

Számtípusok

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2023 szeptember 12

A C++ programokban az adatoknak meg kell adni a típusát.

Számtípusok

Előjeles egész számok (minimum hossz)

- `short` (16 bit)
- `int` (32 bit)
- `long` (32 bit)
- `long long` (64 bit)

Előjeles/előjel nélküli egészek, pl:

- `short`: $[-32768 : 32767]$ vs
`unsigned short`: $[0 : 65535]$

Lebegőpontos számok (floating point)

- `float` (32 bit) - kb. 7 jegy pontos
- `double` (64 bit) - kb. 16 jegy pontos
- `long double` (80bit, a mai operációs rendszerekben 128 biten van tárolva) kb. 19 jegy pontos (ritkán használt)

```
int main() {  
    int a, b = -30;  
    double pi=3.14;  
    unsigned int u=10;  
    return 0;  
}
```

- a definiálás és értékadás összevonható

A legtöbb feladathoz elég, ha `int` és `double` számtípusokat vezetünk be

Műveletek számokkal: összeadás, kivonás

Az eredmény típusa függ az operandusok típusától

- ha az operandusok azonos típusúak, az eredmény is olyan típusú lesz
 - pl.: `int + int = int`
- ha valamelyik pontosabb (több biten ábrázolódik), mint a másik, akkor a fordító előbb a pontosabbra konvertál
 - pl. `int + double = double`
 - pl. `float + double = double`
- vigyázzunk az `unsigned` típusossal!
 - a program futása közben pl egy `unsigned int`-ként definiált változó értéke ne vegyen fel negatív értéket
 - `int + unsigned int = unsigned int`: az előjeles változó előjel nélkülivé konvertálódik a kifejezés kiértékelése során
 - kerüljük az olyan műveleteket, amelyek előjeles és előjel nélküli számtípusokat egyszerre tartalmaznak!

Műveletek egész számokkal: osztás

Az osztás kivezet az egész számok értékkészletéből

- ilyenkor nem történik automatikus típuskonverzió
- helyette: **egészosztás művelet lefelé kerekítéssel**

```
int main() {  
    int a = 12, b = 5;  
    int c=a/b; double d =a/b;  
    cout << "c=" << c << " es d=" << d << endl;  
    return 0;  
}
```

Futtatva:

```
1 $ c=0 es d=0
```

Ha **lebegőpontos osztást** szeretnénk

- legalább az egyik operandus legyen **float** vagy **double**