

## Paveldėjimas

### Užduotis. Prekės.

Faile "Prekes.txt" duota informacija apie gautas prekes: *prekės pavadinimas, prekės tipas (maisto, daržovės, ūkio,...), kaina, kiekis*. Sprendimo eiga:

1. Sudarykite bazinę klasę `Preke`: *prekės pavadinimas, prekės tipas (maisto, daržovės, ūkio,...), kaina*. Klasėje turi būti: konstruktorius be parametru; konstruktorius su parametrais ir numatytomis reikšmėmis; užklotas metodas `ToString()`.
2. Sudarykite išvestinę klasę `Prekes`: *kiekis*. Klasėje turi būti: konstruktorius be parametru; konstruktorius su parametrais ir numatytomis reikšmėmis; užklotas metodas `ToString()`.
3. Sudarykite pagrindinę klasę `Program`. Joje sukurkite skaitymo iš duomenų failo ir spausdinimo lentelę į atsakymų failą metodus.
4. Sukurkite duomenų failą "Prekes.txt".
5. Pagrindiniame metode `Main()` aprašykite objektų dinaminį masyvą (`List`) duomenims saugoti. Patikrinkite įvedimo ir spausdinimo metodų darbą.
6. Pagrindiniame metode `Main()` aprašykite naują objektų dinaminį masyvą (`List`). Jame surašykite visus pradinio dinaminio masyvo elementus.
7. Iš konsolės įveskite norimą prekės tipą ir visus šio tipo elementus pašalinkite iš naujojo sąrašo.
8. Klasę `Preke` papildykite užklotu metodu `CompareTo()` lyginti kainoms ir prekių pavadinimams.
9. Surikiuokite naująjį sąrašą kainų mažėjimo ir alfabetine tvarka. Raskite šio sąrašo prekių kiekio vidurkį.
10. Klasę `Preke` paverskite abstrakčia klase. Joje aprašykite abstraktų metodą `Suma()`.
11. Klasėje `Prekės` realizuokite abstraktų metodą `Suma()` vienos prekės sumai ( $\text{kiekis} * \text{kaina}$ ) rasti.
12. Pagrindinėje klasėje `Program` užrašykite metodą `Sum()` viso dinaminio rinkinio prekių sumai rasti. Pagrindiniame metode `Main()` raskite naujojo sąrašo prekių visą sumą.

Duomenų failas "Prekes.txt":

Pradiniai duomenys:
Duona;maisto; 0,92; 100
Sviestas;maisto;2,42; 200
Bulvės;daržovės;0,27;200
Kopūstai;daržovės;1,09;300
Morkos;daržovės;0,88;150
Burokai;daržovės;0,88;120

Rezultatų failas "Rezultatai.txt":

Rezultatai:			
Pradinis sąrašas			
Pavadinimas	Tipas	Kaina	Kiekis
Duona	maisto	0,92	100
Sviestas	maisto	2,42	200
Bulvės	daržovės	0,27	200
Kopūstai	daržovės	1,09	300
Morkos	daržovės	0,88	150
Burokai	daržovės	0,88	120
Suformuotas sąrašas			
Pavadinimas	Tipas	Kaina	Kiekis
Bulvės	daržovės	0,27	200
Kopūstai	daržovės	1,09	300
Morkos	daržovės	0,88	150
Burokai	daržovės	0,88	120
Rikiuotas suformuotas sąrašas			
Pavadinimas	Tipas	Kaina	Kiekis
Kopūstai	daržovės	1,09	300
Burokai	daržovės	0,88	120
Morkos	daržovės	0,88	150
Bulvės	daržovės	0,27	200
Kiekio vidurkis = 192,50			
Naujo sąrašo prekių visa suma = 618,60			

### Uždavinys.

- Sudarykite bazinę klasę Preke:

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

namespace Pratybos3
{
    public class Preke
    {
        public string pavadinimas { get; set; } // prekės pavadinimas
        public string tipas { get; set; }       // tipo pavadinimas (maisto, drabužiai,...)
        public double kaina { get; set; }       // piniginė vertė
    }
}
```

```

    /** konstruktorius be parametru */
    public Preke()
    {
    }

    /** konstruktorius su parametrais */
    public Preke(string pavadinimas = "", string tipas = "", double kaina = 0.0)
    {
        this.pavadinimas = pavadinimas;
        this.tipas = tipas;
        this.kaina = kaina;
    }

    public override string ToString()
    {
        string eilute;
        eilute = string.Format("{0, -16} {1, -20} {2, 5:f}",
            pavadinimas, tipas, kaina);
        return eilute;
    }
}

```

- Sudarykite išvestinę klasę Prekes:

```

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

namespace Pratybos3
{
    public class Prekes : Preke
    {
        public int kiek { get; set; } // turimas kiekis

        /** konstruktorius be parametru */
        public Prekes()
        {
        }

        /** konstruktorius su parametrais */
        public Prekes(string pav = "", string tipas = "", double kaina = 0.0, int kiek = 0)
            : base(pav, tipas, kaina)
        {
            this.kiek = kiek;
        }

        public override string ToString()
        {
            string eilute;
            eilute = string.Format("{0} {1,6:d}", base.ToString(), kiek);
            return eilute;
        }
    }
}

```

- Sudarykite pagrindinės klasės Program skaitymo iš duomenų failo ir spausdinimo lentelę į atsakymų failą metodus. Pagrindiniame metode Main() aprašykite objektų dinaminį masyvą (List) duomenims saugoti. Patikrinkite įvedimo ir spausdinimo metodų darbą:

```

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

namespace Pratybos3
{
    class Program
    {
        const string CFd1 = "..\\..\\Prekes.txt";
        const string CFr = "..\\..\\Rezultatai.txt";
        static void Main(string[] args)
        {
            if (File.Exists(CFr))
                File.Delete(CFr);
            List<Prekes> PrekiuList = new List<Prekes>();
            Skaityti(CFd1, PrekiuList);
            Spausdinti(CFr, PrekiuList, "Pradinis sąrašas");
        }

        //-----
        /** Skaitomi duomenys iš failo
        @param fv duomenų failo vardas
        @param PrekiuList - objektų rinkinys prekių duomenims saugoti */
        static void Skaityti(string fv, List<Prekes> PrekiuList)
        {
            using (StreamReader reader = new StreamReader(fv, Encoding.GetEncoding(1257)))
            {
                string line;
                while ((line = reader.ReadLine()) != null)
                {
                    string[] parts = line.Split(';');
                    //int ind1 = int.Parse(parts[0]);
                    string pav = parts[0].Trim();
                    string tema = parts[1].Trim();
                    double kaina = double.Parse(parts[2]);
                    int kiek = int.Parse(parts[3]);
                    Prekes pr = new Prekes(pav, tema, kaina, kiek);
                    PrekiuList.Add(pr);
                }
            }
        }

        //-----
        /** Spausdinami duomenys
        @param fv - rezultatu failo vardas
        @param PrekiuList - objektų rinkinys prekių duomenims saugoti
        @param info - lentelės pavadinimas */
        static void Spausdinti(string fv, List<Prekes> PrekiuList, string info)
        {
            const string virsus =
                "-----\r\n"
                + " Pavadinimas          Tipas          Kaina    Kiekis \r\n"
                + "-----";
            using (var fr = File.AppendText(fv))
            {
                fr.WriteLine(info);
                fr.WriteLine(virsus);
                for (int i = 0; i < PrekiuList.Count; i++)
            }
        }
    }
}

```

```

        {
            Prekes pr = PrekiuList[i];
            fr.WriteLine("{0}", pr);
        }
        fr.WriteLine("-----" +
            "-----\r\n");
    }
}
}

```

- Išbandykite programą. Pasitikrinkite atsakymus.

## Antras žingsnis.

- Pagrindiniame metode `Main()` aprašykite naują objektų dinaminį masyvą (`List`). Jame surašykite visus pradinio dinaminio masyvo elementus. Tam sukurkite metodą `Perrašyti`. Iš konsolės įveskite norimą prekės tipą (pvz.: maisto) ir visus šio tipo elementus pašalinkite iš naujojo sąrašo.

```

static void Main(string[] args)
{
    if (File.Exists(CFr))
        File.Delete(CFr);
    List<Prekes> PrekiuList = new List<Prekes>();
    Skaityti(CFd1, PrekiuList);
    Spausdinti(CFr, PrekiuList, "Pradinis sąrašas");

    List<Prekes> NaujasPrekiuList = new List<Prekes>();
    Perrašyti(PrekiuList, NaujasPrekiuList);
    Console.WriteLine("Įveskite prekės tipą");
    string prekėsTipas = Console.ReadLine();
    NaujasPrekiuList.RemoveAll(item => item.tipas == prekėsTipas);
    if (NaujasPrekiuList.Count() > 0)
    {
        Spausdinti(CFr, NaujasPrekiuList, "Suformuotas sąrašas");
    }
    else
    {
        using (var fr = File.AppendText(CFr))
        {
            fr.WriteLine("Naujas sąrašas tuščias");
        }
    }
}

```

- Įterpkite į Program klasę metodą `Perrašyti`:

```

//-----
/** Perrašo vieno dinaminio masyvo elementus į kitą masyvą
@param PrekiuList - turimas objektų rinkinys prekių duomenims saugoti
@param NaujasPrekiuList - formuojamas objektų rinkinys */
static void Perrašyti(List<Prekes> PrekiuList,
    List<Prekes> NaujasPrekiuList)
{
    for (int i = 0; i < PrekiuList.Count; i++)
    {
        Prekes p = new Prekes(PrekiuList[i].pavadinimas,
            PrekiuList[i].tipas, PrekiuList[i].kaina,
            PrekiuList[i].kiek);
        NaujasPrekiuList.Add(p);
    }
}

```

- Išbandykite programą. Pasitikrinkite atsakymus.

### U Trečias žingsnis.

- Klasę Preke papildykite užklotu metodu CompareTo() lyginti kainoms ir prekių pavadinimams.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Pratybos3
{
    public class Preke : IComparable<Preke>
    {
        public string pavadinimas { get; set; } // prekės pavadinimas
        public string tipas { get; set; } // tipo pavadinimas (maisto, drabužiai,...)
        public double kaina { get; set; } // piniginė vertė
        /** konstruktorius be parametrų */
        public Preke()
        {
        }

        /** konstruktorius su parametrais */
        public Preke(string pavadinimas = "", string tipas = "", double kaina = 0.0)
        {
            this.pavadinimas = pavadinimas;
            this.tipas = tipas;
            this.kaina = kaina;
        }

        public override string ToString()
        {
            string eilute;
            eilute = string.Format(" {0, -16} {1, -20} {2, 5:f}",
                pavadinimas, tipas, kaina);
            return eilute;
        }

        public int CompareTo(Preke kita)
        {
            int poz = String.Compare(this.pavadinimas, kita.pavadinimas,
                StringComparison.CurrentCulture);
            if ((this.kaina < kita.kaina) ||
                ((this.kaina == kita.kaina) && (poz > 0)))
                return 1;
            else
                return -1;
        }
    }
}
```

- Surikiuokite naująjį sąrašą kainų mažėjimo ir alfabetine tvarka. Raskite šio sąrašo prekių kiekio vidurkį.

```
static void Main(string[] args)
{
    if (File.Exists(CFr))
        File.Delete(CFr);
    List<Prekes> PrekiuList = new List<Prekes>();
    Skaityti(CFd1, PrekiuList);
    Spausdinti(CFr, PrekiuList, "Pradinis sąrašas");
}
```

```

List<Prekes> NaujasPrekiuList = new List<Prekes>(PrekiuList);
Console.WriteLine("Įveskite prekės tipą");
string prekėsTipas = Console.ReadLine();
NaujasPrekiuList.RemoveAll(item => item.tipas == prekėsTipas);
if (NaujasPrekiuList.Count() > 0)
{
    Spausdinti(CFr, NaujasPrekiuList, "Suformuotas sąrašas");
    NaujasPrekiuList.Sort();
    Spausdinti(CFr, NaujasPrekiuList, "Rikiuotas suformuotas sąrašas");
    using (var fr = File.AppendText(CFr))
    {
        double vid = NaujasPrekiuList.Average(item => item.kiek);
        fr.WriteLine("Kiekio vidurkis = {0,5:f}", vid);
    }
}
else
    using (var fr = File.AppendText(CFr))
    {
        fr.WriteLine("Naujas sąrašas tuščias");
    }
}

```

- Išbandykite programą. Pasitikrinkite atsakymus.

### 🕒 Ketvirtas žingsnis.

- Klasę `Preke` paverskite abstrakčia klase. Joje aprašykite abstraktų metodą `Suma()`.

```

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

namespace Pratybos3
{
    abstract public class Preke : IComparable<Preke>
    {
        public string pavadinimas { get; set; } // prekės pavadinimas
        public string tipas { get; set; } // tipo pavadinimas (maisto, drabužiai,...)
        public double kaina { get; set; } // piniginė vertė
        /** konstruktorius be parametrų */
        public Preke()
        {
        }

        /** konstruktorius su parametrais */
        public Preke(string pavadinimas = "", string tipas = "", double kaina = 0.0)
        {
            this.pavadinimas = pavadinimas;
            this.tipas = tipas;
            this.kaina = kaina;
        }

        /** abstraktus metodas. Bus realizuotas išvestinėje klasėje */
        abstract public double Suma();

        public override string ToString()
        {
            string eilute;
            eilute = string.Format(" {0, -16} {1, -20} {2, 5:f}",
                pavadinimas, tipas, kaina);
            return eilute;
        }
    }
}

```

```

    }

    public int CompareTo(Preke kita)
    {
        int poz = String.Compare(this.pavadinimas, kita.pavadinimas,
                                   StringComparison.CurrentCulture);
        if ((this.kaina < kita.kaina) ||
            ((this.kaina == kita.kaina) && (poz > 0)))
            return 1;
        else
            return -1;
    }
}
}

```

- Klasėje `Prekės` realizuokite abstraktų metodą `Suma()` vienos prekės sumai (kiekis \* kaina) rasti.

```

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

namespace Pratybos3
{
    public class Prekes : Preke
    {
        public int kiek { get; set; } // turimas kiekis

        /** konstruktorius be parametru */
        public Prekes()
        {
        }

        /** konstruktorius su parametrais */
        public Prekes(string pav = "", string tipas = "", double kaina = 0.0, int kiek = 0)
            : base(pav, tipas, kaina)
        {
            this.kiek = kiek;
        }

        public override string ToString()
        {
            string eilute;
            eilute = string.Format("{0} {1,6:d}", base.ToString(), kiek);
            return eilute;
        }

        /** prekės sumos skaičiavimas */
        public override double Suma()
        {
            return kiek * kaina;
        }
    }
}

```

- Pagrindinėje klasėje `Program` užrašykite metodą `Sum()` viso dinaminio rinkinio prekių sumai rasti. Pagrindiniame metode `Main()` raskite naujojo sąrašo prekių visą sumą:



```

using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Pratybos3
{
    class Program
    {
        const string CFd1 = "..\\..\\Prekes.txt";
        const string CFr = "..\\..\\Rezultatai.txt";
        static void Main(string[] args)
        {
            if (File.Exists(CFr))
                File.Delete(CFr);
            List<Prekes> PrekiuList = new List<Prekes>();
            Skaityti(CFd1, PrekiuList);
            Spausdinti(CFr, PrekiuList, "Pradinis sąrašas");

            List<Prekes> NaujasPrekiuList = new List<Prekes>(PrekiuList);
            Console.WriteLine("Įveskite prekės tipą");
            string prekėsTipas = Console.ReadLine();
            NaujasPrekiuList.RemoveAll(item => item.tipas == prekėsTipas);
            if (NaujasPrekiuList.Count() > 0)
            {
                Spausdinti(CFr, NaujasPrekiuList, "Suformuotas sąrašas");
                NaujasPrekiuList.Sort();
                Spausdinti(CFr, NaujasPrekiuList, "Rikiuotas suformuotas sąrašas");
                using (var fr = File.AppendText(CFr))
                {
                    double vid = NaujasPrekiuList.Average(item => item.kiek);
                    fr.WriteLine("Kiekio vidurkis = {0,5:f}", vid);
                    double prekiuSuma = Sum(NaujasPrekiuList);
                    fr.WriteLine("Naujo sąrašo prekių visa suma = {0,5:f}", prekiuSuma);
                }
            }
            else
            {
                using (var fr = File.AppendText(CFr))
                {
                    fr.WriteLine("Naujas sąrašas tuščias");
                }
            }
        }

        //-----
        /** Skaitomi duomenys iš failo
        @param fv duomenų failo vardas
        @param PrekiuList - objektų rinkinys prekių duomenims saugoti */
        static void Skaityti(string fv, List<Prekes> PrekiuList)
        {
            using (StreamReader reader = new StreamReader(fv, Encoding.GetEncoding(1257)))
            {
                string line;
                while ((line = reader.ReadLine()) != null)
                {
                    string[] parts = line.Split(';');
                    //int ind1 = int.Parse(parts[0]);
                    string pav = parts[0].Trim();
                    string tema = parts[1].Trim();
                    double kaina = double.Parse(parts[2]);
                    int kiek = int.Parse(parts[3]);
                    Prekes pr = new Prekes(pav, tema, kaina, kiek);
                }
            }
        }
    }
}

```

```

        PrekiuList.Add(pr);
    }
}

//-----
/** Spausdinami duomenys
@param fv - rezultatu failo vardas
@param PrekiuList - objektų rinkinys prekių duomenims saugoti
@param info - lentelės pavadinimas */
static void Spausdinti(string fv, List<Prekes> PrekiuList, string info)
{
    const string virsus =
        "-----\r\n"
        + " Pavadinimas          Tipas          Kaina    Kiekis \r\n"
        + "-----";
    using (var fr = File.AppendText(fv))
    {
        fr.WriteLine(info);
        fr.WriteLine(virsus);
        for (int i = 0; i < PrekiuList.Count; i++)
        {
            Prekes pr = PrekiuList[i];
            fr.WriteLine("{0}", pr);
        }
        fr.WriteLine("-----" +
            "-----\r\n");
    }
}

//-----
/** Skaičiuojama visa suma
@param PrekiuList - objektų rinkinys prekių duomenims saugoti */
static double Sum(List<Prekes> PrekiuList)
{
    double su = 0;
    for (int i = 0; i < PrekiuList.Count; i++)
        su = su + PrekiuList[i].Suma();
    return su;
}
}
}

```

- Išbandykite programą. Pasitikrinkite atsakymus.

### Savarankiško darbo užduotis.

1. Sukurkite antrą tokios pačios struktūros duomenų failą "Prekes1.txt" .
2. Pagrindiniame metode Main() aprašykite dar vieną objektų dinaminį masyvą (List) papildomiems duomenims saugoti. Įveskite duomenis. Atspausdinkite sąrašą.
3. Sudarykite naują metodą papildomiems duomenims įterpti į surikiuotą sąrašą. Papildomų duomenų sąrašo išgadinti negalima. Atspausdinkite sąrašą.
4. Sudarykite abstraktų metodą kainai padidinti 10%.
5. Sudarykite pagrindinės klasės metodą visoms sąrašo kainoms padidinti.
6. Visų trijų sąrašų kainas padidinkite. Juos atspausdinkite.