

Eseményvezérelt és Vizuális Programozás ZH 1 – 2022. ősz – B csoport

A ZH során egy 3 projektből álló solutiont kell létrehozni, ezekbe készítsd el az alábbi feladatok megoldását, majd a laborokhoz hasonlóan github pull request formájában add be a befejezési időpontig. Figyelj rá, hogy a saját, laborokon is használt, github classroomos repositorydban dolgozz!

A solutionbe az alábbi típusú és nevű 3 projektet hozz létre:

- Console Application (.NET Core), C#: "ConsoleApplication" néven
- Class Library (.NET Standard) C#: "Common" néven
- xUnit Test Project (.NET Core) C#: "Tests" néven

Ne feledd, hogy egyik projektből csak akkor tudod elérni a másikban lévő dolgokat, ha a projekt References részében felveszel rá egy hivatkozást a Solution Explorerben.

A stringek összerakásához használj string interpolationt, az IEnumerable értékeken pedig amikor csak lehet, használj a Linq fluent API-t (pl. Min() metódus a kapott számok legkisebbikének meghatározásához). Ezeket a metódusokat ha a fordító nem látja, "ctrl-."-ra feljánylja a System.Linq névtér használatát, ezek ebben találhatók.

1. Commonban interfész és első implementáció létrehozása

4p

Töröld az alapból létrejövő Class.cs-t.

Hozz létre egy Animal osztályt, ami három tagváltozóval rendelkezik: Type, Weight és Speed. Mindhárom legyen publikus property és konstruktorban inicializáld őket. Adj hozzá egy Favourite statikus property-t, mely visszaadja a te kedvenc állatod tulajdonságaival (faj, súly, sebesség) rendelkező állatot.

Hozz létre egy Zoo osztályt, ami egy privát Animal listát tartalmaz és két publikus metódust: az Insert metódus egy elemet ad az Animal lista elejéhez, a Count pedig visszaadja, hogy hány állat van az állatkertben.

2. Testsben triviális unit teszt létrehozása

3p

A Tests projektben töröld az alapból létrejövő UnitTest1.cs fájlt.

Hozz létre egy ZooTests unit teszt osztályt, benne egy TestInsert nevű teszttel, ami példányosítja a Zoo osztályt, beszúrja a kedvenc állatod háromszor (Animal.Favourite), majd ellenőrz, hogy három tagú-e az állatkert. (Utána commit!)

(Ha a Zoo nem publikus, akkor hiába adsz a Common projektre referenciát, nem fogod tudni példányosítani. Ezen kívül minden unit teszt osztálynak is publikusnak kell lennie.

(Ha valamiért nem működik a programod, a unit teszteket is tudod debugolni úgy, hogy a Test Explorer ablakban a teszten jobb klikk - Debug-ot választasz.)

3. Állatok módosítása

4p

Hozz létre egy IAnimalModifier nevű (publikus) interfészt, benne egyetlen metódussal Modify néven. Paramétere egy Animal típus, visszatérési értéke nincs.

Hozz létre egy implementációját `AnimalFeeder` néven, ami egy `amount` privát tagváltozóval rendelkezik amit konstruktorban kap meg. A `Modify` függvény a megkapott `Animal` súlyát növeli *amount*tel.

A `Tests` projektben hozz létre egy (publikus) `ModifierTests` osztályt, mely egy `TestFeedAnimal` nevű tesztben ellenőrzi, hogy ha 13-mal megnövelünk egy állatot, akkor az valóban hozzáadódik a súlyához (Tipp: teszteld az `Animal` példányosítása után a súlyát, majd a növelés után az újat).

4. Osztályok kiegészítése

6p

Az `Animal` osztályba tegyél egy `GetData` függvényt mely visszatér az állat rövid leírásával az alábbi módon:
A zsiráf sebessége 10 km/h, a súlya pedig 1000 kg.

A `Zoo` osztályba tegyél egy `GetAverageSpeed` függvényt ami visszaadja az állatkert állatainak átlagsebességét és egy `GetHeaviestAnimal` függvényt, ami visszaadja a legnehezebb tagját.

5. Extra funkcionalitás unit teszt

3p

A `ZooTests` osztályt egészítsd ki, mely egy `TestHeaviestAnimal` nevű tesztben ellenőrzi, hogy egy pár tagú állatkertben megváltozik a legnehezebb tag, ha növeled a súlyát egy másiknak. A növelés előtt és után is ellenőrizd, hogy ki ez a tag. (Utána commit!)

6. A konzol alkalmazás készítése

4p

A `ConsoleApplication` projektben hozz létre egy `PrintAndModify` függvényt ami vár egy `IAnimalModifiert` és egy állatot. A függvény 20-szor alkalmazza a módosítást rajta. Minden módosítás után írja ki az állat leírását, például:

A zsiráf sebessége 10 km/h, a súlya pedig 1000 kg.

A zsiráf sebessége 10 km/h, a súlya pedig 1050 kg.

A zsiráf sebessége 10 km/h, a súlya pedig 1100 kg.

...

a `Main()` metódust egészítsd ki úgy, hogy a példányosítson egy `AnimalFeeder` ami 50-nel növeli a súlyát. Ezután hívd meg a `PrintAndModify` függvényt ezt és a kedvenc állatod használva paraméterként! (Utána commit!)

7. Megoldások leadása

1p

A megoldást a laborfeladatokhoz hasonlóan, github pull requestként add be és a laborvezetődet rendeld hozzá reviewerként. Figyelj rá, hogy a forráskód esztétikus is legyen (felesleges üres sorok mellőzése, kikommentezett forráskód ne legyen), a commitokban csak verziókövetendő fájlok legyenek (forráskód, projekt és solution file igen, bin és obj könyvtár és .user file nem). A pull request szövegébe pedig rakj bele egy screenshotot, amin látszanak a Test Explorerben a zöld unit tesztek, valamint a konzolos alkalmazás futási eredménye. (A konzol alkalmazás ablaka ha futtatás után egyből bezárul, akkor Ctrl-F5-tel indítsd el, akkor biztosan nyitva marad.)

A ZH befejezési ideje az utolsó commit és pull request időpontjára vonatkozik, a végső verzióknak fordulnia kell. Későbbi leadásra percenként -2 pont jár.

A leadásnál a konzol és test explorer screenshot fontos, hogy szerepeljen. Ha elmarad, az workflow hiba és további pontlevonással jár.