

## Haladó Szoftverfejlesztés Féléves Feladat

2024/25/1 félév

A tantárgy féléves feladatában minden hallgatónak egy saját adatbázisra épülő rétegzett alkalmazást kell fejlesztenie. A féléves feladat alapvető elvárásai alább kerülnek ismertetésre. A kötelezően elvárt részeknek maradéktalanul meg kell felelni, különben a féléves feladat nem fogadható el.

Az alapvető elvárások a projekttel kapcsolatban

A kész feladatot a moodle rendszerre kell feltölteni. Az előadás kurzus alatt kell keresni a feltöltési lehetőséget. Feltöltési lehetőségek közül mindenkinek a saját laborvezetőjének nevét kell keresni és az **alá feltölteni**.

### **Az alapvető elvárások az üres solution felépítéssel kapcsolatban:**

A féléves feladat kezdetekor el kell indítani a Visual Studio (továbbiakban VS)-t és létre kell hozni egy új Console Appot .NET 8.0-ben! A solution neve legyen: ABC123\_HSZF\_2024251 (ahol ABC123 helyére szintén a neptun kódot kell írni). A project neve legyen: ABC123\_HSZF\_2024251.Console

• **Létre kell hozni a solution-ben az alábbi projekteket (a projekt nevében szerepeljen a solution neve is prefixként, szintén saját neptun kóddal):**

- ABC123\_HSZF\_2024251.Application (Class Library);
- ABC123\_HSZF\_2024251.Model (Class Library);
- ABC123\_HSZF\_2024251.Persistence.MsSql (Class Library);
- ABC123\_HSZF\_2024251.Console (Console App);
- ABC123\_HSZF\_2024251.Test (Class Library).

**Létre kell hozni a solution nevére kattintva (Add) 5 solution folder-t, és ezekbe drag-and-drop módszerrel áthúzni a megfelelő projekteket:**

- Application (Application)
- Domain (Model);
- Infrastructure (MsSql)
- Presentation (Console)
- Tests (Test)

**Legyen beállítva, hogy a Start gomba kattintva a console alkalmazás (a szerver), induljon el. A solution-ön jobb kattintás > Properties > Startup project > Multiple startup projects > Client és Endpoint megjelölése start-ként;**

**Szükséges beállítani a függőségeket! Egy adott projekten jobb kattintás > Add > Project Reference. Az alábbi függőségeket szükséges beállítani. Más függőség nem szerepelhet a rétegek között, ám a Client-hez tesztelési célból felvehető bármelyik, azonban a féléves feladat beadási határidejekor már csak a Models-t ismerheti a Client;**

**Console**-hoz fel kell venni: Application, Infrastructure és Model függőségeket

**Test**-hez fel kell venni: Application, Infrastructure, Modell függőségeket

**Appliciation**-hoz fell kell venni: Model, MsSql függőségeket

**MsSql**-hez fel kell venni: Model függőséget

### **Az alapvető elvárások a szoftverrel kapcsolatban:**

A szoftvernek fordulnia kell a javító tanár számítógépén;

.NET 8.0 verzióban kell írni és adatbázist kell használni Entity Framework Core felett. Más adatbázis és NET verzió nem fogadható el.

A C# nyelvű program osztályait, metódusait, változóit angol nyelven nevezze el. A programban kommentelni lehet magyar nyelven is, de ajánlott angolul;

Model osztályokban legyenek letárolva az idegen kulcsok és használjon Navigation Propertyket LazyLoader-rel ahol lehet! A Linq lekérdezésekben akkor használjunk join-t, ha elkerülhetetlen;

A függőséget csak interfészen át, konstruktor paraméterként kaphatja meg (Dependency Injection)! A Persistence.MsSql a DbContext függőséget csak konstruktor paraméterként kaphatja meg. A függőségek beszúrását a Console projekt végzi, IoC konténer segítségével! Tesztelési célból a konzolos alkalmazásban kézzel példányosíthatóak, de végső megoldásban ezek ne maradjanak a projektben.

A féléves feladatban minimum 10 db Unit tesztet kell létrehozni! Az üzleti logika legyen lefedve unit tesztekkel.

A féléves feladatban minden tesztnek sikeresnek kell lennie! Hogyha 20 tesztet készít, akkor mind a 20-nak sikeresnek kell lennie, nem csak az elvárt 10 db-nak!

Elvárás az alap felhasználói interakció! Ha explicit nem is írják a feladatok, a lekérdezéseket elsősorban a felhasználó indítja el valamilyen egyszerű felhasználói felületről.

A hallgató gondsokodjon arról, hogy megfelelő számú adattal lehessen tesztelni.

Határidő: **12. hét December 1. 23:59:59**

Pótleadás: **13. Hét December 8. 23:59:59**

Feltöltés a moodle-n történik. A javítás első lépése, hogy egy program visszajelzést ad az alapelvárások teljesítéséről. (Fordul a program, a unit tesztek sikeresek, rétegződés teljesül stb). Ennek eredménye teljesült/nem teljesült. Teljesülés esetében jön a második lépése a javításnak, ami a féléves feladat megvédése a laborvezetőnél. Itt elsősorban a kiosztott feladatok szempontjából történik az ellenőrzés. Be kell mutatni a laborvezetőnek, hogy a kiírásban szereplő minden pont teljesült.

## Plusz “pont” lehetőségek

Azon hallgatók akik szeretnének jobb jegyet szerezni a félév során, azok a féléves feladattal ezt megtehetik. Két lehetséges módon.

Fontos, hogy mindkettő csak a szorgalmi időszak alatt szerezhető és csak akkor ha az alábbiak teljesülnek

- A féléves feladat alapkövetelményei maradéktalanul teljesülnek és sikeresen megvédte a féléves feladatot.
- Az előadás zh és labor teljesítmény is legalább elégséges szintű
- A hallgató hiányzásai nem haladják meg a 30%-ot

A jutalmak a féléves teljesítményéhez adódnak hozzá

A jutalmak a következők lehetnek:

- Ha egy hallgató előbb elkészül és a leadási határidő előtt sikeresen megvédi a félévest a laborvezetőjénél, +10%-ot kap
- Ha a témakörhöz köthető 2-3 extra feladatot, easter egget, valami "vicceset" belerak vagy szemmel láthatólag látszik, hogy az elvárások felett teljesített, +5%-ot kaphat a féléves teljesítményéhez.