Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа №4

По дисциплине

“Объектно-ориентированное программирование”

# Тема “Класс Array. Строки. Класс String”

Вариант 15

Выполнил работу

студент группы РИС-19-1б

Миннахметов Э.Ю.

Проверила

доцент кафедры ИТАС

Викентьева О.Л.

Работу выполнил:

Пермь 2020

**1 ПОСТАНОВКА ЗАДАЧИ**

**Постановка задачи 1.**

1. Создать динамический массив (одномерный, двумерный, рваный) из элементов заданного типа. При заполнении массива использовать 2 способа (ручной и с помощью ДСЧ).
2. Массив вывести на печать.
3. Выполнить операции с массивом, указанные в варианте, используя, по возможности, методы класса Array.
4. Результаты обработки вывести на печать.

|  |  |  |
| --- | --- | --- |
| **Тип массива** | **Тип элементов** | **Операция** |
| Рваный | int | Удалить все строки, в которых есть не менее двух нулей. |

**Постановка задачи 2.**

1. Ввести строку символов с клавиатуры. Строка состоит из слов, разделенных пробелами (пробелов может быть несколько) и знаками препинания (, ;:). В строке может быть несколько предложений, в конце каждого предложения стоит знак препинания (.!?).
2. Выполнить обработку строки в соответствии с вариантом.
3. Результаты обработки вывести на печать.

|  |
| --- |
| **Задание** |
| Удалить из строки все слова, которые начинаются и заканчиваются на один и тот же символ. |

**2 ОПИСАНИЕ ЭТАПА ПРОЕКТИРОВАНИЯ**

**2.1 Задача 1**

Интерфейс программы:

* GetArray – создание рваного массива;
* ReadNum, RandNum – получение числа с ввода либо генерация случайного;
* Output – вывод массива;
* DeleteRows – удаление строк с двумя нулями;
* RemoveRow – вспомогательная функция для удаления конкретной строки.

**2.2 Задача 2**

Интерфейс программы:

* Proccessing – удаление слов, начинающихся и заканчивающихся на одну и ту же букву, из предложения.

**3 ИСХОДНЫЙ КОД ПРОГРАММЫ**

**3.1 Задача 1**

using System;

// Вариант 15

namespace Task1

{

delegate void GetNumber(out int number, char simbol, int i = -1, int j = -1);

delegate bool IsValidate(int x, int top);

public class Program

{

private const char c\_cN = 'N';

private const char c\_cM = 'M';

public const char c\_cA = 'A';

private const int c\_iCountZero = 2;

private const int c\_iMinArray = 5;

private const int c\_iMaxArray = 15;

private const int c\_iMinNumber = 0;

private const int c\_iMaxNumber = 9;

public const int c\_iMaxInt = 2147483647;

private const string c\_sElem = "{0} ";

private const string c\_sReadNumber = "Введите {0}: ";

private const string c\_sReadNumberA = "Введите {0}({1},{2}): ";

private const string c\_sIncorrectValue = "Некорректное значение!";

private const string c\_sTask = "Удаление строк с 2 и более нулями";

private const string c\_sContinue = "Продолжить? (д - да, другое - выход)";

private const string c\_sExit = "Спасибо за работу!";

private const string c\_sGetMode =

"Введите способ получения чисел (1 - ввод, - 2 случайное): ";

private static Random s\_rand = new Random();

private static IsValidate IsValid = (x, top) => (x >= 0 && x <= top);

static void Main(string[] args)

{

do

{

int[][] array = GetArray(GetMode());

Output(array);

Console.WriteLine(c\_sTask);

DeleteRows(ref array);

Output(array);

Console.WriteLine(c\_sContinue);

}

while (Console.ReadLine() == "д");

Console.WriteLine(c\_sExit);

Console.ReadKey();

}

static int[][] GetArray(GetNumber GetNum)

{

GetValid(out int n, GetNum, c\_cN, c\_iMaxInt);

int[][] array = new int[n][];

for(int i = 0; i < n; ++i)

{

GetValid(out int m, GetNum, c\_cM, c\_iMaxInt);

array[i] = new int[m];

for (int j = 0; j < m; ++j)

{

GetNum(out int a, c\_cA, i, j);

array[i][j] = a;

}

}

return array;

}

public static void ReadNum(out int number, char simbol, int i, int j)

{

number = 0;

for (bool flag = false; !flag;)

{

if (i == -1)

Console.Write(c\_sReadNumber, simbol);

else

Console.Write(c\_sReadNumberA, simbol, i, j);

string sNum = Console.ReadLine();

flag = int.TryParse(sNum, out number);

if (!flag)

IncorrectValue();

}

}

public static void RandNum(out int number, char simbol, int i, int j)

{

if (simbol == c\_cN || simbol == c\_cM)

number = s\_rand.Next(c\_iMinArray, c\_iMaxArray);

else

number = s\_rand.Next(c\_iMinNumber, c\_iMaxNumber);

}

private static GetNumber GetMode()

{

string key = "";

do

{

Console.Write(c\_sGetMode);

key = Console.ReadLine();

} while (key != "1" && key != "2");

if (key == "1")

return ReadNum;

else

return RandNum;

}

public static void IncorrectValue()

{

Console.WriteLine(c\_sIncorrectValue);

}

private static void GetValid(out int number, GetNumber GetNum, char simbol, int top)

{

while (true)

{

GetNum(out number, simbol);

if (IsValid(number, top))

break;

else

IncorrectValue();

}

}

private static void Output(int[][] array)

{

for (int i = 0, n = array.Length; i < n; ++i)

{

for (int j = 0, k = array[i].Length; j < k; ++j)

Console.Write(c\_sElem, array[i][j]);

Console.WriteLine();

}

Console.WriteLine();

}

public static void DeleteRows(ref int[][] array)

{

for (int i = 0, n = array.Length; i < n; ++i)

{

int z = 0;

for (int j = 0, k = array[i].Length; j < k && z != c\_iCountZero; ++j)

if (array[i][j] == 0)

++z;

if (z == c\_iCountZero)

{

RemoveRow(ref array, i--);

--n;

}

}

}

private static void RemoveRow(ref int[][] array, int i)

{

int n = array.Length - 1;

int[][] temp = array;

array = new int[n][];

for (int j = 0; j < i; ++j)

array[j] = temp[j];

for (int j = i; j < n; ++j)

array[j] = temp[j + 1];

}

}

}

**3.2 Задача 2**

using System;

using System.Text;

using System.Text.RegularExpressions;

namespace Task2

{

public class Program

{

private const string c\_sEnter = "Введите строку";

private const string c\_sResult = "Результат выполнения программы";

private const string c\_sContinue = "Продолжить? (д - да, другое - выход)";

private const string c\_sExit = "Спасибо за работу!";

private static readonly Regex regex = new Regex(@"\b(\w+)");

public static void Main(string[] args)

{

do

{

Console.WriteLine(c\_sEnter);

string str = Console.ReadLine();

Console.WriteLine(c\_sResult);

string result = Proccessing(str);

Console.WriteLine(result);

Console.WriteLine(c\_sContinue);

}

while (Console.ReadLine() == "д");

Console.WriteLine(c\_sExit);

Console.ReadKey();

}

public static string Proccessing(string str)

{

StringBuilder strBuild = new StringBuilder(str);

MatchCollection matches = regex.Matches(str);

foreach(Match match in matches)

{

string word = match.Value;

if (word[0] == word[word.Length - 1])

strBuild.Replace(word, "");

}

return strBuild.ToString();

}

}

}

**3.3 Unit-тест для Task1**

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace Lab6.Tests

{

[TestClass]

public class Task1

{

private static int[][] array = new int[][]{

new int[] { 23, 345, 789, 8, 0, 453, 0, 345, 0 },

new int[] { 45, 45, 21, 5 },

new int[] { 0, 0, 0, 0, 0, 0, 0, 0, 0 },

new int[] { 45, 0, 45, 21, 5 },

new int[] { 45, 45, 21, 5 },

new int[] { 0, 45, 21, 0 }

};

private readonly int[][] result = new int[][]{

new int[] { 45, 45, 21, 5 },

new int[] { 45, 0, 45, 21, 5 },

new int[] { 45, 45, 21, 5 }

};

[TestMethod]

public void Test\_DeleteRows()

{

global::Task1.Program.DeleteRows(ref array);

Assert.AreEqual(array.Length, result.Length);

for(int i = 0, n = array.Length; i < n; ++i)

{

Assert.AreEqual(array[i].Length, result[i].Length);

for (int j = 0, m = array[i].Length; j < m; ++j)

Assert.AreEqual(array[i][j], result[i][j]);

}

}

}

}

**3.3 Unit-тест для Task2**

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace Lab6.Tests

{

[TestClass]

public class Task2

{

private const string test

= "hello world! меня зовут Эльдар. 453 564орпап алвпвалпр345 - helloh world меням зовут ЭльдарЭ Эльдарэ 4534 564орпап алвлпр345Ф";

private const string result

= "hello world! меня зовут Эльдар. 453 564орпап алвпвалпр345 - world зовут Эльдарэ 564орпап алвлпр345Ф";

[TestMethod]

public void Test\_Proccessing()

{

Assert.AreEqual(global::Task2.Program.Proccessing(test), result);

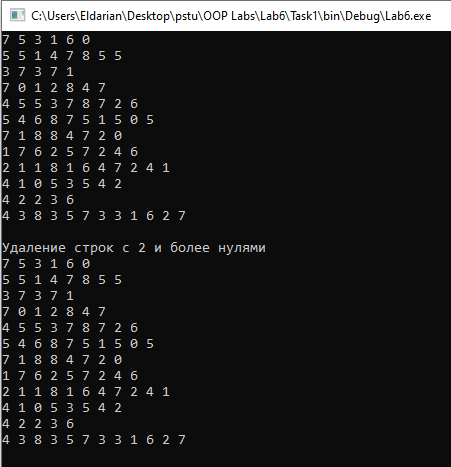
}

}

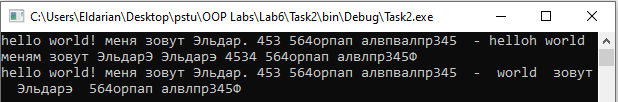
}

**4 СКРИНШОТ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

**4.1 Задача 1**

****

**4.2 Задача 2**

****