**Лабораторная работа №6.**

**Тестирование событий**

**Цель работы**

Приобретение практических навыков использования IoC контейнера Castle Windsor для внедрения зависимости; практика использования шаблонов проектирования; практика использования тестового каркаса NUnit, практика использования изолирующего каркаса NSubstitute.

**Задание на лабораторную работу**

1. Подготовить учебный проект

2. Подключить в проект пакет Castle Windsor

3. Реализовать паттерна команда

4. Добавить и сконфигурировать IoC контейнера

5. Добавить декоратор для команд

6. Добавить декоратора для перехвата исключений

7. Реализовать автономные тесты для разработанных классов

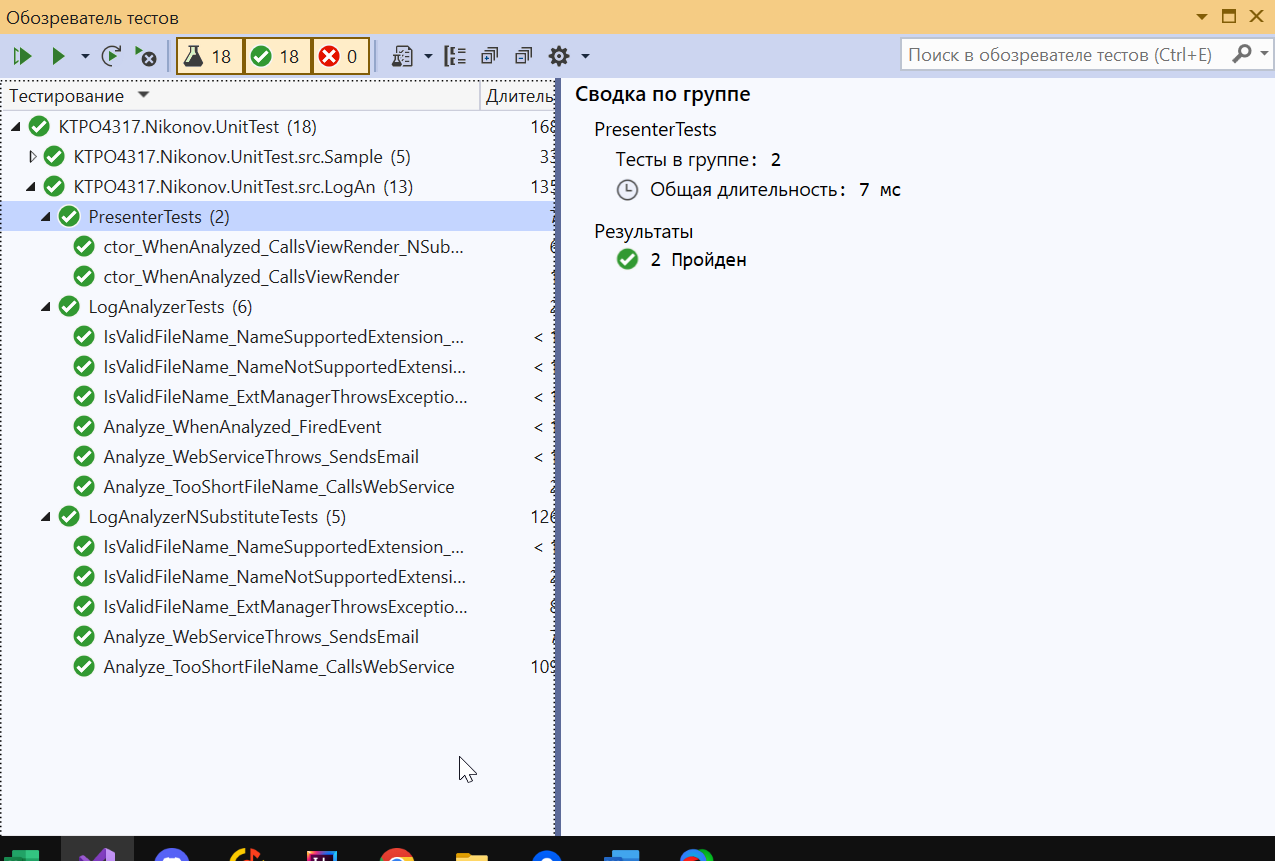
8. На каждом шаге делайте снимки исходного кода создаваемых или изменяемых классов и тестов, окна «Результаты тестов» и «Обозреватель решения» и сохраните в документе MS Word.

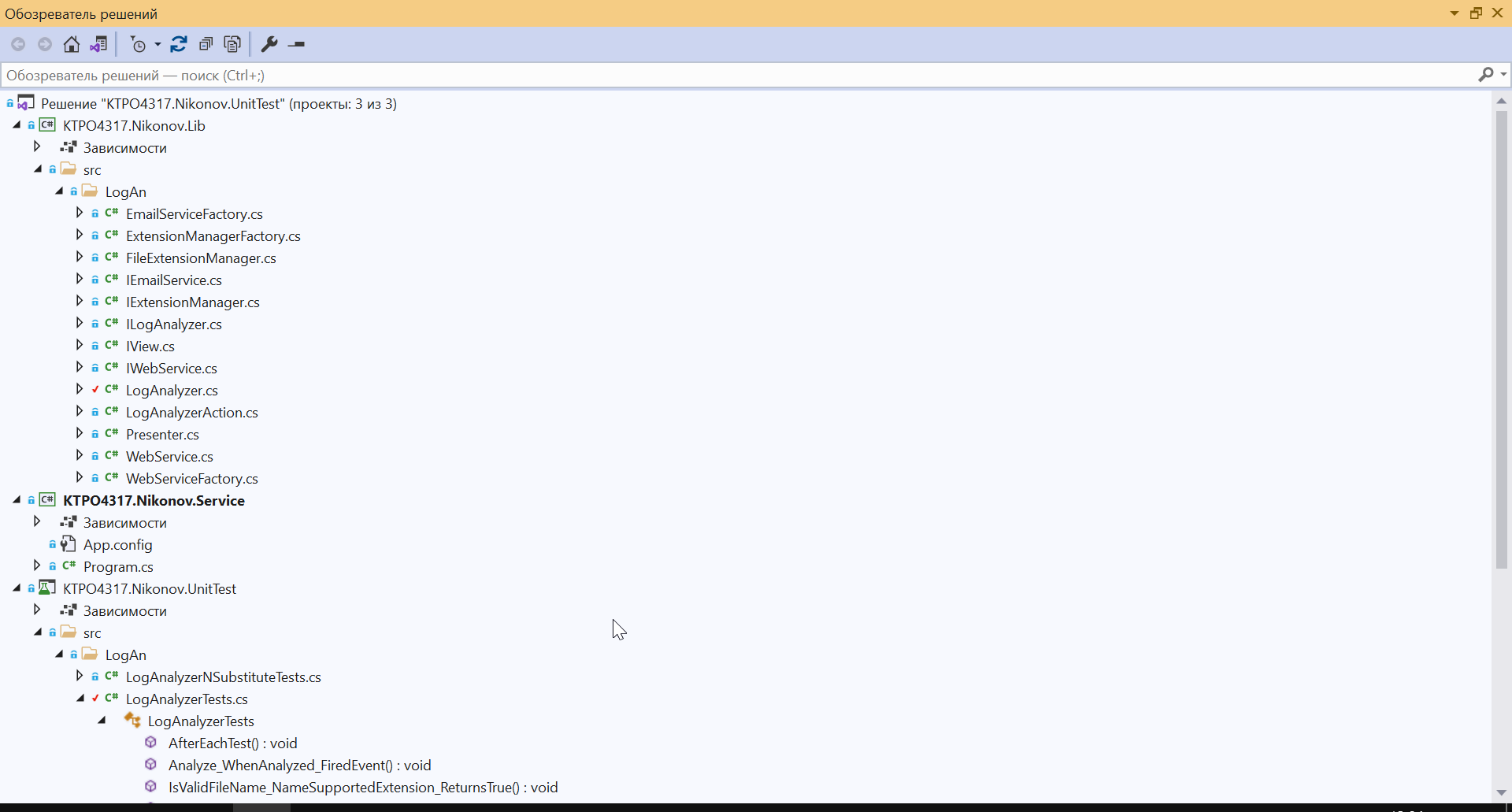
9. Оформить отчет

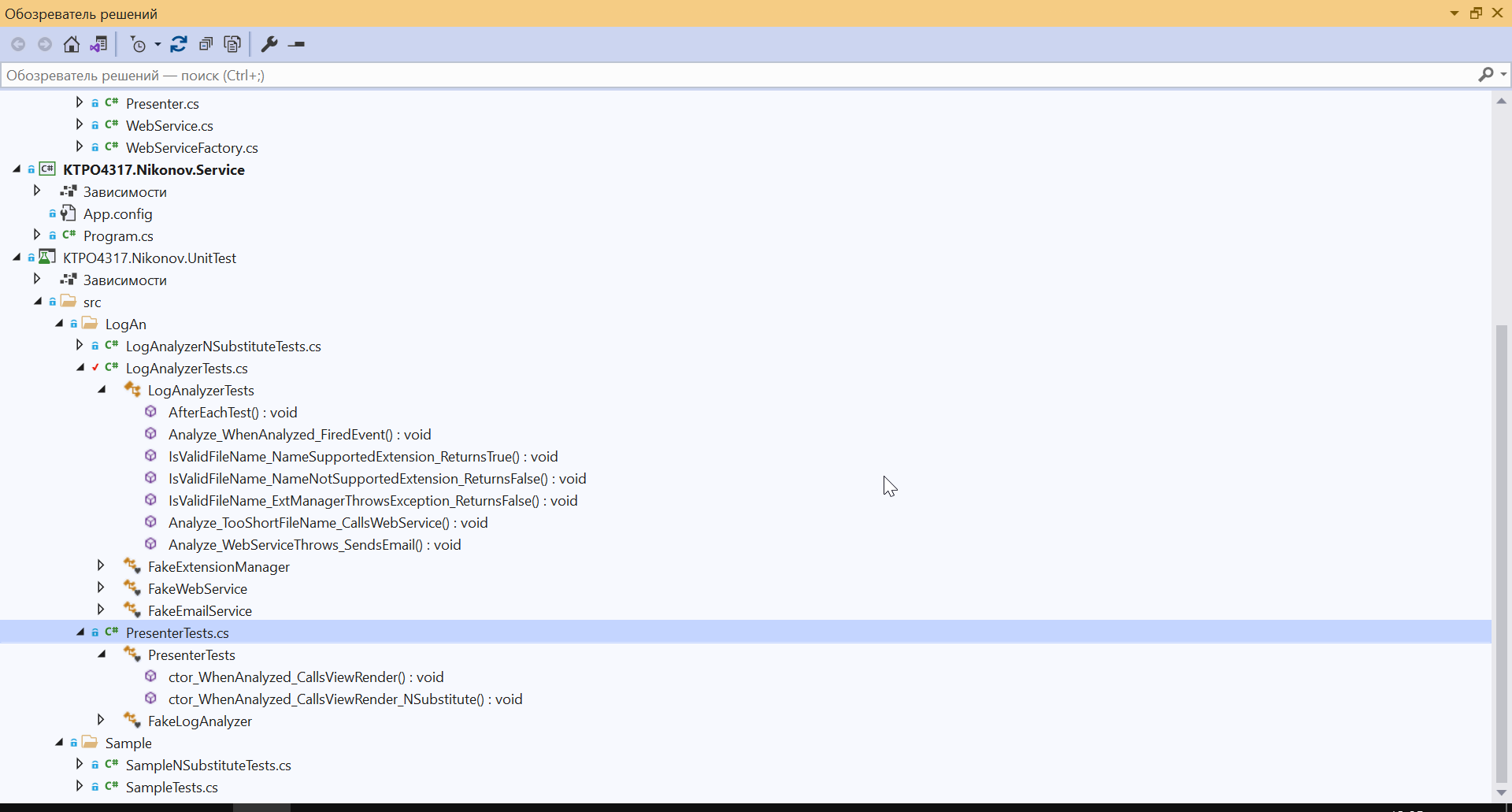
**Ход работы**

**1. Подготовка проекта**

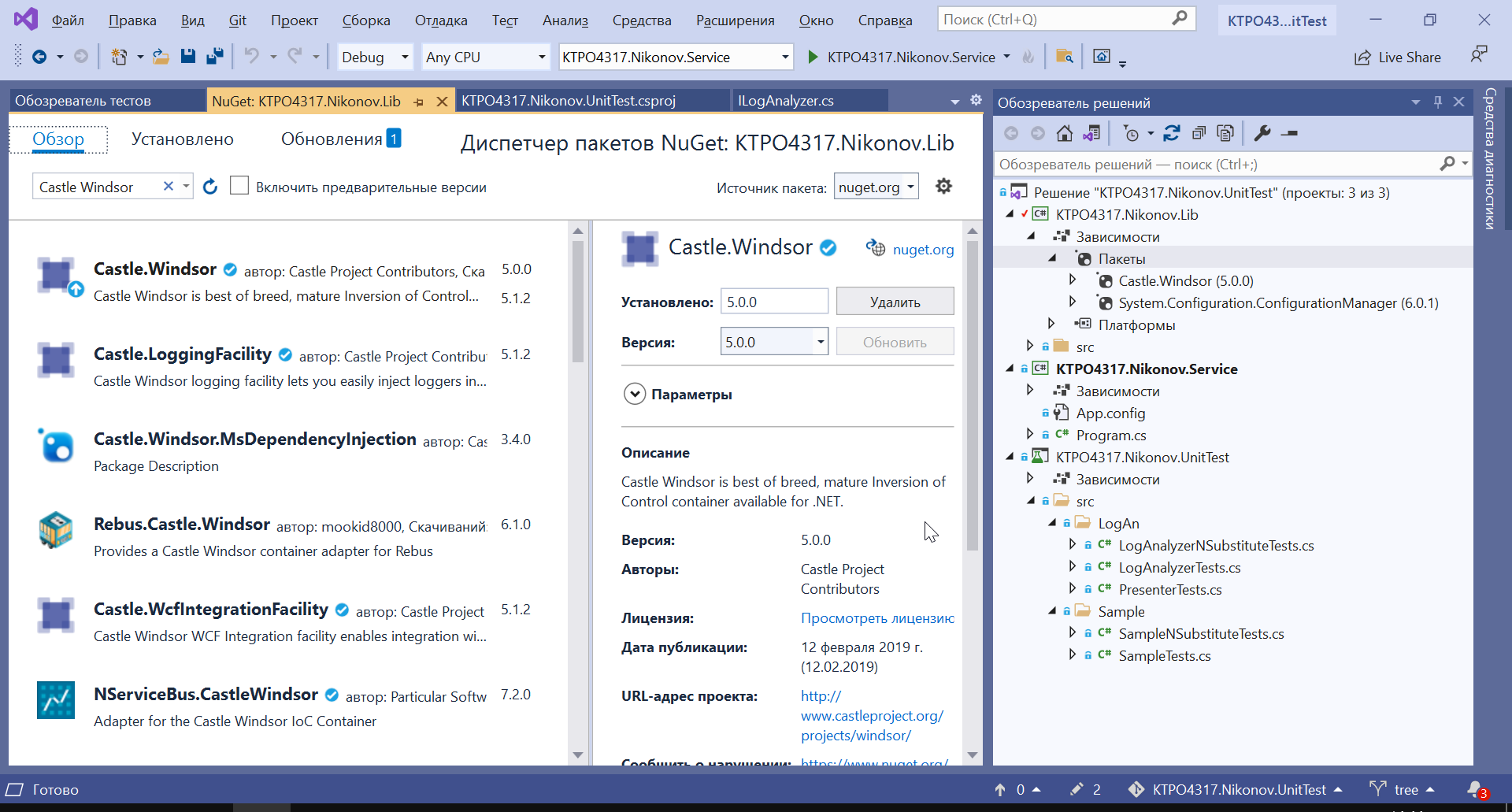
Для выполнения данной лабораторной работы возьмите решение, полученное в результате выполнения лабораторной работы №5. Выполните тесты. Зафиксируйте исходное состояние окна «Обозреватель решения», окна «Результаты тестов»: и сохраните в документе MS Word.







**2. Подключить в проект «.Lib» IoC контейнер Castle Windsor**

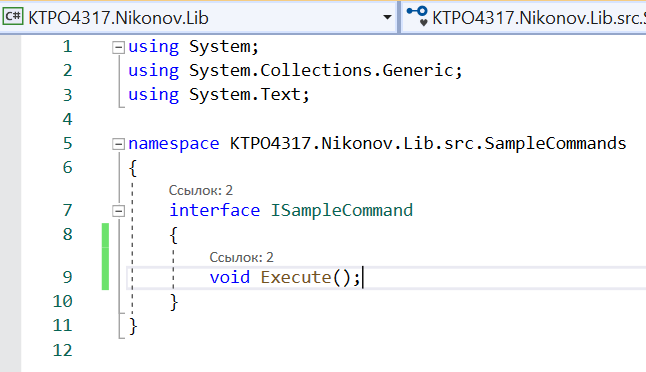


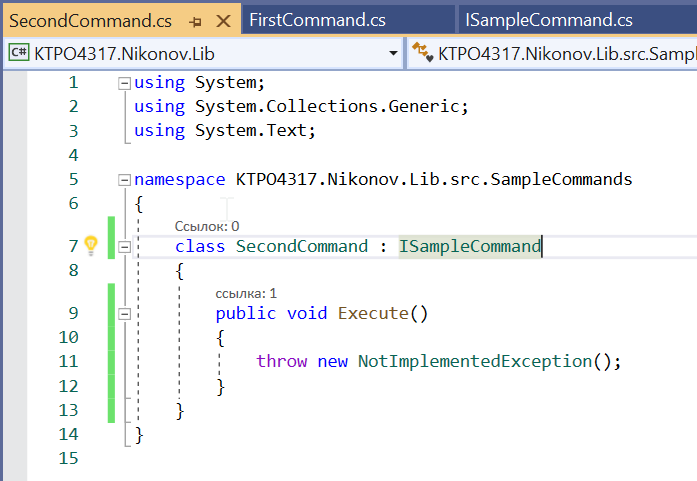
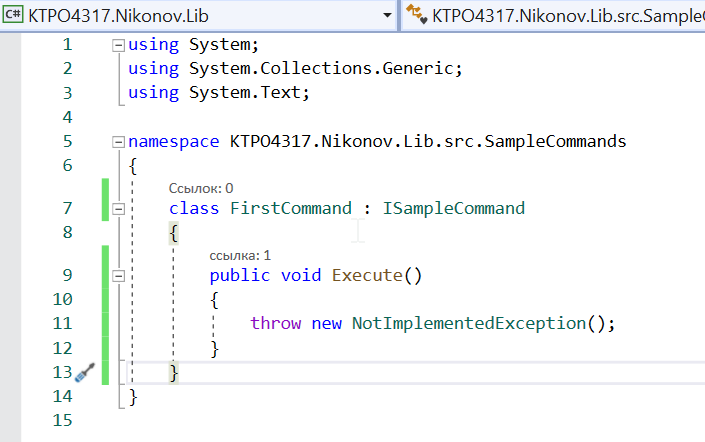
**3. Реализация паттерна команда**

В проекте «.Lib» создайте папку «src/SampleCommands»

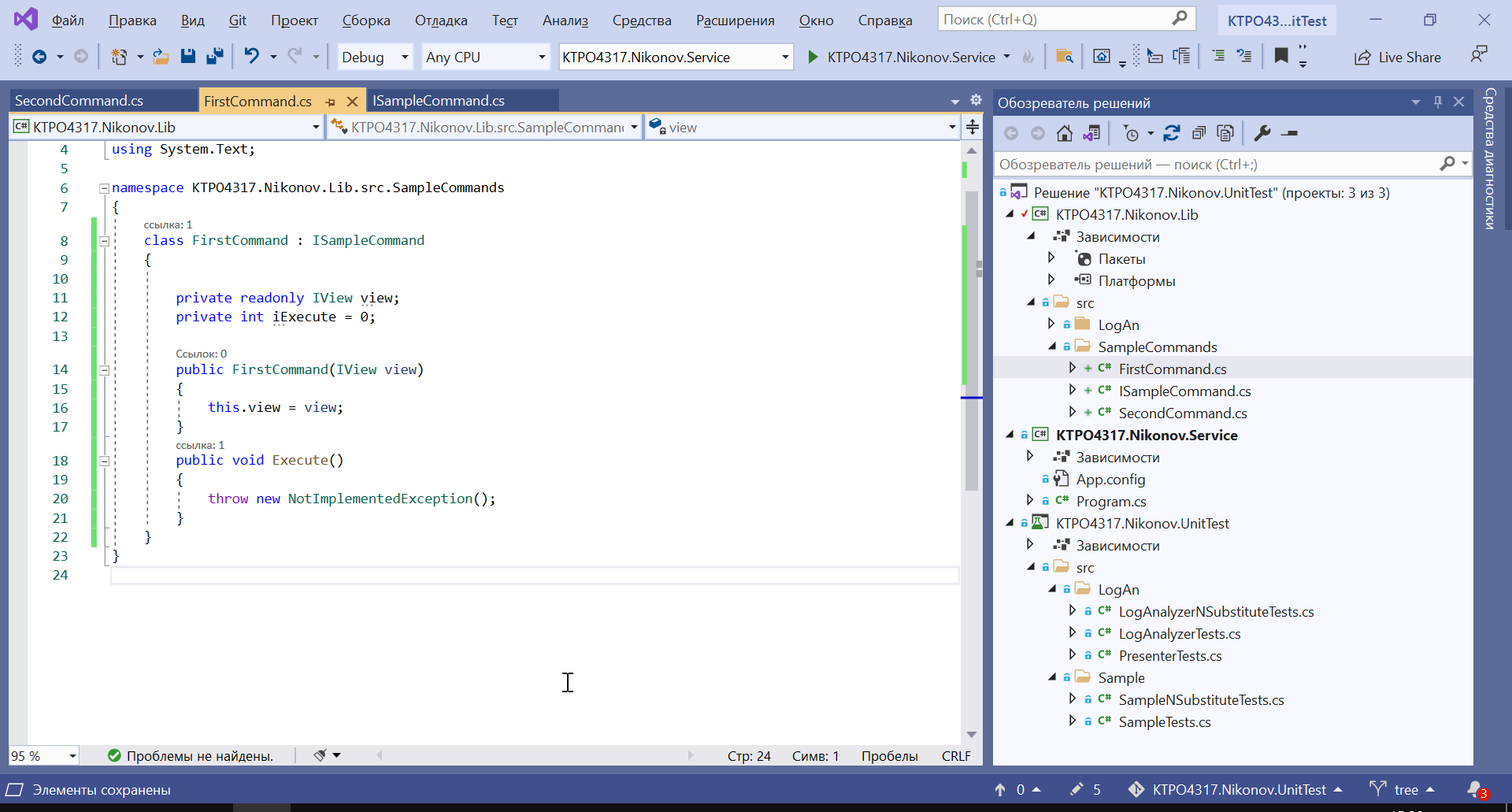
Создайте в ней интерфейс соответствующий шаблону команда:

Добавьте в этой же папке два класса, реализующие этот интерфейс: FirstCommand и SecondCommand

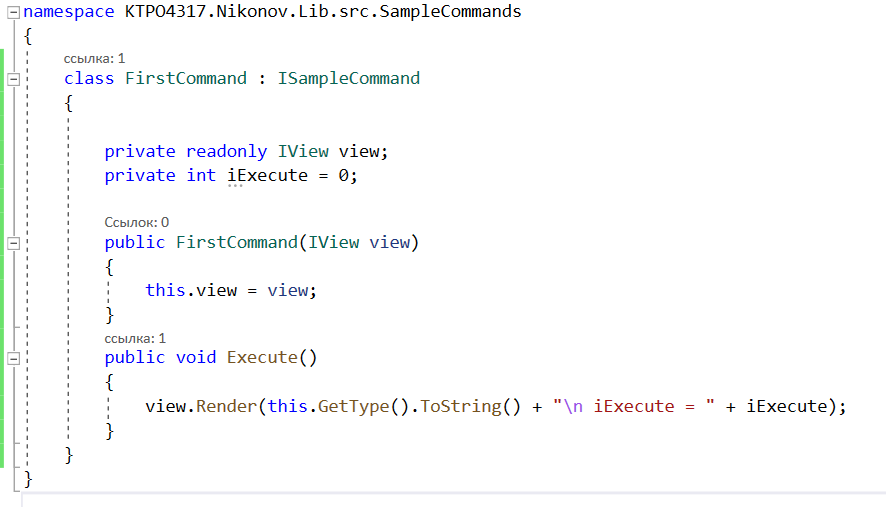




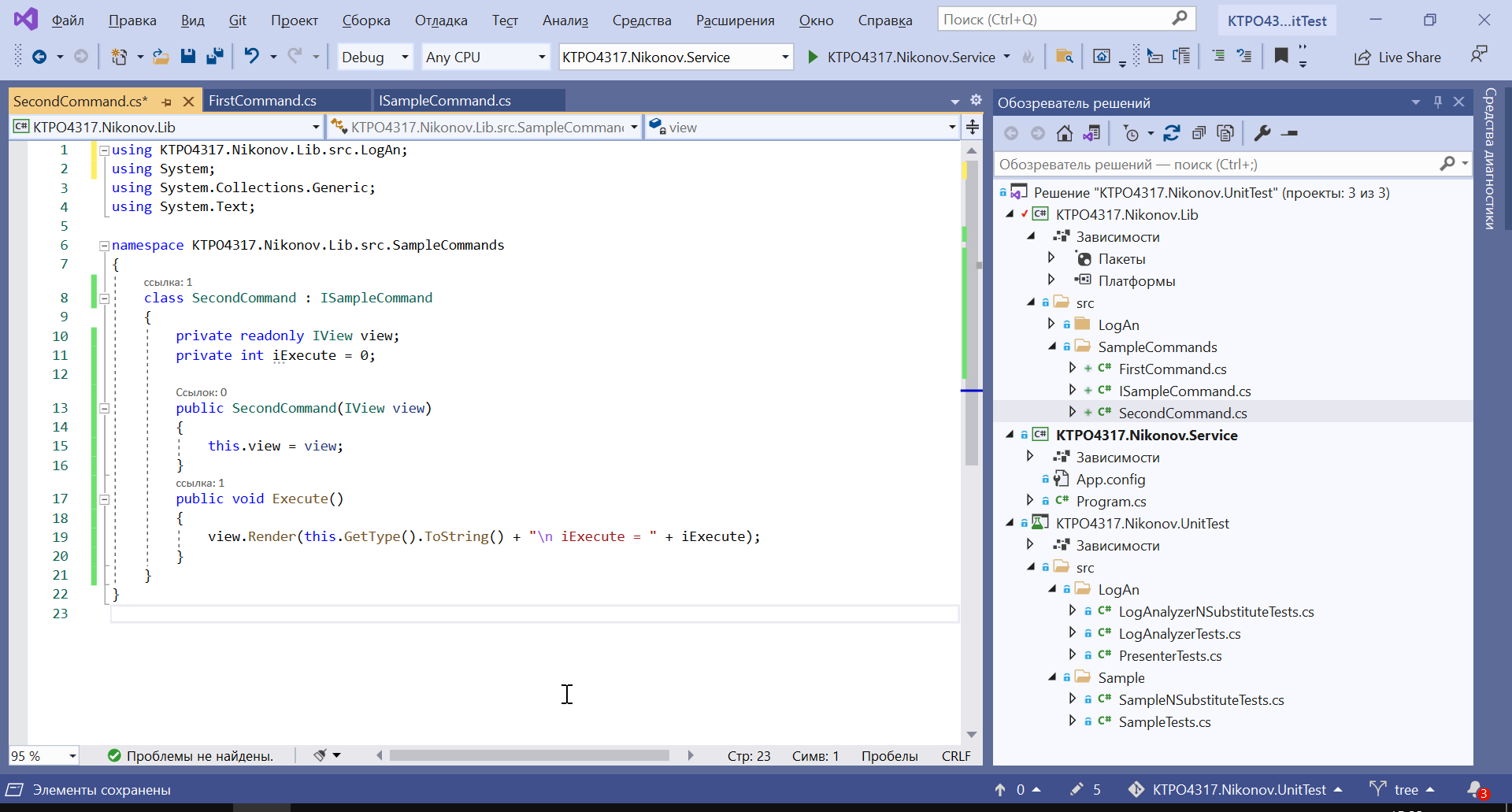
Пусть команда должна уметь выводить имя своего класса, и количество раз вызова метода Execute() для экземпляра класса. Вывод будем осуществлять с помощью интерфейса IView, созданного в предыдущей лабораторной работе. Будем использовать внедрение зависимости через конструктор. Для подсчета количества вызовов организуем поле – счетчик.



Добавьте реализацию метода Execute, добавить приращение счетчика и вызов метода IView.Render().



Аналогичным образом реализуйте SecondCommand.



В проекте «.Service» создайте папку «src\Views»

В этой папке создайте класс ConsoleView, реализующий интерфейс View.

Зафиксируйте в отчете код классов и состояние окна Обозреватель решения.

