Tervezési minta: MVC (Model-View-Controller)

1. Model réteg

Ez a réteg tartalmazza az alkalmazás üzleti logikáját és az adatokat.

Feladatok és osztályok:

- **Board (Tábla osztály):**
  - Tárolja a tábla aktuális állapotát, pl. 2D tömbként (int[][] board).
  - Ellenőrző metódusokat tartalmaz, pl.:
    - boolean checkWin(int row, int col, int player)
    - boolean isColumnFull(int column)
  - A tábla inicializálása és frissítése.
- **Player (Játékos osztály):**
  - Tárolja a játékos adatait (pl. név, azonosító).
  - Egy külön alosztály az emberi és gépi játékosok számára:
    - **HumanPlayer**: Felhasználói input kezelésére szolgáló metódusok.
    - **ComputerPlayer**: Gépi logika (pl. random lépés kiszámítása).
- **GameState (Játékállapot osztály):**
  - Tartalmazza a játék aktuális állapotát (pl. ki következik, vége van-e a játéknak).
  - Mentés és betöltés logikája (XML/JSON formátum).

2. View réteg

A View felelős az adatok megjelenítéséért a felhasználó számára.

Feladatok és osztályok:

- **ConsoleRenderer (Konzolos megjelenítő):**
  - Egyértelmű szöveges tábla kirajzolása (pl. oszlop- és sorszámok megjelenítése).
  - Üzenetek megjelenítése a játékos számára:
    - void displayBoard(int[][] board)
    - void showMessage(String message)
  - Továbbá az aktuális állás és győztes eredmény megjelenítése.

Folytatás a következő oldalon a 3. Réteggel!

3. Controller réteg

A Controller kezeli a felhasználói interakciókat, és koordinálja a Model és a View közötti kommunikációt.

Feladatok és osztályok:

- **GameController (Játékvezérlő):**
  - Felelős a játék folyamatáért:
    - Játékosok váltása, lépések végrehajtása.
    - Input ellenőrzése (pl. oszlopszám érvényessége).
  - Fő metódusok:
    - void playTurn(int player, int column)
    - void startGame()
    - boolean validateMove(int column)
  - Integráció:
    - Meghívja a Model metódusait a lépések feldolgozására.
    - Frissíti a View-t a változások megjelenítésére.