

## Eseményvezérelt és Vizuális Programozás ZH 2020. 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.

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 másokban lévő dolgokat, ha a projekt References részében felveszel rá egy hivatkozást a Solution Explorerben.

### Commonban interfész és default implementáció létrehozása

3p

Töröld az alpból létrejövő Class.cs-t.

Hozz létre egy INumberSequenceSource nevű (publikus) interfészt, benne egyetlen metódussal GenerateNumbers néven. Paramétere nincs, visszatérési értéke IEnumerable<int>.

Hozz létre egy ZeroGenerator nevű (publikus) osztályt, mely implementálja ezt az interfészt, rendelkezik egy int típusú, N nevű propertyvel, és a GenerateNumbers metódusa N darab 0-t ad vissza (yield return segítségével). (Utána commit!)

### Testsben triviális unit teszt létrehozása

3p

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

Hozz létre egy ZeroTests unit teszt osztályt, benne egy BasicTest nevű teszttel, ami példányosítja a ZeroGeneratort, N-et beállítja 10-re és ellenőrzi, hogy 10 számot kap-e válaszul. (Utána commit!)

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

### Konzol alkalmazás elkészítése

3p

A ConsoleApplication projektben hozz létre a Program osztályban egy ShowNumbers metódust, ami paraméterül kap egy INumberSequenceSource objektumot. Egy List<int> objektumba kigyűjti az INumberSequenceSource által legenerált számokat, majd kiírja őket egyesével külön sorba, az alábbi formában:

0. elem: 33

1. elem: 44

2. elem: 55

(Természetesen a feladatnak megfelelő számokkal.)

A stringek összerakásához használj string interpolationt, az IEnumerable értékeken pedig amikor csak lehet, használd 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.)

A Main() metódusból hívd meg a ShowNumbers metódust, hogy a ZeroGenerator egy példánya 5 darab nullát generáljon neki.

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

### **Számgenerálás**

**6p**

A Common projektben hozz létre egy (publikus) INumberSequenceSource leszármazottat PrimeGenerator néven, melynek egy konstruktor paramétere int N, a GenerateNumbers metódusa az első ennyi prímszámot fogja visszaadni. (Utána commit!)

(Ehhez el kell menteni a konstruktorban N értékét egy osztály szintű változóba, valamint érdemes készíteni egy "private bool IsPrime(int x)" prímtesztelő metódust.)

### **Primgenerációs unit teszt**

**3p**

A Tests projektben hozz létre egy (publikus) PrimeTests osztályt, mely egy FivePrimeTest nevű tesztben ellenőrzi, hogy ha 5 prímszámot kérünk a PrimeGenerator osztálytól, akkor helyes választ kapunk (2, 3, 5, 7, 11).

(Ha valamiért nem működik a programod, a unit teszteket 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 PrimeGenerator-tól kérje el az első 100 prímszámot és azok közül a 3-ra végződők darabszámát írja ki. A darabszám meghatározásra és a 3-as végződésre szűrésre használj Linq fluent API-t (mint a .Select(), .First() és társaik). A helyes darabszám 26. (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.