

Vízibicikli-kölcsönző programozásfeladat

Ebben a feladatban egy Balaton-parti vízibicikli-kölcsönző egy napi forgalmi adatai állnak rendelkezésünkre, melyekkel programozási feladatokat kell megoldania. A kölcsönzőben 7 db vízibicikli áll a strandolók rendelkezésére. (A, B, C, D, E, F, G)

A feladat megoldása során vegye figyelembe a következőket:

- A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 5. feladat:)!
- Az egyes feladatokban a kiírásokat a minta szerint készítse el!
- Az ékezetmentes azonosítók és kiírások is elfogadottak.
- Az azonosítókat kis- és nagybetűkkel is kezdheti.
- A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.
- Megoldását készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!

A kolcsonzesek.txt UTF-8 kódolású forrásállomány soraiban egy-egy kölcsönzés adatait tároltuk a következő sorrendben:

- Név, például: Mihály
- Jármű azonosítója (A, B, C, D, E, F, G): például: c
- Elvitel órája: 9
- Elvitel perce: 55
- Visszahozatal órája: 10
- Visszahozatal perce: 56

Az állomány első sora a mezőneveket tartalmazza, az adatokat pontosvesszővel választottuk el:

```
Név;JAzon;EÓra;EPerc;VÓra;Vperc  
Mihály;C;9;55;10;56  
Dávid;A;10;1;10;58  
Ármin;B;10;5;11;4  
Benjámín;D;10;12;10;30  
Bence;F;10;15;10;55  
Kata;D;10;40;11;45
```

1. Készítsen grafikus vagy konzolalkalmazást (projektet) a következő feladatok megoldásához, amelynek projektjét vizibiciklikolcsonzo néven mentse el!
2. Hozzon létre saját osztályt Kolcsonzes azonosítóval, melynek adattagjait felhasználva egy-egy kölcsönzés adatait tudja majd tárolni! A feladatot megoldhatja saját osztály definiálása nélkül is, de megoldása ebben az esetben nem teljes értékű.
3. Készítse el a Kolcsonzes osztály konstruktorát, mely hívásával az osztály adattagjait tudja inicializálni!
4. Olvassa be a kolcsonzesek.txt állomány sorait és tárolja az adatokat egy Kolcsonzes osztályon alapuló összetett adatszerkezetben! Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!
5. Határozza meg és írja ki, hogy hány kölcsönzés adatai található a forrásállományban!
6. Kérjen be egy nevet és írja ki, hogy az illető aznap mettől meddig bérelt vízibiciklit! Elképzelhető, hogy az illető többször is kölcsönzött aznap, ebben az esetben minden kölcsönzés adatát jelenítse meg! Ha aznap egyszer sem kölcsönzött, akkor a „Nem volt ilyen nevű kölcsönző!” szöveg jelenjen meg!

7. Kérjen be egy időpontot óra:perc alakban, majd írja ki, hogy ekkor mely járművek voltak vízén, és azokat meddig és kik kölcsönözték ki! Feltételezheti, hogy a megadott időpontban legalább egy jármű a vízén volt. A kiírásnál használjon vezető nullákat az egyjegyű óra és perc értékek előtt!
8. A járművek kölcsönzési díja egységesen 2400 Ft minden megkezdett félóránként. Határozza meg a napi bevétel összegét és írja a képernyőre!
9. Sajnos az F jelű járművet valaki megrongálta. Hozzon létre F. txt néven UTF-8 kódolású szöveges állományt, amely tartalmazza a lehetséges elkövetőket és azt, hogy mettől meddig volt náluk a jármű! Az állomány szerkezete a kiadott minta szerinti legyen!
10. Készítsen statisztikát, melyben megjeleníti, hogy az egyes járműveket hányszor kölcsönözték a nap során! Oldja meg, hogy a járművek azonosítói ábécérendben kövessék egymást a statisztika kiírásakor!

Képernyőminta találat esetén a 6. feladatban:

```

5. feladat: Napi kölcsönzések száma: 33
6. feladat: Kérek egy nevet: Kata
    Kata kölcsönzései:
    10:40-11:45
    12:48-13:47
    16:27-17:08
7. feladat: Kérek egy időpontot óra:perc alakban: 10:9
    A vízén lévő járművek:
    09:55-10:56 : Mihály
    10:01-10:58 : Dávid
    10:05-11:04 : Ármin
8. feladat: Napi bevétel: 177600 Ft
10. feladat: Statisztika
    A : 6
    B : 4
    C : 4
    D : 6
    E : 3
    F : 6
    G : 4

```

Képernyőminta, ha nincs találat a 6. feladatban:

```

5. feladat: Napi kölcsönzések száma: 33
6. feladat: Kérek egy nevet: Dénes
    Dénes kölcsönzései:
    Nem volt ilyen nevű kölcsönző!
7. feladat: Kérek egy időpontot óra:perc alakban: 12:50
    A vízén lévő járművek:
    12:01-15:05 : Zsombor
    12:05-14:15 : Dániel
    12:30-13:50 : Ádám
    12:40-14:10 : Márk
    12:45-13:15 : Bátor
    12:48-13:47 : Kata
    12:50-13:10 : Gergely
8. feladat: Napi bevétel: 177600 Ft
10. feladat: Statisztika
    A : 6
    B : 4
    C : 4
    D : 6
    E : 3
    F : 6
    G : 4

```

Minta F . txt állomány:

```
10:15-10:55 : Bence
10:42-11:10 : Benjámin
11:16-11:40 : Zsombor
11:52-12:02 : Dániel
12:48-13:47 : Kata
15:00-16:00 : Levente
```

Megoldás:

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace vizibiciklikolcsonzo
{
    //1. Készítsen grafikus vagy konzolalkalmazást (projektet) a következő feladatok
    megoldásához,
    //amelynek projektjét vizibiciklikolcsonzo néven mentse el!
    /*Név;JAzon;EÓra;EPerc;VÓra;Vperc
    Mihály;C;9;55;10;56
    Dávid;A;10;1;10;58*/
    class kolcsonzes
    {
        public string nev { get; set; }
        public char jazon { get; set; }
        public int eora { get; set; }
        public int eperc { get; set; }
        public int vora { get; set; }
        public int vperc { get; set; }
        public kolcsonzes(string sor)
        {
            string[] sorelemek = sor.Split(';');
            this.nev = sorelemek[0];
            this.jazon = Convert.ToChar(sorelemek[1]);
            this.eora = Convert.ToInt32(sorelemek[2]);
            this.eperc = Convert.ToInt32(sorelemek[3]);
            this.vora = Convert.ToInt32(sorelemek[4]);
            this.vperc = Convert.ToInt32(sorelemek[5]);
        }
    }
}
class Program
{
    public static List<kolcsonzes> adatok = new List<kolcsonzes>();
    static void Main(string[] args)
    {
        //4. Olvassa be a kolcsonzesek.txt állomány sorait és tárolja az adatokat
        //egy Kolcsonzes osztályon alapuló összetett adatszerkezetben!
        //Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!
        string[] fajl = File.ReadAllLines("kolcsonzesek.txt", Encoding.UTF8);
        for (int i = 1; i < fajl.Length; i++)
        {
            adatok.Add(new kolcsonzes(fajl[i]));
        }
        int kolcsonzesekszama = adatok.Count();
        //5. Határozza meg és írja ki, hogy hány kölcsönzés adatai találhatóak a
        forrásállományban!
        Console.WriteLine("5. feladat: Napi kölcsönzések száma: {0}",kolcsonzesekszama);
        //6. Kérjen be egy nevet és írja ki, hogy az illető aznap
        * mettől meddig bérelt vízibiciklit!
        * Elképzelhető, hogy az illető többször is kölcsönzött aznap,
        * ebben az esetben minden kölcsönzés adatát jelenítse meg!
```

```

    * Ha aznap egyszer sem kölcsönzött, akkor a „Nem volt ilyen nevű kölcsönző!”
szöveg jelenjen meg!*/
Console.WriteLine("6. feladat: Kérek egy nevet: ");
string keresettnev = Console.ReadLine();
int db = 0;
Console.WriteLine("\t{0} kölcsönzései:", keresettnev);
for (int i = 0; i < kolcsonzesekszama; i++)
{
    if (adatok[i].nev == keresettnev)
    {
        Console.WriteLine("\t{0:00}:{1:00}-{2:00}:{3:00}",
            adatok[i].eora, adatok[i].eperc, adatok[i].vora, adatok[i].vperc);
        db++;
    }
}
if (db == 0) Console.WriteLine("\tNem volt ilyen nevű kölcsönző!");
/*7. Kérjen be egy időpontot óra:perc alakban, majd írja ki,
* hogy ekkor mely járművek voltak vízen, és azokat meddig és kik kölcsönözték ki!
* Feltételezheti, hogy a megadott időpontban legalább egy jármű a vízen volt.
* A kiírásnál használjon vezető nullákat az egyjegyű óra és perc értékek előtt!*/
Console.WriteLine("7. feladat: Kérek egy időpontot óra:perc alakban: ");
string oraperc = Console.ReadLine();
Console.WriteLine("\tA vízen lévő járművek:");
var tmp = oraperc.Split(':');
int elvitelperc, visszaperc, keresettperc;
foreach (var item in adatok)
{
    elvitelperc = item.eora * 60 + item.eperc;
    visszaperc = item.vora * 60 + item.vperc;
    keresettperc = Convert.ToInt32(tmp[0]) * 60 + Convert.ToInt32(tmp[1]);
    if (elvitelperc <= keresettperc && visszaperc >= keresettperc)
    {
        Console.WriteLine("\t{0:00}:{1:00}-{2:00}:{3:00} : {4}",
            item.eora, item.eperc, item.vora, item.vperc, item.nev);
    }
}
/*8. A járművek kölcsönzési díja egységesen 2400 Ft minden megkezdett
félóránként.
* Határozza meg a napi bevétel összegét és írja a képernyőre!*/
//double ossz = 0;
int ossz = 0;
int felorakszama = 0;
foreach (var item in adatok)
{
    ossz = ((item.vora - item.eora) * 60) + (item.vperc - item.eperc);
    //felorakszama += (int)Math.Ceiling(ossz / 30);
    felorakszama += (ossz+29) / 30;
}
Console.WriteLine("8. feladat: Napi bevétel: {0} Ft", felorakszama*2400);
/*9. Sajnos az F jelű járművet valaki megrongálta.
* Hozzon létre F. txt néven UTF-8 kódolású szöveges állományt,
* amely tartalmazza a lehetséges elkövetőket és azt, hogy mettől meddig volt
náluk a jármű!
* Az állomány szerkezete a kiadott minta szerinti legyen!*/
FileStream fnev = new FileStream("f.txt", FileMode.Create);//létrehozásra
megnyitjuk a fájlt
StreamWriter fajlbairo = new StreamWriter(fnev,
System.Text.Encoding.UTF8);//fájlbairáshoz
foreach (var item in adatok)
{
    if(item.jazon == 'F')
    {
        fajlbairo.WriteLine("{0:00}:{1:00}-{2:00}:{3:00} : {4}",
            item.eora, item.eperc, item.vora, item.vperc, item.nev);
    }
}
fajlbairo.Close();//fájl lezárása
fnev.Close();
/*10. Készítsen statisztikát, melyben megjeleníti,

```

```
        * hogy az egyes járműveket hányszor kölcsönözték a nap során!  
        * Oldja meg, hogy a járművek azonosítói ábécérendben kövessék egymást a  
statisztika kiírásakor!*/  
        Console.WriteLine("10. feladat: Statisztika");  
        adatok.GroupBy(x => x.jazon).OrderBy(x => x.Key).ToList().ForEach(x =>  
Console.WriteLine("\t{0} : {1}", x.Key, x.Count()));  
        Console.ReadKey();  
    }  
}  
}
```