

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

По лабораторной работе №4

«Юнит-тестирование»

по дисциплине «Новые технологии в программировании»

Выполнил:

студент гр. 587-1

\_\_\_\_\_ Монгуш С.В.

"\_\_" \_\_\_\_\_ 2020 г.

Проверил:

Доцент кафедры КСУП:

\_\_\_\_\_ Горяинов А.Е.

"\_\_" \_\_\_\_\_ 2021 г.

# 1 Введение

## 1.1 Цель работы

В данной работе необходимо изучить организацию тестирования в разработке ПО, и получить умения написания юнит-тестов.

## 1.2 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 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 Цикломатическая сложность проекта

Цикломатической сложностью называют количество линейно независимых алгоритмических маршрутов через программный код. Расчет цикломатической сложности позволяет определить все требуемые тесты (тестовые случаи) для того или иного класса. Она зависит от ветвлений алгоритма, содержащихся в нем циклов и операторов перехода. Также на цикломатическую сложность влияют вызовы других методов.

## 3 Выполнение юнит-тестов и история коммитов

В процессе проверки и отладки программы бывает необходимо знать время выполнения автоматических проверок. Пакеты библиотеки NUnit позволяют наблюдать не только статус тестов, но и время, затраченное на их выполнение. На рисунке 3.1 представлена информация о проведении тестов.

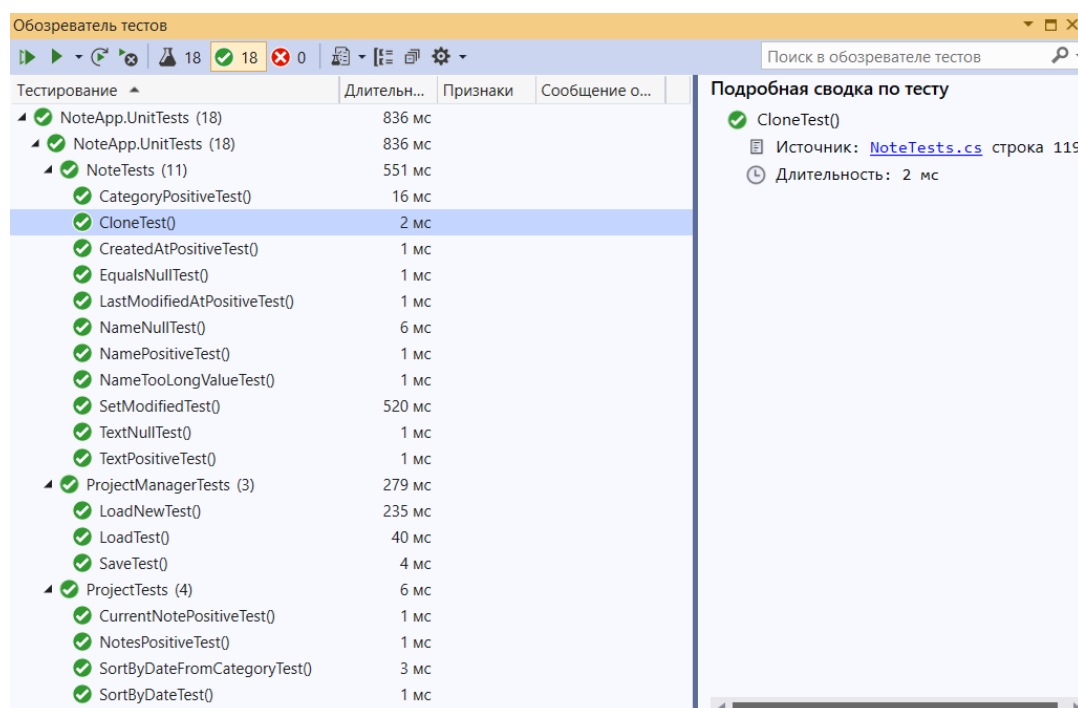
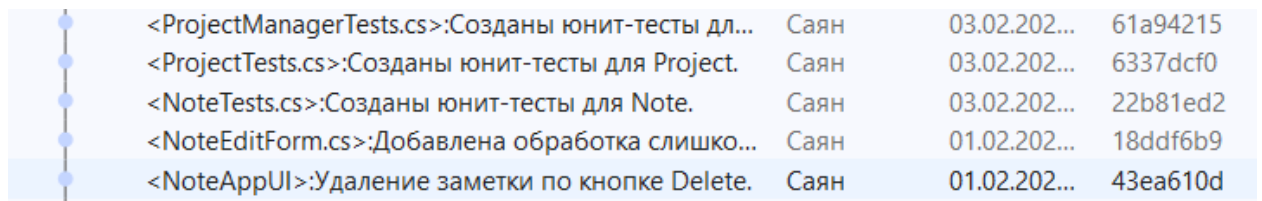


Рисунок 3.1 – Результат выполнения юнит-тестов.

В Visual Studio Community 2019 нельзя проверить на покрытие кода юнит-тестами, поэтому проверка была проведена в среде разработки Rider как представлено на рисунке 3.2.

▼ [ ] NoteApp	91%	8/85
▼ ( ) NoteApp	91%	8/85
▼ ( ) Models	96%	2/52
▼ [ ] Note	97%	1/31
▶ [ ] Name	100%	0/8
▶ [ ] Category	100%	0/2
▶ [ ] Text	100%	0/2
▶ [ ] Created	100%	0/2
▶ [ ] LastModi	100%	0/2
▶ [ ] SetModif	100%	0/3
▶ [ ] Clone()	100%	0/3
▶ [ ] Note()	100%	0/4
▶ [ ] Equals(N	80%	1/5
▼ [ ] Project	95%	1/21
▶ [ ] CurrentM	100%	0/2
▶ [ ] Notes	100%	0/2
▶ [ ] SortByD	100%	0/4
▶ [ ] SortByD	100%	0/5
▶ [ ] Project()	100%	0/1
▶ [ ] Equals(P	86%	1/7
▼ [ ] ProjectManager	82%	6/33
▶ [ ] AppFolder	100%	0/1
▶ [ ] NotesFile	100%	0/1
▶ [ ] Current	100%	0/1
▶ [ ] Create()	100%	0/3
▶ [ ] ProjectMana	100%	0/3
▶ [ ] ProjectMana	100%	0/2
▶ [ ] Load()	75%	3/12
▶ [ ] Save(Project	70%	3/10

Рисунок 3.2 – Покрытие кода юнит-тестами.

A screenshot of a Git commit history table. The table has five columns: a vertical line with blue dots on the left, a description of the commit, the author's name, the date, and the commit hash. The rows are: 1. <ProjectManagerTests.cs>:Созданы юнит-тесты дл... Саян 03.02.202... 61a94215; 2. <ProjectTests.cs>:Созданы юнит-тесты для Project. Саян 03.02.202... 6337dcf0; 3. <NoteTests.cs>:Созданы юнит-тесты для Note. Саян 03.02.202... 22b81ed2; 4. <NoteEditForm.cs>:Добавлена обработка слишко... Саян 01.02.202... 18ddf6b9; 5. <NoteAppUI>:Удаление заметки по кнопке Delete. Саян 01.02.202... 43ea610d.

•	<ProjectManagerTests.cs>:Созданы юнит-тесты дл...	Саян	03.02.202...	61a94215
•	<ProjectTests.cs>:Созданы юнит-тесты для Project.	Саян	03.02.202...	6337dcf0
•	<NoteTests.cs>:Созданы юнит-тесты для Note.	Саян	03.02.202...	22b81ed2
•	<NoteEditForm.cs>:Добавлена обработка слишко...	Саян	01.02.202...	18ddf6b9
•	<NoteAppUI>:Удаление заметки по кнопке Delete.	Саян	01.02.202...	43ea610d

Рисунок 3.3 – История коммитов разработки проекта тестов

## Заключение

Проведенное юнит-тестирование показало корректность работы логики проекта разрабатываемого приложения. Необходимые тесты были организованы с помощью библиотеки тестирования NUnit с использованием инструментов, соответствующих логике работы тестируемых модулей. Результаты обработаны и представлены в требуемой форме.