

## 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 **játékosok** ülnek **asztalok**nál. Az **asztalok** (Asztal) nyilvántartják a pillanatnyi tétet, és az asztalnál ülő **játékosokat**, valamint azt, hogy az adott játék hányadik 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. Az asztal a játék elején egy véletlen célértéket tárol (**goal**). Ha a tét túllépi ezt az értéket, a játéknak vége. Amelyik játékosnál ez bekövetkezik, nyer, ha a tét kevesebb, mint 10%-kal lépi túl goal-t, egyébként veszít. A többi játékos mind veszít.

**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:

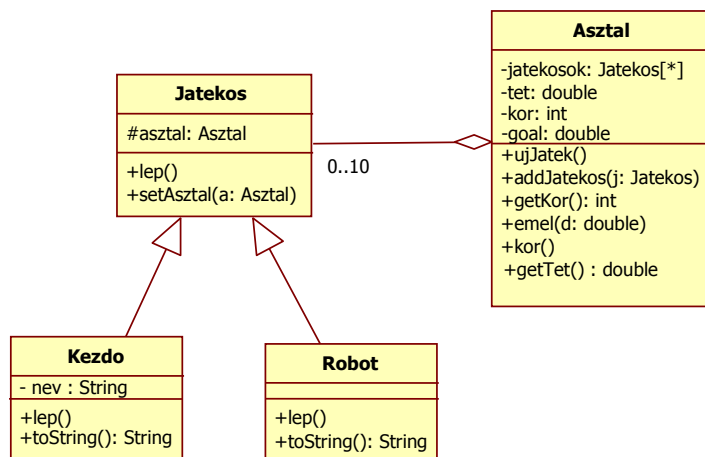
Asztal:

1. Új játékos csatlakozása az asztalhoz (**addJatekos**). Ilyenkor a játékosban beállítja, hogy melyik asztalnál ül. Jelezze, ha az asztal megtelt!
2. Játék kezdése az asztalnál (**ujJatek**). Ennek hatására a tét és a körszámláló nullázódik, a célérték egy véletlen értékre áll (0-100 között).
3. Kör végrehajtása (**kor**), melyben minden asztalnál ülő játékos egyet lép. A kör végén az asztal kiírja a tét aktuális értékét. Amikor vége a játéknak, kiírja, hogy hányas sorszámú játékos nyert. Újabb hívásokra, amíg új játék nem kezdődik, azt írja ki, hogy „Vége a játéknak.”
4. Kör sorszámának lekérdezése (**getKor**).
5. Tét emelése (**emel**) és lekérdezése (**getTet**)

Játékos:

6. Lép (**lep**). A játékos ilyenkor megkérdezheti az asztalt, hogy hányadik körnél tartanak, mennyi az aktuális tét, vagy éppen szólhat az asztalnak, hogy emelje a tétet.

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



## 1 Aggregáció

- Implementáljuk az **Asztal** és a **Jatekos** osztályt!  
Érdemes először két üres osztályt létrehozni a fenti nevekkal, és utána elkezdni felvenni az attribútumokat és a metódusokat.
  - A **Jatekos** **lep** metódusa kiírja, hogy hanyadik körnél járunk, és mekkora a tét.
  - Az **Asztal** a **goal**-t a **Random** osztály segítségével (**nextDouble()** metódus) állítsa be.
- Készítsünk 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.
- Játszassunk az asztalnál 3 kört!

## 2 Heterogén kollekció

- Implementáljuk a **Kezdo** osztályt, amelyik a **Jatekos** leszármazottja! A nevét konstruktorban lehesen állítani, a **toString()**-gel lekérdezni. A lépésekkor írja ki a nevét, hogy hanyadik körnél jár, valamint minden páratlan körben passzoljon, a többiben eggyel emelje a tétet! A név kiírásakor használjuk a felüldefiniált **toString()** metódust is!
- Implementáljuk a **Robot** osztályt, amelyik a **Jatekos** leszármazottja! A lépésekkor írja ki, hogy „Robot”, és hogy hanyadik körnél jár! A „Robot” szót a felüldefiniált **toString()** metódusból kapjuk meg!
- Oldjuk meg, hogy a **Jatekos** osztályból ne lehessen példányt létrehozni!
- A **main()** metódusban **Jatekosok** helyett két kezdő és egy robot játékosat hozunk létre, őket ültessük az asztalhoz, és így játszassunk 3 kört!

## 3 Statikus tagok

- Módosítsuk úgy a **Robot** osztályt, hogy minden robotnak legyen egy saját azonosítója!
- Az azonosítót úgy inicializáljuk a konstruktorban, hogy a robotok azonosítói növekvő számsort alkossanak!
- A robotok **toString()** metódusa a „Robot” szó mögé fűzzék oda az azonosítót is (pl. "Robot13")

## 4 Kivételkezelés

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

- b) Az asztal a **kor()** metódusban dobjon **NincsJatekos** kivételt, ha nem ül senki nála!
- c) A **main()** metódusban hozzon létre egy új asztalt, és játszsasson vele egy kört!  
Figyeljen arra, hogy az esetleges kivételek le legyenek kezelve!

## 5 További játékosok

- a) Implementáljunk egy **Mester** osztályt, amelyik a **Jatekos** leszármazottja, és van mester-fokozata (egy egész érték)! A mester-fokozat konstruktorban állítható. A **toString()**-ben a „Mester” és a mesterfokozat térjen vissza. A lépésekkor írja ki a fokozatát, hogy hányadik körnél jár, és minden körben emelje a tétet annyi százalékkal, amennyi a mester-fokozata!
- b) Implementáljunk egy **Nyuszi** osztályt, amelyik a **Jatekos** leszármazottja! A színét konstruktorban lehesen állítani, a **toString()**-gel lekérdezni.  
A lépésekkor írja ki a színét és hogy hányadik körnél jár! Minden körben emelje a tétet 5-tel addig, amíg a tét 50-nél kisebb. Ha a tét 50 vagy nagyobb, akkor innentől ne emeljen, csak lépésenként írja ki a tét értékét és azt, hogy „Húha!”.
- c) Adjunk az egyik asztalhoz egy mestert és egy nyuszt, játszsassunk velük 10 kört!

## 6 Humán játékos

- a) Implementáljunk egy **Ember** osztályt, amelyik a **Jatekos** leszármazottja! Az Ember a lépések során kiírja az aktuális tétet, és a felhasználótól kérdezi, hogy mennyivel emeljen.
- b) Adjunk az egyik asztalhoz egy Ember-t is, játszsassunk velük 10 kört!

## 7 Finalize

- a) Implementálja a **Jatekos** osztály **finalize()** metódusát: írja ki az objektum azonosítóját valamint a **toString()** metódus visszatérési értékét!
- b) A **main** metódusban nullázza ki az asztal referenciáját, majd indítsa el a garbage collector-t (**System.gc()**)!
- c) Futtassa így az alkalmazást! Meghívódik-e a játékosok finalize metódusa? Ha nem, akkor ennek mi az oka? (Tipp: a szemétygyűjtéskor nem mutathat „élő” referencia az objektumra.)