

## Alkalmazásfejlesztés II. gyakorlat - Házi feladat 1

### A CSOPORT

Követelmények, tudnivalók:

- A feladat kiírásnak megfelelő megvalósításával 5 pontot lehet szerezni.
- A megoldást tömörítve a Coospace gyakorlati színtérbe kell feltölteni NEPTUNKÓD.zip formátumban a feladatnak létrehozott beadási felületen.
- Az archívum tartalmazza az egész Solution-t (.sln és .csproj fájlokat is), azaz kicsomagolás után betölthető és futtatható legyen Visual Studio-ban.
- Tömörítés előtt futtasd le a dotnet clean parancsot, a bin és obj mappák törlendői és a kiadott CSV-eket se tartalmazza a tömörített állomány.
- A program a .NET 8-as verzióját használja.
- A beadott megoldásnak fordulnia és futnia kell, a projekt mappában a dotnet run <argumentumok> parancs hatására induljon el az alkalmazás, ellenkező esetben a projektmunkára automatikusan 0 pont jár.
- Részpontszám szerezhető, azaz ha nem sikerül valamely feladatot teljes mértékben megoldani, attól nem kell kitörölni (de forduljon és fusson a projekt).
- A kiadott CSV-k tartalmazhatnak hibákat. A projekt munka célja nem a CSV hibáinak kézzel történő javítása, így arra nem jár pont. A kiértékelés a kiadott CSV-vel történik.
- Azokban az esetekben, ahol a feladatkiírás nem ad konkrét iránymutatást vagy részletezést, a megoldás módja a hallgató saját belátására és kreativitására van bízva.

- 
1. Készítsd egy platformfüggetlen parancssoros .NET alkalmazást, amely parancssori argumentumként várja a CSV fájl elérési útvonalát. A program pontosan annyi CSV-t várjon amennyit a feladat említ, minden más esetben jelezze a hibát.

Visual Studio-ban is beállíthatók a parancssori argumentumok a következő beállítások segítségével:

- Solution Explorer-ben jobb kattintás a projekt nevére -> Properties
- Debug fül, majd General
- Open debug launch profiles UI -> Command line arguments

```
static void Main(string[] args) {  
    foreach (var arg in args)  
        Console.WriteLine(arg); // 1 2 3  
}  
// dotnet run 1 2 3
```

A felhasználó a programmal a parancssoron keresztül tudjon kommunikálni. A **stop** parancs hatására termináljon a program. Ezen kívül csak a további feladatokban részletezett utasításokat fogadja el a program, minden mást hagyjon figyelmen kívül.

*0.5 pont*

2. Adott az `athletes.csv` állomány, olvasd be fájlt a CSVHelper NuGet csomag segítségével: <https://www.nuget.org/packages/CsvHelper>

Készítsd el a CSV-ben található adattípusoknak megfelelő osztályt külön fájlban, property-k használatával. Egy tetszőleges kollekcióban tárold el a fájl tartalmát.

A CSVHelper használatához segítség: <https://joshclose.github.io/CsvHelper/getting-started/>

*0.5 pont*

3. Készíts egy metódust, amely egy string paramétert vár és a CSV szerinti Team oszlop alapján összeszámolja hány versenyző indult összesen abban a csapatban. A metódus írja ki a kapott eredményt a standard output-ra. A parancs formátuma, amit az alkalmazás vár (ahol CSAPATNÉV a paraméter):

```
team CSAPATNÉV
```

A feladatot LINQ segítségével valósítsd meg.

*1 pont*

4. Készíts egy metódust, amely paraméterben egy string-et és két egész számot vár. A string a CSV szerinti Age, Height vagy Weight oszlopnevek egyike lehet. A metódus listázza ki, hogy mely versenyzőknek van a második és a harmadik paraméter közötti Age/Height/Weight értéke. A metódus írja ki a kapott eredményt a standard output-ra. A parancs formátuma, amit az alkalmazás vár (ahol TULAJDONSÁG, ÉRTÉK1 és ÉRTÉK2 a három paraméter):

```
count TULAJDONSÁG ÉRTÉK1 ÉRTÉK2
```

A feladatot LINQ segítségével valósítsd meg.

*1 pont*

5. Készíts egy metódust, amely parameterben kettő string-et vár. Az első paraméter a CSV szerinti Team, a második pedig a a CSV szerinti Age, Height vagy Weight oszlopnevek egyike lehet. A metódus számolja ki az adott Team színeiben induló versenyzőkhöz tartozó érték átlagát. A metódus írja ki a kapott eredményt a standard output-ra. A parancs formátuma, amit az alkalmazás vár (ahol CSAPATNÉV és TULAJDONSÁG a három paraméter):

```
average CSAPATNÉV TULAJDONSÁG
```

A feladatot LINQ segítségével valósítsd meg.

*1 pont*

6. Minden feladat megoldását írd ki egy CSV fájlba a CSVHelper segítségével, ahol az útvonalat az utolsó parancssori argumentum határozza meg, és a fájl elnevezése kövesse a kiadott parancsot, például: `count-Age-24-28.csv`

A megoldásodat tesztelheted a következő forgatókönyvvvel:

```
dotnet run <path>  
team Argentina  
count Age 24 28  
count Weight 80 90  
average Egypt Age  
stop
```