

# **Connect4 Játék Dokumentáció**

**Készítette: Kovács László**

**Neptun: ZP5IXW**

## Tartalom

1. Bevezetés.....	Hiba! A könyvjelző nem létezik.
2. Követelmények.....	Hiba! A könyvjelző nem létezik.
3. Rendszer Áttekintés.....	Hiba! A könyvjelző nem létezik.
Általános Felépítés .....	Hiba! A könyvjelző nem létezik.
4. Osztályok és Metódusok Részletezése .....	Hiba! A könyvjelző nem létezik.
4.1 Main osztály .....	Hiba! A könyvjelző nem létezik.
4.2 Player osztály .....	Hiba! A könyvjelző nem létezik.
4.3 Reader osztály .....	Hiba! A könyvjelző nem létezik.
4.4 Board osztály .....	Hiba! A könyvjelző nem létezik.
4.5 Game osztály .....	Hiba! A könyvjelző nem létezik.
5. Működési Logika.....	Hiba! A könyvjelző nem létezik.
5.1 Játékindítás .....	Hiba! A könyvjelző nem létezik.
5.2 Játékosok Körváltása .....	Hiba! A könyvjelző nem létezik.
5.3 Győzelem Ellenőrzése .....	Hiba! A könyvjelző nem létezik.
5.4 Ranglista Kezelése .....	Hiba! A könyvjelző nem létezik.
6. Használati Példák .....	Hiba! A könyvjelző nem létezik.
6.1 Indítás .....	Hiba! A könyvjelző nem létezik.
6.2 Játékos Lépése.....	Hiba! A könyvjelző nem létezik.
6.3 Gépi Lépés.....	Hiba! A könyvjelző nem létezik.
6.4 Győzelem.....	Hiba! A könyvjelző nem létezik.

## 1. Bevezetés

A **Connect4** egy népszerű stratégiai játék, amely két játékos között zajlik. Az egyik játékos célja, hogy a táblán négy korongját egy sorban, oszlopban vagy átlósan elhelyezze, miközben akadályozza az ellenfelet ugyanezen cél elérésében. Ez a program egy emberi játékos és a számítógép közötti játékot implementál.

---

## 2. Követelmények

- **Java JDK 8** vagy újabb verzió.
  - **Fejlesztői környezet** (pl. IntelliJ IDEA, Eclipse, VS Code).
  - Operációs rendszer: Windows, MacOS, Linux.
- 

## 3. Rendszer Áttekintés

### Általános Felépítés

A játék logikája és működése az alábbi osztályokra épül:

- **Main:** A program belépési pontja.
  - **Game:** A játék fő logikája, amely kezeli a játékosok körét, az eredményeket és a játék állapotát.
  - **Board:** A játéktábla állapotának és megjelenítésének kezelése.
  - **Player:** A játékos adatainak (név, szimbólum) tárolása.
  - **Reader:** A ranglista adatok olvasására és írására szolgáló segédosztály.
- 

## 4. Osztályok és Metódusok Részletezése

### 4.1 Main osztály

*Célja:*

A program belépési pontja, amely létrehozza és elindítja a játékot.

```
public class Main {  
    public static void main(String[] args) {  
        Game game = new Game();  
        game.start();  
    }  
}
```

*Fő Metódusok:*

- **main(String[] args)**
    - Meghívja a Game osztály start() metódusát, amely elindítja a játékot.
-

## 4.2 Player osztály

### *Célja:*

A játékosok adatainak kezelése, beleértve a nevüket és szimbólumukat.

```
public class Player {  
    private final String name;  
    private final char symbol;  
  
    public Player(String name, char symbol) {  
        this.name = name;  
        this.symbol = symbol;  
    }  
  
    public char getSymbol() {  
        return symbol;  
    }  
}
```

### *Attribútumok:*

- **name:** A játékos neve.
- **symbol:** A játékos szimbóluma (pl. 'Y' vagy 'R').

### *Metódusok:*

- **getSymbol()**  
Visszaadja a játékos szimbólumát.
- 

## 4.3 Reader osztály

### *Célja:*

A ranglista adatok (játékosnév és pontszám) kezelésére szolgáló segédosztály.

```
public class Reader {  
  
    public String Name = "0";  
    public int Score = 0;  
}
```

### *Attribútumok:*

- **Name:** A játékos neve.
  - **Score:** A játékos győzelmeinek száma.
- 

## 4.4 Board osztály

### *Célja:*

A játéktábla állapotának kezelése, a lépések rögzítése és a győzelmi feltételek ellenőrzése.

```

public class Board {
    private final char[][] grid;
    private final int rows;
    private final int columns;

    public Board(int rows, int columns) {
        this.rows = rows;
        this.columns = columns;
        this.grid = new char[rows][columns];
        initialize();
    }

    private void initialize() {
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                grid[i][j] = '.';
            }
        }
    }

    public boolean dropDisk(Player player, int column) { /* ... */ }
    public boolean checkWin(Player player) { /* ... */ }
    public void display() { /* ... */ }
}

```

#### *Attribútumok:*

- **grid:** A játéktábla kétdimenziós tömbje.
- **rows:** Sorok száma.
- **columns:** Oszlopok száma.

#### *Fontos Metódusok:*

- **initialize()**  
Inicializálja a táblát üres pontokkal ('.').
- **dropDisk(Player player, int column)**  
Egy játékos korongját az adott oszlopba ejti, ha az lehetséges.
- **checkWin(Player player)**  
Ellenőrzi, hogy a megadott játékos nyert-e.
- **display()**  
Megjeleníti a tábla aktuális állapotát.

## 4.5 Game osztály

### *Célja:*

A játék fő logikájának vezérlése, beleértve a játékosok körét, a gép lépéseit és az eredménykezelést.

```
public class Game {
    private final Board board;
    private Player humanPlayer;
    private final Player computerPlayer;
    private boolean isHumanTurn;

    public Game() {
        this.board = new Board(6, 7);
        this.computerPlayer = new Player("Gép", 'R');
        this.isHumanTurn = true;
    }
}
```

### *Attribútumok:*

- **board:** A játéktábla példánya.
- **humanPlayer:** Az emberi játékos példánya.
- **computerPlayer:** A gép játékos példánya.
- **isHumanTurn:** Logikai változó, amely jelzi, hogy melyik játékos következik.

### *Fontos Metódusok:*

- **start()**  
A játékot elindítja, kezeli a köröket, a bemeneteket, a lépéseket, és a győzelmi feltételek ellenőrzését.

---

## 5. Működési Logika

### 5.1 Játékindítás

A Game osztály start() metódusa elindítja a játékot, bemutatja a ranglistát, és bekéri az emberi játékos nevét.

### 5.2 Játékosok Körváltása

- Az emberi játékos egy oszlopot választ.
- A gép véletlenszerűen választ egy érvényes oszlopot.
- A lépések után ellenőrzi, hogy valamelyik játékos nyert-e.

### 5.3 Győzelem Ellenőrzése

A Board osztály checkWin() metódusa ellenőrzi, hogy van-e négy azonos szimbólum egy sorban, oszlopban vagy átlósan.

## 5.4 Ranglista Kezelése

A győztes neve hozzáadásra kerül a Ranking.txt fájlhoz, amelyet a játék elején rangsorol és megjelenít.

---

## 6. Használati Példák

### 6.1 Indítás

Üdvözöllek a Connect4 játékban!

Add meg a neved:

>> Játékos1

Szeretnéd látni a jelenlegi rangsort? (Y/N)

>> Y

### 6.2 Játékos Lépése

A te lépésed, válassz egy oszlopot (0-6):

>> 3

### 6.3 Gépi Lépés

Gép lépése következik...

### 6.4 Győzelem

Játékos1 nyert! Gratulálunk!

---

## 7. Fejlesztési Lehetőségek

- **Továbbfejlesztett mesterséges intelligencia:** A gép optimalizált stratégiával játsszon.
  - **Grafikus felhasználói felület (GUI):** Játékosabb élmény biztosítása érdekében.
  - **Többjátékos mód:** Hálózaton keresztüli játék támogatása.
  - **Mentési funkció:** Játékmenet állapotának elmentése és visszatöltése.
-