

KAUNO TECHNOLOGIJOS UNIVERSITETO GIMNAZIJA

Objektinio programavimo pagrindai II (P175B502)

Laboratorinių darbų ataskaitos

Atliko:

III NFQ klasės mokinys

Augustas Mačijauskas

2018 m. lapkričio 21 d.

Priėmė:

Kęstutis Simonavičius, Jūratė Pauliūtė

TURINYS

1. Grafinė vartotojo sąsaja ir algoritmų taikymas (L1)	3
1.1. Darbo užduotis	3
1.2. Grafinės vartotojo sąsajos schema ir paveikslas	3
1.3. Sąsajoje panaudotų komponentų keičiamos savybės	4
1.4. Programos vartotojo vadovas	4
1.5. Programos tekstas	5
1.6. Pradiniai duomenys ir rezultatai	11
2. Dinaminis masyvas (L2)	14
2.1. Darbo užduotis	14
2.2. Grafinės vartotojo sąsajos schema ir paveikslas	14
2.3. Sąsajoje panaudotų komponentų keičiamos savybės	16
2.4. Programos vartotojo vadovas	17
2.5. Programos tekstas	18
2.6. Pradiniai duomenys ir rezultatai	26
3. Paveldėjimas (L3)	30
3.1. Darbo užduotis	43
3.2. Grafinės vartotojo sąsajos schema ir paveikslas	43
3.3. Sąsajoje panaudotų komponentų keičiamos savybės	45
3.4. Programos vartotojo vadovas	47
3.5. Programos tekstas	47
3.6. Pradiniai duomenys ir rezultatai	57
4. Susietasis sąrašas (L4)	64
4.1. Darbo užduotis	64
4.2. Grafinės vartotojo sąsajos schema ir paveikslas	64
4.3. Sąsajoje panaudotų komponentų keičiamos savybės	65
4.4. Programos vartotojo vadovas	66
4.5. Programos tekstas	66
4.6. Pradiniai duomenys ir rezultatai	73
5. Bendrinės klasės (L5)	76
5.1. Darbo užduotis	76
5.2. Grafinės vartotojo sąsajos schema ir paveikslas	76
5.3. Sąsajoje panaudotų komponentų keičiamos savybės	76
5.4. Programos vartotojo vadovas	76
5.5. Programos tekstas	76
5.6. Pradiniai duomenys ir rezultatai	76

1. Grafinė vartotojo sąsaja ir algoritmų taikymas (L1)

1.1. Darbo užduotis

U1-2. Krepšinis.

Krepšinio mokykloje treniruotes lankančių sąrašas yra tekstiniame faile: *būsimo krepšinininko vardas ir pavardė, amžius ir ūgis*. Pirmoje eilutėje yra krepšinio mokyklos pavadinimas. Turime dviejų mokyklų duomenis.

L1+L2+L4.

- Raskite, koks būsimų krepšinininkų amžiaus vidurkis ir koks ūgio vidurkis kiekvienoje mokykloje.
- Surašykite į atskirą rinkinį visus abiejų mokyklų sportininkus, kurių ūgis didesnis už vidurkį.
- Surikiuokite rezultatų sąrašą amžiaus didėjimo tvarka.
- Pašalinkite iš rezultatų sąrašo krepšinininkus, kurių amžius yra didesnis už nurodytą klaviatūra.

L2 papildymas.

- Papildykite surikiuotą rezultatų sąrašą naujais krepšinininkais, kurių ūgis didesnis, už sudaryto sąrašo krepšinininkų ūgio vidurkį. Duomenys yra faile *Naujokai.txt*. Pirmoje eilutėje – vadybininko vardas ir pavardė.

1.2. Grafinės vartotojo sąsajos schema ir paveikslas

Schema:



Paveikslas:

The screenshot shows the application running. The main text area displays the following content:

Pradiniai pirmos mokyklos duomenys:
Tornado mokykla

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Tomas Tomaitis	14	1,96
2	Klaidas Klaidaitis	16	1,98
3	Petras Petraitis	17	1,92
4	Ignas Ignaitis	15	1,99

Pradiniai antros mokyklos duomenys:
Snaiperio krepšinio mokykla

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Jonas Jonaitis	18	1,95
2	Saulius Saulaitis	15	1,89
3	Andrius Andraitis	17	1,99
4	Paulius Paulaitis	16	1,94
5	Mikas Mikaitis	15	1,99

Amžiaus vidurkis pirmoje (Tornado mokykla) mokykloje yra: 15,50 m.
Amžiaus vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 16,20 m.
Ūgio vidurkis pirmoje (Tornado mokykla) mokykloje yra: 1,96 m.
Ūgio vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 1,95 m.
Konteineris studentų, kurių ūgis didesnis už vidurkį:

On the right side, the "Skaityti" button is highlighted. The "Iveskite norimą amžių" text box contains the value "15". The "Skaičiuoti" button is also visible. The "Spaudinti" button is highlighted with a blue border. The "Baigti" button is at the bottom right.

Krepšininkai

Konteineris studentų, kurių ūgis didesnis už vidurkį:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Andrius Andraitis	15	1,98
2	Tomas Tomaitis	15	1,96
3	Klaidas Klaidaitis	14	1,98

Surikiuotas konteineris:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Klaidas Klaidaitis	14	1,98
2	Tomas Tomaitis	15	1,96
3	Andrius Andraitis	15	1,98

Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Klaidas Klaidaitis	14	1,98
2	Tomas Tomaitis	15	1,96
3	Andrius Andraitis	15	1,98

15

Skaityti

Skaičiuoti

Spaudinti

Baigti

1.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
Form1	Text: Krepšininkai	Programos langas
richTextBox1 (tipas RichTextBox)	Font: Courier New, Regular, 10; Script: Baltic;	Atvaizduoti pradinių duomenų ir skaičiavimų turiniui
Button1 (tipas Button)	Name: read Text: Skaityti Font: Times New Roman, Bold, 14; Script: Baltic;	Mygtukas duomenims iš failo nuskaityti
Button2 (tipas Button)	Name: skaiciuoti Text: Skaičiuoti Font: Times New Roman, Bold, 14; Script: Baltic;	Mygtukas skaičiavimams atlikti
Button3 (tipas Button)	Name: print Text: Spausdinti Font: Times New Roman, Bold, 14; Script: Baltic;	Mygtukas skaičiavimų rezultatams spausdinti
Button4 (tipas Button)	Name: baigti Text: Baigti Font: Times New Roman, Bold, 14; Script: Baltic;	Mygtukas programos lango uždarymui
TextBox1 (tipas TextBox)	Text: Įveskite norimą amžių Font: Courier New, Regular, 12; Script: Baltic;	Teksto laukas norimam amžiui užrašyti

1.4. Programos vartotojo vadovas

Programos naudojimo eiga:

- 1) Paleidę programa, spauskite mygtuką „Skaityti“, kuris nuskaitys krepšininkų duomenis iš failo.
- 2) Kai duomenys bus nuskaityti, į žemiau esantį teksto lauką (su tekstu „Įveskite norimą amžių“) įveskite amžių, už kurį vyresnius krepšininkus norite pašalinti.
- 3) Įvedę amžių, spauskite mygtuką „Skaičiuoti“, kuris atliks visus reikiamus skaičiavimus.
- 4) Spauskite mygtuką „Spausdinti“, kuris atspausdins skaičiavimų rezultatus ekrane.
- 5) Spauskite mygtuką „Baigti“, jei norite baigti darbą su programa.

1.5. Programos tekstas

Klasė Krepsininkas:

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

namespace L1
{
    class Krepsininkas
    {
        public string VP { get; set; }
        public int Amžius { get; set; }
        public double Ūgis { get; set; }

        public Krepsininkas(string vrdpav, int amz, double ug)
        {
            VP = vrdpav;
            Amžius = amz;
            Ūgis = ug;
        }

        public override string ToString()
        {
            string eilute;
            eilute = string.Format(" {0, -20}    {1, 2}    {2, 5:f}", VP, Amžius, Ūgis);
            return eilute;
        }

        public static bool operator <=(Krepsininkas k1, Krepsininkas k2)
        {
            return k1.Amžius < k2.Amžius;
        }

        public static bool operator >=(Krepsininkas k1, Krepsininkas k2)
        {
            return k1.Amžius > k2.Amžius;
        }
    }
}
```

Klasė Krepsininkai:

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

namespace L1
{
    /// <summary>
    ///
    /// </summary>
    class Krepsininkai
    {
        const int Max = 500;
        private Krepsininkas[] Krep;
        public int Kiek { get; set; }

        public Krepsininkai()
        {
            Kiek = 0;
            Krep = new Krepsininkas[Max];
        }
    }
}
```

```

    /// <summary>
    /// Gražina atitinkamo indekso vietoje esantį krepšininko objektą;
    /// </summary>
    /// <returns>Krepsin</returns>
    public Krepsininkas ImtiKrepsininka(int i)
    {
        return Krep[i];
    }

    /// <summary>
    /// Papildo krepšininkų masyvą
    /// </summary>
    public void DetiStudenta(Krepsininkas ob)
    {
        if (Kiek + 1 < Max)
            Krep[Kiek++] = ob;
    }

    public void Rikiuoti()
    {
        for (int i = 0; i < Kiek - 1; i++)
        {
            Krepsininkas pagalb = Krep[i];
            int ind = i;
            for (int j = i + 1; j < Kiek; j++)
            {
                if (Krep[j] <= pagalb)
                {
                    pagalb = Krep[j];
                    ind = j;
                }
            }
            Krep[ind] = Krep[i];
            Krep[i] = pagalb;
        }
    }

    public void Šalinti(int amz)
    {
        for (int i = 0; i < Kiek; i++)
        {
            if (Krep[i].Amžius > amz)
            {
                for (int j = i; j < Kiek - 1; j++)
                {
                    Krep[j] = Krep[j + 1];
                }
                Kiek--;
                i--;
            }
        }
    }
}

```

Programos tekstas:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

```

```

namespace L1
{
    public partial class Form1 : Form
    {
        const string duom1 = "..\\..\\duom1_2.txt";
        const string duom2 = "..\\..\\duom2_2.txt";
        const string rez = "..\\..\\rez.txt";

        string mokPav1;
        string mokPav2;
        Krepsininkai mokykla1;
        Krepsininkai mokykla2;
        Krepsininkai naujasKonteineris;

        public Form1()
        {
            InitializeComponent();

            print.Enabled = false;
            skaiciuoti.Enabled = false;

            if (File.Exists(rez))
                File.Delete(rez);
        }

        /// <summary>
        /// Atlieka duomenų nuskaitymą ir surašymą į konteinerius.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void read_Click(object sender, EventArgs e)
        {
            mokykla1 = Skaityti(duom1, out mokPav1);
            mokykla2 = Skaityti(duom2, out mokPav2);

            Spausdinti(rez, mokykla1, mokPav1, "Pradiniai pirmos mokyklos duomenys:");
            Spausdinti(rez, mokykla2, mokPav2, "Pradiniai antros mokyklos duomenys:");

            naujasKonteineris = new Krepsininkai();

            string x = File.ReadAllText(rez);
            richTextBox1.Text = x;

            skaiciuoti.Enabled = true;
        }

        /// <summary>
        /// Atliekami reikiami skaičiavimai, jų rezultatai įvedami į duomenų failą.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void skaiciuoti_Click(object sender, EventArgs e)
        {
            double amzVid1 = AmziausVidurkis(mokykla1);
            SpausdintiTeksta(rez, string.Format("Amžiaus vidurkis pirmoje ({0}) mokykloje yra: {1, 5:f} m.", mokPav1, amzVid1));
            double amzVid2 = AmziausVidurkis(mokykla2);
            SpausdintiTeksta(rez, string.Format("Amžiaus vidurkis antroje ({0}) mokykloje yra: {1, 5:f} m.\\r\\n", mokPav2, amzVid2));

            double ugioVid1 = ŪgioVidurkis(mokykla1);
            SpausdintiTeksta(rez, string.Format("Ūgio vidurkis pirmoje ({0}) mokykloje yra: {1, 5:f} m.", mokPav1, ugioVid1));
            double ugioVid2 = ŪgioVidurkis(mokykla2);
        }
    }
}

```

```

        SpausdintiTeksta(rez, string.Format("Ūgio vidurkis antroje ({0}) mokykloje yra:
{1, 5:f} m.\r\n", mokPav2, ugioVid2));

        double vidur = Math.Round(((ugioVid1 + ugioVid2) / 2), 2);
        UgisDidesnisUzVidurki(mokykla1, naujasKonteineris, vidur);
        UgisDidesnisUzVidurki(mokykla2, naujasKonteineris, vidur);
        Spausdinti(rez, naujasKonteineris, "Abi mokyklos", "Konteineris studentų, kurių
ūgis didesnis už vidurkį:");

        //mokykla1.Rikiuoti();
        //Spausdinti(rez, mokykla1, mokPav1, "Surikiuotas konteineris:");
        //mokykla2.Rikiuoti();
        //Spausdinti(rez, mokykla2, mokPav2, "Surikiuotas konteineris:");
        naujasKonteineris.Rikiuoti();
        Spausdinti(rez, naujasKonteineris, "Abi mokyklos", "Surikiuotas konteineris:");

        int amz = int.Parse(textBox1.Text);
        //mokykla1.Šalinti(amz);
        //Spausdinti(rez, mokykla1, mokPav1, "Pašalinti moksleiviai, kurių amžius didesnis
už nurodytą:");
        //mokykla2.Šalinti(amz);
        //Spausdinti(rez, mokykla2, mokPav2, "Pašalinti moksleiviai, kurių amžius didesnis
už nurodytą:");
        naujasKonteineris.Šalinti(amz);
        Spausdinti(rez, naujasKonteineris, "Abi mokyklos", "Pašalinti moksleiviai, kurių
amžius didesnis už nurodytą:");

        print.Enabled = true;
    }

    /// <summary>
    /// Spaudina tekstą iš duomenų failo į programos langą.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void print_Click(object sender, EventArgs e)
    {
        string x = File.ReadAllText(rez);
        richTextBox1.Text = x;
    }

    /// <summary>
    /// Užėjus ant teksto lauko, jį išvalo.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void textBox1_MouseClick(object sender, MouseEventArgs e)
    {
        textBox1.Text = "";
    }

    /// <summary>
    /// Baigia programos veikimą.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void baigti_Click(object sender, EventArgs e)
    {
        Close();
    }

    /// <summary>
    /// Formuoja naują konteinerį iš krepšininkų, kurių ūgis didesnis už vidurkį.
    /// </summary>
    /// <param name="kont1">Pirmas konteineris</param>
    /// <param name="kont2">Antras kont</param>
    /// <param name="naujas">Trečias kont</param>
    private void UgisDidesnisUzVidurki(Krepsininkai kont, Krepsininkai naujas, double ugis)

```



```

{
    for (int i = 0; i < kont.Kiek; i++)
    {
        if (kont.ImtiKrepsininka(i).Ūgis > ugis)
        {
            naujas.DetiStudenta(kont.ImtiKrepsininka(i));
        }
    }
}

/// <summary>
/// Randa krepšininkų amžiaus vidurkį.
/// </summary>
/// <param name="kont">Konteineris</param>
/// <returns>Amžiaus vidurkis</returns>
static double AmžiausVidurkis(Krepsininkai kont)
{
    double suma = 0;
    for (int i = 0; i < kont.Kiek; i++)
    {
        suma += kont.ImtiKrepsininka(i).Amžius;
    }

    return suma / kont.Kiek;
}

/// <summary>
/// Randa krepšininkų ūgio vidurkį.
/// </summary>
/// <param name="kont">Konteineris</param>
/// <returns>Ūgio vidurkis</returns>
static double ŪgioVidurkis(Krepsininkai kont)
{
    double suma = 0;
    for (int i = 0; i < kont.Kiek; i++)
    {
        suma += kont.ImtiKrepsininka(i).Ūgis;
    }

    return suma / kont.Kiek;
}

/// <summary>
/// Nuskaito duomenų failus.
/// </summary>
/// <param name="fr"></param>
/// <param name="pav"></param>
/// <returns></returns>
private Krepsininkai Skaityti(string fr, out string pav)
{
    Krepsininkai konteineris = new Krepsininkai();
    using(StreamReader reader = new StreamReader(fr, Encoding.GetEncoding(1257)))
    {
        string line;
        line = reader.ReadLine();
        pav = line;

        while((line = reader.ReadLine()) != null)
        {
            string[] parts = line.Split(';');
            string pavVrd = parts[0];
            int amz = int.Parse(parts[1]);
            double ug = double.Parse(parts[2]);
            Krepsininkas krep = new Krepsininkas(pavVrd, amz, ug);
            konteineris.DetiStudenta(krep);
        }

        return konteineris;
    }
}

```

```

    }
}

/// <summary>
/// Sprendžia duomenis į failą lentelės.
/// </summary>
/// <param name="fw"></param>
/// <param name="kont"></param>
/// <param name="pav"></param>
/// <param name="antraste"></param>
private void Sprendinti(string fw, Krepsininkai kont, string pav, string antraste)
{
    const string virsus =
        "-----\r\n" +
        " Nr. Pavardė ir vardas      Amžius      Ūgis \r\n" +
        "-----";

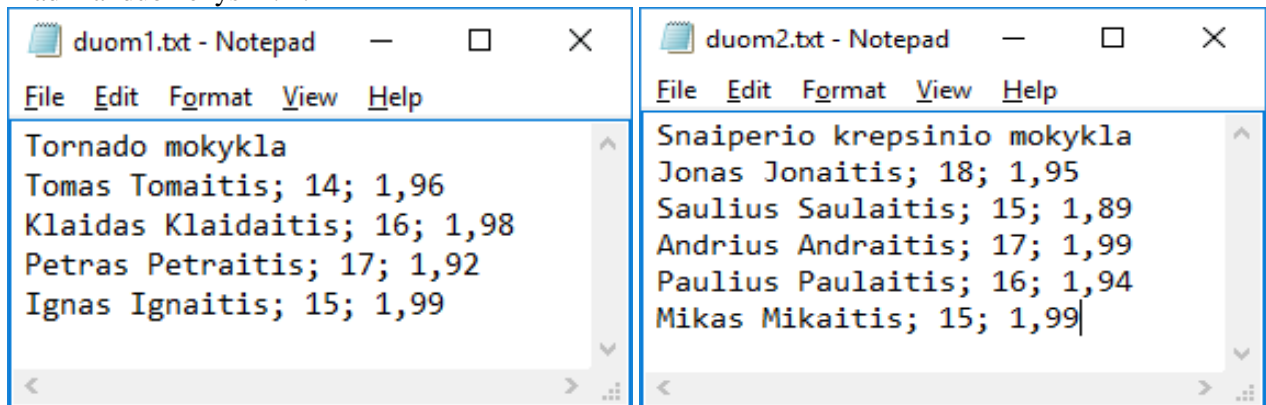
    using (var fr = new StreamWriter(File.Open(fw, FileMode.Append)))
    {
        if (kont.Kiek > 0)
        {
            fr.WriteLine(antraste);
            fr.WriteLine(pav);
            fr.WriteLine(virsus);
            for (int i = 0; i < kont.Kiek; i++)
            {
                Krepsininkas krep = kont.ImtiKrepsininka(i);
                fr.WriteLine("{0, 3} {1}", i + 1, krep);
            }
            fr.WriteLine("-----\r\n");
        }
        else
        {
            fr.WriteLine("Studentų konteineris tuščias!\r\n");
        }
    }
}

/// <summary>
/// Sprendžia tekstą į failą.
/// </summary>
/// <param name="rez"></param>
/// <param name="x"></param>
private void SprendintiTeksta(string rez, string x)
{
    using (StreamWriter fw = new StreamWriter(File.Open(rez, FileMode.Append)))
    {
        fw.WriteLine(x);
    }
}
}
}

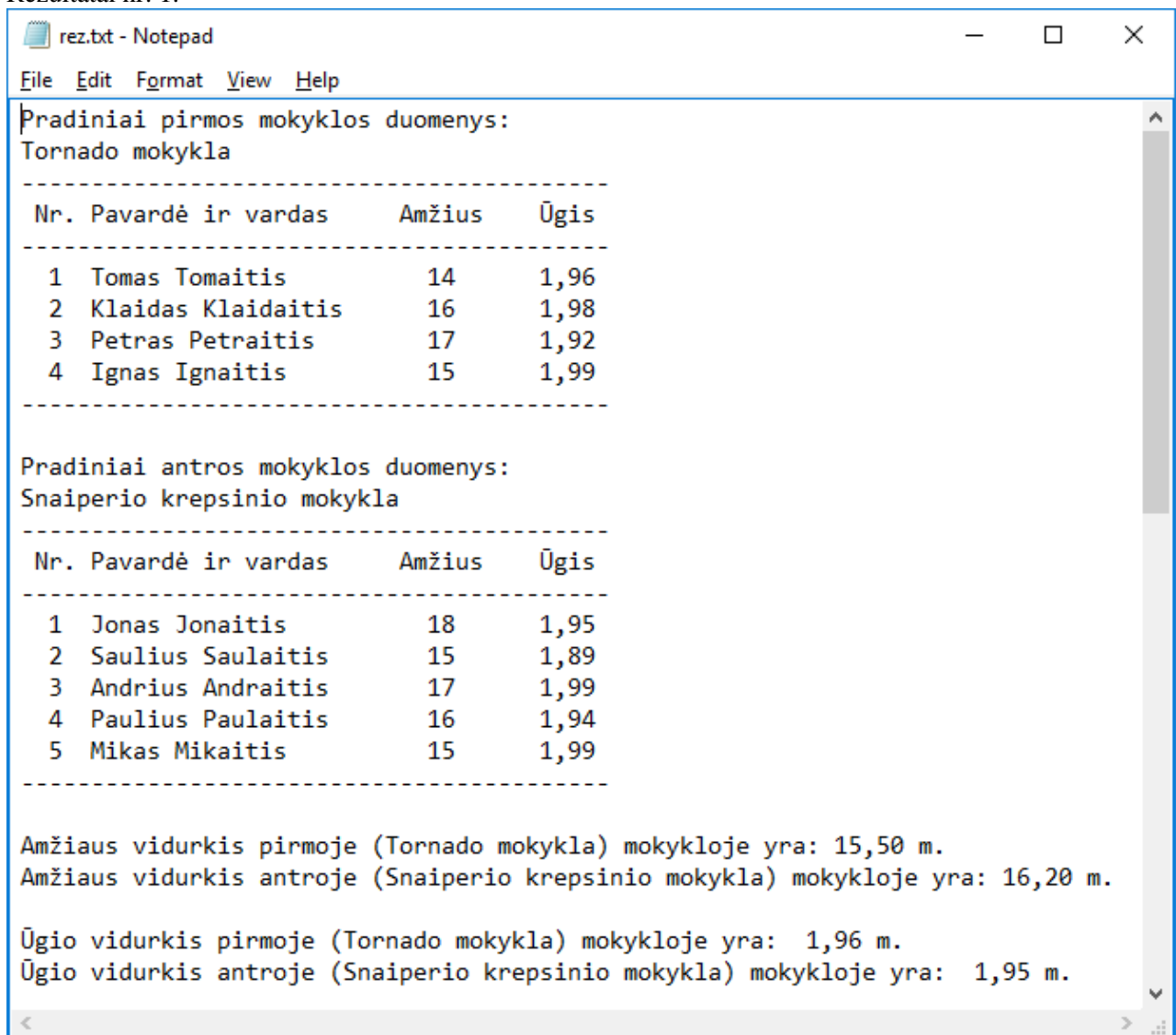
```

1.6. Pradiniai duomenys ir rezultatai

Pradiniai duomenys nr. 1:



Rezultatai nr. 1:



```
rez.txt - Notepad
File Edit Format View Help
Konteineris studentų, kurių ūgis didesnis už vidurkį:
Abi mokyklos

-----
Nr. Pavardė ir vardas    Amžius    Ūgis
-----
1  Klaidas Klaidaitis    16        1,98
2  Ignas Ignaitis        15        1,99
3  Andrius Andraitis     17        1,99
4  Mikas Mikaitis        15        1,99
-----

Surikiuotas konteineris:
Abi mokyklos

-----
Nr. Pavardė ir vardas    Amžius    Ūgis
-----
1  Ignas Ignaitis        15        1,99
2  Mikas Mikaitis        15        1,99
3  Klaidas Klaidaitis     16        1,98
4  Andrius Andraitis     17        1,99
-----

Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

-----
Nr. Pavardė ir vardas    Amžius    Ūgis
-----
1  Ignas Ignaitis        15        1,99
2  Mikas Mikaitis        15        1,99
3  Klaidas Klaidaitis     16        1,98
-----
```

Pradiniai duomenys nr. 2:

duom1_2.txt - Note...	duom2_2.txt - Note...
Sabonio mokykla	Marciulionio mokykla
Jonas Jonaitis; 15; 1,95	Tomas Tomaitis; 15; 1,96
Saulius Saulaitis; 14; 1,89	Klaidas Klaidaitis; 14; 1,98
Andrius Andraitis; 15; 1,98	Petras Petraitis; 15; 1,92
Paulius Paulaitis; 14; 1,94	

Rezultatai nr. 2:

rez.txt - Notepad

File Edit Format View Help

Pradiniai pirmos mokyklos duomenys:
Sabonio mokykla

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Jonas Jonaitis	15	1,95
2	Saulius Saulaitis	14	1,89
3	Andrius Andraitis	15	1,98
4	Paulius Paulaitis	14	1,94

Pradiniai antros mokyklos duomenys:
Marciulionio mokykla

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Tomas Tomaitis	15	1,96
2	Klaidas Klaidaitis	14	1,98
3	Petras Petraitis	15	1,92

Amžiaus vidurkis pirmoje (Sabonio mokykla) mokykloje yra: 14,50 m.
Amžiaus vidurkis antroje (Marciulionio mokykla) mokykloje yra: 14,67 m.

Ūgio vidurkis pirmoje (Sabonio mokykla) mokykloje yra: 1,94 m.
Ūgio vidurkis antroje (Marciulionio mokykla) mokykloje yra: 1,95 m.

rez.txt - Notepad

File Edit Format View Help

Konteineris studentų, kurių ūgis didesnis už vidurkį:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Andrius Andraitis	15	1,98
2	Tomas Tomaitis	15	1,96
3	Klaidas Klaidaitis	14	1,98

Surikiuotas konteineris:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Klaidas Klaidaitis	14	1,98
2	Tomas Tomaitis	15	1,96
3	Andrius Andraitis	15	1,98

Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Klaidas Klaidaitis	14	1,98

2. Dinaminis masyvas (L2)

2.1. Darbo užduotis

U1-2. Krepšinis.

Krepšinio mokykloje treniruotes lankančių sąrašas yra tekstiname faile: *būsimo krepšininko vardas ir pavardė, amžius ir ūgis*. Pirmoje eilutėje yra krepšinio mokyklos pavadinimas. Turime dviejų mokyklų duomenis.

L1+L2+L4.

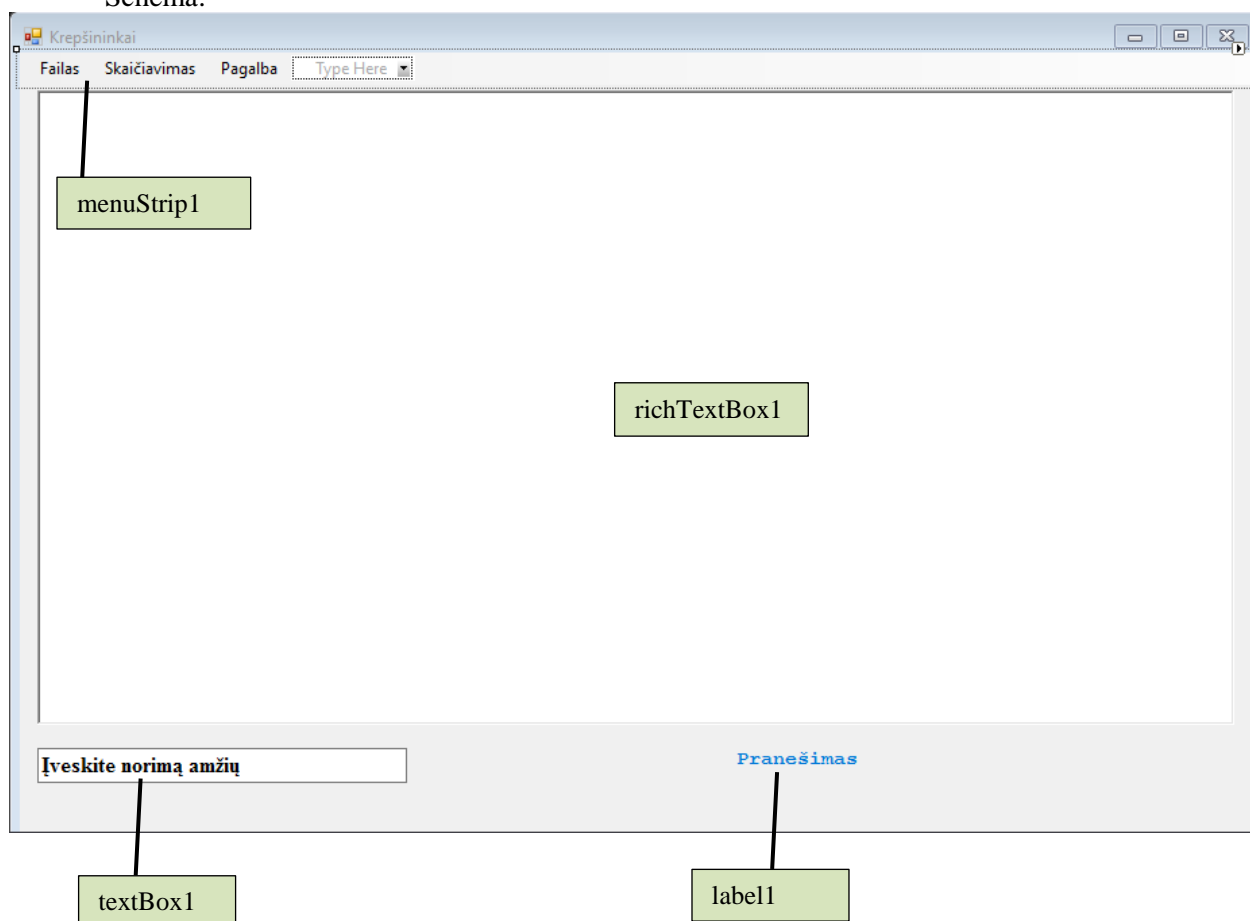
- Raskite, koks būsimų krepšininkų amžiaus vidurkis ir koks ūgio vidurkis kiekvienoje mokykloje.
- Surašykite į atskirą rinkinį visus abiejų mokyklų sportininkus, kurių ūgis didesnis už vidurkį.
- Surikiuokite rezultatų sąrašą amžiaus didėjimo tvarka.
- Pašalinkite iš rezultatų sąrašo krepšininkus, kurių amžius yra didesnis už nurodytą klaviatūra.

L2 papildymas.

- Papildykite surikiuotą rezultatų sąrašą naujais krepšininkais, kurių ūgis didesnis, už sudaryto sąrašo krepšininkų ūgio vidurkį. Duomenys yra faile *Naujokai.txt*. Pirmoje eilutėje – vadybininko vardas ir pavardė.

2.2. Grafinės vartotojo sąsajos schema ir paveikslas

Schema:



Menu punktai:

Failas	Skaičiavimas	Pagalba
Įvesti	Skaičiuoti	Nurodymai vartotojui
Baigti	Spausdinti	Naudojimo sąlygos
	Nauji žaidėjai	
	Išsaugoti	

Paveikslas:

Krepšininkai

Failas Skaičiavimas Pagalba

Pradiniai pirmos mokyklos duomenys:
Tornado mokykla

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Tomas Tomaitis	14	1,96
2	Klaidas Klaidaitis	16	1,98
3	Petras Petraitis	17	1,92
4	Ignas Ignaitis	15	1,99

Pradiniai antros mokyklos duomenys:
Snaiperio krepšinio mokykla

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Jonas Jonaitis	18	1,95
2	Saulius Saulaitis	15	1,89
3	Andrius Andraitis	17	1,99
4	Paulius Paulaitis	16	1,94
5	Mikas Mikaitis	15	1,98

Amžiaus vidurkis pirmoje (Tornado mokykla) mokykloje yra: 15,50 m.
Amžiaus vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 16,20 m.

Ūgio vidurkis pirmoje (Tornado mokykla) mokykloje yra: 1,96 m.

Duomenys atspausdinti.

Krepšininkai

Failas Skaičiavimas Pagalba

Ūgio vidurkis pirmoje (Tornado mokykla) mokykloje yra: 1,96 m.
Ūgio vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 1,95 m.

Konteineris studentų, kurių ūgis didesnis už vidurkį:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Klaidas Klaidaitis	16	1,98
2	Ignas Ignaitis	15	1,99
3	Andrius Andraitis	17	1,99
4	Mikas Mikaitis	15	1,98

Surikiuotas konteineris:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Ignas Ignaitis	15	1,99
2	Mikas Mikaitis	15	1,98
3	Klaidas Klaidaitis	16	1,98
4	Andrius Andraitis	17	1,99

Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

Duomenys atspausdinti.

Krepšininkai

Failas Skaičiavimas Pagalba

1	Ignas Ignaitis	15	1,99
2	Mikas Mikaitis	15	1,98
3	Klaidas Klaidaitis	16	1,98
4	Andrius Andraitis	17	1,99

Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Ignas Ignaitis	15	1,99
2	Mikas Mikaitis	15	1,98

Sąrašas su pridėtais naujais krepšininkais:
Vadybininkas1 Pavarde1

Nr.	Pavardė ir vardas	Amžius	Ūgis
1	Ignas Ignaitis	15	1,99
2	Mikas Mikaitis	15	1,98
3	Vardas2 Pavarde2	15	1,98
4	Vardas1 Pavarde1	18	2,05
5	Vardas3 Pavarde3	18	1,99

15

Duomenys atspausdinti.

2.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
Form1	Text: Krepšininkai	Programos langas
richTextBox1 (tipas RichTextBox)	Font: Courier New, Regular, 11; Script: Baltic;	Atvaizduoti pradinių duomenų ir skaičiavimų turiniui
textBox1 (tipas TextBox)	Text: Įveskite norimą amžių Font: Times New Roman, Bold, 12; Script: Baltic;	Norimo amžiaus įvedimui
label1 (tipas Label)	Text: Pranešimas Font: Courier New, Bold, 11; Script: Baltic; ForeColor: Highlight	Spausdina pranešimus apie programos veikimą.
menuStrip1 (tipas MenuStrip)	Font: Standartiniai nustatymai	Naudojamas meniu punktų, kurie bus naudojami programos veikimo metu, išdėstymui.
failasToolStripMenuItem (tipas ToolStripMenuItem)	Text: Failas Font: Standartiniai nustatymai	Meniu punktas, kuriame yra mygtukai duomenų nuskaitymui ir programos darbo baigimui
ivestiToolStripMenuItem (tipas ToolStripMenuItem)	Name: nuskaityti Text: Failas Font: Standartiniai nustatymai	Meniu punktas duomenims iš failo nuskaityti

baigtiToolStripMenuItem (tipas ToolStripMenuItem)	Name: baigti Text: Baigti Font: Standartiniai nustatymai	Meniu punktas, baigiantis darbą su programa
skaičiavimasToolStripMenuItem (tipas ToolStripMenuItem)	Text: Skaičiavimas Font: Standartiniai nustatymai	Meniu punktas, kuriame yra mygtukai skaičiavimams, spausdinimui, naujų žaidėjų pridėjimui ir saugojimui .csv failą atlikti.
skaiciuotiToolStripMenuItem (tipas ToolStripMenuItem)	Name: skaiciuoti Text: Skaičiuoti Font: Standartiniai nustatymai	Meniu punktas skaičiavimams atlikti
spausdintiToolStripMenuItem (tipas ToolStripMenuItem)	Name: print Text: Spausdinti Font: Standartiniai nustatymai	Meniu punktas gautiems rezultatams atspausdinti
naujiZaidejaiToolStripMenuItem (tipas ToolStripMenuItem)	Name: pridėtiNaujų Text: Nauji žaidėjai Font: Standartiniai nustatymai	Meniu punktas, pridedantis naujus žaidėjus iš failo
issaugotiToolStripMenuItem (tipas ToolStripMenuItem)	Name: išsaugoti Text: Išsaugoti Font: Standartiniai nustatymai	Meniu punktas, leidžiantis išsaugoti rezultatus į .csv failą.
pagalbaToolStripMenuItem (tipas ToolStripMenuItem)	Text: Pagalba Font: Standartiniai nustatymai	Meniu punktas, turintis mygtukus, kurie atvaizduoja nurodymus vartotojui ir naudojimo sąlygas.
nurodymaiVartotojuiToolStripMenuItem (tipas ToolStripMenuItem)	Text: Nurodymai vartotojui Font: Standartiniai nustatymai	Meniu punktas, spausdinantis nurodymus vartotojui.
naudojimoSalygosToolStripMenuItem (tipas ToolStripMenuItem)	Text: Naudojimo sąlygos Font: Standartiniai nustatymai	Meniu punktas, spausdinantis naudojimo sąlygas.

2.4. Programos vartotojo vadovas

1. Atsidarę programą, spauskite Failas - > Įvesti. Pasirinkite failus, iš kurių norite nuskaityti duomenis.
2. Į teksto lauką (pavadinimu „Įveskite norimą amžių“) įveskite amžių, už kurį vyresnius krepšininkus norite pašalinti.
3. Spauskite Skaičiavimas -> Skaičiuoti. Bus atlikti skaičiavimai su pradiniais duomenimis.
4. Spauskite Skaičiavimas -> Spausdinti. Ekrane bus atvaizduoti skaičiavimų rezultatai.
5. Spauskite Skaičiavimas -> Nauji žaidėjai. Prie gautų duomenų bus pridėti nauji žaidėjai, išlaikant rikiavimo tvarką. Ekrane nesimatys jokių pakitimų.
6. Vėl spauskite Skaičiavimas -> Spausdinti. Masyvas su naujais žaidėjais bus atspausdintas ekrane.
7. Spauskite Skaičiavimas -> Išsaugoti. Pasirinkite arba sukurkite naują .csv failą ir rezultatų masyvas bus išsaugotas jame.
8. Jei norite gauti nurodymus vartotojui, spauskite Pagalba -> Nurodymai vartotojui. Naujame lange matysite nuorodas kaip naudotis programa.
9. Norėdami peržiūrėti naudojimo sąlygas, spauskite Pagalba -> Naudojimo sąlygos.

2.5. Programos tekstas

Klasė Krepsininkas:

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

namespace L2
{
    class Krepsininkas
    {
        public string VP { get; set; }
        public int Amžius { get; set; }
        public double Ūgis { get; set; }

        public Krepsininkas(string vrdpav, int amz, double ug)
        {
            VP = vrdpav;
            Amžius = amz;
            Ūgis = ug;
        }

        public override string ToString()
        {
            string eilute;
            eilute = string.Format(" {0, -20}    {1, 2}    {2, 5:f}", VP, Amžius, Ūgis);
            return eilute;
        }

        public static bool operator <=(Krepsininkas k1, Krepsininkas k2)
        {
            int p = String.Compare(k1.VP, k2.VP, StringComparison.CurrentCulture);

            return ((k1.Amžius < k2.Amžius) || (k1.Amžius == k2.Amžius && p < 0));
        }

        public static bool operator >=(Krepsininkas k1, Krepsininkas k2)
        {
            int p = String.Compare(k1.VP, k2.VP, StringComparison.CurrentCulture);

            return ((k1.Amžius > k2.Amžius) || (k1.Amžius == k2.Amžius && p > 0));
        }

        public static bool operator ==(Krepsininkas k1, Krepsininkas k2)
        {
            int p = String.Compare(k1.VP, k2.VP, StringComparison.CurrentCulture);

            return (p == 0 && k1.Amžius == k2.Amžius && k1.Ūgis == k2.Ūgis);
        }

        public static bool operator !=(Krepsininkas k1, Krepsininkas k2)
        {
            int p = String.Compare(k1.VP, k2.VP, StringComparison.CurrentCulture);

            return (p != 0 || k1.Amžius != k2.Amžius || k1.Ūgis != k2.Ūgis);
        }
    }
}
```

Programos tekstas:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace L2
{
    public partial class Form1 : Form
    {
        const string rez = "..\\..\\rez.txt";
        const string naujokai = "..\\..\\Naujokai.txt";
        const string nurodymai = "..\\..\\Nurodymai.txt";
        const string salygos = "..\\..\\NaudojimoSąlygos.txt";

        List<Krepsininkas> mokykla1;
        List<Krepsininkas> mokykla2;
        List<Krepsininkas> naujasKonteineris;
        List<Krepsininkas> naujiKrepsininkai;

        string mokPav1;
        string mokPav2;
        string Vadybininkas;

        public Form1()
        {
            InitializeComponent();

            print.Enabled = false;
            skaiciuoti.Enabled = false;
            pridėtiNaujų.Enabled = false;
            išsaugoti.Enabled = false;

            if (File.Exists(rez))
                File.Delete(rez);
        }

        /// <summary>
        /// Vykdo failų nuskaitymą.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void nuskaityti_Click(object sender, EventArgs e)
        {
            label1.Text = "Vyks ta duomenų įvedimas. Laukite.";
            OpenFileDialog openFileDialog1 = new OpenFileDialog();
            openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
            openFileDialog1.Title = "Pasirinkite pirmąjį duomenų failą";
            DialogResult result = openFileDialog1.ShowDialog();
            if (result == DialogResult.OK)
            {
                string fv = openFileDialog1.FileName;
                mokykla1 = Skaityti(fv, out mokPav1);
            }

            OpenFileDialog openFileDialog2 = new OpenFileDialog();
            openFileDialog2.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
            openFileDialog2.Title = "Pasirinkite antrąjį duomenų failą";
        }
    }
}
```

```

DialogResult result2 = openFileDialog2.ShowDialog();
if (result2 == DialogResult.OK)
{
    string fv = openFileDialog2.FileName;
    mokykla2 = Skaityti(fv, out mokPav2);
}

Spausdinti(rez, mokykla1, mokPav1, "Pradiniai pirmos mokyklos duomenys:");
Spausdinti(rez, mokykla2, mokPav2, "Pradiniai antros mokyklos duomenys:");

naujasKonteineris = new List<Krepsininkas>();

string x = File.ReadAllText(rez);
richTextBox1.Text = x;

skaiciuoti.Enabled = true;
nuskaityti.Enabled = false;
label1.Text = "Duomenys įvesti ir atspausdinti,\ngalima skaičiuoti.";
}

/// <summary>
/// Vykdo skaičiavimus
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void skaiciuoti_Click(object sender, EventArgs e)
{
    label1.Text = "Vykdami skaičiavimai";
    double amzVid1 = AmziausVidurkis(mokykla1);
    SpausdintiTeksta(rez, string.Format("Amžiaus vidurkis pirmoje ({0})" +
    " mokykloje yra: {1, 5:f} m.", mokPav1, amzVid1));
    double amzVid2 = AmziausVidurkis(mokykla2);
    SpausdintiTeksta(rez, string.Format("Amžiaus vidurkis antroje ({0})" +
    " mokykloje yra: {1, 5:f} m.\n", mokPav2, amzVid2));

    double ugioVid1 = ŪgioVidurkis(mokykla1);
    SpausdintiTeksta(rez, string.Format("Ūgio vidurkis pirmoje ({0})" +
    " mokykloje yra: {1, 5:f} m.", mokPav1, ugioVid1));
    double ugioVid2 = ŪgioVidurkis(mokykla2);
    SpausdintiTeksta(rez, string.Format("Ūgio vidurkis antroje ({0})" +
    " mokykloje yra: {1, 5:f} m.\n", mokPav2, ugioVid2));

    double vidur = Math.Round(((ugioVid1 + ugioVid2) / 2), 2);
    UgisDidesnisUzVidurki(mokykla1, naujasKonteineris, vidur);
    UgisDidesnisUzVidurki(mokykla2, naujasKonteineris, vidur);
    Spausdinti(rez, naujasKonteineris, "Abi mokyklos",
    "Konteineris studentų, kurių ūgis didesnis už vidurkį:");

    Rikiuoti(naujasKonteineris);
    Spausdinti(rez, naujasKonteineris, "Abi mokyklos",
    "Surikiuotas konteineris:");

    int amz = int.Parse(textBox1.Text);
    Šalinti(naujasKonteineris, amz);
    Spausdinti(rez, naujasKonteineris, "Abi mokyklos",
    "Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:");

    print.Enabled = true;
    skaiciuoti.Enabled = false;

    label1.Text = "Skaičiavimai atlikti,\ngalima spausdinti.";

    pridėtiNaujų.Enabled = true;
}

/// <summary>
/// Spausdina

```

```

/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void print_Click(object sender, EventArgs e)
{
    string x = File.ReadAllText(rez);
    richTextBox1.Text = x;
    label1.Text = "Duomenys atspausdinti.";
}

/// <summary>
/// Prideda naujus žaidėjus
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void pridėtiNaujų_Click(object sender, EventArgs e)
{
    label1.Text = "Pridedami nauji krepšininkai.";

    naujiKrepšininkai = Skaityti(naujokai, out Vadybininkas);

    for (int i = 0; i < naujiKrepšininkai.Count; i++)
    {
        Krepšininkas krep = naujiKrepšininkai[i];
        int ind = RastiIndeksą(naujasKonteineris, krep);
        naujasKonteineris.Insert(ind, krep);
    }

    Spausdinti(rez, naujasKonteineris, Vadybininkas, "Sąrašas su" +
        " pridėtais naujais krepšininkais:");

    label1.Text = "Duomenys pridėti,\ngalima spausdinti.";

    išsaugoti.Enabled = true;
    pridėtiNaujų.Enabled = false;
}

/// <summary>
/// Leidžia saugoti rezultatus į .csv failą
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void išsaugoti_Click(object sender, EventArgs e)
{
    label1.Text = "Krepšininkai spausdinami į .csv failą.";

    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Filter = "CSV|*.csv";
    saveFileDialog1.Title = "Pasirinkite .csv failą, į kurį norite" +
        " spausdinti";
    DialogResult result = saveFileDialog1.ShowDialog();
    if (result == DialogResult.OK)
    {
        string fv = saveFileDialog1.FileName;
        if (File.Exists(fv))
            File.Delete(fv);
        SpausdintiĮCSVFailą(fv, naujasKonteineris, "Abi mokyklos",
            "Spausdinimas į .csv failą:");
    }

    label1.Text = "Krepšininkai atspaudinti į .csv failą.";
}

/// <summary>
/// Spausdina nurodymus vartotojui
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>

```

```

private void nurodymaiVartotojuiToolStripMenuItem_Click(object sender, EventArgs e)
{
    string x = File.ReadAllText(nurodymai);
    MessageBox.Show(x);
}

/// <summary>
/// Spausdina naudojimo sąlygas
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void naudojimoSąlygosToolStripMenuItem_Click(object sender, EventArgs e)
{
    string x = File.ReadAllText(salygos);
    MessageBox.Show(x);
}

/// <summary>
/// Išvalo teksto lauką ant jo paspaudus
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void textBox1_MouseClick(object sender, MouseEventArgs e)
{
    textBox1.Text = "";
}

/// <summary>
/// Baigia programos darbą
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void baigti_Click(object sender, EventArgs e)
{
    Close();
}

/// <summary>
/// Rikiuoja duomenis
/// </summary>
/// <param name="A"></param>
private void Rikiuoti(List<Krepsininkas> A)
{
    int i = 0;
    bool bk = true;

    while (bk)
    {
        bk = false;
        for (int j = A.Count - 1; j > i; j--)
        {
            if (A[j] <= A[j - 1])
            {
                bk = true;
                Krepsininkas krep = A[j];
                A[j] = A[j - 1];
                A[j - 1] = krep;
            }
        }
        i++;
    }
}

/// <summary>
/// Šalina duomenis
/// </summary>
/// <param name="A"></param>
/// <param name="amz"></param>

```

```

private void Šalinti(List<Krepsininkas> A, int amz)
{
    for (int i = 0; i < A.Count; i++)
    {
        if (A[i].Amžius > amz)
        {
            A.Remove(A[i]);
            i--;
        }
    }
}

/// <summary>
/// Spausdina į .csv failą
/// </summary>
/// <param name="fw"></param>
/// <param name="kont"></param>
/// <param name="pav"></param>
/// <param name="antraste"></param>
private void SpausdintiĮCSVFailą(string fw, List<Krepsininkas> kont, string pav,
string antraste)
{
    using (var fr = new StreamWriter(File.Open(fw, FileMode.Append)))
    {
        if (kont.Count > 0)
        {
            fr.WriteLine("; " + antraste);
            fr.WriteLine("; " + pav);
            fr.WriteLine(" {0, -5} ; {1, -20} ; {2, -8} ; {3, -6} ", "Nr.",
                "Vardas Pavardė", "Amžius", "Ūgis");
            for (int i = 0; i < kont.Count; i++)
            {
                Krepsininkas krep = kont[i];
                fr.WriteLine(" {0, -3} ; {1, -20} ; {2, 3:d} ; {3, 5:f} ",
                    (i + 1).ToString(), krep.VP, krep.Amžius, krep.Ūgis);
            }
        }
        else
        {
            fr.WriteLine("Studentų konteineris tuščias!\n");
        }
    }
}

/// <summary>
/// Randa įterpiamų krepšininkų indeksus
/// </summary>
/// <param name="A"></param>
/// <param name="krep"></param>
/// <returns></returns>
private int RastiIndeksą(List<Krepsininkas> A, Krepsininkas krep)
{
    int ind = A.Count;
    for (int i = 0; i < A.Count; i++)
    {
        if (krep <= A[i])
        {
            ind = i;
        }
    }

    return ind;
}

/// <summary>
/// Formuoja naują konteinerį iš krepšininkų, kurių ūgis didesnis už vidurkį
/// </summary>
/// <param name="kont1">Pirmas konteineris</param>

```

```

/// <param name="kont2">Antras kont</param>
/// <param name="naujas">Trečias kont</param>
private void UgisDidesnisUzVidurki(List<Krepsininkas> kont, List<Krepsininkas> naujas,
double ugis)
{
    for (int i = 0; i < kont.Count; i++)
    {
        if (kont[i].Ugis > ugis)
        {
            naujas.Add(kont[i]);
        }
    }
}

/// <summary>
/// Randa krepšininkų amžiaus vidurkį
/// </summary>
/// <param name="kont">Konteineris</param>
/// <returns>Amžiaus vidurkis</returns>
static double AmziausVidurkis(List<Krepsininkas> kont)
{
    double suma = 0;
    for (int i = 0; i < kont.Count; i++)
    {
        suma += kont[i].Amžius;
    }

    return suma / kont.Count;
}

/// <summary>
/// Randa krepšininkų ūgio vidurkį
/// </summary>
/// <param name="kont">Konteineris</param>
/// <returns>Ūgio vidurkis</returns>
static double ŪgioVidurkis(List<Krepsininkas> kont)
{
    double suma = 0;
    for (int i = 0; i < kont.Count; i++)
    {
        suma += kont[i].Ūgis;
    }

    return suma / kont.Count;
}

/// <summary>
/// Failų nuskaitymas
/// </summary>
/// <param name="fr"></param>
/// <param name="pav"></param>
/// <returns></returns>
private List<Krepsininkas> Skaityti(string fr, out string pav)
{
    List<Krepsininkas> konteineris = new List<Krepsininkas>();
    using (StreamReader reader = new StreamReader(fr, Encoding.GetEncoding(1257)))
    {
        string line;
        line = reader.ReadLine();
        pav = line;

        while ((line = reader.ReadLine()) != null)
        {
            string[] parts = line.Split(';');
            string pavVrd = parts[0];
            int amz = int.Parse(parts[1]);
            double ug = double.Parse(parts[2]);
            Krepsininkas krep = new Krepsininkas(pavVrd, amz, ug);

```



```

        konteineris.Add(krep);
    }

    return konteineris;
}

/// <summary>
/// Spausdina rezultatus į failus
/// </summary>
/// <param name="fw"></param>
/// <param name="kont"></param>
/// <param name="pav"></param>
/// <param name="antraste"></param>
private void Spausdinti(string fw, List<Krepsininkas> kont, string pav,
string antraste)
{
    const string virsus =
        "-----\r\n" +
        " Nr. Pavardė ir vardas      Amžius    Ūgis \r\n" +
        "-----";

    using (var fr = new StreamWriter(File.Open(fw, FileMode.Append)))
    {
        if (kont.Count > 0)
        {
            fr.WriteLine(antraste);
            fr.WriteLine(pav);
            fr.WriteLine(virsus);
            for (int i = 0; i < kont.Count; i++)
            {
                Krepsininkas krep = kont[i];
                fr.WriteLine("{0, 3} {1}", i + 1, krep.ToString());
            }
            fr.WriteLine("-----\n");
        }
        else
        {
            fr.WriteLine("Studentų konteineris tuščias!\n");
        }
    }
}

/// <summary>
/// Spausdina tekstą į failus
/// </summary>
/// <param name="rez"></param>
/// <param name="x"></param>
private void SpausdintiTeksta(string rez, string x)
{
    using (StreamWriter fw = new StreamWriter(File.Open(rez, FileMode.Append)))
    {
        fw.WriteLine(x);
    }
}
}

```

2.6. Pradiniai duomenys ir rezultatai

Pradiniai duomenys nr. 1:

duom1.txt - Notepad	duom2.txt - Notepad
File Edit Format View Help	File Edit Format View Help
Tornado mokykla Tomas Tomaitis; 14; 1,96 Klaidas Klaidaitis; 16; 1,98 Petras Petraitis; 17; 1,92 Ignas Ignaitis; 15; 1,99	Snaiperio krepsinio mokykla Jonas Jonaitis; 18; 1,95 Saulius Saulaitis; 15; 1,89 Andrius Andraitis; 17; 1,99 Paulius Paulaitis; 16; 1,94 Mikas Mikaitis; 15; 1,99

Rezultatai nr. 1:

```

1 Pradiniai pirmos mokyklos duomenys:
2 Tornado mokykla
3 -----
4 Nr. Pavardė ir vardas      Amžius      Ūgis
5 -----
6 1 Tomas Tomaitis          14          1,96
7 2 Klaidas Klaidaitis      16          1,98
8 3 Petras Petraitis        17          1,92
9 4 Ignas Ignaitis          15          1,99
10 -----
11
12 Pradiniai antros mokyklos duomenys:
13 Snaiperio krepsinio mokykla
14 -----
15 Nr. Pavardė ir vardas      Amžius      Ūgis
16 -----
17 1 Jonas Jonaitis          18          1,95
18 2 Saulius Saulaitis       15          1,89
19 3 Andrius Andraitis        17          1,99
20 4 Paulius Paulaitis        16          1,94
21 5 Mikas Mikaitis          15          1,98
22 -----
23
24 Amžiaus vidurkis pirmoje (Tornado mokykla) mokykloje yra: 15,50 m.
25 Amžiaus vidurkis antroje (Snaiperio krepsinio mokykla) mokykloje yra: 16,20 m.
26
27 Ūgio vidurkis pirmoje (Tornado mokykla) mokykloje yra: 1,96 m.
28 Ūgio vidurkis antroje (Snaiperio krepsinio mokykla) mokykloje yra: 1,95 m.
29

```

length: 2793 lines: 73 Ln: 1 Col: 1 Sel: 0|0 Dos\Windows UTF-8 INS

```

C:\Users\amacijauskas\Desktop\Darbai_2\Sprendimai2\L2\rez.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Program.cs rez.txt
30 Konteineris studentų, kurių ūgis didesnis už vidurkį:
31 Abi mokyklos
32 -----
33 Nr. Pavardė ir vardas      Amžius      Ūgis
34 -----
35 1 Klaidas Klaidaitis      16        1,98
36 2 Ignas Ignaitis          15        1,99
37 3 Andrius Andraitis       17        1,99
38 4 Mikas Mikaitis          15        1,98
39 -----
40
41 Surikiuotas konteineris:
42 Abi mokyklos
43 -----
44 Nr. Pavardė ir vardas      Amžius      Ūgis
45 -----
46 1 Ignas Ignaitis          15        1,99
47 2 Mikas Mikaitis          15        1,98
48 3 Klaidas Klaidaitis      16        1,98
49 4 Andrius Andraitis       17        1,99
50 -----
51
52 Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
53 Abi mokyklos
54 -----
55 Nr. Pavardė ir vardas      Amžius      Ūgis
56 -----
57 1 Ignas Ignaitis          15        1,99
58 2 Mikas Mikaitis          15        1,98
59 -----
60
61 Sąrašas su pridėtais naujais krepšininkais:
62 Vadybininkas1 Pavarde1
63 -----
64 Nr. Pavardė ir vardas      Amžius      Ūgis
65 -----
66 1 Ignas Ignaitis          15        1,99
67 2 Mikas Mikaitis          15        1,98
68 3 Vardas2 Pavarde2        15        1,98
69 4 Vardas1 Pavarde1        18        2,05
70 5 Vardas3 Pavarde3        18        1,99
71 -----
72
73
length: 2793 lines: 73 Ln: 1 Col: 1 Sel: 0|0 Dos\Windows UTF-8 INS

```

Pradiniai duomenys nr. 2:

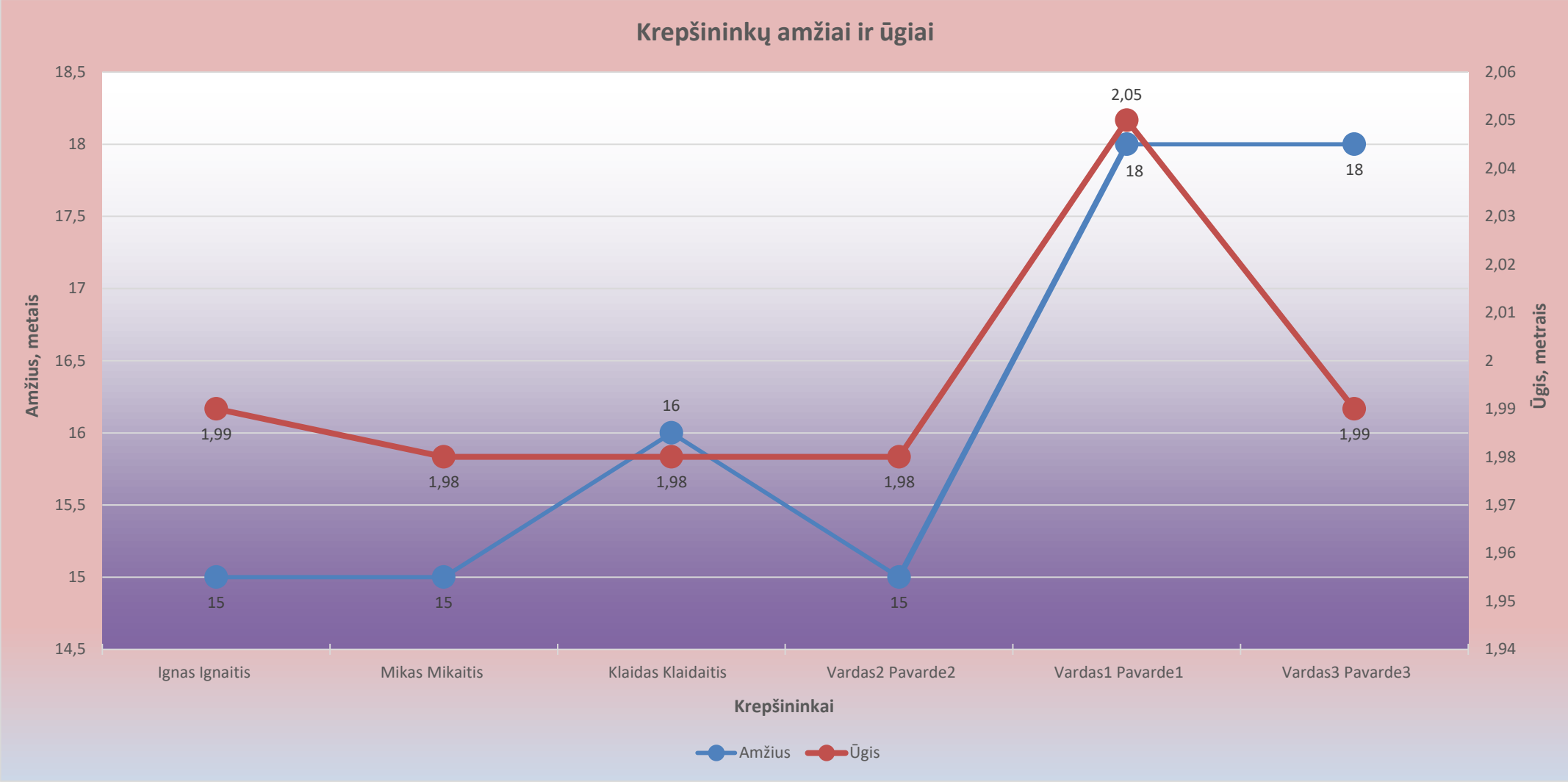
duom1_2.txt - Note...	duom2_2.txt - Note...
Sabonio mokykla Jonas Jonaitis; 15; 1,95 Saulius Saulaitis; 14; 1,89 Andrius Andraitis; 15; 1,98 Paulius Paulaitis; 14; 1,94	Marciulionio mokykla Tomas Tomaitis; 15; 1,96 Klaidas Klaidaitis; 14; 1,98 Petras Petraitis; 15; 1,92

Rezultatai nr. 2:

```

C:\Users\amacijauskas\Desktop\Darbai_2\Sprendimai2\L2\rez.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Program.cs rez.txt
1 Pradiniai pirmos mokyklos duomenys:
2 Sabonio mokykla
3 -----
4 Nr. Pavardė ir vardas Amžius Ūgis
5 -----
6 1 Jonas Jonaitis 15 1,95
7 2 Saulius Saulaitis 14 1,89
8 3 Andrius Andraitis 15 1,98
9 4 Paulius Paulaitis 14 1,94
10 -----
11
12 Pradiniai antros mokyklos duomenys:
13 Marciulionio mokykla
14 -----
15 Nr. Pavardė ir vardas Amžius Ūgis
16 -----
17 1 Tomas Tomaitis 15 1,96
18 2 Klaidas Klaidaitis 14 1,98
19 3 Petras Petraitis 15 1,92
20 -----
21
22 Amžiaus vidurkis pirmoje (Sabonio mokykla) mokykloje yra: 14,50 m.
23 Amžiaus vidurkis antroje (Marciulionio mokykla) mokykloje yra: 14,67 m.
24
25 Ūgio vidurkis pirmoje (Sabonio mokykla) mokykloje yra: 1,94 m.
26 Ūgio vidurkis antroje (Marciulionio mokykla) mokykloje yra: 1,95 m.
27
length: 2686 lines: 71 Ln: 1 Col: 1 Sel: 0|0 Dos\Windows UTF-8 INS
C:\Users\amacijauskas\Desktop\Darbai_2\Sprendimai2\L2\rez.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Program.cs rez.txt
28 Konteineris studentų, kurių ūgis didesnis už vidurkį:
29 Abi mokyklos
30 -----
31 Nr. Pavardė ir vardas Amžius Ūgis
32 -----
33 1 Andrius Andraitis 15 1,98
34 2 Tomas Tomaitis 15 1,96
35 3 Klaidas Klaidaitis 14 1,98
36 -----
37
38 Surikiuotas konteineris:
39 Abi mokyklos
40 -----
41 Nr. Pavardė ir vardas Amžius Ūgis
42 -----
43 1 Klaidas Klaidaitis 14 1,98
44 2 Andrius Andraitis 15 1,98
45 3 Tomas Tomaitis 15 1,96
46 -----
47
48 Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
49 Abi mokyklos
50 -----
51 Nr. Pavardė ir vardas Amžius Ūgis
52 -----
53 1 Klaidas Klaidaitis 14 1,98
54 2 Andrius Andraitis 15 1,98
55 3 Tomas Tomaitis 15 1,96
56 -----
57
58 Sąrašas su pridėtais naujais krepšininkais:
59 Vadybininkas1 Pavarde1
60 -----
61 Nr. Pavardė ir vardas Amžius Ūgis
62 -----
63 1 Klaidas Klaidaitis 14 1,98
64 2 Andrius Andraitis 15 1,98
65 3 Tomas Tomaitis 15 1,96
66 4 Vardas2 Pavarde2 15 1,98
67 5 Vardas1 Pavarde1 18 2,05
68 6 Vardas3 Pavarde3 18 1,99
69 -----
length: 2686 lines: 71 Ln: 10 Col: 43 Sel: 0|0 Dos\Windows UTF-8 INS

```



3. Paveldėjimas (L3)

3.1. Darbo užduotis

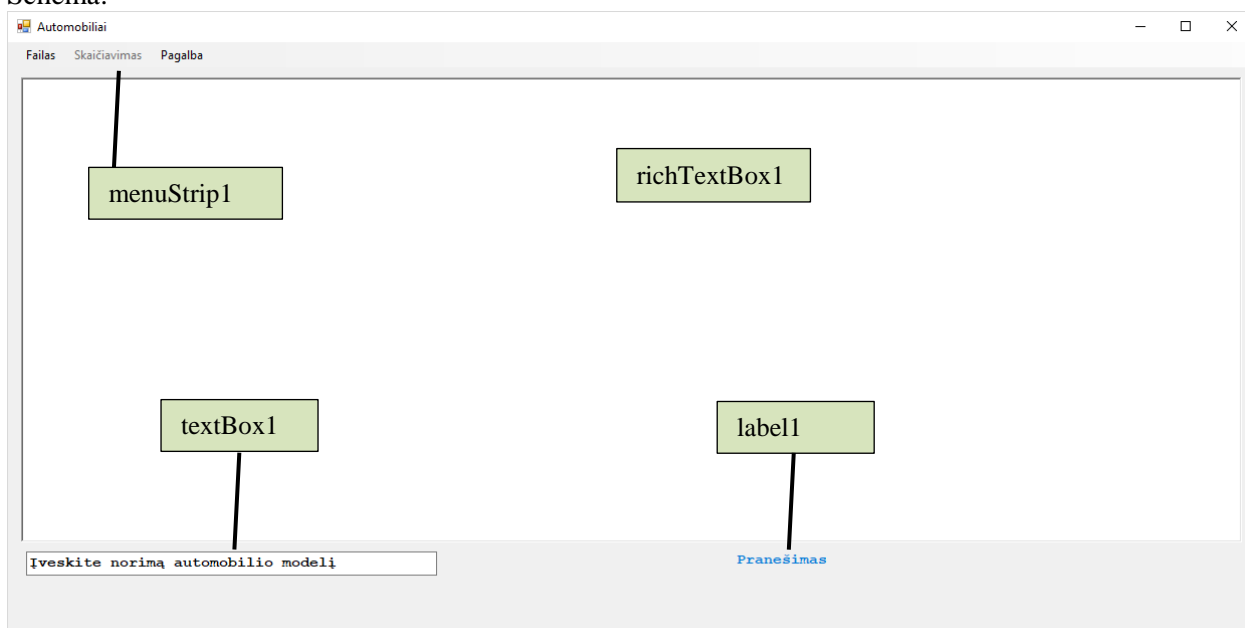
2. **Automobilių parkas.** Turite duomenis apie UAB „Žaibas“ priklausančius automobilius. Duomenų faile pateikta ši informacija: valstybinis numeris, gamintojas, modelis, pagaminimo metai ir mėnuo, techninės apžiūros galiojimo data, kuras, vidutinės kuro sąnaudos (100km).

- Raskite du naujausius automobilius (visi duomenys).
- Sudarykite visų nurodytos markės automobilių sąrašą (visi duomenys).
- Surikiuokite sudarytą sąrašą pagal pasirinktus du požymius.
- Sudarykite visų automobilių gamintojų sąrašą be pasikartojimų.

Visus skaičiavimų rezultatus pateikite rezultatų faile lentelėmis.

3.2. Grafinės vartotojo sąsajos schema ir paveikslas

Schema:



Meniu punktai:

Failas	Skaičiavimas	Pagalba
Įvesti	Skaičiuoti	Informacija
Baigti	Spausdinti	

Paveikslas:

Pradiniai duomenys:

Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	ABC123	Nissan	NX	2004 05	2018 03 23	Benzinas	8,0
2	GHJ456	Nissan	Quasquai	2004 05	2018 03 23	Benzinas	7,0
3	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas	8,0
4	KVK564	Alfa romeo	Turbo10	2005 07	2018 10 28	Dyzelinas	9,0
5	OLE753	Audi	24	2009 06	2018 04 09	Hibridas	3,0
6	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
7	REP427	Toyota	Auris	2009 06	2018 04 09	Hibridas	3,0
8	KEB001	Mercedes	300SL	2011 01	2018 07 22	Dujos	8,0
9	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,0
10	ABC123	Mercedes	W25 Silver Arrow	2011 01	2018 12 31	Dujos	10,0
11	FLM562	Mercedes	W25 Silver Arrow	2011 04	2018 06 22	Dujos	10,0
12	KLG576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
13	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
14	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

Naujausias automobilis:

Valstybinis numeris : KLG576
Gamintojas : Tesla
Modelis : ModelX
Pagaminimo data : 2017 12
Techninė apžiūra : 2018 06 20
Kuro tipas : Elektra

ModelX

Skaičiavimai atlikti. Galima spausdinti.

Automobiliai

Failas
Skaiciavimas
Pagalba

Naujausias automobilis:

Valstybinis numeris : KLG576

Gamintojas : Tesla

Modelis : ModelX

Pagaminimo data : 2017 12

Techninė apžiūra : 2018 06 20

Kuro tipas : Elektra

Kuro sąnaudos : 0

Antras naujausias automobilis:

Valstybinis numeris : ASD653

Gamintojas : Tesla

Modelis : ModelX

Pagaminimo data : 2017 12

Techninė apžiūra : 2018 06 20

Kuro tipas : Elektra

Kuro sąnaudos : 0

ModelX markės automobiliai:

Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
2	KLK576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
3	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
4	LER159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

ModelX

Skaiciavimai atlikti. Galima spausdinti.

Automobiliai

Failas
Skaiciavimas
Pagalba

ModelX markės automobiliai:

Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
2	KLK576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
3	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
4	LER159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

Surikiuoti ModelX markės automobiliai:

Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
2	LER159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0
3	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
4	KLK576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0

Gamintojų sąrašas:

Nissan

Chevrolet

Alfa romeo

Audi

Tesla

Tovota

ModelX

Skaiciavimai atlikti. Galima spausdinti.

Automobiliai

Failas
Skaiciavimas
Pagalba

Gamintojų sąrašas:

Nissan

Chevrolet

Alfa romeo

Audi

Tesla

Toyota

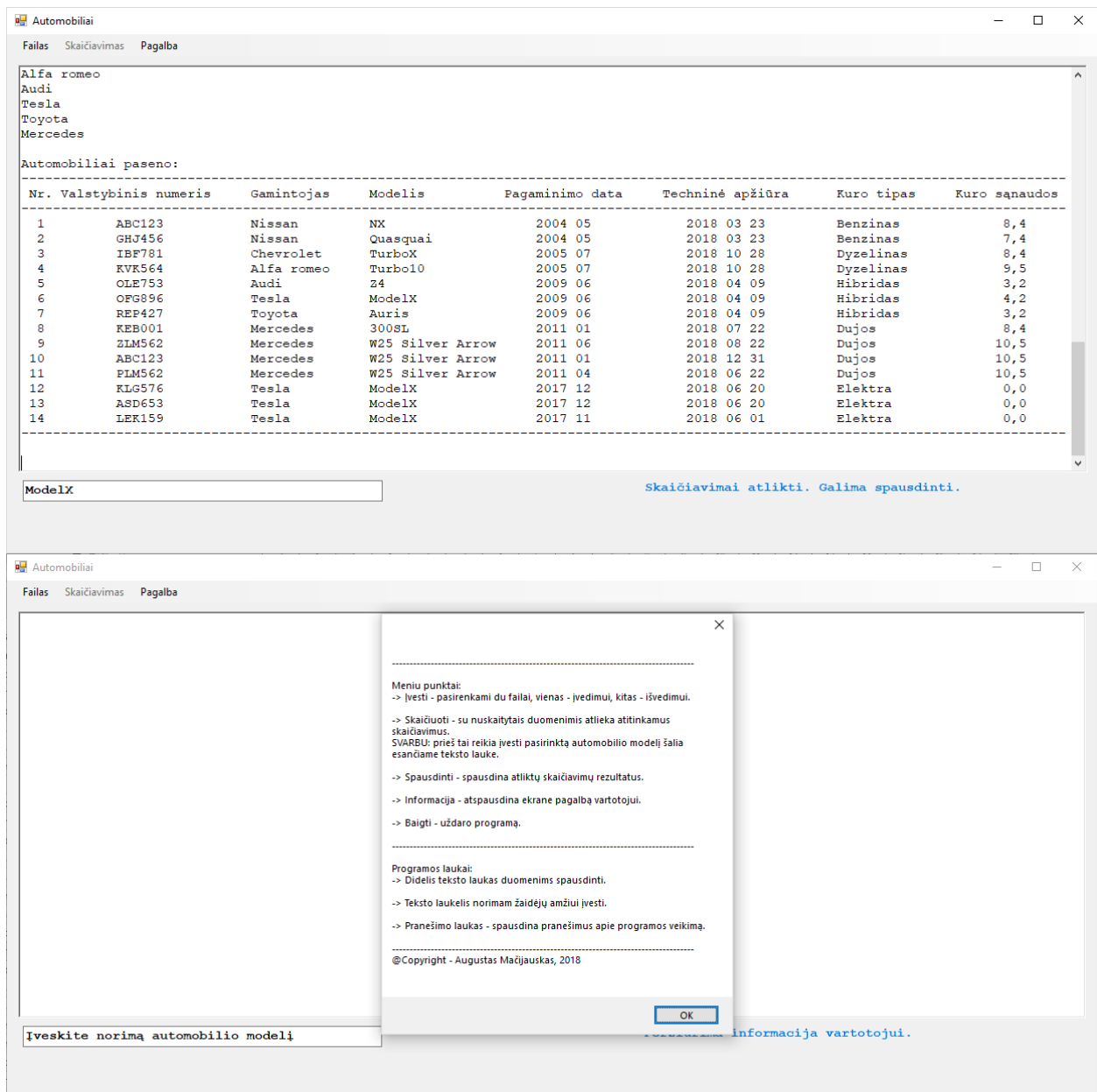
Mercedes

Automobiliai paseno:

Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	ABC123	Nissan	NX	2004 05	2018 03 23	Benzinas	8,4
2	GHJ456	Nissan	Quasquai	2004 05	2018 03 23	Benzinas	7,4
3	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas	8,4
4	KVK564	Alfa romeo	Turbo10	2005 07	2018 10 28	Dyzelinas	9,5
5	OLE753	Audi	Z4	2009 06	2018 04 09	Hibridas	3,2
6	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,2
7	REP427	Toyota	Auris	2009 06	2018 04 09	Hibridas	3,2
8	KEB001	Mercedes	300SL	2011 01	2018 07 22	Dujos	8,4
9	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,5
10	ABC123	Mercedes	W25 Silver Arrow	2011 01	2018 12 31	Dujos	10,5
11	PIM562	Mercedes	W25 Silver Arrow	2011 04	2018 06 22	Dujos	10,5
12	KLK576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
13	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
14	LER159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

ModelX

Skaiciavimai atlikti. Galima spausdinti.



3.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
Form1	Text: Automobiliai	Programos langas
richTextBox1 (tipas RichTextBox)	Font: Courier New, Regular, 11; Script: Baltic;	Atvaizduoti pradinių duomenų ir skaičiavimų turiniui.
textBox1 (tipas TextBox)	Text: Įveskite norimą automobilio modelį Font: Courier New, Bold, 11; Script: Baltic;	Norimo amžiaus įvedimui
label1 (tipas Label)	Text: Pranešimas Font: Courier New, Bold, 11; Script: Baltic; ForeColor: Highlight	Spausdina pranešimus apie programos veikimą.
menuStrip1 (tipas MenuStrip)	Font: Standartiniai nustatymai	Naudojamas meniu punktų, kurie bus naudojami

		programos veikimo metu, išdėstymui.
failasToolStripMenuItem (tipas ToolStripMenuItem)	Font: Standartiniai nustatymai	Meniu punktas, kuriame yra mygtukai duomenų nuskaitymui, atliktų skaičiavimų spausdinimui ir darbo su programa baigimui.
ivestiToolStripMenuItem (tipas ToolStripMenuItem)	Name: Įvesti Text: Įvesti Font: Standartiniai nustatymai	Meniu punktas duomenims iš failo nuskaityti.
spausdintiToolStripMenuItem (tipas ToolStripMenuItem)	Name: spausdinti Text: Spausdinti Font: Standartiniai nustatymai	Meniu punktas gautiems rezultatams atspausdinti.
baigtiToolStripMenuItem (tipas ToolStripMenuItem)	Name: Baigti Text: Baigti Font: Standartiniai nustatymai	Meniu punktas, baigiantis darbą su programa.
skaičiavimasToolStripMenuItem (tipas ToolStripMenuItem)	Text: Skaičiavimas Font: Standartiniai nustatymai	Meniu punktas skaičiavimams atlikti.
spausdintiToolStripMenuItem (tipas ToolStripMenuItem)	Name: Spausdinti Text: Spausdinti Font: Standartiniai nustatymai	Meniu punktas gautiems rezultatams atspausdinti.
pagalbaToolStripMenuItem (tipas ToolStripMenuItem)	Text: Pagalba Font: Standartiniai nustatymai	Meniu punktas, turintis mygtukus, kurie atvaizduoja nurodymus vartotojui ir naudojimo sąlygas.
informacijaToolStripMenuItem (tipas ToolStripMenuItem)	Name: Informacija Text: Informacija Font: Standartiniai nustatymai	Meniu punktas, kurį paspaudus spausdinama pagalba vartotojui.

3.4. Programos vartotojo vadovas

1. Atsidarius programos langui, spauskite Failas -> Įvesti ir pasirinkite du failus: pirmąjį, iš kurio norite skaityti, ir antrąjį, į kurį norite spausdinti.
2. Į apačioje esantį teksto lauką pasirinkite automobilio modelį, kurį norite atrinkti iš viso sąrašo.
3. Spauskite Skaičiavimas -> Skaičiuoti.
4. Spauskite Skaičiavimas -> Spausdinti.
5. Peržiūrėkite gautus rezultatus.
6. Norėdami gauti pagalbos, spauskite Pagalba -> Informacija.

3.5. Programos tekstas

```
Klasė AutomobilioBlueprint.cs:
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace L3_AugustasMačijauskas
{
    abstract class AutomobilioBlueprint
```

```

{
    public string Gamintojas { get; set; }
    public string Modelis { get; set; }
    public DateTime PagaminimoData { get; set; }
    public string Kuras { get; set; }

    public AutomobilioBlueprint(string gam, string mod, DateTime pag, string kur)
    {
        this.Gamintojas = gam;
        this.Modelis = mod;
        this.PagaminimoData = pag;
        this.Kuras = kur;
    }

    public override string ToString()
    {
        return string.Format(" {0, -10}      {1, -7}      {2:yyyy MM}      {3, -
15} ", Gamintojas, Modelis, PagaminimoData, Kuras);
    }

    public abstract void Nusidėvėjimas();
}
}

```

Klasė Automobilis.cs:

```

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

namespace L3_AugustasMačiiauskas
{
    class Automobilis : AutomobilioBlueprint
    {
        public string ValstybinisNumeris { get; set; }
        public DateTime TechninėApžiūra { get; set; }
        public double VidutinėsSąnaudos { get; set; }

        public Automobilis(string valstNr, string gam, string mod, DateTime pag, DateTime
tech, string kur, double vidSan) : base(gam, mod, pag, kur)
        {
            this.ValstybinisNumeris = valstNr;
            this.TechninėApžiūra = tech;
            this.VidutinėsSąnaudos = vidSan;
        }

        public static bool operator <(Automobilis a1, Automobilis a2)
        {
            int poz = string.Compare(a1.ValstybinisNumeris, a2.ValstybinisNumeris,
StringComparison.CurrentCulture);

            return ((a1.TechninėApžiūra < a2.TechninėApžiūra) || ((a1.TechninėApžiūra ==
a2.TechninėApžiūra) && (poz < 0)));
        }

        public static bool operator >(Automobilis a1, Automobilis a2)
        {
            int poz = string.Compare(a1.ValstybinisNumeris, a2.ValstybinisNumeris,
StringComparison.CurrentCulture);

            return ((a1.TechninėApžiūra > a2.TechninėApžiūra) || ((a1.TechninėApžiūra ==
a2.TechninėApžiūra) && (poz > 0)));
        }

        public static bool operator <=(Automobilis a1, Automobilis a2)
        {
            return a1.PagaminimoData < a2.PagaminimoData;
        }
    }
}

```

```

        public static bool operator >=(Automobilis a1, Automobilis a2)
        {
            return a1.PagaminimoData > a2.PagaminimoData;
        }

        public override string ToString()
        {
            return string.Format("{0, 6} {1, -10} {2, -20} {3:yyyy MM}
{4:yyyy MM dd} {5, -15} {6, 5:f1}", ValstybinisNumeris, Gamintojas, Modelis,
PagaminimoData, TechninėApžiūra, Kuras, VidutinėsSąnaudos);
        }

        public override void Nusidėvėjimas()
        {
            VidutinėsSąnaudos *= 1.05;
        }
    }
}

```

Programos kodas:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace L3_AugustasMačijauskas
{
    public partial class Form1 : Form
    {
        List<Automobilis> automobiliai;
        List<Automobilis> markes;
        List<string> gamintojai;
        string rez = "";
        const string info = "..\\..\\informacijaVartotojui.txt";

        public Form1()
        {
            InitializeComponent();
            Skaiciuoti.Enabled = false;
            Spausdinti.Enabled = false;
            skaičiavimasToolStripMenuItem.Enabled = false;
        }

        private void įvestiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            label1.Text = "Vyksta duomenų įvedimas. Prašome palaukti.";
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
            openFileDialog.Title = "Pasirinkite duomenų failą";
            DialogResult result = openFileDialog.ShowDialog();
            if (result == DialogResult.OK)
            {
                string fn = openFileDialog.FileName;
                automobiliai = Read(fn);
            }

            label1.Text = "Duomenys nuskaityti. Pasirinkite failą, \nį kurį norite
spausdinti.";
            SaveFileDialog saveFile = new SaveFileDialog();
            saveFile.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
            saveFile.Title = "Pasirinkite failą, į kurį norite spausdinti";
            result = saveFile.ShowDialog();
        }
    }
}

```

```

if (result == DialogResult.OK)
{
    rez = saveFile.FileName;
    if (File.Exists(rez))
        File.Delete(rez);
    Write(rez, automobiliai, "Pradiniai duomenys:");
}

string x = File.ReadAllText(rez);
richTextBox1.Text = x;

label1.Text = "Duomenys nuskaityti ir atspaudtinti.\nGalima atlikti
skaičiavimus.";

Skaiciuoti.Enabled = true;
Ivesti.Enabled = false;
skaičiavimasToolStripMenuItem.Enabled = true;
}

private List<Automobilis> Read(string fn)
{
    List<Automobilis> ret = new List<Automobilis>();
    using (StreamReader reader = new StreamReader(fn))
    {
        string line;
        string[] parts;
        string vn, gam, mod, kur;
        DateTime pag, tech;
        double san;

        while ((line = reader.ReadLine()) != null)
        {
            parts = line.Split(';');
            vn = parts[0].Trim();
            gam = parts[1].Trim();
            mod = parts[2].Trim();
            pag = DateTime.Parse(parts[3].Trim());
            tech = DateTime.Parse(parts[4].Trim());
            kur = parts[5].Trim();
            san = double.Parse(parts[6].Trim());
            Automobilis naujas = new Automobilis(vn, gam, mod, pag, tech, kur, san);
            ret.Add(naujas);
        }
    }
    return ret;
}

private void Write(string fn, List<Automobilis> kont, string antraste)
{
    const string virsus =
    "-----\r\n" +
    " Nr. Valstybinis numeris    Gamintojas    Modelis    Pagaminimo data
Techninė apžiūra    Kuro tipas    Kuro sąnaudos \r\n" +
    "-----";

    using (var writer = new StreamWriter(File.Open(fn, FileMode.Append)))
    {
        if (kont.Count < 0)
        {
            writer.WriteLine("Studentų konteineris tuščias!\n");
        }
        else
        {
            writer.WriteLine(antraste);
            writer.WriteLine(virsus);
            for (int i = 0; i < kont.Count; i++)

```

```

        {
            Automobilis krep = kont[i];
            writer.WriteLine("{0, 3} {1}", i + 1, krep.ToString());
        }
        writer.WriteLine("-----\n");
    }
}

private void Write(string fn, List<string> gamintojai, string antraste)
{
    using (var writer = new StreamWriter(File.Open(fn, FileMode.Append)))
    {
        writer.WriteLine(antraste);
        foreach (string gamintojas in gamintojai)
        {
            writer.WriteLine(gamintojas);
        }
        writer.WriteLine();
    }
}

private void Write(string fn, string text)
{
    using (var writer = new StreamWriter(File.Open(fn, FileMode.Append)))
    {
        writer.WriteLine(text);
    }
}

private void Baigti_Click(object sender, EventArgs e)
{
    Close();
}

private void Skaiciuoti_Click(object sender, EventArgs e)
{
    label1.Text = "Atliekami skaičiavimai. Prašome palaukti.";
    Automobilis naujausias1;
    Automobilis naujausias2;
    RastiDuNaujausius(automobiliai, out naujausias1, out naujausias2);
    Write(rez, "Naujausias automobilis:\n" +
        $"{ "Valstybinis numeris", 25 } : {naujausias1.ValstybinisNumeris, 20}\n" +
        $"{ "Gamintojas", 25 } : {naujausias1.Gamintojas, 20}\n" +
        $"{ "Modelis", 25 } : {naujausias1.Modelis, 20}\n" +
        $"{ "Pagaminimo data", 25 } : {naujausias1.PagaminimoData.ToString("yyyy
MM"), 20}\n" +
        $"{ "Techninė apžiūra", 25 } : {naujausias1.TechninėApžiūra.ToString("yyyy MM
dd"), 20}\n" +
        $"{ "Kuro tipas", 25 } : {naujausias1.Kuras, 20}\n" +
        $"{ "Kuro sąnaudos", 25 } : {naujausias1.VidutinėsSąnaudos, 20}\n");
    Write(rez, "Antras naujausias automobilis:\n" +
        $"{ "Valstybinis numeris", 25 } : {naujausias2.ValstybinisNumeris, 20}\n" +
        $"{ "Gamintojas", 25 } : {naujausias2.Gamintojas, 20}\n" +
        $"{ "Modelis", 25 } : {naujausias2.Modelis, 20}\n" +
        $"{ "Pagaminimo data", 25 } : {naujausias2.PagaminimoData.ToString("yyyy
MM"), 20}\n" +
        $"{ "Techninė apžiūra", 25 } : {naujausias2.TechninėApžiūra.ToString("yyyy MM
dd"), 20}\n" +
        $"{ "Kuro tipas", 25 } : {naujausias2.Kuras, 20}\n" +
        $"{ "Kuro sąnaudos", 25 } : {naujausias2.VidutinėsSąnaudos, 20}\n");

    string marke = textBox1.Text;
    markes = AtrinktiPagalMarke(automobiliai, marke);
    if (markes.Count > 0)
    {
        Write(rez, markes, $"{marke} markės automobiliai:");
    }
}

```

```

        Rikiuoti(markes);
        Write(rez, marques, $"Surikiuoti {marke} markės automobiliai:");
    }
    else
    {
        Write(rez, "Modelių sąrašas tuščias!\n");
    }

    gamintojai = AtrinktiGamintojus(automobiliai);
    Write(rez, gamintojai, "Gamintojų sąrašas:");

    Paseno(automobiliai);
    Write(rez, automobiliai, "Automobiliai paseno:");

    Skaiciuoti.Enabled = false;
    Spausdinti.Enabled = true;
    label1.Text = "Skaičiavimai atlikti. Galima spausdinti.";
}

private void Paseno(List<Automobilis> automobiliai)
{
    // automobiliai.ForEach(x => x.Nusidėvėjimas());
    foreach (Automobilis auto in automobiliai)
    {
        auto.Nusidėvėjimas();
    }
}

private List<string> AtrinktiGamintojus(List<Automobilis> automobiliai)
{
    List<string> ret = new List<string>();

    for (int i = 0; i < automobiliai.Count; i++)
    {
        if (!ret.Contains(automobiliai[i].Gamintojas))
        {
            ret.Add(automobiliai[i].Gamintojas);
        }
    }

    return ret;
}

private void Rikiuoti(List<Automobilis> automobiliai)
{
    int minIndex;
    for (int i = 0; i < automobiliai.Count - 1; i++)
    {
        minIndex = i;
        for (int j = i + 1; j < automobiliai.Count; j++)
        {
            if (automobiliai[j] < automobiliai[minIndex])
            {
                minIndex = j;
            }

            Automobilis temp = automobiliai[i];
            automobiliai[i] = automobiliai[minIndex];
            automobiliai[minIndex] = temp;
        }
    }
}

private List<Automobilis> AtrinktiPagalMarke(List<Automobilis> automobiliai, string
marke)
{
    List<Automobilis> ret = new List<Automobilis>();
    for (int i = 0; i < automobiliai.Count; i++)

```

```

        {
            if (automobiliai[i].Modelis == marke)
            {
                Automobilis naujas = new Automobilis(automobiliai[i].ValstybinisNumeris,
                    automobiliai[i].Gamintojas, automobiliai[i].Modelis,
                    automobiliai[i].PagaminimoData, automobiliai[i].TechninėApžiūra,
                    automobiliai[i].Kuras, automobiliai[i].VidutinėsSąnaudos);

                ret.Add(naujas);
            }
        }

        return ret;
    }

    private void RastiDuNaujausius(List<Automobilis> automobiliai, out Automobilis a1, out
Automobilis a2)
    {
        a1 = automobiliai[0];
        a2 = automobiliai[1];

        if (a2 >= a1)
        {
            Automobilis temp = a1;
            a1 = a2;
            a2 = temp;
        }

        for (int i = 2; i < automobiliai.Count; i++)
        {
            Automobilis temp = automobiliai[i];
            if (temp >= a1 && temp >= a2)
            {
                a2 = a1;
                a1 = temp;
            }
            else if (temp >= a2)
            {
                a2 = temp;
            }
        }
    }

    private void Spausdinti_Click(object sender, EventArgs e)
    {
        string txt = File.ReadAllText(rez);
        richTextBox1.Text = txt;
        skaičiavimasToolStripMenuItem.Enabled = false;
    }

    private void Informacija_Click(object sender, EventArgs e)
    {
        label1.Text = "Peržiūrima informacija vartotojui.";
        string x = File.ReadAllText(info);
        MessageBox.Show(x);
    }

    private void textBox1_MouseClick(object sender, MouseEventArgs e)
    {
        textBox1.Text = "";
    }
}

```

3.6. Pradiniai duomenys ir rezultatai

Pradiniai duomenys nr.1:

```

zaibas - Notepad
File Edit Format View Help
ABC123;Nissan; NX; 2004-05; 2018 03-23; Benzinas; 8;
GHJ456;Nissan; Quasquai; 2004-05; 2018 03-23; Benzinas; 7;
IBF781;Chevrolet; TurboX; 2005 07; 2018-10-28; Dyzelinas; 8;
KVK564;Alfa romeo; Turbo10; 2005 07; 2018-10-28; Dyzelinas; 9;
OLE753;Audi; Z4; 2009 06; 2018 04-09; Hibridas; 3;
OFG896;Tesla; ModelX; 2009 06; 2018 04-09; Hibridas; 4;
REP427;Toyota; Auris; 2009 06; 2018 04-09; Hibridas; 3;
KEB001;Mercedes; 300SL; 2011 01; 2018-07-22; Dujos; 8;
ZLM562;Mercedes; W25 Silver Arrow; 2011 06; 2018-08-22; Dujos; 10;
ABC123;Mercedes; W25 Silver Arrow; 2011 01; 2018-12-31; Dujos; 10;
PLM562;Mercedes; W25 Silver Arrow; 2011 04; 2018-06-22; Dujos; 10;
KLG576;Tesla; ModelX; 2017 12; 2018 06 20; Elektra; 0;
ASD653;Tesla; ModelX; 2017 12; 2018 06 20; Elektra; 0;
LEK159;Tesla; ModelX; 2017 11; 2018 06 01; Elektra; 0;

```

Rezultatai nr.1:

Pradiniai duomenys:						
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas
1	ABC123	Nissan	NX	2004 05	2018 03 23	Benzinas
2	GHJ456	Nissan	Quasquai	2004 05	2018 03 23	Benzinas
3	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas
4	KVK564	Alfa romeo	Turbo10	2005 07	2018 10 28	Dyzelinas
5	OLE753	Audi	Z4	2009 06	2018 04 09	Hibridas
6	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas
7	REP427	Toyota	Auris	2009 06	2018 04 09	Hibridas
8	KEB001	Mercedes	300SL	2011 01	2018 07 22	Dujos
9	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos
10	ABC123	Mercedes	W25 Silver Arrow	2011 01	2018 12 31	Dujos
11	PLM562	Mercedes	W25 Silver Arrow	2011 04	2018 06 22	Dujos
12	KLG576	Tesla	ModelX	2017 12	2018 06 20	Elektra
13	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra
14	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra

Naujausias automobilis:						
Valstybinis numeris :	KLG576					
Gamintojas :	Tesla					
Modelis :	ModelX					
Pagaminimo data :	2017 12					
Techninė apžiūra :	2018 06 20					
Kuro tipas :	Elektra					
Kuro sąnaudos :	0					

Antras naujausias automobilis:						
Valstybinis numeris :	ASD653					
Gamintojas :	Tesla					
Modelis :	ModelX					
Pagaminimo data :	2017 12					
Techninė apžiūra :	2018 06 20					
Kuro tipas :	Elektra					
Kuro sąnaudos :	0					

ModelX markės automobiliai:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
2	KL6576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
3	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
4	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0
Surikiuoti ModelX markės automobiliai:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
2	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0
3	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
4	KL6576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
Gamintojų sąrašas:							
Nissan							
Chevrolet							
Alfa romeo							
Audi							
Tesla							
Toyota							
Mercedes							
Automobiliai paseno:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	ABC123	Nissan	NX	2004 05	2018 03 23	Benzinas	8,4
2	GHJ456	Nissan	Quasquai	2004 05	2018 03 23	Benzinas	7,4
3	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas	8,4
4	KVK564	Alfa romeo	Turbo10	2005 07	2018 10 28	Dyzelinas	9,5
5	OLE753	Audi	Z4	2009 06	2018 04 09	Hibridas	3,2
6	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,2
7	REP427	Toyota	Auris	2009 06	2018 04 09	Hibridas	3,2
8	KEB001	Mercedes	300SL	2011 01	2018 07 22	Dujos	8,4
9	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,5
10	ABC123	Mercedes	W25 Silver Arrow	2011 01	2018 12 31	Dujos	10,5
11	PLM562	Mercedes	W25 Silver Arrow	2011 04	2018 06 22	Dujos	10,5
12	KL6576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
13	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
14	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

Pradiniai duomenys nr.2:

```

zaibas1 - Notepad
File Edit Format View Help
ABC123;Nissan; NX; 2004-05; 2018 03-23; Benzinas; 8;
GHJ456;Nissan; Quasquai; 2004-05; 2018 03-23; Benzinas; 7;
IBF781;Chevrolet; TurboX; 2005 07; 2018-10-28; Dyzelinas; 8;
VED951;Chevrolet; TurboX; 2006 08; 2018-11-25; Dujos; 6;
CIK384;Chevrolet; TurboX; 2018 09; 2018-09-21; Benzinas; 7;
KVK564;Alfa romeo; Turbo10; 2005 07; 2018-10-28; Dyzelinas; 9;
OFG896;Tesla; ModelX; 2009 06; 2018 04-09; Hibridas; 4;
KEB001;Mercedes; 300SL; 2011 01; 2018-07-22; Dujos; 8;
ZLM562;Mercedes; W25 Silver Arrow; 2011 06; 2018-08-22; Dujos; 10;
ASD653;Tesla; ModelX; 2017 12; 2018 06 20; Elektra; 0;

```

Rezultatai nr.2:

Pradiniai duomenys:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	ABC123	Nissan	NX	2004 05	2018 03 23	Benzinas	8,0
2	GHJ456	Nissan	Quasquai	2004 05	2018 03 23	Benzinas	7,0
3	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas	8,0
4	VED951	Chevrolet	TurboX	2006 08	2018 11 25	Dujos	6,0
5	CIK384	Chevrolet	TurboX	2018 09	2018 09 21	Benzinas	7,0
6	KVK564	Alfa romeo	Turbo10	2005 07	2018 10 28	Dyzelinas	9,0
7	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
8	KEB001	Mercedes	300SL	2011 01	2018 07 22	Dujos	8,0
9	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,0
10	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
Naujausias automobilis:							
Valstybinis numeris :		CIK384					
Gamintojas :		Chevrolet					
Modelis :		TurboX					
Pagaminimo data :		2018 09					
Techninė apžiūra :		2018 09 21					
Kuro tipas :		Benzinas					
Kuro sąnaudos :		7					
Antras naujausias automobilis:							
Valstybinis numeris :		ASD653					
Gamintojas :		Tesla					
Modelis :		ModelX					
Pagaminimo data :		2017 12					
Techninė apžiūra :		2018 06 20					
Kuro tipas :		Elektra					
Kuro sąnaudos :		0					
TurboX markės automobiliai:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas	8,0
2	VED951	Chevrolet	TurboX	2006 08	2018 11 25	Dujos	6,0
3	CIK384	Chevrolet	TurboX	2018 09	2018 09 21	Benzinas	7,0
Surikiuoti TurboX markės automobiliai:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	CIK384	Chevrolet	TurboX	2018 09	2018 09 21	Benzinas	7,0
2	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas	8,0
3	VED951	Chevrolet	TurboX	2006 08	2018 11 25	Dujos	6,0
Gamintojų sąrašas:							
Nissan							
Chevrolet							
Alfa romeo							
Tesla							
Mercedes							
Automobiliai paseno:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	ABC123	Nissan	NX	2004 05	2018 03 23	Benzinas	8,4
2	GHJ456	Nissan	Quasquai	2004 05	2018 03 23	Benzinas	7,4
3	IBF781	Chevrolet	TurboX	2005 07	2018 10 28	Dyzelinas	8,4
4	VED951	Chevrolet	TurboX	2006 08	2018 11 25	Dujos	6,3
5	CIK384	Chevrolet	TurboX	2018 09	2018 09 21	Benzinas	7,4
6	KVK564	Alfa romeo	Turbo10	2005 07	2018 10 28	Dyzelinas	9,5
7	OFG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,2
8	KEB001	Mercedes	300SL	2011 01	2018 07 22	Dujos	8,4
9	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,5
10	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0

4. Polimorfizmas (L3)

4.1. Darbo užduotis

U5_2. Automobilių parkas. VšĮ „Greitis“ plečiasi ir atidaro naujus filialus. Pirmoje eilutėje yra miestas, antroje – 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 mėnesis. Jei techninės apžiūros galiojimas pasibaigęs, tuos automobilius pažymėkite papildomais ženklais.

4.2. Grafinės vartotojo sąsajos schema ir paveikslas

Schema:



Meniu punktai:

Failas	Skaičiavimas	Pagalba
Įvesti	Geriausieji automobiliai	Nurodymai vartotojui
Spausdinti	Seniausieji mikroautobusai	Naudojimo sąlygos
Baigti	Krovininiai automobiliai	
	Techninė apžiūra	

Paveikslas:

Automobilių parkas									
Failas Skaiciavimas Pagalba									
Pradiniai duomenys: Vilnius Gerdimino pr. 50 vilnius@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšiuos data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	FGH567	Toyota	Auris	2015 12	2017 07 21	elektra	4,90	184567
2	Krovininis	JKL901	Seat	HQ	2010 04	2017 11 21	dyzelinas	7,10	490
3	Mikroautobusas	DFF267	Opel	Astra	2010 05	2017 11 13	dujos	8,20	15
4	Mikroautobusas	LMN123	Alfa Romeo	200	2000 08	2018 02 12	benzinas	8,90	17
5	Krovininis	KLM012	Ford	A1	2008 05	2017 12 29	benzinas	6,80	560
6	Lengvasis	IJR903	Nissan	NX	2003 12	2017 12 22	dujos	7,50	116587
7	Mikroautobusas	MNO234	Subaru	X10	2004 11	2018 04 16	dyzelinas	7,60	18
8	Lengvasis	EFG456	BMW	X1	2004 08	2017 05 31	dyzelinas	6,50	164570
Kaunas Ramunėlių g. 123 kaunas@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšiuos data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Mikroautobusas	NOP345	Mini	Cooper	2006 02	2018 06 19	dujos	5,80	17
2	Lengvasis	JKL951	Toyota	Yaris	2005 01	2017 07 17	benzinas	7,40	161753
3	Krovininis	DES564	Renault	Megane	2009 10	2017 01 30	dyzelinas	8,10	750
4	Lengvasis	DCN631	Jeep	F10	2008 03	2017 05 30	dyzelinas	8,20	95614
5	Krovininis	ABE152	Mercedes	H16	2015 11	2017 08 08	dujos	6,90	610
6	Mikroautobusas	UPS456	SAAB	X	2007 06	2018 05 02	dyzelinas	7,30	18
7	Krovininis	PKM856	Mercedes	X3	2011 01	2017 10 04	benzinas	8,50	590
Klaipėda Jūros g. 12 klaipeda@greitis.lt Automobiliai:									
Duomenys atspausdinti.									

Automobilių parkas									
Failas Skaiciavimas Pagalba									
Klaipėda Jūros g. 12 klaipeda@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšiuos data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	CDE234	Audi	E92	2006 05	2016 01 11	dujos	8,10	201333
2	Krovininis	IJK890	Peugeot	3008	2005 03	2017 10 13	dujos	7,90	650
3	Lengvasis	DEF345	Mercedes	Benz	2005 01	2017 04 06	benzinas	8,30	104506
4	Mikroautobusas	PSE456	Volkswagen	Jetta	2000 08	2018 04 16	benzinas	8,60	16
5	Mikroautobusas	KLT528	Lexus	10	2006 08	2017 01 31	dyzelinas	8,60	15
6	Krovininis	HIJ789	Opel	Astra	1999 02	2017 09 22	dyzelinas	8,90	500
Jonava Sodų g. 12 jonava@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšiuos data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	ABC012	Toyota	Avensis	2000 08	2018 07 16	dyzelinas	8,60	123456
2	Lengvasis	BCD123	Volkswagen	Golf	2001 07	2018 08 20	benzinas	8,20	90456
3	Krovininis	GHI678	Nissan	Qashqai	1998 01	2018 02 07	dujos	7,90	500
4	Mikroautobusas	UPS456	Skoda	X	2000 08	2017 03 09	dyzelinas	8,10	16
Duomenys apie geriausius automobilius: Lengvieji: Gamintojas: Volkswagen; Modelis: Golf; Valstybinis numeris: BCD123; Amžius: 17; Papildomas rodiklis: 90456; Krovininiai: Gamintojas: Renault;									
Duomenys atspausdinti.									

Automobilių parkas									
Failas Skaiciavimas Pagalba									
Krovininiai: Gamintojas: Renault; Modelis: Megane; Valstybinis numeris: DES564; Amžius: 9; Papildomas rodiklis: 750; Mikroautobusai: Gamintojas: Subaru; Modelis: X10; Valstybinis numeris: MNO234; Amžius: 14; Papildomas rodiklis: 18; Gamintojas: SAAB; Modelis: X; Valstybinis numeris: UPS456; Amžius: 11; Papildomas rodiklis: 18; Duomenys apie filialus, kuriuose mikroautobusai seniausi: Vilnius Gerdimino pr. 50 vilnius@greitis.lt Klaipėda Jūros g. 12 klaipeda@greitis.lt Filialų krovininių automobilių sąrašai, surikiuoti pagal gamintoją ir modelį: Vilnius Gerdimino pr. 50 vilnius@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšiuos data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	DEK012	Renault	X1	2008 05	2017 12 28	benzinas	6,80	520
Duomenys atspausdinti.									

Automobilių parkas

Failas

Skačiavimas

Pagalba

Filių krovinių automobilių sąrašai, surikiuoti pagal gamintoją ir modelį:

Vilnius
Gedimino pr. 50
vilnius@greitis.lt
Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšildros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	KLM012	Ford	Al	2008 05	2017 12 29	benzinas	6,80	560
2	Krovininis	JKL901	Seat	HQ	2010 04	2017 11 21	dyzelinas	7,10	490

Kaunas
Ramunėlių g. 123
kaunas@greitis.lt
Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšildros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	ADE152	Mercedes	H16	2015 11	2017 08 09	dujos	6,90	610
2	Krovininis	PKM856	Mercedes	X3	2011 01	2017 10 04	benzinas	8,50	590
3	Krovininis	DESS64	Renault	Megane	2009 10	2017 01 30	dyzelinas	8,10	750

Klaipėda
Jūros g. 12
klaipeda@greitis.lt
Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšildros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	HIJ789	Opel	Astra	1999 02	2017 09 22	dyzelinas	8,90	500
2	Krovininis	IJK890	Peugeot	3008	2005 03	2017 10 13	dujos	7,90	650

Jonava
Sodų g. 12
jonava@greitis.lt
Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšildros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	GHI678	Nissan	Qashqai	1998 01	2018 02 07	dujos	7,90	500

Duomenys atspausdinti.

Automobilių parkas

Failas

Skačiavimas

Pagalba

1	Krovininis	KLM012	Ford	Al	2008 05	2017 12 29	benzinas	6,80	560
2	Krovininis	JKL901	Seat	HQ	2010 04	2017 11 21	dyzelinas	7,10	490

Kaunas
Ramunėlių g. 123
kaunas@greitis.lt
Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšildros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	ADE152	Mercedes	H16	2015 11	2017 08 09	dujos	6,90	610
2	Krovininis	PKM856	Mercedes	X3	2011 01	2017 10 04	benzinas	8,50	590
3	Krovininis	DESS64	Renault	Megane	2009 10	2017 01 30	dyzelinas	8,10	750

Klaipėda
Jūros g. 12
klaipeda@greitis.lt
Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšildros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	HIJ789	Opel	Astra	1999 02	2017 09 22	dyzelinas	8,90	500
2	Krovininis	IJK890	Peugeot	3008	2005 03	2017 10 13	dujos	7,90	650

Jonava
Sodų g. 12
jonava@greitis.lt
Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apšildros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	GHI678	Nissan	Qashqai	1998 01	2018 02 07	dujos	7,90	500

Duomenys atspausdinti.

4.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
Form1	Text: Automobilių parkas	Programos langas
richTextBox1 (tipas RichTextBox)	Font: Courier New, Regular, 8; Script: Baltic;	Atvaizduoti pradinį duomenų ir skaičiavimų turinį.
label1 (tipas Label)	Text: Pranešimas Font: Courier New, Bold, 12; Script: Baltic; ForeColor: Highlight	Spausdina pranešimus apie programos veikimą.
menuStrip1 (tipas MenuStrip)	Font: Standartiniai nustatymai	Naudojamas meniu punktu, kurie bus naudojami programos veikimo metu, išdėstymui.
failasToolStripMenuItem	Text: Failas	Meniu punktas, kuriame yra

(tipas ToolStripMenuItem)	Font: Standartiniai nustatymai	mygtukai duomenų nuskaitymui, atliktų skaičiavimų spausdinimui ir darbo su programa baigimui.
ivestiToolStripMenuItem (tipas ToolStripMenuItem)	Name: nuskaityti Text: Nuskaityti Font: Standartiniai nustatymai	Meniu punktas duomenims iš failo nuskaityti.
spausdintiToolStripMenuItem (tipas ToolStripMenuItem)	Name: spausdinti Text: Spausdinti Font: Standartiniai nustatymai	Meniu punktas gautiems rezultatams atspausdinti.
baigtiToolStripMenuItem (tipas ToolStripMenuItem)	Name: baigti Text: Baigti Font: Standartiniai nustatymai	Meniu punktas, baigiantis darbą su programa.
skaičiuotiToolStripMenuItem (tipas ToolStripMenuItem)	Text: Skaičiavimas Font: Standartiniai nustatymai	Meniu punktas skaičiavimams atlikti.
geriausiAutoToolStripMenuItem (tipas ToolStripMenuItem)	Name: geriausiAuto Text: Geriausi automobiliai Font: Standartiniai nustatymai	Meniu punktas, randantis geriausius automobilius kiekvienoje kategorijoje.
seniausiMikroautobusaiToolStripMenuItem (tipas ToolStripMenuItem)	Name: seniausiMikroautobusai Text: Seniausi mikroautobusai Font: Standartiniai nustatymai	Meniu punktas, randantis seniausius mikroautobusus.
krovininiaiAutomobiliaiToolStripMenuItem (tipas ToolStripMenuItem)	Name: krovininiaiAutomobiliai Text: Krovininiai automobiliai Font: Standartiniai nustatymai	Meniu punktas, randantis kiekvieno filialo krovininius automobilius.
techninėApžiūraToolStripMenuItem (tipas ToolStripMenuItem)	Name: techninėApžiūra Text: Techninė apžiūra Font: Standartiniai nustatymai	Meniu punktas, randantis automobilių techninės apžiūros galiojimo pabaigos datas.
pagalbaToolStripMenuItem (tipas ToolStripMenuItem)	Text: Pagalba Font: Standartiniai nustatymai	Meniu punktas, turintis mygtukus, kurie atvaizduoja nurodymus vartotojui ir naudojimo sąlygas.
nurodymaiVartotojuiToolStripMenuItem (tipas ToolStripMenuItem)	Name: nurodymaiVartotojui Text: Nurodymai vartotojui Font: Standartiniai nustatymai	Meniu punktas, spausdinantis nurodymus vartotojui.
naudojimoSąlygosToolStripMenuItem (tipas ToolStripMenuItem)	Name: naudojimoSąlygos Text: Naudojimo sąlygos Font: Standartiniai nustatymai	Meniu punktas, spausdinantis naudojimo sąlygas.

4.4. Programos vartotojo vadovas

1. Paleiskite programą;
2. Atsidariusiame lange spauskite Failas -> Nuskaityti;
3. Norėdami spausdinti, spauskite Failas -> Spausdinti;
4. Toliau galite atlikti skaičiavimus:
 - 4.1 Norėdami rasti geriausius automobilius, spauskite Skaičiavimas -> Geriausi automobiliai;
 - 4.2 Norėdami rasti seniausius mikroautobusus, spauskite Skaičiavimas -> Seniausi mikroautobusai;
 - 4.3 Norėdami rasti filialų krovinius automobilius, spauskite Skaičiavimas -> Krovininiai automobiliai;
 - 4.4 Norėdami rasti automobilių technines apžiūras, spauskite Skaičiavimas -> Techninė apžiūra;
 - 4.5 PASTABA: šio skaičiavimo duomenys ekrane nespausdinami, juos galite rasti faile "Apžiūra.txt";
5. Norėdami peržiūrėti pagalbą vartotojui, spauskite Pagalba -> Nurodymai vartotojui;
6. Norėdami peržiūrėti naudojimo sąlygas, spauskite Pagalba -> Naudojimo sąlygos;
7. Norėdami baigti darbą su programa, spauskite Failas -> Baigti.

4.5. Programos tekstas

Abstrakti bazinė klasė Transportas:

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

namespace AutomobiliųParkas
{
    abstract class Transportas : IComparable<Transportas>
    {
        public string ValstybinisNumeris { get; set; }
        public string Gamintojas { get; set; }
        public string Modelis { get; set; }
        public DateTime PagaminimoData { get; set; }
        public DateTime TechninėApžiūra { get; set; }
        public string Kuras { get; set; }
        public double VidutinėsSąnaudos { get; set; }

        public Transportas(string valstNr, string gam, string mod, DateTime pag, DateTime
tech, string kur, double vid)
        {
            this.ValstybinisNumeris = valstNr;
            this.Gamintojas = gam;
            this.Modelis = mod;
            this.PagaminimoData = pag;
            this.TechninėApžiūra = tech;
            this.Kuras = kur;
            this.VidutinėsSąnaudos = vid;
        }

        public abstract double PapildomasRodiklis();

        public string TechninėApžiūraToString(DateTime techApžiūrosGaliojimoPabaiga)
        {
            return string.Format("{0}; {1}; {2} {3:yyyy MM dd}",
```

```

        this.Gamintojas,
        this.Modelis,
        this.ValstybinisNumeris,
        techApžiūrosGaliojimoPabaiga
    );
}

public override string ToString()
{
    string type = this is Lengvasis ? "Lengvasis" : this is Krovininis ? "Krovininis"
: "Mikroautobusas";
    return string.Format(
        "{0,-15} {1,-10} {2,-15} {3,-15} {4:yyyy MM} {5:yyyy MM dd} {6,-10} {7,4:f2}",
        type, ValstybinisNumeris,
        Gamintojas,
        Modelis,
        PagaminimoData,
        TechninėApžiūra,
        Kuras,
        VidutinėsSąnaudos
    );
}

public int CompareTo(Transportas other)
{
    if (this is Lengvasis && (other is Krovininis || other is Mikroautobusas))
        return 1;
    if (this is Krovininis && other is Lengvasis)
        return -1;
    if (this is Krovininis && other is Mikroautobusas)
        return 1;
    if (this is Mikroautobusas && (other is Lengvasis || other is Krovininis))
        return -1;
    if (this is Lengvasis && other is Lengvasis)
        return ((Lengvasis)this).CompareTo((Lengvasis)other);
    if (this is Krovininis && other is Krovininis)
        return ((Krovininis)this).CompareTo((Krovininis)other);
    if (this is Mikroautobusas && other is Mikroautobusas)
        return ((Mikroautobusas)this).CompareTo((Mikroautobusas)other);

    return 0;
}
}
}

```

Išvestinė klasė Lengvasis:

```

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

namespace AutomobiliųParkas
{
    class Lengvasis : Transportas, IComparable<Lengvasis>
    {
        public double OdometroRodmenys { get; set; }
    }
}

```



```

        public Lengvasis(string valstNr, string gam, string mod, DateTime pag, DateTime tech,
string kur, double vid, double odom) : base(valstNr, gam, mod, pag, tech, kur, vid)
        {
            this.OdometroRodmenys = odom;
        }

        public override double PapildomasRodiklis()
        {
            return this.OdometroRodmenys;
        }

        public override string ToString()
        {
            return base.ToString() + $"{OdometroRodmenys}";
        }

        public int CompareTo(Lengvasis other)
        {
            return -OdometroRodmenys.CompareTo(other.OdometroRodmenys);
        }
    }
}

```

Išvestinė klasė Krovininis:

```

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

namespace AutomobiliųParkas
{
    class Krovininis : Transportas, IComparable<Krovininis>
    {
        public double PriekabosTalpa { get; set; }

        public Krovininis(string valstNr, string gam, string mod, DateTime pag, DateTime tech,
string kur, double vid, double talp) : base(valstNr, gam, mod, pag, tech, kur, vid)
        {
            this.PriekabosTalpa = talp;
        }

        public override double PapildomasRodiklis()
        {
            return this.PriekabosTalpa;
        }

        public override string ToString()
        {
            return base.ToString() + $"{PriekabosTalpa}";
        }

        public int CompareTo(Krovininis other)
        {
            return PriekabosTalpa.CompareTo(other.PriekabosTalpa);
        }

        public static bool operator <(Krovininis k1, Krovininis k2)
        {
            int poz1 = String.Compare(k1.Gamintojas, k2.Gamintojas,
StringComparison.CurrentCulture);
            int poz2 = String.Compare(k1.Modelis, k2.Modelis,
StringComparison.CurrentCulture);
            return ((poz1 < 0) || ((poz1 == 0) && (poz2 < 0)));
        }
    }
}

```

```

        public static bool operator >(Krovininis k1, Krovininis k2)
        {
            int poz1 = String.Compare(k1.Gamintojas, k2.Gamintojas,
StringComparison.CurrentCulture);
            int poz2 = String.Compare(k1.Modelis, k2.Modelis,
StringComparison.CurrentCulture);
            return ((poz1 > 0) || ((poz1 == 0) && (poz2 > 0)));
        }
    }
}

```

Išvestinė klasė Mikroautobusas:

```

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

namespace AutomobiliųParkas
{
    class Mikroautobusas : Transportas, IComparable<Mikroautobusas>
    {
        public int SėdimųVietų { get; set; }

        public Mikroautobusas(string valstNr, string gam, string mod, DateTime pag, DateTime
tech, string kur, double vid, int vietu) : base(valstNr, gam, mod, pag, tech, kur, vid)
        {
            this.SėdimųVietų = vietu;
        }

        public override double PapildomasRodiklis()
        {
            return this.SėdimųVietų;
        }

        public override string ToString()
        {
            return base.ToString() + $"{SėdimųVietų}";
        }

        public int CompareTo(Mikroautobusas other)
        {
            return SėdimųVietų.CompareTo(other.SėdimųVietų);
        }
    }
}

```

Klasė FilialoDuomenys:

```

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

namespace AutomobiliųParkas
{
    class FilialoDuomenys
    {
        private string miestoPavadinimas;
        private string adresas;
        private string elPastas;

        public FilialoDuomenys(string miestas, string adresas, string pastas)
        {
            this.miestoPavadinimas = miestas;
            this.adresas = adresas;
            this.elPastas = pastas;
        }
    }
}

```

```

    }

    public override string ToString()
    {
        return $"{miestoPavadinimas}\n{adresas}\n{elPastas}\n";
    }
}

}

Klasė Filialas:
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AutomobiliųParkas
{
    class Filialas
    {
        private FilialoDuomenys filialas;
        private List<Transportas> automobiliai;

        public Filialas(FilialoDuomenys filialas, List<Transportas> automobiliai)
        {
            this.filialas = filialas;
            this.automobiliai = automobiliai;
        }

        private string PrintCars()
        {
            string ret = "";
            for (int i = 0; i < automobiliai.Count; i++)
            {
                ret += "    " + (i + 1).ToString() + automobiliai[i].ToString() + "\n";
            }

            return ret;
        }

        public FilialoDuomenys FilialoDuomenys()
        {
            return this.filialas;
        }

        public override string ToString()
        {
            const string header = "Automobiliai:\r\n" +
                "-----\r\n" +
                "Nr.    Tipas    Valstybinis numeris    Gamintojas\r\n" +
                "Modelis    Pagaminimo data    Techninės apžiūros data    Kuras    Vidutinės\r\n" +
                "sąnaudos/100km    Papildomas rodiklis \r\n" +
                "-----\r\n";

            return filialas.ToString() + header + this.PrintCars() + "-----\r\n";
        }

        public List<Transportas> FilialoAutomobiliai()
        {
            return automobiliai;
        }
    }
}

```

```
}
```

Programos kodas:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace AutomobiliųParkas
{
    public partial class Form1 : Form
    {
        const string data = "..\\..\\Miestai.txt";
        const string output = "..\\..\\output.txt";
        const string apžiūra = "..\\..\\Apžiūra.txt";
        const string pagalba = "..\\..\\NurodymaiVartotojui.txt";
        const string salygos = "..\\..\\NaudojimoSalygos.txt";
        Dictionary<string, Filialas> filialai;
        List<Transportas> automobiliai;

        public Form1()
        {
            InitializeComponent();

            if (File.Exists(output))
                File.Delete(output);
            if (File.Exists(apžiūra))
                File.Delete(apžiūra);

            spausdinti.Enabled = false;
            label1.Text = "Nuskaitykite failus.";
        }

        private void nuskaityti_Click(object sender, EventArgs e)
        {
            filialai = Read(data);
            automobiliai = Formuoti(filialai);
            Print(output, filialai, "Pradiniai duomenys:");
            nuskaityti.Enabled = false;
            spausdinti.Enabled = true;
            label1.Text = "Duomenys nuskaityti, galima spausdinti.";
        }

        private void spausdinti_Click(object sender, EventArgs e)
        {
            string text = File.ReadAllText(output);
            richTextBox1.Text = text;
            spausdinti.Enabled = false;
            label1.Text = "Duomenys atspausdinti.";
        }

        private void geriausiAuto_Click(object sender, EventArgs e)
        {
            Lengvasis geriausiasLengvasis = (Lengvasis)RastiGeriausiaGrupeje(automobiliai,
            typeof(Lengvasis));
            Krovininis geriausiasKrovininis = (Krovininis)RastiGeriausiaGrupeje(automobiliai,
            typeof(Krovininis));
            Mikroautobusas geriausiasMikroautobusas =
            (Mikroautobusas)RastiGeriausiaGrupeje(automobiliai, typeof(Mikroautobusas));

            printText(output, "Duomenys apie geriausius automobilius:");
        }
    }
}
```

```

        printText(output, "Lengvieji:");
        SpausdintiGeriausius(automobiliai, geriausiasLengvasis, typeof(Lengvasis));
        printText(output, "Krovininiai:");
        SpausdintiGeriausius(automobiliai, geriausiasKrovininis, typeof(Krovininis));
        printText(output, "Mikroautobusai:");
        SpausdintiGeriausius(automobiliai, geriausiasMikroautobusas,
typeof(Mikroautobusas));

        label1.Text = "Geriausi automobiliai surasti, galima spausdinti.";
        spausdinti.Enabled = true;
    }

    private void seniausioMikroautobusai_Click(object sender, EventArgs e)
    {
        double averageAge = SeniausiasFilialas(filialai);
        printText(output, "Duomenys apie filialus, kuriuose mikroautobusai seniausi:");
        SpausdintiSeniausiasFilialus(filialai, averageAge);

        label1.Text = "Filialai su seniausiais mikroautobusais surasti, galima
spausdinti.";
        spausdinti.Enabled = true;
    }

    private void krovininiaiAutomobiliai_Click(object sender, EventArgs e)
    {
        Dictionary<string, Filialas> krovininiaiFilialai =
KrovininiaiAutomobiliai(filialai);
        Print(output, krovininiaiFilialai, "Filialų krovininių automobilių sąrašai,
surikiuoti pagal gamtintoją ir modelį:");

        label1.Text = "Filialų krovininių automobilių sąrašai surasti, galima
spausdinti.";
        spausdinti.Enabled = true;
    }

    private void techninėApžiūra_Click(object sender, EventArgs e)
    {
        TechninėApžiūra(automobiliai);
        label1.Text = "Automobilių techninių apžiūrų datos surastos, jas galite rasti
faile Apžiūra.txt.";
    }

    private void nurodymaiVartotojui_Click(object sender, EventArgs e)
    {
        string text = File.ReadAllText(pagalba);
        MessageBox.Show(text);
    }

    private void naudojimoSąlygos_Click(object sender, EventArgs e)
    {
        string text = File.ReadAllText(salygos);
        MessageBox.Show(text);
    }

    private void baigti_Click(object sender, EventArgs e)
    {
        Close();
    }

    static void TechninėApžiūra(List<Transportas> automobiliai)
    {
        TimeSpan twoYearSpan = DateTime.Now.AddYears(2).Subtract(DateTime.Now);
        TimeSpan yearSpan = DateTime.Now.AddYears(1).Subtract(DateTime.Now);
        TimeSpan halfYearSpan = DateTime.Now.AddMonths(6).Subtract(DateTime.Now);
        for (int i = 0; i < automobiliai.Count; i++)
        {

```

```

        DateTime techninesApziurosPabaiga = automobiliai[i].TechninėApžiūra +
(automobiliai[i] is Lengvasis ? twoYearSpan : (automobiliai[i] is Krovininis ? yearSpan :
halfYearSpan));
        if (techninesApziurosPabaiga < DateTime.Now)
        {
            printText(apziura,
automobiliai[i].TechninėApžiūraToString(techninesApziurosPabaiga) + " TECHNINĖ APŽIŪRA
PASIBAIGUSI");
        }
        if (techninesApziurosPabaiga > DateTime.Now && techninesApziurosPabaiga <
DateTime.Now.AddMonths(1))
        {
            printText(apziura,
automobiliai[i].TechninėApžiūraToString(techninesApziurosPabaiga));
        }
    }
}

static Dictionary<string, Filialas> KroviniaiAutomobiliai(Dictionary<string, Filialas>
filialai)
{
    Dictionary<string, Filialas> krovininiaiFilialai = new Dictionary<string,
Filialas>();
    foreach (KeyValuePair<string, Filialas> entry in filialai)
    {
        List<Transportas> kroviniaiAuto =
filialai[entry.Key].FilialoAutomobiliai().Where(x => x is Krovininis).ToList();
        Sort(kroviniaiAuto);
        Filialas naujasFilialas = new Filialas(filialai[entry.Key].FilialoDuomenys(),
kroviniaiAuto);
        krovininiaiFilialai.Add(entry.Key, naujasFilialas);
    }

    return krovininiaiFilialai;
}

static void Sort(List<Transportas> filialoAutomobiliai)
{
    int ind;
    for (int i = 0; i < filialoAutomobiliai.Count - 1; i++)
    {
        ind = i;
        for (int j = i + 1; j < filialoAutomobiliai.Count; j++)
        {
            if ((Krovininis)filialoAutomobiliai[j] <
(Krovininis)filialoAutomobiliai[ind])
            {
                ind = j;
            }
        }
        var tempObj = filialoAutomobiliai[i];
        filialoAutomobiliai[i] = filialoAutomobiliai[ind];
        filialoAutomobiliai[ind] = tempObj;
    }
}

static void SpausdintiSeniausiusFilialus(Dictionary<string, Filialas> filialai, double
vidAmz)
{
    foreach(KeyValuePair<string, Filialas> entry in filialai)
    {
        if (VidutinisAmžiusFiliale(filialai[entry.Key].FilialoAutomobiliai()) ==
vidAmz)
        {
            printText(output, filialai[entry.Key].FilialoDuomenys().ToString());
        }
    }
}

```

```

static double SeniausiasFilialas(Dictionary<string, Filialas> filialai)
{
    List<string> keys = filialai.Keys.ToList();
    double vidutinisAmžius =
VidutinisAmžiusFiliale(filialai[keys[0]].FilialoAutomobiliai());
    for (int i = 1; i < keys.Count; i++)
    {
        double naujasVid =
VidutinisAmžiusFiliale(filialai[keys[i]].FilialoAutomobiliai());
        if (naujasVid > vidutinisAmžius)
        {
            vidutinisAmžius = naujasVid;
        }
    }

    return vidutinisAmžius;
}

static double VidutinisAmžiusFiliale(List<Transportas> automobiliai)
{
    double suma = 0;
    for (int i = 0; i < automobiliai.Count; i++)
    {
        if (automobiliai[i] is Mikroautobusas)
            suma += (DateTime.Now.Year - automobiliai[i].PagaminimoData.Year);
    }

    return suma / automobiliai.Count;
}

static void SpausdintiGeriausius(List<Transportas> automobiliai, Transportas
geriausias, Type tipas)
{
    double papildomasRodiklis = geriausias.PapildomasRodiklis();
    for (int i = 0; i < automobiliai.Count; i++)
    {
        if (automobiliai[i].GetType() == tipas && automobiliai[i].PapildomasRodiklis()
== papildomasRodiklis)
            printText(output, "Gamintojas: " + automobiliai[i].Gamintojas +
";\nModelis: " + automobiliai[i].Modelis + "; \nValstybinis numeris: " +
automobiliai[i].ValstybinisNumeris + "; \nAmžius: " + (DateTime.Now.Year -
automobiliai[i].PagaminimoData.Year) + "; \nPapildomas rodiklis: " +
automobiliai[i].PapildomasRodiklis() + "; \n");
    }
}

static Transportas RastiGeriausiaGrupeje(List<Transportas> automobiliai, Type tipas)
{
    Transportas geriausiasAuto = automobiliai[0];
    int i;
    for (i = 0; i < automobiliai.Count; i++)
    {
        if (automobiliai[i].GetType() == tipas)
        {
            geriausiasAuto = automobiliai[i];
            break;
        }
    }

    for (int j = i + 1; j < automobiliai.Count; j++)
    {
        if (automobiliai[j].GetType() == tipas &&
automobiliai[j].CompareTo(geriausiasAuto) == 1)
        {
            geriausiasAuto = automobiliai[j];
        }
    }
}

```

```

    }

    return geriausiasAuto;
}

static List<Transportas> Formuoti(Dictionary<string, Filialas> automobiliai)
{
    List<Transportas> naujiAuto = new List<Transportas>();
    foreach (KeyValuePair<string, Filialas> entry in automobiliai)
    {
        foreach(Transportas auto in automobiliai[entry.Key].FilialoAutomobiliai())
        {
            naujiAuto.Add(auto);
        }
    }

    return naujiAuto;
}

static Dictionary<string, Filialas> Read(string data)
{
    Dictionary<string, Filialas> tempDict = new Dictionary<string, Filialas>();
    string[] filialai = File.ReadAllLines(data);
    foreach(string miestas in filialai)
    {
        using (StreamReader reader = new StreamReader("../\\..\\\" + miestas + ".txt"))
        {
            string miestoPav, adresas, elPastas;
            string line;
            miestoPav = reader.ReadLine();
            adresas = reader.ReadLine();
            elPastas = reader.ReadLine();

            string ValstybinisNumeris;
            string Gamintojas;
            string Modelis;
            DateTime PagaminimoData;
            DateTime TechninėApžiūra;
            string Kuras;
            double VidutinėsSąnaudos;

            List<Transportas> automobiliai = new List<Transportas>();

            while ((line = reader.ReadLine()) != null)
            {
                string[] parts = line.Split(';');
                ValstybinisNumeris = parts[1].Trim();
                Gamintojas = parts[2].Trim();
                Modelis = parts[3].Trim();
                PagaminimoData = DateTime.Parse(parts[4].Trim());
                TechninėApžiūra = DateTime.Parse(parts[5].Trim());
                Kuras = parts[6].Trim();
                VidutinėsSąnaudos = double.Parse(parts[7].Trim());

                switch (parts[0].Trim())
                {
                    case "l":
                        double OdometroRodmenys = double.Parse(parts[8].Trim());
                        var naujasAutomobilisL = new Lengvasis(ValstybinisNumeris,
Gamintojas, Modelis, PagaminimoData, TechninėApžiūra, Kuras, VidutinėsSąnaudos,
OdometroRodmenys);

                        automobiliai.Add(naujasAutomobilisL);
                        break;
                    case "k":
                        double PriekabosTalpa = double.Parse(parts[8].Trim());
                        var naujasAutomobilisK = new Krovininis(ValstybinisNumeris,
Gamintojas, Modelis, PagaminimoData, TechninėApžiūra, Kuras, VidutinėsSąnaudos,
PriekabosTalpa);

```



```

        automobiliai.Add(naujasAutomobilisK);
        break;
    case "m":
        int SėdimųVietų = int.Parse(parts[8].Trim());
        var naujasAutomobilisM = new
Mikroautobusas(ValstybinisNumeris, Gamintojas, Modelis, PagaminimoData, TechninėApžiūra,
Kuras, VidutinėsSąnaudos, SėdimųVietų);
        automobiliai.Add(naujasAutomobilisM);
        break;
    default:
        printText(output, "Klaida nuskaityt duomenis!\n");
        break;
    }
}

FilialoDuomenys naujoFilialoDuomenys = new FilialoDuomenys(miestoPav,
adresas, elPastas);
Filialas naujasFilialas = new Filialas(naujoFilialoDuomenys,
automobiliai);
tempDict.Add(miestas, naujasFilialas);
}
}

return tempDict;
}

static void printText(string file, string text)
{
    using (StreamWriter writer = new StreamWriter(file, true))
    {
        writer.WriteLine(text);
    }
}

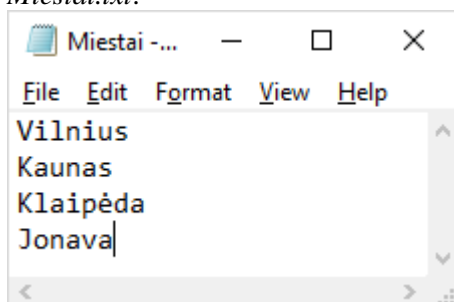
static void Print(string file, Dictionary<string, Filialas> filialai, string title)
{
    using (StreamWriter writer = new StreamWriter(file, true))
    {
        writer.WriteLine(title);
        foreach (KeyValuePair<string, Filialas> entry in filialai)
        {
            writer.WriteLine(filialai[entry.Key].ToString());
        }
    }
}
}
}
}

```

4.6. Pradiniai duomenys ir rezultatai

Pradiniai duomenys nr.1:

Miestai.txt:



Vilnius.txt:

```
Vilnius - Notepad
File Edit Format View Help
Vilnius
Gerdimino pr. 50
vilnius@greitis.lt
l; FGH567; Toyota; Auris; 2015 12; 2017-07-21; elektra; 4,9; 184567;
k; JKL901; Seat; HQ; 2010 04; 2017-11-21; dyzelinas; 7,1; 490;
m; DFE267; Opel; Astra; 2010 05; 2017-11-13; dujos; 8,2; 15;
m; LMN123; Alfa Romeo; 200; 2000 08; 2018-02-12; benzinas; 8,9; 17;
k; KLM012; Ford; A1; 2008 05; 2017-12-29; benzinas; 6,8; 560;
l; IJR983; Nissan; NX; 2003 12; 2017-12-22; dujos; 7,5; 116587;
m; MNO234; Subaru; X10; 2004 11; 2018-04-16; dyzelinas; 7,6; 18;
l; EFG456; BMW; X1; 2004 08; 2017-05-31; dyzelinas; 6,5; 164570;
```

Kaunas.txt:

```
Kaunas - Notepad
File Edit Format View Help
Kaunas
Ramunėlių g. 123
kaunas@greitis.lt
m; NOP345; Mini; Cooper; 2006 02; 2018-06-19; dujos; 5,8; 17;
l; JKL951; Toyota; Yaris; 2005 01; 2017-07-17; benzinas; 7,4; 161753;
k; DES564; Renault; Megane; 2009 10; 2017-01-30; dyzelinas; 8,1; 750;
l; DCN631; Jeep; F10; 2008 03; 2017-05-30; dyzelinas; 8,2; 95614;
k; ADE152; Mercedes; H16; 2015 11; 2017-08-08; dujos; 6,9; 610;
m; UPS456; SAAB; X; 2007 06; 2018-05-02; dyzelinas; 7,3; 18;
k; PKM856; Mercedes; X3; 2011 01; 2017-10-04; benzinas; 8,5; 590;
```

Klaipėda.txt:

```
Klaipėda - Notepad
File Edit Format View Help
Klaipėda
Jūros g. 12
klaipeda@greitis.lt
l; CDE234; Audi; E92; 2006 05; 2016-01-11; dujos; 8,1; 201333;
k; IJK890; Peugeot; 3008; 2005 03; 2017-10-13; dujos; 7,9; 650;
l; DEF345; Mercedes; Benz; 2005 01; 2017-04-06; benzinas; 8,3; 104506;
m; PSE456; Volkswagen; Jetta; 2000 08; 2018-04-16; benzinas; 8,6; 16;
m; KLT528; Lexus; 10; 2006 08; 2017-01-31; dyzelinas; 8,6; 15;
k; HIJ789; Opel; Astra; 1999 02; 2017-09-22; dyzelinas; 8,9; 500;
```

Jonava.txt:

```
Jonava - Notepad
File Edit Format View Help
Jonava
Sodų g. 12
jonava@greitis.lt
1; ABC012; Toyota; Avensis; 2000 08; 2018-07-16; dyzelinas; 8,6; 123456;
1; BCD123; Volkswagen; Golf; 2001 07; 2018-08-20; benzinas; 8,2; 90456;
k; GHI678; Nissan; Qashqai; 1998 01; 2018-02-07; dujos; 7,9; 500;
m; UPS456; Skoda; X; 2000 08; 2017-03-09; dyzelinas; 8,1; 16;
```

Rezultatai nr.1:

output.txt:

Pradiniai duomenys: Vilnius Gerdimino pr. 50 vilnius@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	FGH567	Toyota	Auris	2015 12	2017 07 21	elektra	4,90	184567
2	Krovininis	JKL901	Seat	HQ	2010 04	2017 11 21	dyzelinas	7,10	490
3	Mikroautobusas	DEF267	Opel	Astra	2010 05	2017 11 13	dujos	8,20	15
4	Mikroautobusas	LMI123	Alfa Romeo	200	2000 08	2018 02 12	benzinas	8,90	17
5	Krovininis	KLM012	Ford	A1	2008 05	2017 12 29	benzinas	6,80	560
6	Lengvasis	IDR983	Nissan	NX	2003 12	2017 12 22	dujos	7,50	116587
7	Mikroautobusas	MNO234	Subaru	X10	2004 11	2018 04 16	dyzelinas	7,60	18
8	Lengvasis	EFG456	BMW	X1	2004 08	2017 05 31	dyzelinas	6,50	164570
Kaunas Ramunėlių g. 123 kaunas@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Mikroautobusas	NOP345	Mini	Cooper	2006 02	2018 06 19	dujos	5,80	17
2	Lengvasis	JKL951	Toyota	Yaris	2005 01	2017 07 17	benzinas	7,40	161753
3	Krovininis	DES564	Renault	Megane	2009 10	2017 01 30	dyzelinas	8,10	750
4	Lengvasis	DCN631	Jeep	F10	2008 03	2017 05 30	dyzelinas	8,20	95614
5	Krovininis	ADE152	Mercedes	H16	2015 11	2017 08 08	dujos	6,90	610
6	Mikroautobusas	UPS456	SAAB	X	2007 06	2018 05 02	dyzelinas	7,30	18
7	Krovininis	PKM856	Mercedes	X3	2011 01	2017 10 04	benzinas	8,50	590
Klaipėda Jūros g. 12 klaipeda@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	CDE234	Audi	E92	2006 05	2016 01 11	dujos	8,10	201333
2	Krovininis	IKJ890	Peugeot	3008	2005 03	2017 10 13	dujos	7,90	650
3	Lengvasis	DEF345	Mercedes	Benz	2005 01	2017 04 06	benzinas	8,30	104506
4	Mikroautobusas	PSE456	Volkswagen	Jetta	2000 08	2018 04 16	benzinas	8,60	16
5	Mikroautobusas	KLT528	Lexus	10	2006 08	2017 01 31	dyzelinas	8,60	15
6	Krovininis	HIT789	Opel	Astra	1999 02	2017 09 22	dyzelinas	8,90	500
Jonava Sodų g. 12 jonava@greitis.lt Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	ABC012	Toyota	Avensis	2000 08	2018 07 16	dyzelinas	8,60	123456
2	Lengvasis	BCD123	Volkswagen	Golf	2001 07	2018 08 20	benzinas	8,20	90456
3	Krovininis	GHI678	Nissan	Qashqai	1998 01	2018 02 07	dujos	7,90	500
4	Mikroautobusas	UPS456	Skoda	X	2000 08	2017 03 09	dyzelinas	8,10	16
Duomenys apie geriausius automobilius: Lengvieji: Gamintojas: Volkswagen; Modelis: Golf; Valstybinis numeris: BCD123; Amžius: 17; Papildomas rodiklis: 90456; Krovininiai: Gamintojas: Renault; Modelis: Megane; Valstybinis numeris: DES564; Amžius: 9; Papildomas rodiklis: 750;									

Mikroautobusai:
Gamintojas: Subaru;
Modelis: X10;
Valstybinis numeris: MNO234;
Amžius: 14;
Papildomas rodiklis: 18;

Gamintojas: SAAB;
Modelis: X;
Valstybinis numeris: UPS456;
Amžius: 11;
Papildomas rodiklis: 18;

Duomenys apie filialus, kuriuose mikroautobusai seniausi:
Vilnius
Gerdino pr. 50
vilnius@greitis.lt

Klaipėda
Jūros g. 12
klaipeda@greitis.lt

Filialų krovinių automobilių sąrašai, surikiuoti pagal gamintoją ir modelį:
Vilnius
Gerdino pr. 50
vilnius@greitis.lt

Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	KLM012	Ford	A1	2008 05	2017 12 29	benzinas	6,80	560
2	Krovininis	JKL901	Seat	HQ	2010 04	2017 11 21	dyzelinas	7,10	490

Kaunas
Ramūnėlių g. 123
kaunas@greitis.lt

Automobiliai:

Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	ADE152	Mercedes	H16	2015 11	2017 08 08	dujos	6,90	610
2	Krovininis	PKM856	Mercedes	X3	2011 01	2017 10 04	benzinas	8,50	590
3	Krovininis	DES564	Renault	Megane	2009 10	2017 01 30	dyzelinas	8,10	750

Klaipėda
Jūros g. 12
klaipeda@greitis.lt

Automobiliai:

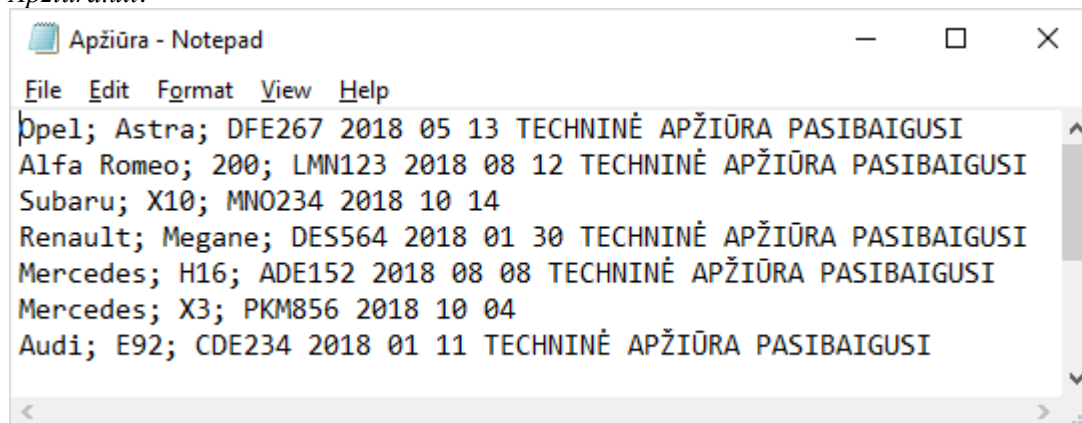
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	HIJ789	Opel	Astra	1999 02	2017 09 22	dyzelinas	8,90	500
2	Krovininis	IJK890	Peugeot	3008	2005 03	2017 10 13	dujos	7,90	650

Jonava
Sodų g. 12
jonava@greitis.lt

Automobiliai:

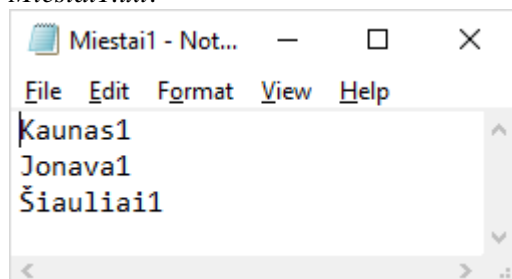
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	GHI678	Nissan	Qashqai	1998 01	2018 02 07	dujos	7,90	500

Apžiūra.txt:

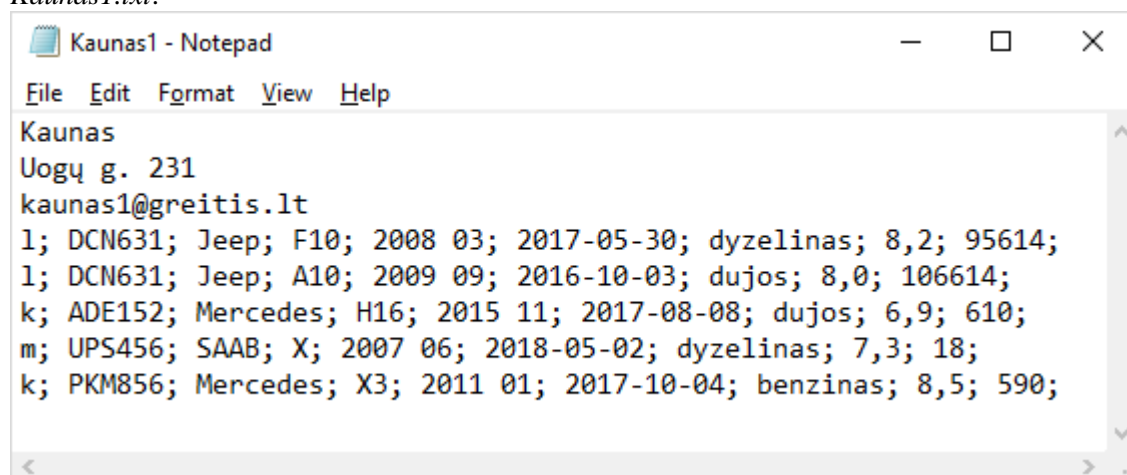


Pradiniai duomenys nr.2:

Miestai1.txt:

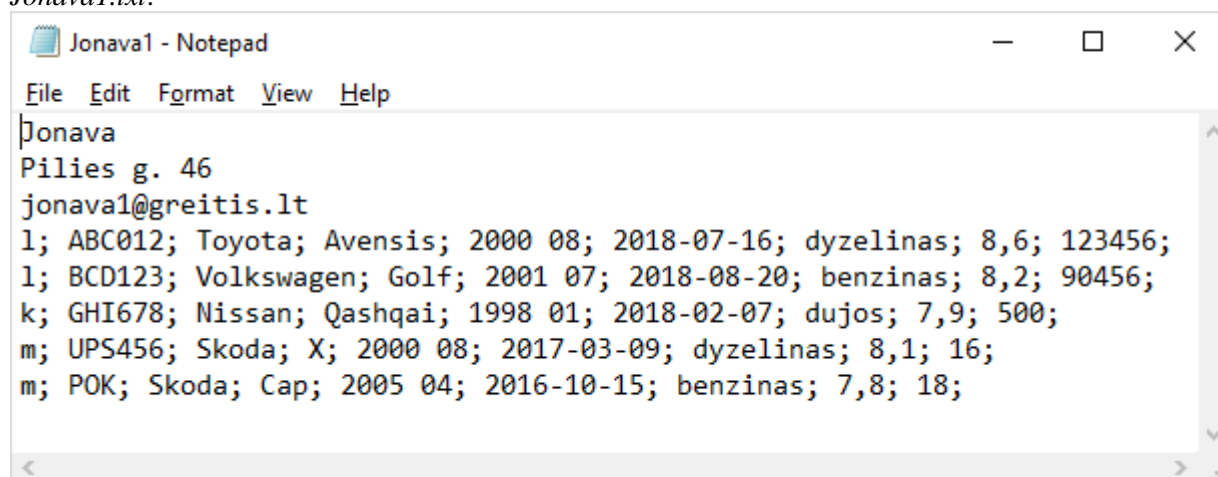


Kaunas1.txt:



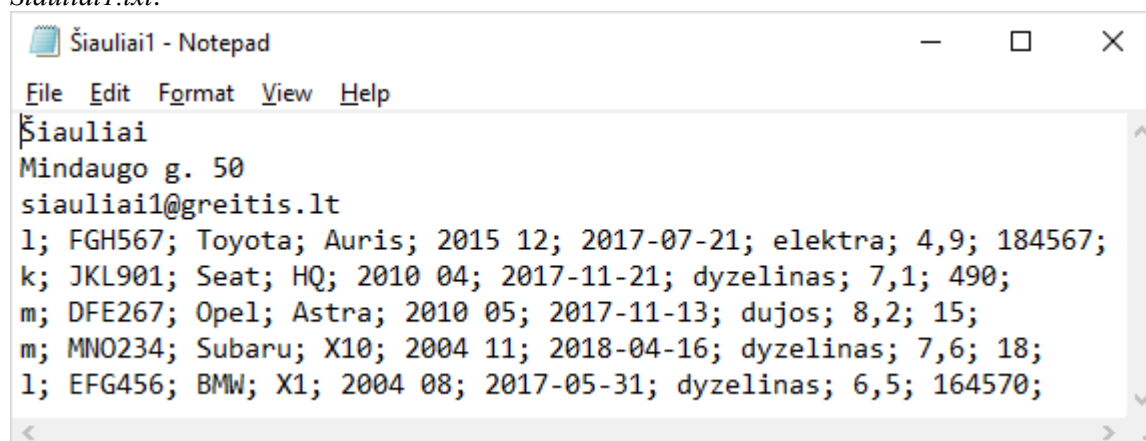
```
Kaunas1 - Notepad
File Edit Format View Help
Kaunas
Uogų g. 231
kaunas1@greitis.lt
l; DCN631; Jeep; F10; 2008 03; 2017-05-30; dyzelinas; 8,2; 95614;
l; DCN631; Jeep; A10; 2009 09; 2016-10-03; dujos; 8,0; 106614;
k; ADE152; Mercedes; H16; 2015 11; 2017-08-08; dujos; 6,9; 610;
m; UPS456; SAAB; X; 2007 06; 2018-05-02; dyzelinas; 7,3; 18;
k; PKM856; Mercedes; X3; 2011 01; 2017-10-04; benzinas; 8,5; 590;
```

Jonava1.txt:



```
Jonava1 - Notepad
File Edit Format View Help
Jonava
Pilies g. 46
jonava1@greitis.lt
l; ABC012; Toyota; Avensis; 2000 08; 2018-07-16; dyzelinas; 8,6; 123456;
l; BCD123; Volkswagen; Golf; 2001 07; 2018-08-20; benzinas; 8,2; 90456;
k; GHI678; Nissan; Qashqai; 1998 01; 2018-02-07; dujos; 7,9; 500;
m; UPS456; Skoda; X; 2000 08; 2017-03-09; dyzelinas; 8,1; 16;
m; POK; Skoda; Cap; 2005 04; 2016-10-15; benzinas; 7,8; 18;
```

Šiauliai1.txt:



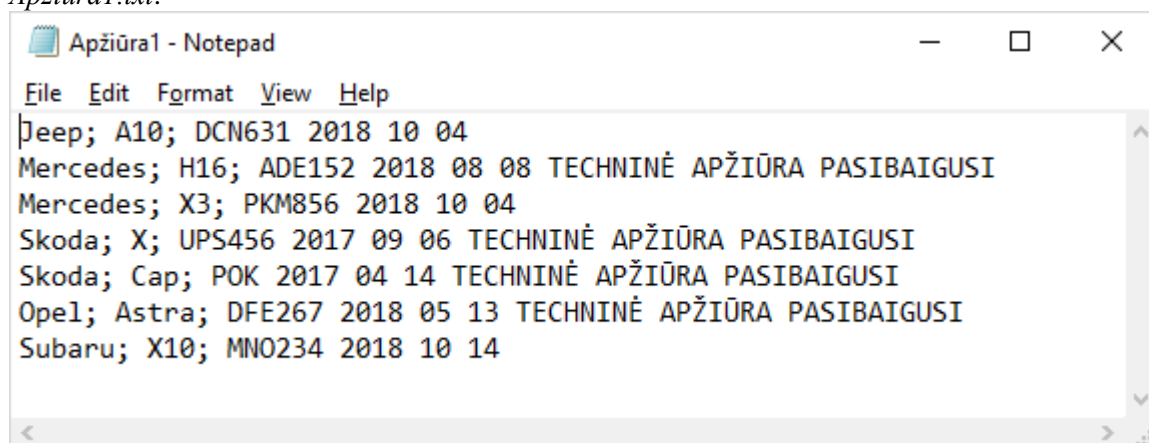
```
Šiauliai1 - Notepad
File Edit Format View Help
Šiauliai
Mindaugo g. 50
siauliai1@greitis.lt
l; FGH567; Toyota; Auris; 2015 12; 2017-07-21; elektra; 4,9; 184567;
k; JKL901; Seat; HQ; 2010 04; 2017-11-21; dyzelinas; 7,1; 490;
m; DFE267; Opel; Astra; 2010 05; 2017-11-13; dujos; 8,2; 15;
m; MNO234; Subaru; X10; 2004 11; 2018-04-16; dyzelinas; 7,6; 18;
l; EFG456; BMW; X1; 2004 08; 2017-05-31; dyzelinas; 6,5; 164570;
```

Rezultatai nr.2:

output1.txt:

Pradiniai duomenys:									
Kaunas									
Uogų g. 231									
kaunas1@greitis.lt									
Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	DCN631	Jeep	F10	2008 03	2017 05 30	dyzelinas	8,20	95614
2	Lengvasis	DCN631	Jeep	A10	2009 09	2016 10 03	dujos	8,00	106614
3	Krovininis	ADE152	Mercedes	H16	2015 11	2017 08 08	dujos	6,90	610
4	Mikroautobusas	UPS456	SAAB	X	2007 06	2018 05 02	dyzelinas	7,30	18
5	Krovininis	PKM856	Mercedes	X3	2011 01	2017 10 04	benzinas	8,50	590
Jonava									
Pilies g. 46									
jonaval@greitis.lt									
Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	ABC012	Toyota	Avensis	2000 08	2018 07 16	dyzelinas	8,60	123456
2	Lengvasis	BCD123	Volkswagen	Golf	2001 07	2018 08 20	benzinas	8,20	90456
3	Krovininis	GHI678	Nissan	Qashqai	1998 01	2018 02 07	dujos	7,90	500
4	Mikroautobusas	UPS456	Skoda	X	2000 08	2017 03 09	dyzelinas	8,10	16
5	Mikroautobusas	POK	Skoda	Cap	2005 04	2016 10 15	benzinas	7,80	18
Šiauliai									
Mindaugo g. 50									
siauliai1@greitis.lt									
Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Lengvasis	FGH567	Toyota	Auris	2015 12	2017 07 21	elektra	4,90	184567
2	Krovininis	JKL901	Seat	HQ	2010 04	2017 11 21	dyzelinas	7,10	490
3	Mikroautobusas	DFE267	Opel	Astra	2010 05	2017 11 13	dujos	8,20	15
4	Mikroautobusas	MNO234	Subaru	X10	2004 11	2018 04 16	dyzelinas	7,60	18
5	Lengvasis	EFG456	BMW	X1	2004 08	2017 05 31	dyzelinas	6,50	164570
Duomenys apie geriausius automobilius:									
Lengvieji:									
Gamintojas: Volkswagen;									
Modelis: Golf;									
Valstybinis numeris: BCD123;									
Amžius: 17;									
Papildomas rodiklis: 90456;									
Krovininiai:									
Gamintojas: Mercedes;									
Modelis: H16;									
Valstybinis numeris: ADE152;									
Amžius: 3;									
Papildomas rodiklis: 610;									
Mikroautobusai:									
Gamintojas: SAAB;									
Modelis: X;									
Valstybinis numeris: UPS456;									
Amžius: 11;									
Papildomas rodiklis: 18;									
Gamintojas: Skoda;									
Modelis: Cap;									
Valstybinis numeris: POK;									
Amžius: 13;									
Papildomas rodiklis: 18;									
Gamintojas: Subaru;									
Modelis: X10;									
Valstybinis numeris: MNO234;									
Amžius: 14;									
Papildomas rodiklis: 18;									
Duomenys apie filialus, kuriuose mikroautobusai seniausi:									
Jonava									
Pilies g. 46									
jonaval@greitis.lt									
Filialų krovininių automobilių sąrašai, surikiuoti pagal gamintoją ir modelį:									
Kaunas									
Uogų g. 231									
kaunas1@greitis.lt									
Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	ADE152	Mercedes	H16	2015 11	2017 08 08	dujos	6,90	610
2	Krovininis	PKM856	Mercedes	X3	2011 01	2017 10 04	benzinas	8,50	590
Jonava									
Pilies g. 46									
jonaval@greitis.lt									
Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	GHI678	Nissan	Qashqai	1998 01	2018 02 07	dujos	7,90	500
Šiauliai									
Mindaugo g. 50									
siauliai1@greitis.lt									
Automobiliai:									
Nr.	Tipas	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninės apžiūros data	Kuras	Vidutinės sąnaudos/100km	Papildomas rodiklis
1	Krovininis	JKL901	Seat	HQ	2010 04	2017 11 21	dyzelinas	7,10	490

Apžiūra1.txt:



Apžiūra1 - Notepad

File Edit Format View Help

Jeep; A10; DCN631 2018 10 04
Mercedes; H16; ADE152 2018 08 08 TECHNINĖ APŽIŪRA PASIBAIGUSI
Mercedes; X3; PKM856 2018 10 04
Skoda; X; UPS456 2017 09 06 TECHNINĖ APŽIŪRA PASIBAIGUSI
Skoda; Cap; POK 2017 04 14 TECHNINĖ APŽIŪRA PASIBAIGUSI
Opel; Astra; DFE267 2018 05 13 TECHNINĖ APŽIŪRA PASIBAIGUSI
Subaru; X10; MNO234 2018 10 14

5. Susietasis sąrašas (L4)

5.1. Darbo užduotys

U1-2. Krepšinis.

Krepšinio mokykloje treniruotes lankančių sąrašas yra tekstiniame faile: *būsimo krepšininko vardas ir pavardė, amžius ir ūgis*. Pirmoje eilutėje yra krepšinio mokyklos pavadinimas. Turime dviejų mokyklų duomenis.

L1+L2+L4.

- Raskite, koks būsimų krepšininkų amžiaus vidurkis ir koks ūgio vidurkis kiekvienoje mokykloje.
- Surašykite į atskirą rinkinį visus abiejų mokyklų sportininkus, kurių ūgis didesnis už vidurkį.
- Surikiuokite rezultatų sąrašą amžiaus didėjimo tvarka.
- Pašalinkite iš rezultatų sąrašo krepšininkus, kurių amžius yra didesnis už nurodytą klaviatūra.

L2 papildymas.

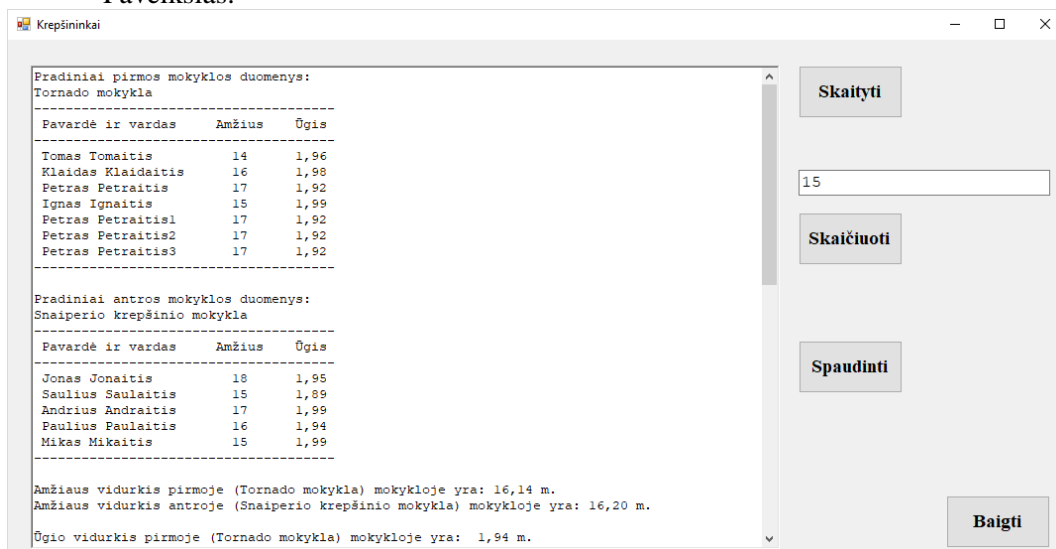
- Papildykite surikiuotą rezultatų sąrašą naujais krepšininkais, kurių ūgis didesnis, už sudaryto sąrašo krepšininkų ūgio vidurkį. Duomenys yra faile *Naujokai.txt*. Pirmoje eilutėje – vadybininko vardas ir pavardė.

5.2. Grafinės vartotojo sąsajos schema ir paveikslas

Schema:



Paveikslas:



Krepšininkai

Ūgio vidurkis pirmoje (Tornado mokykla) mokykloje yra: 1,94 m.
Ūgio vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 1,95 m.

Konteineris krepšininkų, kurių ūgis didesnis už vidurkį: 1,95
Abi mokyklos

Pavardė ir vardas	Amžius	Ūgis
Mikas Mikaitis	15	1,99
Andrius Andraitis	17	1,99
Ignas Ignaitis	15	1,99
Klaidas Klaidaitis	16	1,98
Tomas Tomaitis	14	1,96

Surikiuotas konteineris:
Abi mokyklos

Pavardė ir vardas	Amžius	Ūgis
Andrius Andraitis	17	1,99
Klaidas Klaidaitis	16	1,98
Ignas Ignaitis	15	1,99
Mikas Mikaitis	15	1,99
Tomas Tomaitis	14	1,96

Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

Skaityti

15

Skaičiuoti

Spaudinti

Baigti

Krepšininkai

Mikas Mikaitis	15	1,99
Andrius Andraitis	17	1,99
Ignas Ignaitis	15	1,99
Klaidas Klaidaitis	16	1,98
Tomas Tomaitis	14	1,96

Surikiuotas konteineris:
Abi mokyklos

Pavardė ir vardas	Amžius	Ūgis
Andrius Andraitis	17	1,99
Klaidas Klaidaitis	16	1,98
Ignas Ignaitis	15	1,99
Mikas Mikaitis	15	1,99
Tomas Tomaitis	14	1,96

Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

Ignas Ignaitis	15	1,99
Mikas Mikaitis	15	1,99
Tomas Tomaitis	14	1,96

Skaityti

15

Skaičiuoti

Spaudinti

Baigti

5.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
Form1	Text: Krepšininkai	Programos langas
richTextBox1 (tipas RichTextBox)	Font: Courier New, Regular, 10; Script: Baltic;	Atvaizduoti pradinių duomenų ir skaičiavimų turiniui
Button1 (tipas Button)	Name: read Text: Skaityti Font: Times New Roman, Bold, 14; Script: Baltic;	Mygtukas duomenims iš failo nuskaityti
Button2 (tipas Button)	Name: skaiciuoti Text: Skaičiuoti Font: Times New Roman, Bold, 14; Script: Baltic;	Mygtukas skaičiavimams atlikti
Button3 (tipas Button)	Name: print Text: Spausdinti Font: Times New Roman, Bold, 14;	Mygtukas skaičiavimų rezultatams spausdinti

	Script: Baltic;	
Button4 (tipas Button)	Name: baigti Text: Baigti Font: Times New Roman, Bold, 14; Script: Baltic;	Mygtukas programos lango uždarymui
TextBox1 (tipas TextBox)	Text: Įveskite norimą amžių Font: Courier New, Regular, 12; Script: Baltic;	Teksto laukas norimam amžiui užrašyti

5.4. Programos vartotojo vadovas

1. Paleiskite programą.
2. Spauskite mygtuką „Skaityti“. Ekrane matysite nuskaitytus duomenis.
3. Į teksto lauką įveskite amžių, už kurį vyresnius krepšininkus norite pašalinti.
4. Spauskite „Skaičiuoti“.
5. Spauskite „Spausdinti“.
6. Ekrane galite peržiūrėti skaičiavimų rezultatus.
7. Norėdami baigti darbą su programa, spauskite „Baigti“.

5.5. Programos tekstas

Klasė **Node**:

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

namespace L1
{
    sealed class Node
    {
        public Krepšininkas Duomenys { get; set; }
        public Node Next { get; set; }

        public Node (Krepšininkas duom, Node next)
        {
            this.Duomenys = duom;
            this.Next = next;
        }

        public override string ToString()
        {
            return Duomenys.ToString();
        }
    }
}
```

Klasė **krepšininkas**:

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

namespace L1
{
    class Krepšininkas
    {
        public string VP { get; set; }
        public int Amžius { get; set; }
    }
}
```

```

    public double Ūgis { get; set; }

    public Krepsininkas(string vrdpav, int amz, double ug)
    {
        VP = vrdpav;
        Amžius = amz;
        Ūgis = ug;
    }

    public override string ToString()
    {
        string eilute;
        eilute = string.Format(" {0, -20}    {1, 2}    {2, 5:f}", VP, Amžius, Ūgis);
        return eilute;
    }

    public static bool operator <=(Krepsininkas k1, Krepsininkas k2)
    {
        int poz = string.Compare(k1.VP, k2.VP, StringComparison.CurrentCulture);

        return k1.Amžius > k2.Amžius || (k1.Amžius == k2.Amžius && poz < 0);
    }

    public static bool operator >=(Krepsininkas k1, Krepsininkas k2)
    {
        int poz = string.Compare(k1.VP, k2.VP, StringComparison.CurrentCulture);

        return k1.Amžius < k2.Amžius || (k1.Amžius == k2.Amžius && poz > 0);
    }
}

```

Klasė **krepšininkai**:

```

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

namespace L1
{
    /// <summary>
    ///
    /// </summary>
    class Krepsininkai
    {
        private Node first;
        private Node last;
        private Node link;
        private Node insertionHelper;

        public Krepsininkai()
        {
            this.last = new Node(null, null);
            this.first = new Node(null, last);
            this.insertionHelper = first;
            this.link = null;
        }

        public void Start()
        {
            link = first.Next;
        }

        public bool isEmpty()
        {
            return link.Next == null;
        }
    }
}

```

```

}

public void Next()
{
    link = link.Next;
}

public Krepsininkas ImtiKrepsininka()
{
    return link.Duomenys;
}

public void AddFirst(Krepsininkas duom)
{
    first.Next = new Node(duom, first.Next);
}

public void AddLast(Krepsininkas duom)
{
    insertionHelper.Next = new Node(duom, last);
    insertionHelper = insertionHelper.Next;
}

public void Clear()
{
    while (first.Next != null)
    {
        link = first.Next;
        first.Next.Duomenys = null;
        first = first.Next.Next;
        link = null;
    }

    first = last;
}

public void Sort()
{
    bool switched = true;
    Node temp1, temp2;

    while(switched)
    {
        switched = false;
        temp1 = temp2 = first.Next;
        while(temp2.Next != null)
        {
            if (temp2.Duomenys <= temp1.Duomenys)
            {
                switched = true;
                Krepsininkas temp = temp1.Duomenys;
                temp1.Duomenys = temp2.Duomenys;
                temp2.Duomenys = temp;
            }

            temp1 = temp2;
            temp2 = temp2.Next;
        }
    }
}

public void Filter(int amzius)
{
    Node iterator = first;
    while (iterator.Next != last)
    {
        if (iterator.Next.Duomenys.Amžius > amzius)
        {

```

Programos tekstas:

```
namespace L1
```

69

```

        skaiciuoti.Enabled = true;
    }

    /// <summary>
    /// Atliekami reikiami skaičiavimai, jų rezultatai įvedami į duomenų failą.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void skaiciuoti_Click(object sender, EventArgs e)
    {
        double amzVid1 = AmziausVidurkis(mokykla1);
        SpausdintiTeksta(rez, string.Format("Amžiaus vidurkis pirmoje ({0}) mokykloje yra:
{1, 5:f} m.", mokPav1, amzVid1));
        double amzVid2 = AmziausVidurkis(mokykla2);
        SpausdintiTeksta(rez, string.Format("Amžiaus vidurkis antroje ({0}) mokykloje yra:
{1, 5:f} m.\r\n", mokPav2, amzVid2));

        double ugiovid1 = ŪgioVidurkis(mokykla1);
        SpausdintiTeksta(rez, string.Format("Ūgio vidurkis pirmoje ({0}) mokykloje yra:
{1, 5:f} m.", mokPav1, ugiovid1));
        double ugiovid2 = ŪgioVidurkis(mokykla2);
        SpausdintiTeksta(rez, string.Format("Ūgio vidurkis antroje ({0}) mokykloje yra:
{1, 5:f} m.\r\n", mokPav2, ugiovid2));

        double vidur = Math.Round(((ugiovid1 + ugiovid2) / 2), 2);
        UgisDidesnisUzVidurki(mokykla1, naujasKonteineris, vidur);
        UgisDidesnisUzVidurki(mokykla2, naujasKonteineris, vidur);
        Spausdinti(rez, naujasKonteineris, "Abi mokyklos", $"Konteineris krepšininkų,
kurių ūgis didesnis už vidurį: {vidur}");

        naujasKonteineris.Sort();
        Spausdinti(rez, naujasKonteineris, "Abi mokyklos", "Surikiuotas konteineris:");

        string text = File.ReadAllText(rez);
        richTextBox1.Text = text;

        int amz = int.Parse(textBox1.Text);
        naujasKonteineris.Filter(amz);
        Spausdinti(rez, naujasKonteineris, "Abi mokyklos", "Pašalinti moksleiviai, kurių
amžius didesnis už nurodytą:");

        print.Enabled = true;
    }

    /// <summary>
    /// Spaudina tekstą iš duomenų failo į programos langą.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void print_Click(object sender, EventArgs e)
    {
        string x = File.ReadAllText(rez);
        richTextBox1.Text = x;
    }

    /// <summary>
    /// Užėjus ant teksto lauko, jį išvalo.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void textBox1_MouseClick(object sender, MouseEventArgs e)
    {
        textBox1.Text = "";
    }

    /// <summary>
    /// Baigia programos veikimą.

```

```

/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void baigti_Click(object sender, EventArgs e)
{
    Close();
}

/// <summary>
/// Formuoja naują konteinerį iš krepšinių, kurių ūgis didesnis už vidurkį.
/// </summary>
/// <param name="kont1">Pirmas konteineris</param>
/// <param name="kont2">Antras kont</param>
/// <param name="naujas">Trečias kont</param>
private void UgisDidesnisUzVidurki(Krepsininkai kont, Krepsininkai naujas, double
ugis)
{
    for (kont.Start(); !kont.isEmpty(); kont.Next())
    {
        if (kont.ImtiKrepsininka().Ūgis > ugis)
        {
            naujas.AddFirst(kont.ImtiKrepsininka());
        }
    }
}

/// <summary>
/// Randa krepšinių amžiaus vidurkį.
/// </summary>
/// <param name="kont">Konteineris</param>
/// <returns>Amžiaus vidurkis</returns>
static double AmziausVidurkis(Krepsininkai kont)
{
    double suma = 0;
    int kiekis = 0;

    for (kont.Start(); !kont.isEmpty(); kont.Next())
    {
        suma += kont.ImtiKrepsininka().Amžius;
        kiekis++;
    }

    if (kiekis != 0)
        return suma / kiekis;
    else return 0.0;
}

/// <summary>
/// Randa krepšinių ūgio vidurkį.
/// </summary>
/// <param name="kont">Konteineris</param>
/// <returns>Ūgio vidurkis</returns>
static double ŪgioVidurkis(Krepsininkai kont)
{
    double suma = 0;
    int kiekis = 0;

    for (kont.Start(); !kont.isEmpty(); kont.Next())
    {
        suma += kont.ImtiKrepsininka().Ūgis;
        kiekis++;
    }

    if (kiekis != 0)
        return suma / kiekis;
    else return 0.0;
}

```

```

/// <summary>
/// Nuskaityti duomenų failus.
/// </summary>
/// <param name="fr"></param>
/// <param name="pav"></param>
/// <returns></returns>
private Krepsininkai Skaityti(string fr, out string pav)
{
    Krepsininkai konteineris = new Krepsininkai();
    using(StreamReader reader = new StreamReader(fr, Encoding.GetEncoding(1257)))
    {
        string line;
        line = reader.ReadLine();
        pav = line;

        while((line = reader.ReadLine()) != null)
        {
            string[] parts = line.Split(';');
            string pavVrd = parts[0];
            int amz = int.Parse(parts[1]);
            double ug = double.Parse(parts[2]);
            Krepsininkas krep = new Krepsininkas(pavVrd, amz, ug);
            konteineris.AddLast(krep);
        }

        return konteineris;
    }
}

/// <summary>
/// Spausdina duomenis į failą lentele.
/// </summary>
/// <param name="fw"></param>
/// <param name="kont"></param>
/// <param name="pav"></param>
/// <param name="antraste"></param>
private void Spausdinti(string fw, Krepsininkai kont, string pav, string antraste)
{
    const string virsus =
        "-----\r\n" +
        " Pavardė ir vardas      Amžius      Ūgis \r\n" +
        "-----";

    using (var fr = new StreamWriter(File.Open(fw, FileMode.Append)))
    {
        kont.Start();
        if (!kont.isEmpty())
        {
            fr.WriteLine(antraste);
            fr.WriteLine(pav);
            fr.WriteLine(virsus);
            for (kont.Start(); !kont.isEmpty(); kont.Next())
            {
                Krepsininkas krep = kont.ImtiKrepsininka();
                fr.WriteLine("{0}", krep);
            }
            fr.WriteLine("-----\r\n");
        }
        else
        {
            fr.WriteLine("Krepšininkų konteineris tuščias!\r\n");
        }
    }
}

/// <summary>
/// Spausdina tekstą į failą.
/// </summary>

```



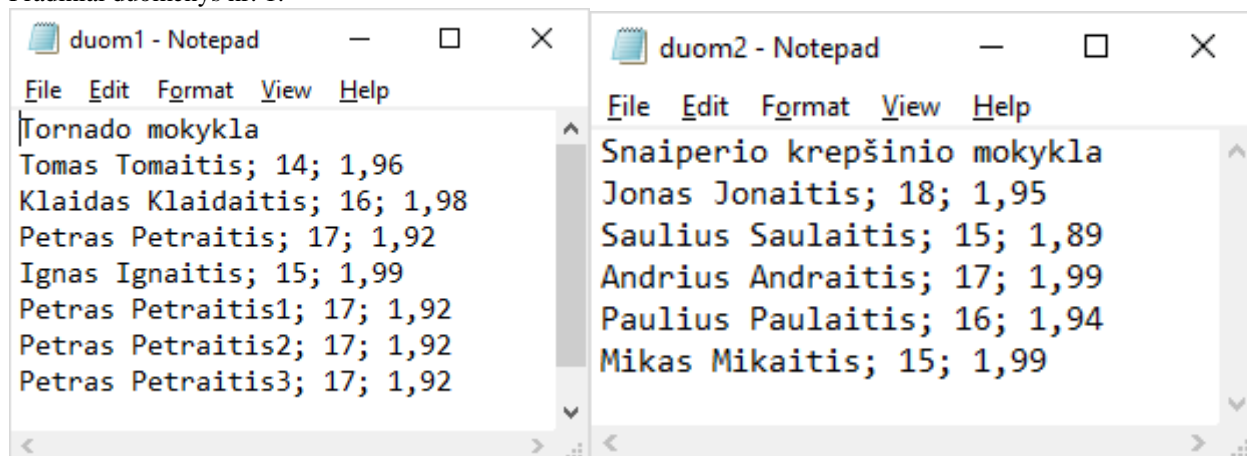
```

/// <param name="rez"></param>
/// <param name="x"></param>
private void SpausdintiTeksta(string rez, string x)
{
    using (StreamWriter fw = new StreamWriter(File.Open(rez, FileMode.Append)))
    {
        fw.WriteLine(x);
    }
}
}
}

```

5.6. Pradiniai duomenys ir rezultatai

Pradiniai duomenys nr. 1:



Rezultatai nr. 1:

rez - Notepad		
File	Edit	Format View Help
Pradiniai pirmos mokyklos duomenys:		
Tornado mokykla		
Pavardė ir vardas	Amžius	Ūgis
Tomas Tomaitis	14	1,96
Klaidas Klaidaitis	16	1,98
Petras Petraitis	17	1,92
Ignas Ignaitis	15	1,99
Petras Petraitis1	17	1,92
Petras Petraitis2	17	1,92
Petras Petraitis3	17	1,92
Pradiniai antros mokyklos duomenys:		
Snaiperio krepšinio mokykla		
Pavardė ir vardas	Amžius	Ūgis
Jonas Jonaitis	18	1,95
Saulius Saulaitis	15	1,89
Andrius Andraitis	17	1,99
Paulius Paulaitis	16	1,94
Mikas Mikaitis	15	1,99

```
rez - Notepad
File Edit Format View Help
Amžiaus vidurkis pirmoje (Tornado mokykla) mokykloje yra: 16,14 m.
Amžiaus vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 16,20 m.

Ūgio vidurkis pirmoje (Tornado mokykla) mokykloje yra: 1,94 m.
Ūgio vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 1,95 m.

Konteineris krepšininkų, kurių ūgis didesnis už vidurkį: 1,95
Abi mokyklos

-----
Pavardė ir vardas    Amžius    Ūgis
-----
Mikas Mikaitis       15        1,99
Andrius Andraitis    17        1,99
Ignas Ignaitis       15        1,99
Klaidas Klaidaitis   16        1,98
Tomas Tomaitis       14        1,96
-----

Surikiuotas konteineris:
Abi mokyklos

-----
Pavardė ir vardas    Amžius    Ūgis
-----
Andrius Andraitis    17        1,99
Klaidas Klaidaitis   16        1,98
Ignas Ignaitis       15        1,99

-----
rez - Notepad
File Edit Format View Help
Tomas Tomaitis       14        1,96
-----

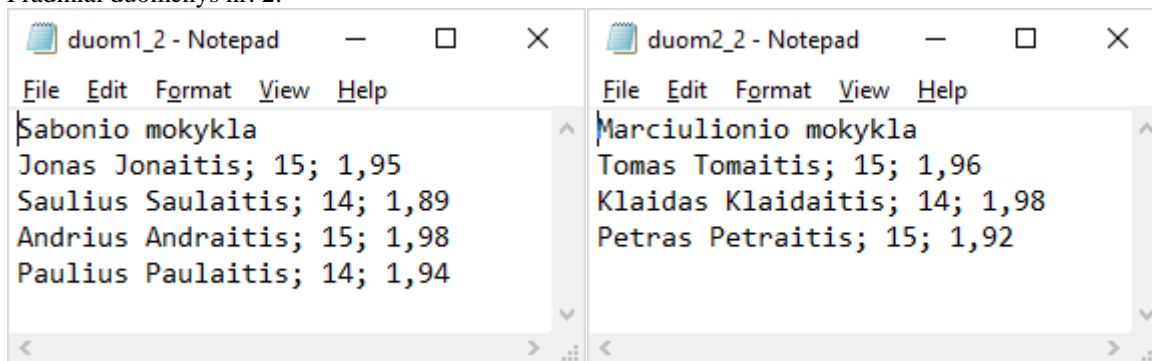
Surikiuotas konteineris:
Abi mokyklos

-----
Pavardė ir vardas    Amžius    Ūgis
-----
Andrius Andraitis    17        1,99
Klaidas Klaidaitis   16        1,98
Ignas Ignaitis       15        1,99
Mikas Mikaitis       15        1,99
Tomas Tomaitis       14        1,96
-----

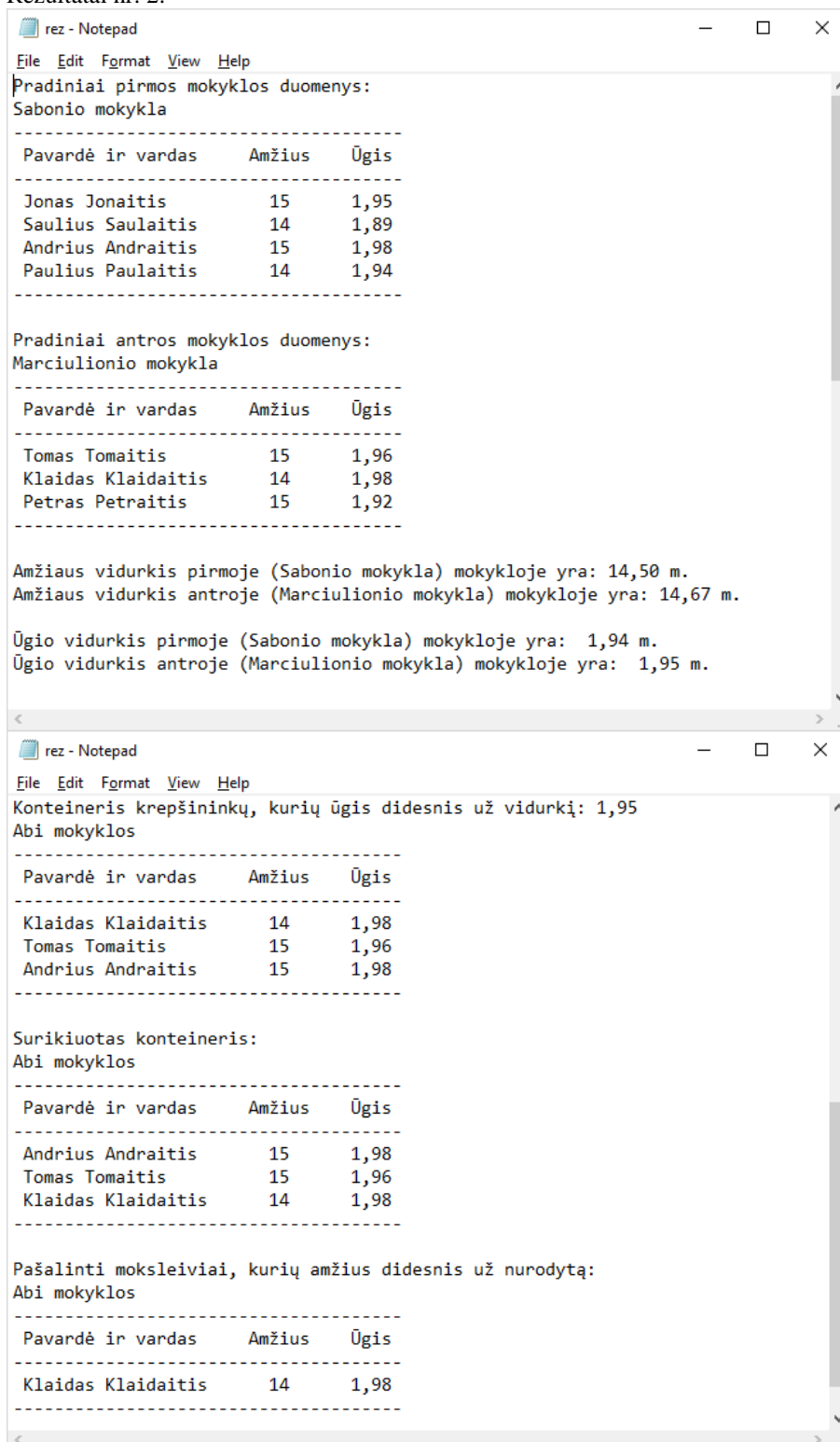
Pašalinti moksleiviai, kurių amžius didesnis už nurodytą:
Abi mokyklos

-----
Pavardė ir vardas    Amžius    Ūgis
-----
Ignas Ignaitis       15        1,99
Mikas Mikaitis       15        1,99
Tomas Tomaitis       14        1,96
-----
```

Pradiniai duomenys nr. 2:



Rezultatai nr. 2:



6. Bendrinės klasės (L5)

6.1. Darbo užduotis

6.2. Grafinės vartotojo sąsajos schema ir paveikslas

6.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė

6.4. Programos vartotojo vadovas

6.5. Programos tekstas

6.6. Pradiniai duomenys ir rezultatai