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:

\_\_\_\_\_\_\_\_ Асмоловский А.В.

Проверил

доцент кафедры КСУП:

\_\_\_\_\_\_\_\_ Коцубинский В.П.

Дата: «\_\_\_»\_\_\_\_\_\_\_\_2020 г.

Томск 2020

Оглавление

[**1 Цель и задачи лабораторной работы** 3](#_Toc57135756)

[**2 Ход лабораторной работы** 4](#_Toc57135757)

[**3 Заключение** 7](#_Toc57135758)

**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 NUnit ,3

4.для 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 (цикл. сложность проекта = 16):

1. Класс Note (цикл. сложность класса =13)
   1. Свойство Name (цикл. сложность = 4)
   2. Свойство Category(цикл. сложность = 2)
   3. Свойство Text (цикл. сложность = 2)
   4. Свойство CreatingTime (цикл. сложность = 1)
   5. Свойство LastChangeTime (цикл. сложность = 2)
   6. Свойство Note (цикл. сложность = 1)
   7. Свойство Clone(цикл. сложность = 1)
2. Класс Project (цикл. сложность класса = 1)
3. Класс ProjectManager(цикл. сложность класса = 2)
   1. Свойство SaveToFile(цикл. сложность = 1)
   2. Свойство LoadFromFile(цикл. сложность = 1)

Исходный код юнит-тестов, покрывающих свойство Name класса NoteApp:

private Note \_note;

[SetUp]

public void InitNote()

{

\_note = new Note();

}

[Test(Description = "Тест геттера Name")]

public void TestNameGet\_CorrectValue()

{

var expected = "Заметка";

\_note.Name = expected;

var actual = \_note.Name;

Assert.AreEqual(expected, actual, "Геттер Name возвращает неправильное название заметки");

}

[Test(Description = "Тест неправильного названия заметки больше 50 символов")]

public void TestNameSet\_Longer50Symbols()

{

var wrongName = "Заметка-Заметка-Заметка-Заметка-Заметка-Заметка-Заметка-Заметка-Заметка-Заметка-Заметка-Заметка";

Assert.Throws<ArgumentException>(() => { \_note.Name = wrongName; },

"Должно возникать исключение, если названия заметки длиннее 50 символов");

}

[Test(Description = "Присваивание пустой строки в поле Name")]

public void TestNameSet\_EmptyStringValue()

{

var expected = "";

\_note.Name = expected;

var actual = "NoName";

Assert.AreNotEqual(expected, actual, "Геттер Name возвращает неправильное название заметки");

}

[Test(Description = "Присваивание нормального названия заметки в поле Name")]

public void TestNameSet\_CorrectValue()

{

var expected = "Нормальная заметка";

\_note.Name = expected;

var actual = \_note.Name;

Assert.AreEqual(expected, actual, "Геттер Name возвращает неправильное название заметки");

}

Время выполнения всех юнит-тестов представлено на Рисунке 2.1:

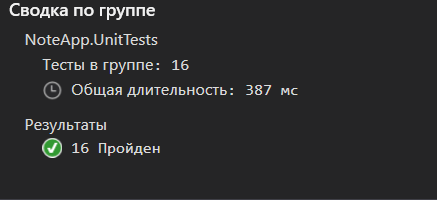


Рисунок 2.1 — Окно выполненных юнит-тестов

Просмотрим историю коммитов ветки develop, чтобы убедиться, что мы все сделали верно:

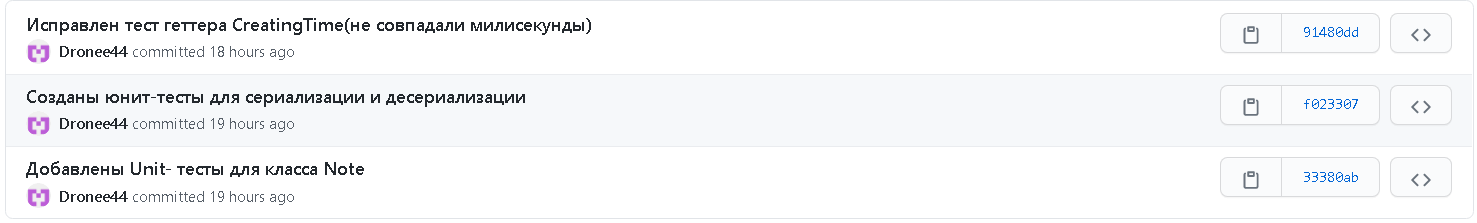


Рисунок 2.2 - История коммитов ветки develop.

# **3 Заключение**

В ходе данной лабораторной работы мы разработали юнит-тесты на каждый класс в нашей бизнес логики, научились высчитывать цикломатическую сложность проекта, а также организацию процесса тестирования в ходе разработки ПО.

Задачи на лабораторную работы выполнены.