

5. Polimorfizmas

Siekiami studijų rezultatai:

- pagrindiniai objektinio programavimo principai;
- objektinio programavimo principų taikymas programų kūrimui;
- uždavinio sprendimo algoritmo sudarymas ar žinomo algoritmo pritaikymas;
- sudaryto algoritmo realizavimas C# programavimo kalba;
- programų derinimas ir testavimas.

Susipažinsite su:

- Objektinio programavimo konstrukcijomis, užtikrinančiomis polimorfizmą;

5.1 Bazinės ir išvestinės klasės, naudojant fiksuoto dydžio masyvus

Užduotis.

- Užduotis ta pati, kaip ir trečiojo laboratorinio darbo pratybose.

Pradiniai duomenys ir rezultatai.

- Duomenų failai ir rezultatų formatai skirtingi. Duomenų failus talpiname į atskirą aplanką **Data**.

Pradiniai duomenys
<p>Failas L3Data-Kaunas.csv:</p> <p>Kaunas</p> <p>D,Reksas,823,Buldogas,Jonas,867424992,2015-07-24,TRUE</p> <p>C,Morkus,658,Siamo,Antanas,865412654,2014-11-24</p> <p>D,Keksas,456,Taksas,Petras,866612345,2014-12-01,FALSE</p> <p>C,Pukė,656,Siamo,Petras,866612345,2014-12-05</p> <p>C,Taira,382,Rusų mėlynoji,Dangira,861253226,2012-05-11</p> <p>D,Miledi,541,Jorkšyro terjeras,Aurelija,867897315,2010-11-14,FALSE</p>
<p>Failas L3Data-Kaunas2.csv:</p> <p>Kaunas</p> <p>D,Dagas,384,Rotveileris,Gytis,867424242,2014-04-24,TRUE</p> <p>D,Princė,842,Jorkšyro terjeras,Augustė,861812330,2014-03-01,FALSE</p> <p>C,Murksė,325,Birmos,Marytė,861855881,2010-08-05</p> <p>C,Rainys,892,Amerikos riestausės,Aldona,861492532,2011-02-26</p> <p>D,Muchtaras,126,Vokiečių aviganis,Ramūnas,861259722,2012-04-15,FALSE</p> <p>D,Haris,458,Jorkšyro terjeras,Sigitas,861933521,2015-04-25,FALSE</p> <p>C,Latė,435,Siamo,Ramunė,867855664,2013-12-04</p>
<p>Failas L3Data-Šiauliai.csv:</p> <p>Šiauliai</p> <p>C,Spurgis,725,Norvegų miškinės,Jurgis,869712120,2012-08-17</p> <p>C,Melisa,829,Persų,Amelija,861489795,2009-10-23</p> <p>D,Dingas,648,Labradoras,Aistė,868922564,2008-05-24,FALSE</p> <p>C,Micis,869,Meino meškėnas,Sandra,868723234,2013-12-20</p> <p>C,Pupsė,892,Persų,Danutė,867289254,2009-09-03</p> <p>D,Princė,549,Pudelis,Zita,861354852,2010-05-23,FALSE</p> <p>D,Rokis,384,Buldogas,Algirdas,867910842,2013-08-30,TRUE</p>
<p>Failas L3Data-Vilnius.csv:</p> <p>Vilnius</p> <p>D,Ama,487,Jorkšyro terjeras,Inga,861299001,2012-08-23,FALSE</p> <p>D,Flintas,403,Vokiečių aviganis,Tadas,879532146,2008-05-25,TRUE</p> <p>C,Liza,292,Sibiro,Alina,867966452,2009-09-13</p> <p>C,Agatas,372,Britų trumpaplaukės,Juozas,867922564,2011-07-12</p> <p>C,Aironas,387,Britų trumpaplaukės,Anupras,867922577,2012-12-12</p> <p>D,Kira,438,Airių terjeras,Vilija,861297520,2011-08-30,FALSE</p> <p>D,Maksis,448,Vokiečių bokseris,Tautvydas,879833551,2010-10-18,TRUE</p>

D,Tobis,687,Šarpėjus,Mantas,861497852,2014-06-05,TRUE						
Rezultatai						
Užregistruoti šunys:						
Kaunas						
823	Buldogas	Reksas	Jonas	(867424992)	2015-07-24	+
456	Taksas	Keksas	Petras	(866612345)	2014-12-01	
541	Jorkšyro terjeras	Miledi	Aurelija	(867897315)	2010-11-14	
384	Rotveileris	Dagas	Gytis	(867424242)	2014-04-24	+
842	Jorkšyro terjeras	Princė	Augustė	(861812330)	2014-03-01	
126	Vokiečių aviganis	Muchtaras	Ramūnas	(861259722)	2012-04-15	
458	Jorkšyro terjeras	Haris	Sigitas	(861933521)	2015-04-25	
Užregistruotos katės:						
Kaunas						
658	Siamo	Morkus	Antanas	(865412654)	2014-11-24	
656	Siamo	Pukė	Petras	(866612345)	2014-12-05	
382	Rusų mėlynoji	Taira	Dangira	(861253226)	2012-05-11	
325	Birmos	Murksė	Marytė	(861855881)	2010-08-05	
892	Amerikos riestausės	Rainys	Aldona	(861492532)	2011-02-26	
435	Siamo	Latė	Ramunė	(867855664)	2013-12-04	
Agresyvūs šunys						
Kaunas: 2						
Agresyvūs šunys						
Vilnius: 3						
Populiariausia šunų veislė						
Kaunas: Jorkšyro terjeras						
Populiariausia kačių veislė						
Vilnius: Britų trumpaplaukės						
Pagal lusto Nr. surūšiuotas visų filialų šunų sąrašas:						
Visi šunys						
126	Vokiečių aviganis	Muchtaras	Ramūnas	(861259722)	2012-04-15	
384	Buldogas	Rokis	Algirdas	(867910842)	2013-08-30	+
384	Rotveileris	Dagas	Gytis	(867424242)	2014-04-24	+
403	Vokiečių aviganis	Flintas	Tadas	(879532146)	2008-05-25	+
438	Airių terjeras	Kira	Vilija	(861297520)	2011-08-30	
448	Vokiečių bokseris	Maksis	Tautvydas	(879833551)	2010-10-18	+
456	Taksas	Keksas	Petras	(866612345)	2014-12-01	
458	Jorkšyro terjeras	Haris	Sigitas	(861933521)	2015-04-25	
487	Jorkšyro terjeras	Ama	Inga	(861299001)	2012-08-23	
541	Jorkšyro terjeras	Miledi	Aurelija	(867897315)	2010-11-14	
549	Pudelis	Princė	Zita	(861354852)	2010-05-23	
648	Labradoras	Dingas	Aistė	(868922564)	2008-05-24	
687	Šarpėjus	Tobis	Mantas	(861497852)	2014-06-05	+
823	Buldogas	Reksas	Jonas	(867424992)	2015-07-24	+
842	Jorkšyro terjeras	Princė	Augustė	(861812330)	2014-03-01	

Programos kūrimo eiga:

- Sukuriamas naujas sprendimas (*solution*) pavadinimu **Lab5**.
- Sukuriamas naujas projektas **Lab5.Step1**, kuris bus skirtas praktikavimuisi naudojant masyvus.
- Sukuriamas aplankas **Data** ir į jį patalpinami duomenų failai, kurie bus projekto dalis.
- Kopijuojami visi klasių failai.

➤ Pirmas žingsnis.

- Bazinė klasė **Animal** yra abstrakti, nes turi abstraktų metodą **isVaccinationExpired()**, nes kačių vakcinavimas įvertinamas mėnesiais (šis metodas skiriasi katėms ir šunims, kuriems vakcinavimas įvertinamas metais). Išvestinės šunų ir kačių klasės privalės realizuoti šį metodą.

- Klasėje **Animal** talpiname metodą **SetData**, skirtą visų gyvūnų bendrųjų duomenų įrašymui ir sukuriame naują šios klasės konstruktorių, kuris kviečia šį metodą. Tuomet klasė **Animal** atrodo taip:

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

abstract class Animal
{
    public string Name { get; set; }
    public int ChipId { get; set; }
    public string Breed { get; set; }
    public string Owner { get; set; }
    public string Phone { get; set; }
    public DateTime VaccinationDate { get; set; }

    public Animal(string name, int chipId, string breed, string owner, string phone,
DateTime vaccinationDate)
    {
        Name = name;
        ChipId = chipId;
        Breed = breed;
        Owner = owner;
        Phone = phone;
        VaccinationDate = vaccinationDate;
    }
    public Animal(string data)
    {
        SetData(data);
    }

    public virtual void SetData(string line)
    {
        string[] values = line.Split(',');
        Name = values[1];
        ChipId = int.Parse(values[2]);
        Breed = values[3];
        Owner = values[4];
        Phone = values[5];
        VaccinationDate = DateTime.Parse(values[6]);
    }

    /// <summary>
    /// Metodas paskelbtas abstrakčiu. Tai reiškia, jog vaikų klasės privalės jį realizuoti
    /// <summary>
    abstract public bool isVaccinationExpired();

    public override bool Equals(object obj)
    {
        return this.Equals(obj as Animal);
    }

    public bool Equals(Animal animal)
    {
        if (Object.ReferenceEquals(animal, null))
        {
            return false;
        }

        if (this.GetType() != animal.GetType())
        {
            return false;
        }
    }
}
```

```

        return (ChipId == animal.ChipId) && (Name == animal.Name);
    }

    public override int GetHashCode()
    {
        return ChipId.GetHashCode() ^ Name.GetHashCode();
    }

    public static bool operator ==(Animal lhs, Animal rhs)
    {
        if (Object.ReferenceEquals(lhs, null))
        {
            if (Object.ReferenceEquals(rhs, null))
            {
                return true;
            }

            return false;
        }
        return lhs.Equals(rhs);
    }

    public static bool operator !=(Animal lhs, Animal rhs)
    {
        return !(lhs == rhs);
    }

    public static bool operator <=(Animal lhs, Animal rhs)
    {
        return (lhs.ChipId <= rhs.ChipId);
    }

    public static bool operator >=(Animal lhs, Animal rhs)
    {
        return (lhs.ChipId >= rhs.ChipId);
    }
}

```

🔗 Antras žingsnis.

- Klasei **Cat** sukuriame papildomą konstruktorių, kviečiantį naująjį **Animal** klasės konstruktorių.
- Realizuojame bazinės klasės abstraktų metodą **isVaccinationExpired()**.
- Pakeičiame **ToString** metodo realizaciją.
- Užklojimo **==** atsisakome, nes jis nieko naujo klasei nesuteikia. Šį užklojimą matysime **Dog** klasėje.
- Tuomet klasė **Cat** atrodo taip:

```

class Cat : Animal
{
    private static int VaccinationDurationMonths = 6;

    public Cat(string name, int chipId, string breed, string owner, string phone, DateTime
vaccinationDate)
        : base(name, chipId, breed, owner, phone, vaccinationDate)
    {
    }

    public Cat(string data)
        : base(data)
    {
        SetData(data);
    }

    /// <summary>
    /// Abstraktaus Animal klasės metodo realizacija
    /// </summary>

```

```

    /// <returns></returns>
    public override bool isVaccinationExpired()
    {
        return VaccinationDate.AddMonths(VaccinationDurationMonths).CompareTo(DateTime.Now)
> 0;
    }

    public override String ToString()
    {
        return String.Format("|{0,-3}|{1,-20}|{2,-9}|{3,-10} ({4})|{5:yyyy-MM-dd}|", ChipId,
Breed, Name, Owner, Phone, VaccinationDate);
    }
}

```

☛ Trečias žingsnis.

- Klasei **Dog** sukuriame papildomą konstruktorių, kviečiantį naująjį **Animal** klasės konstruktorių.
- Realizuojame bazinės klasės abstraktų metodą **isVaccinationExpired()**.
- Pakeičiame **ToString** metodo realizaciją.
- Taip pat užklojame metodą **SetData**, nes yra papildomas laukas šuns aprašyme.
- Tuomet klasė **Dog** atrodo taip:

```

class Dog : Animal
{
    private static int VaccinationDuration = 1;

    public Dog(string name, int chipId, string breed, string owner, string phone, DateTime
vaccinationDate, bool aggressive)
        : base(name, chipId, breed, owner, phone, vaccinationDate)
    {
        Aggressive = aggressive;
    }

    public Dog(string data)
        : base(data)
    {
        SetData(data);
    }

    public override void SetData(string line)
    {
        base.SetData(line);
        string[] values = line.Split(',');
        Aggressive = bool.Parse(values[7]);
    }

    public bool Aggressive { get; set; }

    /// <summary>
    /// Abstraktaus Animal klasės metodo realizacija
    /// </summary>
    /// <returns></returns>
    public override bool isVaccinationExpired()
    {
        return VaccinationDate.AddYears(VaccinationDuration).CompareTo(DateTime.Now) > 0;
    }

    public override String ToString()
    {
        return String.Format("|{0,-3}|{1,-20}|{2,-9}|{3,-10} ({4})|{5:yyyy-MM-dd}|{6}|",
ChipId, Breed, Name, Owner, Phone, VaccinationDate, Aggressive ? '+' : ' ');
    }

    public override bool Equals(object obj)
    {
        return this.Equals(obj as Dog); //kviečiame tipui specifinį metodą toje pačioje
klasėje
    }
}

```

```

    }

    public bool Equals(Dog dog)
    {
        return base.Equals(dog); //kviečiame tėvinės klasės Animal Equals metodą
        //galima papildomai tikrinti pagal tik Dog klasės būdingas savybes, pvz
        //return base.Equals(dog) && this.Aggressive == dog.Aggressive;
    }

    public override int GetHashCode()
    {
        return ChipId.GetHashCode() ^ Name.GetHashCode();
    }

    public static bool operator ==(Dog lhs, Dog rhs)
    {
        if (Object.ReferenceEquals(lhs, null))
        {
            if (Object.ReferenceEquals(rhs, null))
            {
                return true;
            }

            return false;
        }
        return lhs.Equals(rhs);
    }

    public static bool operator !=(Dog lhs, Dog rhs)
    {
        return !(lhs == rhs);
    }
}

```

🔗 Ketvirtas žingsnis.

- Pertvarkome **Branch** klasę. Konteinerinėje klasėje turi būti tik vienas masyvas **Animal**.
- Vietoje buvusių metodų **AddDog** ir **AddCat** sukuriame vieną metodą **AddAnimal**.
- Į klasę talpiname rikiavimo metodą.
- Užklojame operaciją „+“ dviejų **Branch** objektų sujungimui.
- Tuomet klasė **Branch** atrodys taip:

```

class Branch
{
    public string Town { get; private set; }
    private Animal[] Animals;
    public int Count { get; private set; }

    public Branch(string town = "")
    {
        Town = town;
        Animals = new Animal[Program.MaxNumberOfAnimals];
    }

    /// <summary>
    /// Prideda gyvūną į rinkinį
    /// </summary>
    /// <param name="a">Priedamas gyvūnas</param>
    public void AddAnimal(Animal a)
    {
        Animals[Count] = a;
        Count++;
    }
}

```

```

/// <summary>
/// Metodas paima gyvūną pagal nurodytą indeksą
/// </summary>
public Animal GetAnimal(int index)
{
    return Animals[index];
}

/// <summary>
/// Metodas rikiuoja gyvūnų sąrašą
/// </summary>
public void SortAnimals()
{
    for (int i = 0; i < Count - 1; i++)
    {
        int m = i;
        for (int j = i + 1; j < Count; j++)
            if (Animals[j] <= Animals[m])
                m = j;
        Animal a = Animals[i];
        Animals[i] = Animals[m];
        Animals[m] = a;
    }
}

/// <summary>
/// Sujungia du gyvūnų rinkinius į trečią rinkinį
/// </summary>
/// <param name="a">Pirmasis rinkinys</param>
/// <param name="b">Antrasis rinkinys</param>
/// <returns></returns>
public static Branch operator +(Branch a, Branch b)
{
    Branch c = new Branch(a.Town);
    for (int i = 0; i < a.Count; i++)
        c.AddAnimal(a.Animals[i]);
    for (int i = 0; i < b.Count; i++)
        c.AddAnimal(b.Animals[i]);
    return c;
}
}

```

Penktas žingsnis.

- Pertvarkome pagrindinę programą.
- Pirmiausiai pakeičiame duomenų skaitymo metodą. Skaitymą atliks du metodai: 1) ciklas per nurodyto aplanko failus; 2) nurodyto failo skaitymas ir duomenų paskirstymas, panaudojant polimorfizmą.

```

/// <summary>
/// Perskaito filialo duomenis iš failo
/// </summary>
/// <param name="file">Aplanko vardas</param>
/// <param name="branches">Filialai, į kuriuos sudedami duomenys</param>
/// <param name="number">Filialų skaičius</param>
private static void ReadData(string file, Branch[] branches, ref int number)
{
    string[] filePaths = Directory.GetFiles(file, "*.csv");
    foreach (string path in filePaths)
    {
        ReadAnimalData(path, branches, ref number);
    }
}

/// <summary>
/// Perskaito filialo duomenis iš failo
/// </summary>
/// <param name="file">Failo vardas</param>

```

```

/// <param name="branches">Filialas į kurį sudedami duomenys</param>
/// <param name="number">Filialų skaičius</param>
private static void ReadAnimalData(string file, Branch[] branches, ref int number)
{
    using (StreamReader reader = new StreamReader(@file, Encoding.GetEncoding(1257)))
    {
        string line = reader.ReadLine();
        Branch branch = GetBranchByTown(branches, ref number, line);
        while (null != (line = reader.ReadLine()))
        {
            switch (line[0])
            {
                case 'D':
                    branch.AddAnimal(new Dog(line));
                    break;
                case 'C':
                    branch.AddAnimal(new Cat(line));
                    break;
            }
        }
    }
}

```

- Pakeičiame duomenų išvedimo į konsolę metodą **PrintAnimalsToConsole**. Parametras nurodys, kokį gyvūnų tipą norime spausdinti.

```

/// <summary>
/// Išveda filialo duomenis į ekraną
/// </summary>
/// <param name="ba">Filialas</param>
/// <param name="title">Lentelės pavadinimas</param>
/// <param name="type">Spausdinamo gyvūno tipas</param>
static void PrintAnimalsToConsole(Branch ba, string title, char type)
{
    string s = new string('-', ba.GetAnimal(0).ToString().Length);
    Console.WriteLine(title);
    Console.WriteLine(s);
    for (int i = 0; i < ba.Count; i++)
    {
        switch (type)
        {
            case 'D':
            case 'd':
                if (ba.GetAnimal(i) is Dog)
                    Console.WriteLine(ba.GetAnimal(i));
                break;
            case 'C':
            case 'c':
                if (ba.GetAnimal(i) is Cat)
                    Console.WriteLine(ba.GetAnimal(i));
                break;
            default:
                Console.WriteLine(ba.GetAnimal(i));
                break;
        }
    }
    Console.WriteLine(s);
}

```

- Pertvarkome pagrindinės programą **Main** metodą. Jis bus „švarus“: jokių ciklų, jokių išankstinių žinių apie filialų pavadinimus ir jų kiekį:

```

class Program
{
    public const int MaxNumberOfBranches = 10;
    public const int MaxNumberOfAnimals = 50;
    public const int MaxNumberOfBreeds = 50;

    static void Main(string[] args)

```



```

{
    Branch[] branches = new Branch[MaxNumberOfBranches];
    int NumberOfBranches = 0;
    const string DataDir = @"..\..\Data";
    ReadData(DataDir, branches, ref NumberOfBranches);

    Console.WriteLine("Užregistruoti šunys:");
    PrintAnimalsToConsole(branches[0], branches[0].Town, 'D');
    Console.WriteLine();

    Console.WriteLine("Užregistruotos katės:");
    PrintAnimalsToConsole(branches[0], branches[0].Town, 'c');
    Console.WriteLine();

    Console.WriteLine("Agresyvūs šunys\n {0}: {1}", branches[0].Town,
CountAggressive(branches[0]));
    Console.WriteLine("Agresyvūs šunys\n {0}: {1}", branches[1].Town,
CountAggressive(branches[1]));

    Console.WriteLine("Populiariausia šunų veislė\n {0}: {1}", branches[0].Town,
GetMostPopularBreed(GetAnimals("Filialas: {0} Gyvūnas: šuo", branches[0], 'D')));
    Console.WriteLine("Populiariausia kačių veislė\n {0}: {1}", branches[1].Town,
GetMostPopularBreed(GetAnimals("Filialas: {0} Gyvūnas: katė", branches[1], 'C')));
    Console.WriteLine();

    Console.WriteLine("Pagal lusto Nr. surūšiuotas visų filialų šunų sąrašas:");
    Console.WriteLine();
    Branch allDogs = new Branch("Visi šunys");
    GetAllDogs(branches, NumberOfBranches, ref allDogs, "Filialas: {0} Gyvūnas: šuo");
    allDogs.SortAnimals();
    PrintAnimalsToConsole(allDogs, allDogs.Town, '-');
}

```

Ušėštas žingsnis.

- Pertvarkome visus kitus **Main** programos metodus:

```

/// <summary>
/// Suranda filialą pagal pavadinimą
/// </summary>
/// <param name="branches">Filialų rinkinys</param>
/// <param name="number">Filialų kiekis</param>
/// <param name="town">Miesto pavadinimas</param>
/// <returns>Surastas arba naujai sukurtas Branch objektas</returns>
private static Branch GetBranchByTown(Branch[] branches, ref int number, string town)
{
    for (int i = 0; i < number; i++)
    {
        if (branches[i].Town == town)
        {
            return branches[i];
        }
    }
    branches[number++] = new Branch(town);
    return branches[number - 1];
}

/// <summary>
/// Iš gyvūnų sąrašo išrenkami tik šunys
/// </summary>
/// <param name="forma">Kuriamo rinkinio įvardinimas</param>
/// <param name="ba">Filialas</param>
/// <param name="type">Gyvūnų tipas</param>
/// <returns>Branch objektas, kuriame yra tik nurodyto tipo gyvūnai </returns>
private static Branch GetAnimals(string forma, Branch ba, char type)
{
    Branch dogs = new Branch(String.Format(forma, ba.Town));
    for (int i = 0; i < ba.Count; i++)

```

```

        switch (type)
        {
            case 'D':
            case 'd':
                if (ba.GetAnimal(i) is Dog)
                    dogs.AddAnimal(ba.GetAnimal(i));
                break;
            case 'C':
            case 'c':
                if (ba.GetAnimal(i) is Cat)
                    dogs.AddAnimal(ba.GetAnimal(i));
                break;
        }
        return dogs;
    }
}

/// <summary>
/// Suformuojamas filiale užregistruotų veislių rinkinys
/// </summary>
/// <param name="ba">Filialas</param>
/// <param name="breeds">Skirtingų veislių rinkinys</param>
/// <param name="ba">Skirtingų veislių kiekis</param>
private static void GetBreeds(Branch ba, out string[] breeds, out int breedCount)
{
    breeds = new string[MaxNumberOfBreeds];
    breedCount = 0;
    for (int i = 0; i < ba.Count; i++)
    {
        if (!breeds.Contains(ba.GetAnimal(i).Breed))
        {
            breeds[breedCount++] = ba.GetAnimal(i).Breed;
        }
    }
}

/// <summary>
/// Išrenkami tik nurodytos veislės gyvūnai
/// </summary>
/// <param name="ba">Filialas</param>
/// <param name="breed">Veislės pavadinimas</param>
/// <returns>Nurodytos veislės gyvūnų rinkinys</returns>
private static Branch FilterByBreed(Branch ba, string breed)
{
    Branch filtered = new Branch(breed);
    for (int i = 0; i < ba.Count; i++)
        if (ba.GetAnimal(i).Breed == breed)
            filtered.AddAnimal(ba.GetAnimal(i));
    return filtered;
}

/// <summary>
/// Suskaičiuoja agresyvių šunų kiekį filiale
/// </summary>
/// <param name="ba">Filialas</param>
/// <returns>Agresyvių šunų kiekis</returns>
private static int CountAggressive(Branch ba)
{
    int counter = 0;
    for (int i = 0; i < ba.Count; i++)
        if ((ba.GetAnimal(i) is Dog) && (ba.GetAnimal(i) as Dog).Aggressive)
            counter++;
    return counter;
}

/// <summary>
/// Suranda populiariausios filiale veislės pavadinimą
/// </summary>
/// <param name="ba">Filialas</param>

```

```

/// <returns>Veislės pavadinimas</returns>
private static string GetMostPopularBreed(Branch ba)
{
    String popular = "not found";
    int count = 0;

    int breedCount = 0;
    string[] breeds;

    GetBreeds(ba, out breeds, out breedCount);

    for (int i = 0; i < breedCount; i++)
    {
        Branch filtered = FilterByBreed(ba, breeds[i]);
        if (filtered.Count > count)
        {
            popular = breeds[i];
            count = filtered.Count;
        }
    }
    return popular;
}
/// <summary>
/// Iš visų filialų gyvūnų sąrašo išrenkami tik šunys
/// </summary>
/// <param name="ba">Filialai</param>
/// <param name="NumberOfBranches">Filialų kiekis</param>
/// <param name="allDogs">Atrinktas šunų sąrašas</param>
/// <param name="forma">Įvardijimas</param>
private static void GetAllDogs(Branch[] ba, int NumberOfBranches, ref Branch allDogs,
string forma)
{
    for (int i = 0; i < NumberOfBranches; i++)
    {
        Branch bDogs = GetAnimals(forma, ba[i], 'D');
        allDogs += bDogs;
    }
}
}

```

- Įvykdysite programą ir patikrinkite rezultatus.

1 savarankiško darbo užduotis.

- Įmonė pradėjo registruoti ir jūrų kiaulytes. Jūrų kiaulytės, kaip ir katės bei šunys, yra vakcinuojamos, tačiau jos nėra žymimos (nesuteikiamos mikroschemos). Sukurkite klasės **Animal** išvestinę klasę **AnimalMarked**. Perkelkite į ją savybę **chipId**. Pakeiskite klasių **Dog** ir **Cat** bazinės klasės **AnimalMarked**. Sukurkite bazinės klasės **Animal** išvestinę klasę **GuineaPig** (taigi, **Animal** klasę paveldės klasės **AnimalMarked** ir **GuineaPig**, **AnimalMarked** klasę paveldi **Cat** ir **Dog**). Atkreipkite dėmesį, jog po šio pertvarkymo teks pakeisti ir kai kuriuos pagrindinės klasės metodus. Papildykite duomenų failus jūrų kiaulyčių informacija. Atspausdinkite surikiuotus atskirai gyvūnų sąrašus. Šunys ir katės rikiuojami pagal chipID, o jūrų kiaulytės – pagal vardus.

2 savarankiško darbo užduotis.

- Sportas. Duota informacija apie krepšininkus: komandos pavadinimas, pavardė, vardas, žaistų rungtynių skaičius, įmestų taškų skaičius, atkovotų kamuolių skaičius, rezultatyvių perdavimų skaičius. Duota informacija apie futbolininkus: komandos pavadinimas, pavardė, vardas, žaistų rungtynių skaičius, įmuštų įvarčių skaičius, surinktų geltonų kortelių skaičius. Duota informacija apie komandas: komandos pavadinimas, miestas, komandos treneris, žaistų rungtynių skaičius. Atrinkti žaidėjus, kurie žaidė visose komandos rungtynėse ir įmetė taškų (įmušė įvarčių) ne mažiau kaip vidurkis. Krepšininkams papildomas rodiklis – atliko rezultatyvių perdavimų ne mažiau kaip vidurkis, futbolininkams papildomas rodiklis – surinko geltonų kortelių ne daugiau už vidurkį.

5.2 Užduotys

U5_1. Krepšinio rinktinė. Turite ne tik šių, bet ir dviejų ankstesniųjų metų į stovyklas pakviestų krepšininkų sąrašus. Pirmoje eilutėje yra metai, antrėje – stovyklos pradžios data, trečioje – stovyklos pabaigos data. Toliau informacija apie rinktinės narius. Krepšinio rinktinę sudaro ne tik krepšininkai, bet ir pagalbinis personalas – treneriai, gydytojai, masažuotojai ir kt. Sukurkite abstrakčiąją klasę „Narys“ (laukai – vardas, pavardė, gimimo data), kurią paveldės klasės „Krepšininkas“ (papildomi laukai – ūgis, pozicija, klubas, požymis „pakviestas“, požymis „kapitonas“) ir „Personalas“ (papildomas laukas – pareigos).

- Sudarykite rinktinės narių (tiek krepšininkų, tiek personalo), dalyvavusių visose trijose stovyklose, sąrašą ir atspausdinkite jį ekrane.
- Sudarykite visų puolėjų, dalyvavusių rinktinės stovyklose, sąrašą ir ekrane atspausdinkite jų vardus, pavardes bei ūgį. Sudarykite vyr. trenerių sąrašą ir atspausdinkite ekrane visų jų vardus ir pavardes.
- Sudarykite ir surikiuokite rinktinės senjorų sąrašą, pateikdami pilną informaciją apie juos. Krepšininkas yra senjoras, jei jam daugiau, nei 30 metų. Personalas yra senjoras, jei jam daugiau, nei 50 metų. Krepšininkus rikiuokite pagal poziciją, pavardę ir vardą, o personalo narius – pagal pareigas, pavardę ir vardą. Rezultatus įrašykite į failą „Senjorai.csv“.
- Sudarykite visų į rinktinę pakviestų krepšininkų sąrašą. Jų duomenis įrašykite į failą „Rinktinė.csv“. Sudarykite viso į rinktinę pakviesto personalo sąrašą. Jų duomenis įrašykite į failą „Personalas.csv“.

U5_2. Automobilių parkas. VŠĮ „Greitis“ plečiasi ir atidaro naujus filialus. Pirmoje eilutėje yra miestas, antrėje – adresas, trečioje – e-pašto adresas. Toliau informacija apie automobilius. Įmonės automobilių parką sudaro lengvieji automobiliai, krovininiai automobiliai ir mikroautobusai. Sukurkite abstrakčiąją klasę „Transportas“ (laukai – valstybinis numeris, gamintojas, modelis, pagaminimo metai ir mėnuo, atliktos techninės apžiūros data, kuras, vidutinės kuro sąnaudos (100km)), kurią paveldės klasės „Lengvasis“ (papildomas laukas – odometro rodmenys), „Krovininis“ (papildomas laukas – priekabos talpa) ir „Mikroautobusas“ (papildomas laukas – sėdimų vietų skaičius).

- Raskite geriausią transporto priemonę kiekvienoje grupėje, atspausdinkite jos gamintoją, modelį, valstybinius numerius ir amžių. Lengvasis geriausias – nuvažiuota mažiausiai kilometrų, krovininis geriausias – didžiausia priekabos talpa, mikroautobusas geriausias – daugiausia sėdimų vietų.
- Raskite, kuriame filiale mikroautobusai yra seniausi (vidutinis jų amžius didžiausias). Filialo duomenis atspausdinkite ekrane.
- Sudarykite kiekvieno filialo krovininių automobilių sąrašą, surikiuokite pagal gamintoją ir modelį.
- Nustatykite kiekvienai transporto priemonei artimiausios techninės apžiūros datą, jei lengvajam automobiliui techninė apžiūra galioja 2 metus, kroviniam automobiliui – metus, o mikroautobusui – pusę metų. Į failą „Apžiūra.txt“ įrašykite tų transporto priemonių duomenis (transporto priemonių gamintojus, modelius, valstybinius numerius, techninės apžiūros galiojimo pabaigą), kurioms iki techninės apžiūros galiojimo pabaigos liko mažiau kaip 1 mėnėsis. Jei techninės apžiūros galiojimas pasibaigęs, tuos automobilius pažymėkite papildomais ženklais.

U5_3. Studentų atstovybė. Turite ir dviejų ankstesniųjų metų studentų atstovybės narių duomenis. Pirmoje eilutėje yra metai. Studentų atstovybės nariai yra tiek šiuo metu studijuojantys, tiek jau pabaigę studijas. Sukurkite abstrakčiąją klasę „Narys“ (laukai – pavardė, vardas, gimimo data, telefono numeris), kurią paveldės klasės „Studentas“ (papildomi laukai – studento pažymėjimo numeris, kursas, požymis „fuksas“) ir „Senis“ (papildomas laukas – darbovietė).

- Raskite ir atspausdinkite ekrane, kiek kiekvienais metais studentų atstovybė turėjo narių.
- Raskite ir atspausdinkite ekrane, kurį mėnesį bus švenčiama daugiausiai visų studentų atstovybės narių gimtadienių kiekvienais metais.
- Sudarykite ir surikiuokite studentų atstovybės senjorų sąrašą, pateikdami pilną informaciją apie juos. Studentas yra senjoras, jei jam daugiau, nei 22 metai. Senis yra senjoras, jei jam daugiau, nei 25 metai. Studentus rikiuokite pagal kursą, pavardę ir vardą, o senius – pagal darbovietę, pavardę ir vardą. Rezultatus įrašykite į failą „Senjorai.csv“.

- Sudarykite studentų, kurie ankstesniaisiais metais priklausė atstovybei ir buvo studentai, o šiais metais baigė studijas ir tapo seniais. Visus jų duomenis įrašykite į failą „Seniai.csv“.

U5_4. Biblioteka. Turite visų KTU bibliotekų padalinių duomenis. Pirmoje eilutėje yra pavadinimas, antroje – adresas, trečioje – telefonas. Bibliotekoje galima rasti įvairių leidinių – knygų, žurnalų ir laikraščių. Sukurkite abstrakčiąją klasę „Leidinys“ (laukai - pavadinimas, tipas, leidykla, išleidimo metai, puslapių skaičius, tiražas), kurią paveldės klasės „Knyga“ (papildomi laukai - ISBN, autorius), „Žurnalas“ (papildomi laukai – ISBN, numeris, išleidimo mėnesis) ir „Laikraštis“ (papildomi laukai – data, numeris, išleidimo mėnesis ir diena).

- Raskite didžiausiu tiražu išleistą knygą, žurnalą ir laikraštį kiekviename padalinyje, ekrane kiekvienam jų atspausdinkite filialo pavadinimą, leidinio pavadinimą ir tiražą.
- Sudarykite visų leidinių, kuriuos galima rasti tik viename filiale, sąrašą. Visus duomenis apie šiuos leidinius įrašykite į failą „RetiLeidiniai.csv“.
- Sudarykite ir surikiuokite nenaujų leidinių sąrašą, pateikdami pilną informaciją apie juos. Knyga yra nauja, jei nuo išleidimo prabėgo daugiau, nei metai. Žurnalas yra naujas, jei nuo išleidimo prabėgo, daugiau nei mėnesis. Laikraštis yra naujas, jei nuo išleidimo prabėgo daugiau, nei savaitė. Knygas rikiuokite pagal išleidimo metus, žurnalus – pagal išleidimo metus ir mėnesius, o laikraščius – pagal išleidimo metus, mėnesius ir dienas. Rezultatus įrašykite į failą „Nenauji.csv“.
- Raskite visus leidinius, išleistus leidyklos „Technologija“, visus duomenis apie šiuos leidinius įrašykite į failą „Technologija.csv“.

U5_5. Proto mūšis. Proto mūšius organizuojančios studentų atstovybės nusprendė susivienyti ir sudaryti bendrą klausimų bazę. Pirmoje eilutėje nurodytas studentų atstovybės pavadinimas. Toliau yra klausimai. Proto mūšio klausimai gali būti tik dviejų rūšių: testo tipo – su galimais atsakymų variantais ir muzikiniai. Sukurkite abstrakčiąją klasę „Klausimas“ (laukai – tema, sudėtingumas, klausimo autorius, klausimo tekstas, teisingas atsakymas, balai), kurią paveldės klasės „Variantai“ (papildomi laukai – atsakymo variantai) ir „Muzikinis“ (papildomas laukas – failo pavadinimas).

- Raskite, kas sukūrė daugiausiai klausimų, autoriaus vardą bei klausimų kiekį atspausdinkite ekrane.
- Raskite, kas sukūrė daugiausiai klausimų kiekvienoje atstovybėje (bendrai paėmus), autoriaus vardą bei klausimų kiekį atspausdinkite ekrane. Kas sukūrė daugiausia muzikinių klausimų kiekvienoje atstovybėje, autoriaus vardą bei klausimų kiekį atspausdinkite ekrane.
- Sudarykite visų klausimų sąrašą, įrašykite į failą „Klausimai.csv“, testo varianto klausimus išrikiuokite pagal temą ir sudėtingumą, o muzikinius – pagal failo pavadinimą.
- Sudarykite visų klausimų, kurių tema „istorinis“, sąrašą, ir įrašykite juos į failą „Istoriniai.csv“.

U5_6. Nekilnojamojo turto agentūra. Turite ir kitų nekilnojamojo turto agentūrų duomenis. Pirmoje eilutėje yra pavadinimas, antroje – adresas, trečioje – telefonas. Nekilnojamojo turto agentūra parduoda butus ir nuosavus namus. Sukurkite abstrakčiąją klasę „NTObjektas“ (laukai mikrorajonas, gatvė, namo numeris, tipas, pastatymo metai, plotas, kambarių skaičius), kurią paveldės klasės „Butas“ (papildomas laukas - aukštas) ir „Namas“ (papildomas laukas – šildymo būdas).

- Raskite, kurioje gatvėje daugiausiai parduodamų nekilnojamojo turto objektų (namų ir butų), ekrane atspausdinkite gatvės pavadinimą ir parduodamų objektų kiekį.
- Raskite seniausią nekilnojamojo turto objektą, ekrane atspausdinkite visą jo informaciją.
- Raskite, kurių namų ar butų skelbimai yra paskelbti daugiau nei vienoje agentūroje (savininkas tikriausiai labai skuba parduoti, ir bus linkęs nuleisti kainą). Išrikiuokite juos pagal gatvės pavadinimą ir namo numerį. Į failą „Kartojasi.csv“ įrašykite informaciją apie šiuos objektus.
- Sudarykite ir surikiuokite didelių NT objektų sąrašą. Namas yra didelis, jei jo plotas didesnis nei 200 kv.m. Namus rikiuokite pagal plotą ir šildymo būdą. Butas yra didelis, jei jo plotas didesnis nei 90 kv.m. Butus rikiuokite pagal plotą ir aukštą. Rezultatus įrašykite į failą „Dideli.csv“.

U5_7. WCG turnyras. Turite trijų turnyro ratų duomenis. Pirmoje eilutėje yra rato numeris, antroje – data. Toliau pateikta informacija apie to rato rezultatus. Turnyre varžosi kelių skirtingų žaidimų („League of Legends“ ir „Counter Strike“) žaidėjai. Sukurkite klasę „Žaidėjas“ (laukai - vardas, pavardė, komanda), kurią paveldės klasės „LoLŽaidėjas“ (papildomi laukai - pozicija, čempionas, nužudymai(K), mirtys(D),

dalyvavimai nužudymuose(A)) ir “CSŽaidėjas” (papildomi laukai - nužudymai(K), mirtys(D), mėgstamiausias ginklas).

- Raskite žaidėją, pademonstravusį geriausią bendrą (per visus tris ratus) asmeninį rezultatą. LoL žaidėjų palyginimui naudokite vadinamąjį KDA santykį (nužudymai + dalyvavimai nužudymuose)/mirtys t.y. (K+A)/D, o CS žaidėjams KD santykį (K/D). Ekrane atspausdinkite jų vardus, pavardes ir komandos pavadinimą.
- Kai kurie žaidėjai dalyvauja tiek LoL, tiek CS turnyruose. Raskite tuos žaidėjus ir įrašykite į failą „Universalus.csv“.
- Sudarykite bendrą žaidėjų rinktinę. Į šią rinktinę patenka LoL žaidėjai, jei jų nužudymų kiekis yra didesnis už nurodytą. Į šią rinktinę patenka CS žaidėjai, jei jų mirčių kiekis yra mažesnis už nurodytą. Rezultatus įrašykite į failą „Rinktine.csv“.
- Sudarykite visų turnyro dalyvių sąrašą, LoL žaidėjus rikiuokite pagal KDA santykį, o CS žaidėjus – pagal KD santykį. Rezultatus įrašykite į failą „Visi.csv“.

U5_8. Turistų informacijos centras. Turizmo informacijos centre perorganizuoti ir atskirai surašyti duomenys apie kiekviename mieste veikiančius muziejus. Keičiasi duomenų formatas. Pirmoje eilutėje – miestas, antroje – atsakingo asmens vardas ir pavardė. Turizmo informacijos centras teikia informaciją apie lankytinas vietas – muziejus, paminklus ir kita. Sukurkite klasę „LankytinaVieta“ (laukai - pavadinimas, adresas, įkūrimo ar pastatymo metai), kurią paveldės klasės “Muziejus” (papildomas laukas – tipas, darbo dienos, požymis „turi gidą“, bilieto kaina) ir “Paminklas” (papildomas laukas – autorius, kam skirtas).

- Suskaičiuokite, kiek muziejų turi gidus, rezultatą atspausdinkite ekrane.
- Raskite seniausią lankytiną vietą, visą informaciją apie ją atspausdinkite ekrane.
- Sudarykite visų lankytinų vietų sąrašą ir įrašykite į failą „VisosVietos.csv“.
- Sudarykite ir surikiuokite naujų lankytinų vietų sąrašą, pateikdami pilną informaciją apie juos. Muziejus yra naujas, jei nuo įkūrimo prabėgo mažiau, nei 2 metai. Paminklas yra naujas, jei nuo pastatymo prabėgo mažiau nei metai. Muziejus rikiuokite pagal bilieto kainas, paminklus – pagal autorius. Rezultatus įrašykite į failą „Nauji.csv“.

U5_9. IMBD. Turite skirtingų kinomanų mėgėjų peržiūrėtus filmų ir serialų sąrašus. Pirmoje eilutėje yra kino mėgėjo vardas pavardė, antroje – gimimo metai, trečioje – miestas. Sudarykite klasę „Įrašas“ (laukai - pavadinimas, žanras, kino studija, du pagrindiniai aktoriai), kurią paveldės klasės „Filmas“ (papildomi laukai – leidimo metai, režisierius, pajamos) ir „Serialas“ (papildomi laukai – pradžios metai, serijų kiekis, pabaigos metai (jei yra), požymis „ar tęsiasi“).

- Raskite ir atspausdinkite ekrane kiekvieno kino mėgėjo mėgstamiausią aktorį (tai aktorius, kuris atliko daugiausiai vaidmenų peržiūrėtuose filmuose ir serialuose).
- Sudarykite filmų ir serialų, kuriuos peržiūrėjo visi kino mėgėjai, sąrašą. Visus duomenis apie juos įrašykite į failą „MatėVisi.csv“.
- Kiekvienam kino mėgėjui sudarykite rekomenduojamų peržiūrėti filmų ir serialų sąrašą, į kurį įtraukite filmus ir serialus, kurių jis nematė, tačiau matė kiti kino mėgėjai. Rekomendacijų sąrašus įrašykite į failus „Rekomendacija_vardas_pavardė.csv“.
- Sudarykite ir surikiuokite naujų filmų ir serialų sąrašą, pateikdami pilną informaciją apie juos. Filmas yra naujas, jei nuo leidimo metų prabėgo mažiau, nei 2 metai. Serialas yra naujas, jei nuo pradžios metų prabėgo mažiau nei metai. Filmus rikiuokite pagal pajamas, serialus – pagal pradžios ir pabaigos metus. Rezultatus įrašykite į failą „Nauji.csv“.

U5_10. Buitinės technikos parduotuvė. Turite informaciją apie skirtingose buitinės technikos parduotuvėse esančius šaldytuvus. Pirmoje eilutėje yra pavadinimas, antroje – adresas, trečioje – telefonas. Parduotuvėje be šaldytuvų galima įsigyti mikrobangų krosnelių ir elektrinis virdulys. Sukurkite abstrakčią klasę „Prietaisas“ (laukai - gamintojas, modelis, energijos klasė, spalva, kaina), kurią paveldės “Šaldytuvas” (papildomi laukai - talpa, montavimo tipas, požymis „turi šaldiklį“, aukštis, plotis, gylis), “Krosnelė” (papildomi laukai – galingumas, programų skaičius) ir “Virdulys” (papildomi laukai – galia, tūris).

- Suskaičiuokite, kiek skirtingų „Siemens“ šaldytuvų, mikrobangų krosnelių ir virdulių modelių siūlo kiekviena parduotuvė, rezultatą atspausdinkite ekrane.
- Sudarykite dešimties pigiausių pastatomų šaldytuvų, kurių talpa 80 litrų ar didesnė, sąrašą. Ekrane atspausdinkite šaldytuvo gamintoją, modelį, talpą ir kainą.

- Ar yra tokių buitinių prietaisų, kuriuos galima įsigyti tik vienoje parduotuvėje? Atspausdinkite tokių prietaisų sąrašą faile „TikTen.csv“.
- Sudarykite ir surikiuokite brangių buitinių prietaisų sąrašą, pateikdami pilną informaciją apie juos. Šaldytuvas yra brangus, jei jo kaina viršija 1000€. Mikrobangų krosnelė yra brangi, jei jos kaina viršija 500€. Virdulys yra brangus, jei jo kaina viršija 50€ Šaldytuvus rikiuokite pagal aukštį, mikrobangų krosneles – pagal galingumą, o virdulius – pagal galią. Rezultatus įrašykite į failą „Brangus.csv“.

U5_11. Juvelyrikos parduotuvė. Turite informaciją apie skirtingose juvelyrikos parduotuvėse esančius žiedus. Pirmoje eilutėje yra pavadinimas, antroje – adresas, trečioje – telefonas. Parduotuvėje galima įsigyti žiedų, auskarų, grandinėlių. Sukurkite abstrakčią klasę „Juvelyrinis“ (laukai - gamintojas, pavadinimas, metalas, svoris, praba, kaina), kurią paveldės „Žiedas“ (papildomas laukas – dydis), „Auskrai“ (papildomas laukas – užsegimo tipas) ir „Grandinėlė“ (papildomas laukas – ilgis).

- Raskite ir atspausdinkite ekrane, kurioje parduotuvėje yra parduodamas brangiausias žiedas, brangiausi auskarai ir brangiausia grandinėlė.
- Ar yra tokių juvelyrinių gaminių, kurių galima įsigyti tik vienoje juvelyrinėje parduotuvėje? Atspausdinkite jų ir parduotuvių sąrašą faile „Unikalūs.csv“.
- Sudarykite juvelyrinių dirbinių, pigesnių nei 300 eurų, sąrašą. Visus duomenis apie juos įrašykite į failą „300.csv“.
- Sudarykite ir surikiuokite brangių juvelyrinių dirbinių sąrašą, pateikdami pilną informaciją apie juos. Žiedas yra brangus, jei jo kaina viršija 500€. Auskarai yra brangūs, jei jų kaina viršija 300€. Grandinėlė yra brangi, jei jos kaina viršija 150€ Žiedus rikiuokite pagal dydį, auskarus – pagal svorį, o grandinėles – pagal ilgį. Rezultatus įrašykite į failą „Brangus.csv“.

U5_12. Kompiuterinis žaidimas. Žaidimo pasaulyje yra dviejų tipų veikėjai - žaidėjo valdomi herojai bei kompiuterio valdomi „NPC“ (*non playable character*). Sugrupavote žaidimo veikėjus pagal rases, ir surašėte jų duomenis į skirtingus failus. Duomenų formatas dabar toks: pirmoje eilutėje – rasės pavadinimas. Antroje – pradinis miestas. Toliau informacija apie žaidimo veikėjus. Sukurkite abstrakčią klasę „Veikėjas“ (laukai - vardas, klasė, gyvybės taškai, mana, žalos taškai, gynybos taškai), kurią paveldės klasės „Herojus“ (papildomi laukai - jėga, vikrumas, intelektas, ypatinga galia), „NPC“ (papildomi laukai - gildija).

- Raskite, kokios klasės veikėjų yra daugiausia, ekrane atspausdinkite klasės pavadinimą bei toks klasės veikėjų skaičių.
- Pastebėjote, jog grupuojant veikėjus įsivėlė klaidų, ir tas pats veikėjas buvo įtrauktas į kelis skirtingų rasių duomenų failus. Raskite besikartojančius veikėjus, jų vardus surašykite į failą „Klaidos.csv“
- Sudarykite žaidimo veikėjų, kurių gyvybės taškai bent 100, o gynybos taškai – bent 30, sąrašą. Į failą „Tankai.csv“ įrašykite visą informaciją apie juos.
- Sudarykite ir surikiuokite bendrą veikėjų rinktinę. Į šią rinktinę patenka herojai, jei jų intelekto reikšmė viršija nurodytą dydį. Į šią rinktinę patenka NPC, jei jų žalos taškai neviršija nurodyto dydžio. Herojus rikiuokite pagal intelektą, o NPC – pagal žalos taškus. Rezultatus įrašykite į failą „Rinktime.csv“.

U5_13. Krepšinio rinktinė. Turite ne tik šių, bet ir dviejų ankstesnių metų į stovyklas pakviestų krepšininkų sąrašus. Keičiasi duomenų failų formatas. Pirmoje eilutėje metai, antroje – stovyklos pradžios data, trečioje – stovyklos pabaigos data. Toliau informacija apie rinktinės narius. Krepšinio rinktinę sudaro ne tik krepšininkai, bet ir pagalbinis personalas – treneriai, gydytojai, masažuotojai ir kt. Sukurkite abstrakčią klasę „Narys“ (laukai – vardas, pavardė, gimimo data), kurią paveldės klasės „Krepšininkas“ (papildomi laukai - ūgis, pozicija, klubas, požymis „pakviestas“, požymis „kapitonas“) ir „Personalas“ (papildomas laukas - pareigos).

- Raskite krepšininkus, žaidusius Kauno „Žalgiryje“, ekrane atspausdinkite jų vardus, pavardes bei pozicijas. Sąrašą surikiuokite pagal pavardę ir vardą.
- Sudarykite ir surikiuokite rinktinės senjorų sąrašą, pateikdami pilną informaciją apie juos. Krepšininkas yra senjoras, jei jam daugiau, nei 30 metų. Personalo narys yra senjoras, jei jam daugiau, nei 50 metų. Krepšininkus rikiuokite pagal poziciją, pavardę ir vardą, o personalo narius – pagal pareigas, pavardę ir vardą. Rezultatus įrašykite į failą „Senjorai.csv“.

- Sudarykite į rinktinę pakviestų krepšininkų ir pagalbinio personalo narių, kurie dalyvavo visose trejose stovyklose, sąrašą. Jų duomenis įrašykite į failą „Senbuviai.csv“.
- Sudarykite sąrašą krepšininkų, kurių ūgis – 2 metrai ir daugiau, į failą „Aukštaūgiai.csv“ įrašykite krepšininkų vardus, pavardes ir ūgį. Jei krepšininkas buvo pakviestas į rinktinę kelis metus, įtraukite jį į sąrašą tik vieną kartą. Atrinkite personalo narius, kurie yra masažuotojai. Jų duomenis įrašykite į failą „Masažuotojai.csv“.

U5_14. Automobilių parkas. Įmonė UAB „Žaibas“ plečiasi ir atidarė naujus filialus. Pirmoje eilutėje yra miestas, antroje – adresas, trečioje – telefonas. Įmonės automobilių parką sudaro lengvieji automobiliai, krovininiai automobiliai ir autobusai. Sukurkite abstrakčiąją klasę „Transportas“ (laukai – valstybinis numeris, gamintojas, modelis, pagaminimo metai ir mėnuo, atliktos techninės apžiūros data, kuras, vidutinės kuro sąnaudos (100km)), kurią paveldės klasės „Lengvasis“ (papildomas laukas – odometro rodmenys), „Krovininis“ (papildomas laukas – priekabos talpa) ir „Autobusas“ (papildomas laukas – sėdimų vietų skaičius).

- Raskite geriausią transporto priemonę kiekvienoje grupėje, atspausdinkite jos gamintoją, modelį, valstybinius numerius ir amžių. Lengvasis geriausias – nuvažiuota mažiausiai kilometrų, krovininis geriausias – didžiausia priekabos talpa, autobusas geriausias – daugiausia sėdimų vietų.
- Sudarykite visų benzinu varomų lengvųjų automobilių sąrašą, ekrane atspausdinkite jų valstybinį numerį, gamintoją, modelį, bei pagaminimo metus.
- Sudarykite kiekvieno filialo autobusų sąrašą, surikiuokite pagal pagaminimo metus ir gamintoją.
- Nustatykite kiekvienai transporto priemonei artimiausios techninės apžiūros datą, jei lengvajam automobiliui techninė apžiūra galioja 2 metus, kroviniam automobiliui – metus, o autobusui – pusę metų. Į failą „Apžiūra.txt“ įrašykite tų transporto priemonių duomenis (transporto priemonių gamintojus, modelius, valstybinius numerius, techninės apžiūros galiojimo pabaigą), kurioms iki techninės apžiūros galiojimo pabaigos liko mažiau kaip 2 mėnesiai.

U5_15. Studentų atstovybė. Turite ir dviejų ankstesniųjų metų studentų atstovybės narių duomenis. Pirmoje eilutėje metai. Studentų atstovybės nariai yra tiek šiuo metu studijuojantys, tiek jau pabaigę studijas. Sukurkite abstrakčią klasę „Narys“ (laukai - pavardė, vardas, gimimo data, telefono numeris), kurią paveldės klasės „Studentas“ (papildomi laukai - studento pažymėjimo numeris, kursas, požymis „fuksas“) ir „Senis“ (papildomas laukas – darbovietė).

- Raskite vyriausią studentų atstovybės narį, priklausiusį studentų atstovybei, ekrane atspausdinkite jo vardą, pavardę ir amžių.
- Sudarykite visų senių, vyresnių nei 30 metų, sąrašą ir atspausdinkite ekrane.
- Sudarykite ir surikiuokite studentų atstovybės senjorų sąrašą, pateikdami pilną informaciją apie juos. Studentas yra senjoras, jei jam daugiau, nei 22 metai. Senis yra senjoras, jei jam daugiau, nei 27 metai. Studentus rikiuokite pagal kursą, pavardę ir vardą, o senius – pagal darbovietę, pavardę ir vardą. Rezultatus įrašykite į failą „Senjorai.csv“.
- Sudarykite studentų, kurie priklausė atstovybei, sąrašą. Visus jų duomenis įrašykite į failą „Buvo.csv“.

U5_16. Biblioteka. Turite visų KTU bibliotekų padalinių duomenis. Keičiasi duomenų formatas. Pirmoje eilutėje pavadinimas, antroje – adresas, trečioje – telefonas. Bibliotekoje galima rasti įvairių leidinių – knygų, žurnalų ir laikraščių. Sukurkite abstrakčiąją klasę „Leidiny“ (laukai - pavadinimas, tipas, leidykla, išleidimo metai, puslapių skaičius, tiražas), kurią paveldės klasės „Knyga“ (papildomi laukai - ISBN, autorius(-iai)), „Žurnalas“ (papildomi laukai – ISBN, numeris) ir „Laikraštis“ (papildomi laukai – data, numeris).

- Suskaičiuokite, kiek leidinių, senesnių nei 2 metų, yra kiekviename filiale. Rezultatą atspausdinkite ekrane.
- Sudarykite visų leidinių, kurių tipas yra „mokslinis“ sąrašą. Visą informaciją apie juos atspausdinkite ekrane.
- Sudarykite ir surikiuokite nenaujų leidinių sąrašą, pateikdami pilną informaciją apie juos. Knyga yra nauja, jei nuo išleidimo prabėgo daugiau, nei metai. Žurnalas yra naujas, jei nuo išleidimo prabėgo, daugiau nei mėnesis. Laikraštis yra naujas, jei nuo išleidimo prabėgo daugiau, nei

savaitė. Knygas rikiuokite pagal išleidimo metus, žurnalus – pagal išleidimo metus ir mėnesius, o laikraščius – pagal išleidimo metus, mėnesius ir dienas. Rezultatus įrašykite į failą „Nenauji.csv“.

- Sudarykite visų leidinių, kurių tiražas didesnis nei 10 000 vnt., sąrašą, surašykite šių leidinių pavadinimus ir tiražus į failą „PopuliarūsLeidiniai.csv“.

U5_17. Protų mūšis. Protų mūšius organizuojančios studentų atstovybės nusprendė susivienyti ir sudaryti bendrą klausimų bazę. Pirmoje eilutėje nurodytas studentų atstovybės pavadinimas. Toliau yra klausimai. Protų mūšio klausimai gali būti tik dviejų rūšių: su galimais atsakymų variantais ir muzikiniai. Sukurkite abstrakčiąją klasę „Klausimas“ (laukai – tema, sudėtingumas, klausimo autorius, klausimo tekstas, teisingas atsakymas, balai), kurią paveldės klasės „Variantai“ (papildomi laukai – atsakymo variantai) ir „Muzikinis“ (papildomas laukas – failo vardas).

- Raskite, kiek yra I, II ir III sudėtingumo lygio klausimų (visų tipų), rezultatus atspausdinkite ekrane.
- Raskite, kas sukūrė daugiausiai klausimų kiekvienoje atstovybėje (bendrai paėmus), autoriaus vardą bei klausimų kiekį atspausdinkite ekrane.
- Sudarykite sudėtingiausių muzikinių klausimų sąrašą. Įrašykite į failą „SudėtingiMuzikiniai.csv“. Sudarykite bendrai sudėtingiausių klausimų sąrašą. Įrašykite į failą „SudėtingiBendrai.csv“.
- Sudarykite sąrašą klausimų iš temos „Linksmasis“. Įrašykite į failą „Linksmieji.csv“ ir išrikiuokite testo varianto klausimus pagal temą ir sudėtingumą, o muzikinius – pagal failo pavadinimą.

U5_18. Nekilnojamojo turto agentūra. Turite ir kitų nekilnojamojo turto agentūrų duomenis. Keičiasi duomenų formatas. Pirmoje eilutėje pavadinimas, antroje – adresas, trečioje – telefonas. Nekilnojamojo turto agentūra parduoda butus ir nuosavus namus. Sukurkite abstrakčią klasę „NTObjektas“ (laukai mikrorajonas, gatvė, namo numeris, tipas, pastatymo metai, plotas, kambarių skaičius), kurią paveldės klasės „Butas“ (papildomas laukas - aukštas) ir „Namas“ (papildomas laukas – šildymo būdas).

- Raskite, kokio tipo namų ir butų daugiausia siūlo kiekviena agentūra, ekrane atspausdinkite agentūros pavadinimą, namo ar buto tipą, ir to tipo namų ar butų skaičių.
- Raskite, kurių namų ar butų skelbimai yra paskelbti daugiau nei vienoje agentūroje (savininkas tikriausiai labai skuba parduoti, ir bus linkęs nuleisti kainą). Į failą „Kartojasi.csv“ įrašykite informaciją apie šiuos objektus.
- Sudarykite visų mikrorajonų, kuriuose šiuo metu parduojami nekilnojamojo turto objektai, sąrašą, mikrorajonų pavadinimus surašykite faile „Mikrorajonai.csv“.
- Sudarykite ir surikiuokite didelių NT objektų sąrašą. Namas yra didelis, jei jo plotas didesnis nei 200 kv.m. Namus rikiuokite pagal plotą ir šildymo būdą. Butas yra didelis, jei jo plotas didesnis nei 90 kv.m. Butus rikiuokite pagal plotą ir aukštą. Rezultatus įrašykite į failą „Dideli.csv“.

U5_19. WCG turnyras. Turite trijų turnyro ratų duomenis. Keičiasi duomenų formatas. Pirmoje eilutėje rato numeris, antroje – data. Toliau pateikta informacija apie to rato rezultatus. Turnyre varžosi kelių skirtingų žaidimų („League of Legends“ ir „Counter Strike“) žaidėjai. Sukurkite klasę „Žaidėjas“ (laukai - vardas, pavardė, komanda), kurią paveldės klasės „LoLŽaidėjas“ (papildomi laukai - pozicija, čempionas, nužudymai(K), mirtys(D), dalyvavimai nužudymuose(A)) ir „CSŽaidėjas“ (papildomi laukai - nužudymai(K), mirtys(D), mėgstamiausias ginklas).

- Raskite LoL ir CS žaidėją, pademonstravusį geriausią bendrą (per visus tris ratus) asmeninį rezultatą. LoL žaidėjų palyginimui naudokite KDA santykį (nužudymai + dalyvavimai nužudymuose)/mirtys t. y. (K+A)/D, o CS žaidėjams palyginti - KD santykį (K/D). Ekrane atspausdinkite jų vardus, pavardes ir komandos pavadinimą.
- Sudarykite visų turnyre dalyvaujančių komandų sąrašą, ir atspausdinkite faile „Komandos.csv“.
- Sudarykite bendrą žaidėjų rinktinę. Į šią rinktinę patenka LoL žaidėjai, jei jų nužudymų kiekis yra didesnis už nurodytą. Į šią rinktinę patenka CS žaidėjai, jei jų mirčių kiekis yra mažesnis už nurodytą. Rezultatus įrašykite į failą „Rinktine.csv“.
- Sudarykite visų turnyro dalyvių sąrašą, LoL žaidėjus rikiuokite pagal KDA santykį, o CS žaidėjus – pagal KD santykį. Rezultatus įrašykite į failą „Visi.csv“.

U5_20. Turistų informacijos centras. Turizmo informacijos centre perorganizuoti ir atskirai surašyti duomenys apie kiekviename mieste veikiančius muziejus. Pirmoje eilutėje – miestas, antroje – atsakingo asmens vardas ir pavardė. Turizmo informacijos centras teikia informaciją apie lankytinus muziejus ir nacionalinius parkus. Sukurkite klasę „LankytinaVieta“ (laukai - pavadinimas, adresas, metai), kurią

paveldės klasės “Muziejus” (papildomas laukas – tipas, darbo dienos, požymis „turi gidą“, bilieto kaina) ir “Paminklas” (papildomi laukai – autorius, paminklo pavadinimas).

- Suskaičiuokite, kiek lankytinų vietų turi gidus, rezultatą atspausdinkite ekrane.
- Raskite, kokio tipo lankytinas vietas galima aplankyti kiekviename mieste savaitgaliais, ir atspausdinkite jų tipus ekrane.
- Sudarykite paminklų sąrašą, kurių autorius nurodomas, įvedant iš klaviatūros. Į failą „PaminklaiAutorius.csv“ išveskite rezultatus, pateikdami pavadinimą, adresą, metus, autorių, paminklo pavadinimą.
- Sudarykite ir surikiuokite naujų lankytinų vietų sąrašą, pateikdami pilną informaciją apie juos. Muziejus yra naujas, jei nuo įkūrimo prabėgo mažiau, nei 2 metai. Paminklas yra naujas, jei nuo pastatymo prabėgo mažiau nei metai. Muziejus rikiuokite pagal bilieto kainas, paminklus – pagal autorius. Rezultatus įrašykite į failą „Nauji.csv“.

U5_21. IMBD. Turite skirtingų kinomanų mėgėjų peržiūrėtus filmų ir serialų sąrašus. Keičiasi duomenų formatas. Pirmoje eilutėje kino mėgėjo vardas pavardė, antroje - gimimo metai, trečioje - miestas. Sudarykite klasę „Įrašas“ (laukai - pavadinimas, žanras, kino studija, aktoriai), kurią paveldės klasės „Filmas“ (papildomi laukai – leidimo metai, režisierius, pajamos) ir „Serialas“ (papildomi laukai – pradžios metai, serijų kiekis, pabaigos metai (jei yra), požymis „ar tęsiasi“).

- Raskite, kuris kino mėgėjas matė daugiausiai filmų ir serialų, kuriuose vaidino N. Kidman, ekrane atspausdinkite jo vardą, bei matytų filmų ir serialų skaičių.
- Raskite, kiek sąraše esančių filmų ir serialų yra sukurti kino studijos „Warner Bros“, rezultatą atspausdinkite ekrane.
- Kiekvienam kino mėgėjui sudarykite rekomenduojamų peržiūrėti filmų ir serialų sąrašą, į kurį įtraukite filmus ir serialus, kurių jis nematė, tačiau matė kiti kino mėgėjai. Rekomendacijų sąrašus įrašykite į failus „Rekomendacija_vardas_pavardė.csv“.
- Sudarykite ir surikiuokite naujų filmų ir serialų sąrašą, pateikdami pilną informaciją apie juos. Filmas yra naujas, jei nuo leidimo metų prabėgo mažiau, nei 2 metai. Serialas yra naujas, jei nuo pradžios metų prabėgo mažiau nei metai. Filmus rikiuokite pagal pajamas, serialus – pagal pradžios ir pabaigos metus. Rezultatus įrašykite į failą „Nauji.csv“.

U5_22. Buitinės technikos parduotuvė. Turite informaciją apie skirtingose buitinės technikos parduotuvėse esančius šaldytuvus. Keičiasi duomenų formatas. Pirmoje eilutėje pavadinimas, antroje – adresas, trečioje – telefonas. Parduotuvėje be šaldytuvų galima įsigyti mikrobangų krosnelių ir elektrinius virdulys. Sukurkite abstrakčią klasę „Prietaisas“ (laukai - gamintojas, modelis, energijos klasė, spalva, kaina), kurią paveldės “Šaldytuvas” (papildomi laukai - talpa, montavimo tipas, požymis „turi šaldiklį“, aukštis, plotis, gylis), “Krosnelė” (papildomi laukai – galingumas, programų skaičius) ir “Virdulys” (papildomi laukai – galia, tūris).

- Raskite, kokių skirtingų spalvų šaldytuvus galima įsigyti, spalvų sąrašą atspausdinkite ekrane. Raskite, kokių skirtingų spalvų elektrinius virdulius galima įsigyti, spalvų sąrašą atspausdinkite ekrane.
- Raskite pigiausią A+ klasės šaldytuvą, mikrobangų krosnelę ir virdulį, kiekvieno jų visą informaciją atspausdinkite ekrane.
- Sudarykite šaldytuvų, kurių plotis nuo 52 iki 56 cm, sąrašą ir įrašykite visus duomenis apie šiuos šaldytuvus į failą „Tilps.csv“.
- Sudarykite ir surikiuokite brangių buitinių prietaisų sąrašą, pateikdami pilną informaciją apie juos. Šaldytuvas yra brangus, jei jo kaina viršija 1000€. Mikrobangų krosnelė yra brangi, jei jos kaina viršija 500€. Virdulys yra brangus, jei jo kaina viršija 50€ Šaldytuvus rikiuokite pagal aukštį, mikrobangų krosneles – pagal galingumą, o virdulius – pagal galią. Rezultatus įrašykite į failą „Brangus.csv“.

U5_23. Juvelyrikos parduotuvė. Turite informaciją apie skirtingose juvelyrikos parduotuvėse esančius žiedus. Pirmoje eilutėje pavadinimas, antroje – adresas, trečioje – telefonas. Parduotuvėje galima įsigyti žiedų, auskarų, grandinėlių. Sukurkite abstrakčią klasę „JuvelyrinisGaminys“ (laukai - gamintojas, pavadinimas, metalas, svoris, praba, kaina), kurią paveldės “Žiedas” (papildomas laukas – dydis), “Auskarai” (papildomas laukas – užsegimo tipas) ir “Grandinėlė” (papildomas laukas – ilgis).

- Raskite sunkiausią žiedą, auskarus ir grandinėlę. Ekrane atspausdinkite visą informaciją apie kiekvieną jų.
- Raskite, kiek aukščiausios prabos juvelyrinių gaminių yra kiekvienoje parduotuvėje, rezultata atspausdinkite ekrane.
- Ar yra tokių juvelyrinių dirbinių, kurių galima įsigyti visose juvelyrinėse parduotuvėse? Atspausdinkite visą informaciją apie juos faile „Visur.csv“.
- Sudarykite ir surikiuokite brangių juvelyrinių dirbinių sąrašą, pateikdami pilną informaciją apie juos. Žiedas yra brangus, jei jo kaina viršija 500€. Auskarai yra brangūs, jei jų kaina viršija 300€. Grandinėlė yra brangi, jei jos kaina viršija 150€ Žiedus rikiuokite pagal dydį, auskarus – pagal svorį, o grandinės – pagal ilgį. Rezultatus įrašykite į failą „Brangus.csv“.

U5_24. Kompiuterinis žaidimas. Žaidimo pasaulyje yra dviejų tipų veikėjai - žaidėjo valdomi herojai bei kompiuterio valdomi „NPC“ (*non playable character*). Sugrupavote žaidimo veikėjus pagal rases, ir surašėte jų duomenis į skirtingus failus. Duomenų formatas dabar toks: pirmoje eilutėje – rasės pavadinimas. Antroje – pradinis miestas. Toliau informacija apie žaidimo veikėjus. Sukurkite abstrakčią klasę „Veikėjas“ (laukai - vardas, klasė, gyvybės taškai, mana, žalos taškai, gynybos taškai), kurią paveldės klasės „Herojus“ (papildomi laukai - jėga, vikrumas, intelektas, ypatinga galia), „NPC“ (papildomi laukai - gildija).

- Raskite kiekvienos klasės daugiausiai gyvybės taškų turintį veikėją, ekrane atspausdinkite jo vardą, rasę, klasę ir gyvybės taškų kiekį.
- Sudarykite visų veikėjų klasių sąrašą ir įrašykite į failą „Klasės.csv“ .
- Norite, jog kiekviena rasė turėtų bent po vieną kiekvienos klasės herojų ir NPC. Raskite, kokių klasių herojų ar NPC „trūksta“ kiekvienai rasei. Į failą „Trūkstami.csv“ įrašykite kiekvienos rasės pavadinimą, trūkstamų herojų klasių sąrašą, trūkstamų NPC klasių sąrašą.
- Sudarykite ir surikiuokite bendrą veikėjų rinktinę. Į šią rinktinę patenka herojai, jei jų intelekto reikšmė viršija nurodytą dydį. Į šią rinktinę patenka NPC, jei jų žalos taškai neviršija nurodyto dydžio. Herojus rikiuokite pagal intelektą, o NPC – pagal žalos taškus. Rezultatus įrašykite į failą „Rinktime.csv“.