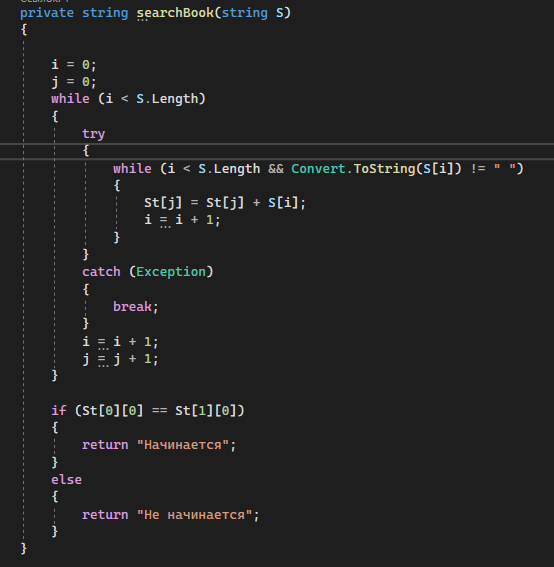
# Отчет о структурном тестировании

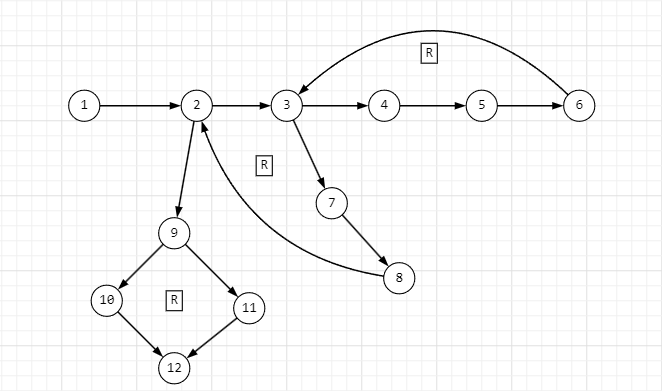
1. Указание название и функций тестируемого приложения.

Приложение Programm 14 и функция searchBook

1. Предоставление кода тестируемого метода.



1. Наличие изображения потокового графа.



1. Наличие цикломатической сложности потокового графа.

V(G) = 14 дуг – 12 + 2 = 4

F = 3 + 1 = 4

1. Наличие определенного базового множества независимых линейных путей.

Путь 1: 1-2-9-10-12

Путь 2: 1-2-9-11-12

Путь 3: 1-2-3-7-8-2-9-10-12

Путь 4: 1-2-3-4-5-6-7-8-2-9-10-12

1. Заполненная таблица для тестовых вариантов.

|  |  |  |
| --- | --- | --- |
| Значение переменной S | Эталонный вариант | Полученный результат |
| кот ток | Начинается | Начинается |
| кот кот | Не начинается | Не начинается |
|  | Завершение программы | Ошибка |
| кот | Завершение программы | Ошибка |

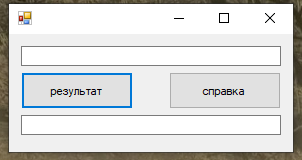
Вывод: Программа смогла справиться с тестированием на 50%. Но ее следует дорабатывать.

# Отчет о функциональном тестировании

1. Указание название и функций тестируемого приложения.

Приложение Programm 14 и функция searchBook

1. Снимок интерфейса приложения.



1. Описание классов эквивалентности для приложения

Ввод слова:

1. Ввод двух слов.

2. Слова должны быть разделены пробелом

3. Ввод английских символов.

4. Ввод русских символов.

5. Ввод цифр.

1. Анализ граничных значений

6. Можно ввести только два слова

7. Нельзя ввести 3 и более слов

8. Нельзя ввести 1 слово

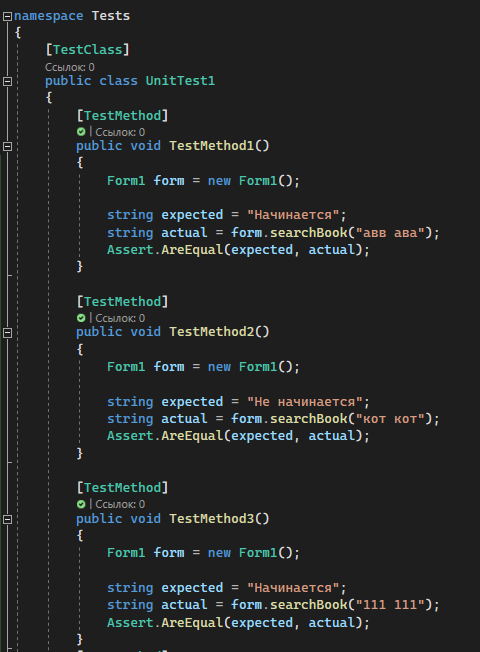
9. Слова обязательно разделены одним пробелом

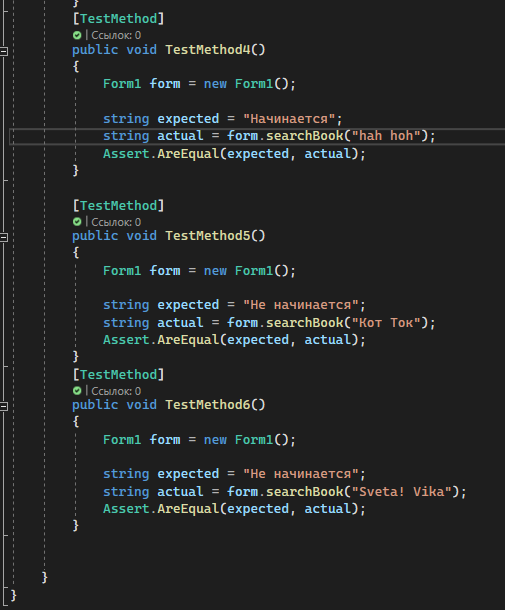
1. Заполненная таблица для тестовых сценариев.

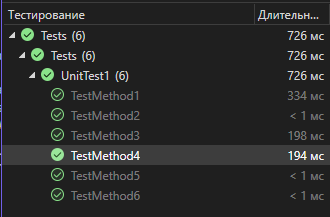
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер**  **теста** | **Описание** | **Шаги теста** | **Тестовые варианты** | **Ожидаемые результаты** |
| 1 | Ввести “авв ава” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” |
| 2 | Ввести “вика света” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Не начинается” |
| 3 | Ввести “hah hoh” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” |
| 4 | Ввести “год ток” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Не начинается” |
| 5 | Ввести “22 22” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” |
| 6 | Ввести “кот ток” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” |
| 7 | Ввести “кот ток кот” | Нажать кнопку “результат” | Ввод слов | Программа завершиться |
| 8 | Ввести “кот” | Нажать кнопку “результат” | Ввод слов | Программа завершиться |
| 9 | Ввести “к к” | Нажать кнопку “результат” | Ввод слов | Программа завершиться |

Вывод: Делая эту практическую я научилась описывать классы эквивалентности, проводить анализ граничных значений и делать тестовые сценарии.

# Создание модульных тестов в MS Visual Studio







# Отчет о функциональном тестировании через интерфейс

1. Указание название и функций тестируемого приложения.
2. Снимок интерфейса приложения.
3. Заполненная таблица для тестовых сценариев на основе отчета из лабораторной №3 Функциональное тестирование.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Описание** | **Шаги теста** | **Тестовые варианты** | **Ожидаемые результаты** | **Полученный результат** | **Статус** |
| 1 | Ввести “авв ава” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” | Программа выполниться. Будет результат “Начинается” | + |
| 2 | Ввести “вика света” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Не начинается” | Программа выполниться. Будет результат “Не начинается” | - |
| 3 | Ввести “hah hoh” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” | Программа выполниться. Будет результат “Начинается” | + |
| 4 | Ввести “год ток” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Не начинается” | Программа выполниться. Будет результат “Не начинается” | + |
| 5 | Ввести “22 22” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” | Программа выполниться. Будет результат “Начинается” | + |
| 6 | Ввести “кот ток” | Нажать кнопку “результат” | Ввод слов | Программа выполниться. Будет результат “Начинается” | Программа выполниться. Будет результат “Не начинается” | - |
| 7 | Ввести “кот ток кот” | Нажать кнопку “результат” | Ввод слов | Программа завершиться | Программа выполниться. Будет результат “Не начинается | - |
| 8 | Ввести “кот” | Нажать кнопку “результат” | Ввод слов | Программа завершиться | Программа завершилась с ошибкой | - |
| 9 | Ввести “к к” | Нажать кнопку “результат” | Ввод слов | Программа завершиться | Программа завершилась с ошибкой | - |

Вывод: Программа работает крайне нестабильно, допуская много ошибок и вылетов.

# Автоматизированное тестирование средствами AutoIT

#include <MsgBoxConstants.au3>

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "avv ava")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test1", "win")

Else

MsgBox($MB\_OK, "test1", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "vika sveta")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test2", "win")

Else

MsgBox($MB\_OK, "test2", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "hah hoh")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test3", "win")

Else

MsgBox($MB\_OK, "test3", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "god tok")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test4", "win")

Else

MsgBox($MB\_OK, "test4", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "22 22")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test5", "win")

Else

MsgBox($MB\_OK, "test5", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "kot tok")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test6", "win")

Else

MsgBox($MB\_OK, "test6", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "kot tok kot hoh")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test7", "win")

Else

MsgBox($MB\_OK, "test7", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1", "kot")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test8", "win")

Else

MsgBox($MB\_OK, "test8", "bad")

EndIf

Run("programm14\bin\Debug\programm14.exe")

Sleep(5000)

ControlSetText("Form1", "", "textBox1" "k k")

ControlClick("Form1", "", "button1")

$result = ControlGetText("Form1", "", "textBox2")

If $result = "It does start" Then

MsgBox($MB\_OK, "test9", "win")

Else

MsgBox($MB\_OK, "test9", "bad")

EndIf

# Автоматизированное тестирование средствами Selenium

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Support.UI;

using System;

[TestFixture]

public class Tests

{

private IWebDriver driver;

private WebDriverWait wait;

[SetUp]

public void Setup()

{

// Инициализация браузера

driver = new ChromeDriver();

wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

}

[Test]

public void Test1()

{

RunTest("авв ава", "Начинается");

}

[Test]

public void Test2()

{

RunTest("вика света", "Не начинается");

}

[Test]

public void Test3()

{

RunTest("hah hoh", "Начинается");

}

[Test]

public void Test4()

{

RunTest("год ток", "Не начинается");

}

[Test]

public void Test5()

{

RunTest("22 22", "Начинается");

}

[Test]

public void Test6()

{

RunTest("кот ток", "Начинается");

}

[Test]

public void Test7()

{

RunTest("кот ток кот", "Программа завершится");

}

[Test]

public void Test8()

{

RunTest("кот", "Программа завершится");

}

[Test]

public void Test9()

{

RunTest("к к", "Программа завершится");

}

[TearDown]

public void TearDown()

{

// Закрытие браузера

driver.Quit();

}

private void RunTest(string inputText, string expectedResult)

{

// Навигация к странице с вашим приложением

driver.Navigate().GoToUrl("URL\_ВАШЕГО\_ПРИЛОЖЕНИЯ");

// Находим элемент ввода слова и вводим текст

IWebElement inputElement = wait.Until(d => d.FindElement(By.Id("textBox1"))); // Замените на соответствующий селектор

inputElement.Clear();

inputElement.SendKeys(inputText);

// Нажимаем кнопку "результат"

IWebElement resultButton = wait.Until(d => d.FindElement(By.Id("button1"))); // Замените на соответствующий селектор

resultButton.Click();

// Ждем некоторое время для завершения выполнения программы

System.Threading.Thread.Sleep(5000);

// Получаем результат из элемента

IWebElement resultElement = wait.Until(d => d.FindElement(By.Id("textBox2"))); // Замените на соответствующий селектор

string actualResult = resultElement.GetAttribute("value");

// Сравниваем ожидаемый и фактический результат

Assert.AreEqual(expectedResult, actualResult, $"Тест: Ввод '{inputText}', Ожидаемый результат '{expectedResult}', Фактический результат '{actualResult}'");

}

}