

## Eseményvezérelt és Vizuális Programozás ZH 2021. ősz – MINTA KONZULTÁCIÓRA

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. Minden feladat után commitolj!

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.

### Commonban interfész és egyszerű implementáció létrehozása

3p

Töröld az alpból létrejövő Class.cs-t.

Hozz létre egy IBoolSource nevű (publikus) interfészt, benne egyetlen metódussal GetBools néven. Paramétere int N, visszatérési értéke IEnumerable<bool>.

Hozz létre egy SimpleGenerator nevű (publikus) osztályt, mely implementálja ezt az interfészt, rendelkezik egy bool típusú, ValueToGenerate nevű propertyvel, és a GetBools metódusa N darab ValueToGenerate értékével megegyező bool értéket ad vissza. (Utána commit!)

### Testsben triviális unit teszt létrehozása

3p

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

Hozz létre egy SimpleGenerator unit teszt osztályt, benne egy BasicTest nevű teszttel, ami példányosítja a SimpleGeneratort és ellenőrzi, hogy megfelelően beállítva helyesen ad-e vissza 10 db true értéket. (Utána commit!)

### Konzol alkalmazás elkészítése

3p

A ConsoleApplication projektben hozz létre a Program osztályban egy ShowBools metódust, ami paraméterként kap egy IBoolSource példányt és egy darabszámot. A metódus meghívja a GetBools metódust és kiírja, hogy a kapott bool felsorolásban hány true érték szerepelt. Használd a Linq fluent API-t.

A Main() metódusból hívd meg a ShowBools metódust, hogy a SimpleGenerator egy példánya adjon neki 5 true értéket.

Ellenőrzésként futtasd le a konzol alkalmazást! (Utána commit!)

**[Művelet elvégzése] 6p**

A Common projektben hozz létre egy (publikus) IBoolSource leszármazottat PatternGenerator néven, melynek a GetBools metódusa N darab bool értéket ad vissza, a „true-false-false” sorozatot ismételve. (Utána commit!)

**Unit teszt**

**3p**

A Tests projektben hozz létre egy (publikus) PatternGeneratorTests osztályt, mely egy BasicTest nevű tesztben ellenőrzi, hogy a PatternGenerator 5 érték generálása esetén helyes sorozatot ad-e vissza.

(Ha valamiért nem működik a programod, a unit tesztet is tudod debuggolni úgy, hogy a Test Explorer ablakban a teszten jobb klikk - Debug-ot választasz.) (Utána commit!)

**A konzol alkalmazás kiegészítése**

**6p**

A ConsoleApplication projektben a Main() metódust egészítsd ki úgy, hogy a ShowBools metódust egy PatternGenerator példánnyal is meghívja, mely 15 bool értéket generál neki. Helyes működés esetén ebben 5 db true érték lesz, vagyis 5-öt kell kiírnia. (Utána commit!)

**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. Későbbi leadásra percenként -1 pont jár.

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