Министерство науки и высшего образования РФ

ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

**ОБЪЕКНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ДИЗАЙН**

Отчёт по лабораторной работе №1

# «Работа с системами контроля версий»

Выполнил:

Студент: 2 курса

группы 20-КБ-ПР2

Филимонов Р.С.

Проверил:

ассистент каф. ИСП

Степанова Е.В.

г. Краснодар 2022

Цель: Изучить подход к автоматизации процесса тестирования с помощью среды разработки Microsoft Visual Studio и фреймворка NUnit.

Задания:

1) Создать класс, реализующий один из алгоритмов (в соответствии с вариантом задания из п.5).

2) Создать проект для автоматизации модульного тестирования с необходимым набором тестов.

3) Запустить проект тестирования и проверить результаты работы.

4) Составить отчет о результатах проведенного тестирования.

Вариант задания: 3

Код программы:

Класс:

public class InsertionSort

{

int[] sort = new int[6];

public InsertionSort(int[] arr)

{

sort = arr;

}

public int[] Sort()

{

if (sort != null)

{

for (int i = 1; i < sort.Length; i++)

for (int j = i; j > 0 && sort[j - 1] > sort[j]; j--) // пока j>0 и элемент j-1 > j, x-массив int

(sort[j - 1], sort[j]) = (sort[j], sort[j - 1]); // меняем местами элементы j и j-1

return sort;

}

else

{

throw new ArgumentException("Массив пустой");

}

}

public int[] ShowArray()

{

return sort;

}

}

Текст модульных тестов:

public class Tests

{

[Test]

public void Test1()

{

int[] test = { 3, 2, 1, 4, 6, 5 };

InsertionSort sort = new InsertionSort(test);

sort.Sort();

int[] extend = { 1, 2, 3, 4, 5, 6 };

Assert.AreEqual(extend, sort.ShowArray());

}

[Test]

public void TestNull()

{

int[] test = null;

InsertionSort sort = new InsertionSort(test);

Assert.Throws<ArgumentException>(() => { test = sort.Sort(); });

}

[Test]

public void TestSubZero()

{

int[] test = { -3, -2, -1, -4, -6, -5 };

InsertionSort sort = new InsertionSort(test);

sort.Sort();

int[] extend = { -6, -5, -4, -3, -2, -1 };

Assert.AreEqual(extend, sort.ShowArray());

}

[Test]

public void TestPlusMinus()

{

int[] test = { -1, -3, -2, 3, 1, 2 };

InsertionSort sort = new InsertionSort(test);

sort.Sort();

int[] extend = { -3, -2, -1, 1, 2, 3 };

Assert.AreEqual(extend, sort.ShowArray());

}

[Test]

public void TestOnlyZero()

{

int[] test = { 0, 0, 0, 0, 0, 0 };

InsertionSort sort = new InsertionSort(test);

sort.Sort();

int[] extend = { 0, 0, 0, 0, 0, 0 };

Assert.AreEqual(extend, sort.ShowArray());

}

}

Результат работы:

