KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos FAKULTETAS

Objektinio programavimo pagrindai II (P175B502)

Laboratorinio darbo ataskaita

Atliko:

IFIN5/3 gr. studentas

Karolis Staniukynas

2016 m. gegužės 27 d.

Priėmė:

Lekt. Simonavičius Kęstutis

KAUNAS 2016

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 3

1.4. Programos vartotojo vadovas 4

1.5. Programos tekstas 4

1.6. Pradiniai duomenys ir rezultatai 8

2. Dinaminis masyvas (L2) 10

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

3. Paveldėjimas (L3) 22

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 23

3.5. Programos tekstas 23

3.6. Pradiniai duomenys ir rezultatai 31

4. Susietasis sąrašas (L4) 33

4.1. Darbo užduotis 33

4.2. Grafinės vartotojo sąsajos schema ir paveikslas 33

4.3. Sąsajoje panaudotų komponentų keičiamos savybės 33

4.4. Programos vartotojo vadovas 34

4.5. Programos tekstas 34

4.6. Pradiniai duomenys ir rezultatai 41

5. Bendrinės klasės (L5) 43

5.1. Darbo užduotis 43

5.2. Grafinės vartotojo sąsajos schema ir paveikslas 43

5.3. Sąsajoje panaudotų komponentų keičiamos savybės 43

5.4. Programos vartotojo vadovas 44

5.5. Programos tekstas 44

5.6. Pradiniai duomenys ir rezultatai 51

# Grafinė vartotojo sąsaja ir algoritmų taikymas (L1)

## 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.

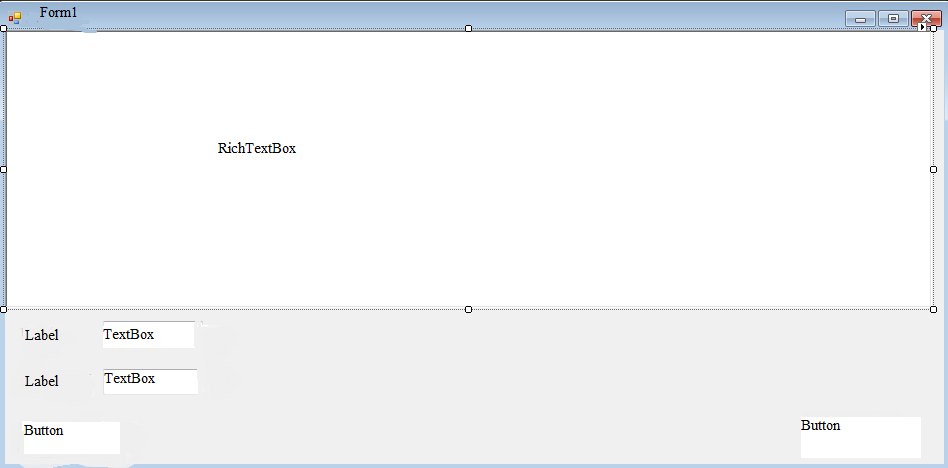
• Raskite kurioje miesto kandidatų sąraše yra daugiau žaidėjų su aukštuoju išsilavinimu.

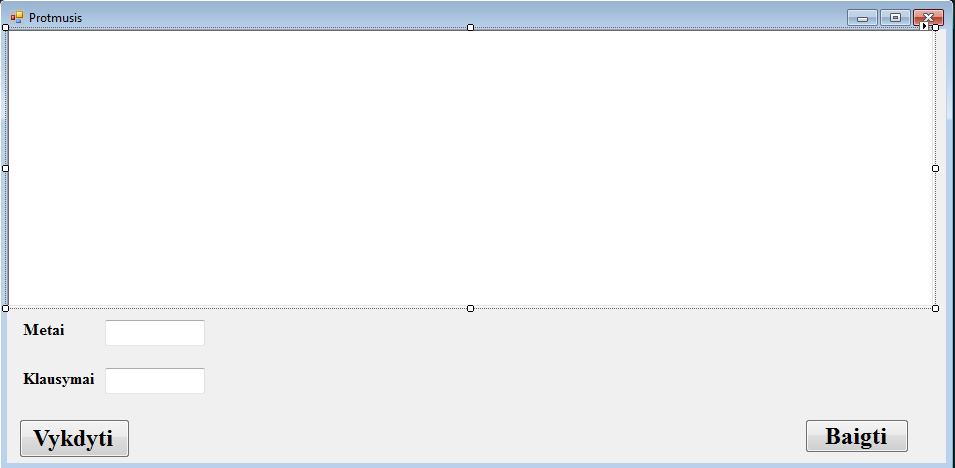
• Sudarykite dvi komandas kuriose žaidėjai būtų ne vyresni kaip m metų ir atsakę į daugiau negu k klausimų.

• Surikiuokite suformuotus sąrašus pagal vardus ir pavardes.

• Pašalinkite iš rezultatų sąrašų kandidatus, kurie neturi aukštojo išsilavinimo

## Grafinės vartotojo sąsajos schema ir paveikslas





## 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 |

## Programos vartotojo vadovas

Paleidus programa, vartotojas turi įvesti metus ir klausymų skaičių. Paspaudus „Vykdyti“ mygtuka Randama kureme miesto kandidatų sąraše yra daugiau žaidėjų su aukštuoju 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 isvedami į duomenų failą ir parodomi programos lange.

## 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 pradinius 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);

}

}

else

{

fv.WriteLine(" ");

fv.WriteLine(name);

fv.WriteLine(" Nera");

}

}

}/// <summary>

/// randa kuriam mieste yra daukiausiai aukstaji issilavinima turinciu kandidatu

/// </summary>

/// <param name="kom1"></param>

/// <param name="kom2"></param>

/// <param name="daug"></param>

static void Rasti(Komandos kom1,Komandos kom2 ,out string daug)

{

daug = " Lygu ";

int kiek=0;

int kiek2=0;

for (int i = 0; i < kom1.Kiek; i++)

if(kom1.ImtiZaideja(i).lav=="aukst")

{

kiek++;

}

for (int i = 0; i < kom2.Kiek; i++)

if (kom2.ImtiZaideja(i).lav == "aukst")

{

kiek2++;

}

if (kiek > kiek2)

daug = "Klaipeda";

if (kiek < kiek2)

daug = "Kaunas";

//else

// daug = "abu lygu";

}

private void Baigti\_Click(object sender, EventArgs e)

{

Close();

}

/// <summary>

/// sudaro komandas

/// </summary>

/// <param name="miest"></param>

/// <param name="m"></param>

/// <param name="k"></param>

/// <returns></returns>

static Komandos komN(Komandos miest,int m,int k)

{

Komandos kom = new Komandos();

for(int i=0;i<miest.Kiek;i++)

if(miest.ImtiZaideja(i).age<m && miest.ImtiZaideja(i).kl>k)

{

kom.DetiZaideja(miest.ImtiZaideja(i));

}

return kom;

}

}

}

## Pradiniai duomenys ir rezultatai

Klaipeda

kestas 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 Klausymai**

**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 Klausymai**

**kestas kestutis v 5 neaukst 5 8**

**tomas 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 Klausymai**

**vardenis pav v 28 aukst 5 4**

**Klaipeda**

**vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai**

**antanas antis v 38 aukst 5 4**

# Dinaminis masyvas (L2)

## 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 sąraše yra daugiau žaidėjų su aukštuoju išsilavinimu.

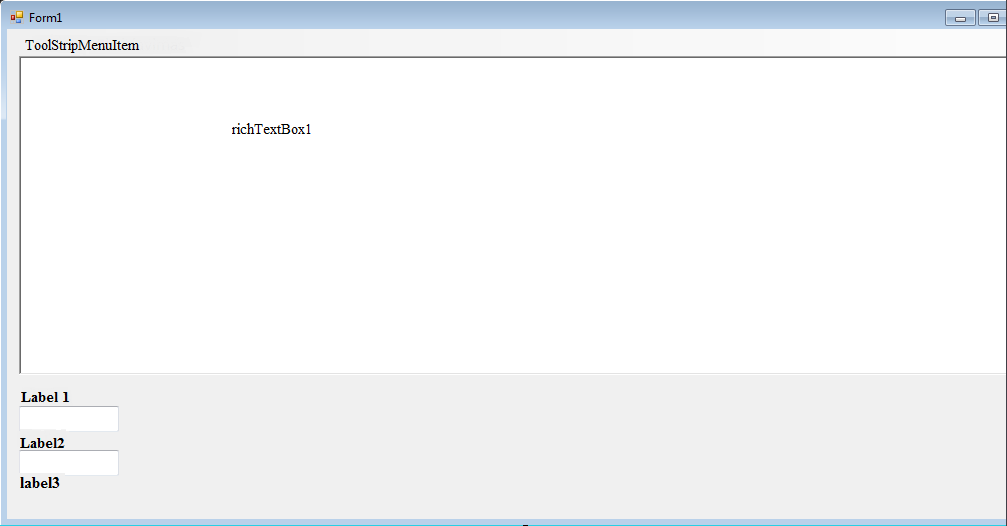
• Sudarykite dvi komandas kuriose žaidėjai būtų ne vyresni kaip m metų ir atsakę į daugiau negu k klausimų. • Surikiuokite suformuotus sąrašus pagal vardus ir pavardes.

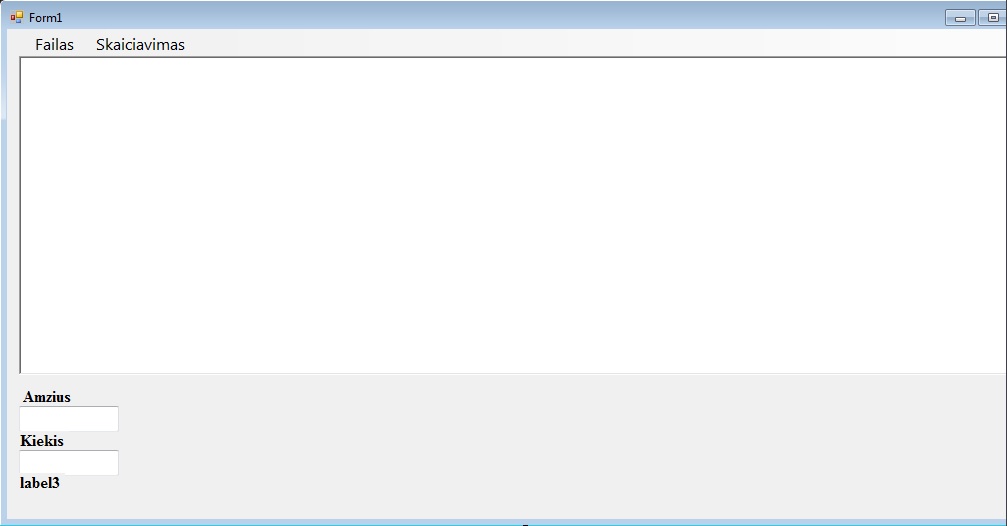
• Pašalinkite iš rezultatų sąrašų kandidatus, kurie neturi aukštojo išsilavinimo.

**L2 papildymas.**

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

## Grafinės vartotojo sąsajos schema ir paveikslas





## 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 isveda 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 |

## 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 buvusius veiksmus „Kur saugoti csv“ tampa aktyvus. Paspaudus jį, reikia pasirinkti kur norėsite išsaugoti .csv formato lentelę. Norint baigti darbą „Failas“ punkte, reikia paspausti „Baigti“.

## 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;

}

/// <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 k)

{

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>

static void Rasti(List<Zaidejai> pirm, List<Zaidejai> antr, out string daug)

{

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>

static void spaust(string fv, List<Zaidejai> info, string antr, string kom)

{

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 rezultatų failą";

DialogResult result = saveFileDialog1.ShowDialog();

if (result == DialogResult.OK) // jeigu pasirinktas failas

{

string fv = saveFileDialog1.FileName;

// Jeigu reikia rezultatų 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 rezultatų failą";

DialogResult result = saveFileDialog1.ShowDialog();

if (result == DialogResult.OK)

// jeigu pasirinktas failas

{

string rez = saveFileDialog1.FileName;

// Jeigu reikia rezultatų 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 salintToolStripMenuItem\_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 (\*.\*)|\*.\*";

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);

}

}

}

}

## Pradiniai duomenys ir rezultatai

Klaipeda

kestas 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;

Kaunas

mew kestutis;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

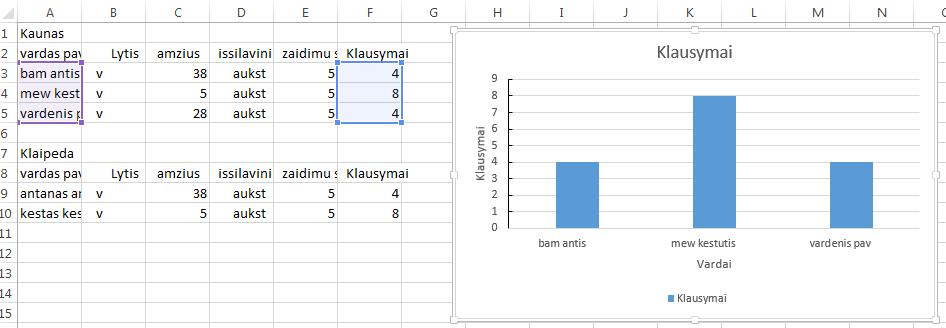
Klaipeda

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

antanas antis , v, 38, aukst, 5, 4,

kestas kestutis , v, 5, aukst, 5, 8,



# Paveldėjimas (L3)

## Darbo užduotis

12. Kompiuterinis žaidimas. Kuriate „fantasy“ tipo kompiuterinį žaidimą. Duomenų faile turite informacija apie žaidimo herojus: vardas, rasė, klasė, gyvybės taškai, mana, gynybos taškai, jėga, vikrumas, intelektas, ypatinga galia.

• Suraskite du daugiausia gyvybės taškų turinčius herojus.

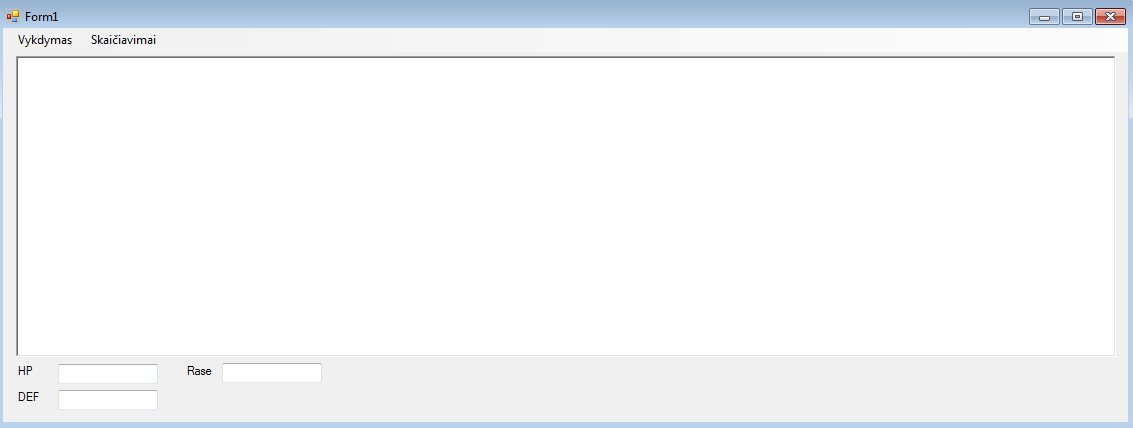
• Sudarykite rasių sąrašą ir kiek kiekviena rasė turi herojų.

• Sudarykite herojų, kurių gyvybės taškai daugiau už x, o gynybos taškai – bent 30, sąrašą (visi duomenys).

• Surikiuokite sudarytą sąrašą pagal pasirinktus du požymius

• Raskite, kuris nurodytos rasės herojus pasižymi geriausiomis charakteristikomis (jėgos, vikrumo ir intelekto suma didžiausia) (visi duomenys). Visus skaičiavimų rezultatus pateikite rezultatų faile lentelėmis.

## Grafinės vartotojo sąsajos schema ir paveikslas



## 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 |
| Vykdyti  TxtToolStripMenuItem | Darbas su failais. | Text:Vykdyti Font: Segoe UI, Regular, 12 Script: Baltic |
| Ivesti  TxtToolStripMenuItem | Darbas su failais. | Text: Ivesti Font: Segoe UI, Regular, 12 Script: Baltic |
| Spausdinti  TxtToolStripMenuItem | Darbas su failais. | Text:Spaustinti Font: Segoe UI, Regular, 12 Script: Baltic |
| Uzdryti  TxtToolStripMenuItem | Uzdaro programa | Text:Uzdaryti Font: Segoe UI, Regular, 12 Script: Baltic |
| Skaiciavimai  TxtToolStripMenuItem | Darbas su duomenim | Text:SkaiciavimaiFont: Segoe UI, Regular, 12 Script: Baltic |
| Daugiausia HP  TxtToolStripMenuItem | Randa su daugiausia HP | Text:Daugiausia HPFont: Segoe UI, Regular, 12 Script: Baltic |
| Atrinkt Rases  TxtToolStripMenuItem | Sudaro rasiu sarasa | Text:Atrinkti RasesFont: Segoe UI, Regular, 12 Script: Baltic |
| Rasti geriausia  TxtToolStripMenuItem | Randa geriausia veikeja | Text: Rasti geriausiaFont: Segoe UI, Regular, 12 Script: Baltic |
| Atrinkti  TxtToolStripMenuItem | Atrenka herojus pagal duom | Text: Atrinkti Font: Segoe UI, Regular, 12 Script: Baltic |
| Rikiuoti  TxtToolStripMenuItem | Surikiuja pagal rases ir vardus | Text:Rikiuoti Font: Segoe UI, Regular, 12 Script: Baltic |
| Label1 | Nusako ką įvesti | Text: HP  Font: Times New Roman, Bold, 12  Script: Baltic |
| Label2 | Nusako ką įvesti | Text: MP  Font: Times New Roman, Bold, 12  Script: Baltic |
| Label3 | Nusako ką įvesti | Text: Rase  Font: Times New Roman, Bold, 12  Script: Baltic |
| textBox1 | Įvedama Hp reikšmė | Font: Courier New, Bold, 12  Script: Baltic |
| textBox2 | Įvedama DEF reikšmė | Font: Courier New, Bold, 12  Script: Baltic |
| textBox3 | Įvedama norima rasė | Font: Courier New, Bold, 12  Script: Baltic |
| Help  TxtToolStripMenuItem | Parodo vartotojo instrukciją | Text:Help Font: Segoe UI, Regular, 12 Script: Baltic |

## Programos vartotojo vadovas

Vykdyti srityje paspaužiame „Įkelti“ mygtuka ir nurodome doumenų failo buvimo vietą, tada paspaudžiame skausdinti ir nurodome kur bus saugomi rezultatai. Atlikus minėtus veiksmus, Skaičiavimai skiltyje veiksmai atliekami mygtukų ičdestymo dvarka(nuo visršaus į apačią). „help“ mygtumas programo lange parodo užduoties sąlygą ir vartojimo instrukciją.

## Programos tekstas

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

abstract public class Herojus:IComparable<Herojus>

{

public string vardas { get; set; }

public string rase { get; set; }

public string klase { get; set; }

public int HP { get; set; }

public int MP { get; set; }

public int def { get; set; }

public Herojus()

{

}

public Herojus(string vardas="", string rase="", string klase="", int HP=0, int MP=0,

int def=0)

{

this.vardas = vardas;

this.rase = rase;

this.klase = klase;

this.HP = HP;

this.MP = MP;

this.def = def;

}

public override string ToString()

{

string eil;

eil = string.Format("|{0,-16}| {1,-16}|{2,-16}|{3,4}|{4,4}|{5,4}|",vardas,rase,klase,HP,MP,def);

return eil;

}

public int CompareTo(Herojus kita)

{

int ras = string.Compare(this.rase, kita.rase, StringComparison.CurrentCulture);

int var = string.Compare(this.vardas, kita.vardas, StringComparison.CurrentCulture);

if ((ras > 0)|| (var > 0))

return 1;

else

return -1;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

class Herojei : Herojus

{

public int strenth { get; set; }

public int agi { get; set; }

public int intel { get; set; }

public string power { get; set; }

public Herojei()

{

}

public Herojei(string vardas = "", string rase = "", string klase = "", int HP = 0, int MP = 0,

int def = 0, int strenth = 0, int agi = 0, int intel = 0, string power = "") : base(vardas, rase, klase, HP, MP, def)

{

this.strenth = strenth;

this.agi = agi;

this.intel = intel;

this.power = power;

}

public override string ToString()

{

string eil;

eil = string.Format("{0} {1,4}|{2,4}|{3,4}|{4,10}|", base.ToString(), strenth,agi, intel,power);

return eil;

}

public double Suma()

{

return strenth + agi + intel;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

class Rases

{

public string rase { get; set; }

public int kiek{get;set;}

public Rases(string rase, int kiek)

{

this.rase = rase;

this.kiek = kiek;

}

public override string ToString()

{

string eil;

eil = string.Format("{0,-16}{1,4}", rase, kiek);

return eil;

}

}

}

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 Lab3

{

public partial class Form1 : Form

{

private List<Herojei> HerojuList;

private List<Rases> rases;

private List<Herojei> atrinkti;

//int pirmas;

//int antras;

string fv;

string info;

public Form1()

{

InitializeComponent();

}

private void uždaryt\_Click(object sender, EventArgs e)

{

Close();

}

private void įvesti\_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)

{

string fr = openFileDialog1.FileName;

HerojuList=Skaityti(fr);

}

}

static List<Herojei> Skaityti(string fv)

{

List<Herojei> HerojuList = new List<Herojei>();

using (StreamReader reader = new StreamReader(fv))

{

string eil;

while((eil=reader.ReadLine())!=null)

{

string[] parts = eil.Split(' ');

string vard = parts[0].Trim();

string rase= parts[1].Trim();

string klase = parts[2].Trim();

int HP = int.Parse(parts[3]);

int MP = int.Parse(parts[4]);

int def = int.Parse(parts[5]);

int st = int.Parse(parts[6]);

int agi = int.Parse(parts[7]);

int intel = int.Parse(parts[8]);

string power = parts[9].Trim();

Herojei hr = new Herojei(vard, rase, klase, HP, MP, def, st, agi, intel, power);

HerojuList.Add(hr);

}

}

return HerojuList;

}

private void spausdinti\_Click(object sender, EventArgs e)

{

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)

{

fv = saveFileDialog1.FileName;

if (File.Exists(fv)) File.Delete(fv);

info = "Pradiniai";

Spausdinti(fv, HerojuList,info);

rezult.LoadFile(fv, RichTextBoxStreamType.PlainText);

}

}

static void Spausdinti(string fv, List<Herojei> HerojuList,string info)

{

using (var fr =new StreamWriter(File.Open(fv,FileMode.Append)))

{

fr.WriteLine(info);

if (HerojuList.Count != 0)

{

fr.WriteLine("|Vardas |rase |Klase |HP |MP |DEF |Jega |AGI |Inte| Power|");

for (int i = 0; i < HerojuList.Count; i++)

{

fr.WriteLine(HerojuList[i]);

}

}

else

fr.WriteLine("nera");

fr.WriteLine(" ");

}

}

/// <summary>

/// Randa daugiausia HP

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void daugiausiaiHP\_Click(object sender, EventArgs e)

{

//int pirmas=-1 , antras=-1;

int daug = 0;

//daug = 0;

int nekreipt = -1;

int pirmas = rasti(HerojuList, daug, nekreipt);

nekreipt = pirmas;

int antras = rasti(HerojuList, daug, nekreipt);

// DaugiausiaiHP(HerojuList, daug,pirmas,antras);

DaugiausiaiHPspaust(fv ,pirmas, antras,HerojuList);

}

static void DaugiausiaiHPspaust(string fv, int pirmas, int antras, List<Herojei> HerojuList)

{

using (var fr = new StreamWriter(File.Open(fv, FileMode.Append)))

{

fr.WriteLine("Daugiausiai HP");

if (HerojuList.Count != 0)

{

fr.WriteLine(" |Vardas |rase |Klase |HP |MP |DEF |Jega |AGI |Inte| Power|");

fr.WriteLine("1 {0}", HerojuList[pirmas]);

fr.WriteLine("2 {0}", HerojuList[antras]);

}

else

fr.WriteLine("Nera");

fr.WriteLine(" ");

}

}

//static void DaugiausiaiHP(List<Herojei> HerojuList,int daug,int pirmas,int antras)

//{

// daug = 0;

// int nekreipt = -1;

// pirmas = rasti(HerojuList, daug, nekreipt);

// nekreipt = pirmas;

// antras= rasti(HerojuList, daug, nekreipt);

//}

static int rasti(List<Herojei> HerojuList, int daug, int nekreipt)

{

int kelintas=0;

daug = 0;

for(int i=0;i<HerojuList.Count;i++)

{

if(HerojuList[i].HP>=daug)

{

if (i != nekreipt)

{

daug = HerojuList[i].HP;

kelintas = i;

}

}

}

return kelintas;

}

//#######################################################################################

/// <summary>

/// sudaro rasiu sarasa

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void atrinktRases\_Click(object sender, EventArgs e)

{

rases = Atrinkt(HerojuList);

//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)

//{

// string fv = saveFileDialog1.FileName;

//if (File.Exists(fv)) File.Delete(fv);

Spausdint(fv,rases);

rezult.LoadFile(fv, RichTextBoxStreamType.PlainText);

//}

}

static List<Rases> Atrinkt(List<Herojei> HerojuList)

{

int keli = 0;

List<Rases> rases = new List<Rases>();

for(int i=0;i<HerojuList.Count;i++)

{

keli = 0;

for(int j=0;j<rases.Count;j++)

{

if(HerojuList[i].rase==rases[j].rase)

{

keli++;

}

}

if(keli==0)

{

string rase = HerojuList[i].rase;

int kiek = kiekRS(rase, HerojuList);

Rases rs = new Rases(rase, kiek);

rases.Add(rs);

}

}

return rases;

}

static int kiekRS(string rase, List<Herojei> HerojuList)

{

int keli=0;

for (int i=0;i<HerojuList.Count;i++)

{

if (HerojuList[i].rase == rase)

keli++;

}

return keli;

}

static void Spausdint(string fv,List<Rases>rases)

{

using (var fr = new StreamWriter(File.Open(fv, FileMode.Append)))

{

fr.WriteLine("Rases");

if (rases.Count != 0)

{

fr.WriteLine("rase Skaicius");

for (int i = 0; i < rases.Count; i++)

{

fr.WriteLine(rases[i]);

}

}

else

fr.WriteLine("nera");

fr.WriteLine(" ");

}

}

//#######################################################################################

/// <summary>

/// atrenka herojus pagal kriterijus

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void atrinktiHerojusPagalDuom\_Click(object sender, EventArgs e)

{

int HP = int.Parse(textBox1.Text);

int DEF = int.Parse(textBox2.Text);

atrinkti = atrinktiDuom(HerojuList, HP, DEF);

info = "Atrinkta";

Spausdinti(fv, atrinkti,info);

rezult.LoadFile(fv, RichTextBoxStreamType.PlainText);

}

static List<Herojei> atrinktiDuom(List<Herojei> HerojuList,int HP,int DEF)

{

List<Herojei> Duom = new List<Herojei>();

for(int i=0;i<HerojuList.Count;i++)

{

if (HerojuList[i].HP > HP && HerojuList[i].def >= DEF)

Duom.Add(HerojuList[i]);

}

return Duom;

}

private void rikiuot\_Click(object sender, EventArgs e)

{

atrinkti.Sort();

info = "Surikiuota";

Spausdinti(fv, atrinkti,info);

rezult.LoadFile(fv, RichTextBoxStreamType.PlainText);

}

//#######################################################################################

/// <summary>

/// randa geriausia

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void rastiGeriausia\_Click(object sender, EventArgs e)

{

string rase = textBox3.Text;

int indeksas = suma(HerojuList,rase);

geriau(fv,HerojuList, indeksas);

rezult.LoadFile(fv, RichTextBoxStreamType.PlainText);

}

static int suma(List<Herojei>HerojuList,string rase)

{

double max = 0;

double sk=0;

int indeksas = -1;

for (int i = 0; i < HerojuList.Count; i++)

if (HerojuList[i].rase == rase)

{

sk = HerojuList[i].Suma();

if (max < sk)

{

max = sk;

indeksas = i;

}

}

return indeksas;

}

static void geriau(string fv,List<Herojei>HerojuList, int indeksas )

{

using (var fr = new StreamWriter(File.Open(fv, FileMode.Append)))

{

fr.WriteLine("Geriausias nurodytos rases herojus (pirmas jei yra daugiau vienoda statistika turinciu) ");

if (indeksas != -1)

fr.WriteLine(HerojuList[indeksas]);

else

fr.WriteLine(" Nera ");

fr.WriteLine(" ");

}

}

private void helpToolStripMenuItem\_Click(object sender, EventArgs e)

{

rezult.LoadFile(help, RichTextBoxStreamType.PlainText);

}

//#######################################################################################

}

}

## Pradiniai duomenys ir rezultatai

//duomenų failas.

keksas zmogus fighter 150 100 200 100 200 100 fire

keksas zmogus fighter 250 100 200 100 200 100 fire

keksas elfas lankas 150 200 50 100 200 100 nature

keksas zmogus fighter 250 100 200 100 200 100 fire

aeksas zmogus fighter 150 100 200 100 200 100 fire

keksas zmogus fighter 350 100 200 100 200 100 fire

aeksas elfas lankas 150 200 50 100 200 100 nature

a orkas berserker 200 200 200 50 50 50 earth

//įvedami duomenys

HP 100 DEF 30 Rase zmogus

//help duomenys

Užduotis:

12. Kompiuterinis žaidimas. Kuriate „fantasy“ tipo kompiuterinį žaidimą. Duomenų faile turite informacija apie žaidimo herojus: vardas, rasė, klasė, gyvybės taškai, mana, gynybos taškai, jėga, vikrumas, intelektas, ypatinga galia.

Suraskite du daugiausia gyvybės taškų turinčius herojus.

Sudarykite rasių sąrašą ir kiek kiekviena rasė turi herojų.

Sudarykite herojų, kurių gyvybės taškai daugiau už x, o gynybos taškai – bent 30, sąrašą (visi duomenys).

Surikiuokite sudarytą sąrašą pagal pasirinktus du požymius

Raskite, kuris nurodytos rasės herojus pasižymi geriausiomis charakteristikomis (jėgos, vikrumo ir intelekto suma didžiausia) (visi duomenys). Visus skaičiavimų rezultatus pateikite rezultatų faile lentelėmis.

Instrukcija:

Vykdyti srityje paspaužiame „Įkelti“ mygtuka ir nurodome doumenų failo buvimo vietą, tada paspaudžiame skausdinti ir nurodome kur bus saugomi rezultatai. Atlikus minėtus veiksmus, Skaičiavimai skiltyje veiksmai atliekami mygtukų ičdestymo dvarka(nuo visršaus į apačią). „help“ mygtumas programo lange parodo užduoties sąlygą ir vartojimo instrukciją.

Pradiniai

|Vardas |rase |Klase |HP |MP |DEF |Jega |AGI |Inte| Power|

|keksas | zmogus |fighter | 150| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 250| 100| 200| 100| 200| 100| fire|

|keksas | elfas |lankas | 150| 200| 50| 100| 200| 100| nature|

|keksas | zmogus |fighter | 250| 100| 200| 100| 200| 100| fire|

|aeksas | zmogus |fighter | 150| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 350| 100| 200| 100| 200| 100| fire|

|aeksas | elfas |lankas | 150| 200| 50| 100| 200| 100| nature|

|a | orkas |berserker | 200| 200| 200| 50| 50| 50| earth|

Daugiausiai HP

|Vardas |rase |Klase |HP |MP |DEF |Jega |AGI |Inte| Power|

1 |keksas | zmogus |fighter | 350| 100| 200| 100| 200| 100| fire|

2 |keksas | zmogus |fighter | 250| 100| 200| 100| 200| 100| fire|

Rases

rase Skaicius

zmogus 5

elfas 2

orkas 1

Geriausias nurodytos rasės herojus (pirmas jei yra daugiau vienoda statistika turinciu)

|keksas | zmogus |fighter | 150| 100| 200| 100| 200| 100| fire|

Atrinkta

|Vardas |rase |Klase |HP |MP |DEF |Jega |AGI |Inte| Power|

|keksas | zmogus |fighter | 150| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 250| 100| 200| 100| 200| 100| fire|

|keksas | elfas |lankas | 150| 200| 50| 100| 200| 100| nature|

|keksas | zmogus |fighter | 250| 100| 200| 100| 200| 100| fire|

|aeksas | zmogus |fighter | 150| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 350| 100| 200| 100| 200| 100| fire|

|aeksas | elfas |lankas | 150| 200| 50| 100| 200| 100| nature|

|a | orkas |berserker | 200| 200| 200| 50| 50| 50| earth|

Surikiuota

|Vardas |rase |Klase |HP |MP |DEF |Jega |AGI |Inte| Power|

|aeksas | elfas |lankas | 150| 200| 50| 100| 200| 100| nature|

|keksas | elfas |lankas | 150| 200| 50| 100| 200| 100| nature|

|a | orkas |berserker | 200| 200| 200| 50| 50| 50| earth|

|aeksas | zmogus |fighter | 150| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 350| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 250| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 250| 100| 200| 100| 200| 100| fire|

|keksas | zmogus |fighter | 150| 100| 200| 100| 200| 100| fire|

# Susietasis sąrašas (L4)

## 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.

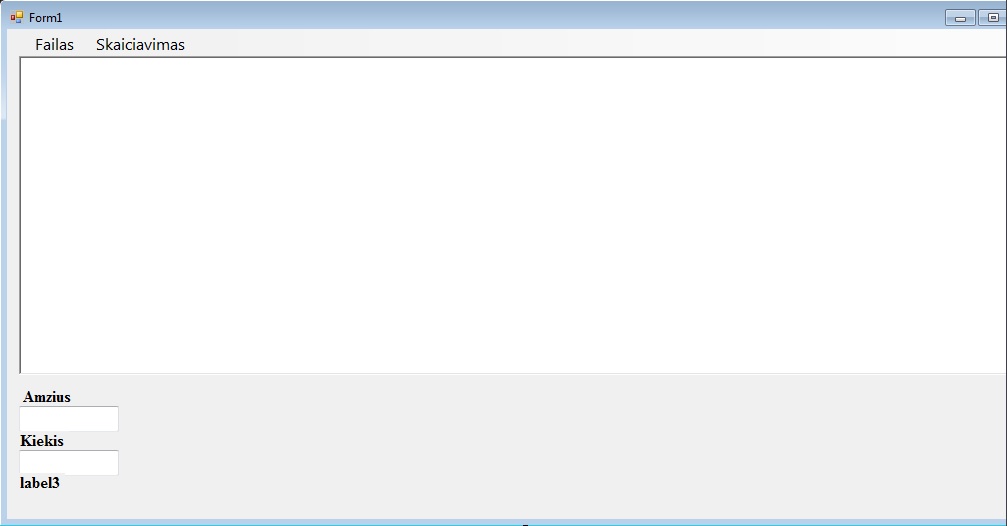
• Raskite kurioje miesto kandidatų sąraše yra daugiau žaidėjų su aukštuoju išsilavinimu.

• 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

## Grafinės vartotojo sąsajos schema ir paveikslas



## 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 |
| Darbas su failaisToolStripMenuItem | Darbas su failais. | Text: Failas Font: Segoe UI, Regular, 12 Script: Baltic |
| IkeltiToolStripMenuItem | Darbas su failais. | Text:Atidaryti Font: Segoe UI, Regular, 12 Script: Baltic |
| BaigtitoolStripMenuItem3 | Uzdaro programą. | Text: Baigti Font: Segoe UI, Regular, 12 Script: Baltic |
| Darbas su duomToolStripMenuItem | Darbas su duomenimis. | Text: Skaiciavimas Font: Segoe UI, Regular, 12 Script: Baltic |
| DaugiausiatoolStripMenuItem4 | Randa daugiausiai. | Font: Segoe UI, Regular, 12 Script: Baltic |
| atrinktiTxtToolStripMenuItem | atrenka | Font: Segoe UI, Regular, 12 Script: Baltic |
| rikiuotTxtToolStripMenuItem | rikiuoja | Font: Segoe UI, Regular, 12 Script: Baltic |
| salintTxtToolStripMenuItem | salina | 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 |

## Programos vartotojo vadovas

Paspaugus „Failas“ galima pasiringti ar norima ivesti duomenis ar uždaryti programą, jei norima ivesti duomenis, reikia apatiniame kaireme programos kampe įvesti reikiamus skaičius. Įvedus duomenis, galima vykdyti „skaičiavimai“(reikia vykdyti etapus, mygtuku eiliskumo tvarka, kaskart paspaudus mygtuką, programos lange pasirodys atsakymas).

## Programos tekstas

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab4

{

public 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 override int GetHashCode()

{

return base.GetHashCode();

}

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 );

}

public static bool operator <(Zaidejas zaid1, Zaidejas zaid2)

{

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

return (p > 0);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab4

{

public sealed class Mazgas

{

public Zaidejas Duomenys { get; set; }

public Mazgas Kitas { get; set; }

public Mazgas()

{ }

public Mazgas(Zaidejas Duomenys,Mazgas Kitas)

{

this.Duomenys = Duomenys;

this.Kitas = Kitas;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab4

{

public sealed class Komandos

{

Mazgas pr;

Mazgas pb;

Mazgas d;

public Komandos()

{

this.pr = null;

this.pb = null;

this.d = null;

}

public void Pradzia()

{

d = pr;

}

public void Kitas()

{

d = d.Kitas;

}

public bool Yra()

{

return d != null;

}

public Zaidejas imti()

{

return d.Duomenys;

}

public void DetiA(Zaidejas inf)

{

var d = new Mazgas(inf, null);

d.Kitas = pr;

pr = d;

}

public void DetiB(Zaidejas inf)

{

var d = new Mazgas(inf, null);

if (pr != null)

{

pb.Kitas = d;

pb = d;

}

else

{

pr = d;

pb = d;

}

}

public void Rikiuoti()

{

bool keista = true;

Mazgas d1, d2;

while (keista)

{

keista = false;

d1 = d2 = pr;

while (d2 != null)

{

if (d2.Duomenys > d1.Duomenys)

{

Zaidejas z = d1.Duomenys;

d1.Duomenys = d2.Duomenys;

d2.Duomenys = z;

keista = true;

}

d1 = d2; d2 = d2.Kitas;

}

}

}

//public void Salinti(Zaidejas tas)

//{

// Mazgas dd = null;

// for(Mazgas d= pr; d== null; d=d.Kitas)

// {

// Zaidejas pag = d.Duomenys;

// if(pag==tas)

// {

// if(dd==null)

// {

// Mazgas a, b;

// a = d;

// b = d.Kitas;

// a.Duomenys = b.Duomenys;

// a.Kitas = b.Kitas;

// b.Kitas = null;

// b = null;

// }

// else

// {

// dd.Kitas = d.Kitas;

// }

// }

// dd = d;

// }

//}

public void Naikint(Zaidejas kel)

{

if(pr!=null)

{

Mazgas dd = pr;

if (pr.Duomenys == kel)

pr = pr.Kitas;

else

{

while(dd.Kitas!=null)

{

if(dd.Kitas.Duomenys==kel)

{

dd.Kitas = dd.Kitas.Kitas;

return;

}

dd = dd.Kitas;

}

}

}

}

}

}

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 Lab4

{

public partial class Form1 : Form

{

const string Kaun = "..\\..\\Kaunas.txt";

const string Kl = "..\\..\\Klaipeda.txt";

const string rez = "..\\..\\Rezultatai.txt";

const string pag = "..\\..\\Help.txt";

Komandos Klaipeda;

Komandos Kaunas;

Komandos kauno;

Komandos klaipedos;

int m, k;

int index = 0;

string kom1;//Klaipedos

string kom2;//Kauno

string daug="Lygu";

public Form1()

{

InitializeComponent();

if (File.Exists(rez))

File.Delete(rez);

}

private void ikeltiToolStripMenuItem\_Click(object sender, EventArgs e)

{

//string kom1;//Klaipedos

//string kom2;//Kauno

//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,index);

Kaunas = SkaitytiKom(Kaun, out kom2, index);

// Klaipeda.Pradzia();

// Kaunas.Pradzia();

Spausdinti(ant1, Kaunas,rez);

Spausdinti(ant2, Klaipeda,rez);

richTextBox1.LoadFile(rez, RichTextBoxStreamType.PlainText);

index++;

}

static Komandos SkaitytiKom(string fv, out string kom,int index)

{

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);

if(index==0)

Komand.DetiB(zaid);

if(index==1)

Komand.DetiA(zaid);

}

}

return Komand;

}

static string Daugiausia(Komandos Klaipeda,Komandos Kaunas)

{

int KL\_daug = rasti(Klaipeda);

int KA\_daug = rasti(Kaunas);

if (KL\_daug > KA\_daug)

return "Klaipeda";

if (KL\_daug < KA\_daug)

return "Kaunas";

else

return "Lygu";

}

private void daugiausiaSuAukstToolStripMenuItem\_Click(object sender, EventArgs e)

{

//daug = "Lygu";

daug = Daugiausia( Klaipeda, Kaunas);

using (var fv =File.AppendText(rez))

{

fv.WriteLine("Daugiausia su aukstuoju yra: {0}",daug);

}

richTextBox1.LoadFile(rez, RichTextBoxStreamType.PlainText);

}

static int rasti(Komandos kom)

{

int daug = 0;

for (kom.Pradzia(); kom.Yra(); kom.Kitas())

{

if (kom.imti().lav == "aukst")

daug++;

}

return daug;

}

private void atrinktToolStripMenuItem\_Click(object sender, EventArgs e)

{

kauno = atrinkt(Kaunas,m,k);

klaipedos = atrinkt(Klaipeda,m,k);

string ant1 = "Atrinkta Kauno komanda";

string ant2 = "Atrinkta Klaipedos komanda";

Spausdinti(ant1, kauno, rez);

klaipedos.Pradzia();

// klaipedos.Salinti();

Spausdinti(ant2, klaipedos, rez);

richTextBox1.LoadFile(rez, RichTextBoxStreamType.PlainText);

}

static Komandos atrinkt (Komandos kom,int m,int k)

{

Komandos komand = new Komandos();

for(kom.Pradzia();kom.Yra();kom.Kitas())

if(kom.imti().age<=m&&kom.imti().kl>k)

{

komand.DetiB(kom.imti());

}

return komand;

}

private void rikiuotToolStripMenuItem\_Click(object sender, EventArgs e)

{

kauno.Pradzia();

klaipedos.Pradzia();

if(kauno.Yra())

kauno.Rikiuoti();

if(klaipedos.Yra())

klaipedos.Rikiuoti();

string ant1 = "surikiuota Kauno komanda";

string ant2 = "surikiuota Klaipedos komanda";

Spausdinti(ant1, kauno, rez);

//klaipedos.Pradzia();

//klaipedos.Salinti();

Spausdinti(ant2, klaipedos, rez);

richTextBox1.LoadFile(rez, RichTextBoxStreamType.PlainText);

}

private void salintiToolStripMenuItem\_Click(object sender, EventArgs e)

{

//kauno.Salinti();

// klaipedos.Salinti();

salinti(kauno);

salinti(klaipedos);

string ant1 = "pasalinta Kauno komanda";

string ant2 = "Pasalinta Klaipedos komanda";

Spausdinti(ant1, kauno, rez);

//klaipedos.Pradzia();

//klaipedos.Salinti();

Spausdinti(ant2, klaipedos, rez);

richTextBox1.LoadFile(rez, RichTextBoxStreamType.PlainText);

}

static void Spausdinti(string antr,Komandos kom,string rez)

{

using (var fv = File.AppendText(rez))

{

fv.WriteLine(antr);

kom.Pradzia();

if (kom.Yra())

{

fv.WriteLine("vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai");

fv.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for (kom.Pradzia(); kom.Yra(); kom.Kitas())

fv.WriteLine(kom.imti().ToString());

}

else

fv.WriteLine("Nera");

fv.WriteLine(" ");

}

}

private void baidtiToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

private void helpToolStripMenuItem\_Click(object sender, EventArgs e)

{

richTextBox1.LoadFile(pag, RichTextBoxStreamType.PlainText);

}

static void salinti(Komandos A)

{

for (A.Pradzia(); A.Yra(); A.Kitas())

{

Zaidejas zaid = A.imti();

if (zaid.lav != "aukst")

{

A.Naikint(zaid);

salinti(A);

break;

}

}

}

//static void SkaitytiAtv(string fv,out Mazgas pr)

//{

// using (var failas = new StreamReader(fv))

// {

// pr = null;

// 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.DetiB(zaid);

// }

// }

// }

//}

}

}

## Pradiniai duomenys ir rezultatai

Kaunas

antis tortas;v;18;aukst;5;3;

vardenis pav;v;28;aukst;5;3;

keksas Tortas;v;38;aukst;5;3;

Klaipeda

kestas kestutis;v;5;neaukst;5;8;

tomas tomutis;v;58;neaukst;5;8;

antanas antis;v;38;neaukst;5;4;

m=60 k=2

Kaunas pradiniai duomenys

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

antis tortas v 18 aukst 5 3

vardenis pav v 28 aukst 5 3

keksas Tortas v 38 aukst 5 3

Klaipeda pradiniai duomenys

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

kestas kestutis v 5 neaukst 5 8

tomas tomutis v 58 neaukst 5 8

antanas antis v 38 neaukst 5 4

Daugiausia su aukstuoju yra: Kaunas

Atrinkta Kauno komanda

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

antis tortas v 18 aukst 5 3

vardenis pav v 28 aukst 5 3

keksas Tortas v 38 aukst 5 3

Atrinkta Klaipedos komanda

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

kestas kestutis v 5 neaukst 5 8

tomas tomutis v 58 neaukst 5 8

antanas antis v 38 neaukst 5 4

surikiuota Kauno komanda

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

antis tortas v 18 aukst 5 3

keksas Tortas v 38 aukst 5 3

vardenis pav v 28 aukst 5 3

surikiuota Klaipedos komanda

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

antanas antis v 38 neaukst 5 4

kestas kestutis v 5 neaukst 5 8

tomas tomutis v 58 neaukst 5 8

pasalinta Kauno komanda

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

antis tortas v 18 aukst 5 3

keksas Tortas v 38 aukst 5 3

vardenis pav v 28 aukst 5 3

pasalinta Klaipedos komanda

vardas pavarde Lytis amzius issilavinimas zaidimu skaicius Klausymai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

nera

# Bendrinės klasės (L5)

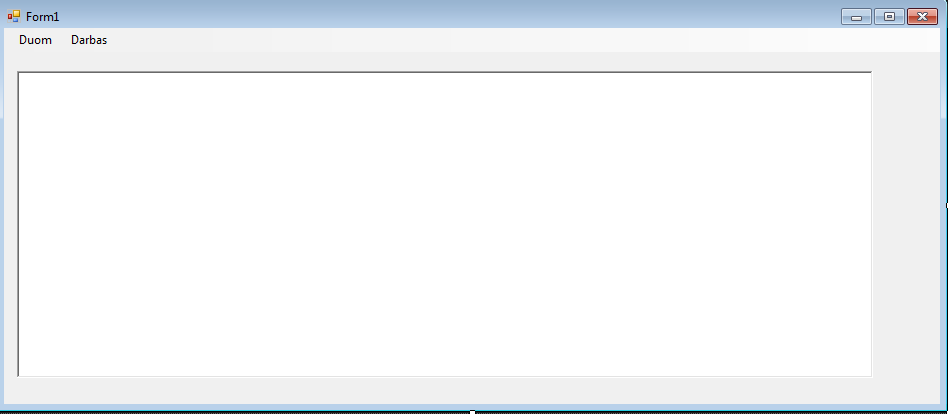
## Darbo užduotis

L5\_12. Studentai. Studentai renkasi modulius. Studentas gali rinktis keletą modulių. Už modulius yra atsakingi dėstytojai. Suformuokite pasirinktų modulių sąrašą. Sudarykite atskirą modulių, kurių studentai nepasirinko, sąrašą. Suraskite, kuris studentas pasirinko daugiausiai modulių. Duomenys:

• Tekstiniame faile U12a.txt yra duomenys apie studentų pasirenkamus modulius: modulio pavadinimas, studento pavardė, vardas, grupė.

• Tekstiniame faile U12b.txt yra duomenys apie modulius: modulio pavadinimas, atsakingo dėstytojo pavardė, vardas, kreditų kiekis.

## Grafinės vartotojo sąsajos schema ir paveikslas



## Sąsajoje panaudotų komponentų keičiamos savybės

|  |  |  |
| --- | --- | --- |
| Komponentas | Savybė | Reikšmė |
| Form1 |  | Text:Form1 |
| richTextBox1 (tipas RichTextBox) |  | Font: Courier New, Bold, 12 Script: Baltic |
| duomToolStripMenuItem | Darbas su failais | Text: Duom Font: Segoe UI, Regular, 12 Script: Baltic |
| IvestiToolStripMenuItem | Ivedami duomenys | Text: Ivesti Font: Segoe UI, Regular, 12 Script: Baltic |
| BaigtiToolStripMenuItem | Uždaroma programa | Text: Baigti Font: Segoe UI, Regular, 12 Script: Baltic |
| SpaustToolStripMenuItem | Spausdinami pradiniai duomenys | Text:Spaust Font: Segoe UI, Regular, 12 Script: Baltic |
| DarbasToolStripMenuItem | Darbas su duomenim | Text: Darbas Font: Segoe UI, Regular, 12 Script: Baltic |
| Atrinkti pasirinktusToolStripMenuItem | Atrenka pasirinktus modulius | Text:Atrinkti pasirinktus Font: Segoe UI, Regular, 12 Script: Baltic |
| Atrinkti nepasirintusToolStripMenuItem | Atrenka nepasirinktus modulius | Text: Atrinkti nepasirinktus Baigti Font: Segoe UI, Regular, 12 Script: Baltic |
| studentasSuDaugiausiaiModuliuToolStripMenuItem | Randa studenta su daugiausiai pasirinktu moduliu | Text: Studentas su daugiausiai Baigti Font: Segoe UI, Regular, 12 Script: Baltic |

## Programos vartotojo vadovas

Ivedant duomenis „Duom“=>“Ivesti“, spausdinant pradinius duomenis „Duom“=>“Spaust“, „Duom“=>“Baigti“ uždaro programą.

“Darbas”=>”Atrinkti pasirinkrus” atrenka duos modulius iš U12b.txt kurie yra pasirinkti U12a.txt failuose. „Darbas“=>”Atrinkti nepasirinktus” galima vykdyti tik atlikus prieš tai buvusį žingsnį. “Darbas” =>”Studentas su daugiausiai moduliu” Randa studenta pasirinkusį daugiausiai modulių.

Kiekvienas žingsnis atnaujina rezultatus programos lange.

## Programos tekstas

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

public class Studentas

{

public Studentas(string mod, string vard,string pav,string grupe)

{

this.mod = mod;

this.vard = vard;

this.pav = pav;

this.grupe = grupe;

}

public string mod { get; set; }

public string vard { get; set; }

public string pav { get; set; }

public string grupe { get; set; }

public override int GetHashCode()

{

return base.GetHashCode();

}

public override string ToString()

{

string eil;

eil = string.Format("|{0,-20} | {1,-7} | {2,-20} | {3,6}|",mod,vard,pav,grupe);

return eil;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

public class Modulis

{

public Modulis(string pavad,string vardpav,int krd)

{

this.pavad = pavad;

this.vardpav = vardpav;

this.krd = krd;

}

public string pavad { get; set; }

public string vardpav { get; set; }

public int krd { get; set; }

public override int GetHashCode()

{

return base.GetHashCode();

}

public override string ToString()

{

string eil;

eil = string.Format("|{0,-20} | {1,-7} | {2,-8} |",pavad, vardpav, krd);

return eil;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

public sealed class Mazgas<Tipas>

// where Tipas :IComparable<Tipas>

{

public Tipas Duomenys { get; set; }

public Mazgas<Tipas> Kitas { get; set; }

public Mazgas(Tipas duomenys,Mazgas<Tipas> adresas)

{

this.Duomenys = duomenys;

this.Kitas = adresas;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

public sealed class Sarasas<Tipas>

{

Mazgas<Tipas> pr;

Mazgas<Tipas> d;

Mazgas<Tipas> pb;

public Sarasas()

{

this.pr = null;

this.d = null;

this.pb = null;

}

public void Pradzia()

{

d = pr;

}

public void Kitas()

{

d = d.Kitas;

}

public bool Yra()

{

return d != null;

}

public Tipas imtiDuom() { return d.Duomenys; }

public void DetiDuom(Tipas inf)

{

var d = new Mazgas<Tipas>(inf, null);

if(pr!=null)

{

pb.Kitas = d;

pb = d;

}

else

{

pr = d;

pb = d;

}

}

}

}

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 Lab5

{

public partial class Form1 : Form

{

const string Stud = "...\\...\\U12a.txt";

const string Mod = "...\\...\\U12b.txt";

const string Rez = "...\\...\\Rez.txt";

Sarasas<Studentas> A,stud;

Sarasas<Modulis> M, parinkt, nepasi;

private void ivestiToolStripMenuItem\_Click(object sender, EventArgs e)

{

A = Skaitytistud(Stud);

M = Skaitytimod(Mod);

}

private void baigtiToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

private void spaustToolStripMenuItem\_Click(object sender, EventArgs e)

{

using (StreamWriter failas = File.AppendText(Rez))

{

failas.WriteLine("Pradiniai");

A.Pradzia();M.Pradzia();

failas.WriteLine("|Modulis |Vardas |Pavarde |Grupe ");

if(A.Yra()==false)

{

failas.WriteLine("nera");

}

else

for (A.Pradzia(); A.Yra(); A.Kitas())

{

failas.WriteLine(A.imtiDuom());

}

failas.WriteLine(" ");

failas.WriteLine("|Pavadinimas |Vardas |Kreditai |");

if (M.Yra()==false)

{

failas.WriteLine("nera");

}

else

for (M.Pradzia(); M.Yra(); M.Kitas())

{

failas.WriteLine(M.imtiDuom());

}

}

rezult.LoadFile(Rez, RichTextBoxStreamType.PlainText);

}

public Form1()

{

InitializeComponent();

if (File.Exists(Rez))

File.Delete(Rez);

}

private void atrinktPasirinktusToolStripMenuItem\_Click(object sender, EventArgs e)

{

parinkt = Atrinkt(A, M);

parinkt.Pradzia();

using (StreamWriter failas =File.AppendText(Rez))

{

failas.WriteLine("Pasirinkti");

if (parinkt.Yra() == false)

{

failas.WriteLine("Nera");

}

else

{

failas.WriteLine("|Pavadinimas |Vardas |Kreditai |");

for (parinkt.Pradzia(); parinkt.Yra(); parinkt.Kitas())

{

failas.WriteLine(parinkt.imtiDuom());

}

}

}

rezult.LoadFile(Rez, RichTextBoxStreamType.PlainText);

}

static Sarasas<Modulis> Atrinkt(Sarasas<Studentas> A,Sarasas<Modulis>M)

{

int keli = 0;

int pas = 0;

var atr = new Sarasas<Modulis>();

for(M.Pradzia(); M.Yra(); M.Kitas())

{

keli = 0;

pas = 0;

for(A.Pradzia();A.Yra();A.Kitas())

{

if (M.imtiDuom().pavad == A.imtiDuom().mod)

{

pas++;

for (atr.Pradzia(); atr.Yra(); atr.Kitas())

{

if (M.imtiDuom().pavad == atr.imtiDuom().pavad)

keli++;

}

}

}

if(keli==0&&pas!=0)

{

Modulis mod = new Modulis(M.imtiDuom().pavad,M.imtiDuom().vardpav,M.imtiDuom().krd);

atr.DetiDuom(mod);

}

}

return atr;

}

private void atrinktNepasirinktusToolStripMenuItem\_Click(object sender, EventArgs e)

{

nepasi = Atrinktnepasi(M, parinkt);

parinkt.Pradzia();

using (StreamWriter failas = File.AppendText(Rez))

{

failas.WriteLine("nepasirinkti");

if (nepasi.Yra() == false)

{

failas.WriteLine(" Nera ");

}

else

{

failas.WriteLine("|Pavadinimas |Vardas |Kreditai |");

for (nepasi.Pradzia(); nepasi.Yra(); nepasi.Kitas())

{

failas.WriteLine(nepasi.imtiDuom());

}

}

}

rezult.LoadFile(Rez, RichTextBoxStreamType.PlainText);

}

static Sarasas<Modulis> Atrinktnepasi(Sarasas<Modulis> M, Sarasas<Modulis> parinkt)

{

int keli = 0;

var nepasirinkti = new Sarasas<Modulis>();

for(M.Pradzia();M.Yra();M.Kitas())

{

keli = 0;

for(parinkt.Pradzia(); parinkt.Yra(); parinkt.Kitas())

{

if(M.imtiDuom().pavad==parinkt.imtiDuom().pavad)

{

keli++;

}

}

if(keli==0)

{

Modulis mod = new Modulis(M.imtiDuom().pavad, M.imtiDuom().vardpav, M.imtiDuom().krd);

nepasirinkti.DetiDuom(mod);

}

}

return nepasirinkti;

}

private void studentasSuDaugiausiaiModuliuToolStripMenuItem\_Click(object sender, EventArgs e)

{

stud= Rasti(A);

stud.Pradzia();

// int kiek = 0;

string st="nera";

if(stud.Yra()==false)

{

using (var failas = File.AppendText(Rez))

{

failas.WriteLine("Daugiausiai moduliu turi");

failas.WriteLine("Nera");

}

}

else

{

//for (stud.Pradzia(); stud.Yra(); stud.Kitas())

//{

//string vardas = stud.imtiDuom().vard;

//string pav = stud.imtiDuom().pav;

st = student(stud, A);

//if (keli > kiek)

//{

// kiek = keli;

// Studentas studentas = new Studentas(A.imtiDuom().mod, A.imtiDuom().vard, A.imtiDuom().pav, A.imtiDuom().grupe);

// st = studentas.ToString();

//}

//}

using (var failas = File.AppendText(Rez))

{

failas.WriteLine("Daugiausiai moduliu turi");

failas.WriteLine(st);

}

}

rezult.LoadFile(Rez, RichTextBoxStreamType.PlainText);

}

static Sarasas<Studentas> Rasti(Sarasas<Studentas> A)

{

var stu = new Sarasas<Studentas>();

A.Pradzia();

for (A.Pradzia();A.Yra();A.Kitas())

{

int keli = 0;

for(stu.Pradzia();stu.Yra();stu.Kitas())

{

if(A.imtiDuom().vard==stu.imtiDuom().vard&&A.imtiDuom().pav==stu.imtiDuom().pav&&A.imtiDuom().grupe==stu.imtiDuom().grupe)

{

keli++;

}

}

if(keli==0)

{

var studentas = new Studentas(A.imtiDuom().mod, A.imtiDuom().vard, A.imtiDuom().pav, A.imtiDuom().grupe);

stu.DetiDuom(studentas);

}

}

return stu;

}

static string student(Sarasas<Studentas> stud,Sarasas<Studentas> A)

{

int keli=0;

int kiek = 0;

string st="Nera";

for(stud.Pradzia();stud.Yra();stud.Kitas())

{

keli=0;

for(A.Pradzia();A.Yra();A.Kitas())

{

if(stud.imtiDuom().pav==A.imtiDuom().pav)

keli++;

}

if(kiek<keli)

{

st = stud.imtiDuom().ToString();

kiek++;

}

}

return st ;

}

static Sarasas<Studentas> Skaitytistud(string fv)

{

var A = new Sarasas<Studentas>();

using (var failas = new StreamReader(fv))

{

string eilute;

while((eilute=failas.ReadLine())!=null)

{

string[] part = eilute.Split(';');

string mod = part[0];

string vard = part[1];

string pav = part[2];

string grupe = part[3];

Studentas S = new Studentas(mod, vard, pav, grupe);

A.DetiDuom(S);

}

}

return A;

}

static Sarasas<Modulis> Skaitytimod(string fv)

{

var M = new Sarasas<Modulis>();

using (var failas = new StreamReader(fv))

{

string eilute;

while((eilute=failas.ReadLine())!=null)

{

string[] part = eilute.Split(';');

string pavad = part[0];

string vardpav = part[1];

int krd = int.Parse(part[2]);

Modulis m = new Modulis(pavad, vardpav, krd);

M.DetiDuom(m);

}

}

return M;

}

}

}

## Pradiniai duomenys ir rezultatai

U12a.txt

prog;karolis;staniukynas;ifin5/3

mat;karolis;staniukynas;ifin5/3

mat;karolis;staniukynas;ifin5/3

mat;tomas;tomas;if

U12b.txt

prog;kestas;6

mat;bulve;6

info;keksas;6

Rez.txt

Pradiniai

|Modulis |Vardas |Pavarde |Grupe

|prog | karolis | staniukynas | ifin5/3|

|mat | karolis | staniukynas | ifin5/3|

|mat | karolis | staniukynas | ifin5/3|

|mat | tomas | tomas | if|

|Pavadinimas |Vardas |Kreditai |

|prog | kestas | 6 |

|mat | bulve | 6 |

|info | keksas | 6 |

Pasirinkti

|Pavadinimas |Vardas |Kreditai |

|prog | kestas | 6 |

|mat | bulve | 6 |

nepasirinkti

Nera

Daugiausiai moduliu turi

|prog | karolis | staniukynas | ifin5/3|