Kutyák

A következő feladatban kutyák állatorvosi adatait tartalmazó szövegfájlok feldolgozását kell elvégeznie. 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 kiírás is elfogadott.
- A program megírásakor a fájlban 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!

1. A feladat megoldásához hozzon létre grafikus vagy konzolalkalmazást (projektet) Kutyák azonosítóval!
2. Három állománnyal kell dolgoznia. Az első KutyaNevek.csv állomány tartalmazza a kutyák lehetséges nevének listáját} Az UTF-8 kódolású fájl soraiban egy azonosító és a kutyanév szerepel pontosvesszővel elválasztva egymástól. A fájlban maximum 500 sor lehetséges. Ügyeljen arra, hogy a fájl első sora az adatok fejlécét tartalmazza! Olvassa be a KutyaNevek. csv állományban található adatokat és tárolja el egy megfelelően megválasztott adatszerkezetben!
3. Határozza meg és írja ki a képernyőre a minta szerint, hogy hány kutyanév található az állományban!
4. A második állomány az UTF-8 kódolású, pontosvesszővel tagolt KutyaFajták.csv tartalmazza az egyes fajták magyar és eredeti nevét[1]. Minden fajta saját azonosítóval van ellátva. A fájl első sora fejlécet tartalmaz. A fájlban maximum 500 sor lehetséges. Olvassa be a KutyaFaj ták. csv állományban található adatokat és tárolja el egy megfelelően megválasztott adatszerkezetben!
5. A harmadik állomány szintén UTF-8 kódolású, pontosvesszővel tagolt szöveges állomány. A fájl neve Kutyák. csv. A fájl első sora fejlécet tartalmaz. A fájlban maximum 500 sor lehetséges. A fájlban soronként a következő adatok találhatók:

   ● a vizsgálat azonosítója, ● a kutya fajtájának azonosítója (a KutyaFajták. csv állomány első oszlopa alapján), ● a kutya nevének azonosítója (a KutyaNevek. csv állomány első oszlopa alapján), ● a kutya életkora ● és az orvosi vizsgálat ideje.

   Feltételezheti, hogy egy kutya csak egyszer szerepel az állományban. Olvassa be a Kutyák.csv állományban található adatokat és tárolja el egy megfelelően megválasztott adatszerkezetben!
6. Határozza meg és írja ki a képernyőre a minta szerint, hogy mennyi a kutyák átlagéletkora! Az életkort kerekítse 2 tizedesjegyre!
7. Határozza meg és írja ki a képernyőre a minta szerint, hogy a legidősebb kutyának mi neve és a fajtája! Feltételezheti, hogy nincs két legidősebb azonos korú kutya.
8. Határozza meg és Írja ki a képernyőre a minta szerint, hogy 2018. január 10-én fajtánként hány kutya volt az állatorvosi rendelőben!
9. Határozza meg és írja ki a képernyőre a minta szerint, hogy melyik nap volt a rendelő a legjobban leterhelve, és hány kutyát láttak el aznap!
10. Névsta tisztika . txt néven hozzon létre egy új UTF-8 kódolású, pontosvesszővel tagolt állományt, amely tartalmazza a vizsgált kutyák nevét és az adott nevű kutyák számát! Az állományban népszerűség alapján csökkenő sorrendben legyenek a nevek, azaz a legtöbb kutya által kapott név legyen elöl!

Minta:

```
3. feladat: Kutyanevek száma: 288
6. feladat: A kutyák átlagéletkora: 9,06
7. feladat: A legidősebb kutya neve és fajtája: Jessy, Smalandi kopó
8. feladat:  január 10-én vizsgált kutya fajták
        Boszniai drótszőrű kopó: 1 kutya
        Bali hegyikutya: 1 kutya
        Fekete-cser mosómedvekopó: 1 kutya
        Tosa inu: 1 kutya
9. feladat: Legjobban leterhelt nap: 2017-06-29: 6 kutya
```

Megoldás:

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

namespace kutyak
{
    class Kutyafajtak
    {
        public Kutyafajtak(string sor)
        {
            string[] sorelemek = sor.Split(';');
            this.fajtaid =Convert.ToInt32(sorelemek[0]);
            this.fajtanev = sorelemek[1];
            this.eredetifajtanev = sorelemek[2];

        }
        //id;név;eredeti név pl: 1;Abruzzói juhászkutya; Cane da pastore Maremmano-
Abruzzese

        public int fajtaid { get; set; }
```

```
    public string fajtanev { get; set; }
    public string eredetifajtanev { get; set; }

}
class Kutyanevek
{
    public Kutyanevek(string sor)
    {
        string[] sorelemek = sor.Split(';');
        this.nevekid = Convert.ToInt32(sorelemek[0]);
        this.kutyanevek = sorelemek[1];

    }
    //id;kutyanév pl: 1;Akina

    public int nevekid { get; set; }
    public string kutyanevek { get; set; }

}
class Kutyak
{
    public Kutyak(string sor)
    {
        string[] sorelemek = sor.Split(';');
        this.KutyaId = Convert.ToInt32(sorelemek[0]);
        this.FajtaId = Convert.ToInt32(sorelemek[1]);
        this.NevId = Convert.ToInt32(sorelemek[2]);
        this.Eletkor = Convert.ToInt32(sorelemek[3]);
        this.Ellenorzes = Convert.ToDateTime(sorelemek[4]);

    }
    //id;fajta_id;név_id;életkor;utolsó orvosi ellenőrzés
    //1;307;107;14;2017.11.27

    public int KutyaId { get; set; }
    public int FajtaId { get; set; }
    public string FajtaNev { get; set; }
    public int NevId { get; set; }
    public string KutyaNev { get; set; }
    public int Eletkor { get; set; }
    public DateTime Ellenorzes { get; set; }

}

class Program
{
    public static List<Kutyanevek> nevek = new List<Kutyanevek>();
    public static List<Kutyafajtak> fajtak = new List<Kutyafajtak>();
    public static List<Kutyak> kutyaadatok = new List<Kutyak>();
    static void Main(string[] args)
    {
        StreamReader olvas = new StreamReader("KutyaNevek.csv", Encoding.UTF8);
        string fejlec = olvas.ReadLine();//ha van fejléc
        while (!olvas.EndOfStream)
        {
            nevek.Add(new Kutyanevek(olvas.ReadLine()));
        }
        int i;//ciklusváltozó

        int nevekszama = nevek.Count;
        StreamReader olvas2 = new StreamReader("KutyaFajtak.csv", Encoding.UTF8);
        string fejlec2 = olvas2.ReadLine();//ha van fejléc
```

```
            while (!olvas2.EndOfStream)
            {
                fajtak.Add(new Kutyafajtak(olvas2.ReadLine()));
            }

            int fajtakszama = fajtak.Count;
            StreamReader olvas3 = new StreamReader("Kutyak.csv", Encoding.UTF8);
            string fejlec3 = olvas3.ReadLine();//ha van fejléc
            while (!olvas3.EndOfStream)
            {
                kutyaadatok.Add(new Kutyak(olvas3.ReadLine()));
            }

            int adatokszama = kutyaadatok.Count;
            int id;
            for (i = 0; i < adatokszama; i++)
            {
                for (int j = 0; j < fajtakszama; j++)
                    if (kutyaadatok[i].FajtaId == fajtak[j].fajtaid)
                        kutyaadatok[i].FajtaNev = fajtak[j].fajtanev;
                for(int j=0;j<nevekszama;j++)
                    if(kutyaadatok[i].NevId==nevek[j].nevekid)
                        kutyaadatok[i].KutyaNev = nevek[j].kutyanevek;
            }
            /*for (i = 0; i < adatokszama; i++)
            {
                Console.WriteLine("{0,-5}{1,-5}{2,-35}{3,-5}{4,-25}{5,-5}{6}",
                    kutyaadatok[i].KutyaId, kutyaadatok[i].FajtaId,
kutyaadatok[i].FajtaNev, kutyaadatok[i].NevId, kutyaadatok[i].KutyaNev,
kutyaadatok[i].Eletkor, kutyaadatok[i].Ellenorzes);
            }*/
            //3.    Határozza meg és írja ki a képernyőre a minta szerint, hogy hány
kutyanév található az állományban!
            Console.WriteLine("3. feladat: Kutyanevek száma: {0}",nevekszama);

            //6.    Határozza meg és írja ki a képernyőre a minta szerint, hogy mennyi
a kutyák átlagéletkora! Az életkort kerekítse 2 tizedesjegyre!
            double atlageletkor = 0;
            for (i = 0; i < adatokszama; i++)
                atlageletkor += kutyaadatok[i].Eletkor;
            Console.WriteLine("6. feladat: A kutyák átlagéletkora:
{0}",Math.Round(atlageletkor/adatokszama,2));

            //7.    Határozza meg és írja ki a képernyőre a minta szerint, hogy a
legidősebb kutyának mi neve és a fajtája!
            //Feltételezheti, hogy nincs két legidősebb azonos korú kutya.
            int max = kutyaadatok[0].Eletkor;
            int maxi = 0;
            for (i = 0; i < adatokszama; i++)
                if(kutyaadatok[i].Eletkor>max)
                {
                    max = kutyaadatok[i].Eletkor;
                    maxi = i;
                }
            Console.WriteLine("7. feladat: A legidősebb kutya neve és fajtája: {0},
{1}", kutyaadatok[maxi].KutyaNev, kutyaadatok[maxi].FajtaNev);

            //8.    Határozza meg és Írja ki a képernyőre a minta szerint, hogy 2018.
január 10-én fajtánként hány kutya volt az állatorvosi rendelőben!
            Console.WriteLine("8. feladat:  január 10-én vizsgált kutya fajták");
            List<Kutyak> ujlista = new List<Kutyak>();
```

```
            for (i = 0; i < adatokszama; i++)
            {
                if(kutyaadatok[i].Ellenorzes== DateTime.Parse("2018.01.10"))
                {
                    ujlista.Add(kutyaadatok[i]);
                }
            }
            ujlista.GroupBy(x => x.FajtaNev).ToList().ForEach(x =>
Console.WriteLine("\t{0}: {1} kutya",x.Key,x.Count()));


            //9.    Határozza meg és írja ki a képernyőre a minta szerint, hogy melyik
nap volt a rendelő a legjobban leterhelve, és hány kutyát láttak el aznap!
            Dictionary<string, int> napoklista = new Dictionary<string, int>();
            foreach (Kutyak kutya in kutyaadatok)
            {
                string key = kutya.Ellenorzes.ToString("yyyy-MM-dd");

                if (napoklista.ContainsKey(key))
                    napoklista[key]++;
                else
                    napoklista.Add(key, 1);
            }
            Console.WriteLine("9. feladat: Legjobban leterhelt nap: " +
napoklista.FirstOrDefault(x => x.Value == napoklista.Values.Max()).Key + ": " +
napoklista.Values.Max() + " kutya");

            //Console.WriteLine("9. feladat: Legjobban leterhelt napok: ");
            //kutyaadatok.GroupBy(x => x.Ellenorzes).OrderByDescending(x =>
x.Count()).ToList().ForEach(x => Console.WriteLine("{0}: {1} kutya",x.Key,x.Count()));

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