



**Kauno technologijos universitetas**  
Informatikos fakultetas

## **Objektinis programavimas I (P175B118)**

Laboratorinių darbų ataskaita

---

**Dovydas Lenkas IFF-4/3**

Studentas

**dr. Giedrius Ziberkas**

Dėstytojas

---

## TURINYS

<b>1. Duomenų klasė.....</b>	<b>3</b>
1.1. Darbo užduotis .....	3
1.2. Programos tekstas.....	3
1.3. Pradiniai duomenys ir rezultatai.....	9
1.4. Dėstytojo pastabos.....	11
<b>2. Skaičiavimų klasė .....</b>	<b>12</b>
2.1. Darbo užduotis .....	12
2.2. Programos tekstas.....	12
2.3. Pradiniai duomenys ir rezultatai.....	12
2.4. Dėstytojo pastabos.....	12
<b>3. Konteineris .....</b>	<b>13</b>
3.1. Darbo užduotis .....	13
3.2. Programos tekstas.....	13
3.3. Pradiniai duomenys ir rezultatai.....	13
3.4. Dėstytojo pastabos.....	13
<b>4. Teksto analizė ir redagavimas .....</b>	<b>14</b>
4.1. Darbo užduotis .....	14
4.2. Programos tekstas.....	14
4.3. Pradiniai duomenys ir rezultatai.....	14
4.4. Dėstytojo pastabos.....	14
<b>5. Paveldėjimas .....</b>	<b>15</b>
5.1. Darbo užduotis .....	15
5.2. Programos tekstas.....	15
5.3. Pradiniai duomenys ir rezultatai.....	15
5.4. Dėstytojo pastabos.....	15

# 1. Duomenų klasė

## 1.1. Darbo užduotis

**U1-17. Proto mūšis.** Studentų atstovybė organizuoja žaidimą „Protų mūšis“. Turite žaidimui paruoštus klausimus. Duomenų faile pateikiama ši informacija: tema, sudėtingumas, klausimo autorius, klausimo tekstas, 4 atsakymo variantai, teisingas atsakymas, balai.

- Raskite, kiek yra I, II ir III sudėtingumo lygio klausimų, rezultatus atspausdinkite ekrane.
- Sudarykite visų klausimų temų sąrašą, surašykite temas į failą „Temos.csv“.
- Sudarykite klausimų rinkinį, kuris turėtų tik skirtingų temų klausimus. Į rinkinį įtraukite ne daugiau kaip 4 klausimus. Į failą „Klausimai.csv“ įrašykite klausimų temas, tekstus ir balų skaičių.

## 1.2. Programos tekstas

### Info.cs

```
using System;
using System.Collections.Generic;

namespace Lab1.U17.Register
{
    class Info
    {
        /// <summary>
        /// Properties for the Info class with public getters and setters
        /// </summary>
        public string Theme { get; set; }
        public string Difficulty { get; set; }
        public string Author { get; set; }
        public string Question { get; set; }
        public List<string> Answers { get; set; }
        public string Correct { get; set; }
        public int Points { get; set; }

        /// <summary>
        /// Constructor to initialize an Info object with provided values
        /// </summary>
        /// <param name="theme">The theme of the question</param>
        /// <param name="difficulty">The difficulty level of the question (I, II,
        III)</param>
        /// <param name="author">The author of the question</param>
        /// <param name="question">The question text</param>
        /// <param name="answers">A list of possible answers</param>
        /// <param name="correctAnswer">The correct answer from the list of possible
        answers</param>
        /// <param name="points">The points awarded for answering the question
        correctly</param>
        public Info(string theme, string difficulty, string author, string question,
        List<string> answers, string correctAnswer, int points)
        {
            Theme = theme;
            Difficulty = difficulty;
            Author = author;
            Question = question;
        }
    }
}
```

```

        Answers = answers;
        Correct = correctAnswer;
        Points = points;
    }
}

```

## InOutUtils.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Text;

namespace Lab1.U17.Register
{
    static class InOutUtils
    {
        /// <summary>
        /// Read data from the specified CSV file and return a list of Info objects
        /// </summary>
        /// <param name="fileName">The path to the CSV file containing question
data</param>
        /// <returns>A list of Info objects representing the questions from the CSV
file</returns>
        public static List<Info> ReadData(string fileName)
        {
            List<Info> Data = new List<Info>();
            string[] Lines = File.ReadAllLines(fileName, Encoding.UTF8); // Read all lines
from the file

            foreach (string Line in Lines)
            {
                string[] Values = Line.Split(';'); // Split the line into values

                // Extract relevant information from the CSV format
                string theme = Values[0];
                string difficulty = Values[1];
                string author = Values[2];
                string question = Values[3];

                // Collect all possible answers in a list
                List<string> answers = new List<string>
                {
                    Values[4], Values[5], Values[6], Values[7]
                };

                string correctAnswer = Values[8]; // Correct answer
                int points = int.Parse(Values[9]); // Points for the question

                // Create an Info object and add it to the list
                Info info = new Info(theme, difficulty, author, question, answers,
correctAnswer, points);
                Data.Add(info);
            }

            return Data; // Return the populated list of Info objects
        }
    }
}

```

## TaskUtils.cs

```
using System;
using System.Collections.Generic;
using System.IO;
//using System.Linq;

namespace Lab1.U17.Register
{
    static class TaskUtils
    {
        /// <summary>
        /// Print a table to a specified text file with values from the data.
        /// </summary>
        /// <param name="data">List of Info objects containing question data.</param>
        /// <param name="fileName">The name of the file to write the output.</param>
        public static void printInitialData(List<Info> data, string fileName)
        {
            using (StreamWriter writer = new StreamWriter(fileName))
            {
                // Write the table header
                writer.WriteLine("|-----|");
                writer.WriteLine("| {0,-10} | {1,-15} | {2,-20} | {3,-35} | {4,-35} | {5,-20} | {6,-10} |", "Tema", "Sudėtingumas", "Autorius", "Klausimas", "Galimi atsakymai", "Teisingas atsakymas", "Taškai");
                writer.WriteLine("|-----|");

                // Write each entry in the data
                foreach (Info info in data)
                {
                    // Combine answers into a single string
                    string possibleAnswers = string.Join(" ", info.Answers);
                    writer.WriteLine("| {0,-10} | {1,-15} | {2,-20} | {3,-35} | {4,-35} | {5,-20} | {6,-10} |",
                        info.Theme,
                        info.Difficulty,
                        info.Author,
                        info.Question,
                        possibleAnswers,
                        info.Correct,
                        info.Points);
                }

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

            // Storage for printing
            private static int I; // Count of difficulty level I
            private static int II; // Count of difficulty level II
            private static int III; // Count of difficulty level III

            /// <summary>
            /// Count how many questions exist for each difficulty level.
            /// </summary>
            /// <param name="data">List of Info objects containing question data.</param>
        }
    }
}
```

```

public static void countDifficulty(List<Info> data)
{
    // Placeholder value
    I = 0;
    II = 0;
    III = 0;

    // Iterate through each Info object and count difficulties manually
    foreach (Info info in data)
    {
        if (info.Difficulty == "I")
        {
            I++;
        }
        else if (info.Difficulty == "II")
        {
            II++;
        }
        else if (info.Difficulty == "III")
        {
            III++;
        }
    }
}

/// <summary>
/// Print the counts of different difficulty questions
/// </summary>
public static void printDifficultyCount()
{
    Console.WriteLine("Skirtingo sudėtingumo klausimų skaičius:");
    Console.WriteLine($"I: {I}");
    Console.WriteLine($"II: {II}");
    Console.WriteLine($"III: {III}");
}

/// <summary>
/// Find unique themes from the list of Info objects.
/// </summary>
/// <param name="data">List of Info objects containing question data.</param>
/// <returns>A HashSet containing unique themes.</returns>
public static HashSet<string> uniqueThemeList(List<Info> data)
{
    HashSet<string> uniqueThemes = new HashSet<string>();

    // Add each theme to the HashSet to ensure uniqueness
    foreach (Info info in data)
    {
        uniqueThemes.Add(info.Theme);
    }

    return uniqueThemes; // Return the unique themes
}

/// <summary>
/// Write the unique themes to a specified CSV file.
/// </summary>
/// <param name="themes">HashSet containing unique themes.</param>
/// <param name="fileName">The name of the file to write the themes.</param>
public static void printUniqueThemes(HashSet<string> themes, string fileName)
{
    using (StreamWriter writer = new StreamWriter(fileName))
    {

```

```

        foreach (string theme in themes)
        {
            writer.WriteLine(theme); // Write each theme on a new line
        }
    }

    /// <summary>
    /// Create a packet of questions containing up to 4 unique themes.
    /// </summary>
    /// <param name="data">List of Info objects containing question data.</param>
    /// <returns>A List of Info objects representing the question packet.</returns>
    public static List<Info> questionPacket(List<Info> data)
    {
        HashSet<string> usedThemes = new HashSet<string>();
        List<Info> questionPacket = new List<Info>();

        foreach (Info info in data)
        {
            if (usedThemes.Count >= 4) break; // Stop if we have 4 unique themes
            if (!usedThemes.Contains(info.Theme))
            {
                questionPacket.Add(info);
                usedThemes.Add(info.Theme);
            }
        }

        return questionPacket;
    }

    /// <summary>
    /// Write the selected questions to a specified CSV file in table format.
    /// </summary>
    /// <param name="questions">List of Info objects representing the
questions.</param>
    /// <param name="fileName">The name of the file to write the questions.</param>
    public static void printQuestionPacket(List<Info> questions, string fileName)
    {
        using (StreamWriter writer = new StreamWriter(fileName))
        {
            // Write table header
            writer.WriteLine("-----|");
            writer.WriteLine("| {0,-10} | {1,-35} | {2,-50} | {3,-6} |", "Tema",
            "Klausimas", "Galimi atsakymai", "Taškai");
            writer.WriteLine("-----|");

            // Write each question with possible answers
            foreach (Info info in questions)
            {
                string possibleAnswers = string.Join(", ", info.Answers);
                writer.WriteLine("| {0,-10} | {1,-35} | {2,-50} | {3,-6} |",
                    info.Theme,
                    info.Question,
                    possibleAnswers,
                    info.Points);
            }

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

```

```
}  
}
```

## Program.cs

```
using System;  
using System.Collections.Generic;  
  
namespace Lab1.U17.Register  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            // Set console encoding to support special characters  
            Console.OutputEncoding = System.Text.Encoding.UTF8;  
  
            // Path to the input CSV file  
            string filePath = "data.csv";  
  
            // Read data from the CSV file into a list of Info objects  
            List<Info> data = InOutUtils.ReadData(filePath);  
  
            // Check if the data list is empty  
            if (data == null || data.Count == 0)  
            {  
                Console.WriteLine("Nėra duomenų.");  
                return; // Exit the program to avoid further processing  
            }  
  
            // Output data from the initial CSV file to a text file  
            TaskUtils.printInitialData(data, "poNuskaitymo.txt");  
  
            // Count the number of questions for each difficulty level  
            TaskUtils.countDifficulty(data);  
  
            // Print the counts of questions for each difficulty level  
            TaskUtils.printDifficultyCount();  
  
            // Find unique themes from the list of questions  
            HashSet<string> uniqueThemes = TaskUtils.uniqueThemeList(data);  
  
            // Write the unique themes to a CSV file  
            TaskUtils.printUniqueThemes(uniqueThemes, "Temos.csv");  
  
            // Create a packet of questions containing up to 4 questions with unique  
themes  
            List<Info> questionPacket = TaskUtils.questionPacket(data);  
  
            // Write the selected questions to a CSV file in the specified table format  
            TaskUtils.printQuestionPacket(questionPacket, "Klausimai.csv");  
        }  
    }  
}
```



### 1.3. Pradiniai duomenys ir rezultatai

#### data.csv (pradiniai duomenys)

Mokslas;I;Jonas Kazlauskas;Kokios spalvos yra  
 bananas?;Raudonos;Mėlynos;Geltonos;Žalios;Geltonos;3  
 Matematika;I;Ieva Petrauskaitė;Kiek yra  $1 + 0$ ?;0;1;2;3;1;3  
 Geografija;II;Petras Jonaitis;Kur yra saulė?;Danguje;Vandenyje;Medyje;Žemėje;Danguje;5  
 Istorija;III;Jonas Kazlauskas;Kas yra Lietuvos  
 sostinė?;Vilnius;Kaunas;Klaipėda;Šiauliai;Vilnius;7  
 Literatūra;II;Ieva Petrauskaitė;Kiek raidžių yra žodyje "katė"?;3;4;5;6;4;5  
 Sportas;I;Petras Jonaitis;Kiek kojų turi žmogus?;2;3;4;5;2;3  
 Muzika;II;Jonas Kazlauskas;Kiek stygų turi gitara?;5;6;7;8;6;5  
 Mokslas;III;Petras Jonaitis;Kuris iš jų karščiausias?;Ugnis;Sniegas;Ledas;Vanduo;Ugnis;7  
 Matematika;I;Ieva Petrauskaitė;Kiek yra  $2 - 1$ ?;0;1;2;3;1;3

#### poNuskaitymo.txt (atspausdinti nuskaityti duomenys)

Tema	Sudėtingumas	Autorius	Klausimas	Galimi atsakymai	Teisingas atsakymas	Taškai
Mokslas	I	Jonas Kazlauskas	Kokios spalvos yra bananas?	Raudonos, Mėlynos, Geltonos, Žalios	Geltonos	3
Matematika	I	Ieva Petrauskaitė	Kiek yra $1 + 0$ ?	0, 1, 2, 3	1	3
Geografija	II	Petras Jonaitis	Kur yra saulė?	Danguje, Vandenyje, Medyje, Žemėje	Danguje	5
Istorija	III	Jonas Kazlauskas	Kas yra Lietuvos sostinė?	Vilnius, Kaunas, Klaipėda, Šiauliai	Vilnius	7
Literatūra	II	Ieva Petrauskaitė	Kiek raidžių yra žodyje "katė"?	3, 4, 5, 6	4	5
Sportas	I	Petras Jonaitis	Kiek kojų turi žmogus?	2, 3, 4, 5	2	3
Muzika	II	Jonas Kazlauskas	Kiek stygų turi gitara?	5, 6, 7, 8	6	5
Mokslas	III	Petras Jonaitis	Kuris iš jų karščiausias?	Ugnis, Sniegas, Ledas, Vanduo	Ugnis	7
Matematika	I	Ieva Petrauskaitė	Kiek yra $2 - 1$ ?	0, 1, 2, 3	1	3

#### Kiek yra I, II ir III sudėtingumo lygio klausimų, rezultatai atspausdinti ekrane.

Skirtingo sudėtingumo klausimų skaičius:

I: 4

II: 3

III: 2

#### Temos.csv (visų klausimų temų sąrašas)

Mokslas  
 Matematika  
 Geografija  
 Istorija  
 Literatūra  
 Sportas  
 Muzika

**Klausimai.csv (skirting temų klausimų rinkinys. Iki keturių klausimų)**

Tema	Klausimas	Galimi atsakymai	Taškai
Mokslas	Kokios spalvos yra bananas?	Raudonos, Mėlynos, Geltonos, Žalios	3
Matematika	Kiek yra $1 + 0$ ?	0, 1, 2, 3	3
Geografija	Kur yra saulė?	Danguje, Vandenyje, Medyje, Žemėje	5
Istorija	Kas yra Lietuvos sostinė?	Vilnius, Kaunas, Klaipėda, Šiauliai	7

**data.csv (antras testas)**

Mokslas;I;Jonas Kazlauskas;Kokios spalvos yra  
bananas?;Raudonos;Mėlynos;Geltonos;Žalios;Geltonos;3

**poNuskaitymo.txt (atspausdinti nuskaityti duomenys)**

Tema	Sudėtingumas	Autorius	Klausimas	Galimi atsakymai	Teisingas atsakymas	Taškai
Mokslas	I	Jonas Kazlauskas	Kokios spalvos yra bananas?	Raudonos, Mėlynos, Geltonos, Žalios	Geltonos	3

**Kiek yra I, II ir III sudėtingumo lygio klausimų, rezultatai atspausdinti ekrane.**

Skirtingo sudėtingumo klausimų skaičius:

I: 1

II: 0

III: 0

**Temos.csv (visų klausimų temų sąrašas)**

Mokslas

**Klausimai.csv (skirting temų klausimų rinkinys. Iki keturių klausimų)**

Tema	Klausimas	Galimi atsakymai	Taškai
Mokslas	Kokios spalvos yra bananas?	Raudonos, Mėlynos, Geltonos, Žalios	3

**1.4. Dėstytojo pastabos**

## **2. Skaičiavimų klasė**

### **2.1. Darbo užduotis**

### **2.2. Programos tekstas**

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

### **2.4. Dėstytojo pastabos**

### **3. Konteineris**

#### **3.1. Darbo užduotis**

#### **3.2. Programos tekstas**

#### **3.3. Pradiniai duomenys ir rezultatai**

#### **3.4. Dėstytojo pastabos**

## **4. Teksto analizė ir redagavimas**

### **4.1. Darbo užduotis**

### **4.2. Programos tekstas**

### **4.3. Pradiniai duomenys ir rezultatai**

### **4.4. Dėstytojo pastabos**

## **5. Paveldėjimas**

### **5.1. Darbo užduotis**

### **5.2. Programos tekstas**

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

### **5.4. Dėstytojo pastabos**