tervezési minták egy oo programozási nyelvben

Mi a tervezési minta?

A tervezési minták (Design Patterns) a szoftverfejlesztés során gyakran előforduló problémákra adott általános, újrahasznosítható megoldások. Nem kész kódblokkok, hanem olyan sablonok és strukturális útmutatók, amelyek segítenek a kód átláthatóságában, a moduláris felépítésben és a hosszú távú karbantarthatóságban.

Alkalmazott tervezési minták és architektúra

1. MVC (Modell-Nézet-Vezérlő) minta

A projekt alapvető struktúráját az MVC minta határozza meg, amely elválasztja az adatokat, a megjelenítést és az irányítást:

Modell (amoba.model): A Board és Player osztályok tárolják a játék állapotát és a logikai szabályokat. Itt történt a legmagasabb szintű egységtesztelés is.

Nézet (View): A Main osztály felel a konzolos megjelenítésért, a tábla kirajzolásáért és a felhasználóval való kommunikációért.

Vezérlő (Controller): A Game osztály koordinálja a játékmenetet, összekötve a felhasználói bemeneteket a modell logikájával.

2. DAO (Data Access Object) minta

Az adatkezelési rétegben a DatabaseManager osztály a DAO minta elveit követi.

Célja: Ez az osztály elszigeteli az SQL műveleteket a szoftver többi részétől. A játék logikájának nem kell ismernie a H2 adatbázis technikai részleteit; csak meghívja a saveWin() metódust, a háttérben zajló adatbázis-műveletek pedig rejtve maradnak.

3. Singleton és Static Utility szemlélet

A projekt a Singleton (Egyke) mintához hasonló megközelítést alkalmaz az adatbázis és a fájlkezelés során.

Megvalósítás: Mivel a DatabaseManager statikus metódusokkal dolgozik, biztosított, hogy az adatbázis-kapcsolat és a ranglista kezelése egyetlen, központi ponton keresztül történjen a program teljes futása alatt, elkerülve a redundáns példányosítást és az erőforrás-ütközéseket.

4. Fájlkezelés (amoba.io csomag)

A GameIO osztály dedikáltan a szöveges fájlokból történő mentésért és betöltésért felel. Ez lehetővé teszi a félbehagyott játékállások elmentését és későbbi folytatását, függetlenül az adatbázisban tárolt ranglistától.

Összegzés

A program felépítése rétegzett és jól elkülönített felelősségi körökre épül (Separation of Concerns). Az alkalmazott tervezési minták lehetővé tették a projekt hatékony tesztelhetőségét (JUnit 5, Mockito), a kódminőség ellenőrzését (Checkstyle) és a moduláris bővíthetőséget.