Gyakorlás - C#

1. Feladat - Egyszer volt Budán kutyavásár

Írjon programot, amely a standard inputról állományvégjelig (EOF-ig) a következő formátumú sorokat olvassa be:

`évszám:város[,város]...`

Az `évszám` egy pozitív egész szám, a `városok` sztringek. Az egyes sorok azt írják le, hogy az adott évben mely városokban rendeztek kutyavásárt.

A programja írja a standard kimenetre lexikografikusan növekvő sorrendben a városok neveit, és minden város mögé a példa kimenetben megadott formában, hogy mely években tartottak az adott városban kutyavásárt! Amennyiben egy városban több kutyavásárt is tartottak volna, akkor ezeknek a kutyavásároknak az évszámait növekvő sorrendben tüntesse fel a kimeneten!

Futási példa, ahol bemenet:

```
1472:Buda
1911:Debrecen,Miskolc
1985:Szeged,Debrecen
```

```
$ Main.exe
```

Kimenet:

```
Buda:1472
Debrecen:1911,1985
Miskolc:1911
Szeged:1985
```

2. Feladat - Szálloda

Készítse el a `Hotel` osztályt, amely egy szállodát reprezentál és a szálloda működéséről tudunk információkat lekérdezni.

Az osztálynak a következő tulajdonságai legyenek:

- **`hotelName`** (szöveg): A szálloda neve (pl. "Sunny Hotel").
- **`rooms`** (egész): A szobák száma.
- **`availableRooms`** (egész): Az elérhető szobák száma.
- **`bookedRooms`** (egész): Az eddig lefoglalt szobák száma.
- **`guests`** (nevek listája): A szállodában tartózkodó vendégek neveit tároló lista vendégek neve szerint ábécé sorrendbe.

A `Hotel` osztály támogassa a következő metódusokat:

- Készítse el a `Hotel` osztályban a példányváltozó értékeinek lekérdező metódusait és legyen formázott a kimenet a példány kiíratásakor.
- A `Hotel` osztály példányosításkor a szálloda nevét és szobák számát adja át, amely alapján töltse fel a többi példányváltozó értékét.
- **`bookRoom()`**: Lefoglal egy szobát a megadott vendég számára. A metódus ellenőrzi, hogy van-e elérhető szoba. Ha nincs elérhető szoba, akkor hibaüzenetet ír ki.
- **`checkoutRoom()`**: A megadott vendég kijelentkezik a szállodából. A metódus ellenőrzi, hogy a vendég szerepel-e a listában. Ha a vendég nem található, akkor hibaüzenetet ír ki.
- **`isRoomAvailable()`**: Ellenőrzi, hogy van-e elérhető szoba a szállodában. Visszatér `true` értékkel, ha van elérhető szoba, egyébként `false`.

Osztály kiíratásának formátuma:

```
Szálloda neve: Sunny Hotel
Összes szoba: 5
Elérhető szobák: 2
Foglalva: 3
Vendégek: Kovács Béla, Nagy Anna, Szabó Péter
```

3. Feladat - Könyvtár rendszer

Egy könyvtári rendszer adatait egy lista formájában tárolják. Az adatok három fő entitásból állnak: *Felhasználók*, *Könyvek*, és *Kölcsönzések*. Az adatokat a következő modellek írják le:

```
class Felhasznalo
{
    public int Id { get; set; }
    public string Nev { get; set; }
    public DateTime RegisztracioDatuma { get; set; }
}

class Konyv
{
    public int Id { get; set; }
    public string Cim { get; set; }
    public string Szerzo { get; set; }
    public int KiadasEve { get; set; }
}

class Kolcsonzes
{
    public int FelhasznaloId { get; set; }
    public int KonyvId { get; set; }
    public DateTime KolcsonzesDatuma { get; set; }
    public DateTime? VisszahozatalDatuma { get; set; } // null, ha még nincs visszahozva
}
```

Az alábbi listák tartalmazzák az adatokat:

```
List<Felhasznalo> felhasznalok = new List<Felhasznalo> { ... };
List<Konyv> konyvek = new List<Konyv> { ... };
List<Kolcsonzes> kolcsonzesek = new List<Kolcsonzes> { ... };
```

Feladatok

- **Összes kölcsönzés**: Írja ki az összes felhasználót és az általuk kölcsönzött könyveket (a könyv címét és szerzőjét). Jelezze azt is, ha egy könyvet még nem hoztak vissza!
- **Legtöbb kölcsönzés**: Melyik felhasználónak van a legtöbb kölcsönzése?
- **Régi könyvek**: Listázza ki azokat a könyveket, amelyek 2000 előtti jelentek meg, és még mindig kölcsönözhetőek (nem kölcsönözték ki vagy már visszahozták őket)!
- **Legrégebbi felhasználó aktív kölcsönzései**: Találja meg a legrégebben regisztrált felhasználót, és listázza ki az összes aktív kölcsönzését (még vissza nem hozott könyveit)!
- **Legutóbbi kölcsönzés**: Írja ki a legutóbbi kölcsönzést: melyik könyvet kölcsönözték ki, ki kölcsönözte, és mikor történt?

4. Feladat - Mérések összehasonlítása

Adott két, ugyanannyi sorból álló állomány (`in1.txt` és `in2.txt`), melyek mérések adatait tartalmazzák. Az első fájl (`in1.txt`) az első méréssorozat eredményeit tartalmazza. Ugyanezeket a méréseket azonos sorrendben megismételtük, s ezt a második fájlban (`in2.txt`) rögzítettük. Vagyis a fájlok *i.* sorában ugyanazon kísérletre kapott két eredmény lett rögzítve.

Egy kísérletet akkor tekintünk sikeresnek, ha a két alkalommal kapott értékek összege meghaladja az 1.0 értéket.

Példa:

```
$ cat in1.txt
0.1
0.5
0.9
0.3
0.6
```

```
$ cat in2.txt
0.2
0.6
0.1
0.9
0.2
```

Az első kísérlet sikertelen, mivel $0.1 + 0.2 < 1.0$. A második és a negyedik kísérlet viszont sikeres lett.

Írjunk programot, ami beolvassa a két input fájlt, s egy `out.txt` nevű állományba kiírja a sikeres kísérletek sorszámát. A program adjon egy kis visszajelzést is.

A fenti példa esetén:

```
$ java Main
-> out.txt létrehozva
-> sikeres kísérletek száma: 2
```

```
$ cat out.txt
2
4
```

Ennek a jelentése: a 2. és a 4. kísérlet sikeres volt.