

2. Java öröklés és kivételek labor

Készítette: Goldschmidt Balázs, BME IIT, 2015.

Az alábbi történet alapján kell Java alkalmazásokat készíteni. A szöveg a 2011. tavaszi Programozás alapjai 2. tárgy nagy ZH-jának záró feladata alapján készült.

Egy számítógéppel felügyelt kaszinó működését szeretnénk modellezni. A kaszinóban a **játékasztalok** (Asztal) nyilvántartják a pillanatnyi tétet, és az asztalnál ülő **játékosokat**, valamint azt, hogy az adott játék hanyadik körében járunk. Egy asztalnál maximum 10 játékos ülhet, de lehet üres asztal is. Bármelyik játékos bármelyik asztalhoz leülhet és ott a leülés sorrendjében játszhat. A játék során az asztalba épített számítógép **körönként**, sorban felszólítja a játékosokat, hogy **lépjenek**. Egy játékos a játék fajtájától függően többféleképpen léphet: passzolhat, tétet emelhet, stb.

Többféle játékos lehet, amelyek eltérő stratégiát alkalmaznak: például kezdő (Kezdo) játékos, aki minden páratlan körben passzol, a többiben eggyel emeli a tétet, vagy egy minden körben passzoló robot (Robot) játékos. Minden játékos nyilvántartja, hogy éppen melyik asztalnál ül. A rendszerben az alábbi műveleteket kell megvalósítani:

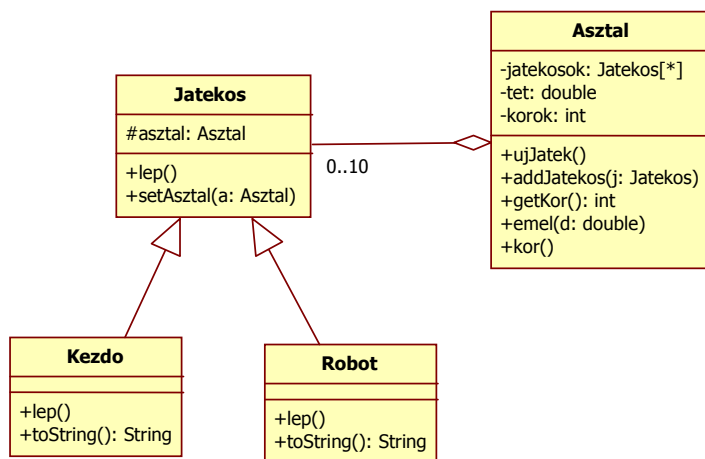
Asztal:

1. Új játékos csatlakozása az asztalhoz (addJatekos). Ilyenkor a játékos is megjegyzi, hogy melyik asztalnál ül.
2. Játék kezdése az asztalnál (ujJatek). Ennek hatására a tét és a körszámláló nullázódik.
3. Kör végrehajtása (kor), melyben minden asztalnál ülő játékos egyet lép. A kör végén az asztal kiírja a tét aktuális értékét.
4. Kör sorszámanak lekérdezése (getKor).
5. Tét emelése (emel)

Játékos:

6. Lép (lep). A játékos ilyenkor megkérdezheti az asztalt, hogy hanyadik körnél tartanak, vagy éppen szólhat az asztalnak, hogy emelje a tétet.

A feladathoz az alábbi osztálydiagram készült (láthatóság: - private, # protected, + public).



1 Aggregáció

Implementálja az **Asztal** és a **Jatekos** osztályt!

Készítsen az osztályok mellé egy **Main** nevű osztályt is, amelyik a **main()** metódust tartalmazza. Ez a metódus hozzon létre egy asztalt, 3 **Jatekost**, és ültesse a játékosokat az asztalhoz.

Játszasson az asztalnál 3 kört!

2 Heterogén kollekció

Implementálja a **Kezdo** és a **Robot** osztályokat, amelyek a **Jatekos** leszármazottai! A lépésekkor mindenki írja ki a típusát, és hogy hányadik körnél jár! A típus kiíratásához definiálja felül és használja a **toString()** metódust is!

Oldja meg, hogy a **Jatekos** osztályból ne lehessen példányt létrehozni!

A **main()** metódusban **Jatekosok** helyett két kezdő és egy robot játékos hozzon létre, őket ültesse az asztalhoz, és így játszasson 3 kört!

3 Statikus tagok

Módosítsa úgy a **Jatekos** osztályt, hogy minden játékosnak saját, egyesével növvő azonosítója legyen!

A játékosok **toString()** metódusa a típusnév mellett ezt az azonosítót is adja vissza (pl. "Kezdo 13")

4 Kivételkezelés

Vegyen fel egy új kivételosztályt **NincsJatekos** néven!

Az asztal a **kor()** metódusban dobjon **NincsJatekos** kivételt, ha nem ül senki nála!

A **main()** metódusban hozzon létre egy új asztalt, és játszasson vele egy kört!

Figyeljen arra, hogy az esetleges kivételek le legyenek kezelve!

5 Finalize

Implementálja a **Jatekos** osztály **finalize()** metódusát: írja ki az objektum azonosítóját valamint a **toString()** metódus visszatérési értékét!

A **main** metódusban nullázza ki az asztal referenciáját, majd indítsa el a garbage collector-t (**System.gc()**)!

Futtassa így az alkalmazást!