

2. Java öröklés és kivételek labor, iMSC

Készítette: Goldschmidt Balázs, BME IIT, 2016.

i. Javadoc kézzel

- Bővítsük ki a *Jatekos* osztályt kézzel javadoc kommentekkel.
Az osztály, az attribútumok és a metódusok mind kapjanak kommentet. A metódusok esetén legyen kommentje a paramétereknek is.
- Próbáljuk ki a javadoc dokumentációt: a *Main.main* metódusban hozzunk létre egy új játékost, és nézzük meg, hogy milyen tagjai milyen dokumentációval lesznek felkínálva:

```
Jatekos j = new Robot();  
j.
```

Itt jönnek fel a találatok .

ii. Javadoc segítségével

- A többi osztálynál használjuk az eclipse automatikus komment-generátorát, ami sajnos nem tudja kitalálni, hogy mit akarunk beleírni, de a paraméterekkel és a visszatérési értékkel segít.
 - jobb klikk a mezőn / Source / Generate Element Comment
 - mezőn a kurzor, majd SHIFT-ALT-J

iii. Weak reference használata

- Készítsünk egy *WeakAsztal* leszármazottat az *Asztal*ból!
A *WeakAsztal* tárolja a játékosokat *WeakReference*-ként. A Java template-ek használata jelen esetben nagyon hasonló a C++ változathoz: *Jatekos* helyett *WeakReference<Jatekos>*-t kell használnunk.

Mivel Java-ban nem lehet template-ből tömböt létrehozni, ezért egy kis trükk kell:

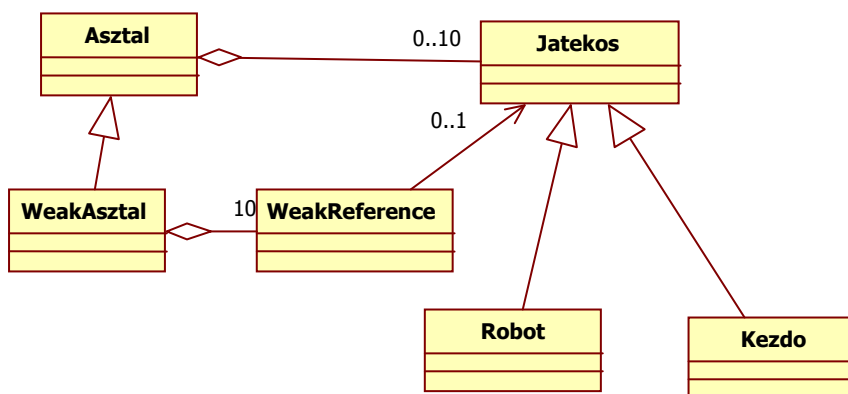
```
WeakReference<Jatekos> jatekosok[]; // típus lehet  
...  
jatekosok = (WeakReference<Jatekos>[])new WeakReference[10];
```

- Módosítsuk úgy az inicializálást, hogy *null* értékeket tartamazó weak referenciákkal legyen a tömb feltöltve
- Módosítsuk úgy az *addJatekos* metódust, hogy a weak referenciát új referenciára cseréli, amiben a paraméterként kapott *Jatekos* található

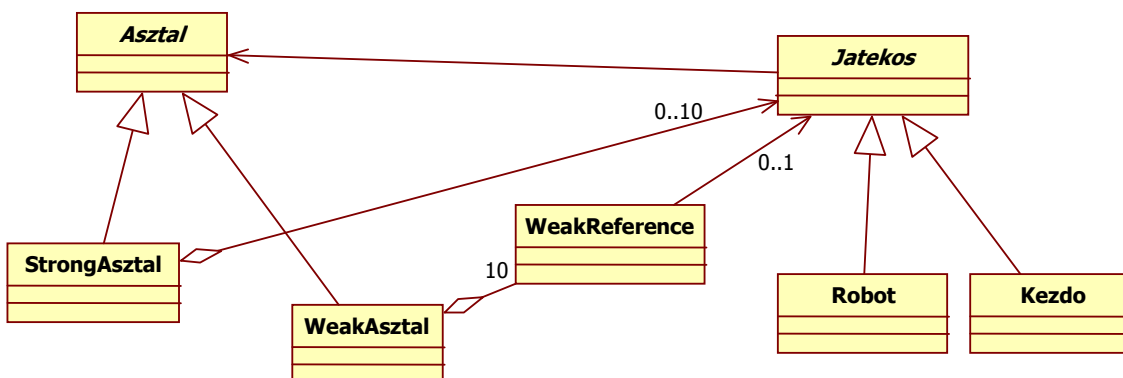
- Hozzunk létre nagyon sok *WeakAsztal*-t úgy, hogy mindegyiknél üljön játékos, de az ezekre a játékosokra mutató referenciákat dobjuk el. Amikor a memória elkezd elfogyni, figyeljük meg, hogy mi történik!
- A fenti pontot hajtsuk végre úgy is, hogy az eredeti Asztal implementációt használjuk! Mi történik ilyenkor?

iv. Abstract Asztal kiemelése

A fenti példa végére az osztályhierarchiánk így néz ki:



Ez a teszteléshez remek volt, de meglehetősen rossz OO tervezési szempontból. A kód módosításával alakítsuk így (ahol az *Asztal* egy abstract osztály, és csak a metódusokat deklarálja, az eredeti *Asztal* implementáció pedig a *StrongAsztalba* kerül):



Próbáljuk ki az eclipse jobb klikkre feljövő menüjében a *Refactor / Extract superclass* opciót!