

Városok – 24 pont

Városokkal kapcsolatos adatokat szeretnénk tárolni és feldolgozni. A `Cities` projektből kiindulva oldd meg a következő feladatokat! Hibakezelésre csak ott van szükség, ahol a feladat azt kifejezetten kéri.

City osztály

- Hozz létre új osztályt `City` néven, és adj hozzá 4 privát adattagot! **1 pont**

- `name`: a város neve (szöveg)
- `country`: a város országa (szöveg)
- `population`: a város lakossága (egész)
- `gdp`: a város GDP-je USD-ben kifejezve (lebegőpontos)

- Készíts egy 4 paraméteres konstruktort, amely segítségével megadhatóak egy objektum tulajdonságai!

Ha a lakosságnak 1-nél kisebb lenne az értéke, akkor dobj a konstruktorban egy `ArgumentException` kivételt! A hibaüzenet legyen: "Hiba: Nem lehet 1-nél kevesebb lakos!" **2 pont**

- Definiál felül a `ToString()` metódust! Egy objektum szöveges formában az alábbi minta szerint épüljön fel! **1 pont**

`<név> lakossága: <lakosság> fő, GDP-je: <gdp> milliárd USD.`

Pl.: Cegléd lakossága: 36125 fő, GPD-je: 1,5 milliárd USD.

- Adj az osztályhoz `Name` és `Country` néven két publikus property-t, amelyekhez csak getter tartozik!

Valósabb képet mutat egy városról a GDP-nek az egy főre átszámított értéke. Írj egy további property-t `GDPPerCapita` néven, amelynek getterje megadja ennek értékét! **2 pont**

$$\text{GDPPerCapita} = \frac{\text{GDP} \cdot 1\,000\,000\,000}{\text{population}}$$

- Tegyük fel, hogy egy országon belül nincsenek azonos nevű városok. Készíts ennek megfelelően egy egyenlőség vizsgálatára alkalmas operátort, amely két várost akkor tekint egyformának, ha nekik a nevük és az országuk megegyezik! **2 pont**

CityCollection osztály

6. Adj a projekthez új osztályt **CityCollection** néven, amely egyetlen privát adattagként egy városokat tartalmazó listával rendelkezik! **1 pont**

A **cities.txt** városok fiktív adatait tartalmazza. minden sorban 4 adat található, melyek rendre a város neve, országa, lakossága, GDP-je milliárd USD-ben számolva. Az adatokat egymástól pontosvessző választja el.

Minta bemenet:

```
Budapest;Magyarország;1750000;150,7  
Debrecen;Magyarország;200000;22,4  
Szeged;Magyarország;160000;19,1  
...
```

7. A **CityCollection** osztály egyetlen konstruktora paraméterben kapja meg egy fájl nevét, majd olvassa be és tárolja el annak tartalmát! **3 pont**
8. Beolvásás során ügyelj a hibakezelésre! Ha egy sor beolvásása sikertelen, a helyes sorokat akkor is tudja beolvasni a program!
- Konverziós művelet miatt keletkező hiba esetén a konzolon "Hiba: a lakosság egész, a gdp lebegőpontos kell legyen!" szöveg jelenjen meg!
- Másfajta kivétel esetén annak üzenetét írasd ki a konzolra! **4 pont**
9. Készíts **PoorestCity()** néven paraméter nélküli publikus metódust az osztályhoz, amely visszatérési értéke az a város, ahol a legkevesebb az egy főre jutó GPD! (Több megoldás esetén tetszőleges megadható.) **2 pont**
10. Tudunk egy **CityCollection** osztályba tartozó objektumot indexelni városok neveivel! Egy város nevét megadva kapjuk meg a hozzá tartozó országot, ha a város megtalálható a listában. Más esetben dobjon **KeyNotFoundException** kivételt "Nincs adat ilyen városról." hibaüzenettel! **4 pont**

Pl.: `collection["Budapest"] == "Magyarország"`

Program osztály

11. Bővítsd a **Program.cs** forráskódot! A program végén kérd be egy város nevét a felhasználótól, majd add meg a hozzá tartozó ország nevét! A felhasználóval a kommunikáció pontosan az alábbi minta szerint történjen:

11. feladat:

Add meg egy város nevét: Miskolc

Az általad megadott város Magyarország területén van.

Hiba esetén a kapott kivétel üzenete jelenjen meg a konzolon!

2 pont

Minta kimenet:

3. feladat:

Cegléd lakossága: 36125 fő, GPD-je: 1,5 milliárd USD.

4. feladat:

Cegléd 1 főre jutó GPD-je: 41522 USD

9. feladat:

A legkisebb GPT/fő-vel rendelkező város: Berlin lakossága: 3700000 fő, GPD-je: 305,8 milliárd USD.

11. feladat:

Add meg egy város nevét: Cegléd

Nincs adat ilyen városról.