

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 2 з дисципліни  
«Основи програмування 2. Модульне програмування»

«Класи та об'єкти»

Варіант 35

Виконав студент ІП-15, Шабанов Метін Шаміль огли  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

Київ 2022

## Лабораторна робота 2

### Класи та об'єкти

#### Варіант 35

Завдання:

35. Розробити клас, який є абстракцією тексту та підтримує операції додавання рядка до тексту і визначення кількості рядків заданої довжини. Створити масив об'єктів даного класу. Доповнити декілька текстів новими рядками. Визначити текст з найбільшою кількістю рядків заданої довжини.

Виконання:

C#:




```
C# Main.cs × C# Text.cs × C# OperateTextObjects.cs × C# Printer.cs ×
1      using System;
2      using System.Collections.Generic;
3
4      namespace Lab3_ClassesAndObjects
5      {
6          class Program
7          {
8              static void Main(string[] args)
9              {
10                 Text[] arrOfTexts = OperateTextObjects.CreateArrOfTexts();
11                 Printer.PrintTexts(arrOfTexts);
12                 OperateTextObjects.AddLinesToTexts(arrOfTexts);
13                 Printer.PrintTexts(arrOfTexts);
14                 Text textWithMostLines = OperateTextObjects.FindTextWithMaxLines(arrOfTexts);
15                 Printer.PrintSpecText(textWithMostLines);
16             }
17         }
18     }
```

```

C# Main.cs × C# Text.cs × C# OperateTextObjects.cs × C# Printer.cs ×
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4
5  namespace Lab3_ClassesAndObjects
6  {
7      13 usages 1 exposing API
8      public class Text
9      {
10
11         1 usage
12         public Text()
13         {
14             Console.WriteLine("Adding new text...");
15             Console.WriteLine("Press F1 to stop adding data.");
16
17             while (Console.ReadKey().Key != ConsoleKey.F1)
18             {
19                 Console.Write("\b");
20                 _text = _text.Append(Console.ReadLine()).ToList();
21             }
22
23             2 usages
24             public List<string> GetText
25             {
26                 get {return _text;}
27             }
28
29             1 usage
30             public void AddLine(string newLine)
31             {
32                 _text = _text.Append(newLine).ToList();
33             }
34
35             1 usage
36             public int FindLinesByLength(int neededLength)
37             {
38                 int counter = 0;
39
40                 for (int i = 0; i < _text.Count; i++)
41                 {
42                     if (_text[i].Length == neededLength)
43                         counter++;
44                 }
45
46                 return counter;
47             }
48         }
49     }
50 }



```

```

C# Main.cs × C# Text.cs × C# OperateTextObjects.cs × C# Printer.cs ×
1      using System;
2      using System.Text.RegularExpressions;
3
4      namespace Lab3_ClassesAndObjects
5      {
6           3 usages
7          public class OperateTextObjects
8          {
9               1 usage
10             public static Text[] CreateArrOfTexts()
11             {
12                 Console.Write("Enter the length of array: ");
13                 int length = _checkForNumber(checkedNum: Console.ReadLine());
14                 Text[] texts = new Text[length];
15
16                 for (int i = 0; i < texts.Length; i++)
17                     texts[i] = new Text();
18
19                 return texts;
20             }
21
22              1 usage
23             public static void AddLinesToTexts(Text[] texts)
24             {
25                 Console.WriteLine("Now start adding lines, press F1 if you don't want to.");
26                 while (Console.ReadKey().Key != ConsoleKey.F1)
27                 {
28                     _addLinesToSpecText(texts);
29                     Console.WriteLine("Press F1 to stop adding lines.");
30                 }
31             }
32         }
33     }

```

```

C# Main.cs × C# Text.cs × C# OperateTextObjects.cs × C# Printer.cs ×
30     public static Text FindTextWithMaxLines(Text[] texts)
31     {
32         Console.Write("Enter the length of line: ");
33         int lengthOfLine = _checkForNumber(checkedNum: Console.ReadLine());
34         int[] counts = _findLines(texts, lengthOfLine);
35
36         while (!_isEmpty(counts))
37         {
38             Console.WriteLine("There are no lines of such length, try again: ");
39             lengthOfLine = _checkForNumber(checkedNum: Console.ReadLine());
40             counts = _findLines(texts, lengthOfLine);
41         }
42
43         int maxLinesIndex = _findIndexOfMax(counts);
44         return texts[maxLinesIndex];
45     }
46
47      5 usages
48     private static int _checkForNumber(string checkedNum)
49     {
50         Regex numPattern = new Regex(pattern: @"^\d$");
51         while (checkedNum == null || !numPattern.IsMatch(checkedNum))
52         {
53             Console.WriteLine("Not a number, try again: ");
54             checkedNum = Console.ReadLine();
55         }
56
57         return int.Parse(checkedNum);
58     }
59
60      1 usage
61     private static void _addLinesToSpecText(Text[] texts)
62     {
63         Console.WriteLine("\nEnter the index of the text to add the line: ");

```

```

C# Main.cs × C# Text.cs × C# OperateTextObjects.cs × C# Printer.cs ×
62     int index = _checkForNumber(checkedNum: Console.ReadLine());
63     while (index > texts.Length - 1)
64     {
65         Console.WriteLine("Too big, try again: ");
66         index = _checkForNumber(checkedNum: Console.ReadLine());
67     }
68
69     texts[index].AddLine(Console.ReadLine());
70 }
71

```

1 usage

```

72 private static int _findIndexOfMax(int[] arr)
73 {
74     int index = 0;
75     for (int i = 0; i < arr.Length; i++)
76         if (arr[i] > arr[index])
77             index = i;
78
79     return index;
80 }
81

```

2 usages

```

82 private static int[] _findLines(Text[] texts, int lengthOfLine)
83 {
84     int[] counts = new int[texts.Length];
85
86     for (int i = 0; i < texts.Length; i++)
87         counts[i] = texts[i].FindLinesByLength(lengthOfLine);
88
89     return counts;
90 }
91

```

1 usage

```

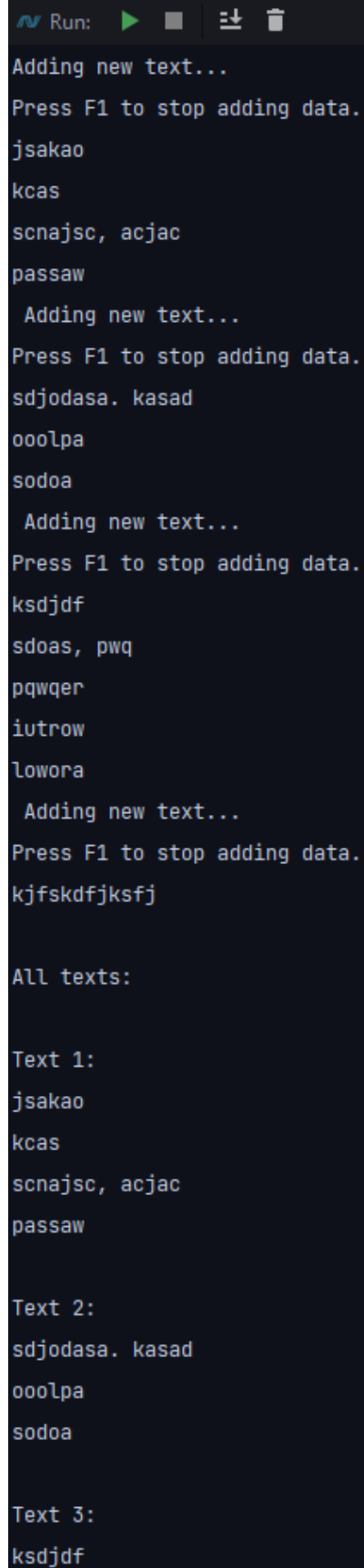
92 private static bool _isEmpty(int[] arr)
93 {
94     int counter = 0;
95     for (int i = 0; i < arr.Length; i++)
96         if (arr[i] != 0)
97             counter++;
98
99     if (counter == 0)
100         return true;
101
102     return false;
103 }
104 }
105 }

```

```
4 namespace Lab3_ClassesAndObjects
5
6 public class Printer
7 {
8     public static void PrintTexts(Text[] texts)
9     {
10         Console.WriteLine("\nAll texts:");
11
12         for (int i = 0; i < texts.Length; i++)
13         {
14             Console.WriteLine($"Text {i + 1}:");
15             List<string> lines = texts[i].GetText;
16             for (int j = 0; j < lines.Count; j++)
17             {
18                 Console.WriteLine(lines[j]);
19             }
20         }
21     }
22
23     public static void PrintSpecText(Text text)
24     {
25         Console.WriteLine("\nThe text with biggest number of lines with given length:");
26         List<string> lines = text.GetText;
27
28         for (int i = 0; i < lines.Count; i++)
29         {
30             Console.WriteLine(lines[i]);
31         }
32     }
33 }
```

## Тестування:

C#:



```
Run: [play] [stop] [file icons]
Adding new text...
Press F1 to stop adding data.
jsakao
kcas
scnajsc, acjac
passaw
    Adding new text...
Press F1 to stop adding data.
sdjodasa. kasad
oolpa
sodoa
    Adding new text...
Press F1 to stop adding data.
ksdjdf
sdoas, pwq
pqwqer
iutrow
lowora
    Adding new text...
Press F1 to stop adding data.
kjfskdfjksfj

All texts:

Text 1:
jsakao
kcas
scnajsc, acjac
passaw

Text 2:
sdjodasa. kasad
oolpa
sodoa

Text 3:
ksdjdf
```

sdoas, pwq

pqwqer

iutrow

lowora

Text 4:

kjfskdfjksfj

Now start adding lines, press F1 if you don't want to.

Enter the index of the text to add the line: 2

jasdha

Press F1 to stop adding lines.

Enter the index of the text to add the line: 4

Too big, try again: 3

oooooooooooooooooooooooooooo

Press F1 to stop adding lines.

All texts:

Text 1:

jsakao

kcas

scnajsc, acjac

passaw

Text 2:

sdjodasa. kasad

ooopa

sodoa

Text 3:

ksdjdf

sdoas, pwq

pqwqer

iutrow

lowora

jasdha

Text 4:

kjfskdfjksfj

oooooooooooooooooooooooooooo

Enter the length of line: 6

The text with biggest number of lines with given length:

ksdjdf

sdoas, pwq

pqwqer

iutrow

lowora

jasdha

Program finished with exit code 0