







A PROGAMOZÁS ALAPJAI 1

5. előadás

Vitéz András
egyetemi adjunktus
BME Híradástechnikai Tanszék
vitez@nit.bme.hu



2012. március 6.



Miről lesz ma szó?

- Típusnév hozzárendelés
- Struktúrák
 - Deklarációja
 - Tárolása
 - Használata
- Operátorok
 - Operátorok precedenciája
 - A C nyelv operátorai
- Mellékhatások szinkronizálása
- Aritmetikai típusok konverziója
 - Egy operandusú
 - Két operandusú
- Típuskonverziók

A programozás alapjai 1 5. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem



Típusnév hozzárendelés

`typedef <típus> <azonosító>;`

A typedef nem hoz létre új típust, csak egy "álcnevet" az adott típus számára a kényelmesebb kezelhetőség érdekében.

Az új típusnévhez rendelt változók típusa az "álcnévvel" azonosított eredeti típus marad.

A programozás alapjai 1 5. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem



Típusnév hozzárendelés

Példa:

```
typedef long hosszu;
long l1;
hosszu l2;
l2 = l1;
```

A programozás alapjai 1 5. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem



A felhasználó által definiálható típusok

Felsorolás	<code>enum</code>
Struktúra	<code>struct</code>
Bitmezők	<code>struct</code>
Unió	<code>union</code>

A programozás alapjai 1 5. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem



Struktúrák

struktúra
(konglomerátum)

logikailag egy egységet alkotó
akár különböző típusú adatokból álló
összetett adattípus

A programozás alapjai 1 5. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Példa:

Dátum

(év,hónap,nap)

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Példa:

Sakkjátszma

(világos,sötét,mikor,hol,lépések,eredmény)

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Példa:

Lépés

(figura,honnan,hová)

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Példa:

Mező

(oszlop,sor)

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Struktúra deklaráció alakja:

```
struct [<struktúra címke>]
{
    <struktúra tag deklarációk>
}
[<struktúra változó azonosítók>];
```

opcionális

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Példa:

```
typedef int honap;
struct datum {
    int ev;
    honap ho;
    short nap;
} d1,d2;

struct datum d3,d4;
```

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Példa:

```
typedef struct {
    int ev;
    honap ho;
    short nap;
} datum_t;

datum_t d1,d2;
```

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Struktúra tag hivatkozás alakja:

<struktúra azonosító>.<tag azonosító>
<struktúra mutató>-><tag azonosító>

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Példa:

```
typedef struct {
    int ev;
    honap ho;
    short nap;
} datum_t;

datum_t d1,*d2;

d2 = &d1;
d1.ho = oktober;
d2->nap = 19;
```

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Belső ábrázolása:

Az egyes tagoknak megfelelő adatok egymás után tárolva

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Szóhatárra illesztett tárolás:

A tárolt adatok elérése legyen gyors

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Tömörített tárolás:

A tárolt adatok között ne maradjon ki hely

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

1. tag

2. tag

3. tag

Szóhatárok

Szóhatárra illesztett tárolás

Tömörített tárolás

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Tagok azonosítójának jelentése

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Megjegyzések:

A struktúra tagok azonosítója megegyezhet változók azonosítójával, mert a környezetből mindig kiderül, hogy melyikről van szó.

Struktúra típusú változó értéke frissíthető egyetlen értékadással.

Függvény paramétere lehet struktúra típusú.

Függvény visszatérési értéke is lehet struktúra.

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Struktúrák

Megjegyzések:

Éppen ezek a tulajdonságok teszik a **struktúrát** a C nyelv leghatékonyabb típusává.

Struktúra típusú változó értéke frissíthető egyetlen értékadással.

Függvény paramétere lehet struktúra típusú.

Függvény visszatérési értéke is lehet struktúra.

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Komplex mintapélda

Definiáljunk adattípust síkbeli vektor ábrázolására!

Készítsünk függvényt, amely kiírja a vektor összetevőit!

Készítsünk függvényt, amely meghatározza két vektor eredőjét!

Készítsünk függvényt vektor nyújtására!

Próbáljuk ki függvényeinket!

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Komplex mintapélda

```
#include <stdio.h>
typedef struct{double x,y;}vektor;
void v_kiir(vektor a){
    printf("(%.2lf;%.2lf)",a.x,a.y);
}
vektor eredo(vektor a,vektor b){
    vektor c;
    c.x=a.x+b.x; c.y=a.y+b.y;
    return c;
}
vektor nyujt(vektor a,double k){
    a.x=k*a.x; a.y=k*a.y;
    return a;
}
int main(){
    vektor v1={2.1,4.3},v2={1.8,-1.5},v3;
    printf("v1="); v_kiir(v1); printf("\n");
    printf("v2="); v_kiir(v2); printf("\n");
    printf("v1+v2="); v_kiir(eredo(v1,v2)); printf("\n");
    v3=nyujt(v1,2.2);
    printf("2.2*v1="); v_kiir(v3);printf("\n");
    return 0;
}
```

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

- Precedencia
Melyik hajtódik először végre?
 $2+3*4$
- Asszociativitás
Balról jobbra vagy jobbról balra köt?
 $11-8-2$
 $a=b=2$

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok precedenciája

$() [] . \rightarrow$
 $! \sim ++ -- + - * \& (<típus>) \text{sizeof}$
 $* / \%$
 $+ -$
 $<< >>$
 $< <= > >=$
 $== !=$

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok precedenciája

$\&$
 $\wedge$
 $|$
 $\&\&$
 $||$
 $?:$
 $= += -= *= /= \% = \&= \wedge= |= <= >=$
 $,$

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok precedenciája

A precedencia zárójelezéssel megtörhető!

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok precedenciája

Az operandusok kiértékelési sorrendje
nem meghatározott!
A mellékhatások kiszámíthatatlanok!

Megoldás:

- zárójelezés
- segédváltozók használata

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

Minden igazságérték jellegű eredmény
 int típusú,
és értéke

0, ha hamis
1, ha igaz

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

 **Operátorok**

függvényhívás

$\langle \text{függvény} \rangle (\langle \text{paraméterek} \rangle)$

Pl.:
 $\text{atlag}(b, 3*a, -5)$

A programozás alapjai 1
6. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

 **Operátorok**

tömbhivatkozás

$\langle \text{tömb} \rangle [\langle \text{index} \rangle]$

Pl.:
 $\text{sor}[i-4]$

A programozás alapjai 1
6. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

 **Operátorok**

struktúrahivatkozás

$\langle \text{struktúra} \rangle . \langle \text{tag} \rangle$
 $\langle \text{struktúra mutató} \rangle - \> \langle \text{tag} \rangle$

A programozás alapjai 1
6. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

 **Operátorok**

logikai tagadás

$! \langle \text{kifejezés} \rangle$

Pl.:
 $a: 01101010 \quad b: 00000000$
 $!a: 00000000 \quad !b: 00000001$

A programozás alapjai 1
6. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

 **Operátorok**

“1-es” komplementálás
 (bitenkénti negálás)

$\sim \langle \text{kifejezés} \rangle$

Pl.:
 $a: 01101010$
 $\sim a: 10010101$

A programozás alapjai 1
6. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

 **Operátorok**

utótagos inkrementálás/dekrementálás

$\langle \text{hivatkozás} \rangle ++$
 $\langle \text{hivatkozás} \rangle --$

felhasznál, majd módosít

A programozás alapjai 1
6. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

előtagos inkrementálás/dekrementálás

`++<hivatkozás>`

`--<hivatkozás>`

módosít, majd felhasznál

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

egyoperandusú plusz/mínusz

`+<aritmetikai kifejezés>`

`-<aritmetikai kifejezés>`

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

indirekció

`*<mutató>`



A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

címképzés

`&<azonosító>`



A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

kényszerített típusmódosítás (casting)

`(<típus>)<kifejezés>`

Konverzió csak aritmetikai típusoknál van!

Pl.:

`5/(double)2`

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

tárolás helyigénye (karakterekben)

`sizeof<kifejezés>`

`sizeof(<típus>)`

A kifejezés nem értékelődik ki!
Casting esetén elég az operátor is!

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

multiplikatív operátorok

$\langle \text{kifejezés} \rangle * \langle \text{kifejezés} \rangle$
 szorzás
 $\langle \text{kifejezés} \rangle / \langle \text{kifejezés} \rangle$
 bennfoglalás vagy osztás !!!
 $\langle \text{kifejezés} \rangle \% \langle \text{kifejezés} \rangle$
 maradékképzés
 Az eredmény típusát az operandusok
 típusa határozza meg!

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

additív operátorok

$\langle \text{kifejezés} \rangle + \langle \text{kifejezés} \rangle$
 $\langle \text{kifejezés} \rangle - \langle \text{kifejezés} \rangle$

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

léptető operátorok

ezt a bitmintát $\langle \text{egész} \rangle \ll \langle \text{nemnegatív egész} \rangle$ ennyi bittel
 $\langle \text{egész} \rangle \gg \langle \text{nemnegatív egész} \rangle$

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

léptető operátorok

$\langle \text{egész} \rangle \ll \langle \text{nemnegatív egész} \rangle$
 $\langle \text{egész} \rangle \gg \langle \text{nemnegatív egész} \rangle$
 Negatív bal operandus esetéről
 a szabvány nem rendelkezik!

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

relációs operátorok

$\langle \text{kifejezés} \rangle < \langle \text{kifejezés} \rangle$
 $\langle \text{kifejezés} \rangle \leq \langle \text{kifejezés} \rangle$
 $\langle \text{kifejezés} \rangle > \langle \text{kifejezés} \rangle$
 $\langle \text{kifejezés} \rangle \geq \langle \text{kifejezés} \rangle$

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

egyenlőségi operátorok

$\langle \text{kifejezés} \rangle == \langle \text{kifejezés} \rangle$
 $\langle \text{kifejezés} \rangle != \langle \text{kifejezés} \rangle$

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

bitenkénti és

$\langle \text{egész kifejezés} \rangle \& \langle \text{egész kifejezés} \rangle$

Pl.:

a :	1 0 1 0 1 1 0 0
b :	0 1 1 0 1 0 0 1
a & b :	0 0 1 0 1 0 0 0

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

bitenkénti kizáró vagy (antivalencia)

$\langle \text{egész kifejezés} \rangle \wedge \langle \text{egész kifejezés} \rangle$

Pl.:

a :	1 0 1 0 1 1 0 0
b :	0 1 1 0 1 0 0 1
a ^ b :	1 1 0 0 0 1 0 1

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

bitenkénti vagy

$\langle \text{egész kifejezés} \rangle | \langle \text{egész kifejezés} \rangle$

Pl.:

a :	1 0 1 0 1 1 0 0
b :	0 1 1 0 1 0 0 1
a b :	1 1 1 0 1 1 0 1

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

logikai és

$\langle \text{egész kifejezés} \rangle \&\& \langle \text{egész kifejezés} \rangle$

Az operandusok **balról jobbra** értékelődnek ki!
Ha az első kifejezés **HAMIS**,
a második **NEM ÉRTÉKELŐDIK KI**,
mellékhatásai elmaradnak!

Pl:

`if(a && b/a>3)` `if(i<n && t[i]>3)`

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

logikai vagy

$\langle \text{egész kifejezés} \rangle || \langle \text{egész kifejezés} \rangle$

Az operandusok **balról jobbra** értékelődnek ki!
Ha az első kifejezés **IGAZ**,
a második **NEM ÉRTÉKELŐDIK KI**,
mellékhatásai elmaradnak!

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

feltételes kifejezés

$\langle \text{kifejezés} \rangle ? \langle \text{kifejezés} \rangle : \langle \text{kifejezés} \rangle$

Csak az egyik kifejezés értékelődik ki!
A kifejezés típusa a **kétooperandusú konverzió**
szabályai szerint határozódik meg!

Pl:

`a = b>5 ? 13 : b-2;`

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

értékadó operátorok (jobbról balra!)

= += -= *= /= %=

&= ^= |= <<= >>=

<hivatkozás>= <kifejezés>

A hivatkozott változónak
módosíthatónak kell lennie!

A kifejezés értéke a módosított változó új értéke.
Ez teszi lehetővé, hogy... a=b=c;

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

értékadó operátorok (jobbról balra!)

= += -= *= /= %=

&= ^= |= <<= >>=

<hivatkozás> <op>= <kifejezés>

≈ <hivatkozás>= <hivatkozás> <op> <kifejezés>

A hivatkozás csak egyszer értékelődik ki!

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operátorok

vessző

<kifejezés>, <kifejezés>

Az operandusok balról jobbra értékelődnek ki!

Az első kifejezés értékét eldobjuk!

A kifejezés értéke és típusa a második kifejezés
értéke illetve típusa lesz!

Ahol a vessző értelme más, zárójelezni kell!

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Mellékhatások szinkronizálása

Sorrend-határ (sequence point):

A program végrehajtásának azon pontja, ahol

- minden előzőleg végrehajtott tevékenység mellékhatásának be kell fejeződnie
- egyetlen későbbi végrehajtható tevékenység mellékhatása sem kezdődhet el

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Mellékhatások szinkronizálása

Sorrend-határ pontok:

- függvény aktuális paramétereinek meghatározása után
- && és || bal operandusa után
- ?: első operandusa után
- vessző operátor mindkét operandusa után
- teljes kifejezés kiértékelésének végén:
 - automatikus változó kezdeti értékesítése után
 - utasítás-határon (pontosvessző)
 - do, while, if, switch és for vezérlő kifejezései után
 - for utasítás másik két kifejezése után
 - return utasítás kifejezésének kiértékelése után

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Aritmetikai típusok konverziója

Egyoperandusú konverzió

- értékadáskor
- függvény hívásakor
(a formális paraméterek aktualizálásakor)

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Aritmetikai típusok konverziója

Kétooperandusú konverzió

- műveletvégzéskor

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Típuskonverziók

Alapelv :

érték megőrzése, ha lehet

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Típuskonverziók

Túlcsordulás esetén

a kapott érték elvileg definiálatlan

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Típuskonverziók

Egészről egészbe

NINCS túlcsordulás jelzés

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Típuskonverziók

Ha a hossz és az előjel is változik,
akkor ebben a sorrendben.

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Típuskonverziók

Ha a hossz és az előjel is változik,
akkor ebben a sorrendben.

int -1 : 1 1 1 1 1 1 1 1

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Ha a hossz és az előjel is változik,
akkor ebben a sorrendben.

1. hossz növelése:

int -1 :

long -1 :

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Ha a hossz és az előjel is változik,
akkor ebben a sorrendben.

1. hossz növelése:

int -1 :

long -1 :

2. előjelesség váltása: unsigned long :

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Ha a hossz és az előjel is változik,
akkor ebben a sorrendben.

1. hossz növelése:

int -1 :

long -1 :

2. előjelesség váltása: unsigned long :

Ha fordított lenne a sorrend:

1. előjelesség váltása: unsigned -1 :

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Ha a hossz és az előjel is változik,
akkor ebben a sorrendben.

1. hossz növelése:

int -1 :

long -1 :

2. előjelesség váltása: unsigned long :

Ha fordított lenne a sorrend:

1. előjelesség váltása: unsigned -1 :

2. hossz növelése: unsigned long :

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Miből	Mibe	Eredmény
char, short	int	mindig, minden művelet előtt
rövidebb int pl. short int	hosszabb int ⇒ int ⇒ long	O.K.
rövidebb unsigned pl. unsigned short unsigned	hosszabb unsigned ⇒ unsigned ⇒ unsigned long	O.K.
hosszabb int	rövidebb int	túlcsordulhat (létszámok elvesznek)
hosszabb unsigned	rövidebb unsigned	túlcsordulhat (létszámok elvesznek)
rövidebb + int	unsigned	O.K.
- int	unsigned	modulo 2 ⁿ , azaz ebből = a + 2 ⁿ (> ~MAX !!!)
unsigned	ugyanakkora int	túlcsordulhat
unsigned	hosszabb int	O.K.

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Lebegőpontosból lebegőpontosba

VAN túlcsordulás jelzés

A programozás alapjai 1
6. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Miből	Mibe	Eredmény
rövidebb lebegőpontos pl.: float ⇒ double ⇒	hosszabb lebegőpontos double long double	O.K.
hosszabb	rövidebb	túlcsordulhat, pontosság csökkenhet

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Egészről lebegőpontosba és vissza

VAN túlcsordulás jelzés

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Miből	Mibe	Eredmény
egész	lebegőpontos	ha lehet, a pontos érték megtartásával, ha nem: a legközelebbi két érték egyikére, ha nem lehet: túlcsordul
lebegőpontos	egész	törtész elhagyásával, túlcsordulhat

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Lépések sorrendje kétoperandusú műveletek végrehajtása előtt

0. Ha egyik tömb vagy függvény, akkor mutatóvá konvertálódik
1. Egyoperandusú konverzió a hossz növelésére
2. A két operandus azonos típusú alakítása

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Egyik operandus	Másik operandus	Közös, új típus
long double	bármilyen	long double
double	bármilyen	double
float	bármilyen	float
unsigned long	bármilyen	unsigned long
long	bármilyen (int, unsigned)	long
unsigned	bármilyen (int)	unsigned
int	bármilyen (int)	int

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Típuskonverziók

Példa:

```
int a=3;
double b=2.3;
a=a*b;
```

3
↓
3.0 * 2.3
6.9
↓
6

A programozás alapjai 1
5. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem