Задание.

Используя возможности VS, разработайте и протестируйте программу сортировки методом Шелла. При этом для ввода и вывода данных, а также запуска и завершения работы программы используйте стандартные элементы пользовательского интерфейса.

Создаем Модульные тесты:

Module1 – Проверка сортировки Шелла

**План Тестирования**

|  |  |
| --- | --- |
| Вход | Выход |
| Module1.doSort(""), | “” |
| Module1.doSort("32 1 4") | 1 4 32 |
| Module1.doSort("32 1 asd") | 1 32 |

В результате должны получиться следующие методы:

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

textBox1.Text = Module1.doSort(textBox1.Text);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

//textBox1.Text = Module1.doSort(textBox1.Text);

}

}

public

class Module1

{

public static String doSort(String s)

{

if (s == "")

{

return "";

}

s = Regex.Replace(s, @"[^0-9,^/\s/g]+", ""); // заменяем все не числа на ""

Char delimiter = ' ';

String[] substrings = s.Split(delimiter);

int[] array4 = new int[substrings.Length];

int count = 0;

for (int i = 0; i < substrings.Length; i++)

{

if (substrings[i] != "")

array4[i - count] = Int32.Parse(substrings[i]);

else

{

Array.Resize(ref array4, array4.Length - 1);

count++;

}

}

int step = array4.Length / 2;

while (step > 0)

{

for (int i = 0; i < (array4.Length - step); i++)

{

int j = i;

while (j >= 0 && array4[j] > array4[j + step])

{

int temp = array4[j];

array4[j] = array4[j + step];

array4[j + step] = temp;

j--;

}

}

step = step / 2;

}

String result = "";

foreach (var num in array4)

{

result += num + " ";

}

return result;

}

}

}

Код Тестого модуля

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using lab1;

namespace UnitTestLab1

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1()

{

Assert.AreEqual(Module1.doSort(""), "");

Assert.AreEqual(Module1.doSort("32 1 4"), "1 4 32 ");

Assert.AreEqual(Module1.doSort("32sd 1 4sd sda"), "1 4 32 ");

Assert.AreEqual(Module1.doSort("32 1 asd"), "1 32 ");

}

}

}