PROGRAMOZÁSI TECHNOLÓGIA

1. BEADANDÓ

5. feladat

Capitaly társasjáték szimuláció

Dokumentáció

Készítette:

Horánszki Patrik Donát

CJJ14N

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 házat 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 elégendő 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, melyik játékos esik ki **másodszorra** a játékból!

Megoldási terv

Csomagok, Osztályok szerkezete

- map
  - **Map**
  - fields
    - ***Field***
    - **Luck**
    - **Property**
    - **Service**
  - players
    - ***Player***
    - **Careful**
    - **Greedy**
    - **Tactician**
- exceptions
  - **InvalidDataException**
- ConsoleColors
  - **ConsoleColors**

Game

Main

Megoldási terv

map csomag

map.Map

Reprezentálja a játék térképét, amely mezőkből áll. Feladata a mezők kezelésének biztosítása, beleértve a következő mezőre lépést a dobott kocka alapján.

map.fields.Field *(absztrakt)*

Definiálja a mezők alapvető tulajdonságait. Minden mező, mint például a szerencse, ingatlan és szolgáltatás, ennek az osztálynak az leszármazottja. Tartalmazza a getType és getValue metódusokat, amelyeket a származtatott osztályoknak implementálniuk kell.

map.fields.Luck

Szerencse mezőket reprezentál, amelyek a játék során jutalmakat adnak a játékosoknak. Tartalmaz egy value attribútumot, amely meghatározza a szerencse mező értékét.

map.fields.Property

Ingatlan mezőket reprezentál. Kezeli az ingatlan állapotát (pl. van-e rajta ház, van-e gazdája) és az ingatlan megvásárlásának logikáját. Tartalmazza a buyProperty és buyHouse metódusokat a tranzakciók kezelésére.

map.fields.Service

Szolgáltatás mezőket reprezentálja, amelyek a játékosok által kifizetendő összegeket jelentenek. Tartalmaz egy value attribútumot, amely meghatározza a szolgáltatás költségét.

Megoldási terv

players csomag

players.Player *(absztrakt)*

Játékosokat reprezentál, tartalmazza a játékos nevét, pénzét és a játékban betöltött szerepét. Kezeli a játékos pénzügyeit és interakcióit a mezőkkel. a playField metódust a származtatott osztályoknak implementálniuk kell. A játékosok lépéseit a konzolra tudja írni a writeActionToConsole segítségével.

players.Careful

Óvatos játékos típust reprezentálja, amely a pénzügyi döntéseiben óvatos. Stratégiája figyelembe veszi a játékos jelenlegi pénzügyi helyzetét, és döntéseit ennek megfelelően alakítja: minden második alkalommal él a vásárlás lehetőségével.

players.Greedy

Mohó játékos típust reprezentálja, amely a maximális profit elérésére törekszik. Hajlamos kockázatos döntéseket hozni, hogy gyorsan növelje a vagyonát: Minden vásárlási lehetőséggel él.

players.Tactician

Taktikus játékos típust reprezentálja, amely csak akkor él a vásárlási lehetőséggel, ha a vásárlás esetén aktuális vagyona legalább fele megmarad.

Megoldási terv

exceptions csomag

exceptions.InvalidDataException

Egyedi kivételt definiál, amelyet akkor dobnak, amikor érvénytelen adatokat észlelnek a játékfájl beolvasása során. Használata segít az adatok stabil, pontos beolvasásában, és a hibák kezelésében.

ConsoleColors csomag

ConsoleColors.ConsoleColors

Konzol színek definiálására szolgál ANSI escape kódok segítségével. Lehetővé teszi a színes szöveg kiírását a konzolra, ezzel javítva a felhasználói élményt.

default csomag

Game

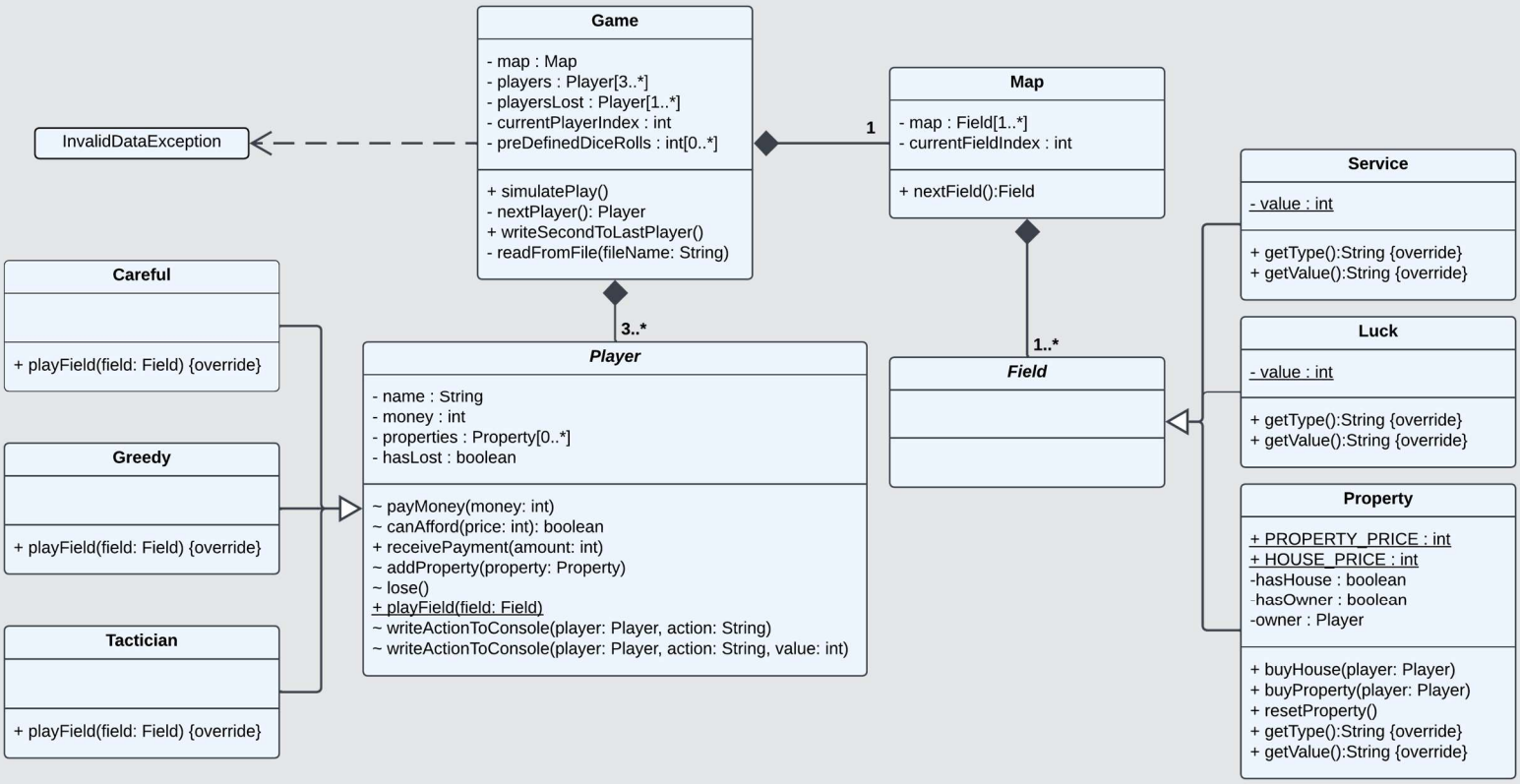
A játék fő logikáját kezeli, beleértve a fájlból olvasást a readFromFile privát metódussal, játékosok körüli interakciókat, lépéseket, és a játék menetét.

Main

A program belépési pontja, ahol a játék inicializálása és indítása történik. Itt definiálják a játék indításához szükséges összes kezdeti beállítást.

Osztálydiagram

UML



Tesztesetek

edge-case

Nem létező fájl *(bármilyen fájl ami nem szerepel a root könyvtárban)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN nem létező fájlt megadunk

THEN FileNotFound kivétellel hibaüzenetet kapunk

```
Game configuration file:
test.txt
File not found: test.txt

Process finished with exit code -1
```

Üres fájl *(testEmpty.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN üres fájlt megadunk

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testEmpty.txt
Invalid data in testEmpty.txt: The file must not be empty!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testWrongNumberOfFields.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, ahol nincs legalább 1 mező

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testWrongNumberOfFields.txt
Invalid data in testWrongNumberOfFields.txt: The number of fields must be at least 1!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testServiceWithoutValue.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, ahol a szolgáltatás mezőnek nincs értéke

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testServiceWithoutValue.txt
Invalid data in testServiceWithoutValue.txt: service must have a value!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testLuckWithoutValue.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, ahol a szerencse mezőnek nincs értéke

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testLuckWithoutValue.txt
Invalid data in testLuckWithoutValue.txt: luck must have a value!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testPropertyWithValue.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, ahol az ingatlan mezőnek van értéke

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testPropertyWithValue.txt
Invalid data in testPropertyWithValue.txt: Property must not have a value!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testWrongFieldName.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, amiben szerepel egy olyan mező, ami nincs a játékban

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testWrongFieldName.txt
Invalid data in testWrongFieldName.txt: money is not a valid field! (property/service/value)

Process finished with exit code -1
```

Rossz bemeneti fájl *(testNotEnoughPlayers.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, amiben nem szerepel legalább 3 játékos

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testNotEnoughPlayers.txt
Invalid data in testNotEnoughPlayers.txt: The number of players must be at least 3!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testPlayerWithoutData.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, amiben a játékosnak nincs neve vagy játékstílusa

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testPlayerWithoutData.txt
Invalid data in testPlayerWithoutData.txt: The player must have a name and a playstyle: Name (greedy/careful/tactician)

Process finished with exit code -1
```

Rossz bemeneti fájl *(testPlayerWithWrongPlaystyle.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, amiben a játékosnak nem létező játékstílusa van

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testPlayerWithWrongPlaystyle.txt
Invalid data in testPlayerWithWrongPlaystyle.txt: happy is not a valid playstyle! (greedy, careful, tactician)

Process finished with exit code -1
```

Rossz bemeneti fájl *(testWrongPredefinedDiceRolls.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, amiben előre definiált dobássorozat esetén nem érvényes kockadobás szerepel a listában

THEN InvalidDataException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testWrongPredefinedDiceRolls.txt
Invalid data in testWrongPredefinedDiceRolls.txt: 7 is not a valid dice roll!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testFirstLineWithoutFieldCount.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, ahol az első sorban nem szerepel a mezők száma

THEN InputMismatchException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testFirstLineWithoutFieldCount.txt
The first line must be the number of fields in the map!

Process finished with exit code -1
```

Rossz bemeneti fájl *(testWrongNumberFormat.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, szám helyett valami más van

THEN NumberFormatException kivétellel hibaüzenetet kapunk

```
Game configuration file:
testWrongNumberFormat.txt
For input string: "3000Forint" is not a valid number!

Process finished with exit code -1
```

Normál futás *(testRandom.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, ahol nincs előre definiált dobássorozat, a véletlenre bízzuk a kimenetelt

THEN megkapjuk a konzolon a játék menetét, illetve azt, aki másodjára kiesett

```
Game configuration file:
testRandom.txt
James received a lucky payment (200)! Balance: 10200
Hannah bought a property! Balance: 9000
Arthur received a lucky payment (500)! Balance: 10500
James bought a property! Balance: 9200
Hannah received a lucky payment (200)! Balance: 9200
Arthur payed for a service (3000)! Balance: 7500
James payed the property tax to the owner! Balance: 8700
Hannah payed the property tax to the owner! Balance: 9200
Arthur payed for a service (2700)! Balance: 4800
James bought a house! Balance: 5200
Hannah bought a property! Balance: 8200
Arthur payed for a service (2700)! Balance: 2100
James payed the property tax to the owner! Balance: 4700
Hannah bought a property! Balance: 7700
Arthur received a lucky payment (200)! Balance: 2300
James payed for a service (2900)! Balance: 1800
Hannah payed for a service (3000)! Balance: 4700
Arthur lost the game!
James payed the property tax to the owner! Balance: 1300
James lost the game!

James finished second to last!

Process finished with exit code 0
```

Normál futás *(testPredefined.txt)*

AS A felhasználó

I WANT TO fájlt beolvasni

GIVEN konzol prompt

WHEN egy olyan fájlt adunk meg, ahol van előre definiált dobássorozat

THEN megkapjuk a konzolon a játék menetét, illetve azt, aki másodjára kiesett

```
Game configuration file:
testPreDefined.txt
James bought a property! Balance: 9000
Hannah payed for a service (2700)! Balance: 7300
Arthur bought a property! Balance: 9000
James payed for a service (3000)! Balance: 6000
Hannah payed for a service (2700)! Balance: 4600
Arthur received a lucky payment (500)! Balance: 9500
James bought a property! Balance: 5000
Hannah payed for a service (2900)! Balance: 1700
Arthur payed the property tax to the owner! Balance: 9000
James payed for a service (2900)! Balance: 2600
Hannah payed the property tax to the owner! Balance: 1200
Arthur received a lucky payment (400)! Balance: 9400
Arthur payed the property tax to the owner! Balance: 8900
James received a lucky payment (400)! Balance: 4000
Hannah payed the property tax to the owner! Balance: 700
Arthur received a lucky payment (200)! Balance: 9100
James received a lucky payment (500)! Balance: 5000
Hannah lost the game!
Arthur payed for a service (2900)! Balance: 6200
James payed the property tax to the owner! Balance: 4500
Arthur payed the property tax to the owner! Balance: 6200
James received a lucky payment (400)! Balance: 5400
Arthur payed the property tax to the owner! Balance: 5700
James payed for a service (2700)! Balance: 3200
Arthur payed the property tax to the owner! Balance: 5200
James received a lucky payment (200)! Balance: 3900
Arthur payed for a service (2900)! Balance: 2300
Arthur received a lucky payment (400)! Balance: 2700
Arthur payed for a service (2700)! Balance: 0
James received a lucky payment (500)! Balance: 4400
Arthur lost the game!

Arthur finished second to last!

Process finished with exit code 0
```