

Asztali alkalmazások fejlesztése

11. évfolyam – javító vizsga

A **Bitzeria** nevű étteremben helyben is vásárolhatunk, de interneten keresztül is rendelhetünk ételt. Az eladásokból különböző adatokat tárolnak, ezekkel kell dolgoznod!



Indulj ki a **Forras-Bitzeria** projektből! A mappa nevében a Forras szót írd át a saját nevedre! Pl.: KovacsJanos-Bitzeria.

A **Program.cs** állományban elkülönül egymástól egy **A** és egy **BC** része a forráskódnak. Ez a két része a feladatsornak egymástól független, bármilyen sorrendben megoldható.

Ügyelj rá a képernyőre kiírást igénylő feladatoknál, hogy az adatok **pontosan a minták szerint** jelenjenek meg! A megoldás során külön függvényt, eljárást csak akkor szükséges készítened, ha azt a feladat szövege kéri.

A. Helyben vásárlás

Az étterem minden nap feljegyzi, hogy a vendégek mennyit költöttek náluk. A vásárlások nevű lista naponként lebontva külön listákban tartalmazza a vevők számláinak értékeit.

```
List<List<int>> vasarlasok = new List<List<int>>
{
    new List<int> { 4500, 5200, 3899, 4700, 5000, 5699 },
    new List<int> { 6200, 5700, 4800, 5100, 5900 },
    new List<int> { 4999, 5100, 6500, 4200, 5300, 6100, 4700 },
    new List<int> { 5000, 5800, 4620, 5400, 6000, 7199 },
    new List<int> { 4830, 5100, 5600, 6300, 5900 },
    new List<int> { 5300, 4900, 5800, 6100, 7000, 5500, 6400 },
    new List<int> { 4700, 5200, 5900, 6500, 5600, 6100 },
    new List<int> { 5400, 5800, 5000, 6200, 6700 },
    new List<int> { 4910, 5399, 6000, 7200, 5600, 6100, 6600 },
    new List<int> { 5200, 4800, 5400, 5900, 6300, 7000 }
};
```

Például az első feljegyzett nap első vendége 4500 Ft-ot költött, a második vendég 5200 Ft értékben vásárolt.

1. Jelenítsd meg az adatokat konzolon a minta szerint! Ne használd ki, hogy 10 nap adatait tárolták, a megoldásnak tetszőlegesen sok nap esetén működnie kell! (Az egyes napokon belül is változó lehet a vendégek száma.) **4 pont**
2. Számold ki, hogy átlagosan mennyit költött egy vásárló! Az eredményt a minta szerint, egészre kerekítve jelenítsd meg! **3 pont**
3. Minden vásárlónak, aki 5000 Ft-nál többet költött az étteremben 12% esélye van nyerni egy kupont, amelyet következő rendelése alkalmával felhasználhat. Szimuláld véletlenszám generálással a kuponok sorsolását, majd jelenítsd meg, hogy hányan kaptak kupont! **4 pont**

B. Rendelés

4. Készíts osztályt **Rendelés** néven, amely négy **privát** adattaggal rendelkezik: **1 pont**

- nev: a megrendelő neve (szöveg)
- kupon: használtak-e kupont a rendeléshez? (logikai)
- költség: a rendelés költsége (egész)
- megjegyzes: a rendeléshez leadott további megjegyzés (szöveg)

5. Az osztályhoz egyetlen, 4 paraméteres konstruktor tartozzon! **1 pont**

6. Készíts Nev, Kupon és Költség néven publikus property-ket, amelyek mindegyike csak **getter**rel rendelkezik! **1 pont**

7. A kuponok egységesen 15%-kal csökkentik a vásárlás költségét, így használatuk esetében a vevőnek kevesebbet kell fizetnie.

Vegyél fel egy további **egész típusú** publikus property-t Fizetendo néven, amelyhez szintén csak **getter** tartozik. Ennek értéke a vevő által ténylegesen fizetendő összeg. (Ez kupon nélküli rendeléseknél megegyezik a rendelés költségével.)

Az étterem nem egész értékű fizetendő esetén csak a lefelé kerekített értéket kéri el a vásárlótól. **3 pont**

8. Definiáld felül az alapértelmezett ToString() metódust! Egy rendelést a következő formában szeretnénk megjeleníteni: először a vásárló neve, majd zárójelben a Fizetendo érték, végül pedig a megjegyzés! Például: **2 pont**

Kovacs Bela (7352 Ft): Csípős nélkül kérem a gyrost!

C. Forgalom

9. Készíts osztályt **Forgalom** néven, amely egyetlen **privát** adattagja egy rendelések nevű, **Rendelés** osztályú objektumokat tartalmazó lista! **1 pont**

10. A rendelésekről tárolt információk az **adatok.txt** szöveges állományban találhatóak.

A fájlban minden rendelés két sorban került rögzítésre. Először a vásárló neve szerepel, ezt követi a kuponhasználat (1 = használt kupon, 0 = nem használt kupon), majd a rendelés költsége. A három adat pontosvesszővel tagolt. A második sorban a megjegyzés szövege van.

Az **adatok.txt** első 6 sora:

```
Kovacs Bela;1;8650
Csípős nélkül kérem a gyrost!
Szabo Anna;0;4720
Extra sajtot kérek a pizzára!
Nagy Peter;1;11300
-
```

Például az 1. sor szerint Kovacs Bela rendelése 8650 Ft értékű, de használt kupont, így fizetnie 7352 Ft-ot kell. Nagy Peter volt az első vevő, aki nem írt megjegyzést.

Feltehető, hogy a név azonosítja a vásárlót, de egy ember többször is rendelhet az étteremből.

A **Forgalom** osztály konstruktora egyetlen szöveges paraméterként egy fájl nevét várja, amelyből beolvassa az adatokat, és eltárolja a rendelések listában! **6 pont**

11. Hozz létre publikus metódust `Legdrágabb()` néven, amely visszatérési értéke az a rendelés objektum, amiért a vásárlónak a legtöbbet kellett fizetnie! **4 pont**

(Vigyázat! Nem feltétlen ennek a költsége a legnagyobb.)

12. Készíts egy `KupontalanokFajlba()` nevű, visszatérési érték nélküli metódust, amely egy `nevek.txt` nevű fájlba kiírja azon vásárlók neveit, akik még sosem használtak kupont a vásárláshoz! A megfelelő vásárlók neveit csak egyszer jelenítsd meg!

A megoldáshoz bármilyen tanult adatszerkezet, függvény felhasználható, szükség esetén segédfüggvény is készíthető! **6 pont**

(Vigyázat! Más jelent logikailag a „sosem használt” és a „volt amikor nem használt”.)

13. A **Program.cs** állományban hívd meg a **Forgalom** osztály metódusait! **1 pont**

Minta kimenet:

```
1. feladat:
1. nap: 4500 5200 3899 4700 5000 5699
2. nap: 6200 5700 4800 5100 5900
3. nap: 4999 5100 6500 4200 5300 6100 4700
4. nap: 5000 5800 4620 5400 6000 7199
5. nap: 4830 5100 5600 6300 5900
6. nap: 5300 4900 5800 6100 7000 5500 6400
7. nap: 4700 5200 5900 6500 5600 6100
8. nap: 5400 5800 5000 6200 6700
9. nap: 4910 5399 6000 7200 5600 6100 6600
10. nap: 5200 4800 5400 5900 6300 7000
```

```
2. feladat:
Átlagos fizetendő: 5598 Ft
```

```
3. feladat:
Kiosztott kuponok száma: 5 db
```

```
11. feladat:
Legdrágább rendelés:
Nagy Peter (13600 Ft): Ne tegyenek uborkát a hamburgerbe!
```