

Tervezési Minták Dokumentáció - Amőba Játék Név: Demeter Attila László
Neptun kód: JY133Z

A beadandó elkészítése során az alábbi tervezési mintákat és elveket alkalmaztam a kód strukturálása és a karbantarthatóság érdekében:

1. Value Object (Érték Objektum) Minta Az alkalmazásban a Position osztály valósítja meg ezt a mintát.

- Leírás: A Position osztály egy koordináta-párt (sor, oszlop) reprezentál. Az osztály állapota a létrehozás után nem változtatható meg (immutable), mivel a mezői final kulcsszóval vannak ellátva, és nincsenek setter metódusai.
- Indoklás: A játék során a pozíciók önmagukban nem változnak, csak újak jönnek létre. Két pozíció akkor egyenlő, ha a koordinátaik megegyeznek, nem pedig akkor, ha ugyanaz a memóriacímük. Ezt biztosítja a felülírt equals() és hashCode() metódus.

2. Data Access Object (DAO) Minta Az alkalmazásban a HighScoreManager osztály valósítja meg ezt a mintát (a persistence csomagban).

- Leírás: Ez az osztály felelős az adatbázissal való közvetlen kommunikációért. Minden SQL specifikus kód (kapcsolódás, INSERT, UPDATE, SELECT) ebbe az osztályba van tokozva.
- Indoklás: A fő programnak (Main) nem kell tudnia arról, hogy az adatok H2 adatbázisban, fájlban vagy felhőben vannak tárolva. A Main csak meghívja a saveWin() metódust, a technikai részleteket a DAO osztály elrejt. Ez növeli a kód átláthatóságát és cserélhetőségét.

3. MVC (Model-View-Controller) Architektúra elv Bár egyszerű konzolos alkalmazásról van szó, a kód szétválasztása követi az MVC elveket:

- Model: A hu.nye.progtech.model csomag (Board, Position) tartalmazza az üzleti logikát és az adatokat. Nem tudnak arról, hogyan vannak megjelenítve.
- View: A konzolos kiírások (System.out.println) felelnek a megjelenítésért. A Board osztály toString() metódusa biztosítja a tábla szöveges reprezentációját a nézet számára.
- Controller: A Main osztály vezérli a játék menetét, fogadja a felhasználói inputot (Scanner), és frissíti a modellt a kapott adatok alapján.