

Elemi típusok és változók

A számítógép munkájának és így a programozásnak is egyik legalapvetőbb építőkövei az elemi műveletek, melyeket elemi típusú adatokon végzünk.

Típus	Típus neve Java-ban	Értékészlet	Példa
Logikai érték	boolean	true vagy false	
Egész szám	short	-32768 .. 32767	57, 30000, -18
	int	-231 .. 231	1073741824
	long	-263 .. 263	1152921504606846976
Valós szám	float		234.5f
	double		-1.034, 3.78E48
Karakter (egy betű)	char	'\u0000' .. '\uffff'	'A', ';'

A szöveget tartalmazó elemi adatok String típusúak. Ez nem elemi típus, hanem osztály (a nagy kezdőbetű is erre utal), viszont a gyakorlatban szinte úgy használjuk, mint a valódi elemi típusokat.

Az elemi típusokkal sok hasznos elemi művelet végezhető. Ezeket első körben aszerint csoportosítjuk, hogy milyen típusú adatok között értelmezettek, ill. hogy milyen típusú lesz az eredmény.

A matematikai műveletek számok között értelmezettek. Példák:

```
jshell> 1 * 2 + 3 * 4 - 5
$1 ==> 9

jshell> 1 * (2 + 3) * (4 - 5)
$2 ==> -5

jshell> 10 / 3
$3 ==> 3
```

Az utolsó példa eredménye meglepő lehet. Matematikai értelemben nem helyes, viszont itt az a szabály érvényesül, hogy egész számok közötti osztás eredménye is egész marad. Matematikailag helyes eredményt úgy kaphatunk, ha az operandusok eredeti típusát is valósként határozzuk meg. Például:

```
jshell> 10.0 / 3
$4 ==> 3.3333333333333335
```

Számok között gyakran használunk összehasonlító operátorokat is, amelyeknek az eredménye egy logikai érték. Példák:

```
jshell> 89 > 79
$5 ==> true

jshell> 1231321 <= 1231231
$6 ==> false
```

Összehasonlító operátorok:

==	egyenlő
!=	nem egyenlő
>	nagyobb
>=	nagyobb vagy egyenlő
<	kisebb
<=	kisebb vagy egyenlő

Logikai operátorok az és (&&) illetve az vagy (||). Példák:

```
jshell> (7 == 8) || (9 > 10)
$7 ==> false

jshell> ((2 > 3) || (0 == (6 - 2 * 3))) && true
$8 ==> true
```

A + operátor értelmezett a szöveges típusú adatokra is (String), ami értelemszerűen a szövegek összefűzését jelenti. Ez bizonyos mértékig működik szöveg és más típusú adatok között is. Példák:

```
jshell> "Félegyázi" + ' ' + "Márton"
$9 ==> "Félegyázi Márton"

jshell> "ie " + 173
$10 ==> "ie 173"

jshell> "négyzetgyök alatt kettő: " + Math.pow(2, 0.5)
$11 ==> "négyzetgyök alatt kettő: 1.4142135623730951"
```

(az utolsó példánál a Math.pow() függvény a hatványozást végzi, a feledik hatvány meg a gyökvonással egyenértékű; a hatványozó függvény eredménye valós (double) típusú)

Azt a viselkedést, hogy szöveges típusú adathoz bármit hozzáadva szöveges típusú eredményt kapunk, kihasználhatjuk szöveges típusra alakítás céljából is (nem szép, de működik):

```
jshell> 77 - 7 * (5 - 2) * 3.141592645 + ""
$12 ==> "11.02655445500001"
```

Változók

A programozás során is használhatunk az matematikában, fizikában megszokott változókat. Ezek olyan egy vagy több betűvel jelölt értékek, amelyek tényleges értéke lehet ismeretlen vagy akár egy konstans is (például $ax^2 + bx + c$ vagy $E = mc^2$).

A Java programozás során használt változók típussal is rendelkeznek. Ez azért fontos, mert amikor definiáljuk a változót, főleg akkor, amikor még nem is tudjuk megadni a kezdő értékét sem, a futtató környezetnek tudnia kell azt, hogy a leendő érték tárolására mekkora memóriaterületet készítsen elő.

Ez a gyakorlatban két fontos dolgot jelent:

1. Mielőtt használni szeretnénk egy változót, előbb létre kell hoznunk.
2. Amikor létrehozunk egy változót, meg kell adjuk a típusát.

```
jshell> a = 731
Error:
cannot find symbol
symbol:   variable a
a = 731
^
```

Használni próbálok egy létre nem hozott változót
→ Error

```
jshell> int a
a ==> 0
```

Változó létrehozása

```
jshell> a = 731
a ==> 731
```

Változó használata

Változó létrehozása és kezdeti értékkel való ellátása egy lépésben is történhet (de van alapértelmezett kezdeti értéke is).

```
jshell> double x = 2.35
x ==> 2.35
```

A létrehozott változókkal műveleteket is végezhetünk, ill. kezdeti értékadásnál is használhatunk összetett kifejezéseket.

A változók neve állhat több betűből is (meg számból és alulvonásból), de mindenképp kisbetűvel kell kezdődnie.

```
jshell> double befogo1 = 3
befogo1 ==> 3.0

jshell> double befogo2 = 4
befogo2 ==> 4.0
```

Ez is négyzetgyökvonás

```
jshell> double atfogo = Math.sqrt(befogo1 * befogo1 + befogo2 * b
efogo2)
atfogo ==> 5.0

jshell> befogo1 * befogo2 / 2
$19 ==> 6.0
```

Ez itt a „terület”, bár nem mentettük le változóba

A változóknak nem csak a neve „változó”, hanem az értékük is megváltozhat a program futása során (nem konstansok).

```
jshell> int x = 9
x ==> 9

jshell> 7 * x
$21 ==> 63

jshell> x = 13
x ==> 13

jshell> 7 * x
$23 ==> 91
```

Itt x értékét 9-ről 13-ra változtatom

Ha használom, már az új értéke érvényesül

Amikor egy változó új értéket kap, korábbi értékét a változtatás során még felhasználhatjuk. Matematikai szemmel ez elég furán néz ki, viszont ne felejtsük, hogy az = jel nem egy tényállást jelöl vagy vizsgál, hanem egy tényállást hoz létre (értékkadó, értéket beállító operátor).

```
jshell> int r = 91
r ==> 91

jshell> r = r + 7
r ==> 98
```

Itt az r értéke 91-ről 98-ra változik

Feladatok (minden feladatot végezz el változók használatával és anélkül is)

1. Ha az alma egységára 230 Ft, mennyibe kerül 2.5 kg alma.
2. Számold ki egy 10 cm átmérőjű és 25 cm magas fahasáb térfogatát.
 - a henger térfogata = alapterület * magasság
 - az alap kör területe = átmérő * π
 - a π értéke a Java-ban a Math.PI konstansban érhető el
3. Hány kg az előbbi fadarab, ha a fajsúlya 0.47.
4. Mennyi az éves kamata 27000 Ft-nak, ha a kamatláb 7.3%? És mennyi lesz a teljes összeg a végén?
5. Egy 320 km/h sebességgel haladó hurrikán mekkora távolságot tesz meg fél nap alatt? És mennyi idő kell neki 1000 km megtételéhez?
6. Ha a havi nagyjából állandó kiadásaim így alakulnak:
 - víz: 6300 Ft
 - villany: 7200 Ft
 - gáz: 3500 Ft
 - internet: 4050 Ft
 - BKV bérlet: 10500 Ft
 - telefonszámla: 7630 Ft
 - egy tank benzin: 14000 Ft
 - sör: 29000 Ft

Mennyi az éves állandó kiadásaim összege?