МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное

образовательное учреждение

высшего образования

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

**по дисциплине**

**“** **Тестирование программного обеспечения”**

**Закодированные тесты пользовательского интерфейса.**

Выполнил: студент гр. ФИб-4302-51-00 Шатунов Д. И. \_\_\_\_\_\_

Проверил: старший преподаватель кафедры ПМиИ Фищева И.Н. \_\_\_\_\_

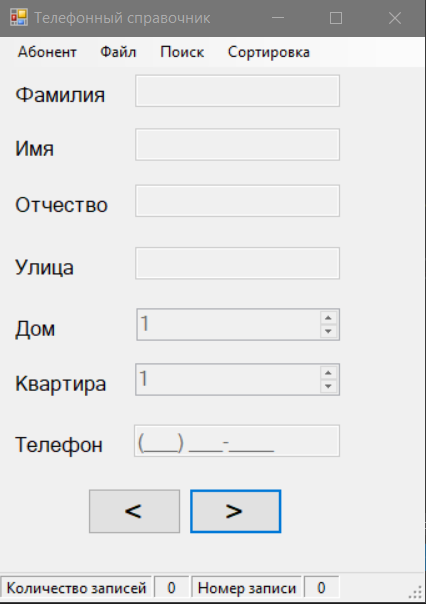
Киров 2022

**Задание**

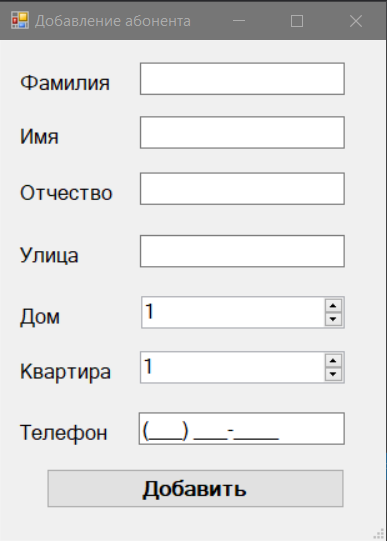
Необходимо взять программу, которую Вы написали для курсового проекта по Объектно-ориентированному программированию. В этой программе нас будет интересовать пользовательский интерфейс. Необходимо выполнить тестирование работоспособности пользовательского интерфейса. Для этого с помощью Coded UI создайте набор тестов по основной функциональности программы.

**Информация о тестируемой программе**

Для тестирования была взята программа «Телефонный справочник». Программа состоит из MainForm



И подформы для добавления абонента



**Сгенерированные тесты**

**Добавление абонента**

[TestMethod]

public void AddNewAbonent()

{

Mouse.Click(uIMenuStrip1Window, new Point(210, 72)); //нажимаем кнопку абонент в менюстрип

Mouse.Click(uIItemWindow, new Point(211, 66)); //из меню выбираем пункт добавить

uILastNameTextBoxEdit.Text = this.RecordedMethod1Params.UILastNameTextBoxEditText; //вводим значение фамилии

Mouse.Click(uINameTextBoxWindow, new Point(115, 84));//нажимаем на текстбокс с именем

uINameTextBoxEdit.Text = this.RecordedMethod1Params.UINameTextBoxEditText; //вводим значение

Mouse.Click(uIPatronymicTextBoxWindow, new Point(100, 70)); //нажимаем на текстбокс с отчеством

uIPatronymicTextBoxEdit.Text = this.RecordedMethod1Params.UIPatronymicTextBoxEditText; //вводим значение

Mouse.Click(uIStreetTextBoxWindow, new Point(94, 86)); //нажимаем на текстбокс с улицей

uIStreetTextBoxEdit.Text = this.RecordedMethod1Params.UIStreetTextBoxEditText; //вводим значение

Mouse.Click(uI\_\_\_\_\_\_\_\_\_\_Window, new Point(92, 132)); //нажимаем на текстбокс с телефоном

uIPhoneMaskedTextBoxEdit.Text = this.RecordedMethod1Params.UIPhoneMaskedTextBoxEditText; //вводим значение

Mouse.Click(uIДобавитьWindow, new Point(180, 136)); //нажимаем кнопку добавить }

**Изменение абонента**

[TestMethod]

public void ChangeAbonent()

{

Mouse.Click(uIMenuStrip1Window, new Point(208, 72)); //вызываем меню приложения нажатием кнопки абонент

Mouse.Click(uIItemWindow, new Point(222, 101)); //нажимаем кнопку изменить в меню

Mouse.Click(uINameTextBoxWindow, new Point(262, 77)); //нажимаем на поле имени которое хотим изменить

uINameTextBoxEdit.Text = this.ChangeAbonentParams.UINameTextBoxEditText; //вводим новое значение

Mouse.Click(uIИзменитьWindow, new Point(205, 147)); //нажимаем кнопку изменить для подтверждения изменений

}

**Удаление абонента**

[TestMethod]

public void DeleteAbonent()

{

Mouse.Click(uIMenuStrip1Window, new Point(201, 75)); //вызываем меню приложения нажатием кнопки абонент

Mouse.Click(uIItemWindow, new Point(222, 122)); //нажимаем кнопку удалить в меню

}

**Сохранение абонента в файл**

[TestMethod]

public void SaveAbonent()

{

Mouse.Click(uIMenuStrip1Window, new Point(268, 75)); )); //вызываем меню приложения нажатием кнопки файл

Mouse.Click(uIItemWindow1, new Point(249, 105)); //нажимаем кнопку сохранить в меню

uIИмяфайлаComboBox.EditableItem = this.SaveAbonentParams.UIИмяфайлаComboBoxEditableItem; //в открывшемся окне сохранения вводим название файла

Mouse.Click(uIСохранитьButton, new Point(39, 7)); //нажимаем кнопку сохранить

}

**Открытие абонента из файла**

[TestMethod]

public void OpenAbonent()

{

Mouse.Click(uIMenuStrip1Window, new Point(272, 66)); //вызываем меню приложения нажатием кнопки файл

Mouse.Click(uIItemWindow1, new Point(220, 69)); //нажимаем кнопку открыть в меню

uIПросмотрэлементовList.SelectedItemsAsString = this.OpenAbonentParams.UIПросмотрэлементовListSelectedItemsAsString; //в окне открытия файлов выбираем нужный файл

Mouse.Click(uIОткрытьButton, new Point(38, 13)); // нажимаем кнопку открыть}

**Поиск абонента**

[TestMethod]

public void FindAbonent()

{

Mouse.Click(uIMenuStrip1Window, new Point(333, 63)); //вызываем меню приложения нажатием кнопки поиск

Mouse.Click(uIItemWindow2, new Point(256, 65)); //выбираем пункт найти по фио и нажимаем его

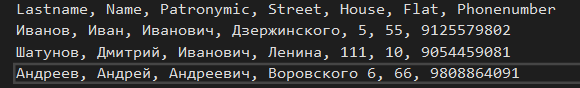
uILastNameTextBoxEdit.Text = this.FindAbonentParams.UILastNameTextBoxEditText; //вводим в поле фамилии нужное значение

Mouse.Click(uIНайтиWindow, new Point(227, 107)); //нажимаем кнопку найти

}

**Добавление абонента из файла**

Тестовый файл



[TestMethod]

public void AddAbonentFromFile()

{

Mouse.Click(uIMenuStrip1Window, new Point(210, 72)); //вызываем меню приложения нажатием кнопки абонент

Mouse.Click(uIItemWindow, new Point(211, 66)); //из меню выбираем пункт добавить

uILastNameTextBoxEdit.Text = TestContext.DataRow["Lastname"].ToString(); //Поле фамилия заполняет согласно тестовому файлу

Mouse.Click(uINameTextBoxWindow, new Point(115, 84)); //нажимаем на поле имя

uINameTextBoxEdit.Text = TestContext.DataRow["Name"].ToString(); //поле имя заполняется из тестового файла

Mouse.Click(uIPatronymicTextBoxWindow, new Point(100, 70)); //нажимаем на поле отчество

uIPatronymicTextBoxEdit.Text = TestContext.DataRow["Patronymic"].ToString(); //поле отчество заполняется из тестового файла

Mouse.Click(uIStreetTextBoxWindow, new Point(94, 86)); //нажимаем на поле улица

uIStreetTextBoxEdit.Text = TestContext.DataRow["Street"].ToString(); //поле улица заполняется из тестового файла

Mouse.Click(uIHouseTextBoxWindow, new Point(94, 86)); //нажимаем на поле дом

uIHouseTextBoxEdit.Text = TestContext.DataRow["House"].ToString(); //поле дом заполняется из тестового файла

Mouse.Click(uIFlatTextBoxWindow, new Point(94, 86)); //нажимаем на поле квартира

uIFlatTextBoxEdit.Text = TestContext.DataRow["Flat"].ToString(); //поле квартира заполняется из тестового файла

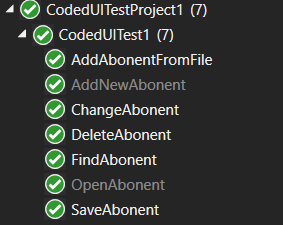
Mouse.Click(uI\_\_\_\_\_\_\_\_\_\_Window, new Point(92, 132)); //нажимаем на поле телефон

uIPhoneMaskedTextBoxEdit.Text = TestContext.DataRow["Phonenumber"].ToString(); //поле номера телефона заполняется из файла

Mouse.Click(uIДобавитьWindow, new Point(180, 136)); //нажимаем кнопку добавить

}

**Результаты тестирования**

****

**Вывод**

Во время тестирования ошибок в приложении не было найдено. Но не было проведено тестирование, когда абонентов в книге много, например 1000. С помощью закодированных тестов пользовательского интерфейса можно проверять работу пользовательского интерфейса приложения. Эти тесты позволяют убедиться, что все приложение, в том числе интерфейс, работает правильно. Закодированные тесты пользовательского интерфейса полезны, если интерфейс содержит логику проверки или другую логику.