

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ОТЧЕТ

К лабораторной работе №4 (Юнит-тестирование)
по дисциплине «Системное программное обеспечение»

Выполнил студент

группы 538:

_____ Семёнов Н.Ю.

Проверил

доцент кафедры КСУП:

_____ Коцубинский В.П.

Дата: «__» _____ 2020 г.

Томск 2020

Оглавление

1 Цель и задачи лабораторной работы	3
2 Ход лабораторной работы	4
3 Заключение	8

1 Цель и задачи лабораторной работы

Цель работы:изучить организацию тестирования в разработке ПО, и получить умения написания юнит-тестов.

Задачи:

1.Изучить организацию процесса тестирования в разработке ПО, виды тестирования, сроки их проведения и ответственных исполнителей.

2.Изучить основные атрибуты и классы библиотеки NUnitдля написания юнит-тестов.

3.Научиться рассчитывать цикломатическую сложность методов, классов и проектов, оценивать степень покрытия кода тестами.

4.Написать юнит-тесты для классов логики приложения с использованием библиотеки NUnit.

2 Ход лабораторной работы

Расчёт цикломатической сложности проекта NoteApp.

NoteApp (цикл. сложность проекта = 20):

1. Класс Note (цикл. сложность класса = 13)
 - Свойство Name (цикл. сложность = 4)
 - Свойство Category (цикл. сложность = 2)
 - Свойство Text (цикл. сложность = 2)
 - Свойство TimeCreation (цикл. сложность = 1)
 - Свойство TimeLastChange (цикл. сложность = 2)
 - Свойство Note (цикл. сложность = 1)
 - Свойство Clone (цикл. сложность = 1)
2. Класс Project (цикл. сложность класса = 5)
 - Свойство SortWithSelectionCategory (цикл. сложность = 5)
3. Класс ProjectManager (цикл. сложность класса = 2)
 - Свойство WritingToFile (цикл. сложность = 1)
 - Свойство ReadingFromFile (цикл. сложность = 1)

Исходный код юнит-теста, отвечающего за сериализацию:

```
[SetUp]
    public void InitNote()
    {
        _notesForSave = new Project();
        _notesForLoad = new Project();
        _actualList = new List<Note>();

        _notesForSave.Glossary.Add(new Note("Заголовок1",
Category.Work, "Текст1", DateTime.Today) { TimeLastChange =
DateTime.Today });
        _notesForSave.Glossary.Add(new Note("Заголовок3",
Category.Work, "Текст3", DateTime.Today) { TimeLastChange =
DateTime.Today });
```

```

        _notesForSave.Glossary.Add(new Note("Заголовок2",
Category.Work, "Текст2", DateTime.Today) { TimeLastChange =
DateTime.Today });

        _notesForSave.Glossary.Add(new Note("Заголовок4",
Category.Work, "Текст4", DateTime.Today) { TimeLastChange =
DateTime.Today });
    }

```

```

[Test(Description = "Тест сериализации")]
public void TestSerialize_CorrectValue()
{
    var expected = _notesForSave;
    var path = _path;
    var actualList = _actualList;

    ProjectManager.WritingToFile(expected, _path);

    var actual = _notesForLoad;

    JsonSerializer serializer = new JsonSerializer();

    using (StreamReader sr = new StreamReader(path))

    using (JsonReader reader = new JsonTextReader(sr))

        actual =
        (Project)serializer.Deserialize<Project>(reader);

    for (int i = 0; i < actual.Glossary.Count; i++)
    {
        actualList.Add(actual.Glossary[i]);
    }

    for (int i = 0; i < expected.Glossary.Count; i++)

```

```

    {
        Assert.AreEqual(expected.Glossary[i].Name,
actualList[i].Name);
        Assert.AreEqual(expected.Glossary[i].Category,
actualList[i].Category);
        Assert.AreEqual(expected.Glossary[i].Text,
actualList[i].Text);

Assert.AreEqual(expected.Glossary[i].TimeLastChange,
actualList[i].TimeLastChange);

Assert.AreEqual(expected.Glossary[i].TimeCreation,
actualList[i].TimeCreation);
    }
}

```

Время выполнения всех юнит-тестов представлено на Рисунке 2.1:

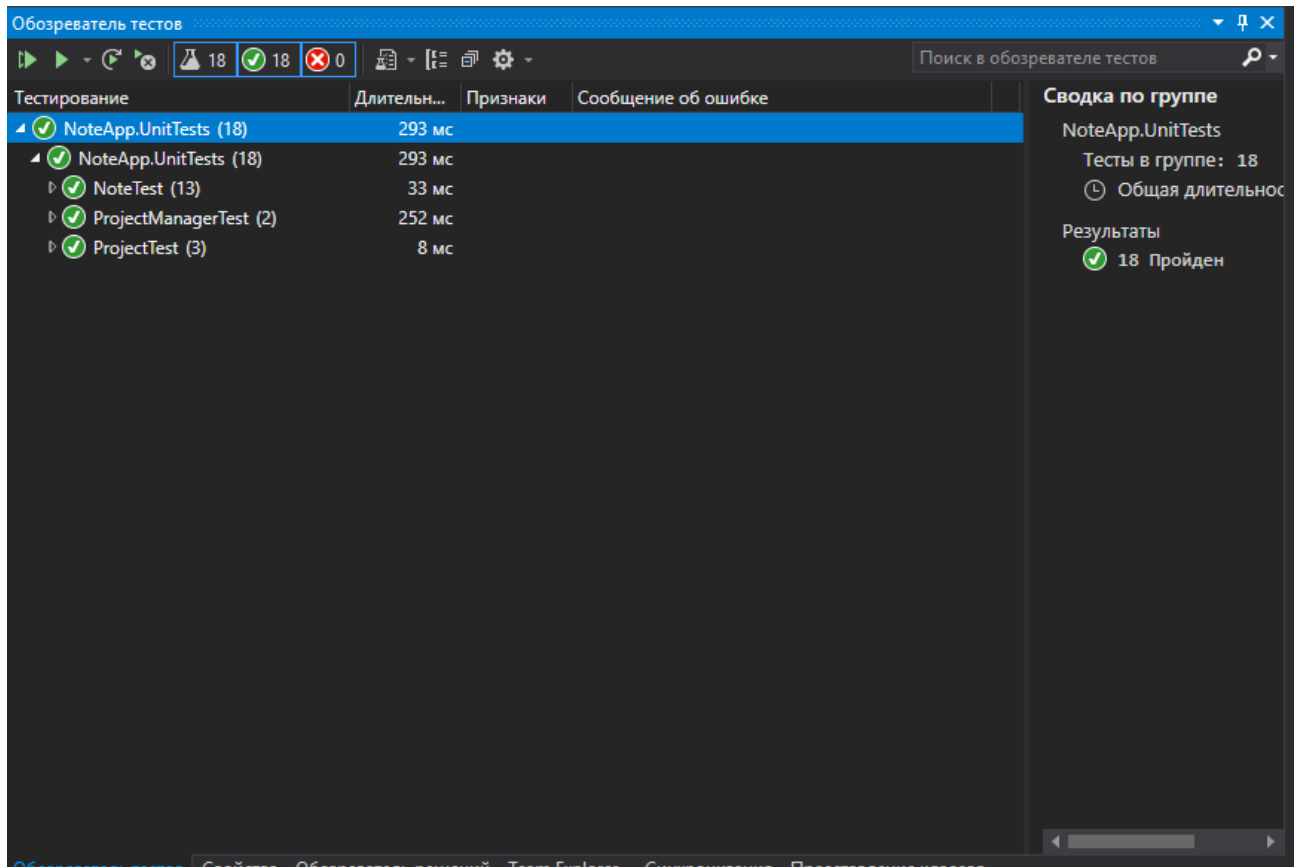


Рисунок 2.1 – Окно выполненных юнит-тестов

На рисунке 2.2 представлена текущая история коммитов ветки `develop`.

Создан сценарий сборки установочного пакета
SNY538 committed yesterday 4b2d964 <>

Добавлены тесты для сериализации и десериализации
SNY538 committed yesterday 5f52db9 <>

Commits on Nov 26, 2020

Добавлены отчёты 1,2,3
SNY538 committed 5 days ago 5602e25 <>

Commits on Nov 23, 2020

Добавлен проект NoteApp.UnitTests.
SNY538 committed 8 days ago 0ce48fe <>

Commits on Nov 18, 2020

Исправлена проблема с CategoryBox
SNY538 committed 13 days ago 13fafea <>

Добавлена защита от неправильного ввода
SNY538 committed 13 days ago ce56403 <>

Commits on Nov 17, 2020

Была создана адаптивная вёрстка, которая удовлетворяет ТЗ
SNY538 committed 14 days ago 6ff9303 <>

Commits on Nov 9, 2020

Добавлена библиотека Newtonsoft.Json.NET. Добавлен класс сериализации
SNY538 committed 22 days ago 240002b <>

Commits on Nov 8, 2020

Разработана логика приложения
SNY538 committed 23 days ago 1df9eca <>

Commits on Nov 4, 2020

Добавлен проект логики
SNY538 committed 27 days ago 2316470 <>

Создано решение проекта
SNY538 committed 27 days ago 6b72df1 <>

Initial commit
SNY538 committed 27 days ago Verified de1a189 <>

Рисунок 2.2 - История коммитов ветки `develop`.

3 Заключение

В ходе данной лабораторной работы были разработаны юнит-тесты на каждый класс бизнес логики проекта. Была рассчитана цикломатическая сложность проекта логики, а также рассмотрен процесс организации тестирования в ходе разработки ПО.

Все поставленные задачи на лабораторную работы были выполнены.