

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**

**Objektinio programavimo pagrindai II (P175B502)**  
***Laboratorinio darbo ataskaita***

Atliko:

IFIN5/3 gr. studentas

Karolis Staniukynas

2016 m. kovo 23 d.

Priėmė:

Lekt. Simonavičius Kęstutis

# TURINYS

<b>1. Grafinė vartotojo sąsaja ir algoritmų taikymas (L1)</b>	<b>3</b>
1.1. Darbo užduotis	3
1.2. Grafinės vartotojo sąsajos schema ir paveikslas	3
1.3. Sąsajoje panaudotų komponentų keičiamos savybės	3
1.4. Programos vartotojo vadovas	4
1.5. Programos tekstas	4
1.6. Pradiniai duomenys ir rezultatai	8
<b>2. Dinaminis masyvas (L2)</b>	<b>10</b>
2.1. Darbo užduotis	10
2.2. Grafinės vartotojo sąsajos schema ir paveikslas	10
2.3. Sąsajoje panaudotų komponentų keičiamos savybės	11
2.4. Programos vartotojo vadovas	11
2.5. Programos tekstas	11
2.6. Pradiniai duomenys ir rezultatai	19
<b>3. Paveldėjimas (L3)</b>	<b>22</b>
3.1. Darbo užduotis	22
3.2. Grafinės vartotojo sąsajos schema ir paveikslas	22
3.3. Sąsajoje panaudotų komponentų keičiamos savybės	22
3.4. Programos vartotojo vadovas	22
3.5. Programos tekstas	22
3.6. Pradiniai duomenys ir rezultatai	22
<b>4. Susietasis sąrašas (L4)</b>	<b>23</b>
4.1. Darbo užduotis	23
4.2. Grafinės vartotojo sąsajos schema ir paveikslas	23
4.3. Sąsajoje panaudotų komponentų keičiamos savybės	23
4.4. Programos vartotojo vadovas	23
4.5. Programos tekstas	23
4.6. Pradiniai duomenys ir rezultatai	23
<b>5. Bendrinės klasės (L5)</b>	<b>24</b>
5.1. Darbo užduotis	24
5.2. Grafinės vartotojo sąsajos schema ir paveikslas	24
5.3. Sąsajoje panaudotų komponentų keičiamos savybės	24
5.4. Programos vartotojo vadovas	24
5.5. Programos tekstas	24
5.6. Pradiniai duomenys ir rezultatai	24

# 1. Grafinė vartotojo sąsaja ir algoritmų taikymas (L1)

## 1.1. Darbo užduotis

U1-12. Žaidimas. Dabar labai populiarus žaidimas „Protmūšis“, kuriame dalyviai atsakinėja į įvairius klausimus. Klaipėdiečiai ir kauniečiai planuoja dalyvauti žaidime ir nori sudaryti komandą. Apie kiekvieną kandidatą žinoma informacija: vardas ir pavardė, amžius, lytis, išsilavinimas, keliuose žaidimuose dalyvavo, į kelis klausimus atsakė teisingai. Faile pirmoje eilutėje yra komandos pavadinimas. L1+L2+L4.

- Raskite kurioje miesto kandidatų sąrašė yra daugiau žaidėjų su aukštesniu išsilavinimu.
- Sudarykite dvi komandas kuriose žaidėjai būtų ne vyresni kaip m metų ir atsakę į daugiau negu k klausimų.
- Surikiuokite suformuotus sąrašus pagal vardus ir pavardes.
- Pašalinkite iš rezultatų sąrašų kandidatus, kurie neturi aukštojo išsilavinimo

## 1.2. Grafinės vartotojo sąsajos schema ir paveikslas

The image shows two overlapping Windows application windows. The top window, titled 'Form1', has a large 'RichTextBox' control in the center. Below it, there are two rows of 'Label' and 'TextBox' controls, and two 'Button' controls at the bottom. The bottom window, titled 'Protmusis', has two rows of 'Label' and 'TextBox' controls (labeled 'Metai' and 'Klausymai'), and two buttons labeled 'Vykdėti' and 'Baigti' at the bottom.

## 1.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
-------------	--------	---------

Form1	Programos langas	Text:Protmusis
Vykdyti (tipas Button)	Vygdo programa	Text: Vykdyti Font: Times New Roman, Bold, 18 Script: Baltic
Baigti (tipas Button)	Uždaro programa	Text: Baigti Font: Times New Roman, Bold, 18 Script: Baltic
Metai (tipas Label)	Nurodo ką įrašyti i textBox	Text: Metai Font: Times New Roman, Bold, 12 Script: Baltic
Klausymai (tipas Label)	Nurodo ką įrašyti i textBox	Text: Klausymai Font: Times New Roman, Bold, 12 Script: Baltic
Rez (tipas RichTextBox)	Atvaizduoti failo turinį	Font: Courier New, Bold, 12 Script: Baltic
textBox1	Metam įvesti	Font: Courier New, Bold, 12 Script: Baltic
textBox2	Klausymų skaičiui įvesti	Font: Courier New, Bold, 12 Script: Baltic

#### 1.4. Programos vartotojo vadovas

Paleidus programa, vartotojas turi įvesti metus ir klausymų skaičių. Paspaudus „Vykdyti“ mygtuka Randama kureme miesto kandidatų sąrašas yra daugiau žaidėjų su aukštesniu išsilavinimu. Sudaromos dvi komandas kuriose žaidėjai būtų ne vyresni kaip m metų ir atsakę į daugiau negu k klausimų. Surikiuoja suformuotus sąrašus pagal vardus ir pavardes. Pašalina iš rezultatų sąrašų kandidatus, kurie neturi aukštojo išsilavinimo. Rezultatai išvedami į duomenų failą ir parodomi programos lange.

#### 1.5. Programos tekstas

```
//Klase
class Zaidejas
{
    public string PavVard { get; set; }
    public string lyt { get; set; }
    public int age { get; set; }
    public string lav { get; set; }
    public int zaid_sk { get; set; }
    public int kl { get; set; }
    public Zaidejas(string vard,string lyt,int age,string lav,int zaid_sk,int
kl)
    {
        PavVard = vard;
        this.lyt = lyt;
        this.age = age;
        this.lav = lav;
        this.zaid_sk = zaid_sk;
        this.kl = kl;
    }
    public string imtlav() { return lav; }
    public override string ToString()
    {
        string eilute;
        eilute = string.Format("{0,-20}{1,5} {2,8}{3,12} {4,15} {5,15}",
PavVard, lyt, age, lav, zaid_sk, kl);
        return eilute;
    }
    public static bool operator>(Zaidejas zaid1,Zaidejas zaid2)
    {

```

```

        int p = String.Compare(zaid1.PavVard, zaid2.PavVard,
StringComparison.CurrentCulture);

        return (p < 0 || p == 0);

    }
    public static bool operator <(Zaidejas zaid1, Zaidejas zaid2)
    {
        int p = String.Compare(zaid1.PavVard, zaid2.PavVard,
StringComparison.CurrentCulture);

        return (p > 0 || p == 0);

    }

}
//Konteineris
class Komandos
{
    const int Cn = 500;
    private Zaidejas[] Kom;
    public int Kiek { get; set; }
    public Komandos()
    {
        Kiek = 0;
        Kom = new Zaidejas[Cn];
    }
    public Zaidejas ImtiZaideja(int i) { return Kom[i]; }
    public void DetiZaideja(Zaidejas zaid) { Kom[Kiek++] = zaid; }
    /// <summary>
    /// rikiuja pagal abc
    /// </summary>
    public void Rikiuoti()
    {
        for (int i = 0; i < Kiek; i++)
            for (int j = i + 1; j < Kiek; j++)
            {
                if (Kom[i] < Kom[j])
                {
                    Zaidejas zaid = Kom[i];
                    Kom[i] = Kom[j];
                    Kom[j] = zaid;
                }
            }
    }
    /// <summary>
    /// is sarasu pasalina nuos kurie neatitinka kriterijaus
    /// </summary>
    public void Salinti()
    {
        int z = 0;
        for (int i = 0; i < Kiek; i++)
            if (Kom[i].lav == "aukst")
                Kom[z++] = Kom[i];
        Kiek = z;
    }

}

}

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 U1_12.Žaidimas
{
    public partial class Form1 : Form
    {
        const string Kaun = "..\\..\\Kaunas.txt";
        const string Kl = "..\\..\\Klaipeda.txt";
        const string CFr = "..\\..\\Rezultatai.txt";
        Komandos Klaipeda;
        Komandos Kaunas;
        Komandos kauno;
        Komandos klaipedos;
        int m, k;
        public Form1()
        {
            InitializeComponent();
            if (File.Exists(CFr))
                File.Delete(CFr);
            // string kom1;
            // string kom2;
        }

        private void Vykdyti_Click(object sender, EventArgs e)
        {
            Vykdyti.Enabled = false;
            string kom1;
            string kom2;
            string daug;
            string ant1 = "Kaunas pradiniai duomenys";
            string ant2 = "Klaipeda pradiniai duomenys";
            m = int.Parse(textBox1.Text);
            k = int.Parse(textBox2.Text);

            Klaipeda = SkaitytiKom(Kl, out kom1);
            Kaunas = SkaitytiKom(Kaun, out kom2);

            Spausdinti(CFr, Kaunas, ant1);
            Spausdinti(CFr, Klaipeda, ant2);

            Rasti(Klaipeda, Kaunas, out daug);

            kauno = komN(Kaunas, m, k);
            klaipedos = komN(Klaipeda, m, k);

            kauno.Rikiuoti();
            klaipedos.Rikiuoti();
            Spausdinti(CFr, kauno, kom2 + " surikiuota abc");
            Spausdinti(CFr, klaipedos, kom1 + " surikiuota abc");
            kauno.Salinti();
            klaipedos.Salinti();
            spaust(CFr, daug);
            Spausdinti(CFr, kauno, kom2 + " Pasalinta");
            Spausdinti(CFr, klaipedos, kom1 + " Pasalinta");
            rez.LoadFile(CFr, RichTextBoxStreamType.PlainText);
        }
        /// <summary>
        /// skaito is failu ir deda i konteineri
        /// </summary>
        /// <param name="fv"></param>
        /// <param name="kom"></param>
        /// <returns></returns>
        static Komandos SkaitytiKom(string fv, out string kom)
    }
}

```

```

{
    Komandos Komand = new Komandos();

    using (StreamReader duom = new StreamReader(fv,
Encoding.GetEncoding(1257)))
    {
        string eilute = duom.ReadLine();
        kom = eilute;
        while ((eilute = duom.ReadLine()) != null)
        {
            string[] part = eilute.Split(';');
            string pavVard = part[0];
            string lyt = part[1];
            int age = int.Parse(part[2]);
            string lav = part[3];
            int zaid_sk = int.Parse(part[4]);
            int kl = int.Parse(part[5]);
            Zaidejas zaid = new Zaidejas(pavVard, lyt, age, lav, zaid_sk,
kl);

            Komand.DetiZaideja(zaid);
        }
    }
    return Komand;
}
/// <summary>
/// spausdina pirma rezultata
/// </summary>
/// <param name="fr"></param>
/// <param name="daug"></param>
static void spaust(string fr, string daug)
{
    using (var fv = new StreamWriter(File.Open(fr, FileMode.Append),
Encoding.GetEncoding(1257)))
    {
        fv.WriteLine(" ");
        fv.WriteLine("daugiausiai turinciu aukstaji issilavinima yra {0}
mieste", daug);
    }
}
/// <summary>
/// spausdina pradinis doum ir rezultatu lenteles
/// </summary>
/// <param name="fr"></param>
/// <param name="Komand"></param>
/// <param name="name"></param>
static void Spausdinti(string fr, Komandos Komand, string name)
{
    using (var fv = new StreamWriter(File.Open(fr, FileMode.Append),
Encoding.GetEncoding(1257)))
    {
        if (Komand.Kiek != 0)
        {
            fv.WriteLine(" ");
            fv.WriteLine(name);
            fv.WriteLine("vardas pavarde          Lytis    amzius
issilavinimas    zaidimu skaicius    Klausymai");
            fv.WriteLine("
");
            for (int i = 0; i < Komand.Kiek; i++)
            {
                Zaidejas zaid = Komand.ImtiZaideja(i);
                fv.WriteLine(zaid);
            }
        }
    }
}

```





Klaipeda  
 keostas kestutis;v;5;neaukst;5;8;  
 tomas tomutis;v;58;aukst;5;4;  
 antanas antis;v;38;aukst;5;4;

Kaunas  
 antis tortas;v;18;neaukst;5;4;  
 vardenis pav;v;28;aukst;5;4;  
 keksas Tortas;v;38;neaukst;5;4;

m=40 k=3

**Kaunas pradiniai duomenys**

vardas pavarde	Lytis	amzius	issilavinimas	zaidimu skaicius	
<b>Klausymai</b>					
antis tortas	v	18	neaukst	5	4
vardenis pav	v	28	aukst	5	4
keksas Tortas	v	38	neaukst	5	4

**Klaipeda pradiniai duomenys**

vardas pavarde	Lytis	amzius	issilavinimas	zaidimu skaicius	
<b>Klausymai</b>					
keostas kestutis	v	5	neaukst	5	8
tommas tomutis	v	58	aukst	5	4
antanas antis	v	38	aukst	5	4

daugiausiai turinciu aukstaji issilavinima yra Klaipeda mieste

**Kaunas**

vardas pavarde	Lytis	amzius	issilavinimas	zaidimu skaicius	
<b>Klausymai</b>					
vardenis pav	v	28	aukst	5	4

**Klaipeda**

vardas pavarde	Lytis	amzius	issilavinimas	zaidimu skaicius	
<b>Klausymai</b>					
antanas antis	v	38	aukst	5	4

## 2. Dinaminis masyvas (L2)

### 2.1. Darbo užduotis

U1-12. Žaidimas. Dabar labai populiarus žaidimas „Protmūšis“, kuriame dalyviai atsakinėja į įvairius klausimus. Klaipėdiečiai ir kauniečiai planuoja dalyvauti žaidime ir nori sudaryti komandą. Apie kiekvieną kandidatą žinoma informacija: vardas ir pavardė, amžius, lytis, išsilavinimas, keliuose žaidimuose dalyvavo, į kelis klausimus atsakė teisingai. Faile pirmoje eilutėje yra komandos pavadinimas. L1+L2+L4.

- Raskite kurioje miesto kandidatų sąrašė yra daugiau žaidėjų su aukštesniu išsilavinimu.
- Sudarykite dvi komandas kuriose žaidėjai būtų ne vyresni kaip m metų ir atsakę į daugiau negu k klausimų.
- Surikiuokite suformuotus sąrašus pagal vardus ir pavardes.
- Pašalinkite iš rezultatų sąrašų kandidatus, kurie neturi aukštojo išsilavinimo.

#### L2 papildymas.

- Papildykite vieną iš surikiuotų rezultatų sąrašų naujais žaidėjais, duomenys apie kuriuos yra faile.

### 2.2. Grafinės vartotojo sąsajos schema ir paveikslas

The image displays two screenshots of a Windows Forms application, both titled 'Form1'. The top screenshot shows a form with a menu bar containing 'ToolStripMenuItem'. Below the menu bar is a large text area labeled 'richTextBox1'. At the bottom of the form, there are three labels: 'Label 1', 'Label2', and 'label3', each followed by an input field. The bottom screenshot shows the same form with a menu bar containing 'Failas' and 'Skaiciavimas'. Below the menu bar is a large text area. At the bottom of the form, there are three labels: 'Amzius', 'Kiekis', and 'label3', each followed by an input field.

### 2.3. Sąsajoje panaudotų komponentų keičiamos savybės

Komponentas	Savybė	Reikšmė
Form1	Programos Langas	Text:Form1
richTextBox1 (tipas RichTextBox)	Atvaizduoti duomenų failo turinį.	Font: Courier New, Bold, 12 Script: Baltic
failasToolStripMenuItem	Darbas su failais.	Text: Failas Font: Segoe UI, Regular, 12 Script: Baltic
atidarytiToolStripMenuItem	Darbas su failais.	Text:Atidaryti Font: Segoe UI, Regular, 12 Script: Baltic
toolStripMenuItem3	Uzdaro programą.	Text: Baigti Font: Segoe UI, Regular, 12 Script: Baltic
skaiciavimasToolStripMenuItem	Darbas su duomenimis.	Text: Skaiciavimas Font: Segoe UI, Regular, 12 Script: Baltic
toolStripMenuItem4	Vykdo skaičiavimus ir išveda duomenis i txt failą bei programos langą.	Text: Vykdyti Font: Segoe UI, Regular, 12 Script: Baltic
kurSaugotiTxtToolStripMenuItem	Duomenys saugomi csv faile	Text: Kur saugoti csv Font: Segoe UI, Regular, 12 Script: Baltic
textBox1	Įvedamas amžius	Font: Courier New, Bold, 12 Script: Baltic
textBox2	Įvedamas klausimų skaičius	Font: Courier New, Bold, 12 Script: Baltic
label1	Nusako ką įvesti	Text: Amžius Font: Times New Roman, Bold, 12 Script: Baltic
label2	Nusako ką įvesti	Text: Kiekis Font: Times New Roman, Bold, 12 Script: Baltic
label3	Parodo ats	Font: Times New Roman, Bold, 12 Script: Baltic

### 2.4. Programos vartotojo vadovas

Duomenys įkeliami nuvedus pelės rodyklę ant menu lango „Failas“ ir paspaudus „Atidaryti“.

Failai pasirenkami eilės tvarka (Kaunas.txt, Klaipeda.txt, Papild.txt) ir ne kitaip. Tada menu punkte „Skaiciavimai“ tampa aktyvus, pirmumas yra reikiamas mygtukui „Atrinkti“ tada galima atlikti tolesnius reikiamus skaičiavimus, kiekvieno etapo metu atsiveria langas kuriame reikia pasirinkti rezultato failo pavadinimą ir jo saugojimo vietą. Atlikus prieš tai buvusių veiksmus „Kur saugoti csv“ tampa aktyvus. Paspaudus jį, reikia pasirinkti kur norėsite išsaugoti .csv formato lentelę. Norint baigti darbą „Failas“ punkte, reikia paspausti „Baigti“.

### 2.5. Programos tekstas

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

namespace lab2_
{
```

```

class Zaidejai: IComparable<Zaidejai>
{
    public string PavVard { get; set; }
    public string lyt { get; set; }
    public int age { get; set; }
    public string lav { get; set; }
    public int zaid_sk { get; set; }
    public int kl { get; set; }
    public Zaidejai(string vard, string lyt, int age, string lav, int zaid_sk,
int kl)
    {
        PavVard = vard;
        this.lyt = lyt;
        this.age = age;
        this.lav = lav;
        this.zaid_sk = zaid_sk;
        this.kl = kl;
    }
    public override string ToString()
    {
        string eilute;
        eilute = string.Format("{0,-20},{1,5},{2,8},{3,12},{4,15},{5,15}",
PavVard, lyt, age, lav, zaid_sk, kl);
        return eilute;
    }
    public override bool Equals(object objektas)
    {
        Zaidejai stud = objektas as Zaidejai;
        return stud.PavVard == PavVard;
    }
    // Užklotas metodas GetHashCode()
    public override int GetHashCode()
    {
        return base.GetHashCode();
    }
    public int CompareTo(Zaidejai kitas)
    {
        int poz = String.Compare(this.PavVard, kitas.PavVard,
StringComparison.CurrentCulture);

        if (poz > 0)
            return 1;
        if (poz < 0)
            return -1;
        else
            return 0;
    }

    public static bool operator >(Zaidejai zaid1,Zaidejai zaid2)
    {
        int p = String.Compare(zaid1.PavVard, zaid2.PavVard,
StringComparison.CurrentCulture);
        return (p < 0 || p == 0);
    }
    public static bool operator <(Zaidejai zaid1, Zaidejai zaid2)
    {
        int p = String.Compare(zaid1.PavVard, zaid2.PavVard,
StringComparison.CurrentCulture);
        return (p > 0 || p == 0);
    }
}

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 lab2_
{
    public partial class Form1 : Form
    {
        private List<Zaidejai> Kaunas;
        private List<Zaidejai> Klaipeda;
        private List<Zaidejai> Kauno;
        private List<Zaidejai> Klaipedos;
        private List<Zaidejai> Iterpti;
        string kom1;
        string kom2;
        string kom;
        public Form1()
        {
            InitializeComponent();
        }

        private void atidarytiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog1 = new OpenFileDialog();
            openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*";
            openFileDialog1.Title = "Pasirinkite duomenų failą";
            DialogResult result = openFileDialog1.ShowDialog();
            if (result == DialogResult.OK) // jeigu pasirinktas failas
            {
                string fv = openFileDialog1.FileName;
                Kaunas = skaityti(fv, out kom1);
            }
            OpenFileDialog openFileDialog2 = new OpenFileDialog();
            openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*";
            openFileDialog1.Title = "Pasirinkite duomenų failą";
            DialogResult results = openFileDialog2.ShowDialog();
            if (result == DialogResult.OK) // jeigu pasirinktas failas
            {
                string fv = openFileDialog2.FileName;
                Klaipeda = skaityti(fv, out kom2);
            }
            OpenFileDialog openFileDialog3 = new OpenFileDialog();
            openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*";
            openFileDialog1.Title = "Pasirinkite duomenų failą";
            DialogResult resultat = openFileDialog3.ShowDialog();
            if (result == DialogResult.OK) // jeigu pasirinktas failas
            {
                string fv = openFileDialog3.FileName;
                Iterpti = skaityti(fv, out kom);
            }
            skaiciavimasToolStripMenuItem.Enabled = true;
        }
    }
}

```

```

k)
    /// <summary>
    /// Is atskiro failo papildo tam tikra lista
    /// </summary>
    /// <param name="komand"></param>
    /// <param name="Iterp"></param>
    static void Papild(List<Zaidejai> komand, List<Zaidejai> Iterp, int m, int
    {
        int q = 0;
        for (int i = 0; i < Iterp.Count; i++)
        {
            q = 0;
            if (Iterp[i].age < m && Iterp[i].kl > k && Iterp[i].lav != "neaukst")
            {
                for (int j = 0; j < komand.Count; j++)
                {
                    if (Iterp[i] > komand[j])
                    {
                        komand.Insert(j, Iterp[i]);
                        q++;
                        break;
                    }
                }
                if (q == 0)
                    komand.Insert(komand.Count, Iterp[i]);
            }
        }
    }
    /// <summary>
    /// Isvedama i csv faila
    /// </summary>
    /// <param name="kom"></param>
    /// <param name="fv"></param>
    /// <param name="pav"></param>
    static void isvest(List<Zaidejai> kom, string fv, string pav)
    {
        using (var fr = new StreamWriter(File.Open(fv, FileMode.Append)))
        {
            fr.WriteLine(pav);
            if (kom.Count == 0)
            {
                fr.WriteLine("Nera");
            }
            else
            {
                fr.WriteLine("vardas pavarde,          Lytis,    amzius,
issilavinimas,  zaidimu skaicius,  Klausymai,");
                for (int i = 0; i < kom.Count; i++)
                {
                    Zaidejai zaid = kom[i];
                    fr.WriteLine(zaid);
                }
                fr.WriteLine(" ");
            }
        }
    }
    /// <summary>
    /// salina netinkamus kandidatus
    /// </summary>
    /// <param name="kom"></param>
    static void Salinti(List<Zaidejai> kom)
    {

```

```

        bool ne = false;
        for (int i = 0; i < kom.Count; i++)
        {
            if (ne == true)
            {
                i--;
            }
            ne = false;
            if (kom[i].lav == "neaukst")
            {
                kom.RemoveAt(i);
                ne = true;
            }
        }

    }

    /// <summary>
    /// atrenka
    /// </summary>
    /// <param name="kom"></param>
    /// <param name="m"></param>
    /// <param name="k"></param>
    /// <returns></returns>
    static List<Zaidejai> atrinkti(List<Zaidejai> kom, int m, int k)
    {
        List<Zaidejai> info = new List<Zaidejai>();
        for (int i = 0; i < kom.Count; i++)
        {
            if (kom[i].age < m && kom[i].kl > k)
                info.Add(kom[i]);
        }
        return info;
    }

    /// <summary>
    /// skaito
    /// </summary>
    /// <param name="fv"></param>
    /// <param name="kom"></param>
    /// <returns></returns>
    static List<Zaidejai> skaityti(string fv, out string kom)
    {
        List<Zaidejai> info = new List<Zaidejai>();
        using (StreamReader srautas = new StreamReader(fv))
        {
            string eilute;
            kom = srautas.ReadLine();
            while ((eilute = srautas.ReadLine()) != null)
            {
                string[] dalis = eilute.Split(';');
                string pavVard = dalis[0];
                string lyt = dalis[1];
                int age = int.Parse(dalis[2]);
                string lav = dalis[3];
                int zaid_sk = int.Parse(dalis[4]);
                int kl = int.Parse(dalis[5]);
                Zaidejai zaid = new Zaidejai(pavVard, lyt, age, lav, zaid_sk,
kl);

                info.Add(zaid);
            }
        }

        return info;
    }
}

```

```

#####PATAISYTA#####
/// <summary>
/// randa kuriame sarase yra daugiausiai su aukstuoju issilavinimu
/// </summary>
/// <param name="pirm"></param>
/// <param name="antr"></param>
/// <param name="daug"></param>
daug) static void Rasti(List<Zaidejai> pirm, List<Zaidejai> antr, out string
{
    daug = "Lygu";
    int kiek1 =kuris(pirm);
    int kiek2 =kuris(antr);
    if ( kiek1> kiek2)
        daug = "Kaunas";
    if (kiek1 < kiek2)
        daug = "Klaipeda";

}
static int kuris(List<Zaidejai> kom)
{
    int kiekis=0;
    for (int i = 0; i < kom.Count; i++)
        if (kom[i].lav == "aukst")
            kiekis++;
    return kiekis;
}

#####
/// <summary>
/// spausdina i txt faila
/// </summary>
/// <param name="fv"></param>
/// <param name="info"></param>
/// <param name="antr"></param>
/// <param name="kom"></param>
kom) static void spaust(string fv, List<Zaidejai> info, string antr, string
{
    using (var fr = new StreamWriter(File.Open(fv, FileMode.Append)))
    {
        if (info.Count == 0)
        {
            fr.WriteLine(antr);
            fr.WriteLine(kom);
            fr.WriteLine("Nera");
        }
        else
        {
            fr.WriteLine(antr);
            fr.WriteLine(kom);
            fr.WriteLine("vardas pavarde            Lytis    amzius
issilavinimas  zaidimu skaicius  Klausymai");
        }
    }
    fr.WriteLine("
");
    for (int i = 0; i < info.Count; i++)
    {
        Zaidejai zaid = info[i];
        fr.WriteLine(zaid);
    }
    fr.WriteLine(" ");
}
}

```



```

// BAIGTI
private void toolStripMenuItem3_Click(object sender, EventArgs e)
{
    Close();
}
/// <summary>
/// praso nurodyti kur saugoti csv faila
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void kurSaugotiTxtToolStripMenuItem_Click(object sender, EventArgs
e)
{
    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Filter = "csv files (*.csv)|*.csv|All files
(*.*)|*.*";
    saveFileDialog1.Title = "Pasirinkite rezultatu faila";
    DialogResult result = saveFileDialog1.ShowDialog();
    if (result == DialogResult.OK) // jeigu pasirinktas failas
    {
        string fv = saveFileDialog1.FileName;
        // Jeigu reikia rezultatu failas išvalomas
        if (File.Exists(fv))
            File.Delete(fv);
        isvest(Kauno, fv, kom1);
        isvest(Klaipedos, fv, kom2);
    }
}

private void toolStripMenuItem2_Click(object sender, EventArgs e)
{
    int m = int.Parse(textBox1.Text);
    int k = int.Parse(textBox2.Text);
    string daug;
    Rasti(Kaunas, Klaipeda, out daug);
    label3.Text = "Daugiausiai su aukstuoju yra " + daug;
    label3.Visible = true;

    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*";
    saveFileDialog1.Title = "Pasirinkite rezultatu faila";
    DialogResult result = saveFileDialog1.ShowDialog();
    if (result == DialogResult.OK)
        // jeigu pasirinktas failas
    {
        string rez = saveFileDialog1.FileName;
        // Jeigu reikia rezultatu failas išvalomas
        if (File.Exists(rez))
            File.Delete(rez);
        Kauno = atrinkti(Kaunas, m, k);
        Klaipedos = atrinkti(Klaipeda, m, k);

        string antr = "Atrinkti";
        spaust(rez, Kauno, antr, kom1);
        spaust(rez, Klaipedos, antr, kom2);
        spausdinti(daug, rez);
        rezultatai.LoadFile(rez, RichTextBoxStreamType.PlainText);
    }
    kurSaugotiTxtToolStripMenuItem.Enabled = true;
}

```

```

public void spausdinti(string daug, string fv)
{
    using (var fr = new StreamWriter(File.Open(fv, FileMode.Append)))
    {
        fr.WriteLine(" Daugiausiai su aukstuoju yra "+daug);
    }
}

private void rikiuotToolStripMenuItem_Click(object sender, EventArgs e)
{
    int m = int.Parse(textBox1.Text);
    int k = int.Parse(textBox2.Text);
    string daug;
    Rasti(Kaunas, Klaipeda, out daug);

    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*"";
    saveFileDialog1.Title = "Pasirinkite rezultatų failą";
    DialogResult result = saveFileDialog1.ShowDialog();
    if (result == DialogResult.OK)
    // jeigu pasirinktas failas
    {
        string rez = saveFileDialog1.FileName;
        // Jeigu reikia rezultatų failas išvalomas

        Kauno.Sort();
        Klaipedos.Sort();
        Iterpti.Sort();
        string antr = "surikiuota";
        spaust(rez, Kauno, antr, kom1);
        spaust(rez, Klaipedos, antr, kom2);
        rezultatai.LoadFile(rez, RichTextBoxStreamType.PlainText);
    }
}

private void salintiToolStripMenuItem_Click(object sender, EventArgs e)
{
    int m = int.Parse(textBox1.Text);
    int k = int.Parse(textBox2.Text);
    string daug;
    Rasti(Kaunas, Klaipeda, out daug);

    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*"";
    saveFileDialog1.Title = "Pasirinkite rezultatų failą";
    DialogResult result = saveFileDialog1.ShowDialog();
    if (result == DialogResult.OK)
    // jeigu pasirinktas failas
    {
        string rez = saveFileDialog1.FileName;
        // Jeigu reikia rezultatų failas išvalomas

        Salinti(Kauno);
        Salinti(Klaipedos);

        string antr = "pasalinta";
        spaust(rez, Kauno, antr, kom1);
        spaust(rez, Klaipedos, antr, kom2);
        rezultatai.LoadFile(rez, RichTextBoxStreamType.PlainText);
    }
}

```

```

    }
}

private void papildytiToolStripMenuItem_Click(object sender, EventArgs e)
{
    int m = int.Parse(textBox1.Text);
    int k = int.Parse(textBox2.Text);
    string daug;
    Rasti(Kaunas, Klaipeda, out daug);

    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*.txt";
    saveFileDialog1.Title = "Pasirinkite rezultatų failą";
    DialogResult result = saveFileDialog1.ShowDialog();
    if (result == DialogResult.OK) // jeigu pasirinktas failas
    {
        string rez = saveFileDialog1.FileName;
        // Jeigu reikia rezultatų failas išvalomas

        if (kom == kom1)
            Papild(Kauno, Iterpti,m,k);
        if (kom == kom2)
            Papild(Klaipedos, Iterpti,m,k);

        string antr = "Iterpta";

        spaust(rez, Kauno, antr, kom1);
        spaust(rez, Klaipedos, antr, kom2);
        rezultatai.LoadFile(rez, RichTextBoxStreamType.PlainText);
    }
}
}
}

```

## 2.6. Pradiniai duomenys ir rezultatai

Klaipeda  
 keštas keštutis;v;5;neaukst;5;8;  
 tomas tomutis;v;58;aukst;5;4;  
 antanas antis;v;38;aukst;5;4;

Kaunas  
 antis tortas;v;18;neaukst;5;4;  
 vardenis pav;v;28;aukst;5;4;  
 keksas Tortas;v;38;neaukst;5;4;

Kaunas  
 mew keštutis;v;5;aukst;5;8;  
 tomas tomutis;v;58;aukst;5;4;  
 bam antis;v;38;aukst;5;4;

m=40 k=3

Atrinkti  
Kaunas  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

antis tortas	,	v,	18,	neaukst,	5,	4,
vardenis pav	,	v,	28,	aukst,	5,	4,
keksas Tortas	,	v,	38,	neaukst,	5,	4,

Atrinkti  
Klaipeda  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

kestas kestutis	,	v,	5,	aukst,	5,	8,
antanas antis	,	v,	38,	aukst,	5,	4,

Daugiausiai su aukstuoju yra Klaipeda

surikiuota  
Kaunas  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

antis tortas	,	v,	18,	neaukst,	5,	4,
keksas Tortas	,	v,	38,	neaukst,	5,	4,
vardenis pav	,	v,	28,	aukst,	5,	4,

surikiuota  
Klaipeda  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

antanas antis	,	v,	38,	aukst,	5,	4,
kestas kestutis	,	v,	5,	aukst,	5,	8,

pasalinta  
Kaunas  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

vardenis pav	,	v,	28,	aukst,	5,	4,
--------------	---	----	-----	--------	----	----

pasalinta  
Klaipeda  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

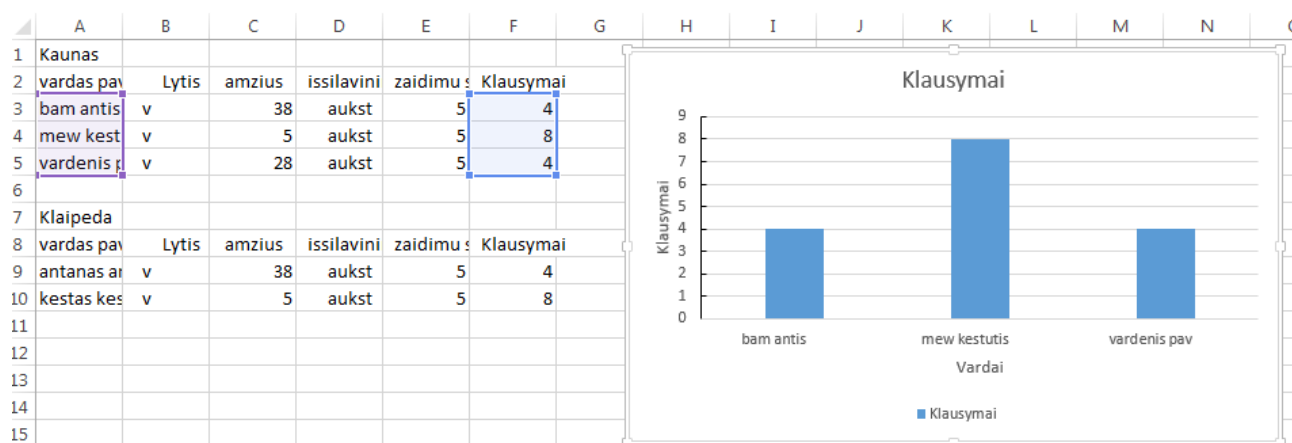
antanas antis	,	v,	38,	aukst,	5,	4,
kestas kestutis	,	v,	5,	aukst,	5,	8,

Iterpta  
Kaunas  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

bam antis	,	v,	38,	aukst,	5,	4,
mew kestutis	,	v,	5,	aukst,	5,	8,
vardenis pav	,	v,	28,	aukst,	5,	4,

Iterpta  
Klaipeda  
vardas pavarde                      Lytis      amzius      issilavinimas      zaidimu skaicius      Klausymai

antanas antis , v, 38, aukst, 5, 4,  
 keštas kešutis , v, 5, aukst, 5, 8,



### **3. Paveldėjimas (L3)**

#### **3.1. Darbo užduotis**

#### **3.2. Grafinės vartotojo sąsajos schema ir paveikslas**

#### **3.3. Sąsajoje panaudotų komponentų keičiamos savybės**

Komponentas	Savybė	Reikšmė

#### **3.4. Programos vartotojo vadovas**

#### **3.5. Programos tekstas**

#### **3.6. Pradiniai duomenys ir rezultatai**

## **4. Susietasis sąrašas (L4)**

### ***4.1. Darbo užduotis***

### ***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***