KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Objektinis programavimas II (P175B123)

Darbų aplankas

Atliko:

<GRUPĖ> gr. studentas

<Vardas Pavardė>

2017 m. gegužės 15 d.

Priėmė:

Doc. Aštrys Kirvaitis

KAUNAS 2017

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

## 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

# Dinaminis atminties valdymas (L2)

## Darbo užduotis

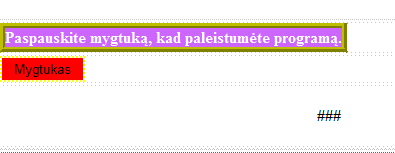
LD\_24. Detalės. Internetinėje parduotuvėje pirkėjai užsisakinėja robotų gamybai reikalingus įtaisus. Suraskite populiariausią įtaisą, kiek tokių įtaisų parduota ir už kokią sumą. Sudarykite tik vienos rūšies įtaisus pirkusių pirkėjų sąrašą, nupirktų įtaisų skaičių ir už juos sumokėtų pinigų sumą. Duomenys:

• tekstiniame faile U24a.txt yra informacija apie parduotuvėje parduodamus įtaisus: įtaiso kodas, įtaiso pavadinimas, įtaiso kaina;

• tekstiniame faile U24b.txt yra informacija apie pirkėjus: pirkėjo pavardė, vardas, pirkto įtaiso kodas, pirktų įtaisų kiekis.

Į kitą rinkinį atrinkite įtaisus, kurių parduota ne mažiau kaip n vienetų ir kurių vieneto kaina ne didesnė kaip k litų (n ir k įvedami klaviatūra).

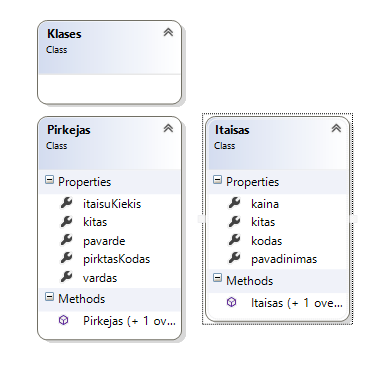
## Grafinės vartotojo sąsajos schema



## Sąsajoje panaudotų komponentų keičiamos savybės

|  |  |  |
| --- | --- | --- |
| Komponentas | Savybė | Reikšmė |
| Label1 | BackgroundColor, Font, TextColor | Purple, Bold, White, |
| Button1 | BackgroundColor, Bordercolor | Red, Yellow. |
| Table1 |  |  |

## Klasių diagrama



## Programos vartotojo vadovas

Įkėlus failą į programos aplanką ir paspaudus mygtuką, programa suranda visas reikiamas reikšmes.

## Programos tekstas

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.IO;

/// <summary>

/// Tadas Laurinaitis IFF-6/8

/// </summary>

namespace Lab2web

{

public partial class Forma1 : System.Web.UI.Page

{

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

{

}

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

{

Main();

}

const int maxItaisuSk = 5;

const string failas1 = @"..\..\U24a.txt";

const string failas2 = @"..\..\U24b.txt";

const string file = @"D:\Rezultatai.txt";

static void Main()

{

Itaisas visiItaisai = skaitytiItaisuDuomenis(failas1);

Pirkejas visiPirkejai = skaitytiPirkejuDuomenis(failas2);

int popItaisuKiek, vienosRusiesKiekis = 0;

double suma1, suma2 = 0;

string popItaisoPav = populiariausiasItaisas(visiItaisai, visiPirkejai, out popItaisuKiek, out suma1);

string[] vienosRusiesPirkejai = tikVienosRusies("ranka", visiItaisai, visiPirkejai, out vienosRusiesKiekis, out suma2);

Itaisas kitasRinkinys = KitasRinkinys(1, 1000, visiItaisai, visiPirkejai);

Spausdinimas(file, popItaisoPav, popItaisuKiek, suma1, suma2, vienosRusiesPirkejai, vienosRusiesKiekis, kitasRinkinys);

}

/// <summary>

/// perskaito Itaisu duomenis

/// </summary>

/// <param name="failas"></param>

/// <returns></returns>

static Itaisas skaitytiItaisuDuomenis(string failas)

{

Itaisas visiItaisai = new Itaisas();

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

{

string line = reader.ReadLine();

if (line != null)

{

string[] values = line.Split(';');

visiItaisai.kodas = values[0];

visiItaisai.pavadinimas = values[1];

visiItaisai.kaina = int.Parse(values[2]);

}

else if (line == null)

{

return null;

}

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

{

string[] values = line.Split(';');

Itaisas itaisas = new Itaisas();

itaisas.kodas = values[0];

itaisas.pavadinimas = values[1];

itaisas.kaina = int.Parse(values[2]);

itaisas.kitas = visiItaisai;

visiItaisai = itaisas;

}

}

return visiItaisai;

}

/// <summary>

/// perskaito pirkeju duomenis

/// </summary>

/// <param name="failas"></param>

/// <returns></returns>

static Pirkejas skaitytiPirkejuDuomenis(string failas)

{

Pirkejas visiPirkejai = new Pirkejas();

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

{

string line = reader.ReadLine();

if (line != null)

{

string[] values = line.Split(';');

visiPirkejai.pavarde = values[0];

visiPirkejai.vardas = values[1];

visiPirkejai.pirktasKodas = values[2];

visiPirkejai.itaisuKiekis = int.Parse(values[3]);

}

else if (line == null)

{

return null;

}

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

{

string[] values = line.Split(';');

Pirkejas pirkejas = new Pirkejas();

pirkejas.pavarde = values[0];

pirkejas.vardas = values[1];

pirkejas.pirktasKodas = values[2];

pirkejas.itaisuKiekis = int.Parse(values[3]);

pirkejas.kitas = visiPirkejai;

visiPirkejai = pirkejas;

}

}

return visiPirkejai;

}

/// <summary>

/// suranda populiariausio itaiso pavadinima, taip pat grazina ju skaiciu ir kainu suma

/// </summary>

/// <param name="visiItaisai"></param>

/// <param name="visiPirkejai"></param>

/// <param name="ind2"></param>

/// <param name="suma"></param>

/// <returns></returns>

static string populiariausiasItaisas(Itaisas visiItaisai, Pirkejas visiPirkejai, out int ind2, out double suma)

{

int ind1 = 0;

ind2 = 0;

suma = 0;

string popItaisPav = "";

Pirkejas visiPirkejai2 = new Pirkejas();

Itaisas visiItaisai2 = new Itaisas();

visiItaisai2 = visiItaisai;

while (visiItaisai != null)

{

visiPirkejai2 = visiPirkejai;

ind1 = 0;

while (visiPirkejai2 != null)

{

if (visiItaisai.kodas == visiPirkejai2.pirktasKodas)

{

ind1 += visiPirkejai2.itaisuKiekis;

}

visiPirkejai2 = visiPirkejai2.kitas;

}

if (ind1 >= ind2)

{

ind2 = ind1;

popItaisPav = visiItaisai.pavadinimas;

}

visiItaisai = visiItaisai.kitas;

}

while (visiItaisai2 != null)

{

if (popItaisPav == visiItaisai2.pavadinimas)

{

suma = ind2 \* visiItaisai2.kaina;

}

visiItaisai2 = visiItaisai2.kitas;

}

return popItaisPav;

}

/// <summary>

/// suranda tik vienos rusies itaiso pirkejus, ju kieki, pinigu suma

/// </summary>

/// <param name="itaisoPavadinimas"></param>

/// <param name="visiItaisai"></param>

/// <param name="visiPirkejai"></param>

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

/// <param name="suma"></param>

/// <returns></returns>

static string[] tikVienosRusies(string itaisoPavadinimas, Itaisas visiItaisai, Pirkejas visiPirkejai, out int kiekis, out double suma)

{

string kodas = "";

double kaina = 0;

suma = 0;

kiekis = 0;

int count = 0;

string[] vardai = new string[maxItaisuSk];

while (visiItaisai != null)

{

if (itaisoPavadinimas == visiItaisai.pavadinimas)

{

kodas = visiItaisai.kodas;

kaina = visiItaisai.kaina;

}

visiItaisai = visiItaisai.kitas;

}

if (kodas == "" && kaina == 0)

{

Console.WriteLine("Itaiso tokiu pavadinimu nera.");

}

while (visiPirkejai != null)

{

if (visiPirkejai.pirktasKodas == kodas)

{

vardai[count] = visiPirkejai.vardas + " " + visiPirkejai.pavarde;

kiekis += visiPirkejai.itaisuKiekis;

suma += visiPirkejai.itaisuKiekis \* kaina;

count++;

}

visiPirkejai = visiPirkejai.kitas;

}

return vardai;

}

/// <summary>

/// atrenka itaisus pagal uzduoties nurodymus

/// </summary>

/// <param name="n"></param>

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

/// <param name="visiItaisai"></param>

/// <param name="visiPirkejai"></param>

/// <returns></returns>

static Itaisas KitasRinkinys(int n, double k, Itaisas visiItaisai, Pirkejas visiPirkejai)

{

int sk = 0;

Itaisas naujasRinkinys = new Itaisas();

Pirkejas visiPirkejai2 = new Pirkejas();

while (visiItaisai != null)

{

if (visiItaisai.kaina <= k)

{

sk = 0;

visiPirkejai2 = visiPirkejai;

while (visiPirkejai2 != null)

{

if (visiItaisai.kodas == visiPirkejai2.pirktasKodas)

{

sk = sk + visiPirkejai2.itaisuKiekis;

}

visiPirkejai2 = visiPirkejai2.kitas;

}

if (sk >= n)

{

Itaisas naujesnis = new Itaisas();

naujesnis.kodas = visiItaisai.kodas;

naujesnis.pavadinimas = visiItaisai.pavadinimas;

naujesnis.kaina = visiItaisai.kaina;

naujesnis.kitas = naujasRinkinys;

naujasRinkinys = naujesnis;

}

}

visiItaisai = visiItaisai.kitas;

}

return naujasRinkinys;

}

/// <summary>

/// spausdina rezultatus faile

/// </summary>

/// <param name="file"></param>

/// <param name="popItPav"></param>

/// <param name="popItKiek"></param>

/// <param name="suma1"></param>

/// <param name="suma2"></param>

/// <param name="tikVienosRusies"></param>

/// <param name="vienosRusiesKiekis"></param>

/// <param name="kitasRinkinys"></param>

static void Spausdinimas(string file, string popItPav, int popItKiek, double suma1, double suma2, string[] tikVienosRusies, int vienosRusiesKiekis, Itaisas kitasRinkinys)

{

using (StreamWriter writer = new StreamWriter(@file))

{

writer.WriteLine("Populiariausio prietaiso pavadinimas, jo pardavimo skaičius ir kaina: pavadinimas - {0}, skaicius - {1}, kaina - {2}euro.", popItPav, popItKiek, suma1);

writer.WriteLine("Vienos rusies pirkeju sarasas, nupirktu itaisu skaicius ir uz juos sumoketu pinigu suma: ");

for(int i = 0; i < tikVienosRusies.Length-1; i++)

{

writer.WriteLine("|{0, 15}|", tikVienosRusies[i]);

}

writer.WriteLine("Itaisu skaicius: {0} Sumoketa suma: {1}euro.", vienosRusiesKiekis, suma2);

while (kitasRinkinys != null)

{

if (kitasRinkinys.pavadinimas != "")

{

writer.Write("|{0, 4}", kitasRinkinys.pavadinimas);

kitasRinkinys = kitasRinkinys.kitas;

}

}

}

}

}

}

## Pradiniai duomenys ir rezultatai

Pradinių duomenų failai:

U24a.txt

Kodas;pavadinimas;vieneto kaina;

24fa;ranka;15;

25fb;koja;14;

26fc;sirdis;75;

27fd;petis;18;

28fe;nosis;142;

29ff;akis;155;

U24b.txt

Pavarde;Vardas;irenginio kodas;pirktas kiekis;

Zemaitis;Kazimieras;24fa;16;

Petraitis;Kazimieras;24fa;2;

Keturakis;Kazimieras;24fa;2;

Jonaitis;Kazimieras;25fb;4;

Zaliasis;Kazimieras;26fc;8;

Mentaitis;Kazimieras;27fd;1;

Brigaitis;Kazimieras;29ff;12;

Ausraitis;Kazimieras;28fe;7;

Kinderis;Kazimieras;25fb;55;

Karaitis;Kazimieras;29ff;13;

Zvaigzdaitis;Kazimieras;25fb;5;

Selmaitis;Kazimieras;24fa;9;

Vienaitis;Kazimieras;29ff;3;

Rezultatai:

Rezultatai.txt

Populiariausio prietaiso pavadinimas, jo pardavimo skaičius ir kaina: pavadinimas - koja, skaicius - 64, kaina - 896euro.

Vienos rusies pirkeju sarasas, nupirktu itaisu skaicius ir uz juos sumoketu pinigu suma:

|Kazimieras Selmaitis|

|Kazimieras Keturakis|

|Kazimieras Petraitis|

|Kazimieras Zemaitis |

Itaisu skaicius: 29 Sumoketa suma: 435euro.

|ranka|koja|sirdis|petis|nosis|akis|

## Dėstytojo pastabos

Rezultatu

# Bendrinės klasės ir sąsajos (L3)

## Darbo užduotis

## Grafinės vartotojo sąsajos schema

## Sąsajoje panaudotų komponentų keičiamos savybės

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

## Klasių diagrama

## Programos vartotojo vadovas

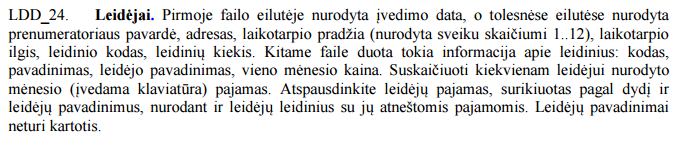
## Programos tekstas

## Pradiniai duomenys ir rezultatai

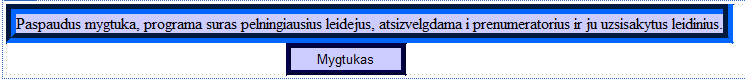
## Dėstytojo pastabos

# Kolekcijos ir išimčių valdymas (L4)

## Darbo užduotis



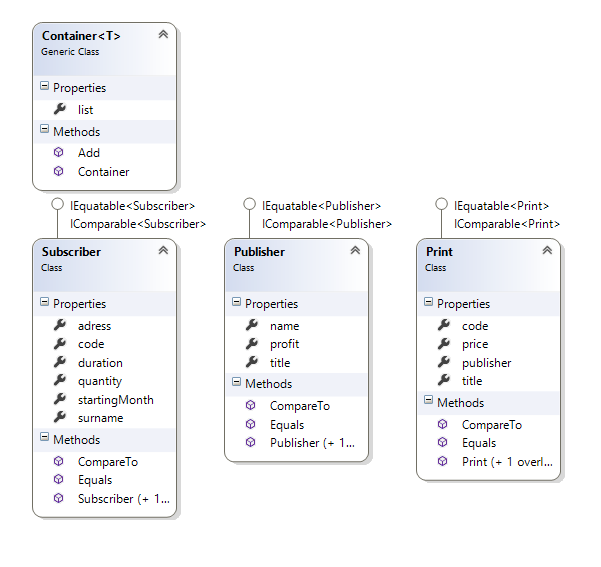
## Grafinės vartotojo sąsajos schema



## Sąsajoje panaudotų komponentų keičiamos savybės

|  |  |  |
| --- | --- | --- |
| Komponentas | Savybė | Reikšmė |
| Label1 | BorderWidth, BorderColour,BackgroundColour | 5px, Blue, Cyan |
| Button1 | BorderWidth, BorderColour,BackgroundColour | 3px, Dark Blue, Cyan |
|  |  |  |

## Klasių diagrama



## Programos vartotojo vadovas

Programa apskaicuoja pelningiausius leidejus, pagal leidinius ir leidiniu prenumeratorius. Tekstiniuose failuose nurodomi prenumeratoriu duomenys (pavarde, kiekis, data, trukme, kodas, adresas) ir leidiniu duomenys (kodas, kaina, leidejas, pavadinimas).

## Programos tekstas

Print.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Lab4Web

{

public class Print : IEquatable<Print>, IComparable<Print>

{

public string code { get; set; }

public string title { get; set; }

public string publisher { get; set; }

public double price { get; set; }

public Print() { }

public Print(string code, string title, string publisher, double price)

{

this.code = code;

this.title = title;

this.publisher = publisher;

this.price = price;

}

public bool Equals(Print other)

{

return (price == other.price);

}

public int CompareTo(Print other)

{

return (price.CompareTo(other.price));

}

}

}

Subscriber.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Lab4Web

{

public class Subscriber : IEquatable<Subscriber>, IComparable<Subscriber>

{

public string surname { get; set; }

public string adress { get; set; }

public int startingMonth { get; set; }

public int duration { get; set; }

public string code { get; set; }

public int quantity { get; set; }

public Subscriber() { }

public Subscriber(string surname, string adress, int startingMonth, int duration, string code, int quantity)

{

this.surname = surname;

this.adress = adress;

this.startingMonth = startingMonth;

this.duration = duration;

this.code = code;

this.quantity = quantity;

}

public bool Equals(Subscriber other)

{

return (code == other.code);

}

public int CompareTo(Subscriber other)

{

return (code.CompareTo(other.code));

}

}

Publisher.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Lab4Web

{

public class Publisher : IEquatable<Publisher>, IComparable<Publisher>

{

public string name { get; set; }

public double profit { get; set; }

public string title { get; set; }

public Publisher() { }

public Publisher(string name, double profit, string title)

{

this.name = name;

this.profit = profit;

this.title = title;

}

public bool Equals(Publisher other)

{

return (profit == other.profit);

}

public int CompareTo(Publisher other)

{

return (profit.CompareTo(other.profit));

}

}

}

Container.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Lab4Web

{

public class Container<T> where T : IEquatable<T>, IComparable<T>

{

public List<T> list { get; set; }

public Container()

{

list = new List<T>();

}

public void Add(T data)

{

list.Add(data);

}

}

}

## Pradiniai duomenys ir rezultatai

Pradiniai duomenys:

| Pavarde | Adresas | Prad. Men. | Trukme | Kodas | Kiekis |

-------------------------------------------------------------------------------------------------

| Antanaitis | Jono-5 | 2 | 6 | 4A | 1 |

| Mantainas | Sid-8 | 5 | 5 | 4A | 2 |

| Kombainas | Zuko-13 | 8 | 4 | 4A | 22 |

| Miestelenas | Jono-415 | 12 | 3 | 4B | 3 |

| Anglaitis | Jono-75 | 2 | 2 | 4B | 14 |

| Giedraitis | Jonso-5 | 7 | 12 | 4C | 1 |

| Bliumas | Sieed-8 | 9 | 2 | 4C | 2 |

| Zomsinas | Zxcuko-13 | 8 | 1 | 4D | 1 |

| Nielsenas | Jkolno-415 | 12 | 5 | 4D | 5 |

| Leslis | Jonlolo-75 | 5 | 6 | 4D | 7 |

| Kodas | Pavadinimas | Leidejas | Kaina |

---------------------------------------------------------------

| 4A | Ratai | Aibe | 14 |

| 4B | Sodas | Alba | 9 |

| 4C | Elektronika | Fortas | 4 |

| 4D | Miestas | Sigma | 13 |

Surikiuoti rezultatai:

| Pelnas | Leidejas | Leidinys |

----------------------------------------------

| 308 | Aibe | Ratai |

| 65 | Sigma | Miestas |

| 27 | Alba | Sodas |

| 4 | Fortas | Elektronika |

## Dėstytojo pastabos

Peržiūrėti failų išdėstymo tvarką, pataisyti vartotojo vadovą, grafinės sąsajos schemą.

Testas - 2

# 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