

Eseményvezérelt és Vizuális Programozás ZH 1 — 2023. ősz — 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, github classroomos repositorydban dolgozz!

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

- Console Application (.NET 7.0), C#: "ConsoleApplication" néven
- Class Library (.NET 7.0) C#: "Common" néven
- xUnit Test Project (.NET 7.0) C#: "Tests" néven

A stringek összerakásához használj string interpolationt, az IEnumerable értékek visszaadásához pedig használj yield returnt (ahol van értelme)!

1. Commonban interfész és triviális implementáció létrehozása

Töröld az alpból létrejövő Class.cs-t.

Hozz létre egy Person osztályt, ami három tagváltozóval rendelkezik: Name (string), Age (int) és Height (int). Mindhárom legyen publikus property. Adj hozzá egy GetMe statikus függvényt, mely visszaadja a te tulajdonságaiddal (név, kor, magasság (lehetnek kitalált értékek is)) rendelkező embert.

Hozz létre egy publikus IPersonSource interface-t, ami rendelkezik egy „IEnumerable<Person> GetPersons()” metódussal.

Hozz létre egy TrivialPersonSource osztályt, mely implementálja a fenti interfészt. Az osztály GetPersons() metódusa a Person.GetMe() eredményét adja vissza N-szer, ahol N-t a TrivialPersonSource példányosításakor a konstruktának kell átadni paraméterként és az eltárolja. (Ezután git alatt commitolj egyet.)

2. Testsben unit teszt létrehozása

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

Hozz létre egy TrivialPersonSourceTests unit teszt osztályt, benne egy TestCount nevű teszttel, ami példányosítja a TrivialPersonSource osztályt úgy, hogy az 12 személyt adjon vissza. Ezután a unit teszt ellenőrizze, hogy a visszaadott személyek száma tényleg ennyi-e. (Utána commit!)

(Ha a Common projektben egy osztály nem publikus, akkor hiába adsz a Common projektre referenciát, a unit tesztekéből 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 debuggolni úgy, hogy a Test Explorer ablakban a teszten jobb klikk - Debug-ot választasz.)

3. Commonban feltölthető ember nyilvántartó készítése

Hozz létre egy AdvancedPersonSource nevű osztályt, mely szintén implementálja az IPersonSource interfészt. Belül rendelkezzen egy List<Person> tárolóval. Az AdvancedPersonSource.GetPersons() ennek az elemeit adja vissza. A listára új elemet felvenni egy Add(string name, int age, int height) metódussal lehessen.

4. Person.ToString elkészítése

Az esztétikus megjelenítés érdekében a készíts egy Person.ToString() metódust (minden osztály örökli az object-ből), mely string interpolation (\$) használatával a következő minta szerinti formában írja ki az adatokat: „Minta Mókus, 3 éves, 41 cm magas”. (Utána commit!)

5. Összetettebb unit test

A Tests projektben hozz létre egy újabb unit teszt osztályt AdvancedPersonSourceTests néven. Ebben egy unit teszt azt ellenőrizze, hogy ha 3 embert (tetszőleges adatokkal) felveszel az Add metódussal, akkor utána tényleg 3 db Person példányt ad vissza a GetPersons() metódus. (Utána commit!)

6. A konzol alkalmazás készítése

A ConsoleApplication indulásakor hozz létre egy példányt a TrivialPersonSource és az AdvancedPersonSource osztályokból. A TrivialPersonSource 3-szor adja vissza az adataidat, az AdvancedPersonSource pedig 4 tetszőleges adatokkal feltöltött Person. Írd ki a konzolra mindkét PersonSource visszaadott Personjait úgy, hogy implicit (nem kiírva) használod a Person.ToString()-et. (Utána commit!)

7. Megoldások leadása

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 verziónak 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.