

KAUNO TECHNOLOGIJOS UNIVERSITETO GIMNAZIJA

Objektinio programavimo pagrindai II (P175B502)

Laboratorinių darbų ataskaitos

Atliko:

III NFQ klasės mokinys

Augustas Mačijauskas

2018 m. gegužės 31 d.

Priėmė:

Kęstutis Simonavičius, Jūratė Pauliūtė

KAUNAS 2018

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	30
3.2.	Grafinės vartotojo sąsajos schema ir paveikslas	30
3.3.	Sąsajoje panaudotų komponentų keičiamos savybės	33
3.4.	Programos vartotojo vadovas	34
3.5.	Programos tekstas	34
3.6.	Pradiniai duomenys ir rezultatai	40
4.	Susietasis sąrašas (L4)	44
4.1.	Darbo užduotis	44
4.2.	Grafinės vartotojo sąsajos schema ir paveikslas	44
4.3.	Sąsajoje panaudotų komponentų keičiamos savybės	44
4.4.	Programos vartotojo vadovas	44
4.5.	Programos tekstas	44
4.6.	Pradiniai duomenys ir rezultatai	44
5.	Bendrinės klasės (L5)	45
5.1.	Darbo užduotis	45
5.2.	Grafinės vartotojo sąsajos schema ir paveikslas	45
5.3.	Sąsajoje panaudotų komponentų keičiamos savybės	45
5.4.	Programos vartotojo vadovas	45
5.5.	Programos tekstas	45
5.6.	Pradiniai duomenys ir rezultatai	45

1. Grafinė vartotojo sąsaja ir algoritmų taikymas (L1)

1.1. 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.

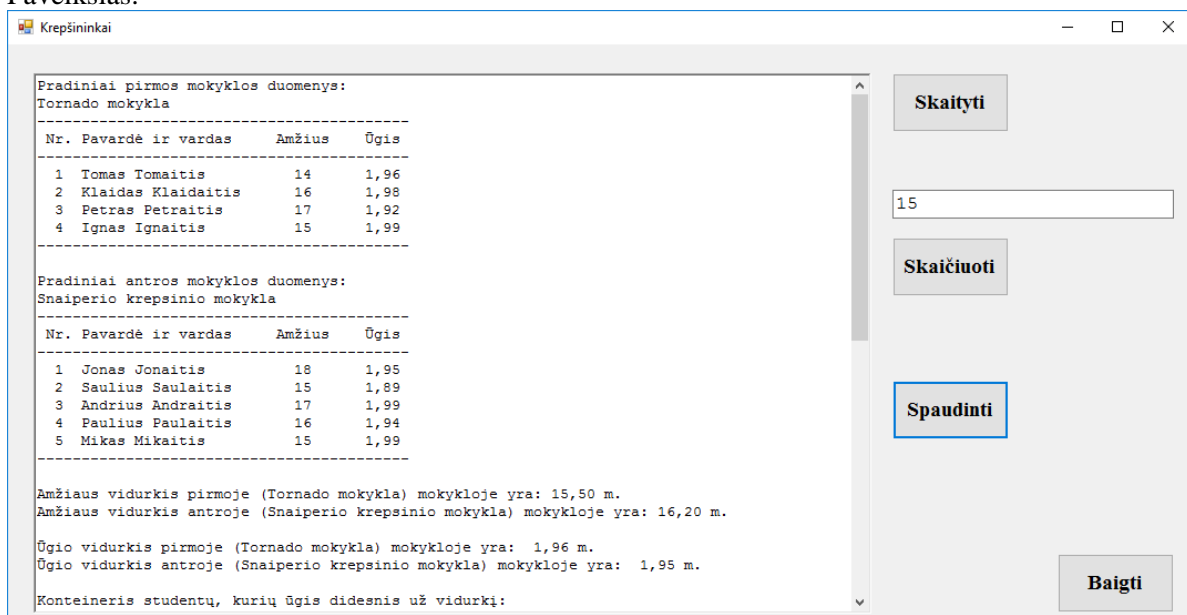
- Papildykite surikiuotą rezultatų sąrašą naujais krepšininkais, kurių ūgis didesnis, už sudaryto sąrašo krepšininkų ūgio vidurkį. Duomenys yra faile *Naujokai.txt*. Pirmoje eilutėje – vadybininko vardas ir pavardė.

1.2. Grafinės vartotojo sąsajos schema ir paveikslas

Schema:



Paveikslas:



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" +
        "-----";

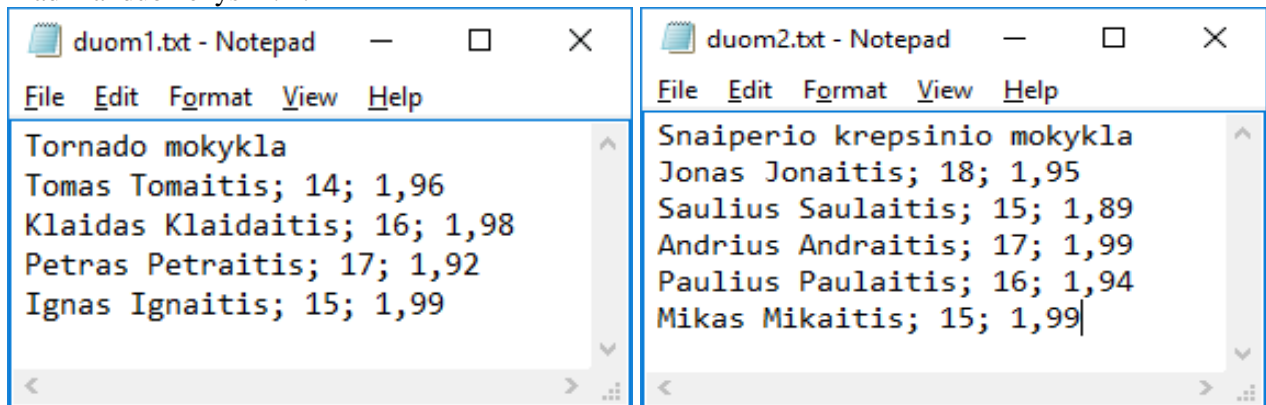
    using (var fr = new StreamWriter(File.Open(fw, FileMode.Append)))
    {
        if (kont.Kiek > 0)
        {
            fr.WriteLine(antraste);
            fr.WriteLine(pav);
            fr.WriteLine(virsus);
            for (int i = 0; i < kont.Kiek; i++)
            {
                Krepsininkas 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);
    }
}
}
}

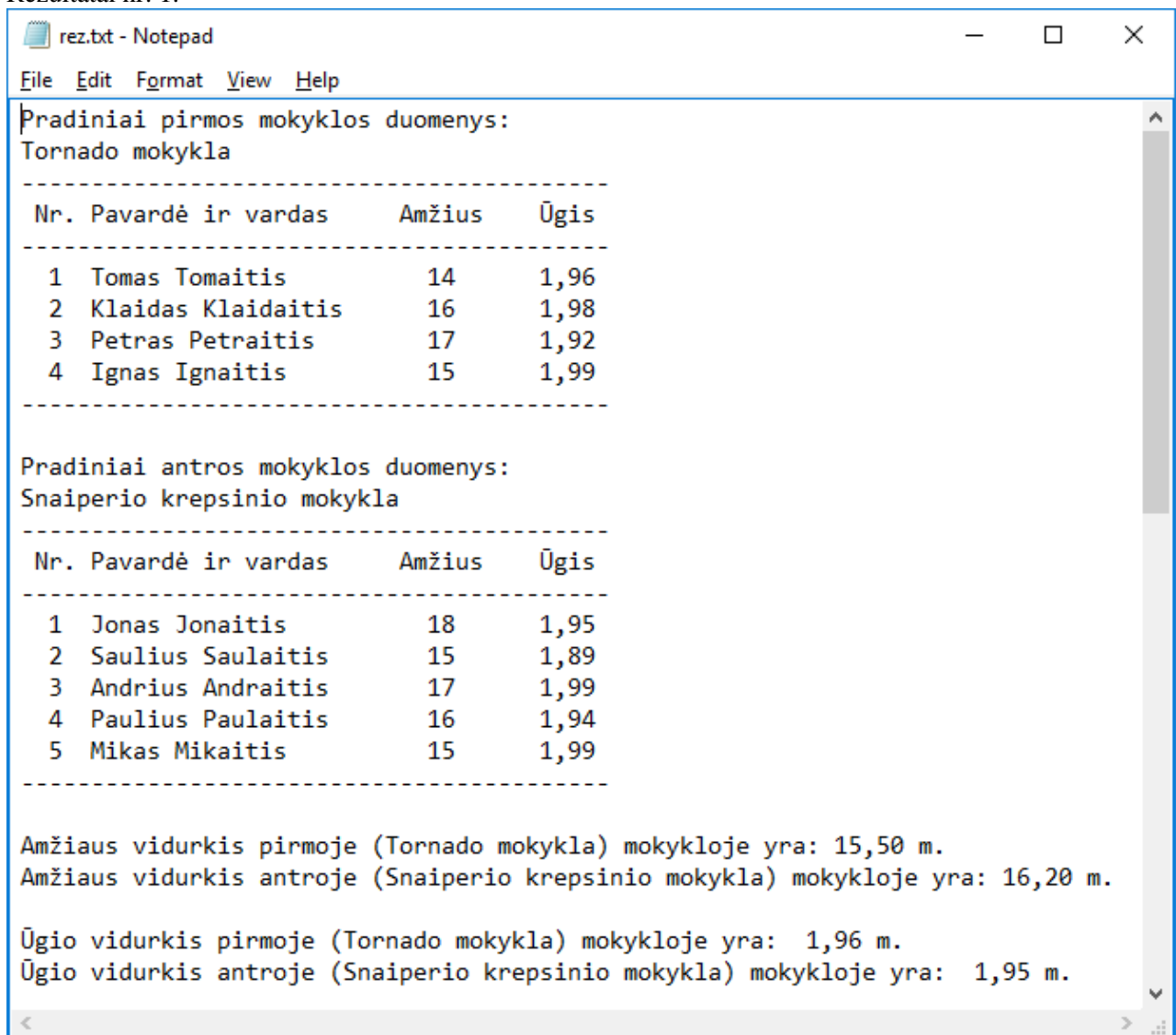
```

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  Klaudas 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  Klaudas 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  Klaudas 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	Klaudas 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.

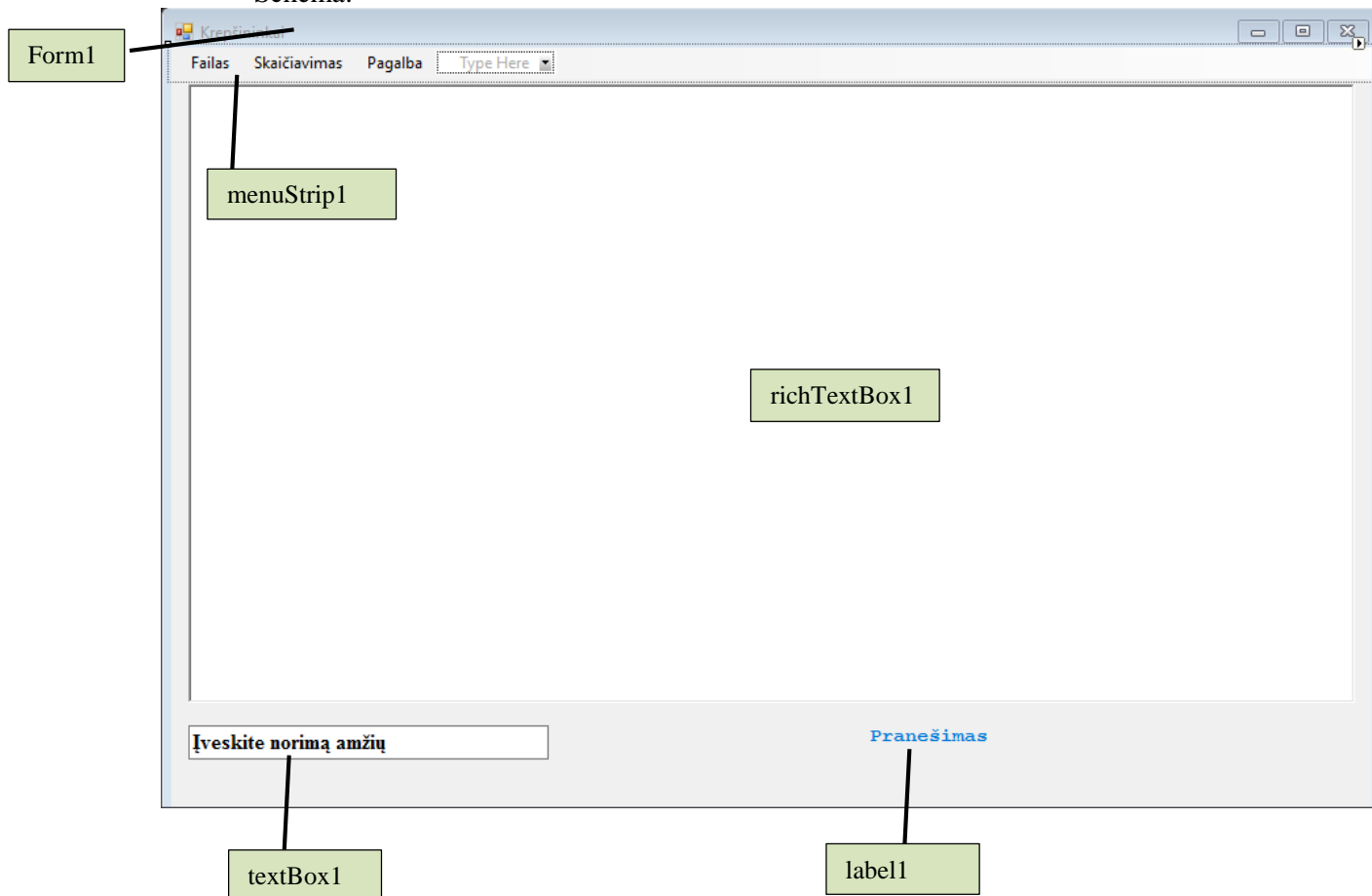
- Raskite, koks būsimų krepšininkų amžiaus vidurkis ir koks ūgio vidurkis kiekvienoje mokykloje.
- Surašykite į atskirą rinkinį visus abiejų mokyklų sportininkus, kurių ūgis didesnis už vidurkį.
- Surikiuokite rezultatų sąrašą amžiaus didėjimo tvarka.
- Pašalinkite iš rezultatų sąrašo krepšininkus, kurių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

Ūgio vidurkis antroje (Snaiperio krepšinio mokykla) mokykloje yra: 1,95 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 ToolStripMenu)	Text: Failas Font: Standartiniai nustatymai	Meniu punktas, kuriame yra mygtukai duomenų nuskaitymui ir programos darbo baigimui
ivestiToolStripMenuItem (tipas ToolStripMenu)	Name: nuskaityti Text: Įvesti Font: Standartiniai nustatymai	Meniu punktas duomenims iš failo nuskaityti

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

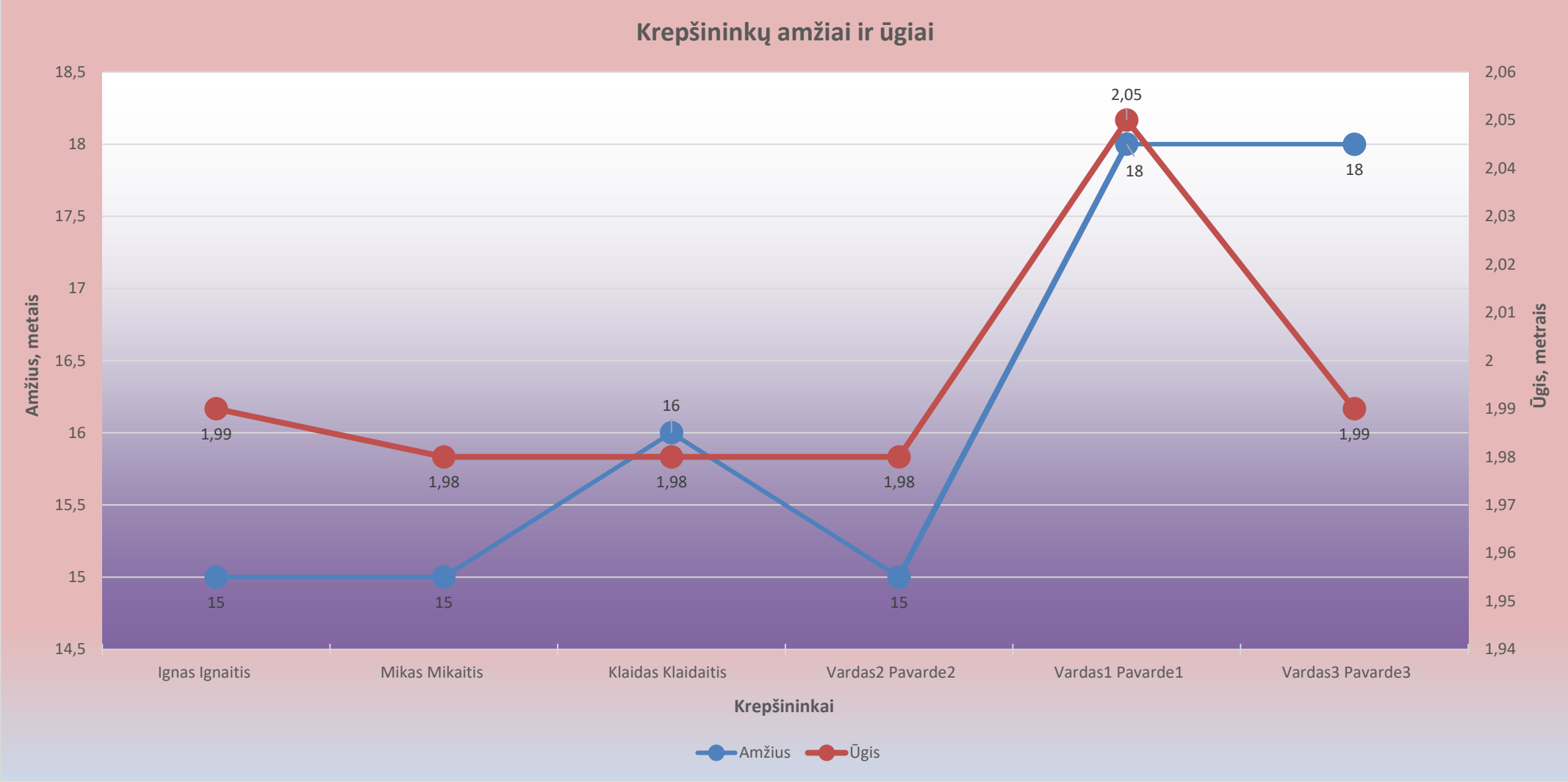
duom1_2.txt - Note...	duom2_2.txt - Note...
Sabonio mokykla Jonas Jonaitis; 15; 1,95 Saulius Saulaitis; 14; 1,89 Andrius Andraitis; 15; 1,98 Paulius Paulaitis; 14; 1,94	Marciulionio mokykla Tomas Tomaitis; 15; 1,96 Klaidas Klaidaitis; 14; 1,98 Petras Petraitis; 15; 1,92

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

```



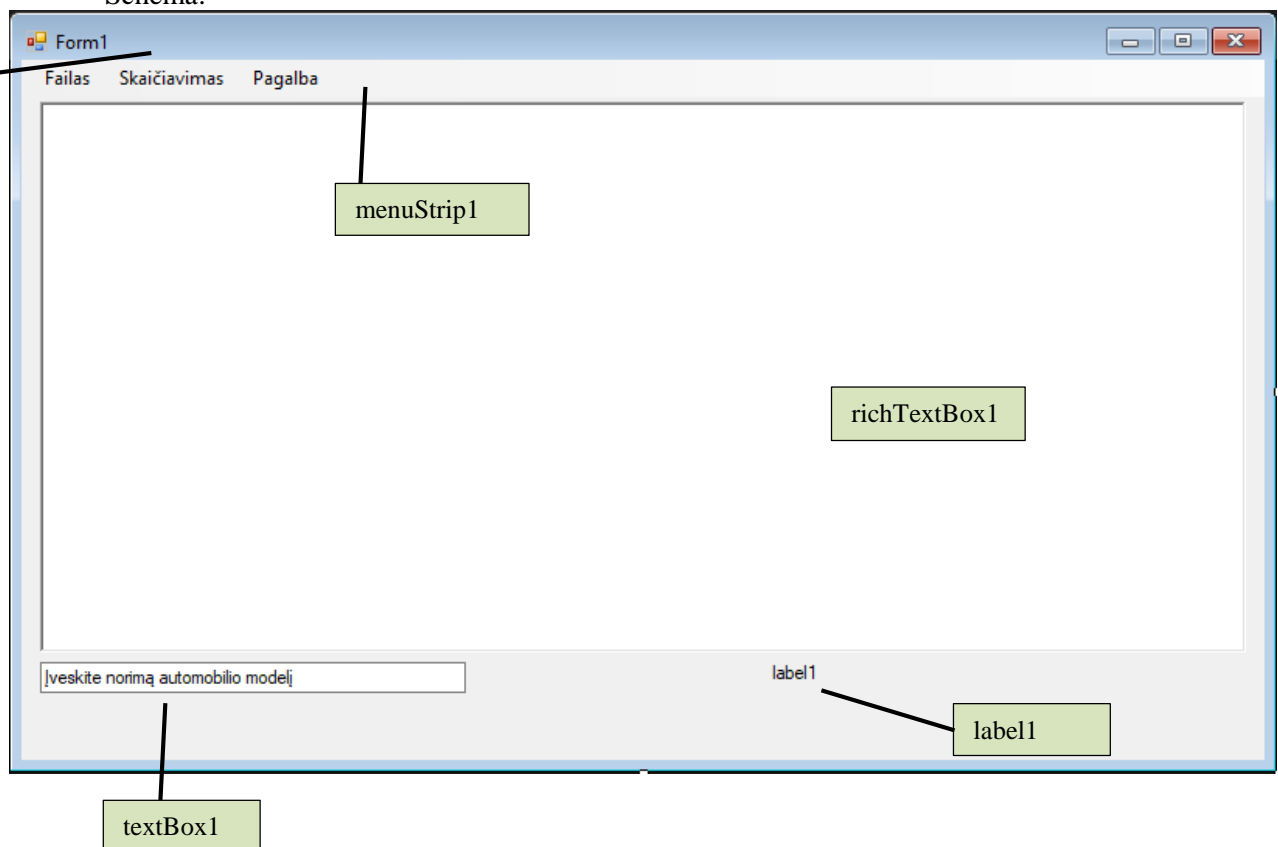
3. Paveldėjimas (L3)

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 sąajos schema ir paveikslas

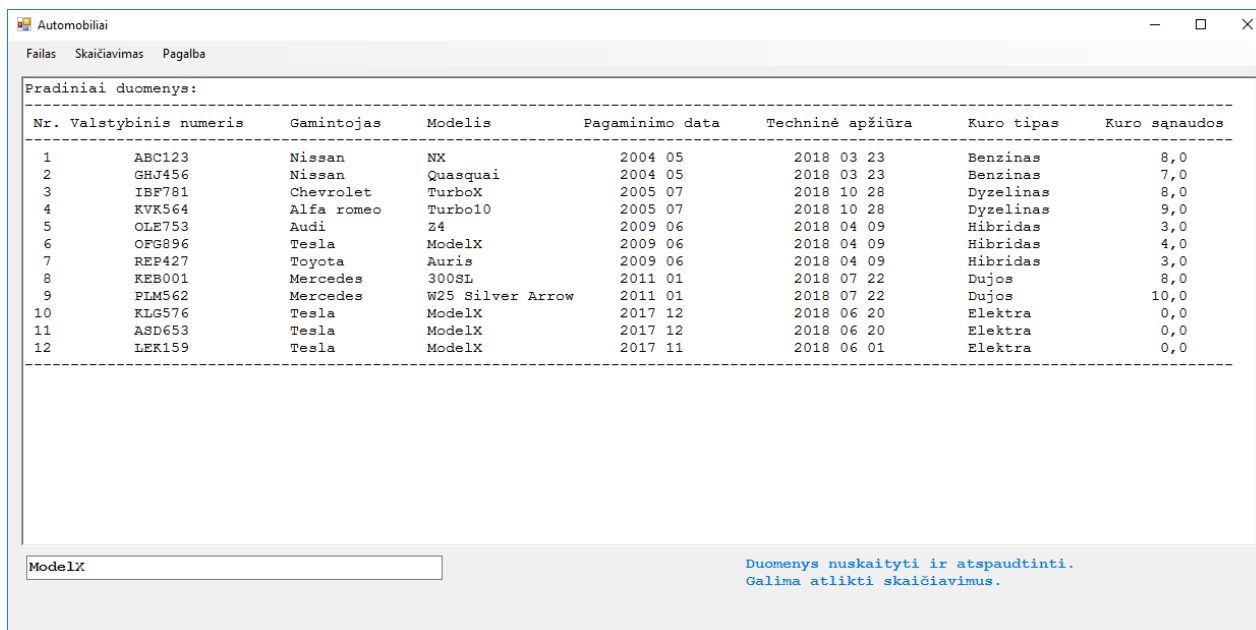
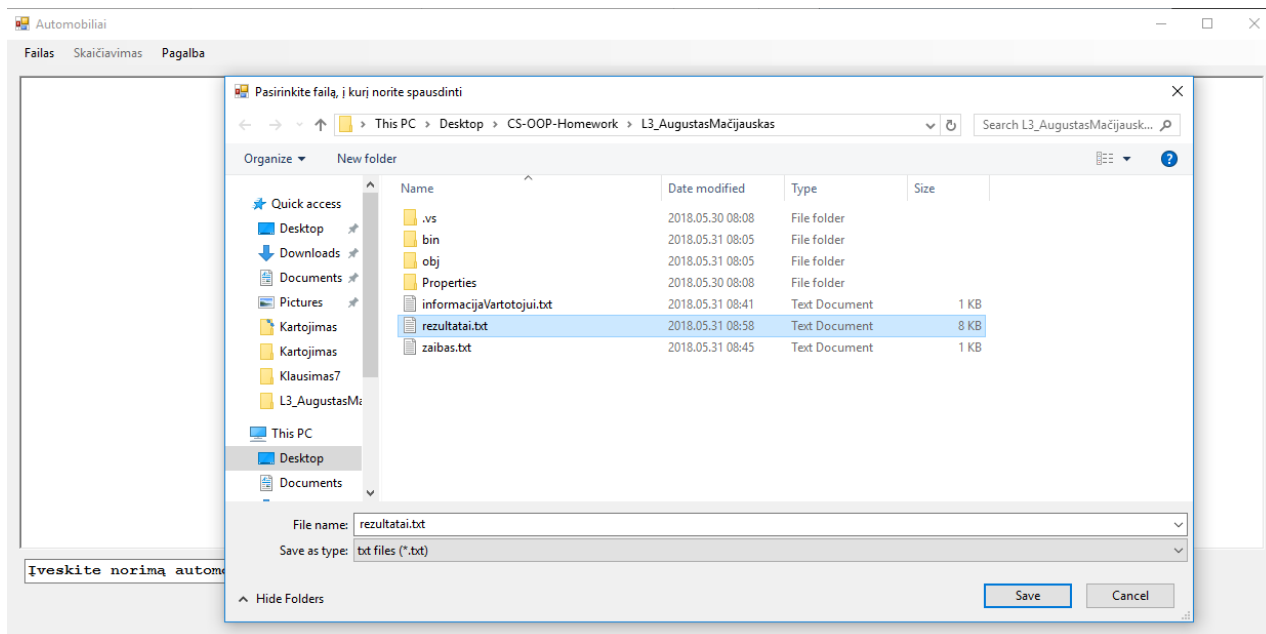
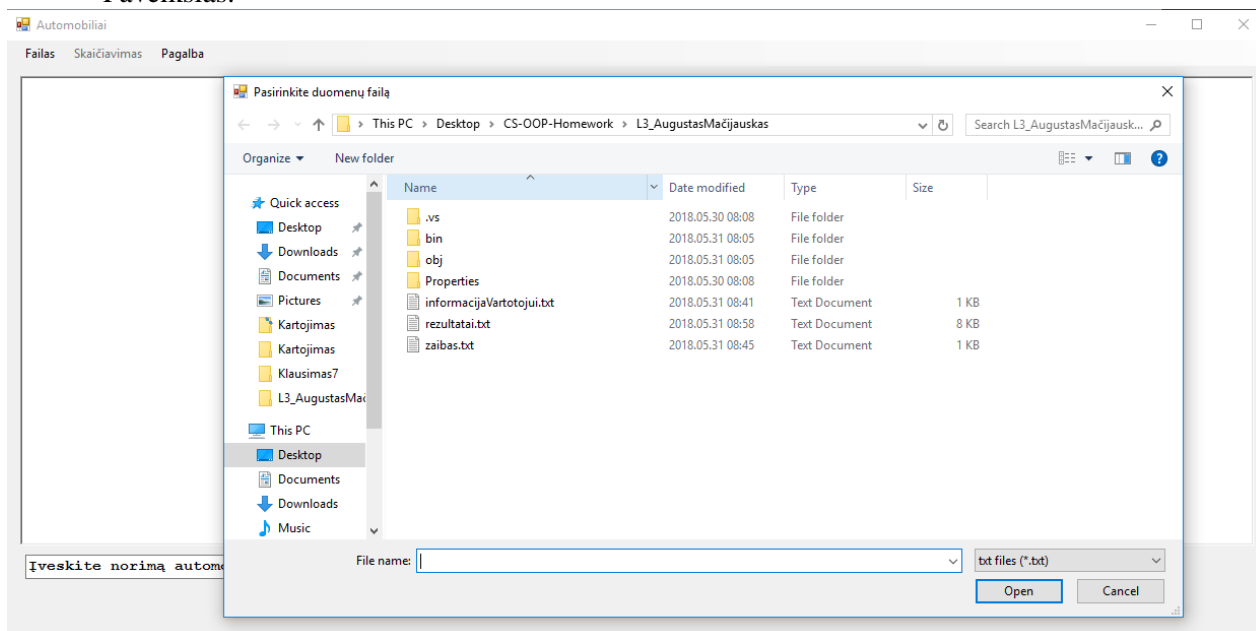
Schema:



Menu punktai:

Failas	Skaičiavimas	Pagalba
Įvesti	Skaičiuoti	Informacija
Baigti	Spausdinti	

Paveikslas:



Automobiliai

Failas Skaiciavimas Pagalba

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	OPG896	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	PLM562	Mercedes	W25 Silver Arrow	2011 01	2018 07 22	Dujos	10,0
10	KLGS76	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
11	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
12	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

Naujausias automobilis:

Valstybinis numeris : KLGS76

Gamintojas : Tesla

Modelis : ModelX

Pagaminimo data : 2017 12

Techninė apžiūra : 2018 06 20

Kuro tipas : Elektra

Kuro sąnaudos : 0

ModelX

Skaiciavimai atlikti. Galima spausdinti.

Automobiliai

Failas Skaiciavimas Pagalba

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	OPG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
2	KLGS76	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	OPG896	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	KLGS76	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0

ModelX

Skaiciavimai atlikti. Galima spausdinti.

Automobiliai

Failas Skaiciavimas Pagalba

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	OPG896	Tesla	ModelX	2009 06	2018 04 09	Hibridas	4,0
2	KLGS76	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	OPG896	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	KLGS76	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0

ModelX

Skaiciavimai atlikti. Galima spausdinti.

Automobiliai

Failas Skaičiavimas 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	PLM562	Mercedes	W25 Silver Arrow	2011 01	2018 07 22	Dujos	10,5
10	KLK576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
11	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
12	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

ModelX

Skaičiavimai atlikti. Galima spausdinti.

3.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
Form1	Text: Automobiliai Name: 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 amžių Font: Courier New, Bold, 11; Script: Baltic;	Norimo automobilio modelio 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 ToolStripMenu)	Text: Failas Font: Standartiniai nustatymai	Meniu punktas, kuriame yra mygtukai duomenų nuskaitymui ir programos darbo baigimui
investiToolStripMenuItem (tipas ToolStripMenu)	Name: Ivesti Text: Įvesti Font: Standartiniai nustatymai	Meniu punktas duomenims iš failo nuskaityti ir rezultatų failui pasirinkti
baigtiToolStripMenuItem (tipas ToolStripMenu)	Name: Baigti Text: Baigti Font: Standartiniai nustatymai	Meniu punktas, baigiantis darbą su programa
skaičiavimasToolStripMenuItem (tipas ToolStripMenu)	Text: Skaičiavimas Font: Standartiniai nustatymai	Meniu punktas, kuriame yra mygtukai skaičiavimams ir spausdinimui atlikti
skaiciuotiToolStripMenuItem (tipas ToolStripMenu)	Name: Skaiciuoti Text: Skaičiuoti Font: Standartiniai nustatymai	Meniu punktas skaičiavimams atlikti

spausdintiToolStripMenuItem (tipas ToolStripMenu)	Name: Spausdinti Text: Spausdinti Font: Standartiniai nustatymai	Menu punktas gautiems rezultatams atspausdinti
pagalbaToolStripMenuItem (tipas ToolStripMenu)	Text: Pagalba Font: Standartiniai nustatymai	Menu punktas, turintis mygtukus, kurie atvaizduoja nurodymus vartotojui ir naudojimo sąlygas.
informacijaToolStripMenuItem (tipas ToolStripMenu)	Text: Informacija Font: Standartiniai nustatymai	Menu punktas, spausdinantis pagalbą vartotojui.

3.4. Programos vartotojo vadovas

PASTABA: norėdami peržiūrėti pagalbą vartotojui, spauskite Pagalba -> Informacija. Norėdami baigti darbą su programa, spauskite Failas -> Baigti.

1. Atidarę programą, spauskite Failas -> Įvesti.
2. Pasirinkite duomenų failą, iš kurio norite nuskaityti.
3. Pasirinkite rezultatų failą, į kurį norite spausdinti.
4. Į teksto laukelį „Įveskite norimą automobilio modelį“ įveskite savo norimą automobilio modelį.
5. Spauskite Skaičiavimas -> Skaičiuoti. Bus atlikti skaičiavimai.
6. Kai skaičiavimai bus atlikti, spauskite Skaičiavimas -> Spausdinti.
7. Programos veikimas baigtas. Galite peržiūrėti gautus rezultatus ir uždaryti programą.

3.5. Programos tekstas

Abstrakti klasė AutomobilioBlueprint:

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

namespace L3_AugustasMačijskas
{
    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();
    }
}
```

AutomobilioBlueprint klasę paveldinti išvestinė klasė:

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

namespace L3_AugustasMačijauskas
{
    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 tekstas:

```
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 Autumobiliai : Form
    {
        List<Automobilis> automobiliai;
        List<Automobilis> markes;
        List<string> gamintojai;
        string rez = "";
        const string info = "..\\..\\informacijaVartotojui.txt";

        public Autumobiliai()
        {
            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 openFile = new OpenFileDialog();
            openFile.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
            openFile.Title = "Pasirinkite duomenų failą";
            DialogResult result = openFile.ShowDialog();
            if (result == DialogResult.OK)
            {
                string fn = openFile.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;
            Įvesti.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\r\n" +
        "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, markes, $"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;
        skaiciavimasToolStripMenuItem.Enabled = false;
    }

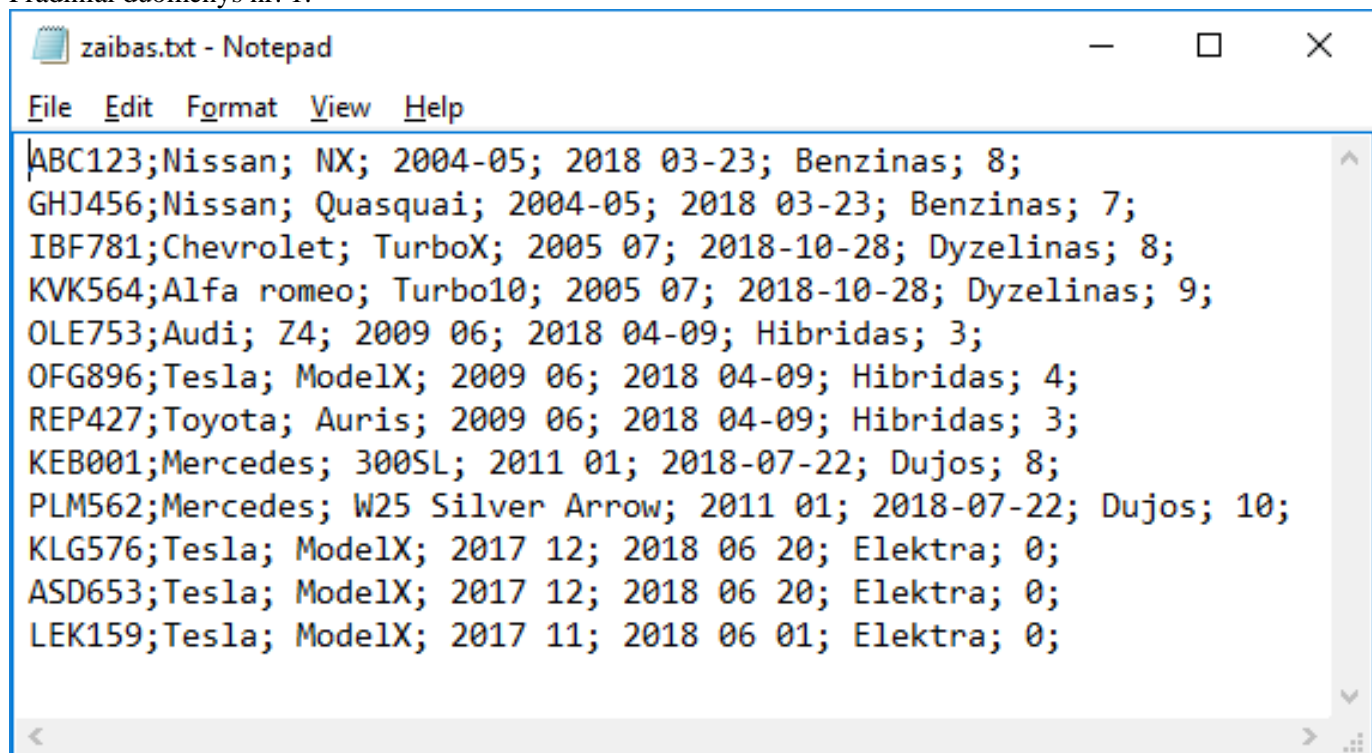
    private void Informacija_Click(object sender, EventArgs e)
    {
        label1.Text = "Peržiūrima informacija vartotojui.";
        string x = File.ReadAllText(info);
        MessageBox.Show(x);
    }

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

```

3.6. Pradiniai duomenys ir rezultatai

Pradiniai duomenys nr. 1:



Rezultatai nr. 1:

rezultatai.txt - Notepad

File Edit Format View Help

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	Z4	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	PLM562	Mercedes	W25 Silver Arrow	2011 01	2018 07 22	Dujos	10,0
10	KLG576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
11	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
12	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

Naujausias automobilis: Valstybinis numeris : KLG576 Gamintojas : Tesla

Antras naujausias automobilis: Valstybinis numeris : ASD653 Gamintojas : Tesla

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	KLG576	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	KLG576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0

rezultatai.txt - Notepad

File Edit Format View Help

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	KLG576	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	PLM562	Mercedes	W25 Silver Arrow	2011 01	2018 07 22	Dujos	10,5
10	KLG576	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
11	ASD653	Tesla	ModelX	2017 12	2018 06 20	Elektra	0,0
12	LEK159	Tesla	ModelX	2017 11	2018 06 01	Elektra	0,0

Pradiniai duomenys nr. 2:

```

zaibas.txt - 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. 2:

rezultatai Nr. 2:

rezultatai.txt - Notepad

File Edit Format View Help

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	Z4	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	PLM562	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

Antras naujausias automobilis: Valstybinis numeris : ASD653 Gamintojas : Tesla

W25 Silver Arrow markės automobiliai:

Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,0
2	ABC123	Mercedes	W25 Silver Arrow	2011 01	2018 12 31	Dujos	10,0
3	PLM562	Mercedes	W25 Silver Arrow	2011 04	2018 06 22	Dujos	10,0

Surikiuoti W25 Silver Arrow markės automobiliai:

Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	PLM562	Mercedes	W25 Silver Arrow	2011 04	2018 06 22	Dujos	10,0
2	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,0
3	ABC123	Mercedes	W25 Silver Arrow	2011 01	2018 12 31	Dujos	10,0

rezultatai.txt - Notepad							
File Edit Format View Help							
Surikiuoti W25 Silver Arrow markės automobiliai:							
Nr.	Valstybinis numeris	Gamintojas	Modelis	Pagaminimo data	Techninė apžiūra	Kuro tipas	Kuro sąnaudos
1	PLM562	Mercedes	W25 Silver Arrow	2011 04	2018 06 22	Dujos	10,0
2	ZLM562	Mercedes	W25 Silver Arrow	2011 06	2018 08 22	Dujos	10,0
3	ABC123	Mercedes	W25 Silver Arrow	2011 01	2018 12 31	Dujos	10,0
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	KLGS76	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

4. Susietasis sąrašas (L4)

4.1. 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.

4.2. Grafinės vartotojo sąsajos schema ir paveikslas

4.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė

4.4. Programos vartotojo vadovas

4.5. Programos tekstas

4.6. Pradiniai duomenys ir rezultatai

5. Bendrinės klasės (L5)

5.1. Darbo užduotis

5.2. Grafinės vartotojo sąsajos schema ir paveikslas

5.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė

5.4. Programos vartotojo vadovas

5.5. Programos tekstas

5.6. Pradiniai duomenys ir rezultatai