

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Алтайский государственный университет»
Институт математики и информационных технологий
Кафедра информатики
Курс «Программная инженерия»

Отчет о лабораторной работе №6
Создание классов

Выполнил:
Сердюк Е. И.
группа 4746

Барнаул 2020

Вариант 9

Задача 1

9. Разработать класс для представления объекта строки. Определить конструктор с одним параметром целого типа – длина строки, который можно использовать как конструктор умолчания. Определить конструктор, который копирует в новую строку n первых символов другой строки и который можно использовать как конструктор копирования. Определить деструктор. Определить преобразования из вещественного числа в строку, представляющую это число, и из строки в вещественное число

Исходный код

```
using System;
public class ObjectString
{
    private int length; // Length of the string
    public char[] str; // String

    public int Length // Properties
    {
        get => length;
        set => length = value;
    }

    public ObjectString(int length = 6) // Default constructor
    {
        this.length = length;
        str = new char[length];

        // Manual input
        // do // If an invalid variable is entered -> the input starts again
        // {
        //     try
        //     {
        //         Console.Write("Input the string:\n");
        //         for (int i = 0; i < length; i++)
        //         {
        //             Console.Write("String[{0}]: ", i);
        //             str[i] = Convert.ToChar(Console.ReadLine());
        //         }
        //     }
        //     catch
        //     {
        //         Console.Write("Error, the wrong type of variable\n");
        //         continue;
        //     }
        //     flag = true;
        // } while (flag != true);

        // Random input
        Random rnd = new Random();
        for (int i = 0; i < length; i++)
        {
            str[i] = Convert.ToChar(rnd.Next('a', 'a' + 26)); // Random character from
the english alphabet
        }

    }

    // Copy constructor
    public ObjectString(ObjectString objectStr, int n = 5)
```

```

{
    length = objectStr.length;
    str = new char[length];
    for (int i = 0; i < n; i++)
    {
        str[i] = objectStr.str[i];
    }
}

// Destructor
~ObjectString() { }

// Transforming ObjectString to Double
public double[] ToDouble(ObjectString objstr)
{
    double[] doubleStr; // new double string
    doubleStr = new double[objstr.length];
    for (int i = 0; i < objstr.length; i++)
    {
        doubleStr[i] = str[i];
    }
    return doubleStr;
}

// Transforming Double to ObjectString
public ObjectString FromDouble(double[] doubleStr)
{
    ObjectString String = new ObjectString(length);
    for (int i = 0; i < doubleStr.Length; i++)
    {
        String.str[i] = (char)doubleStr[i];
    }
    return String;
}

// Output method
public void output()
{
    Console.WriteLine();
    for (int i = 0; i < length; i++)
    {
        Console.Write("{0}", str[i]);
    }
}
}
class Program
{
    static void Main()
    {
        // create objString
        ObjectString str1 = new ObjectString();
        ObjectString str2 = new ObjectString(str1);

        // output the strings
        Console.Write("\nstr 1: ");
        str1.output();

        Console.Write("\nstr 2: ");
        str2.output();

        // double array init
        double[] doubleStr;
        doubleStr = new double[str1.Length]; // Especially for str1

        // transformed ObjString str1

```

```
doubleStr = str1.ToDouble(str1);

// output double array
Console.WriteLine("\nstr1 to Double: ");
for (int i = 0; i < doubleStr.Length; i++)
    Console.Write("{0} ", doubleStr[i]);

ObjectString str3 = new ObjectString();

// transformed from double back to ObjString
str3 = str3.FromDouble(doubleStr);
Console.Write("\nDouble to str3: ");
str3.output();
    }
}
```