Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

**ТЕСТИРОВАНИЕ И ОТЛАДКА  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Отчет по лабораторной работе №3

«Тестирование методом белого ящика»

Выполнил:

студент 3 курса

группы 19-КБ-ПР1

Кравцов Олег Юрьевич

Проверил:

ст. преп. каф. ИСП

А. Г. Волик

Краснодар

2021

**1 Цель работы**

Изучить подход к тестированию методом белого ящика.

**2 Задание**

1) Создать класс (в соответствии с вариантом задания из п.5), реализующий обработку с текстовой информации.

2) Построить управляющий граф программы (УГП).

3) Используя знания о структуре программы определить набор тестов для тестирования методом белого ящика.

4) Указать какие пути реализуются (проверяются) тестом (см. таблица 3.1).

5) Указать какие критерии покрываются путем (см. таблица 3.2).

6) Оценить качество полученных тестовых данных с точки зрения покрытия кода тестами.

7) Протестировать метод с использованием средств автоматизации тестирования.

8) Составить отчет о результатах проведенного тестирования.

**3 Вариант задания**

Вариант №12.

Создать класс, реализующий метод для замены всех вхождений символа в строке

**4 Текст программы**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

namespace Лр\_\_\_3\_\_ТОПО\_Кравцов\_19\_КБ\_ПР1\_

{

public static class Lr\_3

{

public static void Main(string[]args)

{

}

public static string Sep(string word, string letter, string replace)

{

if (word == "")//1

return "";//2

char[] wordarr = word.ToArray();//3

for (int i = 0;//4

i < word.Length;//5

i++)//6

{

if (wordarr[i] == Convert.ToChar(letter))//7

{

wordarr[i] = Convert.ToChar(replace);//8

}

}

return new string(wordarr);//9

}

}

}

**5 Текст модульных тестов**

using NUnit.Framework;

namespace TestProject1

{

public class Tests

{

[Test]

public void CheckEmptywordTest()

{

string word = "";

string letter = "в";

string replace = "ю";

string actually = Лр\_\_\_3\_\_ТОПО\_Кравцов\_19\_КБ\_ПР1\_.Lr\_3.Sep(word, letter, replace);

string expect = "";

Assert.AreEqual(actually, expect);

}

[Test]

public void CheckLetterTest()

{

string word = "Приветствую!";

string letter = "в";

string replace = "ю";

string actually = Лр\_\_\_3\_\_ТОПО\_Кравцов\_19\_КБ\_ПР1\_.Lr\_3.Sep(word, letter, replace);

string expect = "Приюетстюую!";

Assert.AreEqual(actually, expect);

}

[Test]

public void CheckLetterErrorTest()

{

string word = "Приветствую!";

string letter = "в";

string replace = "ю";

string actually = Лр\_\_\_3\_\_ТОПО\_Кравцов\_19\_КБ\_ПР1\_.Lr\_3.Sep(word, letter, replace);

string expect = "Приюетствую!";

Assert.AreEqual(actually, expect);

}

[Test]

public void CheckSymbolTest()

{

string word = "Привет ствую!";

string letter = " ";

string replace = "ю";

string actually = Лр\_\_\_3\_\_ТОПО\_Кравцов\_19\_КБ\_ПР1\_.Lr\_3.Sep(word, letter, replace);

string expect = "Приветююствую!";

Assert.AreEqual(actually, expect);

}

}

}

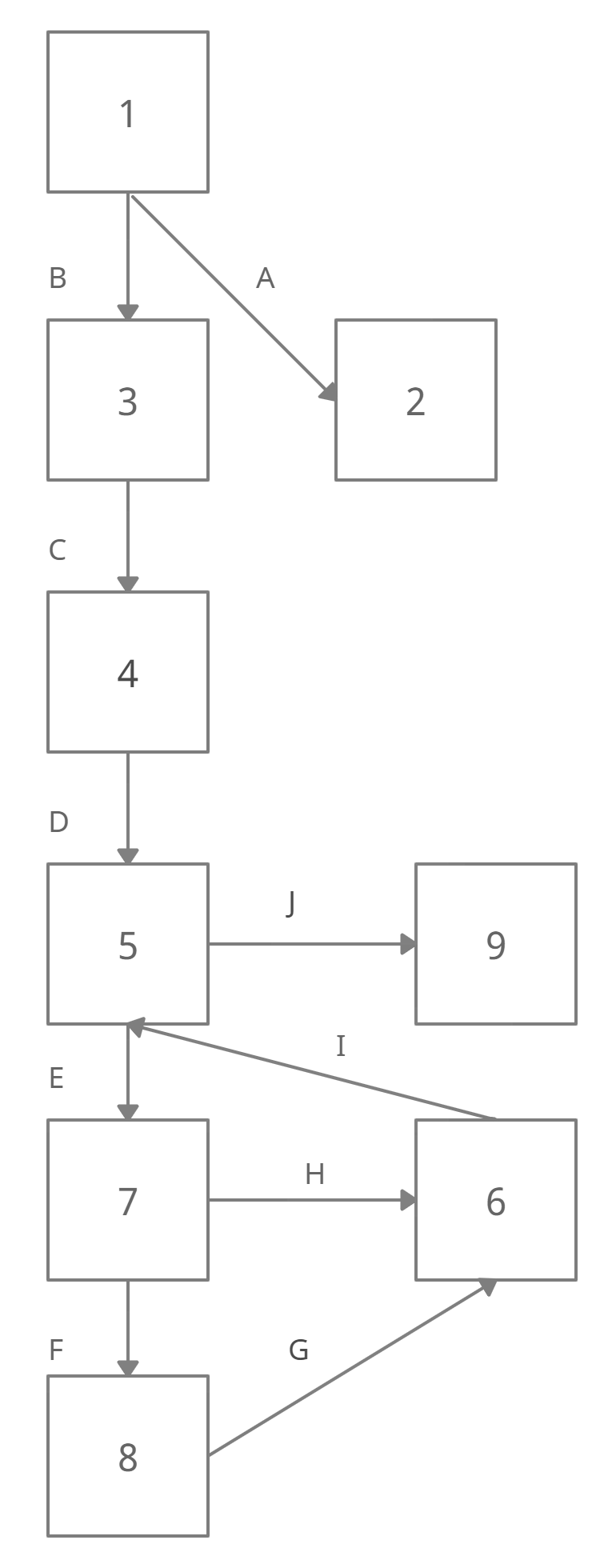


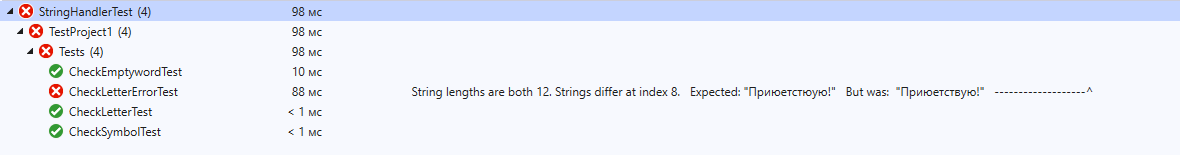
Рис 1 – УГП

Таблица 1 — Пути программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Путь | word | letter | replace | Результат |
| 1 |  | \_ |  |  | \_ |
| 2 | 1,3,4,5,7,8,6,5,7,8,6,5,7,6,5,9/B,C,D,E,F,G,D,F,G,D,E,H,D,J | aaz | a | b | bbz |

|  |  |  |
| --- | --- | --- |
| 1 | Покрытие операторов | 2 |
| 2 | Покрытие решений | 1,2 |
| 3 | Покрытие условий | 1,2 |
| 4 | Покрытие путей | 1,2 |

**6 Результаты тестирования программы**



**7 Выводы**

В результате проделанной работы был изучен метод белого ящика, позволяющий опираясь на спецификации программы получать достаточный для тестирования тестовый набор. Так, был разработан класс, который делал определенные действия над строкой, а также был составлен тест к данному классу.