

Programozás

Alonso¹

A következő feladatban a Formula-1 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: 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 úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!

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

- a versenyző neve (név), például: Fernando Alonso
 - a versenyző születési dátuma (születés), például: 1981.07.29
 - a versenyző nemzetisége (nemzetiség), például: spanyol
 - a versenyző nevének rövidítése (kód), például: ALO
- Csak az utóbbi években aktív versenyzők nevének van rövidítése, a többiek esetében a kód mező értéke üres.

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

1. Készítsen konzolalkalmazást (projektet) a következő feladatok megoldásához, melynek projektjét Alonso néven mentse el!
2. Hozzon létre saját osztályt, melynek segítségével el tudja tárolni egy-egy versenyző adatait (versenyző neve, születési dátuma, nemzetisége, rajtszáma (ha van))! *Ha nem tud osztályt létrehozni, akkor az 3–10. feladatokat megoldhatja anélkül is, de ebben az esetben a megoldás nem lesz teljes értékű.*
3. Hozzon létre konstruktort, melynek segítségével az Ön által létrehozott osztály adattagjai feltölthetők!
4. Olvassa be a `versenyzok.csv` állomány sorait és tárolja az adatokat (versenyzőket) az Ön által létrehozott osztályból képzett adatszerkezetben, amely használatával a további feladatok megoldhatók! Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!

¹ Források: Wikipédia

https://hu.wikipedia.org/wiki/A_Formula-1_rajtszámainak_listája

5. Határozza meg és írja ki a képernyőre a minta szerint, hogy az állomány hány adatsort tartalmaz!
6. Határozza meg és írja ki a minta szerint, hogy melyik pilótához tartozik az ALO rövidítés (kód)! Feltételezheti, hogy szerepel ilyen egyedi rövidítés az állományban.
7. Az állományban rögzített adatok alapján gyanúsán sok versenyző született január 1-én. Határozza meg és írja ki a minta szerint, hogy mely versenyzők születtek január 1-én! A kiírásban a versenyzők neve és születési dátuma is szerepeljen! A megjelenítésnél nem kell rendezést alkalmaznia.
8. Kérjen be a felhasználótól a minta szerint egy szöveges adatot (rövidítést)! Az adatbevitelt mindaddig ismételje, amíg üres karakterláncot ad meg a felhasználó!
9. Keresse meg és írja ki a minta szerint azon versenyző adatait, akihez az előző feladatban bekért rövidítés tartozik! Ha nem található ilyen versenyző, akkor a „Nem található a keresett versenyző!” szöveget írja ki!
10. Határozza meg és írja ki a minta szerint, hogy mely nemzetek adtak több mint 25 versenyzőt a Formula-1-es sportágnak! A válaszokat vesszővel elválasztva, egymás mellé írja ki! A kiírásban felesleges vessző ne szerepeljen! A nemzetek felsorolása tetszőleges sorrendben történhet.

Minta találat esetén

```

5. feladat: 847
6. feladat: Fernando Alonso
7. feladat:
    Baumgartner Zsolt (1981. 01. 01.)
    Jean-Marc Gounon (1963. 01. 01.)
    Hans-Joachim Stuck (1951. 01. 01.)
    Jacky Ickx (1945. 01. 01.)
    Carlo Franchi (1938. 01. 01.)
    Hap Sharp (1928. 01. 01.)
    Marcel Balsa (1909. 01. 01.)
8. feladat: Kérek egy rövidítést:
MAG
9. feladat:
    név: Jan Magnussen
    nemzetiség: dán
    születési dátum: 1973. 07. 04.
10. feladat: brit, német, francia, brazil, olasz, amerikai

```

Minta nem létező rövidítés esetén

```
5. feladat: 847
6. feladat: Fernando Alonso
7. feladat:
    Baumgartner Zsolt (1981. 01. 01.)
    Jean-Marc Gounon (1963. 01. 01.)
    Hans-Joachim Stuck (1951. 01. 01.)
    Jacky Ickx (1945. 01. 01.)
    Carlo Franchi (1938. 01. 01.)
    Hap Sharp (1928. 01. 01.)
    Marcel Balsa (1909. 01. 01.)
8. feladat: Kérek egy rövidítést:
HUN
9. feladat:
Nem található a keresett versenyző!
10. feladat: brit, német, francia, brazil, olasz, amerikai
```

Megoldás2:

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

namespace Alonso
{
    internal class adatsor
    {
        /*név;születés;nemzetiség;kód
Lewis Hamilton;1985.01.07;brit;HAM*/
        //Hozzon létre saját osztályt, melynek segítségével el tudja tárolni egy-egy versenyző adatait
        //(versenyző neve, születési dátuma, nemzetisége, kódja)
        public string nev { get; set; }
        public DateTime szuletes { get; set; }
        public string nemzetiseg { get; set; }
        public string kod { get; set; }
        public adatsor(string sor)
        {
            //Hozzon létre konstruktort, melynek segítségével az Ön által létrehozott osztály adatait feltölthetők!
            string[] sorelemek = sor.Split(';');
            this.nev= sorelemek[0];
            this.szuletes = Convert.ToDateTime(sorelemek[1]);
            this.nemzetiseg = sorelemek[2];
            this.kod = sorelemek[3];
        }
    }
}
```

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Security.Cryptography.X509Certificates;

namespace Alonso
{
    internal class Program
    {
        public static List<adatsor> adatok=new List<adatsor>();
        static void Main(string[] args)
        {
            //Olvassa be a versenyzők. csv állomány sorait
            string[] fajlbol = File.ReadAllLines("versenyzok.csv",Encoding.UTF8).Skip(1).ToArray();
            int i = 0, adatokszama = 0;
            //tárolja az adatokat (versenyzőket) az Ön által létrehozott osztályból képzett adatszerkezetben,
            //amely használatával a további feladatok megoldhatók!
            //Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!
```

```

for ( i = 0; i < fajlbol.Count(); i++)
{
    adatok.Add(new adatsor(fajlbol[i]));
}
//Határozza meg és írja ki a képernyőre a minta szerint, hogy az állomány hány adatsort tartalmaz!
adatokszama = adatok.Count();
Console.WriteLine("5. feladat:" +adatokszama);
//Határozza meg és írja ki a minta szerint, hogy melyik pilótához tartozik az ALO rövidítés (kód)!
//Feltételezheti, hogy szerepel ilyen egyedi rövidítés az állományban.
i = 0;
int sorszam = 0;
bool van=false;
while(i<adatokszama&&!van)
{
    if (adatok[i].kod=="ALO")
    {
        van = true;
        sorszam = i;
    }
    i++;
}
if (van) Console.WriteLine("6. feladat: " + adatok[sorszam].nev);
Console.WriteLine("6. másképpen:");
Console.WriteLine(adatok.Where(x => x.kod == "ALO").First().nev);
//Határozza meg és írja ki a minta szerint, hogy mely versenyzők születtek január 1-én!
//A kiírásban a versenyzők neve és születési dátuma is szerepeljen!
Console.WriteLine("7. feladat:");
foreach (var a in adatok)
{
    if(a.szuleset.Month==1&&a.szuleset.Day==1)
        Console.WriteLine("{0} ({{1:yyyy.MM.dd}})", a.nev,a.szuleset);
}
Console.WriteLine("7. másképpen:");
adatok.Where(x => x.szuleset.Month == 1 && x.szuleset.Day == 1).
    ToList().ForEach(x => Console.WriteLine("{0} ({{1:yyyy.MM.dd}})", x.nev, x.szuleset));
//Kérjen be a felhasználótól a minta szerint egy szöveges adatot (rövidítést)!
//Az adatbevitelt mindaddig ismétlje, amíg üres karakterláncot ad meg a felhasználó!
string rovidites = "";
Console.WriteLine("8. feladat: Kérek egy rövidítést: ");
do
{
    rovidites = Console.ReadLine();
}while(rovidites=="");
//Keresse meg és írja ki a minta szerint azon versenyző adatait,
//akihez az előző feladatban bekért rövidítés tartozik!
//Ha nem található ilyen versenyző, akkor a „Nem található a keresett versenyző !” szöveget írja ki!
Console.WriteLine("9. feladat:");
i = 0;
sorszam = 0;
van = false;
while (i < adatokszama && !van)
{
    if (adatok[i].kod == rovidites)
    {
        van = true;
        sorszam = i;
    }
    i++;
}
if (van)
{
    Console.WriteLine("név: " + adatok[sorszam].nev);
    Console.WriteLine("nemzetiség: " + adatok[sorszam].nemzetiseg);
    Console.WriteLine("születési dátum: {0:yyyy.MM.dd}", adatok[sorszam].szuleset);
}
else Console.WriteLine("Nem található a keresett versenyző");
Console.WriteLine("9. másképpen:");//az összes versenyzőt írja ki
int db = 0;
adatok.Where(x => x.kod == rovidites).ToList().
    ForEach(x => Console.WriteLine("név: {0}" +
        "\n\nnemzetiség: {1}" +
        "\n\nszületési dátum: {2:yyyy.MM.dd}", x.nev, x.nemzetiseg, x.szuleset,db++));
if (db == 0) Console.WriteLine("Nem található a keresett versenyző");
//Határozza meg és írja ki a minta szerint, hogy mely nemzetek adtak több mint 25 versenyzőt a Formula-1-es sportágnak!
//A válaszokat vesszővel elválasztva, egymás mellé írja ki! A kiírásban felesleges vessző ne szerepeljen!
List<string> nemzetek=new List<string>();//új lista
adatok.GroupBy(x=>x.nemzetiseg).Where(x=>x.Count(>25).ToList().ForEach(x=>nemzetek.Add(x.Key));//új listához adom
Console.WriteLine("10. feladat:");
for ( i = 0; i < nemzetek.Count(); i++)
{
    if(i<nemzetek.Count()-1) Console.WriteLine(nemzetek[i]+", ");
    else Console.WriteLine(nemzetek[i]);
}

```

```
//Mely nemzeteknek hány pilótája szerepel az adatok között?
```

```
Console.WriteLine("\n11. feladat: Statisztika:");
```

```
adatok.GroupBy(x => x.nemzetiseg).OrderByDescending(x=>x.Count()).ToList().ForEach(x => Console.WriteLine(x.Key + ":" + x.Count()));
```

```
Console.ReadKey();
```

```
}
```

```
}
```

```
}
```