KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

OBJEKTINIS PROGRAMAVIMAS II (P175B123)

Laboratorinio darbo ataskaita

Atliko:
IFF-5/7 gr. Studentė
Viktorija Ražaitė
2016 m. vasario 24 d.
Priėmė:
Jurgis Pralgauskis

KAUNAS 2016

TURINYS

1. Rekursija

1.1. Darbo užduotis

LD_7. Žodžių paieška.

Parašykite programą, kuri perskaitytų iš tekstinio failo "Trecias.txt" tekstą po vieną simbolį (1 < simbolių kiekis <= 2000) ir jais užpildytų masyvą A[n,n]. Į masyvą nerašomi eilutės pabaigos, naujos eilutės ir failo pabaigos simboliai. n parenkamas toks mažiausias, kad tekstas tilptų į kvadratinę matricą. Jei paskutinei eilutei trūksta simbolių, užpildote tarpais. Po to programa iš tekstinio failo "Zodziai.txt", kuriame kiekvienas žodis yra atskiroje eilutėje ir prasideda nuo 1 pozicijos, perskaito eilinį žodį, kurio ilgis k <= n/2. Reikia nustatyti, kiek kartų šis žodis kartojasi lentelėje A[n, n]. Paieška turėtų būti atliekama horizontaliai iš kairės į dešinę, vertikaliai iš viršaus žemyn ir pagal dešinę diagonalę žemyn ir į dešinę (ne tik pagrindinė diagonalė). Žodžiai iš eilutės (stulpelio) į eilutę (stulpelį) nekeliami. Žodžiai nepersidengia, bet trumpesnis žodis gali būti ilgesniojo žodžio dalimi.

Ekrane atspausdinkite parinktą n reikšmę ir kiekvieno žodžio pasikartojimų skaičių.

Trecias.txt	Matrica								
Berzas,sula;;sula;;klevu saldial lapasula	B	e	r	z	a	s	,	s	u
a aula, ar suart zemes vaikai du	l	a	;	;	s	u	l	a	;
Zodziai.txt	;	k	l	e	v	u		s	a
Sula	l	d	i	a	l		l	a	p
Alus	a	s	u	l	a			a	
Atsakymas			a	u	l	a	,		a
n = 9	r			s	u	a	r	t	
sula 3	z	e	m	e	s		v	a	i
alus 2	k	a	i		d	u			

1.2. Programos tekstas

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class FormL1 : System.Web.UI.Page
{
    public class Simboliai
    {
        public char s { get; set; }

        public Simboliai(char s)
        {
            this.s = s;
        }
    }
    public class Konteinerine
    {
        const int Max = 45; // max stulpeliu ir eiluciu
        private Simboliai[,] Simb;
        public Konteinerine()
        {
            Simb = new Simboliai[Max, Max];
        }

        public void Deti(int i, int j, Simboliai a)
        {
            Simb[i, j] = a;
        }

        public Simboliai Imti(int i, int j)
        {
            return Simb[i, j];
```

```
        }
    }
    const int MaxZodziu = 2000;
    const int MaxMatrica = 45;
    private const string f1 = @"F:\2tras\objektinis\L1\Trecias.txt";
    private const string f2 = @"F:\2tras\objektinis\L1\Zodziai.txt";
    private const string f3 = @"F:\2tras\objektinis\L1\Rezultatai.txt";
    //Konteinerine simbol = new Konteinerine();//dvimatis
    //string[,] Masyvas = new string[MaxMatrica, MaxMatrica];

    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void TextBox1_TextChanged(object sender, EventArgs e)
    {

    }
    /// <summary>
    /// Skaiciuoja, kiek zodziu yra vertikaliai
    /// </summary>
    /// <param name="zodziuC">kiek yra duotuju zodziu</param>
    /// <param name="zodziuMasyvas">duotuju zodziu masyvas</param>
    /// <param name="simboliai">simboliu konteineris</param>
    /// <param name="n">matricos dydis n*n</param>
    /// <param name="ZodziuSkaicius">Rastu zodziu skaicius</param>
    /// <param name="i">parametras, kuris nurodo, kuris zodis bus ieskomas</param>
    static void RecursionHORIZONTALE(int zodziuC, string[] zodziuMasyvas,
Konteinerine simboliai, int n, ref int[] ZodziuSkaicius, int i)
    {

        char[] zodRaides = zodziuMasyvas[i].ToCharArray();
        for (int k = 0; k < n; k++) //stulpeliai ||
        {
            bool tikrinaH = true;
            int nuoKurio = 0;

            for (int l = 0; l < n; l++) //eilutes --
            {
                if (zodRaides.Length <= (n - l)) // patikrina ar tas zodis tilps
iki eilutes pabaigos
                    if ((char.ToLower(zodRaides[0])) ==
char.ToLower(simboliai.Imti(l, k).s))
                    {
                        tikrinaH = true;
                        for (int m1 = 1; m1 < zodziuMasyvas[i].Length; m1++)
                        {
                            if ((m1 + l <= n) && (tikrinaH))
                            {
                                if ((char.ToLower(zodRaides[m1])) ==
char.ToLower(simboliai.Imti(l + m1, k).s))
                                    nuoKurio = l + m1;
                                else tikrinaH = false;
                            }
                        }

                        if (tikrinaH) ZodziuSkaicius[i] += 1;
                    }
            }
        }

        if ((i + 1) < zodziuC)
            RecursionHORIZONTALE(zodziuC, zodziuMasyvas, simboliai, n, ref
ZodziuSkaicius, i + 1);

    }
```

```
    /// <summary>
    /// Skaiciuoja, kiek zodziu yra horizontaliai
    /// </summary>
    /// <param name="zodziuC">kiek yra duotuju zodziu</param>
    /// <param name="zodziuMasyvas">duotuju zodziu masyvas</param>
    /// <param name="simboliai">simboliu konteineris</param>
    /// <param name="n">matricos dydis n*n</param>
    /// <param name="ZodziuSkaicius">Rastu zodziu skaicius</param>
    /// <param name="i">parametras, kuris nurodo, kuris zodis bus ieskomas</param>
    static void RecursionVERTIKALE(int zodziuC, string[] zodziuMasyvas,
Konteinerine simboliai, int n, ref int[] ZodziuSkaicius, int i)
    {
        bool tikrinaV = true;
        int nuoKurioV = 0;


        char[] zodRaides = zodziuMasyvas[i].ToCharArray();
        for (int l = 0; l < n; l++) //stulpeliai ||
        {
            for (int k = 0; k < n; k++) //eilutes --
            {
                if (zodRaides.Length <= (n - k)) // patikrina ar tas zodis tilps
iki stulpelio pabaigos
                    if ((char.ToLower(zodRaides[0])) ==
char.ToLower(simboliai.Imti(l, k).s))
                    {
                        tikrinaV = true;
                        for (int m1 = 1; m1 < zodziuMasyvas[i].Length; m1++)
                        {
                            if ((m1 + k <= n) && (tikrinaV))
                            {
                                if ((char.ToLower(zodRaides[m1])) ==
char.ToLower(simboliai.Imti(l, k + m1).s))
                                    nuoKurioV = k + m1;
                                else tikrinaV = false;
                            }
                        }

                        if (tikrinaV) ZodziuSkaicius[i] += 1;
                    }
            }
        }
        if ((i + 1) < zodziuC)
            RecursionVERTIKALE(zodziuC, zodziuMasyvas, simboliai, n, ref
ZodziuSkaicius, i + 1);

    }
    /// <summary>
    /// Skaiciuoja, kiek zodziu yra pagal diagonale
    /// </summary>
    /// <param name="zodziuC">kiek yra duotuju zodziu</param>
    /// <param name="zodziuMasyvas">duotuju zodziu masyvas</param>
    /// <param name="simboliai">simboliu konteineris</param>
    /// <param name="n">matricos dydis n*n</param>
    /// <param name="ZodziuSkaicius">Rastu zodziu skaicius</param>
    /// <param name="i">parametras, kuris nurodo, kuris zodis bus ieskomas</param>
    static void RecursionDIAGONALE(int zodziuC, string[] zodziuMasyvas,
Konteinerine simboliai, int n, ref int[] ZodziuSkaicius, int i)
    {
        bool tikrinaD = true;
        int nuoKurioD = 0;


        char[] zodRaides = zodziuMasyvas[i].ToCharArray();
        for (int k = 0; k < n; k++) //stulpeliai ||
        {
```

```
            for (int l = 0; l < n; l++) //eilutes --
            {
                if ((zodRaides.Length <= (n - l)) && (zodRaides.Length <= (n -
k))) // patikrina ar tas zodis tilps iki eilutes ir stulpelio pabaigos
                    if ((char.ToLower(zodRaides[0])) ==
char.ToLower(simboliai.Imti(l, k).s))
                    {
                        tikrinaD = true;
                        for (int m1 = 1; m1 < zodziuMasyvas[i].Length; m1++)
                        {
                            if ((m1 + l <= n) && (tikrinaD) && (m1 + k <= n))
                            {
                                if ((char.ToLower(zodRaides[m1])) ==
char.ToLower(simboliai.Imti(l + m1, k + m1).s))
                                    nuoKurioD = l + m1;
                                else tikrinaD = false;
                            }
                        }

                        if (tikrinaD) ZodziuSkaicius[i] += 1;
                    }
            }
        }
        if ((i + 1) < zodziuC)
            RecursionDIAGONALE(zodziuC, zodziuMasyvas, simboliai, n, ref
ZodziuSkaicius, i + 1);

    }
    /// <summary>
    /// Paspaudus mygtuka "start" bus atliekami veiksmai nurodyti siame void'e
    /// </summary>
    protected void Button1_Click(object sender, EventArgs e)
    {
        TextBox7.Text = " tarpas ";
        Konteinerine simboliai = new Konteinerine();
        string[] zodziuMasyvas = new string[MaxZodziu];
        int zodziuC;
        int n;
        Skaityti(simboliai, zodziuMasyvas, out zodziuC, out n);

        //-------surasys, kiek kartu buvo rastas kiekvienas zodis
        int[] ZodziuSkaicius = new int[zodziuC];
        for (int i = 0; i < zodziuC; i++)
        {
            ZodziuSkaicius[i] = 0;
        }

        //-----HORIZONTALIAI
        int iHoriz = 0;
        RecursionHORIZONTALE(zodziuC, zodziuMasyvas, simboliai, n, ref
ZodziuSkaicius, iHoriz);


        //--------------VERTIKALIAI
        int iVertik = 0;
        RecursionVERTIKALE(zodziuC, zodziuMasyvas, simboliai, n, ref
ZodziuSkaicius, iVertik);

        //--------------PAGAL DIAGONALE
        int iDiogon = 0;
        RecursionDIAGONALE(zodziuC, zodziuMasyvas, simboliai, n, ref
ZodziuSkaicius, iDiogon);

        string eill = "kvadratine matrica: " + n + " x " + n + "\r\n";
        for (int i = 0; i < zodziuC; i++)
        {
```

```
            eill += "zodis '" + zodziuMasyvas[i] + "' pasikartoja " +
ZodziuSkaicius[i] + " kartus";
            eill += "\r\n";

        }
        using (var writer = File.AppendText(f3))
        {
            writer.WriteLine();
            writer.WriteLine("Gauti rezultatai: ");
            writer.WriteLine();
            writer.WriteLine(eill);
        }
        TextBox6.Text = "Rezultatai:";
        TextBox3.Text = eill;
    }

    /// <summary>
    /// Skaito duomenis is duom failu
    /// </summary>
    /// <param name="simboliai">simboliu konteineris</param>
    /// <param name="zodziuMasyvas">duotuju zodziu masyvas</param>
    /// <param name="n">matricos dydis n*n</param>
    /// <param name="zodziuC">kiek yra duotuju zodziu</param>
    public void Skaityti(Konteinerine simboliai, string[] zodziuMasyvas, out int
zodziuC, out int nn)
    {
        int MaxEil = 50;
        string line;
        int simboliuSk = 0;
        string[] eilutes = new string[MaxEil];
        int teksEil = 0; // kiek duotam faile yra eiluciu

        using (StreamReader reader = new StreamReader(f1))
        {
            while ((line = reader.ReadLine()) != null)
            {
                simboliuSk += line.Length;
                eilutes[teksEil++] = line;
            }
        }
        //------------randa kvadratines matricos n x n
        double n = Math.Sqrt(simboliuSk);
        n = Math.Round(n, 1);
        if (n % Math.Round(n, 0) == 0) n = Math.Round(n, 0);
        else
        {
            if (n % Math.Round(n, 0) >= 0.5) n = Math.Round(n, 0);
            else n = n = Math.Round(n, 0) + 1;
        }
        //------------sudeda teksta i viena eilute
        nn = Convert.ToInt32(n);
        string failas = "";
        for (int i = 0; i < teksEil; i++)
        {
            failas = failas + eilutes[i];
        }
        //------------slpit'ina kas simboli ir deda i kvadratine matrica
        string[] sym = new string[simboliuSk];

        char[] symbol = failas.ToCharArray();

        int count = 0; char a;
        for (int jj = 0; jj < n; jj++)
            for (int ii = 0; ii < n; ii++)
            {
                if (count < failas.Length)
```

```
                    a = symbol[count];
                else
                    a = ' ';
                Simboliai ss = new Simboliai(a);
                simboliai.Deti(ii, jj, ss);
                count++;
            }

        //-------------skaito antro failo zodzius
        zodziuC = 0;
        string line1; string eilZod = "";
        using (StreamReader reader = new StreamReader(f2))
        {
            while ((line1 = reader.ReadLine()) != null)
            {
                if (line1.Length <= n / 2)
                    zodziuMasyvas[zodziuC++] = line1;
                eilZod += line1 + "\r\n";
            }
        }

        if (File.Exists(f3))
            File.Delete(f3);
        //------------suraso pradinius duomenis i rezultatu faila
        string eil = "";
        using (var writer = File.AppendText(f3))
        {
            writer.WriteLine("Duotas tekstas: ");
            writer.WriteLine();
            for (int i = 0; i < teksEil; i++)
            {
                writer.WriteLine(eilutes[i]);
            }
            writer.WriteLine("-----------------------------------");
            writer.WriteLine();
            writer.WriteLine("Zodziu matrica  ");
            writer.WriteLine();
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    writer.Write(" " + simboliai.Imti(j, i).s);
                    eil += " " + simboliai.Imti(j, i).s;
                }
                writer.WriteLine();
                eil += "\r\n";
            }
            writer.WriteLine("-----------------------------------");
            writer.WriteLine();
            writer.WriteLine("Duoti zodziai:");
            writer.WriteLine();
            for (int i = 0; i < zodziuC; i++)
            {
                writer.WriteLine(zodziuMasyvas[i]);
            }
            writer.WriteLine("-----------------------------------");
            TextBox1.Text = eil;
            TextBox2.Text = "Pradiniai duomenys";
            TextBox5.Text = "Duoti zodziai";
            TextBox4.Text = eilZod;
        }
    }
}
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="FormL1.aspx.cs"
Inherits="FormL1" %>
```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body style="background-color: #ADDFFF;">
    <form id="form1" runat="server">
    <div>
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
Text="START" />
    </div>
        <h1>LD1_7 ŽODŽIŲ PAIEŠKA</h1>
        <p>

            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>

            &nbs
p;         
            <asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
        </p>
        <p>
        <asp:TextBox ID="TextBox1" runat="server"
OnTextChanged="TextBox1_TextChanged" Height="146px" TextMode="MultiLine"
Width="201px"></asp:TextBox>
                    
        <asp:TextBox ID="TextBox4" runat="server"
OnTextChanged="TextBox1_TextChanged" Height="146px" TextMode="MultiLine"
Width="193px"></asp:TextBox>
        </p>
        <p>
            <asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>
        </p>
        <p>
         <asp:TextBox ID="TextBox3" runat="server" Height="168px"
TextMode="MultiLine" Width="466px"></asp:TextBox>

            &nbs
p;    
            <asp:TextBox ID="TextBox7" runat="server"></asp:TextBox>
        </p>
        <p>
             </p>
        <p>
             </p>
    </form>
</body>
</html>
```

1.3. Pradiniai duomenys ir rezultatai

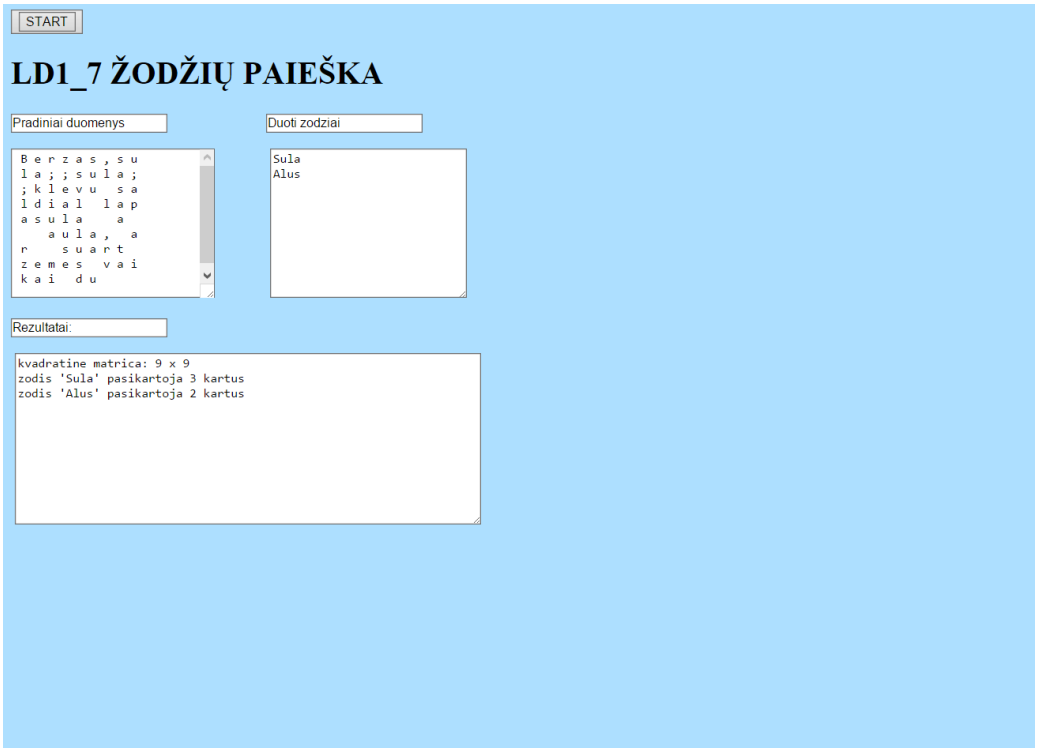
Pradiniai duomenys

```
Failas: Trecias.txt

Berzas,sula;;sula;;klevu saldial lapasula
 a   aula, ar  suart zemes vaikai du

Failas:  Zodziai.txt

Sula
Alus
```

Rezultatai

```
Failas: Rezultatai.txt

Duotas tekstas:

Berzas,sula;;sula;;klevu saldial lapasula
 a   aula, ar  suart zemes vaikai du
-----------------------------------

Zodziu matrica

 B e r z a s , s u
 l a ; ; s u l a ;
 ; k l e v u   s a
 l d i a l   l a p
 a s u l a     a
     a u l a ,   a
 r     s u a r t
 z e m e s   v a i
 k a i   d u
-----------------------------------

Duoti zodziai:

Sula
Alus
-----------------------------------

Gauti rezultatai:

kvadratine matrica: 9 x 9
zodis 'Sula' pasikartoja 3 kartus
zodis 'Alus' pasikartoja 2 kartus
```

1.4. Grafinės vartotojo sąsajos schema



1.5. Grafinės vartotojo sąsajos elementų pakeistos savybės

Komponentas	ID	text	Back Color	OnClick	TextMode
Button	Button1	START	#ADDFFF	Button1_Click	
TextBox	TextBox1		#FFCCFF		MultiLine
TextBox	TextBox2				
TextBox	TextBox3				
TextBox	TextBox4				
TextBox	TextBox5		#FFCCFF		MultiLine
TextBox	TextBox6				

1.6. Klasių diagramos

Veiksmai
+RecursionHORIZONTALE(in zodziuC:integer, in zodziuMasyvas:string[], in simboliai:Konteinerine, in n:integer, inout ZodziuSkaicius:integer[], in i:integer) +RecursionVERTIKALE(in zodziuC:integer, in zodziuMasyvas:string[], in simboliai:Konteinerine, in n:integer, inout ZodziuSkaicius:integer[], in i:integer) +RecursionDIAGONALE(in zodziuC:integer, in zodziuMasyvas:string[], in simboliai:Konteinerine, in n:integer, inout ZodziuSkaicius:integer[], in i:integer) +Page_Load(in sender:object, in e:EventArgs) +Page_Init(in sender:object, in e: EventArgs) +TextBox1_TextChanged(in sender:object, in e: EventArgs) +TextBox2_TextChanged(in sender:object, in e: EventArgs) +Button1_Click(in sender:object, in e: EventArgs)

Skaityti
+ Skaityti(){query}

1.7. Programos vartotojo vadovas

Sukuriamas tekstinis duomenų failas Trecias.txt ir paršomas tekstas. Tekstas surašomas į nxn matricą, n parenkamos toks mažiausias, kad tekstas tilptų į kvadratinę matricą, jei trūksta simbolių, užpildoma tarpais. Užpildžius duomenų failą, galima įjungti programą. Įjungę programą paspaudžiame mygtuką „START", kuris į ekraną išveda pradinius duomenis skiltyje „Pradiniai duomenys" ir išveda rezultatus skiltyje „Rezultatai". Baigę savo darbą, galime išeiti iš programos spausdami raudoną x mygtuką dešiniam viršutiniame programos kampe. Įvykdžius programą, taip pat sukuriamas tekstinis rezultatų failas Rezultatai.txt.

2. Susietasis sąrašas

3. Bendrinis susietasis sąrašas

4. Bendrinės kolekcijos

4.1. Darbo užduotis

4.2. Programos tekstas

4.3. Pradiniai duomenys ir rezultatai

4.4. Grafinės vartotojo sąsajos schema

4.5. Grafinės vartotojo sąsajos elementų pakeistos savybės

4.6. Klasių diagramos

4.7. Programos vartotojo vadovas

5. Deklaratyvusis programavimas