

Blackjack Játék Specifikáció

Feladat rövid szöveges ismertetése

A feladat egy egyszerű Blackjack játék megvalósítása Java nyelven, amely grafikus felhasználói felülettel (GUI) rendelkezik. A játék célja, hogy a játékos kártyáinak összértéke közelebb legyen a 21-hez, mint az osztóé, anélkül, hogy túllépné a 21-et. A játékos különböző műveleteket hajthat végre, mint például kártya húzása (Hit), megállás (Stand), duplázás (Double) és osztás (Split). A játékosnak lehetősége van fogadni, és a játék végén nyerhet vagy veszíthet a fogadása alapján. A játék grafikus felülete lehetővé teszi a játékos számára, hogy könnyen kezelje a játékot és kövesse a játék állapotát.

Use-case-ek

1. Játék indítása

- A felhasználó elindítja az alkalmazást, és megjelenik a főképernyő.
- A felhasználó a tét megadása után a "Deal" gombra kattintva elindítja a játékot.

2. Kártya húzása (Hit)

- A játék során a felhasználó a "Hit" gombra kattintva húz egy új kártyát.
- A játék frissíti a játékos kártyáit és pontszámát.

3. Megállás (Stand)

- A játék során a felhasználó a "Stand" gombra kattintva befejezi a körét.
- Az osztó folytatja a játékot a szabályok szerint, majd a játék véget ér.

4. Duplázás (Double)

- A játék során a felhasználó a "Double" gombra kattintva megduplázza a fogadását és húz egy új kártyát.
- A játék frissíti a játékos kártyáit és pontszámát, majd az osztó folytatja a játékot.

5. Osztás (Split)

- Ha a játékos két azonos értékű kártyát kap, a "Split" gombra kattintva két külön kézre oszthatja a kártyáit.
- A játék frissíti a játékos kártyáit és pontszámát mindkét kézre.

Megoldási ötlet vázlatos ismertetése

A játék megvalósítása Java nyelven történik, a grafikus felhasználói felületet a Swing könyvtár segítségével hozzuk létre. Az alkalmazás MVC (Model-View-Controller) architektúrát követ, ahol a **Dealer** osztály az osztó logikáját, a **GameUI** osztály a felhasználói felületet, és a **CoinPurse** osztály a játékos egyenlegét kezeli.

• Technológiai megoldások:

- Java nyelv

- Swing könyvtár a GUI megvalósításához
- JUnit a teszteléshez

- **Fájlformátumok:**

- A kártyák és zsetonok képei PNG formátumban kerülnek tárolásra a **resources** mappában.
- A játék állapotát és a játékos egyenlegét a memóriában tároljuk, nem használunk külső fájlokat vagy adatbázisokat.

- **Főbb osztályok:**

- **Card**: A kártyák reprezentációja, beleértve a rangot és a színt.
- **CoinPurse**: A játékos egyenlegét és fogadásait kezeli.
- **Dealer**: Az osztó logikáját kezeli, beleértve a kártyák húzását és a játék végének ellenőrzését.
- **Deck**: A kártyapaklit kezeli, beleértve a kártyák keverését és húzását.
- **GameUI**: A grafikus felhasználói felületet kezeli, beleértve a gombok és panelek létrehozását és frissítését.
- **Hand**: A játékos és az osztó kezét kezeli, beleértve a kártyák hozzáadását és az értékek számítását.
- **Main**: Az alkalmazás belépési pontja, amely elindítja a játékot.
- **Rank**: A kártyák rangjait definiálja.
- **Suit**: A kártyák színeit definiálja.

- **Főbb funkciók:**

- Kártyák húzása és megjelenítése
- Fogadások kezelése és frissítése
- Játék állapotának frissítése és megjelenítése
- Játék végének kezelése és üzenetek megjelenítése