
Java X.

Beágyazott osztályok a Java-ban

Ficsor Lajos

Miskolci Egyetem

Általános Informatikai Tanszék

Utolsó módosítás: 2005. 11. 08.

Osztálydefiníció a programon belül

- Az eddigi példákban egymás mellé rendelt osztály és interface definíciókat láttunk.
- A Java nyelvben ezen kívül ezek a definíciók beágyazhatók
 - osztály hatáskörbe (tagosztályok)
 - utasításblokkba (lokális osztályok)
 - egy utasításba (névtelen osztályok)
- A továbbiakban csak a tagosztályokkal foglalkozunk.

Tagosztály

- Osztálydefiníció egy osztály (befoglaló osztály) hatáskörében.
- A befoglaló és a tagosztály kölcsönösen hozzáférnek egymás tagjaihoz (a **private** tagokhoz is). A nevek minősítése nem szükséges.
- A tagosztályok is öröklődnek.
- A befoglaló osztályon kívüli hozzáférés a tagosztály hozzáférési kategóriájának megfelelően lehetséges. A tagosztály tagjaira való hivatkozáshoz a nevet minősíteni kell.

Statikus tagosztály

- **Static** módosítóval
- Emellett még **public**, **protected**, **private** módosítója is lehet.
- A program csomagon belüli tagolásához: egy típus, amely a befoglaló osztály hatáskörében van.
- Használható a befoglaló osztályon belül típusként.
- Ha nem **private**, a programban ugyanúgy használható, mint bármely más osztály. (Kivéve a bonyolultabb hivatkozást.)

Statikus tagosztály (folyt.)

- Alkalmazási területei:
 - Egy segédosztályt el akarunk rejteni a külvilág elől. (**private** tagosztály).
 - Egy osztály megvalósításakor egy olyan segédosztályra van szükség, amelyeknek hozzá kell férnie az osztály privát tagjaihoz.
 - Ki akarjuk fejezni, hogy egy osztály vagy *interface* egy másiknak logikai alárendeltje.

Nem statikus tagosztály

- A befoglaló és a tagosztály példányainak kapcsolatát implementálhatjuk vele.
 - A tagosztály egy adott példányához a befoglaló osztály **pontosan egy** példánya tartozik.
 - Ellenkező irányban: A befoglaló osztály egy példányához a tagosztály tetszőleges számú (beleértve a nullát is) példánya tartozhat.

Nem statikus tagosztály (folyt.)

- A fenti kapcsolat automatikusan rögzítődik, és fennáll mindaddig, amíg a befoglaló osztály példánya "él".
 - Ha például egy "külső" osztály metódusa példányosít egy beágyazott típusú objektumot, az a befoglaló objektumhoz kapcsolódik, és túléli a metódust (nem úgy, mint egy egyszerű osztály lokális példánya.)
- Alkalmazási példákat majd később látunk.