

Национальный Исследовательский Университет
«Московский Энергетический Институт»
Кафедра прикладной математики и искусственного интеллекта

Тема: Автоматизация тестирования приложения с графическим интерфейсом с применением технологии Coded UI Test.

Студент: Ростовых Александра

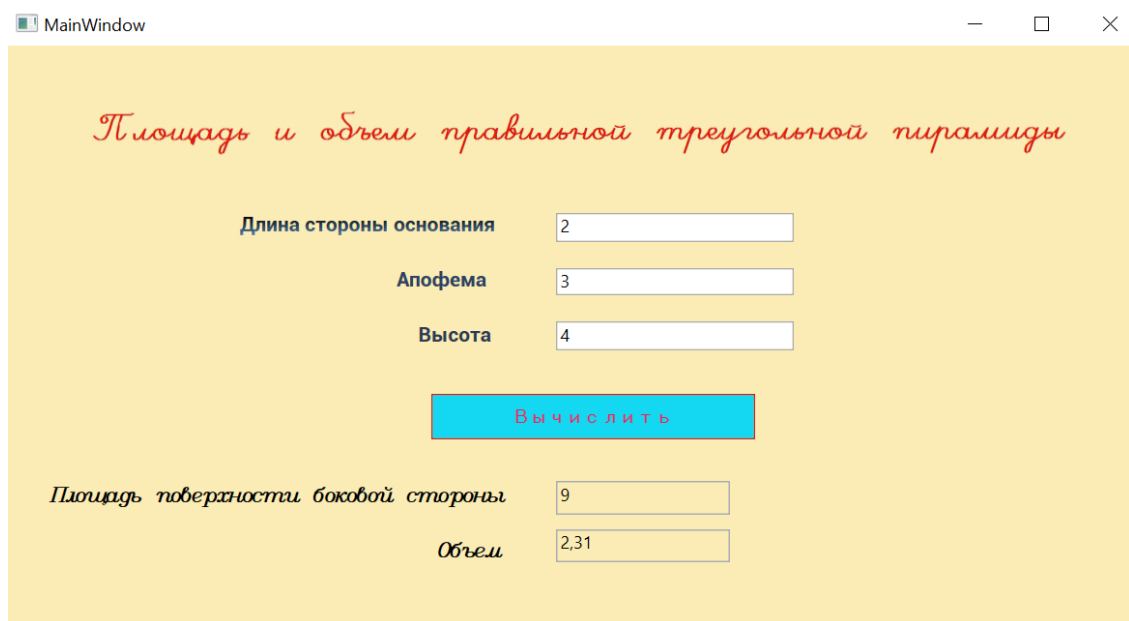
Москва 2021

Цель работы

Научиться автоматизировать тестирование приложений через графический интерфейс пользователя с применением технологии Coded UI Test.

1. Проанализировать графический интерфейс приложения, которое будет тестироваться.

Создадим WPF приложение на языке C#, которое вычисляет площадь боковой поверхности и объем правильной треугольной пирамиды по введенным пользователем длине стороны основания, апофеме и высоте пирамиды, после нажатия кнопки «Вычислить».



MainWindow

Площадь и объем правильной треугольной пирамиды

Длина стороны основания 2

Апофема 3

Высота 4

Вычислить

Площадь поверхности боковой стороны 9

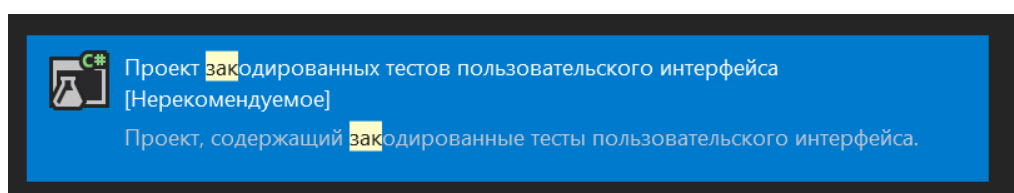
Объем 2,31

2. Разработать список автоматизированных GUI тест-кейсов и подготовить тестовые данные.

Тестироваться будет следующее: пользователь вводит длину стороны основания, апофему и высоту пирамиды, затем нажимает кнопку «Вычислить», после чего внизу должен отобразиться верный результат.

3. Создать проект Coded UI Test в Microsoft Visual Studio.

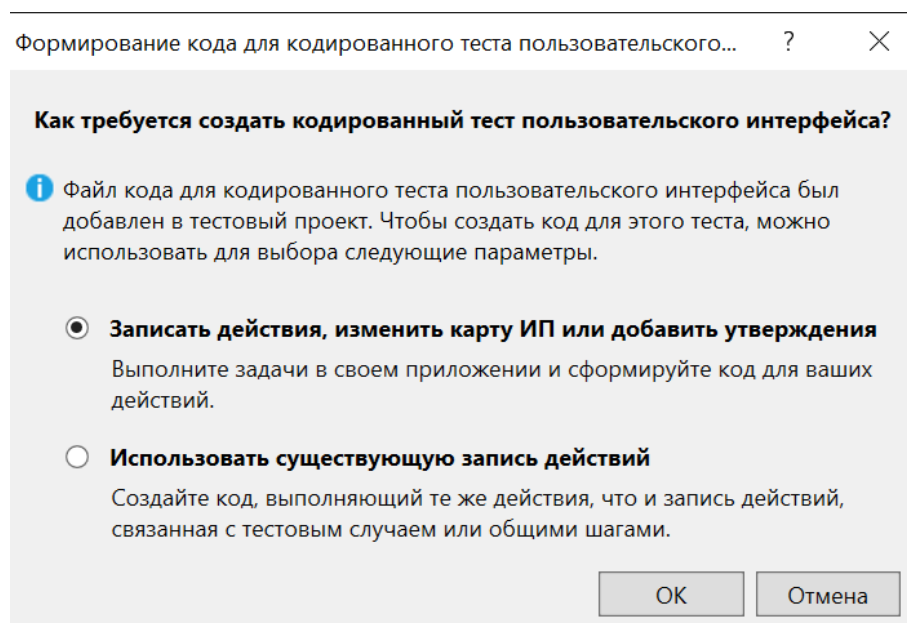
Добавляем в проект проект закодированных тестов пользовательского интерфейса:



Далее появляется окно, в котором можно выбрать 2 варианта:

- Записать новые действия,
- Воспользоваться записью старых действий.

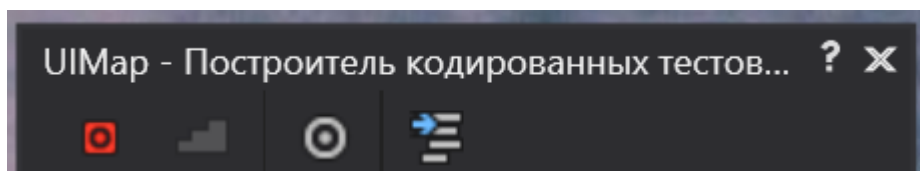
Выбираем 1 вариант и нажимаем ОК.



В нижнем правом углу рабочего стола появляется окно построителя кодированных тестов.

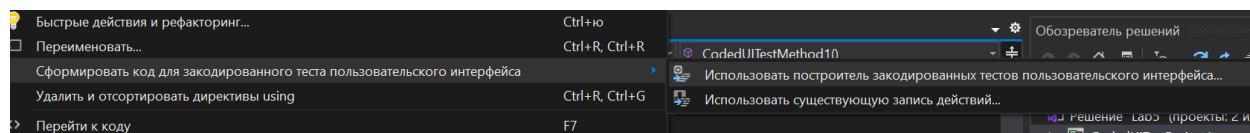
На нем 4 кнопки:

- Начать запись.
- Просмотреть записанные шаги.
- Кнопка для изменения карты пользовательского интерфейса и добавления утверждения.
- Сгенерировать код.

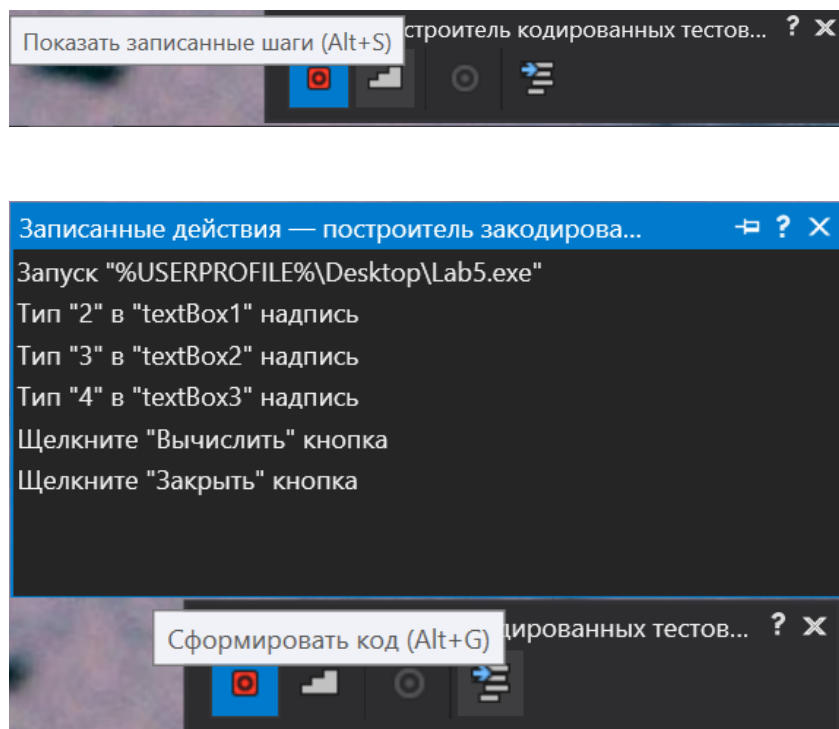


4. Записать действия и получить генерируемый средой код теста.

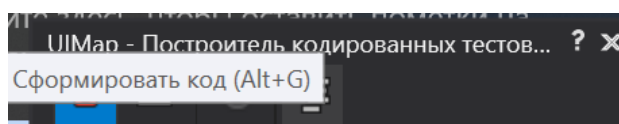
Можем открыть построитель тестов из метода в тестовом классе щелчком правой кнопкой мыши и выбором соответствующего пункта.



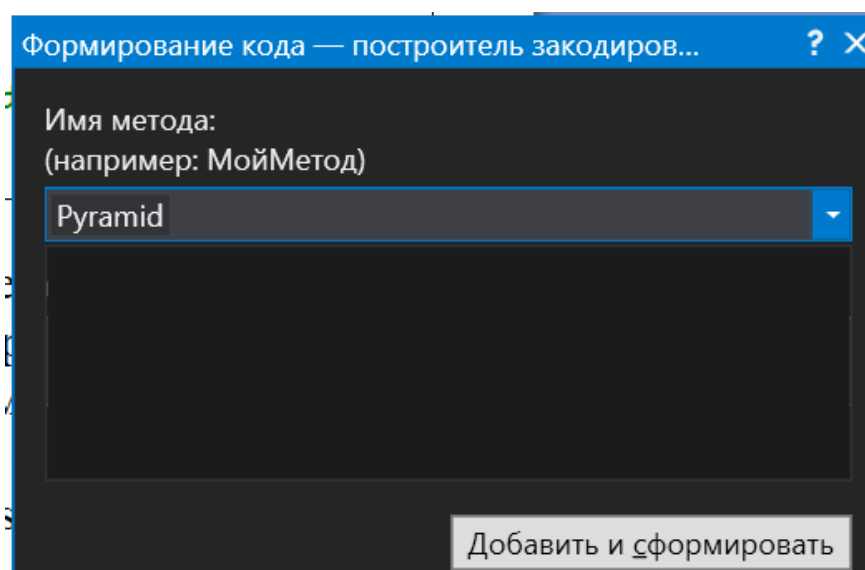
Затем нажимаем «Запись» и открываем наше приложение и вводим туда значения. После закрытия приложения заканчиваем запись и смотрим записанные шаги.



Затем нажимаем кнопку «Сформировать код»



Называем метод:



И видим в тестовом классе новый код:

```
[TestMethod]
public void CodedUITestMethod1()
{
    this.UIMap.Pyramid();
    // Чтобы создать код для этого теста, выберите в контекстном меню команду
    "Формирование кода для кодированного теста пользовательского интерфейса", а затем
    выберите один из пунктов меню.
}
```

Посмотрим так же файл UIMap.Designer.cs и реализацию метода Pyramid() в нем.

```
public void Pyramid()
{
    #region Variable Declarations
    WpfEdit uITextBox1Edit = this.UIMainWindowWindow.UITextBox1Edit;
    WpfEdit uITextBox2Edit = this.UIMainWindowWindow.UITextBox2Edit;
    WpfEdit uITextBox3Edit = this.UIMainWindowWindow.UITextBox3Edit;
    WpfButton uIВычислитьButton = this.UIMainWindowWindow.UIВычислитьButton;
    WinButton uIЗакрытьButton = this.UIMainWindowWindow1.UIЗакрытьButton;
    #endregion

    // Запуск "%USERPROFILE%\Desktop\Lab5.exe"
    ApplicationUnderTest uIMainWindowWindow =
    ApplicationUnderTest.Launch(this.PyramidParams.UIMainWindowWindowExePath,
    this.PyramidParams.UIMainWindowWindowAlternateExePath);

    // Тип "2" в "textBox1" надпись
    uITextBox1Edit.Text = this.PyramidParams.UITextBox1EditText;

    // Тип "3" в "textBox2" надпись
    uITextBox2Edit.Text = this.PyramidParams.UITextBox2EditText;

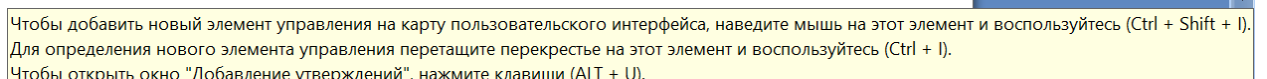
    // Тип "4" в "textBox3" надпись
    uITextBox3Edit.Text = this.PyramidParams.UITextBox3EditText;

    // Щелкните "Вычислить" кнопка
    Mouse.Click(uIВычислитьButton, new Point(161, 10));

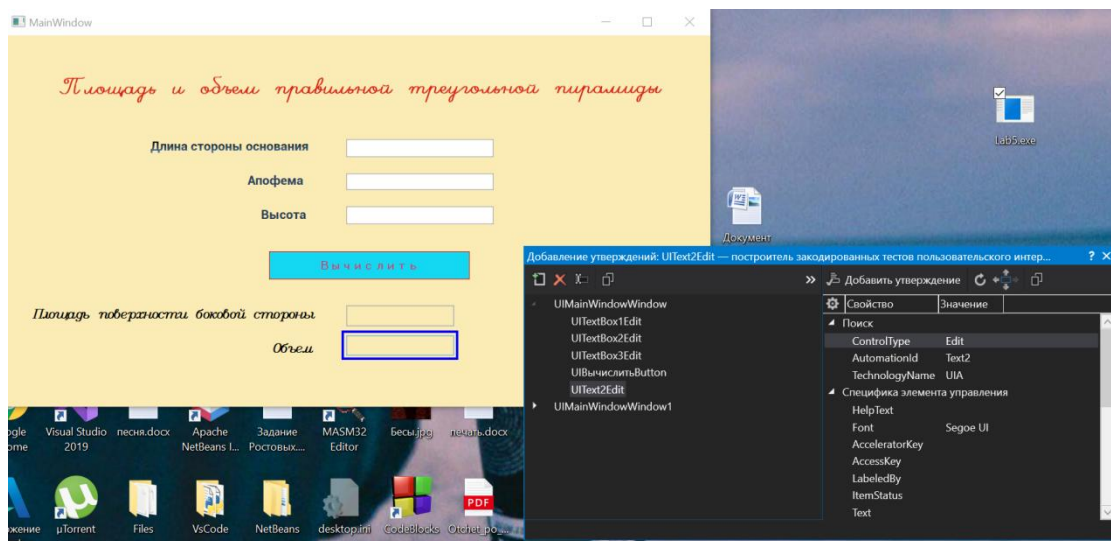
    // Щелкните "Закрыть" кнопка
    Mouse.Click(uIЗакрытьButton, new Point(32, 6));
}
```

5. Настроить проверки

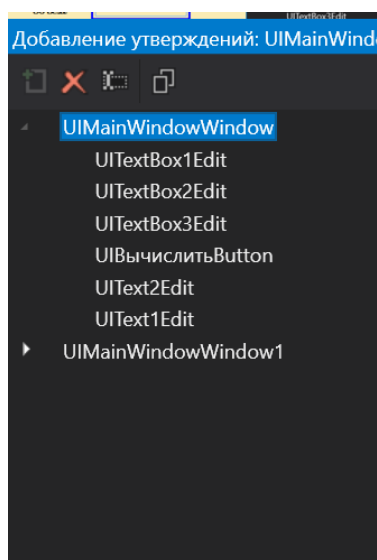
Теперь надо добавить элементы управления TextBox, для того чтобы мы записали для них утверждения, ведь в них будет проверяться результат. Нажимаем на серый круг (кнопка на строителе тестов) и добавляем поочередно оба элемента. Ниже представлены скриншоты, как это происходит, мы просто выбираем элемент в нашем приложении и добавляем его к элементам в строителе тестов.



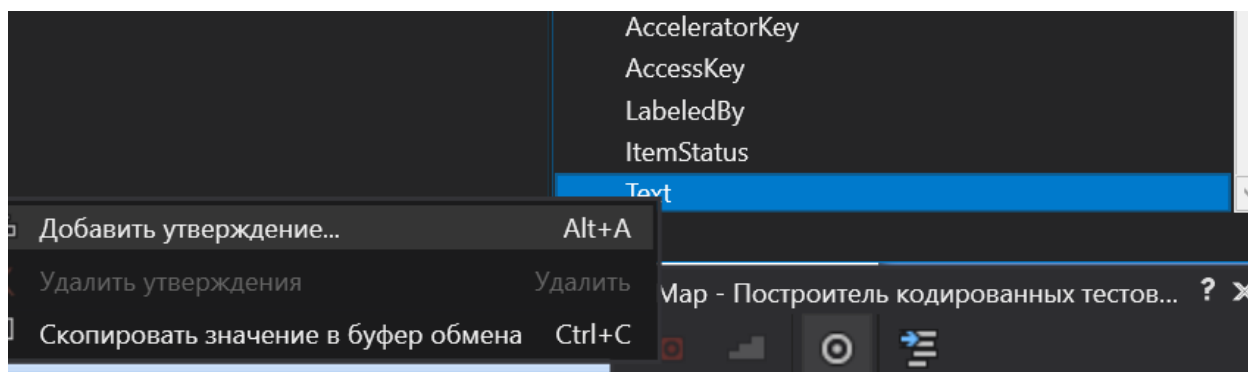
Чтобы добавить новый элемент управления на карту пользовательского интерфейса, наведите мышь на этот элемент и воспользуйтесь (Ctrl + Shift + I). Для определения нового элемента управления перетащите перекрестье на этот элемент и воспользуйтесь (Ctrl + I). Чтобы открыть окно "Добавление утверждений", нажмите клавиши (ALT + U).



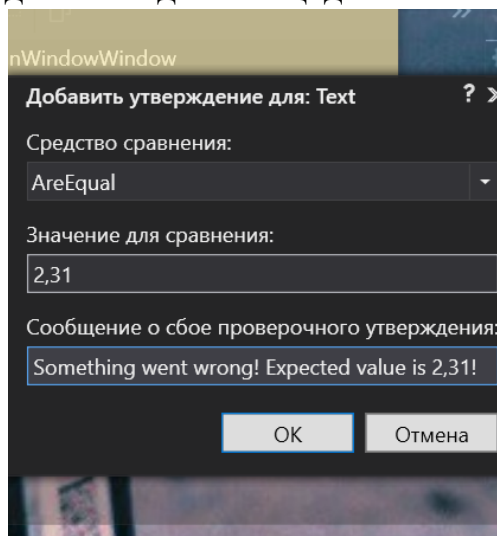
Итог: оба элемента добавлены.



Далее нужно добавить утверждения, ищем свойство Text для наших TextBox'ов и нажимаем «Добавить утверждение»



Выбираем средство AreEqual, например сначала для объема, вводим ожидаемое значение и сообщение об ошибке. Аналогично повторяем действия для площади.



Нажимаем «сформировать код», называем метод AssertMethod1, заходим в CodedUITest1.cs и смотрим получившийся код:

```
[TestMethod]
public void CodedUITestMethod1()
{
    this.UIMap.Pyramid();
    this.UIMap.AssertMethod1();
    // Чтобы создать код для этого теста, выберите в контекстном меню команду
    "Формирование кода для кодированного теста пользовательского интерфейса", а затем
    выберите один из пунктов меню.
}
```

А так же можем просмотреть в файле UIMap.Designer.cs:

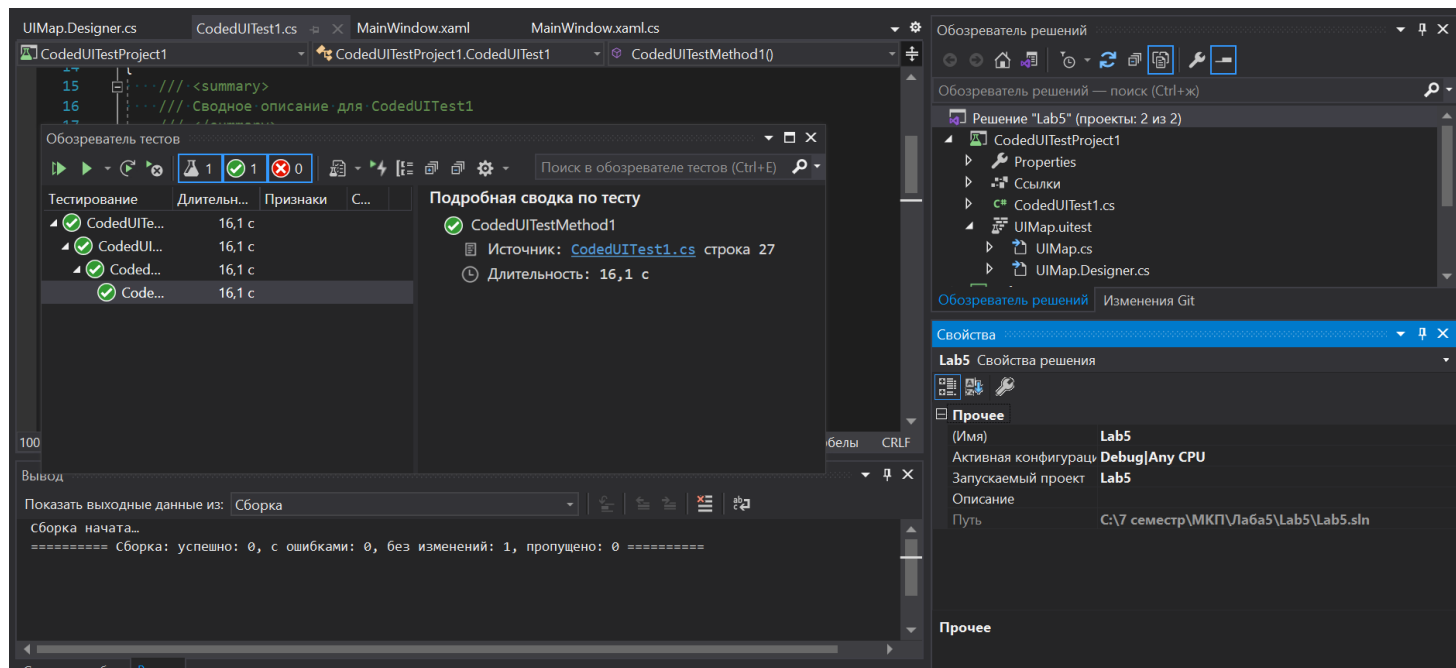
```
public void AssertMethod1()
{
    #region Variable Declarations
    WpfEdit uIText1Edit = this.UIMainWindowWindow.UIText1Edit;
    WpfEdit uIText2Edit = this.UIMainWindowWindow.UIText2Edit;
    #endregion

    // Убедитесь, что свойство Text "Text1" надпись равняется "9"
    Assert.AreEqual(this.AssertMethod1ExpectedValues.UIText1EditText,
        uIText1Edit.Text, "Something went wrong! Expected value is 9!");

    // Убедитесь, что свойство Text "Text2" надпись равняется "2,31"
    Assert.AreEqual(this.AssertMethod1ExpectedValues.UIText2EditText,
        uIText2Edit.Text, " Something went wrong! Expected value is 2,31!");
}
```

6. Запустить тест

Запускаем тест и смотрим на результат:



Тест успешно пройден!