**B csoport**

Az elkészült teljes solution-t csomagold **ZIP** formátumban. {Neptun kód}-{csoport} pl: ABC123-B. A megoldásra 90 perc van.   
0. feladat: Elkészítendő osztály a car.txt fileban található.

**JSON Beolvasás**

Készíts a Worker osztálynak egy statikus import metódust, amely List<Car> tér vissza! Ez a statikus metódus olvassa be a mellékelt **car.json** fájlt és adja vissza az autókat listaként!

**Attributum**

Hozzon létre egy saját attributumot DataDescription néven  
 Csak tulajdonságokon lehessen alkalmazni!  
 Legyen egy nyilvános tulajdonsága, amelyben a descriptiont lehet tárolni.  
 Alkalmazd néhány propertyn.

**Helper osztály**

a) Készítsen egy Helper osztályt AttributeHelper néven, mely képes visszaadni egy tulajdonság leírását.  
Segítség: GetPropertyDescription<T>(string propertyName), ahol T az osztály, amely tulajdonságát szeretnénk kifejezni, propertyName a tulajdonság neve, melynek a DataDescription attribútumát szeretnénk megjeleníteni.

**Unit Test**

a) Alakítson ki egy olyan különálló DLL-t, melyben unittesztelést hajthatunk végre a félév során tanultak  
alapján.

b) Készítsen egy osztályt az AttributeHelper osztályhoz, melyben háromféleképpen leteszteli a  
GetPropertyDescription () metódus működését:

a. olyan osztály tulajdonságára, melyen található (saját) DataDescription attribútum  
b. olyan osztály tulajdonságra, melyen nem található DataDescription attribútum   
c. null értékre. Az esetlegesen keletkező kivételt kezelje le olyan módszerrel, mint amit a félév során  
tanultunk

**Linq + Mentések:**

1. Melyik a legolcsóbb használt autó?
2. Kérem a három legidősebb autót. Egy List<OldCar> típús gyűjteményt adjunk vissza. (név, év, típús)
3. Típúsonként mennyi az átlag lóerő szám?
4. Hány darab fekete Jeep van a gyűjteményben?
5. Számoljuk ki az autók új árát a rajtuk felhasználható kedvezménnyel price \* (100- discount \*0,01) és adjuk vissza egy DiscountCar nevű objektumba (Név, diszkontár). Mentsük el egy XML file-ba az így kapott gyűjteményt.
6. 2000 utáni évjáratú és legfeljebb 800-ba kerülő autókat gyűjtsük ki egy CheapNewCar gyűjteménybe (name, price, year). Az eredményhez csatlakozzunk egy InMemoryDB-hez és mentsük el egy új táblába.

<PackageReference Include="Newtonsoft.Json" Version="13.0.3" />  
<PackageReference Include="Microsoft.EntityFrameworkCore" Version="5.0.14" />  
<PackageReference Include="Microsoft.EntityFrameworkCore.Proxies" Version="5.0.14" />  
<PackageReference Include="Microsoft.EntityFrameworkCore.InMemory" Version="5.0.14" />  
<PackageReference Include="Microsoft.NET.Test.Sdk" Version="17.7.2" />  
<PackageReference Include="NUnit" Version="3.13.3" />  
<PackageReference Include="NUnit3TestAdapter" Version="4.5.0" />