

OBJEKTINIO PROGRAMAVIMO PAGRINDAI I (P175B117)

Laboratorinių darbų ataskaita

Atliko:

IIINFQ klasės mokinys

Augustas Mačijauskas

2017 m. rugsėjo 21 d.

Priėmė:

Kęstutis Simonavičius, Jūratė Pauliūtė

TURINYS

1. Pažintis su klase.....	3
1.1. Darbo užduotis	3
1.2. Programos tekstas.....	3
1.3. Pradiniai duomenys ir rezultatai.....	6
2. Objektų rinkinys	7
2.1. Darbo užduotis	7
2.2. Programos tekstas.....	7
2.3. Pradiniai duomenys ir rezultatai.....	7
3. Konteinerinė klasė.....	8
3.1. Darbo užduotis	8
3.2. Programos tekstas.....	8
3.3. Pradiniai duomenys ir rezultatai.....	8
4. Teksto analizė ir redagavimas	9
4.1. Darbo užduotis	9
4.2. Programos tekstas.....	9
4.3. Pradiniai duomenys ir rezultatai.....	9
5. Susieti rinkiniai.....	10
5.1. Darbo užduotis	10
5.2. Programos tekstas.....	10
5.3. Pradiniai duomenys ir rezultatai.....	10

1. Pažintis su klase

1.1. Darbo užduotis

- Sukurkite klasę `Studentas`, kuri turėtų kintamuosius amžiui ir ūgiui saugoti. Trys studentai nutarė treniruotis žaisti krepšinį. Raskite, koks aukščiausio studento amžius ir koks jauniausio studento ūgis.
- Papildykite klasę `Studentas` kintamuoju, skirtu studento svoriui saugoti. Sukurkite klasę `Liftas`, kuri turėtų kintamuosius lifto keliamosios galios reikšmei ir talpai saugoti. Per kelis kartus visi studentai pakils liftu į reikiamą aukštą?
- Papildykite klasę `Liftas` metodais `Dėti()`, kurie leistų keisti lifto keliamąją galią ir talpą. Ar visi studentai vienu metu bus pakelti į reikiamą aukštą, jeigu lifto keliamoji galia bus padvigubinta? O jeigu talpa bus padvigubinta?

1.2. Programos tekstas

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

namespace Labaratorinis_U2_2
{
    class Studentas
    {
        private int amzius, //amzius studento
                  svoris; //studento svoris

        double ugis; //studento ugis

        public Studentas(int amzius, double ugis, int svoris)
        {
            this.amzius = amzius;
            this.ugis = ugis;
            this.svoris = svoris;
        }

        public int koksAmzius() { return amzius; }

        public double koksUgis() { return ugis; }

        public int koksSvoris() { return svoris; }
    }

    class Liftas
    {
        private int galia,
                  talpa;
        public Liftas(int galia, int talpa)
        {
            this.galia = galia;
            this.talpa = talpa;
        }

        public int kokiaGalia() { return galia; }

        public int kokiaTalpa() { return talpa; }

        public void DidintiGalia(int x)
        {
            galia *= x;
        }
    }
}
```

```

    public void DidintiTalpa(int x)
    {
        talpa *= x;
    }

    public void MazintiTalpa(int x)
    {
        talpa /= x;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Studentas[] studentai = new Studentas[3];

        studentai[0] = new Studentas(20, 1.78, 76);
        studentai[1] = new Studentas(19, 1.85, 82);
        studentai[2] = new Studentas(21, 1.79, 73);
        Liftas l = new Liftas(150, 3);

        for (int i = 0; i < studentai.Length; i++)
        {
            if (studentai[i].koksUgis() == auksciausioAmzius(studentai))
            {
                Console.WriteLine("Auksciausio studento amzius: {0,6:d}",
studentai[i].koksAmzius());
            }
        }

        for (int j = 0; j < studentai.Length; j++)
        {
            if (studentai[j].koksAmzius() == jauniausioUgis(studentai))
            {
                Console.WriteLine("Jauniausio studento ugis:    {0,6:f2}",
studentai[j].koksUgis());
            }
        }
        Console.WriteLine("");

        int visuStudentuSvoris = svoris(studentai);
        int kiekStudentu = studentai.Length;
        int kiekKartuKelsimes;
        kiekKartuKelsimes = kiekKartuReiksKelti(kiekStudentu, studentai, l);

        Console.WriteLine("Vidutinis studentu svoris:    {0,4:d}", visuStudentuSvoris /
kiekStudentu);
        Console.WriteLine("");

        int kiekKilimu = kiekKilsim(l, studentai, visuStudentuSvoris, kiekKartuKelsimes);
        Console.WriteLine("Studentai pakils per:    {0,5:d} k.", kiekKilimu);
        Console.WriteLine("");

        l.DidintiTalpa(2);
        kiekKilimu = kiekKilsim(l, studentai, visuStudentuSvoris, kiekKartuKelsimes);
        Console.WriteLine("Padidinus talpa studentai pakils per: {0,5:d} k.", kiekKilimu);
        l.MazintiTalpa(2); // Graziname talpa i pradine verte, kad nepasikeistu galios
skaiciavimas

        l.DidintiGalia(2);
        kiekKilimu = kiekKilsim(l, studentai, visuStudentuSvoris, kiekKartuKelsimes);
        Console.WriteLine("Padidinus galia studentai pakils per: {0,5:d} k.", kiekKilimu);
        Console.WriteLine("");
    }

    static int kiekKartuReiksKelti(int kiekStudentu, Studentas[] studentai, Liftas l)
    {

```

```

        int kiekKartuReiksKelti;
        if (kiekStudentu % l.kokiaTalpa() == 0)
        {
            kiekKartuReiksKelti = kiekStudentu / l.kokiaTalpa();
        }
        else
        {
            kiekKartuReiksKelti = kiekStudentu / l.kokiaTalpa() + 1;
        }
        return kiekKartuReiksKelti;
    }

    static int kiekKilsim(Liftas l, Studentas[] studentai, int visuStudentuSvoris, int x)
    {
        int kiekKilimu = 0;
        if (x * l.kokiaGalia() > visuStudentuSvoris)
        {
            kiekKilimu = x;
        }
        else
        {
            kiekKilimu = x + 1;
        }
        return kiekKilimu;
    }

    static int jauniausioUgis(Studentas[] studentai)
    {
        int jauniausioAmzius = studentai[0].koksAmzius();
        for (int i = 1; i < studentai.Length; i++)
        {
            if (studentai[i].koksAmzius() < jauniausioAmzius)
            {
                jauniausioAmzius = studentai[i].koksAmzius();
            }
        }
        return jauniausioAmzius;
    }

    static int svoris(Studentas[] studentai)
    {
        int visuSvoris = 0;

        for (int i = 0; i < studentai.Length; i++)
        {
            visuSvoris += studentai[i].koksSvoris();
        }
        return visuSvoris;
    }

    static double auksciausioAmzius(Studentas[] studentai)
    {
        double auksciausioUgis = studentai[0].koksUgis();
        for (int i = 1; i < studentai.Length; i++)
        {
            if (studentai[i].koksUgis() > auksciausioUgis)
            {
                auksciausioUgis = studentai[i].koksUgis();
            }
        }
        return auksciausioUgis;
    }
}

```

1.3. Pradiniai duomenys ir rezultatai

Pirmi pradiniai duomenys:

	Studentas1	Studentas2	Studentas3
Amžius	20	19	21
Ūgis	1.78	1.85	1.79
Svoris	76	82	73

	Liftas1
Keliamoji galia	150
Talpa	3

Rezultatai su pirmaisiais duomenimis:

```
C:\WINDOWS\system32\cmd.exe
Auksciausio studento amžius:      19
Jauniausio studento ūgis:      1,85

Vidutinis studentu svoris:      77

Studentai pakils per:      2 k.

Padidinus talpa studentai pakils per:      2 k.
Padidinus galia studentai pakils per:      1 k.

Press any key to continue . . .
```

Antri pradiniai duomenys:

	Studentas1	Studentas2	Studentas3
Amžius	20	19	19
Ūgis	1.85	1.85	1.79
Svoris	75	83	76

	Liftas1
Keliamoji galia	175
Talpa	2

Rezultatai su antraisiais duomenimis:

```
C:\WINDOWS\system32\cmd.exe
Auksciausio studento amžius:      20
Auksciausio studento amžius:      19
Jauniausio studento ūgis:      1,85
Jauniausio studento ūgis:      1,79

Vidutinis studentu svoris:      78

Studentai pakils per:      2 k.

Padidinus talpa studentai pakils per:      2 k.
Padidinus galia studentai pakils per:      2 k.

Press any key to continue . . .
```

2. Objektų rinkinys

2.1. Darbo užduotis

2.2. Programos tekstas

2.3. Pradiniai duomenys ir rezultatai

3. Konteinerinė klasė

3.1. Darbo užduotis

3.2. Programos tekstas

3.3. Pradiniai duomenys ir rezultatai

4. Teksto analizė ir redagavimas

4.1. Darbo užduotis

4.2. Programos tekstas

4.3. Pradiniai duomenys ir rezultatai

5. Susieti rinkiniai

5.1. Darbo užduotis

5.2. Programos tekstas

5.3. Pradiniai duomenys ir rezultatai