

1. Duomenų klasė

Siekiami studijų rezultatai:

- pagrindiniai objektinio programavimo principai;
- objektinio programavimo principų taikymas programų kūrimui;
- sudaryto algoritmo realizavimas C# programavimo kalba;
- programų derinimas ir testavimas.

Susipažinsite su:

- *CSV* (angl. comma separated values) formato failais, kuriuose saugomi objektų duomenys ir į kuriuos surašomi skaičiavimų rezultatai;
- dinaminio masyvo tipu *List*, kuris bus naudojamas objektų rinkiniui saugoti;
- datai ir laikui saugoti skirtu duomenų tipu *DateTime*;
- išvardijimo duomenų tipu;
- didžiausios (mažiausios) reikšmės paieškos algoritmu;
- unikalių reikšmių (be pasikartojimų) radimo algoritmu.

1.1. Pažintis su objektų rinkiniu

Užduotis. Gyvūnų registras.

Gyvūnų duomenų bazėje registruojami šunys. Žinomas kiekvieno šuns *registracijos nr.*, *vardas*, *veislė*, *gimimo data*, *lytis*.

Parašykite programą, kuri:

- Nuskaitytų pradinį duomenis iš failo ir atspausdintų šunų sąrašą ekrane, suformatuota lentele.
- Suskaičiuotų ir atspausdintų šunų kiekį. Taip pat atspausdintų kiek iš jų patinų, kiek patelių.
- Rastų seniausią šunį, ir atspausdintų ekrane jo vardą, veislę ir amžių.
- Surastų ir atspausdintų registre esančių šunų veislių sąrašą (be pasikartojimų).
- Paprašytų vartotojo pasirinkti veislę.
- Atrinktų vartotojo pasirinktos veislės šunis. Atspausdintų jų sąrašą ekrane, suformatuota lentele.
- Atspausdintų pasirinktos veislės sąrašą *CSV* faile, failo pavadinimas — pasirinkta veislė.

Pradiniai duomenys ir rezultatai.

Duomenys surašyti tekstiniam faile *Dogs.csv*. Kiekvienoje eilutėje užrašytas: *šuns registracijos nr.*, *vardas*, *veislė*, *gimimo data*, *lytis*.

Pradiniai duomenys
123;Reksas;Buldogas;2014-01-01;Male
124;Margis;Dalmantinas;2014-02-28;Male
125;Bitė;Senbernaras;2008-07-17;Female
320;Rikis;Taksas;2012-01-07;Male
415;Pifas;Taksas;2014-07-07;Male

Tiek pradinį duomenų, tiek rezultatų faile naudojamas *CSV* formatas. Naudojamas skyriklis – kabliataškis. Failų koduotė — *UTF8*.

Programos kūrimo eiga.

- Lyties išvardijimo tipo ir duomenų klasės kūrimas.
- Pradinių duomenų skaitymo ir spausdinimo ekrane metodų įgyvendinimas.
- Skaičiavimo metodų įgyvendinimas.
- Rezultatų spausdinimo į failą įgyvendinimas.

Pasiruošimas.

- Sukurkite sprendimą (*solution*) pavadinimu **Lab1.Exercises**.
- Sukurkite jame projektą pavadinimu **Lab1. Exercises.Register**.

Pirmas žingsnis. Duomenų klasės kūrimas.

- Išanalizuokime pradinius duomenis ir pasirinkime tipus jiems saugoti. Registracijos nr. naudosime *int*, vardui ir veislei — *string*, gimimo datai — *DateTime*. Lyčiai saugoti sukursime išvardijimo tipą *Gender*.
- Sukurkite naują išvardinimo tipą ***Gender*** (atskirame faile **Gender.cs**).

```
namespace Lab1.Exercises.Register
{
    enum Gender
    {
        Male = 1,
        Female = 2,
    }
}
```

- Sukurkite duomenų klasę ***Dog*** (atskirame faile **Dog.cs**). Joje pridėkite savybes *ID*, *Name*, *Breed*, *BirthDate* ir *Gender*. Sukurkite konstruktorių su tokiais pat parametrais.

```
class Dog
{
    public int ID {get; set;}
    public string Name {get; set;}
    public string Breed {get; set;}
    public DateTime BirthDate {get; set;}
    public Gender Gender {get; set;}

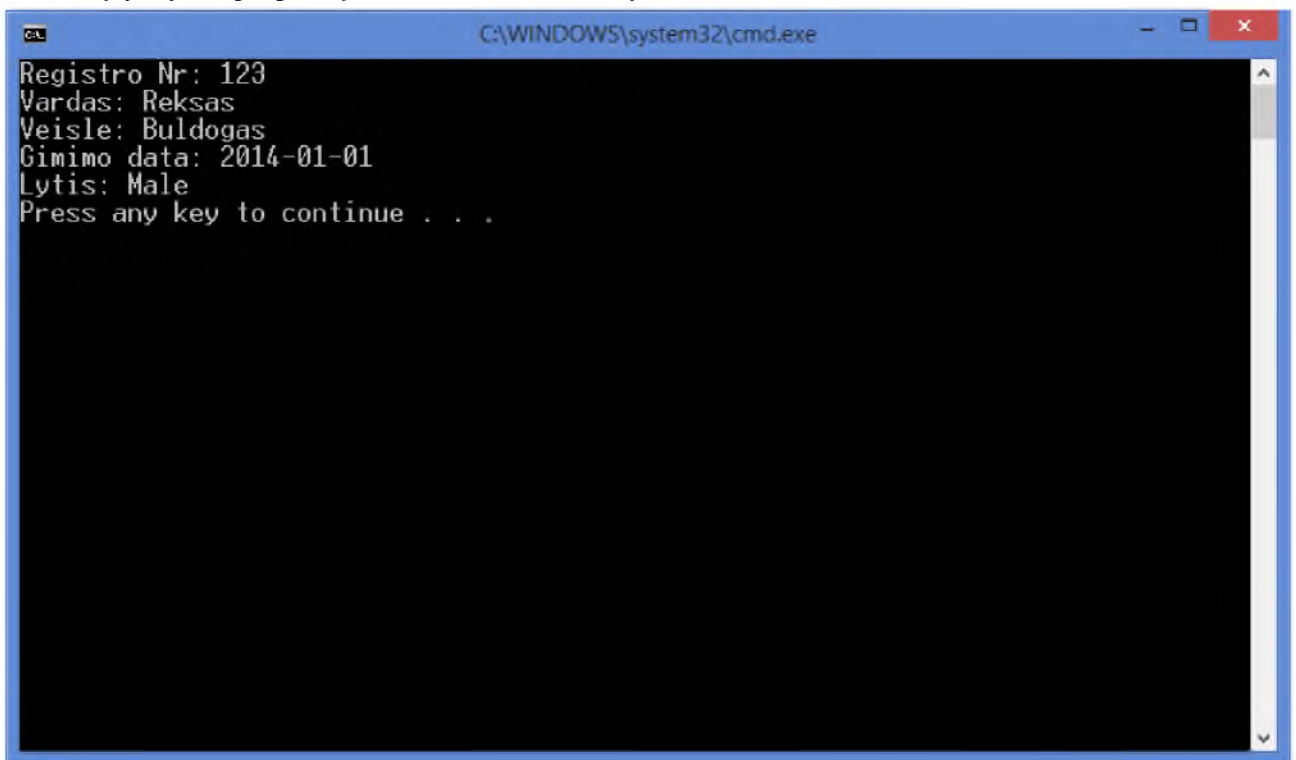
    public Dog(int id, string name, string breed, DateTime birthDate, Gender gender)
    {
        this.ID = id;
        this.Name = name;
        this.Breed = breed;
        this.BirthDate = birthDate;
        this.Gender = gender;
    }
}
```

- Išmėginkite duomenų klasę. **Program.cs** faile ištrinkite automatiškai sugeneruotą metodo *Main* kodą. Tuomet *Main* metode sukurkite *Dog* tipo objektą. Galiausiai atspausdinkite šio objekto duomenis į ekraną.

```
static void Main(string[] args)
{
    Dog test = new Dog(123, "Reksas", "Buldogas", new DateTime(2014,1,1), Gender.Male);

    Console.WriteLine("Registro Nr: {0}", test.ID);
    Console.WriteLine("Vardas: {0}", test.Name);
    Console.WriteLine("Veislė: {0}", test.Breed);
    Console.WriteLine("Gimimo data: {0:yyy-MM-dd}", test.BirthDate);
    Console.WriteLine("Lytis: {0}", test.Gender);
}
```

- Įvykdykite programą. Ekrane turėtumėte matyti šuns duomenis.



```
C:\WINDOWS\system32\cmd.exe
Registro Nr: 123
Vardas: Reksas
Veisle: Buldogas
Gimimo data: 2014-01-01
Lytis: Male
Press any key to continue . . .
```

- Pašalinkite **Program.cs** failo *Main* metodo turinį (patį metodą palikite).

Antras žingsnis. Pradinių duomenų skaitymas ir spausdinimas ekrane.

- Paruoškite ir pridėkite į projektą duomenų failą.
- Įvedimo ir išvedimo operacijoms sukurkite statinę klasę *InOutUtils* (atskirame faile *InOutUtils.cs*).
- Būtinai įtraukite *System.IO* biblioteką, skirtą darbui su failais ir duomenų srautais.

`using System.IO;`

- Parašykite duomenų skaitymo iš failo metodą.

```
static class InOutUtils
{
    public static List<Dog> ReadDogs(string fileName)
    {
        List<Dog> Dogs = new List<Dog>();
        string[] Lines = File.ReadAllLines(fileName, Encoding.UTF8);
        foreach (string line in Lines)
        {
            string[] Values = line.Split(';');
            int id = int.Parse(Values[0]);
            string name = Values[1];
            string breed = Values[2];
            DateTime birthDate = DateTime.Parse(Values[3]);

            Gender gender;
            Enum.TryParse(Values[4], out gender); //tries to convert value to enum

            Dog dog = new Dog(id, name, breed, birthDate, gender);
            Dogs.Add(dog);
        }

        return Dogs;
    }
}
```

```
double value = double.Parse(doubleToParse, CultureInfo.InvariantCulture);
//It does not depend on settings
// using System.Globalization;
```

- Parašykite duomenų spausdinimo į ekraną formatuojant lentelę metodą ir patalpinkite klasėje *InOutUtils*.

```
public static void PrintDogs(List<Dog> Dogs)
{
    Console.WriteLine(new string('-', 74));
    Console.WriteLine("| {0,8} | {1,-15} | {2,-15} | {3,-12} | {4,-8} |",
        "Reg.Nr.", "Vardas", "Veislė", "Gimimo data", "Lytis");
    Console.WriteLine(new string('-', 74));
    foreach (Dog dog in Dogs)
    {
        Console.WriteLine("| {0,8} | {1,-15} | {2,-15} | {3,-12:yyyy-MM-dd} | {4,-8} |",
            dog.ID, dog.Name, dog.Breed, dog.BirthDate, dog.Gender);
    }
    Console.WriteLine(new string('-', 74));
}
```

- Papildykite *Main* metodą kreipiniais į parašytus metodus.

```
static void Main(string[] args)
{
```

```

List<Dog> allDogs = InOutUtils.ReadDogs(@"Dogs.csv");

Console.WriteLine("Registro informacija:");
InOutUtils.PrintDogs(allDogs);

Console.WriteLine("Iš viso šunų: {0}", allDogs.Count);
}

```

- Įvykdyskite programą. Ekrane turėtumėte matyti pradinis duomenis atspausdintus lentele bei šunų kiekį.

```

C:\WINDOWS\system32\cmd.exe
Registro informacija:
-----
| 123 | Reksas      | Buldogas      | 2014-01-01 | Male |
| 124 | Margis      | Dalmantinas   | 2014-02-28 | Male |
| 125 | Bite        | Senbernaras   | 2008-07-17 | Female |
| 320 | Rikis       | Taksas        | 2012-01-07 | Male |
| 415 | Pifas       | Taksas        | 2014-07-07 | Male |
-----
Is viso sunu: 5
Press any key to continue . . .

```

Trečias žingsnis. Kiekio skaičiavimas.

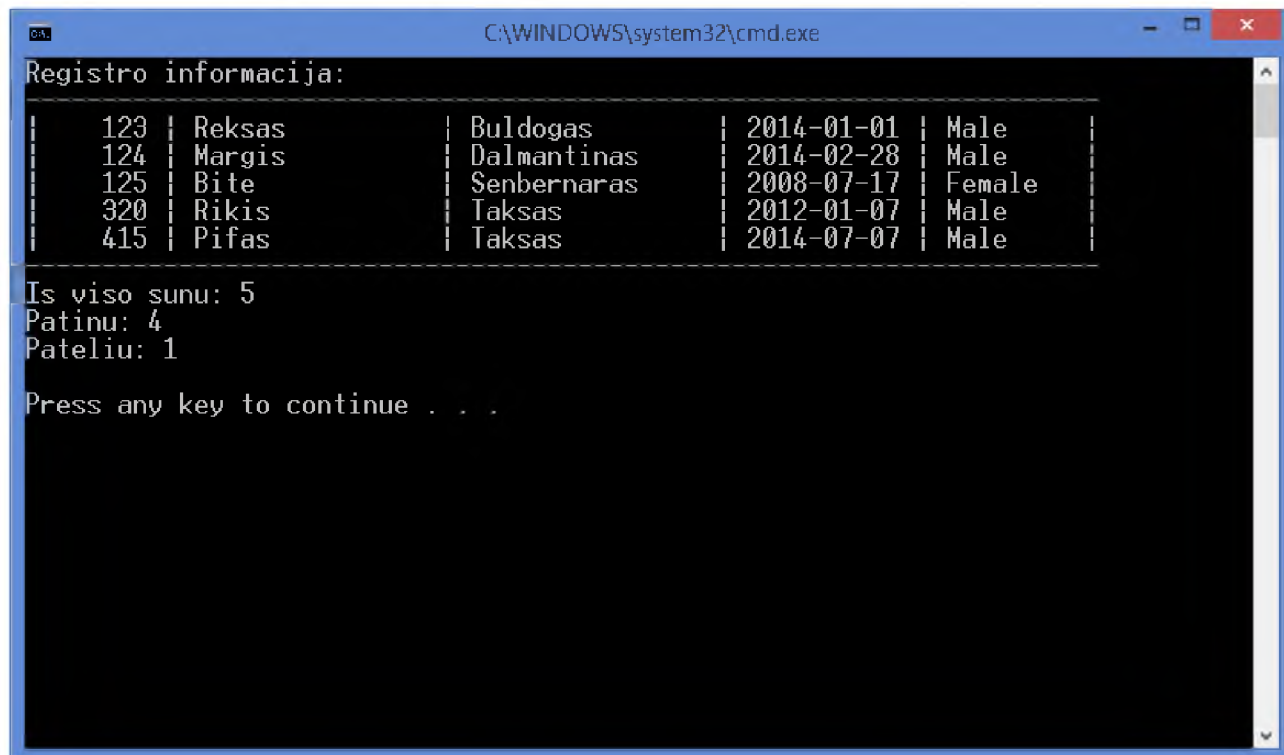
- Skaičiavimo metodams sukurkite statinę klasę *TaskUtils* (atskirame faile *TaskUtils.cs*).
- Šioje klasėje parašykite metodą, skaičiuojantį kiek yra duotos lyties šunų.

```
static class TaskUtils
{
    public static int CountByGender(List<Dog> Dogs, Gender gender)
    {
        int count = 0;
        foreach (Dog dog in Dogs)
        {
            if (dog.Gender.Equals(gender))
            {
                count++;
            }
        }
        return count;
    }
}
```

- Papildykite *Main* metodą:

```
Console.WriteLine("Patinų: {0}", TaskUtils.CountByGender(allDogs, Gender.Male));
Console.WriteLine("Patelių: {0}", TaskUtils.CountByGender(allDogs, Gender.Female));
Console.WriteLine();
```

- Įvykdysite programą ir patikrinkite gautus rezultatus.



```
C:\WINDOWS\system32\cmd.exe

Registro informacija:
+-----+-----+-----+-----+-----+
| 123 | Reksas | Bulldogas | 2014-01-01 | Male |
| 124 | Margis | Dalmantinas | 2014-02-28 | Male |
| 125 | Bite | Senbernaras | 2008-07-17 | Female |
| 320 | Rikis | Taksas | 2012-01-07 | Male |
| 415 | Pifas | Taksas | 2014-07-07 | Male |
+-----+-----+-----+-----+-----+

Is viso sunu: 5
Patinu: 4
Pateliu: 1

Press any key to continue . . .
```

Ketvirtas žingsnis. Min/Max reikšmės radimas.

- Klasę *TaskUtils* papildykite metodu, skirtu rasti seniausią šunį.

```
public static Dog FindOldestDog(List<Dog> Dogs)
{
    Dog oldest = Dogs[0]; // means least value
    for (int i = 1; i < Dogs.Count; i++)
    {
        if (DateTime.Compare(Dogs[i].BirthDate, oldest.BirthDate) < 0)
        {
            oldest = Dogs[i];
        }
    }
    return oldest;
}
```

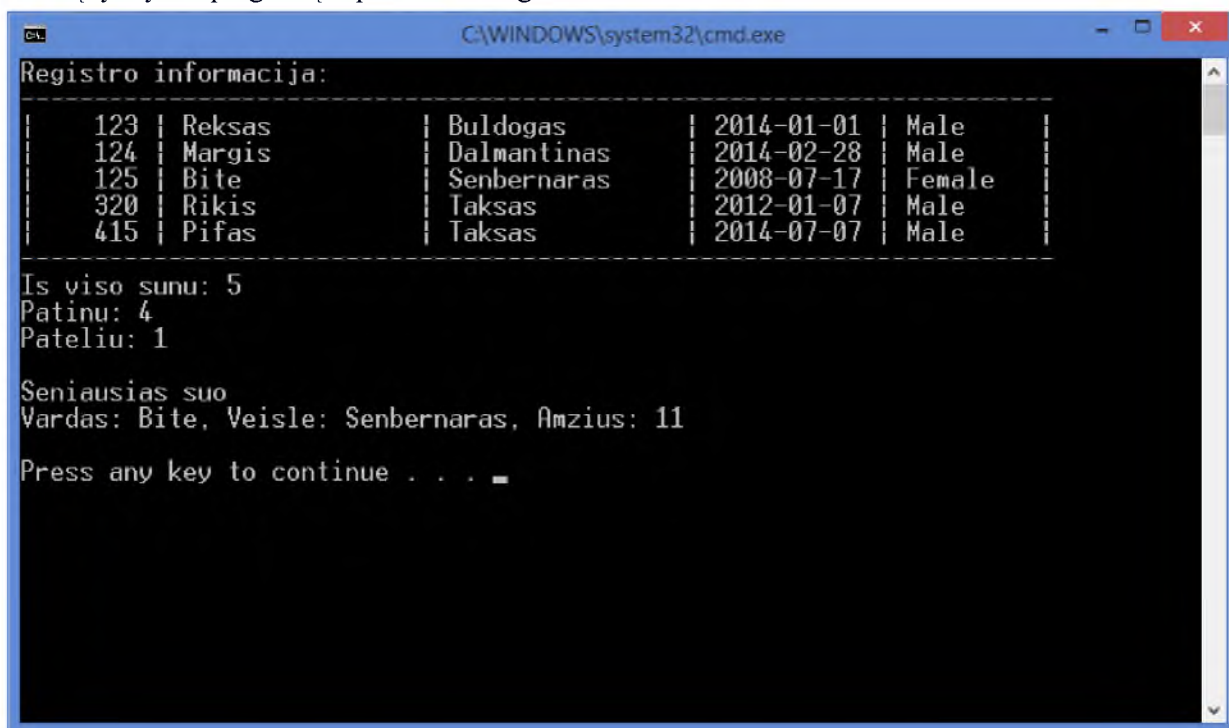
- Užduotis reikalauja atspausdinti seniausio šuns amžių, tačiau dabar turime tik jo gimimo datą. Papildykite klasę *Dog* metodu *CalculateAge*.

```
public int CalculateAge()
{
    DateTime today = DateTime.Today;
    int age = today.Year - this.BirthDate.Year;
    if (this.BirthDate.Date > today.AddYears(-age))
    {
        age--;
    }
    return age;
}
```

- Papildykite *Main* metodą kreipiniu į *FindOldestDog* metodą ir gautų rezultatų spausdinimu:

```
Dog oldest = TaskUtils.FindOldestDog(allDogs);
Console.WriteLine("Seniausias šuo");
Console.WriteLine("Vardas: {0}, Veislė: {1}, Amžius: {2}",
    oldest.Name, oldest.Breed, oldest.CalculateAge());
```

- Ivykdykite programą ir patikrinkite gautus rezultatus.



```
C:\WINDOWS\system32\cmd.exe

Registro informacija:
-----
| 123 | Reksas      | Buldogas      | 2014-01-01 | Male |
| 124 | Margis      | Dalmantinas   | 2014-02-28 | Male |
| 125 | Bite        | Senbernaras   | 2008-07-17 | Female |
| 320 | Rikis       | Taksas        | 2012-01-07 | Male |
| 415 | Pifas       | Taksas        | 2014-07-07 | Male |
-----

Is viso sunu: 5
Patinu: 4
Pateliu: 1

Seniausias suo
Vardas: Bite, Veisle: Senbernaras, Amzius: 11

Press any key to continue . . .
```

Penktas žingsnis. Unikalių reikšmių radimas.

- Parašykite metodą, kuris suranda visas skirtingas veisles. Talpinkite į klasę *TaskUtils*.

```
public static List<string> FindBreeds(List<Dog> Dogs)
{
    List<string> Breeds = new List<string>();
    foreach (Dog dog in Dogs)
    {
        string breed = dog.Breed;
        if (!Breeds.Contains(breed)) // uses List method Contains()
        {
            Breeds.Add(breed);
        }
    }
    return Breeds;
}
```

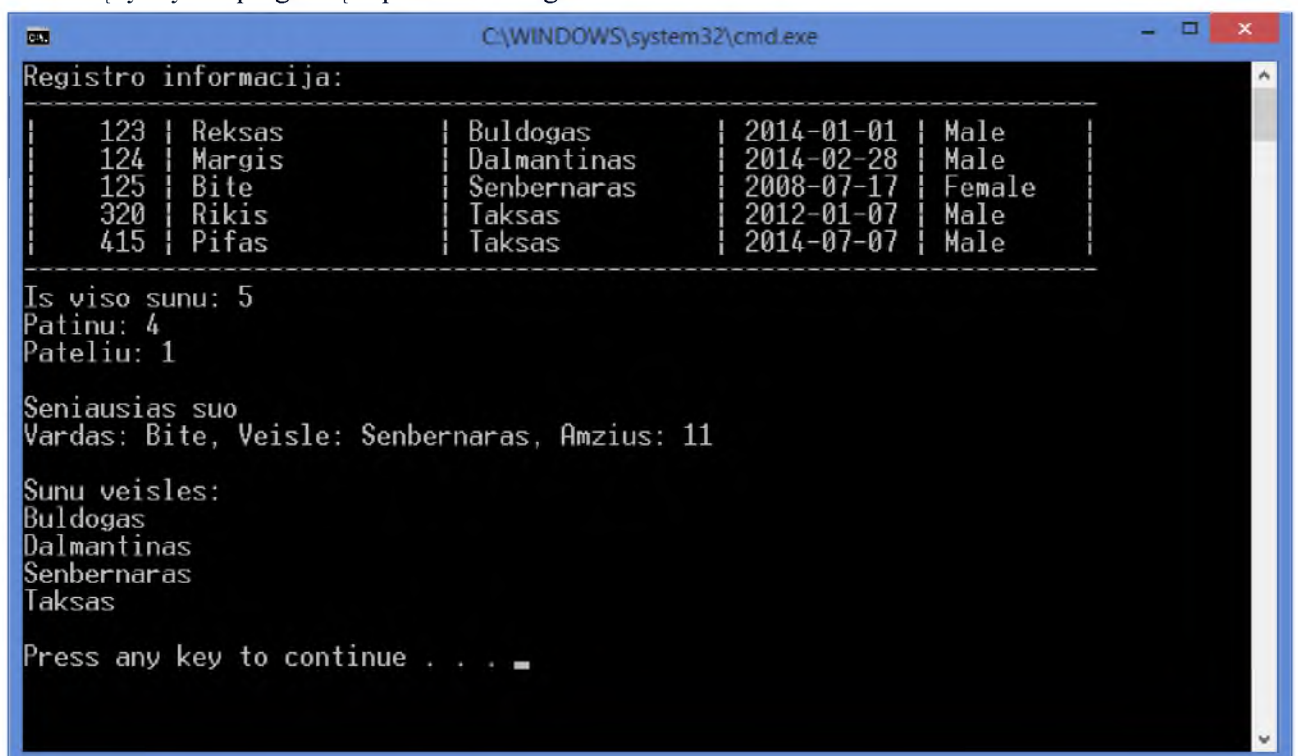
- Papildykite klasę *InOutUtils* metodu, skirtu veislių sąrašui spausdinti.

```
public static void PrintBreeds(List<string> breeds)
{
    foreach (string breed in breeds)
    {
        Console.WriteLine(breed);
    }
}
```

- Papildykite metodą *Main* naujų metodų iškvietimui:

```
List<string> Breeds = TaskUtils.FindBreeds(allDogs);
Console.WriteLine("Šunų veislės:");
InOutUtils.PrintBreeds(Breeds);
Console.WriteLine();
```

- Ivykdykite programą ir pasitikrinkite gautus rezultatus.



```
C:\WINDOWS\system32\cmd.exe

Registro informacija:
-----
| 123 | Reksas      | Buldogas      | 2014-01-01 | Male |
| 124 | Margis      | Dalmantinas   | 2014-02-28 | Male |
| 125 | Bite        | Senbernaras   | 2008-07-17 | Female |
| 320 | Rikis       | Taksas        | 2012-01-07 | Male |
| 415 | Pifas       | Taksas        | 2014-07-07 | Male |
-----

Is viso sunu: 5
Patinu: 4
Pateliu: 1

Seniausias suo
Vardas: Bite, Veisle: Senbernaras, Amzius: 11

Sunu veisles:
Buldogas
Dalmantinas
Senbernaras
Taksas

Press any key to continue . . .
```


Šeštas žingsnis. Filtravimas.

- Papildykite *Main* metodą vartotojo pasirinktos šunų veislės perskaitymui iš konsolės.

```
Console.WriteLine("Kokios veislės šunis atrinkti?");  
string selectedBreed = Console.ReadLine();
```

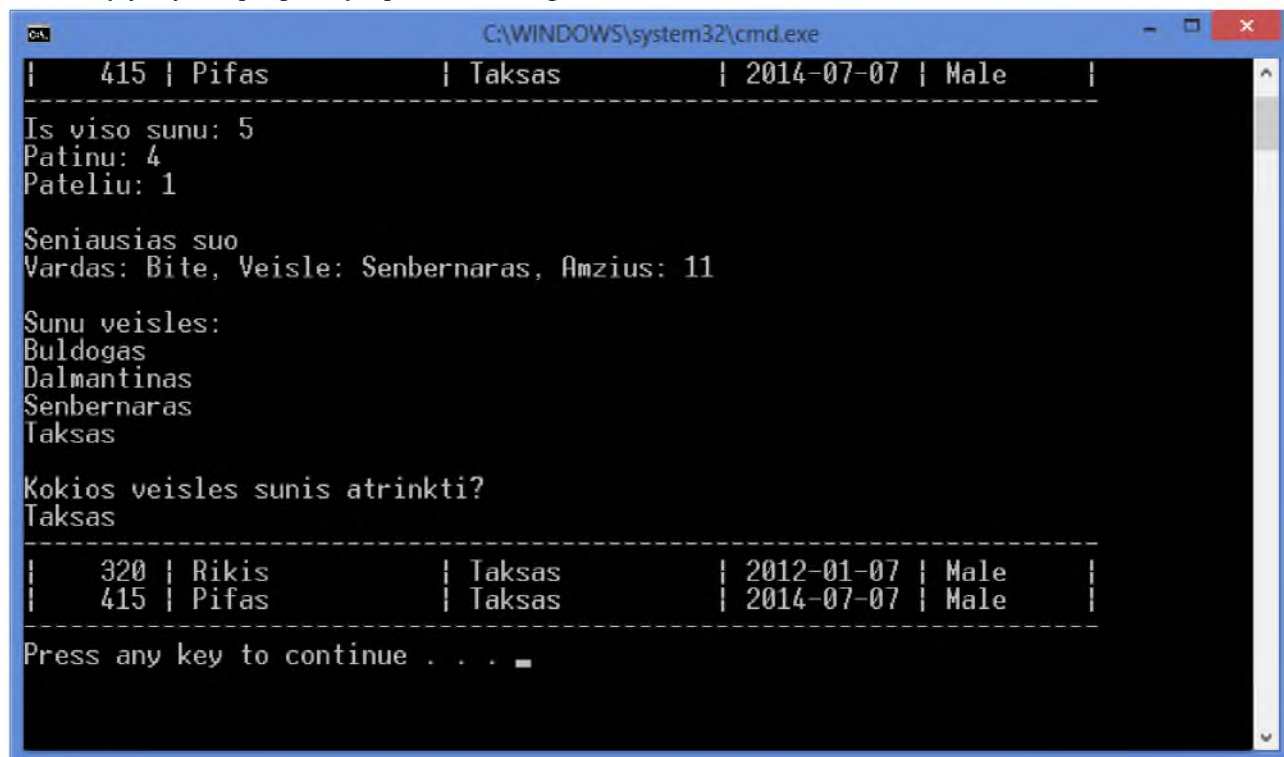
- Parašykite metodą, atrenkantį nurodytos veislės šunis. Talpinkite į klasę *TaskUtils*.

```
public static List<Dog> FilterByBreed(List<Dog> Dogs, string breed)  
{  
    List<Dog> Filtered = new List<Dog>();  
    foreach (Dog dog in Dogs)  
    {  
        if (dog.Breed.Equals(breed)) // uses string method Equals()  
        {  
            Filtered.Add(dog);  
        }  
    }  
    return Filtered;  
}
```

- Papildykite *Main* metodą:

```
List<Dog> FilteredByBreed = TaskUtils.FilterByBreed(allDogs, selectedBreed);  
InOutUtils.PrintDogs(FilteredByBreed);
```

- Įvykdykite programą ir patikrinkite gautus rezultatus.



```
C:\WINDOWS\system32\cmd.exe  
| 415 | Pifas | Taksas | 2014-07-07 | Male |  
-----  
Is viso sunu: 5  
Patinu: 4  
Pateliu: 1  
  
Seniausias suo  
Vardas: Bite, Veisle: Senbernaras, Amzius: 11  
  
Sunu veisles:  
Buldogas  
Dalmantinas  
Senbernaras  
Taksas  
  
Kokios veisles sunis atrinkti?  
Taksas  
-----  
| 320 | Rikis | Taksas | 2012-01-07 | Male |  
| 415 | Pifas | Taksas | 2014-07-07 | Male |  
-----  
Press any key to continue . . .
```

Septintas žingsnis. Spausdinimas į failą.

- Parašykite metodą, skirtą šunų spausdinimui į CSV failą.

```
public static void PrintDogsToCSVFile(string fileName, List<Dog> Dogs)
{
    string[] lines = new string[Dogs.Count + 1];
    lines[0] = String.Format("{0};{1};{2};{3};{4}",
        "Reg.Nr.", "Vardas", "Veislė", "Gimimo data", "Lytis");
    for (int i = 0; i < Dogs.Count; i++)
    {
        lines[i + 1] = String.Format("{0};{1};{2};{3};{4}",
            Dogs[i].ID, Dogs[i].Name, Dogs[i].Breed, Dogs[i].BirthDate, Dogs[i].Gender);
    }
    File.WriteAllLines(fileName, lines, Encoding.UTF8);
}
```

- Papildykite *Main* metodą:

```
string fileName = selectedBreed + ".csv";
InOutUtils.PrintDogsToCSVFile(fileName, filteredByBreed);
```

- Ivykdykite programą. Patikrinkite, ar buvo sukurtas naujas CSV failas. Atidarykite jį su Excel, ir patikrinkite rezultatus.

	A	B	C	D	E	F
1	320	Rikis	Taksas	2012-01-07	Male	
2	415	Pifas	Taksas	2014-07-07	Male	
3						
4						
5						

1 savarankiško darbo užduotis

Žinoma, kiek kuris turistinės grupės narys (vardas, pavardė) turi pinigų. Duomenys tekstiniame faile. Kiekvienas narys bendroms grupės išlaidoms skiria ketvirtadalį turimų pinigų. Kiek pinigų bus iš viso surinkta bendroms grupės išlaidoms? Kuris grupės narys skyrė daugiausiai pinigų grupės išlaidoms? Jei yra keli nariai, išvardinkite visus.