

Eseményvezérelt és Vizuális Programozás ZH 1 – 2022. őszi – A 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 másiban 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 felajánlja a System.Linq névtér használatát, ezek ebben találhatóak.

1. Commonban interfész és első implementáció létrehozása

4p

Töröld az alapból létrejövő Class.cs-t.

Hozz létre egy Person osztályt, ami három tagváltozóval rendelkezik: Name, Age és Height. Mindhárom legyen publikus property és konstruktorban inicializáld őket. Adj hozzá egy GetMe statikus függvényt, mely visszaadja a te tulajdonságaidal (név, kor, magasság) rendelkező embert.

Hozz létre egy Classroom osztályt, ami egy privát Person listát tartalmaz és két publikus metódust: az Add metódus egy elemet ad a Person listához, a Count pedig visszaadja, hogy hány ember van az osztályban.

2. Testsben triviális unit teszt létrehozása

3p

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

Hozz létre egy ClassroomTests unit teszt osztályt, benne egy TestAddition nevű teszttel, ami példányosítja a Classroom osztályt, beszúr téged háromszor (Person.GetMe()), majd ellenőrzi, hogy három tagú-e az osztály. (Utána commit!)

(Ha a Classroom 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 tesztek is tudod debugolni úgy, hogy a Test Explorer ablakban a teszten jobb klikk - Debug-ot választasz.)

3. Emberek módosítása

4p

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

Hozz létre egy implementációját PersonGrower néven, ami egy N property-vel rendelkezik. A Modify függvény a megkapott Person magasságát növelje N-nel.

A Tests projektben hozz létre egy (publikus) ModifierTests osztályt, mely egy TestGrowPerson nevű tesztben ellenőrzi, hogy ha 5 centiméterrel megnövelünk egy embert, akkor az valóban hozzáadódik a méretéhez (Tipp: teszteld a Person példányosítása után a méretet, majd a növelés után az újat).

4. Osztályok kiegészítése

6p

A Person osztályba tegyél egy GetDescription függvényt mely visszatér az ember rövid leírásával az alábbi módon (a dátum formátuma nem lényeges): Aladár (130 cm) születésnapja 2007. 10. 05.

A Classroom osztályba tegyél egy GetAverageAge függvényt ami visszaadja az osztály átlagéletkorát és egy GetTallestPerson függvényt, ami visszaadja a legmagasabb tagját.

5. Extra funkcionalitás unit teszt

3p

A ClassroomTests osztályt egészítsd ki, mely egy TestHeightInClass nevű tesztben ellenőrzi, hogy egy pár tagú osztályban megváltozik a legmagasabb tag ha megnöveled a magasságát. 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 IPersonModifiert. A függvény használja GetMe-ből kapott példányt és 10-szer alkalmazza a módosítást rajta. Minden módosítás után írja ki a Person leírását, például:

Aladár (130 cm) születésnapja 2007. 10. 05.

Aladár (135 cm) születésnapja 2007. 10. 05.

Aladár (140 cm) születésnapja 2007. 10. 05.

...

a Main() metódust egészítsd ki úgy, hogy a példányosítson egy PersonGrowert ami 5-tel növeli a magasságot. Ezután hívd meg a PrintAndModify függvényt ezt 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.