

## Pilóták programozásfeladat

A következő feladatban a Forma-I pilótáinak adataiból készített szöveges állományból kell adatokat kinyernie.

A megoldás során vegye figyelembe a következőket:

- A képernyőre Írást igénylő részfeladatok eredményének megjelenítése előtt Írja a képernyőre a feladat sorszámát (például: 3. feladat:)!
- Az egyes feladatokban a kiírásokat a minta szerint készítse el!
- Az ékezetmentes azonosítók és kiírások is elfogadottak.
- A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.
- Megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!

A pilotak.csv UTF-8 kódolású forrásállomány soraiban a következő sorrendben találja meg az adatokat:

- a pilóta neve (név), például: Lewis Hamilton
- a pilóta születési dátuma (születési\_dátum), például: 1985 . 01 . 07
- a pilóta nemzetisége (nemzetiség), például: brit
- a pilóta rajtszáma (rajtszám), például: 44  
Csak az aktuális évben aktív pilótáknak van rajtszámuk, a többiek esetében a rajtszám mező értéke üres. Több pilótának is lehet azonos rajtszáma.

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

1. Készítsen konzolalkalmazást (projektet) a következő feladatok megoldásához, melynek projektjét VersenyzoK néven mentse el!
2. Olvassa be a pilotak.csv állomány sorait és tárolja az adatokat egy olyan összetett adatszerkezetben (pl. vektor, lista stb.), amely használatával a további feladatok megoldhatók! Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!
3. Határozza meg és írja ki a képernyőre a minta szerint, hogy az állomány hány adatsort tartalmaz!
4. Határozza meg és írja ki a minta szerint, hogy az állomány utolsó sorában melyik pilóta neve szerepel!
5. Határozza meg és írja ki a minta szerint, hogy mely pilóták születtek a XIX. században (azaz 1901. január 1-je előtt)! Feltételezheti, hogy a van olyan pilóta, aki 1901 előtt született. Írja ki a minta szerint a pilóták születési dátumát is!
6. Határozza meg, és írja ki a minta szerint, hogy a legkisebb értékű rajtszám pilótájának mi a nemzetisége! Ne vegye figyelembe a rajtszám nélküli pilóták adatait! Feltételezheti, hogy a legkisebb értékű rajtszámot csak egy pilóta kapta meg az idényben!
7. Egy-egy rajtszámot több pilóta is megkaphat az idényben. Határozza meg és írja ki a minta szerint, hogy melyek ezek a rajtszámok!

Minta

```

3. feladat: 847
4. feladat: Alexander Albon
5. feladat:
    Louis Chiron (1899.08.03)
    Arthur Legat (1898.11.01)
    Hans von Stuck (1900.12.27)
    Philippe Étancelin (1896.12.28)
    Bill Aston (1900.03.29)
    Adolf Brudes (1899.10.15)
    Piero Dusio (1899.10.13)
    Luigi Fagioli (1898.06.09)
    Clemente Biondetti (1898.08.18)
6. feladat: belga
7. feladat: 88 99 10 4 28

```

Megoldás:

```

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

namespace Versenyzo
{
    class Forma1
    {
        public Forma1(string sor)
        {
            string[] sozelemek = sor.Split(';');
            this.Nev = sozelemek[0];
            this.Szuldat = Convert.ToDateTime(sozelemek[1]);
            this.Nemzetiseg = sozelemek[2];
            if (sozelemek[3] != "")
                this.Rajtszam = Convert.ToInt32(sozelemek[3]);
            else this.Rajtszam = 0;
        }
        //név;születési_dátum;nemzetiség;rajtszám
        //Lewis Hamilton;1985.01.07;brit;44
        public string Nev { get; set; }
        public DateTime Szuldat { get; set; }
        public string Nemzetiseg { get; set; }
        public int Rajtszam { get; set; }
    }

    class Program
    {
        public static List<Forma1> adatok = new List<Forma1>();
        static void Main(string[] args)
        {
            StreamReader olvas = new StreamReader("pilotak.csv", Encoding.UTF8);
            string fejléc = olvas.ReadLine();
            while (!olvas.EndOfStream)
            {
                adatok.Add(new Forma1(olvas.ReadLine()));
            }
            int i, j;
            int adatokszama = adatok.Count;
            /*Console.WriteLine("név          születési_dátum      nemzetiség
rajtszám");
            for(i=0;i<adatokszama;i++)
                Console.WriteLine("{0,-25}{1,-25}{2,-15}{3}",
                    adatok[i].Nev, adatok[i].Szuldat, adatok[i].Nemzetiseg,
                    adatok[i].Rajtszam);*/

```

```

//3. Határozza meg és írja ki a képernyőre a minta szerint, hogy az
állomány hány adatsort tartalmaz!
Console.WriteLine("3. feladat: {0}", adatokszama);

//4.Határozza meg és írja ki a minta szerint, hogy az állomány utolsó
sorában melyik pilóta neve szerepel!
Console.WriteLine("4. feladat: {0}", adatok[adatokszama-1].Nev);

//5. Határozza meg és írja ki a minta szerint, hogy mely pilóták
születtek a XIX. században (azaz 1901. január 1-je előtt)!
//Feltételezheti, hogy a van olyan pilóta, aki 1901 előtt született. Írja
ki a minta szerint a pilóták születési dátumát is!
Console.WriteLine("5. feladat:");
for (i = 0; i < adatokszama; i++)
    if(adatok[i].Szuldat.Year<1901)
        Console.WriteLine("\t{0} ({1:yyyy.MM.dd})", adatok[i].Nev,
adatok[i].Szuldat);

/*6. Határozza meg, és írja ki a minta szerint, hogy a legkisebb értékű
rajtszám pilótájának mi a nemzetisége!
* Ne vegye figyelembe a rajtszám nélküli pilóták adatait! Feltételezheti,
* hogy a legkisebb értékű rajtszámot csak egy pilóta kapta meg az
idényben!*/

int minrajtszam = adatok[0].Rajtszam;
int mini = 0;
for (i = 1; i < adatokszama; i++)
    if(adatok[i].Rajtszam!=0)
    {
        if (adatok[i].Rajtszam < minrajtszam)
        {
            minrajtszam = adatok[i].Rajtszam;
            mini = i;
        }
    }
Console.WriteLine("6. feladat: {0}", adatok[mini].Nemzetiseg);

//7. Egy-egy rajtszámot több pilóta is megkaphat az idényben.
//Határozza meg és írja ki a minta szerint, hogy melyek ezek a rajtszámok!
Console.Write("7. feladat: ");
List<Formal> ujlista = new List<Formal>();
for (i = 0; i < adatokszama; i++)
    if (adatok[i].Rajtszam!=0)
        ujlista.Add(adatok[i]);
ujlista.GroupBy(x => x.Rajtszam).Where(x => x.Count() >
1).ToList().ForEach(x => Console.Write("{0} ",x.Key));

Console.ReadKey();
}
}
}

```