KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Objektinis programavimas II (P175B123)

Darbų aplankas

Atliko:

IFF-7/7 gr. studentas

Matas Goštautas

2018 m. vasario 19 d.

Priėmė:

Doc. Lina Narbutaitė

KAUNAS 2018

TURINYS

1. Rekursija (L1) 2

1.1. Darbo užduotis 2

1.2. Grafinės vartotojo sąsajos schema 2

1.3. Sąsajoje panaudotų komponentų keičiamos savybės 2

1.4. Klasių diagrama 2

1.5. Programos vartotojo vadovas 2

1.6. Programos tekstas 2

1.7. Pradiniai duomenys ir rezultatai 2

1.8. Dėstytojo pastabos 2

2. Dinaminis atminties valdymas (L2) 2

2.1. Darbo užduotis 2

2.2. Grafinės vartotojo sąsajos schema 2

2.3. Sąsajoje panaudotų komponentų keičiamos savybės 2

2.4. Klasių diagrama 2

2.5. Programos vartotojo vadovas 2

2.6. Programos tekstas 2

2.7. Pradiniai duomenys ir rezultatai 2

2.8. Dėstytojo pastabos 2

3. Bendrinės klasės ir sąsajos (L3) 2

3.1. Darbo užduotis 2

3.2. Grafinės vartotojo sąsajos schema 2

3.3. Sąsajoje panaudotų komponentų keičiamos savybės 2

3.4. Klasių diagrama 2

3.5. Programos vartotojo vadovas 2

3.6. Programos tekstas 2

3.7. Pradiniai duomenys ir rezultatai 2

3.8. Dėstytojo pastabos 2

4. Kolekcijos ir išimčių valdymas (L4) 2

4.1. Darbo užduotis 2

4.2. Grafinės vartotojo sąsajos schema 2

4.3. Sąsajoje panaudotų komponentų keičiamos savybės 2

4.4. Klasių diagrama 2

4.5. Programos vartotojo vadovas 2

4.6. Programos tekstas 2

4.7. Pradiniai duomenys ir rezultatai 2

4.8. Dėstytojo pastabos 2

5. Deklaratyvusis programavimas (L5) 2

5.1. Darbo užduotis 2

5.2. Grafinės vartotojo sąsajos schema 2

5.3. Sąsajoje panaudotų komponentų keičiamos savybės 2

5.4. Klasių diagrama 2

5.5. Programos vartotojo vadovas 2

5.6. Programos tekstas 2

5.7. Pradiniai duomenys ir rezultatai 2

5.8. Dėstytojo pastabos 2

# Rekursija (L1)

## Darbo užduotis

**LD\_14.Salos.**

Robotas iš didelio aukščio nufotografavo stačiakampį Žemės paviršiaus plotą. Gautas vaizdas buvo sudalintas kvadratiniais langeliais n × m (1 ≤ n, m ≤ 100). Langelis tamsus, jeigu jis atitinka sausumos dalelę ir šviesus, jeigu atitinka vandens plotelį.

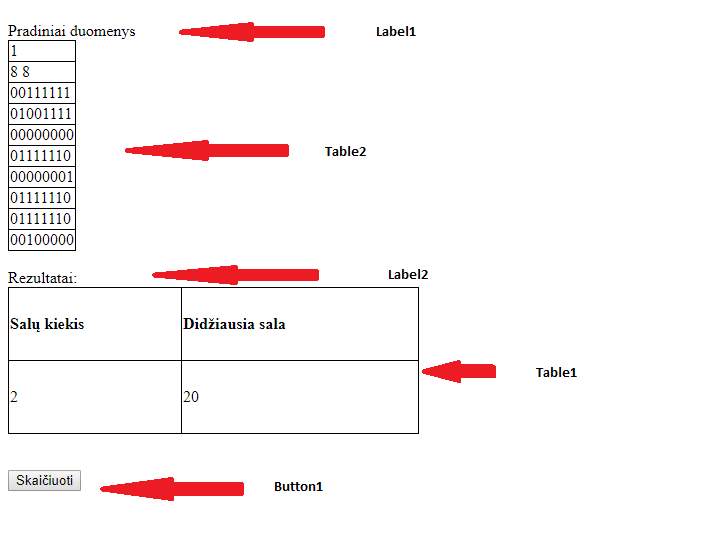
Parašykite programą, kuri suskaičiuotų, kiek yra salų ir kokio dydžio didžiausia sala. Sala – tai gretimų tamsių langelių visuma, iš visų pusių apribota šviesiais. Sala taip pat laikoma tamsių langelių visuma, esanti ant nufotografuoto ploto ribos (ne iš visų pusių supa vanduo).

Programa turi analizuoti k (1 ≤ k ≤ 10) tokių žemės plotų nuotraukų.

**Duomenys** surašyti tekstiniame faile U3.txt, kur pirmoje eilutėje yra k. Toliau – kiekvieno stačiakampio duomenys: pirmoje stačiakampio aprašo eilutėje yra n ir m reikšmės, o kitose eilutėse – stačiakampio langelių reikšmės eilutėmis. Tamsų langelį atitinka simbolis ‘1’, o šviesų ‘0‘.

**Rezultatai.** Ekrane spausdinti kiekvienoje eilutėje po du skaičius: kiekvieno paviršiaus ploto salų skaičių ir didžiausios salos langelių skaičių

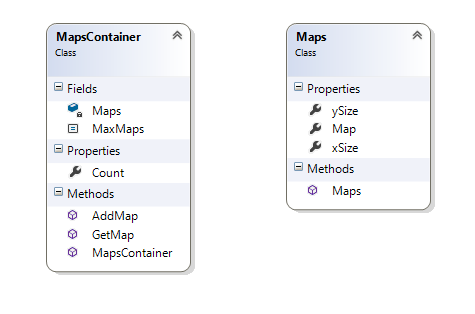
## Grafinės vartotojo sąsajos schema



## Sąsajoje panaudotų komponentų keičiamos savybės

|  |  |  |
| --- | --- | --- |
| Komponentas | Savybė | Reikšmė |
| Label1 | Text | Duomenys: |
| Label2 | Text | Rezultatai: |
| Table1 | BorderColor  BorderStyle  BorderWidth  GridLines | Black  Solid  1 px  Both |
| Table2 | BorderColor  BorderStyle  BorderWidth  GridLines | Black  Solid  1px  Both |
| Button1 | Text | Skaičiuoti |

## Klasių diagrama



## Programos vartotojo vadovas

Surašome duomenis tekstiniame faile U3.txt, kur pirmoje eilutėje yra nufotografuotų plotų skaičius. Toliau – kiekvieno stačiakampio duomenys: pirmoje stačiakampio aprašo eilutėje yra nuotraukų y ir x koordinatės, o kitose eilutėse – stačiakampio langelių reikšmės eilutėmis. Tamsų langelį atitinka simbolis ‘1’, o šviesų ‘0‘.

Įjungę programą paspaudžiame mygtuką „Skaičiuoti“. Lentelėje po užrašu „Duomenys:“ yra surašomi duomenys iš U3.txt tekstinio failo, o po „Rezultatai“ yra surašyti programos suskaičiuoti rezultatai.

## Programos tekstas

Map.cs klasė:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

public class Maps

{

public Maps(int[,] map, int xsize, int ysize)

{

Map = map;

xSize = xsize;

ySize = ysize;

}

public int[,] Map { get; set; }

public int xSize { get; set; }

public int ySize { get; set; }

}

MapsContainer.cs konteinerinė klasė:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

public class MapsContainer

{

public const int MaxMaps = 50;

private Maps[] Maps;

public int Count { get; private set; }

public MapsContainer(int size)

{

Maps = new Maps[size];

}

public void AddMap(Maps map)

{

Maps[Count++] = map;

}

public Maps GetMap(int index)

{

return Maps[index];

}

}

1lab.aspx:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="1lab.aspx.cs" Inherits="\_1lab" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<br />

<asp:Label ID="Label1" runat="server" Text="Pradiniai duomenys"></asp:Label>

<asp:Table ID="Table2" runat="server" BorderColor="Black" BorderStyle="Solid" BorderWidth="1px" GridLines="Both" Height="30px" HorizontalAlign="Justify" Width="29px">

</asp:Table>

<br />

<asp:Label ID="Label2" runat="server" Text="Rezultatai:"></asp:Label>

<br />

<asp:Table ID="Table1" runat="server" Height="147px" Width="411px" BorderColor="Black" BorderStyle="Solid" BorderWidth="1px" ForeColor="Black" GridLines="Both">

</asp:Table>

<br />

<br />

<asp:Button ID="Button1" runat="server" Text="Skaičiuoti" OnClick="Button1\_Click" />

</div>

</form>

</body>

</html>

1lab.aspx.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.IO;

public partial class \_1lab : System.Web.UI.Page

{

public const int xMax = 100; //maksimalus x ašies dydis

public const int yMax = 100; //maksimalys y ašies dydis

const string data = "App\_Data/U2.txt"; //duomenų failas

protected void Page\_Load(object sender, EventArgs e)

{

}

/// <summary>

/// Spausdina salos rezultatus į Table1

/// </summary>

/// <param name="count"> salų skaičius plote</param>

/// <param name="max"> didžiausios salos plote dydis</param>

private void Table(string count, string max)

{

TableCell Max = new TableCell();

Max.Text = max;

TableCell Count = new TableCell();

Count.Text = count;

TableRow row = new TableRow();

row.Cells.Add(Count);

row.Cells.Add(Max);

Table1.Rows.Add(row);

}

/// <summary>

/// Spausdina rezultatus į Rezultatai.txt

/// </summary>

/// <param name="results"> rezultatai</param>

/// <param name="Count"> Plotų kiekis</param>

private void ToFile(int[,]results, int Count)

{

string path = Server.MapPath("App\_Data/Rezultatai.txt");

using (StreamWriter writer = new StreamWriter(path))

{

writer.WriteLine("{0,-15}{1,-15}", "Salų skaičius", "Didž. sala");

writer.WriteLine("------------------------------");

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

{

writer.WriteLine("{0,-15}{1,-15}", results[i,0], results[i,1]);

}

}

}

/// <summary>

/// Spausdina duomenis

/// </summary>

/// <param name="results"> rezultatai</param>

/// <param name="Count"> Plotų kiekis </param>

private void Print(int[,] results, int Count)

{

TableRow row = new TableRow();

TableCell pavadinimas = new TableCell();

pavadinimas.Text = "<b>Salų kiekis</b>";

row.Cells.Add(pavadinimas);

TableCell kaina = new TableCell();

kaina.Text = "<b>Didžiausia sala</b>";

row.Cells.Add(kaina);

Table1.Rows.Add(row);

ToFile(results, Count);

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

{

Table(results[i, 0].ToString(), results[i, 1].ToString());

}

}

/// <summary>

/// Gražina rezultatus

/// </summary>

/// <param name="count"> Žemėlapių kiekis</param>

/// <returns> gražina žemėlapius</returns>

public int[,] Islands(out int count)

{

MapsContainer Maps = Reading();

PrintDataFile(Maps);

int[,] results = new int[Maps.Count, 2];

Do(Maps, ref results);

count = Maps.Count;

PrintDataWeb();

return results;

}

/// <summary>

/// Skaito duomenų faila

/// </summary>

/// <returns> gražina žemėlapių konteinerį</returns>

private static MapsContainer Reading()

{

using (StreamReader reader = new StreamReader(@data))

{

int count = int.Parse(reader.ReadLine());

MapsContainer a = new MapsContainer(count);

for (int i = 0; i < count; i++)

{

string[] dims = reader.ReadLine().Split();

int[] dim = new int[3];

dim[1] = int.Parse(dims[0]) + 2; // dim[1] - Y ašis ilgis

dim[2] = int.Parse(dims[1]) + 2; // dim[2] - X ašis ilgis

int[,] Empty = new int[xMax, yMax];

int[,] area = new int[xMax, yMax];

area = Empty;

for (int j = 1; j < dim[1] - 1; j++)

{

string line = reader.ReadLine();

for (int k = 1; k < dim[2] - 1; k++)

{

area[k, j] = int.Parse(line[k - 1].ToString());

}

}

Maps map = new Maps(area, dim[2], dim[1]);

a.AddMap(map);

}

return a;

}

}

/// <summary>

/// Atlieka skaičiavimo veiksmus

/// </summary>

/// <param name="Map"> Žemėlapių konteineris</param>

/// <param name="results"> rezultatų dvimatys masyvas [salų skaičius,2]</param>

private static void Do(MapsContainer Map, ref int[,] results)

{

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

{

int[,] Empty = new int[xMax, yMax];

int[,] area = new int[xMax, yMax];

area = Empty;

area = Map.GetMap(i).Map;

int IslandCount = 0;

int MaxTwos = 0;

for (int j = 1; j < Map.GetMap(i).ySize - 1; j++)

{

for (int k = 1; k < Map.GetMap(i).xSize - 1; k++)

{

if (area[k, j] == 1)

{

area[k, j] = 2;

ConvertNearbySquares(ref area, k, j);

IslandCount++;

int Twos = CountTwos(area, Map.GetMap(i).xSize, Map.GetMap(i).ySize);

if (MaxTwos < Twos)

{

MaxTwos = Twos;

}

ClearTwos(ref area, Map.GetMap(i).xSize, Map.GetMap(i).ySize);

}

}

}

results[i, 0] = IslandCount;

results[i, 1] = MaxTwos;

}

}

/// <summary>

/// Metodas su rekursija. Suranda visus langelį liečiančius langelius, ir tuos langelius liečiančius langelius ir t.t.

/// </summary>

/// <param name="area"> perduodamas žemėlapis, kuriame 0 - vanduo, 1 - dar netikrinta žemė, 2 - jau patikrinta žemė</param>

/// <param name="xpos"> langelio x koordinate </param>

/// <param name="ypos"> langelio y koordinate </param>

private static void ConvertNearbySquares(ref int[,] area, int xpos, int ypos)

{

area[xpos, ypos] = 2;

for (int y = -1; y <= 1; y++)

{

for (int x = -1; x <= 1; x++)

{

if (area[xpos + x, ypos + y] == 1)

{

ConvertNearbySquares(ref area, xpos + x, ypos + y);

}

}

}

}

/// <summary>

/// Kad būtų galima skaičiuot kitos salos dydį, ši sala panaikinama iš žemėlapio masyvo (konteineris išlieka nepakeistas)

/// </summary>

/// <param name="Map"> Žemėlapio masyvas</param>

/// <param name="x"> Žemėlapio didžiausia x koordinate </param>

/// <param name="y"> Žemėlapio didžiausia y koordinate </param>

private static void ClearTwos(ref int[,] Map, int x, int y)

{

for (int i = 0; i < y; i++)

{

for (int j = 0; j < x; j++)

{

if (Map[j, i] == 2)

Map[j, i] = 0;

}

}

}

/// <summary>

/// Suskaičiuoja visus dvejetus dvimatyje masyve, t.y. suskaičiuoja salos dydį

/// </summary>

/// <param name="Map"> žemėlapio dvimatis masyvas </param>

/// <param name="x"> žemėlapio didžiausia x koordinatė </param>

/// <param name="y"> žemėlapio didžiausia y koordinatė </param>

/// <returns></returns>

private static int CountTwos(int[,] Map, int x, int y)

{

int count = 0;

{

for (int i = 0; i < y; i++)

{

for (int j = 0; j < x; j++)

{

if (Map[j, i] == 2)

count++;

}

}

}

return count;

}

/// <summary>

/// Paspaudus Button1 atlieka skaičiavimus ir spausdinimą

/// </summary>

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

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

protected void Button1\_Click(object sender, EventArgs e)

{

int Count;

string path = Server.MapPath(data);

int[,] results = Islands(out Count);

Print(results, Count);

}

/// <summary>

/// spausdina duomenis ekrane

/// </summary>

private void PrintDataWeb()

{

string[] lines = File.ReadAllLines(Server.MapPath(data));

foreach (string line in lines)

{

TableRow row = new TableRow();

TableCell Line = new TableCell();

Line.Text = line;

row.Cells.Add(Line);

Table2.Rows.Add(row);

}

}

/// <summary>

/// spausdina duomenis faile

/// </summary>

/// <param name="Map"></param>

private void PrintDataFile(MapsContainer Map)

{

string path = Server.MapPath("App\_Data/Duomenys.txt");

using (StreamWriter writer = new StreamWriter(path))

{

writer.WriteLine("Salų kiekis: {0,10}", Map.Count);

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

{

writer.WriteLine("n ir m reikšmės: {0,5}{1,5}", Map.GetMap(i).ySize, Map.GetMap(i).xSize);

for (int j = 1; j < Map.GetMap(i).ySize-1; j++)

{

writer.Write("Stačiakampio duomenys: "); //24

for (int k = 1; k < Map.GetMap(i).xSize-1; k++)

{

writer.Write(Map.GetMap(i).Map[k, j]);

}

writer.WriteLine();

}

}

}

}

## Pradiniai duomenys ir rezultatai

**I variantas.**

|  |
| --- |
| U3.txt |
| 4  2 6  110100  000000  9 4  1000  1000  0110  1100  1100  0000  1010  1000  0000  10 7  0100110  0111000  0010110  1000000  1011100  1001110  0011000  1000100  0000100  0000000  2 2  00  01 |

|  |
| --- |
| Duomenys.txt |
| Salų kiekis: 4  n ir m reikšmės: 4 8  Stačiakampio duomenys: 110100  Stačiakampio duomenys: 000000  n ir m reikšmės: 11 6  Stačiakampio duomenys: 1000  Stačiakampio duomenys: 1000  Stačiakampio duomenys: 0110  Stačiakampio duomenys: 1100  Stačiakampio duomenys: 1100  Stačiakampio duomenys: 0000  Stačiakampio duomenys: 1010  Stačiakampio duomenys: 1000  Stačiakampio duomenys: 0000  n ir m reikšmės: 12 9  Stačiakampio duomenys: 0100110  Stačiakampio duomenys: 0111000  Stačiakampio duomenys: 0010110  Stačiakampio duomenys: 1000000  Stačiakampio duomenys: 1011100  Stačiakampio duomenys: 1001110  Stačiakampio duomenys: 0011000  Stačiakampio duomenys: 1000100  Stačiakampio duomenys: 0000100  Stačiakampio duomenys: 0000000  n ir m reikšmės: 4 4  Stačiakampio duomenys: 00  Stačiakampio duomenys: 00 |

|  |
| --- |
| Rezultatai.txt |
| Salų skaičius Didž. sala  ------------------------------  2 2  3 8  4 10  0 0 |

|  |
| --- |
| Vartotojo sąsaja |
| https://i.gyazo.com/9db3f28ab402c3557011f35f1151a758.png |

**II variantas.**

|  |
| --- |
| U3.txt |
| 1  8 8  00111111  01001111  00000000  01111110  00000001  01111110  01111110  00100000 |

|  |
| --- |
| Duomenys.txt |
| Salų kiekis: 1  n ir m reikšmės: 10 10  Stačiakampio duomenys: 00111111  Stačiakampio duomenys: 01001111  Stačiakampio duomenys: 00000000  Stačiakampio duomenys: 01111110  Stačiakampio duomenys: 00000001  Stačiakampio duomenys: 01111110  Stačiakampio duomenys: 01111110  Stačiakampio duomenys: 00100000 |

|  |
| --- |
| Rezultatai.txt |
| Salų skaičius Didž. sala  ------------------------------  2 20 |

|  |
| --- |
| Vartotojo sąsaja |
| https://i.gyazo.com/716d7f7c903ef89cb520fc6298608386.png |

## Dėstytojo pastabos

# Dinaminis atminties valdymas (L2)

## Darbo užduotis

## Grafinės vartotojo sąsajos schema

## Sąsajoje panaudotų komponentų keičiamos savybės

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

## Klasių diagrama

## Programos vartotojo vadovas

## Programos tekstas

## Pradiniai duomenys ir rezultatai

## Dėstytojo pastabos

# Bendrinės klasės ir sąsajos (L3)

## Darbo užduotis

## Grafinės vartotojo sąsajos schema

## Sąsajoje panaudotų komponentų keičiamos savybės

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

## Klasių diagrama

## Programos vartotojo vadovas

## Programos tekstas

## Pradiniai duomenys ir rezultatai

## Dėstytojo pastabos

# Kolekcijos ir išimčių valdymas (L4)

## Darbo užduotis

## Grafinės vartotojo sąsajos schema

## Sąsajoje panaudotų komponentų keičiamos savybės

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

## Klasių diagrama

## Programos vartotojo vadovas

## Programos tekstas

## Pradiniai duomenys ir rezultatai

## Dėstytojo pastabos

# Deklaratyvusis programavimas (L5)

## Darbo užduotis

## Grafinės vartotojo sąsajos schema

## Sąsajoje panaudotų komponentų keičiamos savybės

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

## Klasių diagrama

## Programos vartotojo vadovas

## Programos tekstas

## Pradiniai duomenys ir rezultatai

## Dėstytojo pastabos