

## **Dokumentáció**

### **A 3. beadandóhoz**

#### ***Feladat***

Szimuláljuk az alábbi egyszerűsített Monopoly társasjátékot! Adott néhány eltérő stratégiájú játékos és egy körpálya, amelyen különféle mezők sorakoznak egymás után. A pályát körbe-körbe újra és újra bejárják a játékosok úgy, hogy egy kockával dobva mindig annyit lépnek, amennyit a kocka mutat. A mezők három félék lehetnek: ingatlanok, szolgáltatások és szerencse mezők. Az ingatlant meg lehet vásárolni 1000 Petákért, majd újra rálépve ház is lehet rá építeni 4000 Petákért. Ha ezután más játékos erre a mezőre lép, akkor a mező tulajdonosának fizet: ha még nincs rajta ház, akkor 500 Petákot, ha van rajta ház, akkor 2000 Petákot. A szolgáltatás mezőre lépve a banknak kell befizetni a mező paramétereként megadott összeget. A szerencse mezőre lépve a mező paramétereként megadott összegű pénzt kap a játékos. Háromféle stratégiájú játékos vesz részt a játékban. Kezdetben mindenki kap egy induló tőkét (10000 Peták), majd A „mohó” játékos ha egy még gazdátlan ingatlan mezőjére lépett, vagy övé az ingatlan, de még nincs rajta ház, továbbá van elég tőkéje, akkor vásárol. Az „óvatos” játékos egy körben csak a tőkéjének a felét vásárolja el, a „taktikus” játékos minden második vásárlási lehetőséget kihagyja. Ha egy játékosnak fizetnie kell, de nincs elegendő pénze, akkor kiesik a játékból, házai elvesznek, ingatlanjai megvásárolhatókká válnak. A játék paramétereit egy szövegfájlból olvassuk be. Ez megadja a pálya hosszát, majd a pálya egyes mezőit. Minden mezőről megadjuk annak típusát, illetve ha szolgáltatás vagy szerencse mező, akkor annak pénzdíját. Ezt követően a fájl megmutatja a játékosok számát, majd sorban minden játékos nevét és stratégiáját. A tesztelhetőséghez fel kell készíteni a megoldó programot olyan szövegfájl feldolgozására is, amely előre rögzített módon tartalmazza a kockadobások eredményét.

***Írjuk ki, hogy adott számú kör után mely versenyzők birtokolják az egyes ingatlanokat!***

## *Specifikáció*

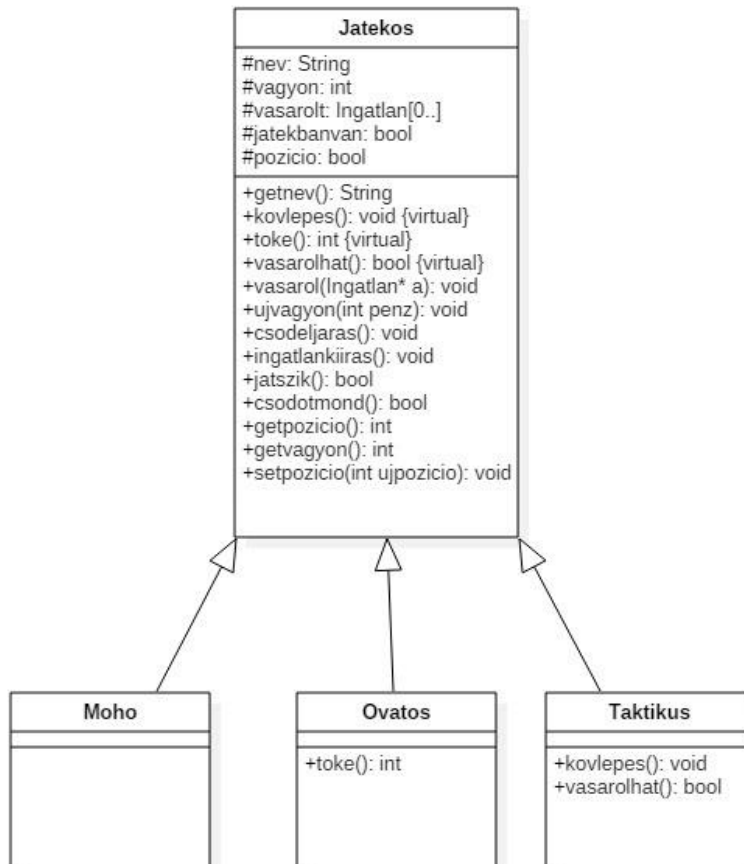
A játékosok leírásához bevezettem négy osztályt: a játékosok általános tulajdonságait leíró őszosztályt (*Jatekos*) és ebből származtattam a konkrét stratégiájú játékosokat, a *Moho*, az *Ovatos* és a *Taktikus* játékos. Attól függetlenül, hogy a játékosok milyen stratégiát alkalmaznak számos közös tulajdonságaik vannak, mint például a nevük, a vagyonuk, vásárolt ingatlanjaik és a pályán lévő pozíciójuk.

A speciális játékos osztályok az őszosztály konstruktorával inicializálják a nevet, a vagyont, a jatekbanvan bool típusú változót igazra és a pozíciót 0-ra, illetve ezen kívül a Taktikus játékos konstruktora meg a vasar változót 0-ra állítja, majd ahol szükséges felüldefiniálják a *toke()*, a *kovlepes()* és a *vasarolhat()* metódusokat

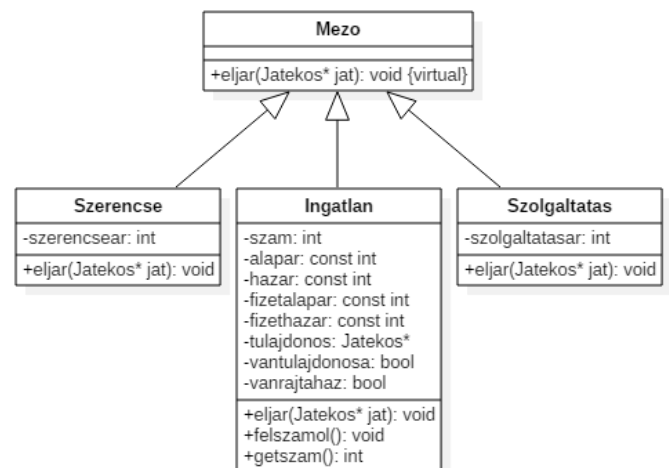
A mezők leírásához szintén négy osztályt vezettem be: a mezők általános tulajdonságait leíró őszosztályt, majd ebből származtattam a konkrét mezőket, a *Ingatlan*, a *Szerencse* és a *Szolgáltatás* mezőt. Ezeknek a legfontosabb eljárásul az *eljar(Jatekos\* jat)*, mely az Ingatlannál eldönti, hogy az adott játékos meg kell (vagy meg tudja) vásárolni az adott mezőt, vagy fizetnie kell az ingatlan tulajdonosának. A szerencse és a szolgáltatás mezők pedig rendre hozzáadnak vagy elvesznek a játékos vagyonából.

## Terv

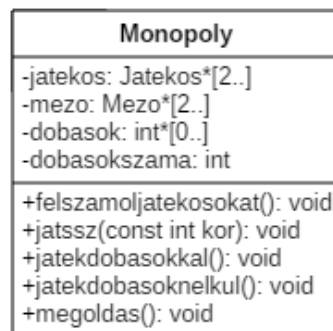
Az alábbi rajz bemutatja a *Jatekos* ösosztályt és a belőle leszármazó 3 osztályt:



Az alábbi rajz, pedig a *Mezo* ösosztályból leszármazó három osztályt mutatja be. Az *eljar(Jatekos\* jat)* metódus egy, a játékosra mutató pointert vár és azzal elvégzi azt, amit kell.



Az utolsó osztály pedig a magáé a játéke, amit Monopolynek nevezünk.



A Monopoly osztály egy *Jatekos*\* és egy *Mezo*\* típusú vektort tartalmaz. Ezekben tároljuk a játékosokat, illetve a pályákat. A *felszamoljatekosokat()* függvény kitörli a vektorból a becsődött játékosokat, a *jatssz(const int kor)* lejátszik a játékból adott mennyiségű kört, amennyiben az előre megadott dobások száma 0. A *megoldas()* függvény kiírja a feladat által kért megoldást.

Állapottér:  $\text{mezo}:\text{Mezo}^n$ ,  $\text{jatekos}:\text{Jatekos}^m$ ,  $\text{dobasok}:\text{int}^k$

Előfeltétel:  $\text{mezo}=\text{mezo}'$ ,  $\text{jatekos}=\text{jatekos}' \wedge \text{dobasokszama}=\text{dobasokszama}'$ , ha  $\text{dobasokszama}>0 \rightarrow \text{dobasok}=\text{dobasok}'$

Utófeltétel:  $\forall i \in [1..kor]$ ,  $\forall j \in [1..m]$  ( $\text{jatekos}[j].\text{pozicio}=\text{jatekos}[j].\text{pozicio}+\text{dobas}$ ) mod  $n$   $\wedge$   $\text{mezo}[\text{jatekos}[j].\text{pozicio}].\text{eljar}(\text{jatekos}[j])$

**jatekdobasokkal()**

**mezokszama:=n**

**cdobas:=0**

**cdobas<dobasokszama**

**i:=1..m**

**cposition:=jatekos[i]->getpozicio()**

**cposition:=(cposition+dobasok[cdobas]) mod mezokszama**

**jatekos[i]->setpozicio(cposition)**

**mezo[cposition]->eljar(jatekos[i])**

**cdobas:=cdobas+1**

**felszamoljatekosokat()**

**jatekdobasoknelkul(int kor)**

**mezokszama:=n**

**i:=1..kor**

**j:=1..m**

**cposition:=jatekos[j]->getpozicio()**

**dobas:=random(1-6)**

**cposition:=(cposition+dobas) mod mezokszama**

**jatekos[j]->setpozicio(cposition)**

**mezo[cposition]->eljar(jatekos[j])**

**felszamoljatekosokat()**

A fenti két algoritmus valósítja meg a játszást előre megadott vagy véletlenszerűen generált dobásokkal.

## ***Tesztelési terv***

1. Létrehozás + Játékosok konstruktorai
  - Ellenőrzöm, hogy mindegyik játékos megkapja a szükséges kezdőparamétereket (név, vagyon, kezdőpozíció és igazra állított jatekbanvan bool változó).
2. Mezők konstruktor
  - Ellenőrzöm, hogy a megfelelő típusú mezőt hozza létre
3. Játékosok viselkedése.
  - Ellenőrzöm a *kovlepes()*, *vasarolhat()* függvények helyes működését.
4. Játékosok kiesése
  - Ellenőrzöm, hogy amennyiben egy játékos pénz nélkülük marad tényleg nem szerepel többet a jatekos tömbben.