – minden memóriacím tartalma megváltoztatható,

Memória



↑ objektum kezdőcím

Szlávi Péter – Zsakó László:

Programozási nyelvek a közoktatásban.

2015.04.14

14/22





Nyelvi alapfogalmak – a típusfogalom fejlődése



Az első magasszintű nyelvek

- Itt a **típus** nem más, mint a **változó reprezentációjának** (értsd ez alatt: memóriába való leképezésének) **a rögzítése.**
- Kényelmesebben leírhatók az adatok, előredefiniált, *elemi* készletből választhatók ki típusaik.
- Megjelenhetnek az összetett típusok – ennek következménye a deklaráció.

Memória



↑
objektum



Szlávi Péter – Zsakó László:

Programozási nyelvek a közoktatásban.

2015.04.14

15/22



Nyelvi alapfogalmak – a típusfogalom fejlődése



FORTRAN

- egész, valós, duplapontos, logikai, komplex
- tömb, ekvivalencia
- implicit deklaráció: egyes típusoknál az azonosító első betűje eldönti a változó hovatartozását (i,j,k,l,m,n – egész)
- explicit deklaráció: ahol automatikusan nem dől el, illetve a tömböknél

BYTE	SZEMÉLY (30)
LOGICAL	NEM
INTEGER	DÁTUM (3)
EQUIVALENCE	(SZEMÉLY (1) , NEM) , (SZEMÉLY (2) , KOR)



Szlávi Péter – Zsakó László:

Programozási nyelvek a közoktatásban.

2015.04.14

16/22



Nyelvi alapfogalmak – a típusfogalom fejlődése



COBOL

A Cobol-ban a változóknak nincs típusa, van viszont egy maszkja (PIC), hogy milyen értékek tárolhatók benne.

A fontosabb elemek:

- A az angol ABC betűi vagy a szóköz.
- B szóköz.
- V tizedespont helye az adaton belül (a tizedespont nem tárolódik).
- S az adat előjeles.





Nyelvi alapfogalmak – a típusfogalom fejlődése



COBOL

- P tizedespont helye az adaton kívül, és "P" helyén 0-k állnak. Ezek a 0-k nem tárolódnak.
- X alfanumerikus karakter.
- 9 számjegy

Példa:

egyik PIC X(4).

másik PIC S99.





Nyelvi alapfogalmak – a típusfogalom fejlődése



COBOL

- Egyetlen típuskonstrukciós eszköz a rekord.
- A rekord mezőit szintenként adjuk meg – a 01-es szintűek a rekordok.

01 születés.

02 datum.

03 év PIC 9(4).

03 hó PIC 9(2).

03 nap PIC 9(2).

02 hely PIC A(40).





Nyelvi alapfogalmak – a típusfogalom fejlődése



A típusfogalom fejlődése

- BASIC: típus = memóriareprezentáció (vö. névtelen típus), csak elemiek érhetőek el, összetett típusok, kezdőérték
- FORTRAN: tömb, unió
- ALGOL,... : összetettek is elérhetőek
- COBOL,... : rekord, file
- Pascal: típus = memóriába képezés szabálya, definiálható a struktúra





Nyelvi alapfogalmak – a típusfogalom fejlődése



A típusfogalom fejlődése

- ELAN: elemi típus, halmaz, dinamikus típusok, konstans, operátor
- Modula-2 : típus = szerkezet + műveletek, a modul, mint a típusalkotás eszköze
- Ada: implementáció és reprezentáció elrejtése
- Osztályok: típusfogalom (struktúra és műveletek), típusalkotás újdonsága – származtatás





Programozási nyelvek a
közoktatásban
7. előadás vége

