KAUNO TECHNOLOGIJOS UNIVERSITETAS
Informatikos fakultetas

OBJEKTINIO PROGRAMAVIMO PAGRINDAI I (P175B117)

Laboratorinių darbų ataskaita

Atliko:
IFIN1/2 gr. studentas
Martynas Burneika
2021 m. gruodžio 23 d.
Priėmė:
Lektorė, Sturienė Rima

KAUNAS 2021

TURINYS

1. Pažintis su klase

1.1. Darbo užduotis

U2–2. Liftas

• Sukurkite klasę Studentas, kuri turėtų kintamuosius amžiui ir ūgiui saugoti. Trys studentai nutarė treniruotis žaisti krepšinį. Raskite, koks aukščiausio studento amžius ir koks jauniausio studento ūgis.

• Papildykite klasę Studentas kintamuoju, skirtu studento svoriui saugoti. Sukurkite klasę Liftas, kuri turėtų kintamuosius lifto keliamosios galios reikšmei ir talpai saugoti. Per kelis kartus visi studentai pakils liftu į reikiamą aukštą?

• Papildykite klasę Liftas metodais Dėti(), kurie leistų keisti lifto keliamąją galią ir talpą. Ar visi studentai vienu metu bus pakelti į reikiamą aukštį, jeigu lifto keliamoji galia bus padvigubinta? O jeigu talpa bus padvigubinta?

1.2. Programos tekstas

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

namespace Laboras_1____
{/// <summary>
/// Ši klasė skirta duomenims apie studentą
/// </summary>
    class Studentas

    {
        private double ūgis, amžius, svoris; //studento duomenys
        public Studentas(double ūgis, double amžius, double svoris)
        {
            this.ūgis = ūgis;
            this.amžius = amžius;
            this.svoris = svoris;
        }
        public double Imtiūgį() { return ūgis; }
        public double Imtiamžių() { return amžius; }
        public double ImtiSvorį() { return svoris; }
    }
    /// <summary>
    /// Ši klasė skirta duomenim apie liftą.
    /// </summary>
class Liftas
    {
        private double galia, talpa;
        public Liftas(double galia, double talpa)
        {
            this.galia = galia;
            this.talpa = talpa;

        }
        public double Imtigalią() { return galia; }
        public double Imtitalpą() { return talpa; }
        public void Dėtigalią(double galia) { this.galia = galia; }
        public void Dėtitalpą(double talpa) { this.talpa = talpa; }
    }
class Program
    {
        static void Main(string[] args)
        {
            Studentas s1, s2, s3;
            double u1, a1, kg1;
```

```
        // Pirmo studento duomenų įvedimas
        Console.WriteLine("Įveskite 1-ojo studento ūgį centimetrais:");
        u1 = double.Parse(Console.ReadLine());
        Console.WriteLine("Įveskite 1-ojo studento amžių metais:");
        a1 = double.Parse(Console.ReadLine());
        Console.WriteLine("Įveskite 1-ojo studento svorį kilogramais:");
        kg1 = double.Parse(Console.ReadLine());
        //Pradiniai studentų duomenys
        s1 = new Studentas(u1, a1, kg1);
        s2 = new Studentas(180, 24, 60);
        s3 = new Studentas(190, 23, 80);
        //Lentelė pradiniams studentų duomenims įšvesti
        SpausdLentele(s1, s2, s3);
        //Tikrina ar yra studentų su vienodais ūgiais
        if (s1.Imtiūgį() == s3.Imtiūgį())
            Spausdinti2(s1, s3);
        else if (s2.Imtiūgį() == s3.Imtiūgį())
            Console.Write("Aukščiausi studentai yra 2 ir 3.\n Jų ūgis yra:"
           + "{0}\n" + " antram studentui yra:" + "{1} metų\n"
          + " trečiam studentui yra:" + "{2}"
           + " metų\n", s2.Imtiūgį(), s2.Imtiamžių(), s3.Imtiamžių());
        else if (s1.Imtiūgį() == s2.Imtiūgį())
            Console.Write("Aukščiausi studentai yra 1 ir 2.\n Jų ūgis yra:"
           + "{0}\n" + " pirmam studentui yra:"
           + "{1} metų\n" + " antram studentui yra:" + "{2}"
           + " metų\n", s1.Imtiūgį(), s1.Imtiamžių(), s2.Imtiamžių());
        else
            Console.WriteLine("aukščiausio studento ūgis(cm)=" + "{0}\n"
                + " jo amžius =" +
                "{1}", SkaičAmžių(s1, s2, s3),
                Skaičiavimas((SkaičAmžių(s1, s2, s3)), s1, s2, s3));
        //Tikrina ar yra studentų su vienodu amžiumi
        if (s1.Imtiamžių() == s3.Imtiamžių())
            Spausdinti1(s1, s3);
        else if (s1.Imtiamžių() == s2.Imtiamžių())
            Console.WriteLine("Jauniausi studentai yra 1 ir 2.\n Jų amžius yra:"
                + "{0}" + "\n pirmo studento ūgis yra:" + "{1} "
           + "\n antro studento ūgis yra:" +
                "{2} \n", s1.Imtiamžių(), s1.Imtiūgį(), s2.Imtiūgį());
        else if (s2.Imtiamžių() == s3.Imtiamžių())
            Console.WriteLine("Jauniausi studentai yra 2 ir 3.\n Jų amžius yra:"
                + "{0}" + "\n antro studento ūgis yra:" + "{1} "
           + "\n trečio studento ūgis yra:" +
                "{2} \n", s2.Imtiamžių(), s2.Imtiūgį(), s3.Imtiūgį());
        else
            Console.WriteLine("jauniausiam studentui yra =" + "{0}\n" + "jo ūgis(cm) ="
                + "{1}", SkaičŪgį(s1, s2, s3),
                IfŪgis((SkaičŪgį(s1, s2, s3)), s1, s2, s3));
        Liftas l1;
        //Pradiniai lifto duomenys
        l1 = new Liftas(200, 2);
        //Lentelė pradiniams lifto duomenims įšvesti
        SpausdLentele2(l1);
        //Spausdina per kiek kartu liftas pakels studentus
        Console.WriteLine("liftas studentus pakels per: " + "{0}"
            + " kartus", ELiftas(l1, s1, s2, s3));
        l1.Dėtigalią(l1.Imtigalią() * 2);
        //Spausdina per kiek kartu liftas pakels studentus jei galia 2x
        Console.WriteLine("Jei galia 2x liftas studentus pakels per: "
            + "{0}" + " kartus", ELiftas(l1, s1, s2, s3));
        l1.Dėtigalią(l1.Imtigalią() / 2);
        l1.Dėtitalpą(l1.Imtitalpą() * 2);
        //Spausdina per kiek kartu liftas pakels studentus jei talpa 2x
        Console.WriteLine("Jei talpa 2x liftas studentus pakels per: "
            + "{0}" + " kartus", ELiftasAntras(l1, s1, s2, s3));
    }
```

```
/// <summary>
/// Suranda aukščiausia studentą
/// </summary>
/// <param name="s1"> pirmas studentas </param>
/// <param name="s2">antras studentas</param>
/// <param name="s3">trečias studentas</param>
/// <returns>Gražina aukščiausią studento ūgį</returns>
public static double SkaičAmžių(Studentas s1, Studentas s2, Studentas s3)
{
    double a;
    a = Math.Max(Math.Max(s1.Imtiūgį(), s2.Imtiūgį()), s3.Imtiūgį());
    return a;

}
/// <summary>
/// Suranda aukščiausio studento amžių
/// </summary>
/// <param name="s1"> pirmas studentas </param>
/// <param name="s2">antras studentas</param>
/// <param name="s3">trečias studentas</param>
/// <param name="SkaičAmžių">metodas surandantis aukščiausią studentą</param>
/// <returns>Gražina aukščiausio studento amžių</returns>
public static double Skaičiavimas(double SkaičAmžių, Studentas s1,
Studentas s2, Studentas s3)
{ double ats1;
    if (SkaičAmžių == s3.Imtiūgį())
        ats1 = s3.Imtiamžių();
    else if (SkaičAmžių == s2.Imtiūgį())
        ats1 = s2.Imtiamžių();
    else
        ats1 = s1.Imtiamžių();
    return ats1;
}
/// <summary>
/// Suranda jauniausią studentą
/// </summary>
/// <param name="s1"> pirmas studentas </param>
/// <param name="s2">antras studentas</param>
/// <param name="s3">trečias studentas</param>
/// <returns>Gražina jauniausio studento amžių</returns>
public static double SkaičŪgį(Studentas s1, Studentas s2, Studentas s3)
{
    double b;
    b = Math.Min(Math.Min(s1.Imtiamžių(), s2.Imtiamžių()), s3.Imtiamžių());
    return b;
}
/// <summary>
/// Suranda jauniausio studento ūgį
/// </summary>
/// <param name="s1"> pirmas studentas </param>
/// <param name="s2">antras studentas</param>
/// <param name="s3">trečias studentas</param>
/// <param name="SkaičŪgį">metodas surandantis jauniausią studentą</param>
/// <returns>Gražina jauniausio studento ūgį</returns>
public static double IfŪgis(double SkaičŪgį, Studentas s1, Studentas s2, Studentas s3)
{
    double ats2;
    if (SkaičŪgį == s2.Imtiamžių())
        ats2 = s2.Imtiūgį();
    else if (SkaičŪgį == s3.Imtiamžių())
        ats2 = s3.Imtiūgį();
    else
        ats2 = s1.Imtiūgį();
    return ats2;
}
```

```
/// <summary>
/// Suranda per kiek kartų liftas pakels studentus
/// </summary>
/// <param name="s1"> pirmas studentas </param>
/// <param name="s2">antras studentas</param>
/// <param name="s3">trečias studentas</param>
/// <param name="l1">liftas</param>
/// <returns>Gražina jauniausio studento ūgį</returns>
public static double ELiftas(Liftas l1, Studentas s1, Studentas s2, Studentas s3)
{
    double Kartai;
    if (l1.Imtitalpą() >= 3 && l1.Imtigalią() >= (s1.ImtiSvorį() +
       s2.ImtiSvorį() + s3.ImtiSvorį()))
        Kartai = 1;
    else if (((l1.Imtigalią() < (s1.ImtiSvorį() + s2.ImtiSvorį() + s3.ImtiSvorį()))
        && (l1.Imtigalią() >= (s1.ImtiSvorį() + s2.ImtiSvorį())) ||
             (l1.Imtigalią() >= (s1.ImtiSvorį() + s3.ImtiSvorį())) ||
             (l1.Imtigalią() >= (s2.ImtiSvorį() + s3.ImtiSvorį()))) &&
l1.Imtitalpą() >= 3)
        Kartai = 2;
    else if (l1.Imtitalpą() >= 3 && (l1.Imtigalią() <= (s1.ImtiSvorį()
       + s2.ImtiSvorį()) ||
            l1.Imtigalią() <= (s1.ImtiSvorį() + s3.ImtiSvorį()) ||
            l1.Imtigalią() <= (s2.ImtiSvorį() + s3.ImtiSvorį())))
        Kartai = 3;
    else if (((l1.Imtigalią() < (s1.ImtiSvorį() + s2.ImtiSvorį() + s3.ImtiSvorį()))
        && (l1.Imtigalią() >= (s1.ImtiSvorį() + s2.ImtiSvorį())) ||
             (l1.Imtigalią() >= (s1.ImtiSvorį() + s3.ImtiSvorį())) ||
             (l1.Imtigalią() >= (s2.ImtiSvorį() + s3.ImtiSvorį()))) &&
       l1.Imtitalpą() == 2)
        Kartai = 2;
    else if (l1.Imtitalpą() == 1 && (l1.Imtigalią() >= s1.ImtiSvorį())
        || (l1.Imtigalią() >= s2.ImtiSvorį()) || (l1.Imtigalią() >= s3.ImtiSvorį()))
        Kartai = 3;
    else
        Kartai = 3;
    return Kartai;
}
/// <summary>
/// Metodas duomenims išvesti jei jauniausi studentai yra 1 ir 3
/// </summary>
/// <param name="s1"> pirmas studentas </param>
/// <param name="s3">antras studentas</param>
public static void Spausdinti1(Studentas s1, Studentas s3)
{
    Console.Write("Jauniausi studentai yra 1 ir 3.\n Jų amžius yra:"
            + "{0}" + "\n pirmo studento ūgis yra:"
      + "{1} " + "\n trecio studento ūgis yra:" +
            "{2} \n", s1.Imtiamžių(), s1.Imtiūgį(), s3.Imtiūgį());
}
 /// <summary>
 /// Metodas duomenims išvesti jei aukščiausi studentai yra 1 ir 3
 /// </summary>
 /// <param name="s1"> pirmas studentas </param>
 /// <param name="s3">antras studentas</param>
public static void Spausdinti2(Studentas s1, Studentas s3)
{
    Console.Write("Aukščiausi studentai yra 1 ir 3.\n Jų ūgis yra:"
       + "{0}\n" + " pirmam studentui yra:" + "{1} metų\n"
      +  " trečiam studentui yra:" + "{2}"
       + " metų\n", s1.Imtiūgį(), s1.Imtiamžių(), s3.Imtiamžių());
}
```

```
        /// <summary>
        /// metodas lentelei spausdinti
        /// </summary>
        /// <param name="s1"> pirmas studentas </param>
        /// <param name="s2">antras studentas</param>
        /// <param name="s3">trečias studentas</param>
       public static void SpausdLentele(Studentas s1, Studentas s2, Studentas s3)
        {
            Console.WriteLine("-------------------------------------------------------------");
            Console.WriteLine("I    X     I  1studentas  I  2studentas  I    3studentas I");
            Console.WriteLine("-------------------------------------------------------------");
            Console.WriteLine("I Ūgis     I     {0}      I     {1}      I      {2}      I",
                s1.Imtiūgį(), s2.Imtiūgį(), s3.Imtiūgį());
            Console.WriteLine("-------------------------------------------------------------");
            Console.WriteLine("I Amžius   I     {0}       I     {1}       I      {2}       I",
                s1.Imtiamžių(), s2.Imtiamžių(), s3.Imtiamžių());
            Console.WriteLine("-------------------------------------------------------------");
            Console.WriteLine("I Svoris   I     {0}       I     {1}       I      {2}       I",
                s1.ImtiSvorį(), s2.ImtiSvorį(), s3.ImtiSvorį());
            Console.WriteLine("-------------------------------------------------------------");
        }
        /// <summary>
        /// metodas lentelei spausdinti
        /// </summary>
        /// <param name="l1"> liftas </param>
       public static void SpausdLentele2(Liftas l1)
        {
            Console.WriteLine("-------------------------------");
            Console.WriteLine("I    X   I galia   I   talpa  I");
            Console.WriteLine("-------------------------------");
            Console.WriteLine("I liftas I   {0}   I   {1}      I", l1.Imtigalią(),
                l1.Imtitalpą());
            Console.WriteLine("-------------------------------");
        }

    }

}
```

1.3. Pradiniai duomenys ir rezultatai

Testas nr. 1

```
Iveskite 1-ojo studentio ugi centimetrais:
191
Iveskite 1-ojo studento amziu metais:
24
Iveskite 1-ojo studento svori kilogramais:
80
-------------------------------------------------------------
I    X     I  1studentas  I  2studentas  I    3studentas I
-------------------------------------------------------------
I Ugis     I     191      I     180      I      190      I
-------------------------------------------------------------
I Amzius   I     24       I     24       I      23       I
-------------------------------------------------------------
I Svoris   I     80       I     70       I      80       I
-------------------------------------------------------------
auksciausio studento ugis(cm)=191
 jo amzius =24
Jauniausi studentai yra 1 ir 2.
 Ju amzius yra:24
 pirmo studento ugis yra:191
 antro studento ugis yra:180

-------------------------------
I    X   I galia   I   talpa  I
-------------------------------
I liftas I   200   I   3      I
```

```
------------------------------
liftas studentus pakels per: 2 kartus
Jei galia 2x liftas studentus pakels per: 1 kartus
Jei talpa 2x liftas studentus pakels per: 2 kartus
Press any key to continue . . .
```

Testas nr. 2

```
Iveskite 1-ojo studento ugi centimetrais:
178
Iveskite 1-ojo studento amziu metais:
23
Iveskite 1-ojo studento svori kilogramais:
84
------------------------------------------------------------
I    X     I  1studentas  I  2studentas  I    3studentas I
------------------------------------------------------------
I Ugis     I     178      I     200      I      175      I
------------------------------------------------------------
I Amzius   I     23       I     26       I      24       I
------------------------------------------------------------
I Svoris   I     84       I     92       I      70       I
------------------------------------------------------------
auksciausio studento ugis(cm)=200
 jo amzius =26
jauniausiam studentui yra =23
jo ugis(cm) =178
------------------------------
I    X   I galia   I   talpa  I
------------------------------
I liftas I   300   I   2      I
------------------------------
liftas studentus pakels per: 2 kartus
Jei galia 2x liftas studentus pakels per: 2 kartus
Jei talpa 2x liftas studentus pakels per: 1 kartus
Press any key to continue . . .
```

Testas nr. 3

```
Iveskite 1-ojo studento ugi centimetrais:
180
Iveskite 1-ojo studento amziu metais:
21
Iveskite 1-ojo studento svori kilogramais:
88
------------------------------------------------------------
I    X     I  1studentas  I  2studentas  I    3studentas I
------------------------------------------------------------
I Ugis     I     180      I     168      I      180      I
------------------------------------------------------------
I Amzius   I     21       I     21       I      25       I
------------------------------------------------------------
I Svoris   I     88       I     69       I      80       I
------------------------------------------------------------
Auksciausi studentai yra 1 ir 3.
 Ju ugis yra:180
 pirmam studentui yra:21 metu
 treciam studentui yra:25 metu
Jauniausi studentai yra 1 ir 2.
 Ju amzius yra:21
 pirmo studento ugis yra:180
 antro studento ugis yra:168

------------------------------
I    X   I galia   I   talpa  I
------------------------------
I liftas I   170   I   4      I
------------------------------
liftas studentus pakels per: 2 kartus
Jei galia 2x liftas studentus pakels per: 1 kartus
Jei talpa 2x liftas studentus pakels per: 2 kartus
Press any key to continue . . .
```

1.4. Dėstytojo pastabos

Pradiniu duomenų spausdinimas ne metode (pataisyta)

Patikslinti skaičiavimus (pataisyta)

2. Objektų rinkinys

2.1. Darbo užduotis

U3–2. Krepšinis

• Krepšinio mokykloje treniruotes lankančių sąrašas yra tekstiniame faile: būsimo krepšininko vardas ir pavardė, amžius ir ūgis. Pirmoje eilutėje yra krepšinio mokyklos pavadinimas. Sukurkite klasę Krepšininkas, kuri turėtų kintamuosius vardui su pavarde, amžiui bei ūgiui saugoti. Raskite, koks būsimų krepšininkų amžiaus vidurkis ir koks ūgio vidurkis.

• Papildykite programą veiksmais su dviejų krepšinio mokyklų duomenimis. Kiekvienos mokyklos duomenys saugomi atskiruose failuose. Kurioje mokykloje aukščiausias sportininkas? Surašykite į atskirą rinkinį visus abiejų mokyklų sportininkus, kurių ūgis didesnis už vidurkį.

2.2. Programos tekstas

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

namespace Laboras2
/// <summary>
/// Ši klasė skirta duomenims apie krepšininkus
/// </summary>
{class krepsininkas
    {   //krepsininko duomenys
        private string vardas;
        private string pavarde;
        private int amzius;
        private int ugis;
        public krepsininkas(string vardas, string pavarde, int amzius, int ugis)
        {
            this.vardas = vardas;
            this.pavarde = pavarde;
            this.amzius = amzius;
            this.ugis = ugis;
        }
        public string ImtiVarda() { return vardas; }
        public string ImtiPavarde() { return pavarde; }
        public int ImtiAmziu() { return amzius;}
        public int ImtiUgi() { return ugis; }

    }
class Program
    {
          // konstantos
        const int Cn = 100;
        const string CFd1 = "Duom6.txt";
        const string CFd2 = "Duom7.txt";
        const string CFrez = "..\\..\\Rez.txt";

        static void Main(string[] args)
        {   //ištrina rezultatų failą jeigu egzistuoja
            if (File.Exists(CFrez))
                File.Delete(CFrez);
            //kuriami nuskaitomi ir spausdinami masyvai
            krepsininkas[] K1 = new krepsininkas[Cn];
            int nkiek1;
            string pav1;
            Skaityti(CFd1, K1, out nkiek1, out pav1);
            Spausdinti(CFrez, K1, nkiek1, pav1);
```

```
            krepsininkas[] K2 = new krepsininkas[Cn];
            int nkiek2;
            string pav2;
            Skaityti(CFd2, K2, out nkiek2, out pav2);
            Spausdinti(CFrez, K2, nkiek2, pav2);
            SpausdintiRezultatus(CFrez, K1, pav1, nkiek1)
            SpausdintiRezultatus(CFrez, K2, pav2, nkiek2);
              //Tikrina ar nera vienodo ugio krepsininku
            using (var fr = File.AppendText(CFrez))
            {
              if (K1[Auksciausias(K1, nkiek1)].ImtiUgi() <
                  K2[Auksciausias(K2, nkiek2)].ImtiUgi())
                  fr.WriteLine("Auksciausias sportininkas yra ''{0}'' mokykloje", pav2);
              else if (K1[Auksciausias(K1, nkiek1)].ImtiUgi() ==
                 K2[Auksciausias(K2, nkiek2)].ImtiUgi())
                 fr.WriteLine("Auksciausi sportininkai yra ir mokyklose ''{0}'' ir ''{1}''  ",
                 pav1, pav2);
              else
                 fr.WriteLine("\nAuksciausias sportininkas yra ''{0}'' mokykloje", pav1);
            }
            krepsininkas[] K3 = new krepsininkas[Cn];
            int kiek2 = 0;
            Formuoti(K1, nkiek1, K3, ref kiek2, UgioVidurkis(K1, nkiek1));
            Formuoti(K2, nkiek2, K3, ref kiek2, UgioVidurkis(K2, nkiek2));
              //Tikrinama ar suformuotas masyvas egzistuoja
            if (kiek2 > 0)
                Spausdinti(CFrez, K3, kiek2, "\nSportininkai kuriu ugis didesnis uz vidurki");
            else
                using (var fr = File.AppendText(CFrez))
                    fr.WriteLine("\nRinkinys neformuojamas");
        }
        /// <summary>
        /// Nuskaito duomenų failus
        /// </summary>
        /// <param name="K">Masyvas į kurį skaitys</param>
        /// <param name="kiek">kintamasis kuris išsaugo masyvo ilgio reikšmę</param>
        /// <param name="fv">Duomenų failo pavadinimas</param>
static void Skaityti(string fv, krepsininkas[] K, out int kiek, out string pav)
        {
            using (StreamReader reader = new StreamReader(fv))
            {
                string vardas;
                string pavarde;
                int amzius;
                int ugis;
                string line;
                line = reader.ReadLine();
                string[] parts;
                pav = line;
                int i = 0;
                while ((line = reader.ReadLine()) != null && (i < Cn))
                {
                    parts = line.Split(';');
                    vardas = (parts[0]);
                    pavarde = (parts[1]);
                    amzius = int.Parse(parts[2]);
                    ugis = int.Parse(parts[3]);
                    K[i] = new krepsininkas(vardas, pavarde, amzius, ugis);
                    i++;
                }
                kiek = i;
            }
        }
```

```
        /// <summary>
        /// Spausdina pradinius duomenis
        /// </summary>
        /// <param name="K">Masyvas kuri spausdinsim</param>
        /// <param name="kiek">masyvo ilgis</param>
        /// <param name="fv">Rezultatų failo pavadinimas</param>
        /// <param name="pav">krepsinio mokyklos pavadinimas</param>
public static void Spausdinti(string fv, krepsininkas[]K, int kiek, string pav)
        {
                const string virsus =
                          "|-----------------|--------------|--------------|---------|\r\n"
                        + "|     Vardas      |    Pavarde   |     Amzius   |   Ugis  | \r\n"
                        + "|-----------------|--------------|--------------|---------|";
            using (var fr = File.AppendText(fv))
            {
                fr.WriteLine("  {0}", pav);
                fr.WriteLine(virsus);
                krepsininkas tarp;
                for (int i = 0; i < kiek; i++)
                {
                     tarp = K[i];
                fr.WriteLine("|{0,-17}|{1,-15}|{2,-15}|{3,-9}|",
                                tarp.ImtiVarda(), tarp.ImtiPavarde(),
                                tarp.ImtiAmziu(), tarp.ImtiUgi());
                fr.WriteLine("|-----------------|--------------|--------------|---------|");
                }
            }
        }
        /// <summary>
        /// Spausdina rezultatus
        /// </summary>
        /// <param name="fv">Rezultatų failo pavadinimas</param>
        /// <param name="K">Masyvas kuriame yra visi ilgiai</param>
        /// <param name="kiek">Masyvo ilgio kintamasis</param>
        /// <param name="pav">krepsinio mokyklos pavadinimas</param>
public static void SpausdintiRezultatus(string fv, krepsininkas[] K, string pav,
        int kiek)
        {
          using (var fr = File.AppendText(fv))
           {
        fr.WriteLine("\nSporto mokyklos ''{0}'' busimu krepsininku amziaus vidurkis: {1,6:f2}",
                    pav, AmziausVidurkis(K, kiek));
        fr.WriteLine("Sporto mokyklos ''{0}'' busimu krepsininku ugio vidurkis: {1,9:f2}",
                    pav, UgioVidurkis(K, kiek));
           }
        }
        /// <summary>
        /// Suranda krepsininku amziaus vidurki
        /// </summary>
        /// <param name="K">Masyvas kuriame yra visi ilgiai</param>
        /// <param name="kiek">Masyvo ilgio kintamasis</param>
public static double AmziausVidurkis(krepsininkas[]K, int kiek)
      {
            double suma = 0;
            for (int i = 0; i < kiek; i++)
                suma = suma + K[i].ImtiAmziu();
            if (kiek > 0)
                return suma / kiek;
            else return -1;
        }
        /// <summary>
        /// Suranda krepsininku vidurki
        /// </summary>
        /// <param name="K">Masyvas kuriame yra visi ilgiai</param>
        /// <param name="kiek">Masyvo ilgio kintamasis</param>
public static double UgioVidurkis(krepsininkas[] K, int kiek)
        {
```

```
            double suma = 0;
            for (int i = 0; i < kiek; i++)
                suma = suma + K[i].ImtiUgi();
            double a = suma / kiek;
            if (kiek > 0)
                return a;
            else return -1;
        }
        /// <summary>
        /// Suranda auksciausia krepsininka
        /// </summary>
        /// <param name="K">Masyvas kuriame yra visi ilgiai</param>
        /// <param name="kiek">Masyvo ilgio kintamasis</param>
 static int Auksciausias(krepsininkas[]K, int kiek)
        {
            int n = 0;
            for (int i = 0; i < kiek; i++)
                if (K[i].ImtiUgi() > K[n].ImtiUgi())
                    n = i;
            return n;
        }
        /// <summary>
        /// Formuoja nauja masyva
        /// </summary>
        /// <param name="K">Masyvas kuriame yra visi ilgiai</param>
        /// <param name="kiek">Masyvo ilgio kintamasis</param>
        /// <param name="K3">Naujas masyvas</param>
        /// <param name="kiek2">Naujo masyvo ilgio kintamasis</param>
        /// <param name="UgioVidurkis">Kreipinys i metoda</param>
 static void Formuoti(krepsininkas[] K, int kiek, krepsininkas[] K3, ref int kiek2,
        double UgioVidurkis)
        {
            for (int i = 0; i < kiek; i++)
            {
                if (K[i].ImtiUgi() > UgioVidurkis)
                {
                    K3[kiek2] = K[i];
                    kiek2++;
                }

            }
        }
    }
}
```

2.3. Pradiniai duomenys ir rezultatai

Testas nr.1

Tornadas

Vardas	Pavarde	Amzius	Ugis
Jonas	Jonaitis	18	190
Matas	Matutis	19	190
Rokas	Rokutis	20	190
Lukas	Lukenas	19	190
Titas	Titenas	17	190

```
  Žalgirirs
|-----------------|---------------|---------------|---------|
|     Vardas      |    Pavarde    |     Amzius    |   Ugis  |
|-----------------|---------------|---------------|---------|
|rytis            | rytenas       |20             |190      |
|-----------------|---------------|---------------|---------|
|nojus            | nojenas       |21             |190      |
|-----------------|---------------|---------------|---------|
|zenius           | zeniukas      |22             |190      |
|-----------------|---------------|---------------|---------|
|tadas            | tadenas       |19             |190      |
|-----------------|---------------|---------------|---------|
|rokas            | rukelis       |18             |190      |
|-----------------|---------------|---------------|---------|

Sporto mokyklos ''Tornadas'' busimu krepsininku amziaus vidurkis:  18.60
Sporto mokyklos ''Tornadas'' busimu krepsininku ugio vidurkis:    190.00

Sporto mokyklos ''Žalgirirs'' busimu krepsininku amziaus vidurkis:  20.00
Sporto mokyklos ''Žalgirirs'' busimu krepsininku ugio vidurkis:    190.00

Aksciausi sportininkai yra ir mokyklose ''Tornadas'' ir ''Žalgirirs''

Rinkinys neformuojamas
```

Testas nr. 2

```
  Tornadas
|-----------------|---------------|---------------|---------|
|     Vardas      |    Pavarde    |     Amzius    |   Ugis  |
|-----------------|---------------|---------------|---------|
|Jonas            | Jonaitis      |18             |200      |
|-----------------|---------------|---------------|---------|
|Matas            | Matutis       |19             |199      |
|-----------------|---------------|---------------|---------|
|Rokas            | Rokutis       |20             |191      |
|-----------------|---------------|---------------|---------|
|Lukas            | Lukenas       |19             |189      |
|-----------------|---------------|---------------|---------|
|Titas            | Titenas       |17             |195      |
|-----------------|---------------|---------------|---------|
  Žalgirirs
|-----------------|---------------|---------------|---------|
|     Vardas      |    Pavarde    |     Amzius    |   Ugis  |
|-----------------|---------------|---------------|---------|
|rytis            | rytenas       |20             |190      |
|-----------------|---------------|---------------|---------|
|nojus            | nojenas       |21             |195      |
|-----------------|---------------|---------------|---------|
|zenius           | zeniukas      |22             |196      |
|-----------------|---------------|---------------|---------|
|tadas            | tadenas       |19             |200      |
|-----------------|---------------|---------------|---------|
|rokas            | rukelis       |18             |210      |
|-----------------|---------------|---------------|---------|

Sporto mokyklos ''Tornadas'' busimu krepsininku amziaus vidurkis:  18.60
Sporto mokyklos ''Tornadas'' busimu krepsininku ugio vidurkis:    194.80

Sporto mokyklos ''Žalgirirs'' busimu krepsininku amziaus vidurkis:  20.00
Sporto mokyklos ''Žalgirirs'' busimu krepsininku ugio vidurkis:    198.20

Aksciausias sportininkas yra ''Žalgirirs'' mokykloje
```

Sportininkai kuriu ugis didesnis uz vidurki

Vardas	Pavarde	Amzius	Ugis
Jonas	Jonaitis	18	200
Matas	Matutis	19	199
Titas	Titenas	17	195
tadas	tadenas	19	200
rokas	rukelis	18	210

Testas nr.3

Tornadas

Vardas	Pavarde	Amzius	Ugis
Jonas	Jonaitis	18	210
Matas	Matutis	19	199
Rokas	Rokutis	20	191
Lukas	Lukenas	19	199
Titas	Titenas	17	194

Žalgirirs

Vardas	Pavarde	Amzius	Ugis
rytis	rytenas	20	190
nojus	nojenas	21	185
zenius	zeniukas	22	191
tadas	tadenas	19	202
rokas	rukelis	18	210

Sporto mokyklos ''Tornadas'' busimu krepsininku amziaus vidurkis: 18.60
Sporto mokyklos ''Tornadas'' busimu krepsininku ugio vidurkis: 198.60

Sporto mokyklos ''Žalgirirs'' busimu krepsininku amziaus vidurkis: 20.00
Sporto mokyklos ''Žalgirirs'' busimu krepsininku ugio vidurkis: 195.60

Auksciausi sportininkai yra ir mokyklose ''Tornadas'' ir ''Žalgirirs''

Sportininkai kuriu ugis didesnis uz vidurki

Vardas	Pavarde	Amzius	Ugis
Jonas	Jonaitis	18	210
Matas	Matutis	19	199
Lukas	Lukenas	19	199
tadas	tadenas	19	202
rokas	rukelis	18	210

2.4. Dėstytojo pastabos

3. Konteinerinė klasė

3.1. Darbo užduotis

U4-2. Mobiliojo ryšio kortelės Norėdamas palyginti mobiliojo ryšio operatorių siūlomas išankstinio mokėjimo korteles Sirvydas surinko šią informaciją į tekstinį failą.

Faile eilutėmis yra kortelių duomenys: kortelės (tinklo) pavadinimas, pradinė suma kortelėje, tarifas savame tinkle, tarifas į kitus tinklus, SMS žinučių tarifas savame tinkle ir į kitus tinklus.

Parašykite programą, kuri spausdintų kortelių duomenis lentele, surastų kortelę, kurios SMS žinučių tarifai į kitus tinklus mažiausi.

Papildykite programą veiksmais, kurie leistų atrinkti korteles, kurios leidžia skambinti ir siųsti SMS žinutes savame tinkle nemokamai, ir šį sąrašą surikiuoti pagal pradinę sumą mažėjimo tvarka ir kortelės pavadinimą abėcėliškai.

3.2. Programos tekstas

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

namespace P4lab
{
    /// <summary>
    /// Ši klasė skirta duomenims apie korteles
    /// </summary>
    class kortele
    {
        private double Suma, TarifSav,
            TarifKit, SmsTarifSav, SmsTarifKit;
        private string pav;

        public kortele(string pav ,double Suma,
            double TarifSav, double TarifKit,
            double SmsTarifSav, double SmsTarifKit)
        {
            this.pav = pav;
            this.Suma = Suma;
            this.TarifSav = TarifSav;
            this.TarifKit = TarifKit;
            this.SmsTarifSav = SmsTarifSav;
            this.SmsTarifKit = SmsTarifKit;
        }
        public override string ToString()
        {
            string eilute;
            eilute = string.Format("|{0,9} | {1,5:f} | {2,6:f} | {3,6:f} | {4,5:f}
              |{5,6:f}|",
                 pav, Suma, TarifSav, TarifKit, SmsTarifSav, SmsTarifKit);
            return eilute;
        }
        public double ImtiSmsTarifKit() { return SmsTarifKit; }
        public double ImtiTarifSav() { return TarifSav; }
        public double ImtiSmsTarifSav() { return SmsTarifSav; }

        /// <summary>
        /// Užklotas operatorius
        /// </summary>
        /// <param name="k1">kortele</param>
        /// <returns>grąžina palyginimą pagal SMS ir skambučių tarifus savame tinkle</returns>
```

```
    public static bool operator !(kortele k1)
    {
        if ((k1.ImtiTarifSav() == 0) && (k1.ImtiSmsTarifSav() == 0))
            return false;
        return true;
    }

    /// <summary>
    /// Užklotas operatorius
    /// </summary>
    /// <param name="kt1">pirma kortele</param>
    /// <param name="kt2"> antra kortele</param>
    /// <returns>grąžina palyginimą pagal pradinę sumą ir pavadinimą</returns>
    public static bool operator >= (kortele kt1, kortele kt2)
    {
        int p = string.Compare(kt1.pav, kt2.pav,
            StringComparison.CurrentCulture);
        if (kt1.Suma > kt2.Suma || kt1.Suma == kt2.Suma && p > 0)
            return true;
        return false;

    }
    public static bool operator <=(kortele kt1, kortele kt2)
    {
        int p = string.Compare(kt1.pav, kt2.pav,
            StringComparison.CurrentCulture);
        if (kt1.Suma > kt2.Suma || kt1.Suma == kt2.Suma && p < 0)
            return true;
        return false;
    }
}

/// <summary>
/// Ši klasė yra konteinerinė
/// </summary>
class korteleskontnr
{
    const int Cmax = 100;
    private kortele[] kort;
    private int n;

    public korteleskontnr()
    {
        n = 0;
        kort = new kortele[Cmax];
    }

    //sąsajos metodai
    public int Imti() { return n; }
    public kortele Imtikort(int i) { return kort[i]; }
    public void Deti(kortele ob) { kort[n++] = ob; }
    /// <summary>
    /// metodas skirtas surikiuoti masyvui
    /// </summary>
    public void Rikiuoti()
    {
        for(int i = 0; i < n-1; i++)
        {
            kortele min = kort[i];
            int im = i;
            for (int j =i; j<n; j++)
                if(kort[j] <= min)
                {
                    im = j;
                    min = kort[j];
                }
            kort[im] = kort[i];
            kort[i] = min;
```

```
        }
    }
}

class Program
{
    // konstantos
    const string CFd = "duom.txt";
    const string CFrez = "..//..//rez.txt";
    static void Main(string[] args)
    {
        //ištrina rezultatų failą jeigu egzistuoja
        if (File.Exists(CFrez))
            File.Delete(CFrez);

        korteleskontnr korteles = new korteleskontnr();
        skaityti(ref korteles, CFd);
        spausdinti(korteles, CFrez);
        korteleskontnr kortelesmin = new korteleskontnr();
        minkeli(korteles,ref kortelesmin);
        using (var fr = File.AppendText(CFrez))
        {
            fr.WriteLine("kortelė, kurios SMS žinučių tarifai į kitus tinklus mažiausi.");
            for (int i = 0; i < kortelesmin.Imti(); i++)
            {
                fr.WriteLine("|-----------------------------------------------------|");
                fr.WriteLine("{0}", kortelesmin.Imtikort(i).ToString());
            }
        }            korteleskontnr korteles1 = new korteleskontnr();
        formuoti(korteles, ref korteles1);
        korteles1.Rikiuoti();
        if (korteles1.Imti() > 0)
            spausdinti(korteles1, CFrez);
        else
            using (var fr = File.AppendText(CFrez))
                fr.WriteLine("\ntokiu korteliu nera");

    }

    /// <summary>
    /// Nuskaito duomenų failus
    /// </summary>
    /// <param name="korteles">konteineris</param>
    /// <param name="fv">Duomenų failo pavadinimas</param>
    static void skaityti(ref korteleskontnr korteles , string fv)
    {
        using (StreamReader reader = new StreamReader(fv))
        {
            string line;
            line = reader.ReadLine();
            string[] parts;
            while ((line = reader.ReadLine()) != null)
            {
                parts = line.Split(' ');
                string pav = parts[0];
                double suma = double.Parse(parts[1]);
                double tarifsav = double.Parse(parts[2]);
                double TarifKit = double.Parse(parts[3]);
                double SmsTarifSav = double.Parse(parts[4]);
                double SmsTarifKit = double.Parse(parts[5]);
                kortele kort = new kortele(pav, suma, tarifsav, TarifKit, SmsTarifSav,
            SmsTarifKit);
                korteles.Deti(kort);
            }
        }
    }

    /// <summary>
```

```
/// Spausdina duomenis lentele
/// </summary>
/// <param name="korteles">konteineris</param>
/// <param name="fv">Rezultatų failo pavadinimas</param>

static void spausdinti(korteleskontnr korteles, string fv)
{
    string virsus = "\n|-----------------------------------------------------|\r\n"
                   +"|              informacija apie korteles              |\r\n"
                   + "|-----------------------------------------------------|\r\n"
                   + "|    pav    | suma  | tar sav| tar kit|smstsav|smstkit|";
    using (var fr = File.AppendText(fv))
    {
        fr.WriteLine(virsus);
        for (int i = 0; i < korteles.Imti(); i++)
        {
            fr.WriteLine("|-----------------------------------------------------|");
            fr.WriteLine("{0}", korteles.Imtikort(i).ToString());
        }
        fr.WriteLine("|-----------------------------------------------------|\n");
    }
}

/// <summary>
/// Suranda kortele su mažiausiu SMS tarifu į kitus tinklus
/// </summary>
/// <param name="korteles">konteineris</param>
static double min(korteleskontnr korteles)
{
    double min =99999;
    for (int i = 1; i < korteles.Imti(); i++)
    {
        if(korteles.Imtikort(i).ImtiSmsTarifKit() < min)
            min = korteles.Imtikort(i).ImtiSmsTarifKit();
    }
    return min;
}

/// <summary>
/// Formuoja nauja konteineri
/// </summary>
/// <param name="A">senas konteineris</param>
/// <param name="B">Naujas konteineris</param>
static void formuoti(korteleskontnr A, ref korteleskontnr B)
{
    for (int i = 0; i < A.Imti(); i++)
        if (!A.Imtikort(i))
            ;
        else
            B.Deti(A.Imtikort(i));

}

/// <summary>
/// Formuoja nauja konteinerį su mažiausiu SMS tarifu į kitus tinklus kortelėmis
/// </summary>
/// <param name="k1">senas konteineris</param>
/// <param name="k2">Naujas konteineris</param>
static void minkeli(korteleskontnr k1,ref korteleskontnr k2)
{
    for (int i = 0; i < k1.Imti(); i++)
        if (k1.Imtikort(i).ImtiSmsTarifKit() == min(k1))
            k2.Deti(k1.Imtikort(i));
}



/// <summary>
```

```
        /// Spausdina rezultatus lentele
        /// </summary>
        /// <param name="kortelesmin">konteineris</param>
        /// <param name="fv">Rezultatų failo pavadinimas</param>
        static void spausdintiRez(korteleskontnr kortelesmin, string fv)
        {
            using (var fr = File.AppendText(CFrez))
            {
                fr.WriteLine("kortelė, kurios SMS žinučių tarifai į kitus tinklus mažiausi.");
                for (int i = 0; i < kortelesmin.Imti(); i++)
                {
                    fr.WriteLine("|-----------------------------------------------------|");
                    fr.WriteLine("{0}", kortelesmin.Imtikort(i).ToString());
                }
            }
        }
    }
}
```

3.3. Pradiniai duomenys ir rezultatai

Testas nr.1

```
|-----------------------------------------------------|
|              informacija apie korteles              |
|-----------------------------------------------------|
|    pav    | suma  | tar sav| tar kit|smstsav|smstkit|
|-----------------------------------------------------|
|    telia | 14.00 |   1.00 |   0.15 |  0.00 |   0.05|
|-----------------------------------------------------|
|   pildyk | 14.00 |   1.00 |   0.20 |  0.00 |   0.02|
|-----------------------------------------------------|
|     ezys | 12.00 |   1.00 |   0.12 |  0.00 |   0.02|
|-----------------------------------------------------|
|    tele2 | 12.00 |   2.00 |   0.12 |  0.00 |   0.02|
|-----------------------------------------------------|
|     bite | 13.00 |   1.00 |   0.10 |  0.00 |   0.04|
|-----------------------------------------------------|
| teledema | 11.00 |   1.00 |   0.10 |  0.00 |   0.06|
|-----------------------------------------------------|

kortelė, kurios SMS žinučių tarifai į kitus tinklus mažiausi.
|-----------------------------------------------------|
|   pildyk | 14.00 |   1.00 |   0.20 |  0.00 |   0.02|
|-----------------------------------------------------|
|     ezys | 12.00 |   1.00 |   0.12 |  0.00 |   0.02|
|-----------------------------------------------------|
|    tele2 | 12.00 |   2.00 |   0.12 |  0.00 |   0.02|

tokiu korteliu nera
```

Testas nr.2

```
|-------------------------------------------------|
|              informacija apie korteles          |
|-------------------------------------------------|
|    pav   | suma  | tar sav| tar kit|smstsav|smstkit|
|-------------------------------------------------|
|    telia | 14.00 |   0.00 |   0.15 |  0.00 |   0.05|
|-------------------------------------------------|
|   pildyk | 14.00 |   0.00 |   0.20 |  0.00 |   0.02|
|-------------------------------------------------|
|     ezys | 12.00 |   0.00 |   0.12 |  0.00 |   0.02|
|-------------------------------------------------|
|    tele2 | 12.00 |   2.00 |   0.12 |  0.00 |   0.02|
|-------------------------------------------------|
|     bite | 13.00 |   1.00 |   0.10 |  0.00 |   0.04|
|-------------------------------------------------|
| teledema | 11.00 |   0.00 |   0.10 |  0.00 |   0.06|
|-------------------------------------------------|

kortelė, kurios SMS žinučių tarifai į kitus tinklus mažiausi.
|-------------------------------------------------|
|   pildyk | 14.00 |   0.00 |   0.20 |  0.00 |   0.02|
|-------------------------------------------------|
|     ezys | 12.00 |   0.00 |   0.12 |  0.00 |   0.02|
|-------------------------------------------------|
|    tele2 | 12.00 |   2.00 |   0.12 |  0.00 |   0.02|

|-------------------------------------------------|
|              informacija apie korteles          |
|-------------------------------------------------|
|    pav   | suma  | tar sav| tar kit|smstsav|smstkit|
|-------------------------------------------------|
|   pildyk | 14.00 |   0.00 |   0.20 |  0.00 |   0.02|
|-------------------------------------------------|
|    telia | 14.00 |   0.00 |   0.15 |  0.00 |   0.05|
|-------------------------------------------------|
|     ezys | 12.00 |   0.00 |   0.12 |  0.00 |   0.02|
|-------------------------------------------------|
| teledema | 11.00 |   0.00 |   0.10 |  0.00 |   0.06|
|-------------------------------------------------|
```

Testas nr.2

```
|-------------------------------------------------|
|              informacija apie korteles          |
|-------------------------------------------------|
|    pav   | suma  | tar sav| tar kit|smstsav|smstkit|
|-------------------------------------------------|
|    telia | 14.00 |   0.00 |   0.15 |  0.00 |   0.05|
|-------------------------------------------------|
|   pildyk | 14.00 |   0.00 |   0.20 |  0.00 |   0.02|
|-------------------------------------------------|
|     ezys | 14.00 |   0.00 |   0.12 |  0.00 |   0.04|
|-------------------------------------------------|
|    tele2 | 12.00 |   2.00 |   0.12 |  0.00 |   0.04|
|-------------------------------------------------|
|     bite | 13.00 |   1.00 |   0.10 |  0.00 |   0.04|
|-------------------------------------------------|
| teledema | 14.00 |   0.00 |   0.10 |  0.00 |   0.06|
|-------------------------------------------------|




kortelė, kurios SMS žinučių tarifai į kitus tinklus mažiausi.
|-------------------------------------------------|
```

```
|   pildyk | 14.00 |   0.00 |   0.20 |  0.00 |   0.02|


|-----------------------------------------------------|
|              informacija apie korteles              |
|-----------------------------------------------------|
|    pav   | suma  | tar sav| tar kit|smstsav|smstkit|
|-----------------------------------------------------|
|     ezys | 14.00 |   0.00 |   0.12 |  0.00 |   0.04|
|-----------------------------------------------------|
|   pildyk | 14.00 |   0.00 |   0.20 |  0.00 |   0.02|
|-----------------------------------------------------|
| teledema | 14.00 |   0.00 |   0.10 |  0.00 |   0.06|
|-----------------------------------------------------|
|    telia | 14.00 |   0.00 |   0.15 |  0.00 |   0.05|
|-----------------------------------------------------|
```

3.4. Dėstytojo pastabos

4. Teksto analizė ir redagavimas

4.1. Darbo užduotis

U5-2.

- Nelyginis žodžių skaičius Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami.
- Žodžiai eilutėse skiriami bent vienu tarpu.
- Tarpai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės.
- Eilutėse, kuriose yra nelyginis žodžių skaičius n, n / 2 + 1 žodį pakeisti žodžiu „xxooxx“.

4.2. Programos tekstas

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

namespace p5lab
{
    class Program
    {
        //constants
        const string Cfd = "Text.txt";
        const string Cfrez = "..\\..\\Results.txt";
        const string Cfana = "..\\..\\Analysis.txt";

        static void Main(string[] args)
        {
            //deletes results file if it exists
            if (File.Exists(Cfrez))
                File.Delete(Cfrez);
            //deletes analysis file if it exists
            if (File.Exists(Cfana))
                File.Delete(Cfana);
            char[] separators = { ' ', '.', ',', '!', '?', ':', ';', '(', ')', '\t' };
            ReadWrite(Cfd, Cfrez, Cfana, separators);
        }
        /// <summary>
        /// Reads the data file and write in the results and analysis files
        /// </summary>
        /// <param name="fv">data file</param>
        /// <param name="rfv">results file</param>
        /// <param name="afv">analysis file</param>
        /// <param name="separators">separators</param>
        static void ReadWrite(string fv,string rfv,string afv, char[] separators)
        {
            int linecount = 0; // number of lines with odd numbers
            int n = 0; //number of words in a line
            string word = "";
            string line;
            using (var fra = File.CreateText(afv))
            {
               using (var fr = File.CreateText(rfv))
               {
                    using (StreamReader reader = new StreamReader(fv,
    Encoding.GetEncoding(1257)))
                    {
                      fr.WriteLine("-----------------------------------------------");
                        fr.WriteLine("Pradinis tekstas");
```

```
                fr.WriteLine("-----------------------------------------------");
                while ((line = reader.ReadLine()) != null)
                {
                fr.WriteLine(line);
                }
                fr.WriteLine("===============================================");
            }
            using (StreamReader reader = new StreamReader(fv,
        Encoding.GetEncoding(1257)))
            {
               fr.WriteLine("Tekstas po redagavimo");
               fr.WriteLine("-----------------------------------------------");
               while ((line = reader.ReadLine()) != null)
               {
                  string[] parts = line.Split(separators,
        StringSplitOptions.RemoveEmptyEntries);
                  WordsCount(parts, line, separators, out n);
                  //checks if lines have even or odd number of words
                  if (n % 2 !=0)
                  {
                        linecount++;
                        WordFinder(line, separators, parts, n, out word);
                        fra.WriteLine("{0} ------> XXOOXX", word);
                        fr.WriteLine(line.Replace(word, "XX00XX"));
                  }
                  else
                  fr.WriteLine(line);
                  n = 0;
               }
               //if there is no lines with odd number of words
               if (linecount == 0)
               {
                  fr.WriteLine("\neiluciu su nelyginiu zodziu skaiciumi nera");
                  fr.WriteLine("-----------------------------------------------");
                  fra.WriteLine("eiluciu su nelyginiu zodziu skaiciumi nera");
               }
            }
         }
      }
}
/// <summary>
/// finds out how many words are in a line of text
/// </summary>
/// <param name="parts"> word array</param>
/// <param name="line">text line</param>
/// <param name="separators">separators</param>
/// <param name="n">number of words in a line</param>
static int WordsCount(string[] parts, string line, char[] separators, out int n)
{
    n = 0;
    foreach (string word in parts)
    {
        n++;
    }
    return n;
}
/// <summary>
/// finds a certain word in a line
/// </summary>
/// <param name="line">text line</param>
/// <param name="separators">separators</param>
/// <param name="parts"> word array</param>
/// <param name="n">number of words in a line</param>
/// <param name="word">the word that was found</param>
```

```
        static string WordFinder(string line, char[] separators, string[] parts, int n, out
              string word)
        {
            word = "";
            int wordnr = 0;
            foreach (string word1 in parts)
            {
                wordnr++;
                if (wordnr == n / 2 + 1)
                {
                    word = word1;
                    break;
                }
            }
            return word;
        }
    }
}
```

4.3. Pradiniai duomenys ir rezultatai

Testas nr.1

```
------------------------------------------------
Pradinis tekstas
------------------------------------------------
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur.
Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.
================================================
Tekstas po redagavimo
------------------------------------------------
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt XX00XX labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation XX00XX laboris nisi ut aliquip ex ea
commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur.
Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia XX00XX mollit anim id est laborum.

Analizės failas
ut ------> XXOOXX
ullamco ------> XXOOXX
deserunt ------> XXOOXX
```

Testas nr.2

```
------------------------------------------------
Pradinis tekstas
------------------------------------------------
Lorem ipsum dolor sit amet, consectetur adipiscing elit, zodis
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. zodis
Excepteur sint occaecat cupidatat non proident, zodis
sunt in culpa qui officia deserunt mollit anim id est laborum.
================================================
Tekstas po redagavimo
------------------------------------------------
```

Lorem ipsum dolor sit XX00XX, consectetur adipiscing elit, zodis
sed do eiusmod tempor incididunt XX00XX labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation XX00XX laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate XX00XX esse cillum dolore eu fugiat nulla pariatur. zodis
Excepteur sint occaecat XX00XX non proident, zodis
sunt in culpa qui officia XX00XX mollit anim id est laborum.

Analizės failas
amet ------> XXOOXX
ut ------> XXOOXX
ullamco ------> XXOOXX
velit ------> XXOOXX
cupidatat ------> XXOOXX
deserunt ------> XXOOXX

Testas nr. 3

Pradinis tekstas

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. zodis
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. zodis
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum. zodis
==========
Tekstas po redagavimo

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. zodis
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. zodis
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum. zodis

eiluciu su nelyginiu zodziu skaiciumi nera

Analizės failas

eiluciu su nelyginiu zodziu skaiciumi nera.

4.4. Dėstytojo pastabos

5. Susieti rinkiniai

5.1. Darbo užduotis

U6–2.

- Futbolas Pirmoje failo eilutėje nurodytas futbolo komandų skaičius. Tolesnėse eilutėse pateikta informacija apie futbolo komandas: pavadinimas, miestas, trenerio pavardė, vardas. Žemiau pateikta I rato rezultatų lentelė, išreikšta pelnytais įvarčiais.
- Suskaičiuokite kiekvienos komandos surinktų taškų skaičių, jei už pergalę skiriami 3 taškai, o už lygiąsias – 1 taškas.
- Sudarykite komandų turnyrinę lentelę – surikiuokite surinktų taškų mažėjimo tvarka. Jei komandos surinko taškų vienodai, aukščiau ta komanda, kuri turi daugiau pergalių.
- Suraskite daugiausiai įvarčių pelniusią komandą.
- Suraskite komandas, kurios daugiausiai rungtynių nepraleido įvarčių.

5.2. Programos tekstas

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

namespace ConsoleApp10
{
    class Team //class for team characteristics
    {
        private string clubn, city, surname, name;
        private int  winpoints, teamgoals;
        public Team()
        {
            clubn = "";
            city = "";
            surname = "";
            name = "";
            winpoints = 0;
            teamgoals = 0;
        }

        public void Set(string clubn, string city,
           string surname, string name)
        {
            this.clubn = clubn;
            this.city = city;
            this.surname = surname;
            this.name = name;
        }

        public string Getclubn() { return clubn; } // returns club name
        public string Getcity() { return city; } // returns city
        public string Getsurname() { return surname; } // returns surname
        public string Getname() { return name; }  //returns name
        public int Getwinpoints() { return winpoints; } //returns win points
        public int Getteamgoals() { return teamgoals; }// returns team goals
        public void Setwinpoints(int Wpnt) { winpoints = Wpnt; } // Sets win points
        public void Setteamgoals(int Tgol) { teamgoals = Tgol; }// sets team goals
        //overloaded operator
        public override string ToString()
        {
            string line;
            line = string.Format("|{0,-20}|{1,-15}|{2,-10}|{3,-10}|",
            clubn, city, surname, name);
```

```
            return line;
        }
        //Arange operators <= and >=
        public static bool operator <=(Team st1, Team st2)
        {
            int v1, v2;
            v1 = st1.Getwinpoints(); v2 = st2.Getwinpoints();
            return (v1 > v2);
        }
        public static bool operator >=(Team st1, Team st2)
        {
            int v1, v2;
            v1 = st1.Getwinpoints(); v2 = st2.Getwinpoints();
            return (v1 < v2);
        }

    }
    //Container
    class Matrix
    {
        //Constants
        const int CMaxlin = 1000;
        const int CMaxSt = 1000;
        private int[,] A; // data matrix
        private Team[] TeamT;
        public int n { get; set; } // line number
        public int m { get; set; } // column number

        public Matrix()
        {
            n = 0;
            m = 0;
            A = new int[CMaxlin, CMaxSt];
            TeamT = new Team[CMaxlin];
        }

        public void Set(Team ob) { TeamT[n++] = ob; }
        public Team Get(int nr) { return TeamT[nr]; }
        //changes matrix values
        public void SetWWW(int i, int j, int r) { A[i, j] = r; }
        //returns matrix values
        public int GetWWW(int i, int j) { return A[i, j]; }
        //-------------------------------------------------
        //Finds how many goals team scored
        public void TeamGoals()
        {
            int goal = 0;
            Team kom;
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    goal = goal + GetWWW(i, j);
                }
                kom = Get(i);
                kom.Setteamgoals(goal);
                goal = 0;
            }
        }
        //Finds victory points
        public void Wins()
        {
            int point = 0;
            Team kom;
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
```

```
                if (GetWWW(i, j) > GetWWW(j, i))
                    point = point + 3;
                else if (GetWWW(i, j) == GetWWW(j, i))
                    point = point + 1;
            }
            kom = Get(i);
            kom.Setwinpoints(point - 1);
            point = 0;
        }
    }
    //Finds which team had most matches without conceding a goal
    public string ZeroConceded(Matrix team)
    {
        int cnt = 0;
        string ats = "";
        for (int j = 0; j < m; j++)
        {
            for (int i = 0; i < n; i++)
            {
                if((GetWWW(j , i)==0)&&(GetWWW(i, j) >= 0))
                {
                    cnt++;
                    ats = string.Format("Team: {0} has most matches without conceding a
      goal.",
                    team.Get(i).Getclubn());
                }
                else ats = "All teams have conceded a goal";
            }
        }return ats;
    }
    //Aranges teams by points
    public void Arange()
    {
        for (int i = 0; i < n - 1; i++)
        {
            Team min = TeamT[i];
            int im = i;
            for (int j = i + 1; j < n; j++)
                if (min >= TeamT[j])
                {
                    min = TeamT[j];
                    im = j;
                }
            TeamT[im] = TeamT[i];
            TeamT[i] = min;
        }
    }
}
class Program
{
    //data and results files
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";

    static void Main(string[] args)
    {
        File.Delete(CFr);
        Matrix team = new Matrix();
        Read(CFd, ref team);
        team.Wins();
        team.TeamGoals();
        Print(CFr, team);
    }

    //Reads form file
    static void Read(string fd, ref Matrix team)
    {
        int nn, points;
```

```
    string line, clubn, name, surname, city;
    using (StreamReader reader = new StreamReader(fd))
    {
        line = reader.ReadLine();
        string[] parts;
        nn = int.Parse(line);
        team.m = nn;
        for (int i = 0; i < nn; i++)
        {
            line = reader.ReadLine();
            parts = line.Split(';');
            clubn = parts[0];
            city = parts[1];
            surname = parts[2];
            name = parts[3];
            Team kom;
            kom = new Team();
            kom.Set(clubn, city, surname, name);
            team.Set(kom);
        }
        for (int i = 0; i < nn; i++)
        {
            line = reader.ReadLine();
            parts = line.Split(';');
            for (int j = 0; j < nn; j++)
            {
                points = int.Parse(parts[j]);
                team.SetWWW(i, j, points);
            }
        }
        team.n = nn;
    }
}
//Finds which team scored the most goals
static string MostGoals(Matrix team)
{
    string ats = "";
    int k = 0;
    for (int i = 0; i < team.m; i++)
    {
        if (team.Get(i).Getteamgoals() > team.Get(k).
        Getteamgoals())
            k = i;
    }
    ats = string.Format("{0} scored: {1} goals ",
    team.Get(k).Getclubn(),
    team.Get(k).Getteamgoals());
    return ats;
}


//Prints results to results file
static void Print(string fv, Matrix team)
{
    using (var fr = File.AppendText(fv))
    {
        string br = new string('-', 60);
        fr.WriteLine("First data");
        fr.WriteLine("\nTeams:");
        fr.WriteLine(br);
        for (int i = 0; i < team.n; i++)
        {
            fr.WriteLine(team.Get(i).ToString());
            fr.WriteLine(br);
        }
        fr.WriteLine("Data matrix");
        for (int i = 0; i < team.m; i++)
        {
```

```
                    for (int j = 0; j < team.n; j++)
                    {
                        fr.Write(team.GetWWW(i, j) + ";");
                    }
                    fr.WriteLine("");
                }
                fr.WriteLine("\n");
                team.Arange();
                fr.WriteLine(br + "-----");
                for (int i = 0; i < team.n; i++)
                {
                    fr.WriteLine(team.Get(i).ToString() +
                    team.Get(i).Getwinpoints() + "  |");
                    fr.WriteLine(br + "---");
                }
                fr.WriteLine("\n");
                fr.WriteLine("Team with most goals:");
                fr.WriteLine(MostGoals(team));
                fr.WriteLine("\nTeam With most matches without conceded goals:");
                fr.WriteLine(team.ZeroConceded(team));
            }
        }
    }
}
```

5.3. Pradiniai duomenys ir rezultatai

Testas nr.1

```
First data

Teams:
-------------------------------------------------------------
|Liverpool FC        | Liverpool     | Klopp    | Jurgen   |
-------------------------------------------------------------
|Chelsea FC          | London        | Tuchel   | Thomas   |
-------------------------------------------------------------
|Manchester United   | Manchester    | Rangnick | Ralf     |
-------------------------------------------------------------
|Manchester City     | Manchester    | Guardiola| Josep    |
-------------------------------------------------------------
|Arsenal             | London        | Arteta   | Mikel    |
-------------------------------------------------------------
Data matrix
0;1;0;4;0;
2;0;0;1;0;
2;0;0;5;1;
3;1;1;0;1;
0;0;1;0;0;


-------------------------------------------------------------------
|Manchester United   | Manchester    | Rangnick | Ralf     |8  |
-------------------------------------------------------------
|Chelsea FC          | London        | Tuchel   | Thomas   |6  |
-------------------------------------------------------------
|Liverpool FC        | Liverpool     | Klopp    | Jurgen   |4  |
-------------------------------------------------------------
|Manchester City     | Manchester    | Guardiola| Josep    |4  |
-------------------------------------------------------------
|Arsenal             | London        | Arteta   | Mikel    |3  |
-------------------------------------------------------------


Team with most goals:
Manchester United scored: 8 goals
```

```
Team With most matches without conceded goals:
Team: Arsenal has most matches without conceding a goal.
```

Testas nr.2

```
First data

Teams:
------------------------------------------------------------
|Liverpool FC        | Liverpool     | Klopp    | Jurgen   |
------------------------------------------------------------
|Chelsea FC          | London        | Tuchel   | Thomas   |
------------------------------------------------------------
|Manchester United   | Manchester    | Rangnick | Ralf     |
------------------------------------------------------------
|Manchester City     | Manchester    | Guardiola| Josep    |
------------------------------------------------------------
|Arsenal             | London        | Arteta   | Mikel    |
------------------------------------------------------------
Data matrix
0;1;2;4;0;
2;0;1;1;0;
2;0;0;5;1;
3;1;1;0;1;
1;0;1;0;0;


---------------------------------------------------------------------
|Chelsea FC          | London        | Tuchel   | Thomas   |8  |
---------------------------------------------------------------
|Manchester United   | Manchester    | Rangnick | Ralf     |5  |
---------------------------------------------------------------
|Arsenal             | London        | Arteta   | Mikel    |5  |
---------------------------------------------------------------
|Manchester City     | Manchester    | Guardiola| Josep    |4  |
---------------------------------------------------------------
|Liverpool FC        | Liverpool     | Klopp    | Jurgen   |4  |
---------------------------------------------------------------


Team with most goals:
Manchester United scored: 8 goals

Team With most matches without conceded goals:
Team: Liverpool FC has most matches without conceding a goal.
```

Testas nr.3

```
First data

Teams:
------------------------------------------------------------
|Liverpool FC        | Liverpool     | Klopp    | Jurgen   |
------------------------------------------------------------
|Chelsea FC          | London        | Tuchel   | Thomas   |
------------------------------------------------------------
|Manchester United   | Manchester    | Rangnick | Ralf     |
------------------------------------------------------------
|Manchester City     | Manchester    | Guardiola| Josep    |
------------------------------------------------------------
|Arsenal             | London        | Arteta   | Mikel    |
------------------------------------------------------------
Data matrix
0;1;2;4;2;
2;0;1;1;0;
2;0;0;5;1;
3;1;1;0;1;
1;0;1;0;0;
```

```
-----------------------------------------------------------------
|Chelsea FC          | London        | Tuchel   | Thomas   |8  |
-----------------------------------------------------------------
|Liverpool FC        | Liverpool     | Klopp    | Jurgen   |7  |
-----------------------------------------------------------------
|Manchester United   | Manchester    | Rangnick | Ralf     |5  |
-----------------------------------------------------------------
|Manchester City     | Manchester    | Guardiola| Josep    |4  |
-----------------------------------------------------------------
|Arsenal             | London        | Arteta   | Mikel    |2  |
-----------------------------------------------------------------


Team with most goals:
Liverpool FC scored: 9 goals

Team With most matches without conceded goals:
Team: Arsenal has most matches without conceding a goal.
```

5.4. Dėstytojo pastabos