

# Юнит-тестирование (модульное тестирование)

«Выдача кредита банком»

## Цель лабораторной работы:

Получить практические навыки модульного тестирования кода программных компонентов.

## Создание проекта тестирования

Модульное тестирование используется разработчиками для проверки правильности работы методов классов. Модульные тесты сохраняются в системе контроля версий и выполняются при каждом построении приложения. Кроме того, модульные тесты являются основой регрессионного тестирования, которое выполняется при добавлении новых возможностей или модификации приложения.

Для реализации модульного тестирования разрабатываемого приложения создадим тестовый проект модульного тестирования. Для этого добавим в решение Interface проект модульного теста - UnitTestProjectLibrary (рисунок 1)

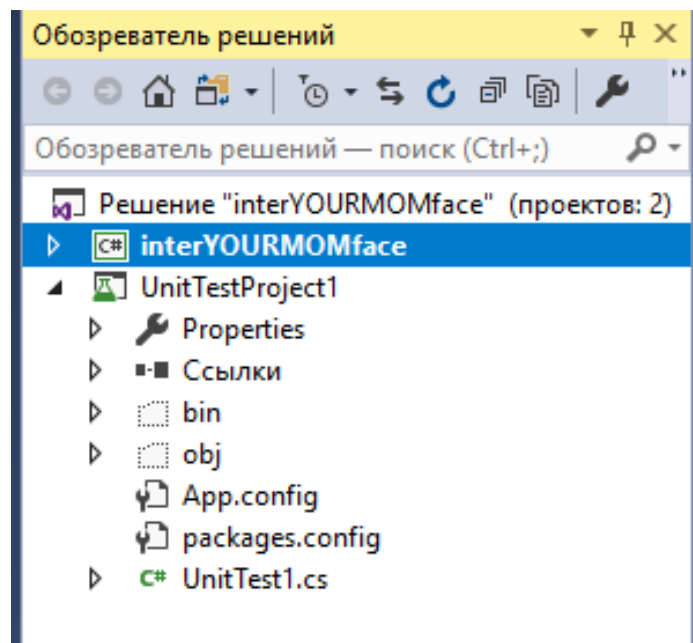


Рисунок 1 – Проект модульного тестирования

## Создание теста:

При создании проекта модульного теста генерируется класс для тестирования UnitTest (рисунок 2)

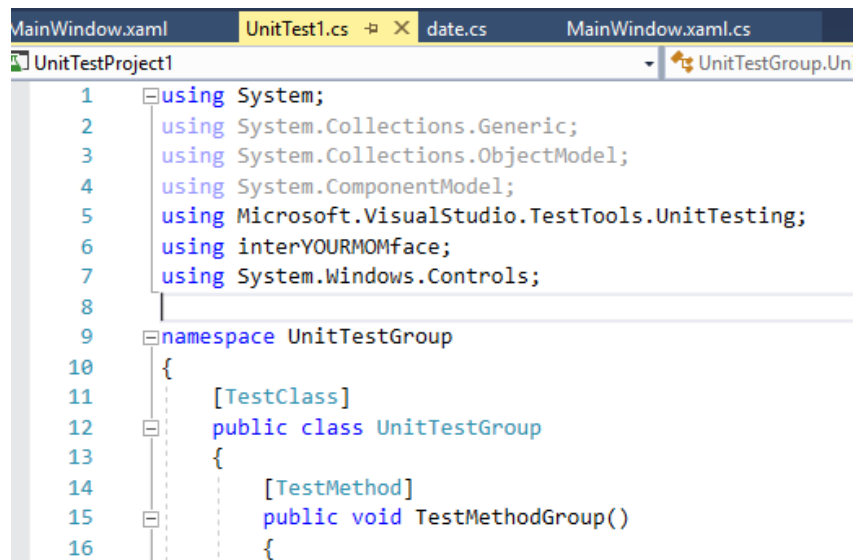


Рисунок 2 – Код созданного класса модульного тестирования

Исходя из содержания классов основной программы, были разработаны следующие юнит-тесты:

1. Тестирование метода, использующегося для вывода данных о всех заемщиках в элемент DataGridView (рисунок 3)

```

[TestMethod]
public void dgv()
{
    Model1Container db = new Model1Container();
    DataGrid dgv = new DataGrid();

    interYOURMOMface.itemsourse.dgv(db, dgv);
    Assert.IsNotNull(dgv.Items); //проверка на вывод данных в dgv
}

```

Рисунок 3 – Тестирование вывода данных о всех заемщиках

2. Тестирование метода, использующегося для расчета итоговой суммы кредитования с учетом процентной ставки (рисунок 4)

```

public void sum()
{
    TextBox text = new TextBox();
    TextBox text2 = new TextBox();
    ComboBox combo = new ComboBox();

    text.Text = "15000";
    combo.Items.Add(12);
    combo.SelectedItem = 12;

    interYOURMOMface.rem.remainer(text, combo, text2);
    Assert.IsNotNull(text2.Text); //проверка на подсчет итоговой суммы кредитования
}

```

Рисунок 4 – Тестирование расчета итоговой суммы кредитования

### 3. Тестирование метода, использующегося для подсчета количества месяцев кредитования (рисунок 5)

```
[TestMethod]
public void period()
{
    DatePicker start = new DatePicker();
    DatePicker end = new DatePicker();
    DatePicker start_card = new DatePicker();
    DatePicker end_card = new DatePicker();
    TextBox sum_text = new TextBox();

    start.SelectedDate = DateTime.Now.Date;
    end.SelectedDate = DateTime.Now.Date;
    start_card.SelectedDate = DateTime.Now.Date;
    end_card.SelectedDate = DateTime.Now.Date;

    interYOURMOMface.date.date_rem(end, start, sum_text, start_card, end_card);
    Assert.IsNotNull(sum_text.Text); //подсчет кол-ва месяцев кредитования
}
```

Рисунок 5 – Тестирование подсчета количества месяцев кредитования

Результаты тестирования формируются классом Assert, который производит сравнение ожидаемых результатов с фактическими.

После построения тестового проекта созданные тесты отображаются в обозревателе тестов (рисунок 6).

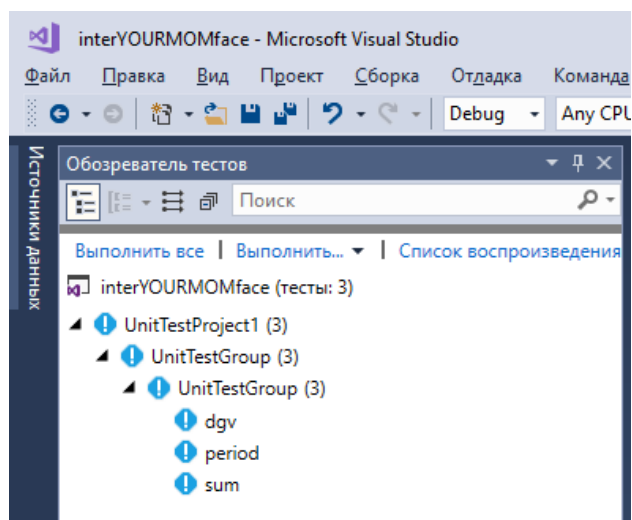


Рисунок 6 – Созданные тесты в обозревателе

При выполнении теста в обозревателе тестов отображаются параметры его выполнения (рисунок 7).

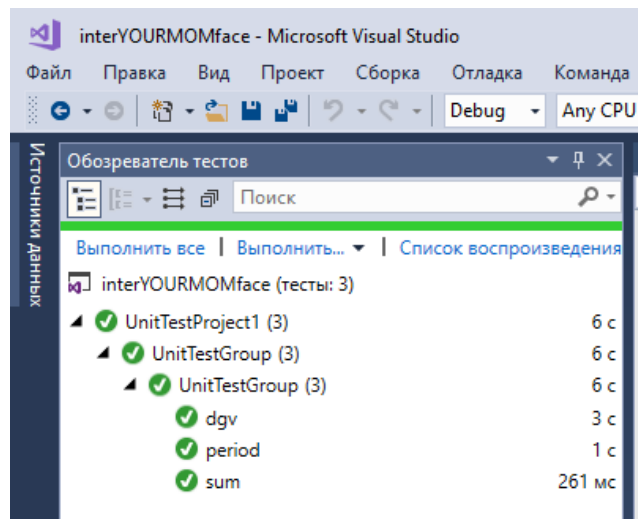


Рисунок 7 – Выполненные тесты в обозревателе тестов

### **Вывод:**

Получил практические навыки модульного тестирования кода программных компонентов.