

Neptun kód: ASPP08

Feladat

Szimuláljuk az alábbi egyszerűsített Capitaly 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.

Adjuk meg, melyik játékos nyeri meg a játékot és mekkora vagyona (mennyi a tőkéje, milyen ingatlanokat birtokol) van ekkor!

Elemzés¹

Minden játékos jellemzői: az egyedi neve (string), category (String), money (int), fields (ArrayList<Field>), currentPos (int). A játékban részt vevő játékosok fajtái a következők: „mohó”, „óvatos”, „taktikus”.

Vásárlás	Kör	Pénz	Price
mohó	mindegy	> 0	< money

¹ Ez az elemzés rész a hallgatói beadandó dokumentációjából elhagyható, az átalakítási táblázatokat a tervezés részben elég feltüntetni.

óvatos	mindegy	> 0	< money / 2
taktikus	páros	> 0	< money

	Él
mohó	0 < money
óvatos	0 < money
taktikus	0 < money

Minden Field jellemzői: ownerID (int), category (String), price (int), hasHouse (boolean), owned (boolean). A játékban részt vevő játékosok fajtái a következők: „mohó”, „óvatos”, „taktikus”.

Terv¹

Szükségünk lesz egy absztrakt Player (játékos) osztályra a name, category konstruktoron keresztül fogjuk megadni. Ezen felül létre kell hoznunk a megfelelő metódusokat, getName (), getCategory (), getMoney() , divMoney(int money) , addMoney(int money), getCurrentPos(), setCurrentPos(int pos), toString(), buyField(Field field), removeFields(), getFields() , isAlive (él-e?).

A Player osztályt a Greedy, Cautious és Tactician osztályok szülőosztálya lesz

Szükségünk lesz továbbá egy absztrakt Field osztályra a category, price konstruktoron keresztül fogjuk megadni. Ezen felül létre kell hoznunk a megfelelő metódusokat, getPrice(), getCategory (), getHasHouse() , setHasHouse(boolean a) , getOwned (), setOwned(boolean a), getOwnerID (), toString(), setOwnerID(int ID)

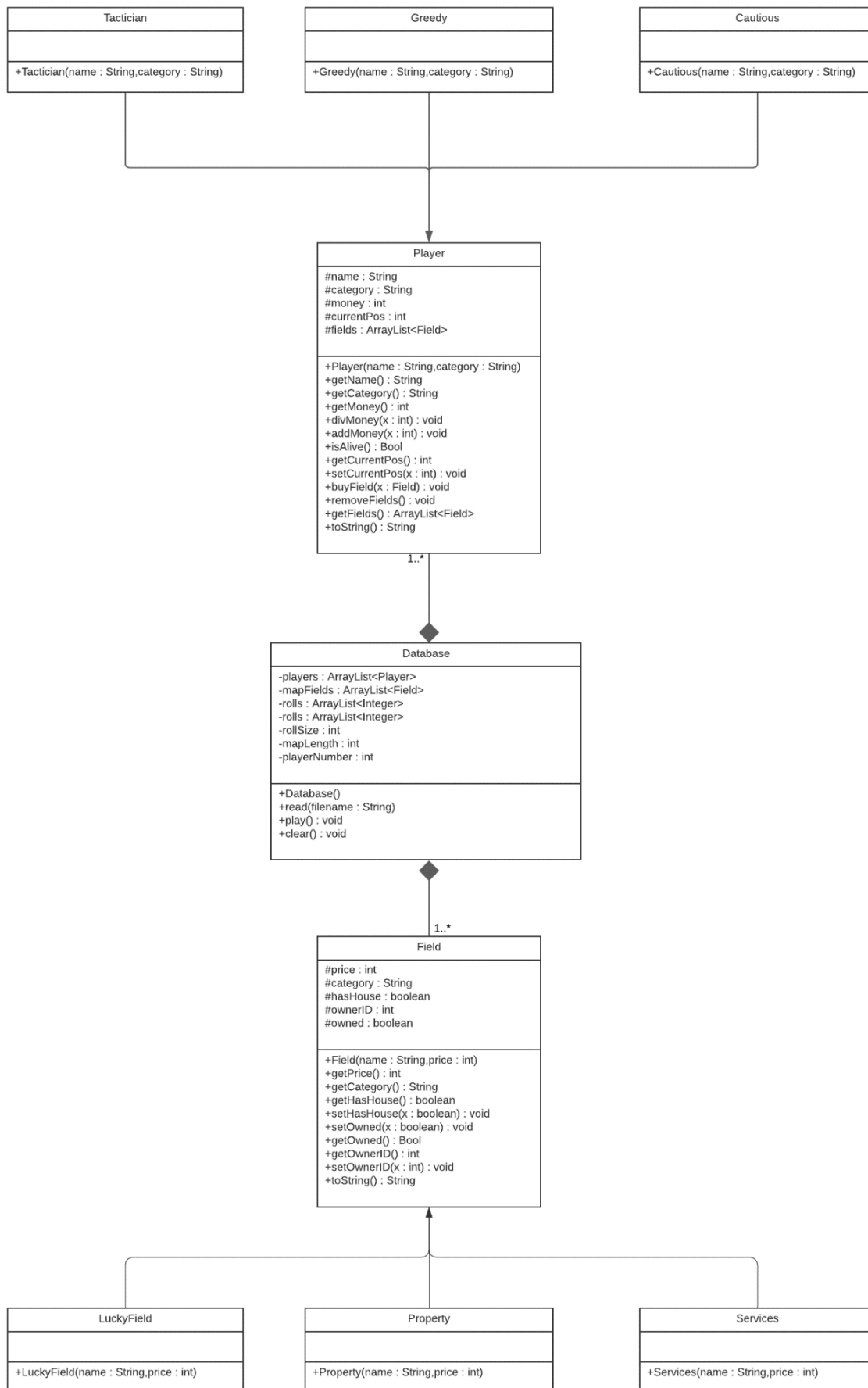
A Field osztályt a Services, LuckyField és Property osztályok szülőosztálya lesz

A Database play() metódusa fogja futtatni a játékot.

play() metódus:

Ciklusban amíg a játékosok száma több mint 1, minden kör elején vagy a beolvasott dobásokból válasszuk ki a következőt (ha elfogyott körbe megy) vagy random számot dobunk neki. A dobás alapján lépünk, majd megvizsgáljuk, hogy milyen mezőre léptünk és milyen típusú a játékosunk, Ez alapján vagy fizet vagy pénzt kap vagy vásárol vagy nem csinál semmit. Megnézzük életben maradt-e a játékos, ha nem kivesszük. Majd kiválasztjuk a következő játékost és ezt így végig.

¹ A szöveges magyarázatra a hallgatói beadandók dokumentumaiban nincs szükség.



Tesztelési terv

Üres fájl	data2.txt	Invalid input!
Hibás fájl	data4.txt	Invalid input!
Nincs fájl	data5.txt	File not found!
Dobások nélkül	data.txt	Más elvárt kimenet mindig
Dobásokkal	data3.txt	<ul style="list-style-type: none"> ○ Player{ name=Player2, category=C, money=9000, fields=[Field{ category=P, price=1000}, Field{ category=P, price=1000}, Field{ category=P, price=1000}, Field{ category=P, price=1000}, Field{ category=P, price=1000}]}
Vásárlás + fizetés	<pre> players.get(0).setCurrentPos(0); players.get(0).buyField(mapFields.get(0)); players.get(1).setCurrentPos(0); players.get(1).divMoney(500); players.get(0).addMoney(500); </pre>	<ul style="list-style-type: none"> ○ Player{ name=Player1, category=G, money=10000, fields=[]} ○ Player{ name=Player2, category=C, money=10000, fields=[]} ○ Player{ name=Player1, category=G, money=10500, fields=[Field{ category=P, price=1000}]} ○ Player{ name=Player2, category=C, money=9500, fields=[]}