

Haladó Fejlesztési Technikák

Gyakorlati Zárthelyi Dolgozat – 2023.02.06

Az elkészült teljes anyagot ZIP formátumban töltsse fel az oktató által megadott helyre.

Előkészületek

Az első 5 percben készítsen elő egy CoursesDatabaseApp nevű konzol alkalmazást, és egy CoursesDatabaseApp.Test nevű DLL alkalmazást!

A CoursesDatabaseApp alkalmazáshoz telepítse az alábbi NUGET csomagokat:

- `Microsoft.EntityFrameworkCore`
- `Microsoft.EntityFrameworkCore.InMemory`
- `Microsoft.EntityFrameworkCore.Proxies`

A CoursesDatabaseApp.Test alkalmazáshoz telepítse az alábbi NUGET csomagokat:

- `NUnit`
- `NUnit3TestAdapter`
- `Microsoft.NET.Test.Sdk`

Osztályok elkészítése (4 pont)

Készítse el az alábbi osztályokat a felsorolt adattagokkal:

Instructor:

- *Id*: int (automatikusan növelt)
- *Name*: string
- *Neptun*: string
- *InstructorCourses*: InstructorCourse lista (LazyLoading, virtuális adattag, amely nem szerepel az adatbázisban!)

Course:

- *Id*: int (automatikusan növelt)
- *Title*: string
- *Credits*: int
- *InstructorCourses*: InstructorCourse lista (LazyLoading, virtuális adattag, amely nem szerepel az adatbázisban!)

InstructorCourse: egy kapcsolótábla az alábbi adattagokkal:

- *Id*: int (automatikusan növelt)
- *InstructorId*: int, amely majd idegen kulcs lesz
- *CourseId*: int, amely majd idegen kulcs lesz
- *Instructor*: Instructor (LazyLoading, virtuális adattag, amely nem szerepel az adatbázisban!)
- *Course*: Course (LazyLoading, virtuális adattag, amely nem szerepel az adatbázisban!)

In-Memory adatbázis elkészítése (10 pont)

Készítsen el egy in-memory database-t a data.txt-ben található seed adatok alapján egy `CoursesDbContext` nevű osztály segítségével.

Megj.: Mivel az `InstructorCourse` táblának 2 idegen kulcsa is van, ezért a `modelBuilder.Entity<InstructorCourse>()` esetében az `Instructor` és a `Course` tekintetében is rendelkezni kell egy-a-többhöz kapcsolatról (1 `Instructor`, több `InstructorCourse`; 1 `Course`, több `InstructorCourse`, ahol az `InstructorId` illetve `CourseId` az idegen kulcsok)!

Megj. 2: Továbbá, mivel az összes lekérdezés csak az `InstructorCourse` osztályon keresztül fog történni, ezért elegendő csak ehhez az egy osztályhoz tartozó `DbSet`-et létrehozni.

LINQ lekérdezések (10 pont)

Az `InstructorCourse` `DbSet` alapján a `Main(string[] args)` függvényben írjon LINQ lekérdezéseket az alábbi információk kigyűjtésére:

- Az összes kurzus (course) listája, úgy, hogy ismétlődés ne legyen benne (2 pont)
- Az összes olyan kurzus listája, melyet Jane Doe nevű oktató tanít (2 pont)
- Az összes olyan oktató (instructor) neve, akik legalább 3 kurzust tanítanak (6 pont)

Attribútum és validáció reflexióval (10 pont)

Hozzon létre egy attribútumot, és valósítsa meg az attribútum alapján történő validációt:

- Hozza létre a `NeptunAttribute()` attribútumot – amellyel az `Instructor` osztályban levő `Neptun` adattagot ellátva biztosítható, hogy egy Neptun kód csakis 6 karakterből állhat, melyek mindegyike vagy nagybetű, vagy szám (valid példa: ABN27X, invalid példák: ABn28X vagy ABN27XQ) (2 pont)
- A validáláshoz hozzon létre egy `IValidation` interfészt, melynek `public bool Validate(object instance, PropertyInfo prop)` metódusa nincs kifejtve: ez alkalmas lesz arra, hogy az adott instance-nek lekérdezzük a prop által meghatározott tulajdonságát, és igaz/hamis visszatéréssel megmondjuk, hogy teljesül-e rá az adott feltétel (1 pont)
- Ezt az interfészt implementálja a `NeptunValidation(NeptunAttribute)` osztály, mely konstruktora a kapott `NeptunAttribute` objektumot eltárolja, és megvalósítja a `Validate` metódust. A metódus implementációjában célszerű a `string` objektumok `Length` attribútumát, illetve a `char` típus `IsUpper()` illetve `IsDigit()` metódusait felhasználni (3 pont)
- A `Validator` osztálynak a `public bool Validate(object instance)` metódusa végigmegy az instance összes tulajdonságán és lekéri mindegyik attribútumait b.) végigmegy ezeken az attribútumokon, és ha bármelyik kasztolható `NeptunAttribute` típusra, akkor meghívja annak `Validate` metódusát az adott instance adott property-jére c.) ha ez bármelyik esetben hamis értékkel tér vissza, akkor hamis értéket; ellenkező esetben a ciklusok végén igaz értéket ad vissza. (3 pont)
- Miután az `Instructor` osztály `Neptun` adattagjára applikálta az attribútumot, a `Main()` függvényben hozzon létre 3 `Instructor` típusú objektumot, melyek közül 1 valid és 2 invalid. A `Validator` osztály példányával validálja ezeket, és írja ki az eredményt a konzolra! (1 pont)

Unit tesztek (6 pont)

Készítsen egy Test DLL-t, melyben az előző feladat e.) pontjában készített példákat unit test keretében validálja (3 teszteset).