МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе №4

на тему: «Тестирование пользовательского интерфейса»

по дисциплине: «Качество и тестирование программного обеспечения»

Вариант 7

Выполнила: Карпикова С.П.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71ПГ

Проверили: Ужаринский А.Ю., Конюхова О.В.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Орел, 2020 г.

**Задание:** разработать сценарии использования и провести тестирование пользовательского интерфейса приложения.

**Ход работы**

**Программа на С#:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab4

{

unsafe class DoubleHash

{

public const int TABLE\_SIZE = 20;

public const int PRIME = 7;

int[] hashTable;

int curr\_size;

public bool isFull()

{

return (curr\_size == TABLE\_SIZE);

}

int hash1(int key)

{

return (key % TABLE\_SIZE);

}

int hash2(int key)

{

return (PRIME - (key % PRIME));

}

public DoubleHash()

{

hashTable = new int[TABLE\_SIZE];

curr\_size = 0;

for (int i = 0; i < TABLE\_SIZE; i++)

hashTable[i] = -1;

}

public void insert(int key)

{

if (isFull())

return;

int index = hash1(key);

if (hashTable[index] != -1)

{

int index2 = hash2(key);

int i = 1;

for (; ;)

{

int newIndex = (index + i \* index2) %

TABLE\_SIZE;

if (hashTable[newIndex] == -1)

{

hashTable[newIndex] = key;

break;

}

i++;

}

}

else

hashTable[index] = key;

curr\_size++;

}

public string get()

{

string result = "";

for (int i = 0; i < TABLE\_SIZE; i++)

{

if (hashTable[i] != -1)

result += i + " --> " + hashTable[i] + Environment.NewLine;

else

result += i + Environment.NewLine;

}

return result;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab4

{

public partial class Form1 : Form

{

DoubleHash h;

public Form1()

{

InitializeComponent();

h = new DoubleHash();

}

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

{

h.insert(Convert.ToInt32(textBox1.Text));

textBox1.Text = "";

button2.Enabled = true;

button1.Enabled = false;

textBox1.Select();

}

private void button2\_Click(object sender, EventArgs e)

{

textBox2.Text = h.get();

}

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

{

if (textBox1.Text.Length > 0) {

button1.Enabled = true;

} else {

button1.Enabled = false;

}

}

}

}

В данной программе возможен единственный сценарий использования, тест которого представлен далее в таблице.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Действие** | **Реакция системы** | **Рисунок** | **Результат** |
| 1 | Щелкните на текстовое поле для ввода значения и введите «1». | Кнопка «Добавить» станет доступна пользователю для нажатия. | 1 | Верно |
| 2 | Сотрите введенное значение «1» из текстового поля. | Кнопка «Добавить» станет недоступна пользователю для нажатия. | 2 | Верно |
| 3 | Снова введите в текстовое поле значение «1» и нажмите кнопку «Добавить» | Кнопка «Добавить» станет недоступна, текстовое поле для ввода очистится, кнопка «Получить все элементы» станет доступна пользователю для нажатия. | 3 | Верно |
| 4 | Введите в текстовое поле значение «1» и нажмите кнопку «Добавить» | Кнопка «Добавить» станет недоступна, текстовое поле для ввода очистится. | 4 | Верно |
| 5 | Нажмите на кнопку «Получить все элементы». | В нередактируемом текстовом поле появятся все значения хэш-таблицы. | 5 | Верно |
| 6 | Добавьте элемент «2» в текстовое поле и нажмите кнопку «Добавить». | Кнопка «Добавить» станет недоступна, текстовое поле для ввода очистится, поле со значениями хэш таблицы останется прежним. | 6 | Верно |
| 7 | Нажмите на кнопку «Получить все элементы». | В нередактируемом текстовом поле появятся новые значения хэш-таблицы. | 7 | Верно |

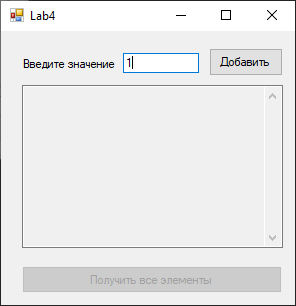


Рисунок 1 – Пользовательский интерфейс, состояние №1

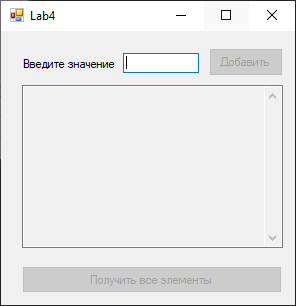


Рисунок 2 – Пользовательский интерфейс, состояние №2

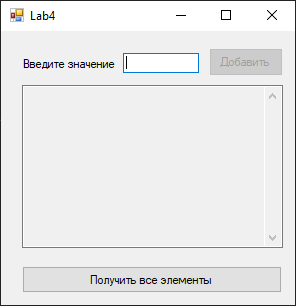
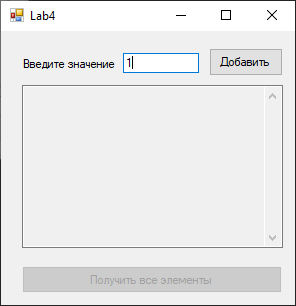


Рисунок 3 – Пользовательский интерфейс, состояние №3

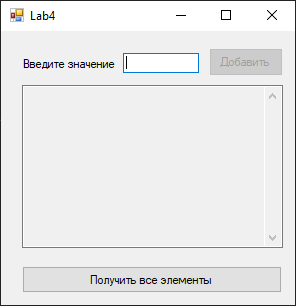
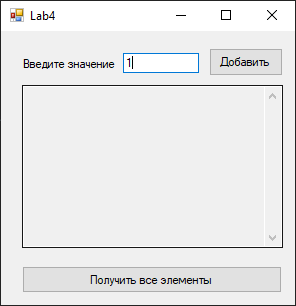


Рисунок 4 – Пользовательский интерфейс, состояние №4

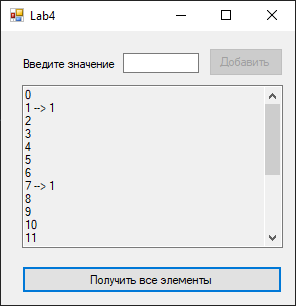


Рисунок 5 – Пользовательский интерфейс, состояние №5

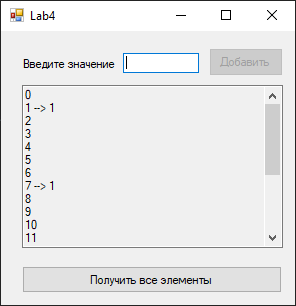
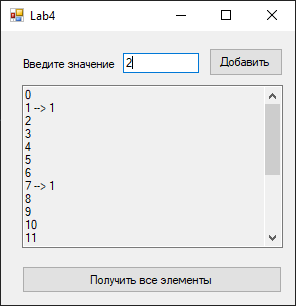


Рисунок 6 – Пользовательский интерфейс, состояние №6

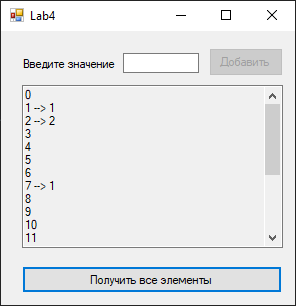


Рисунок 7 – Пользовательский интерфейс, состояние №7

Рассмотрим передачу фокуса управления между элементами пользовательского интерфейса по нажатию на кнопку таб.

При запуске программы фокус установлен на текстовом поле ввода значения (рисунок 2).

Далее при нажатии на кнопку таб фокус переносится на кнопку «Добавить» (рисунок 8). При нажатии на кнопку, фокус возвращается назад на текстовое поле для дальнейшего заполнения таблицы.

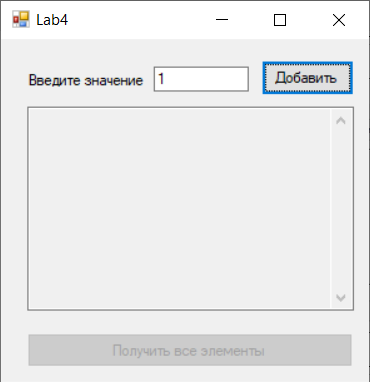


Рисунок 8 – Пользовательский интерфейс, фокус на кнопке «Добавить»

Рассмотрим передачу фокуса при выборе кнопки «Получить все элементы». При заполненном текстовом поле фокус установится сначала на кнопку «Добавить», а затем на кнопку «Получить все элементы» (рисунок 9). В случае, если текстовое поле пустое, то фокус переместится сразу на нужную кнопку (рисунок 5).

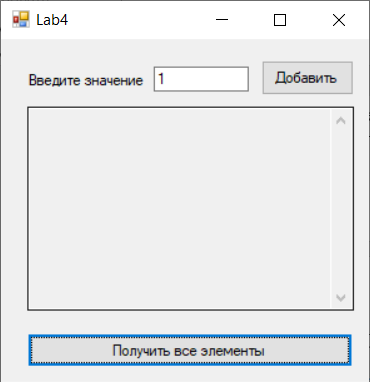
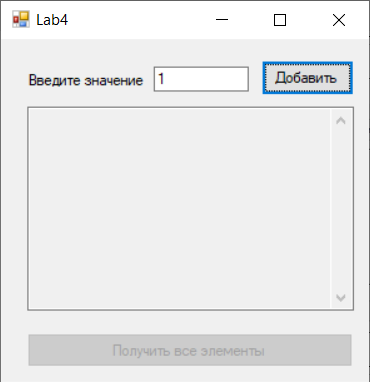


Рисунок 9 – Пользовательский интерфейс, фокус на кнопке «Получить все элементы»