



Java



Ismétlés nyelvi eszközök

Kifejezések (operandusok és operátorok)

- szintaktika
- ismert értékekből új értékeket határozzunk meg
- két komponense van típus és érték

Operandusok és operátorok

Operandusok: az értékek

literál

int x=12;

konstans

final double G=8.14;

változó

int x;

metódus hívás

f1(x,y);

Operandusok és operátorok

Aritmetikai operátorok: műveleti jelek

*	<i>int x=a*b;</i>
/	<i>double x=a/b;</i>
+	<i>int x=a+b;</i>
-	<i>int x=a-b;</i>
%	<i>byte x=a%b;</i>

Kerek zárójelek:

a műveletek végrehajtási sorrendjét befolyásolják.

*int x = 3*a+b;*

int x = 3(a+b);*

Implicit konverzió

egy aritmetikai operátor egyik operandusa egész a másik pedig lebegőpontos, akkor az eredmény is lebegőpontos lesz.

Explicit konverzió

a programozó ebben az esetben a kifejezés értékére "ráerőltet" egy típust.

Összehasonlítanak két értéket, és meghatározzák a köztük lévő kapcsolatot.

>	(nagyobb),
>=	(nagyobb vagy egyenlő),
<	(kisebb),
<=	(kisebb vagy egyenlő),
==	(egyenlő-e),
!=	(nem egyenlő)

Logikai operátorok

&& logikai és *ha(a=10 && b=8)...*

|| logikai vagy *ha(a=10 || b=8)...*

! logikai nem *ha(a && !(c||b))...*

Egyéb operátorok

[] tömbképző operátor
int szamok[];

. minősített hivatkozás
System.out.println();

new új objektum létrehozása
java.util.Scanner sc=
new java.util.Scanner(System.in);

Alprogramok

A program célszerű kisebb egységekre bontása

- a program különböző pontjairól aktivizálhatók
- áttekinthetőbb, olvashatóbb a forrásprogram,
- a többször előforduló tevékenységeket elegendő egyszer megírni

Alprogramok

A programot célszerű kisebb egységekre bontása

- Jávában metódus
- meghívásuk a metódus nevére való hivatkozással

Egy metódus lehet *eljárás* vagy *függvény* aszerint, hogy van-e visszatérési értéke.

Eljárás

- Az eljárásnál nincsen visszatérési érték, egyszerűen végrehajtott az eljárás nevére való hivatkozással.
- A végrehajtás után a program a következő utasítással folytatódik
- Az eljárás visszatérési típusa *void* (*semleges*), ami azt jelenti , hogy a visszatérési típus nincs definiálva.

példa eljárásra

```
private static void waiting() {  
    System.out.print("Entert!");  
    System.console().readLine();  
}
```

```
public static void main(String[] args) {  
    ...  
    waiting();  
}
```

Függvény

- A függvénynél van visszatérési érték.
- A függvényt is egyszerűen a nevére való hivatkozással hívunk meg, de visszaad egy értéket, melyet a függvény neve képvisel.

Függvény

```
private static double K(int r){  
    return 2*r.Mtah.PI;  
}
```

```
public static void main(String[]args) {  
    int r=10;  
    double kerulet = K(r); // kerület  
}
```