

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**

**OBJEKTINIS PROGRAMAVIMAS I (P175B118)**  
***Laboratorinio darbo ataskaita***

Atliko:

IFF-5/7 gr. studentas

Viktorija Ražaitė

2015 m. rugsėjo 25 d.

Priėmė:

Lektorius Dr. Mindaugas Jančiukas

# TURINYS

<b>1. Objektų rinkinys .....</b>	<b>3</b>
1.1. Darbo užduotis .....	3
1.2. Programos tekstas.....	3
1.3. Pradiniai duomenys ir rezultatai.....	6
1.3.1 Pradiniai duomenys .....	6
1.3.2 Rezultatai .....	6
<b>2. Konteineris.....</b>	<b>7</b>
2.1. Darbo užduotis .....	7
2.2. Programos tekstas.....	7
2.3. Pradiniai duomenys ir rezultatai.....	7
<b>3. Susieti objektų rinkiniai .....</b>	<b>8</b>
3.1. Darbo užduotis .....	8
3.2. Programos tekstas.....	8
3.3. Pradiniai duomenys ir rezultatai.....	8
<b>4. Teksto analizė ir redagavimas .....</b>	<b>9</b>
4.1. Darbo užduotis .....	9
4.2. Programos tekstas.....	9
4.3. Pradiniai duomenys ir rezultatai.....	9
<b>5. Sudėtingesnis konteineris .....</b>	<b>10</b>
5.1. Darbo užduotis .....	10
5.2. Programos tekstas.....	10
5.3. Pradiniai duomenys ir rezultatai.....	10

# 1. Objektų rinkinys

## 1.1. Darbo užduotis

7. WCG turnyras. Kaune vyksta atrankinis kompiuterinio žaidimo „League of Legends“ turnyras. Žaidime dvi penkių žaidėjų komandos kovoja tarpusavyje valdydamos skirtingus čempionus, siekdamos sunaikinti priešininkų bazės gilumoje stovintį „Nexus“. Duomenų faile pateikta informacija apie pirmo rato dalyvius ir jų rezultatus: vardas, pavardė, komanda, pozicija, čempionas, sunaikinimai(K), dalyvavimai sunaikinimuose(A).

- Raskite žaidėją, pademonstravusį geriausią asmeninį rezultatą. Palyginimui naudokite vadinamąjį KA rodiklį (K+A). Ekrane atspausdinkite jo vardą, pavardę, komandos pavadinimą, poziciją bei naudotą čempioną.

- Raskite, kuris čempionas buvo naudotas „universaliausiai“ (daugiausiai skirtingų pozicijų). Ekrane atspausdinkite čempiono pavadinimą, bei kokiose pozicijose jis buvo naudotas.

- Sudarykite „Top“ pozicijoje žaidusių žaidėjų sąrašą, į failą „Top.csv“ įrašykite žaidėjų komandos pavadinimus, pavardes, vardus, naudoto čempiono pavadinimą.

## 1.2. Programos tekstas

```
Klasė
using System;

namespace _7Uzd
{
    class Turnyras
    {
        public string Vardas { get; set; }
        public string Pavarde { get; set; }
        public string Komanda { get; set; }
        public string Pozicija { get; set; }
        public string Cemp { get; set; }
        public int Sunaik { get; set; }
        public int Zuvo { get; set; }
        public int Dalyvav { get; set; }

        public Turnyras()
        {
        }

        public Turnyras(string vardas, string pavarde, string komanda, string
pozicija, string cemp, int sunaik, int zuvo, int dalyvav)
        {
            Vardas = vardas;
            Pavarde = pavarde;
            Komanda = komanda;
            Pozicija = pozicija;
            Cemp = cemp;
            Sunaik = sunaik;
            Zuvo = zuvo;
            Dalyvav = dalyvav;
        }
    }
}

Programa

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

```

namespace _7Uzd
{
    class Program
    {
        public const int MaxZaid = 50; //maksimalus zaideju skaicius

        static void Main(string[] args)
        {
            Turnyras[] turnyras;
            int turnCount = 0;
            int maxRod;
            ReadData(out turnyras, out turnCount); //nuskaitomi duomenys is
            duomenu failo
            GerAsmRez(turnyras, turnCount, out maxRod); // maxRod - didziausias
            rodiklis GerAsmRez - Geriausias asmeninis rezultatas
            Console.WriteLine(" ");
            // Lenteles paruosimas
            Console.WriteLine("-----");
            Console.WriteLine("-----Zaidejas(zaidejai)-pasieke-geriausia-
            asmenini-rezultata-----");
            Console.WriteLine("-----");
            Console.WriteLine("  Vardas, pavarde      | Komanda      | Pozicija |
            Cempionas ");
            Console.WriteLine("-----");
            for (int i = 0; i < turnCount; i++)
            {
                //Apskaiciuojamas KA rodiklis
                if (turnyras[i].Sunaik + turnyras[i].Dalyvav == maxRod)
            //Duomenu surasymas i faila
                {
                    Console.WriteLine(" {0} {1} | {2}      | {3}      | {4}          ",
                    turnyras[i].Vardas, turnyras[i].Pavarde, turnyras[i].Komanda,
                    turnyras[i].Pozicija, turnyras[i].Cemp);
                    Console.WriteLine("-----");
                }
            }
            Univers(turnyras, turnCount);           // universaliausio cempiono
            ieskojimas
            TopPozicija(turnyras, turnCount);       // Top pozicijos zaideju radimas

        } // Randami zaidejai zaidziantys top pozicijoje
        private static void TopPozicija(Turnyras[] turnyras, int turnCount)
        {
            using (StreamWriter writer = new StreamWriter(@"Top.csv")) //duomenys
            bus rasomi i excel programa
            {
                writer.WriteLine("Komanda,Pavarde,Vardas,Cempionas");
            //Lenteles paaiskinimai
                for (int i = 0; i < turnCount; i++)           // ima is masyvo
            zaideju pozicijos pavadinimus ir lygina
                {
                    // jei pozicija lygi
            zodziui 'Top', tuomet
                    if (turnyras[i].Pozicija == "Top") // i excel irasomas
            komandos pavadinimas, zaidejo pavarde vardas ir cempionas kuriame zaidziama
                    writer.WriteLine("{0},{1},{2},{3}", turnyras[i].Komanda,
            turnyras[i].Pavarde, turnyras[i].Vardas, turnyras[i].Cemp);
                }
            }
        } // Universaliausio cempiono ieskojimas, tai tas kuriame zaidzia
            daugiausiai zaideju skirtingomis pozicijomis
        private static void Univers(Turnyras[] turnyras, int turnCount)
        {

```

```

        int[] PozicMas = new int[turnCount]; // sukuriamas naujas masyvas,
jame bus saugoma tam tikro cempiono skirtingu poziciju skaicius
        for (int i = 0; i < turnCount; i++)
            for (int j = i+1; j < turnCount; j++)
                { //lyginama, jei cempiono vardai tokie patys ir jei skiriasi
pozicijos
                    if ((turnyras[i].Cemp == turnyras[j].Cemp) &&
(turnyras[i].Pozicija != turnyras[j].Pozicija))
                        {
skaiciaus
                            PozicMas[i] += 1; //pridedamas vienetas prie poziciju
                                }
                        }
                    }
        int max = 0; // ieskoma didziausios reiksmes numeriukas
        for (int i = 0; i < turnCount; i++)
        {
            if (PozicMas[i] > max) //jei poziciju skaicius didesnis uz max,
kuris pradzioje yra lygus 0
                max = PozicMas[i]; // tuomet jis pakeiciamas i didesni
        }
        Console.WriteLine(); // Surasomi duomenys, kurie bus spausdinami
ekrane
        Console.WriteLine("-----");
        Console.WriteLine("Cempionas, kuris buvo naudotas universaliausiai:|
{0} ", turnyras[max-1].Cemp);
        Console.WriteLine("-----");
        Console.WriteLine("Pozicijos, kuriose buvo zaidziama: | {0}
", turnyras[max-1].Pozicija);
        for (int i = 0; i < turnCount; i++) //Renkamos skirtingos pozicijo.
Einama per visa masyva ir ieskoma, jei
        { // cempionato, turincio daugiausiai poziciju numeriukas
sutampa, tuomet tikrinama ar poziciju pavadinimai skiriasi
            if ((turnyras[max-1].Cemp == turnyras[i].Cemp) && (turnyras[max-
1].Pozicija != turnyras[i].Pozicija))
                Console.WriteLine("{0} ", turnyras[i].Pozicija); // Ir
rezultatas spausdinamas ekrane
            }
        Console.WriteLine(""); //| lenteles uzbaigimas, jis nera ciklo,
kadangi neaisku, kuris skaicius bus paskutinis ciklo
        Console.WriteLine("-----");
        Console.WriteLine("-----");
    } // funkcija nustatanti geriausia asmenini rezultata pagal KA
rodikli
    private static void GerAsmRez(Turnyras[] turnyras, int turnCount, out int
maxRod)
    {
        maxRod = 0; //Pradzioje rodiklis prilyginamas 0
        for (int i = 0; i < turnCount; i++)
        { //jei sunaikinimu skaicius su dalyvavimo skaiciu yra
didesnis uz maxRodikli
            if (turnyras[i].Sunaik + turnyras[i].Dalyvav > maxRod)
                maxRod = turnyras[i].Sunaik+turnyras[i].Dalyvav; //tuomet jis
priskiriamas maxRodikliui
            }
        }
        //failo nuskaitymas
        private static void ReadData(out Turnyras[] turnyras, out int turnCount)
//parsines sias reiksmes
        {
            turnCount = 0; //kiek is viso yra zaideju
            turnyras = new Turnyras[MaxZaid]; //maksimalus zaideju skaicius
            using (StreamReader reader = new StreamReader("Data.csv"))
            {

```

```

string line = null;
while (null != (line = reader.ReadLine()))
{
    string[] values = line.Split(','); // kaip bus skiriama eilute
    string vardas = values[0];
    string pavarde = values[1];
    string komanda = values[2];
    string pozicija = values[3];
    string cemp = values[4];
    int sunaik = int.Parse(values[5]);
    int zuvo = int.Parse(values[6]);
    int dalyvav = int.Parse(values[7]);
    Turnyras turnyr = new Turnyras(vardas, pavarde, komanda,
    pozicija, cemp, sunaik, zuvo, dalyvav);
    turnyras[turnCount++] = turnyr; //skaitliuka didina vienetu
}
}
}
}
}

```

### 1.3. Pradiniai duomenys ir rezultatai

#### 1.3.1 Pradiniai duomenys

Failas : Data.csv

```

Arnas,Sofauskas,FFBL,Top,Jarvan IV,3,0,5
Gailė,Pagalvytė,FFBL,Mid,Annie,1,1,5
Jonas,Kėdėpius,FFBL,AD,Ashe,3,1,5
Radvilė,Stalėiukaitė,FFBL,Support,Taric,0,0,8
Ignas,Lovauskas,FFBL,Jungle,Amumu,1,1,6
Indrė,Langaitė,Rainbow Dash,Top,Jarvan IV,8,0,10
Diana,Paveikslaitė,Rainbow Dash,Mid,Ahri,3,4,3
Dainius,Lentauskas,Rainbow Dash,AD,Vein,6,1,10
Vytenis,Dėpiauskas,Rainbow Dash,Support,Annie,2,1,17
Raigardas,Knygius,Rainbow Dash,Jungle,Lee Sin,1,2,11
Giedrius,Palangiauskas,FuriKuri,Top,Annie,3,1,10
Audrius,Dalgiauskas,FuriKuri,Mid,Ahri,6,1,8
Gintaras,Grėbliauskas,FuriKuri,AD,Corki,3,2,8
Jomantė,Dėklaitė,FuriKuri,Support,Sonna,0,4,13
Vladimiras,Lygiauskas,FuriKuri,Jungle,Jarvan IV,2,5,8
Audronė,Grindpiūtė,Girls United,Top,Jarvan IV,3,2,8
Simona,Tinklaitė,Girls United,Mid,Annie,5,0,8
Ieva,Pjūklaitė,Girls United,AD,Vein,8,0,8
Rasa,Plaktukaitė,Girls United,Support,Soraka,1,1,14
Raminta,Foteliūtė,Girls United,Jungle,Udyr,1,1,7

```

#### 1.3.2 Rezultatai

Failas : Top.csv

Failas : cmd.exe

## **2. Konteineris**

### ***2.1. Darbo užduotis***

### ***2.2. Programos tekstas***

### ***2.3. Pradiniai duomenys ir rezultatai***

### **3. Susieti objektų rinkiniai**

#### ***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. Sudėtingesnis konteineris**

### ***5.1. Darbo užduotis***

### ***5.2. Programos tekstas***

### ***5.3. Pradiniai duomenys ir rezultatai***