
Java V.

Osztályszintű tagok

A **final** minősítő

Ficsor Lajos

Miskolci Egyetem

Általános Informatikai Tanszék

Utolsó módosítás: 2005. 10. 05.



Példányváltozó

- Az eddig megismert adattagokból minden objektumnak saját "készlete" van.
- Ezeket ezért szokás **példányváltozó**knak is nevezni.
- Az adattagok deklarációja elé kitehető a **static** minősítő.

Osztályváltozó

- A statikus adattag az osztály valamennyi objektuma számára **egy** példányban létezik, azok osztottan használják.
- Miután az osztály egészére vonatkoznak, szokás **osztályváltozó**knak is hívni. (Globális változó az osztályra nézve.)
- Nem a **this** pszeudó változó segítségével történik az elérése.
- Az objektumok előtt kell létrejönnie.

Osztályváltozó (folyt.)

- Inicializálása az osztály inicializálás során történik.
 - A JVM akkor inicializál egy osztályt, amikor először történik rá aktív hivatkozás (példányosítás, metódus hívás vagy változó hozzáférés).
- Az osztályon belül a nevével hivatkozhatunk.
- Hivatkozhatunk rá bármely objektum példánnyal.
- Az osztály bármely metódusa használhatja.
- Mivel egyetlen példány létrehozása nélkül is létezik, a hivatkozásban osztálynevet is használhatunk.

Osztályváltozó (példa)

```
public class Szamozott {  
    private static int m_iDarab = 0;  
    private int m_iSorszam;  
    public Szamozott() {  
        m_iSorszam = ++m_iDarab;  
    }  
}
```

Minden objektum számontartja, hogy hányadikként jött létre. A számozást a konstruktor végzi.

Osztálymetódus

- Bár az eddigi metódusok csak egy példányban léteznek minden példány számára, de működésükhöz egy aktuális példány szükséges.
- Ezeket hívhattuk volna példánymetódusoknak is.
- Metódus is kaphat **static** minősítést: **osztálymetódus**.
- Az ilyen metódus a példányváltozókhoz és a **this** pszedóváltozóhoz nem férhet hozzá.
 - Következmény: csak osztályváltozókat és más osztálymetódusokat használhat.

Osztálymetódus (folyt.)

- Akkor is végrehajtható, ha az osztálynak nem léteznek példányai.
- Hivatkozás:
 - osztályon belül a nevével
 - minősített hivatkozásnál bármely példány vagy az osztály nevével.
- Egy program indulásakor nincs, ami példányt hozzon létre, ezért a **main** metódusnak **static** minősítésűnek kell lennie.

Osztálymetódus (példa)

Az előző **Szamoszott** osztály egy metódusa lehet:

```
public static int hanyanVagyunk() {  
    return m_iDarab;  
}
```

- Ez a metódus bármely **Szamoszott** típusú példánnyal, vagy a

Szamoszott.hanyanVagyunk()

formában is meghívható.

A **final** minősítő

- A **final** minősítő többféle célra is használható a Java nyelvben.
- Jelentése mindig "valami olyasmi, ami később nem változtatható meg".
- Rokon a C nyelv **const** minősítőjével, de attól sokoldalúbb.

final adattagok

Deklaráció:

final típus azonosító=inicializáló kifejezés

- Az inicializáló kifejezés csak olyan elemeket tartalmazhat, amelyek az addigi deklarációk ismeretében feldolgozhatóak.
- Minden példányosítás során végrehajtódik, tehát a változó értéke objektumonként egyedi lehet.
- Egyetlen metódus sem változtathatja meg az értékét.

static final adattagok

Deklaráció:

static final típus azonosító=inicializáló kif.

- Az inicializáló kifejezés csak konstansokat és olyan **static** adattagokat tartalmazhat, amelyek már deklaráltak.
- Csak egyszer, az osztály inicializálása során hajtódik végre, tehát a változó értéke az osztály minden példánya számára ugyanaz.
- Egyetlen metódus sem változtathatja meg az értékét.

Megjegyzések:

- Az egyszerű típusú **final** adattagok egy konstans értéket képviselnek. (Nagyjából megfelelnek a C **const** változóinak.)
- Az osztálytípusú **final** adattagok egy **konstans referenciát** képviselnek.
 - Nem változtatható meg az értéke, hogy egy másik példányra hivatkozzon.
 - Semmi sem akadályozza azonban azt, hogy a hivatkozott objektumot megváltoztassuk.
 - A Java-ban egy objektum nem deklarálható konstansnak.

Kezdőérték nélküli `final` adattag

- Másik elnevezése **üres konstans**, angol elnevezése: **blank final**.
- A deklarációból az inicializáló kifejezés (az = jellel együtt) elhagyható.
- Az adattag az első használata előtt inicializálandó. (A fordító ezt ellenőrzi!)
- Veszélyes lehet, ezért az ajánlás:
 - ne használjuk,
 - ha használjuk, akkor **minden** konstruktor állítsa be az értékét.