Лабораторная работа №4.

Использование изолирующих каркасов.

**Цель работы**

Приобретение практических навыков использования изолирующих каркасов для создания объектов заглушек и подставных объектов при автономном тестировании модулей, практика использования тестового каркаса NUnit, практика использования изолирующего каркаса NSubstitute.

**Задание на лабораторную работу**

1. Подготовить учебный проект

2. Подключить изолирующий каркас

3. Изучить примы использования NSubstitute, реализовав демонстрационные тесты.

4. Создать реализацию тестов из лабораторной работы №2, заменив рукописные поддельные объекты на динамические.

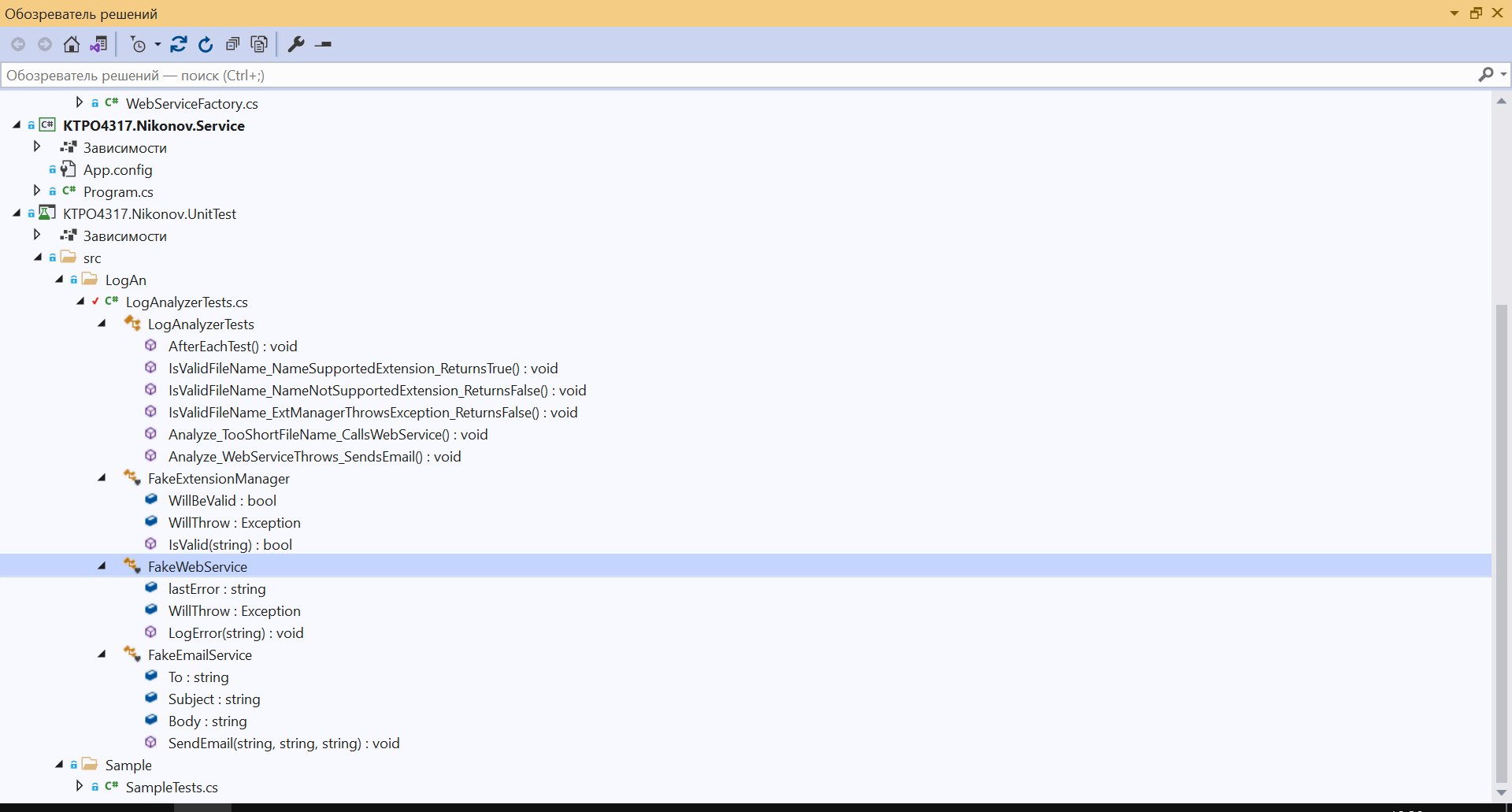
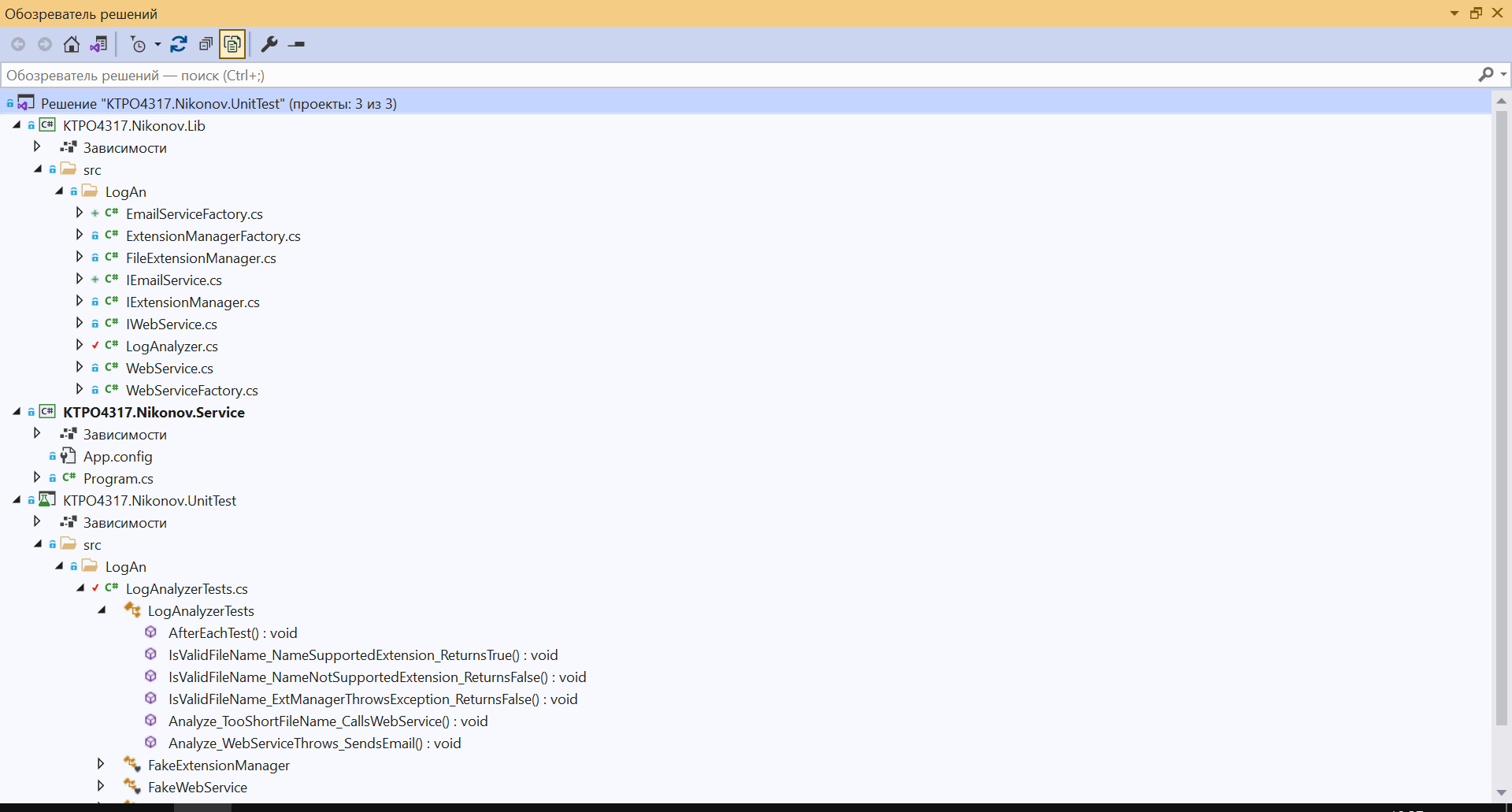
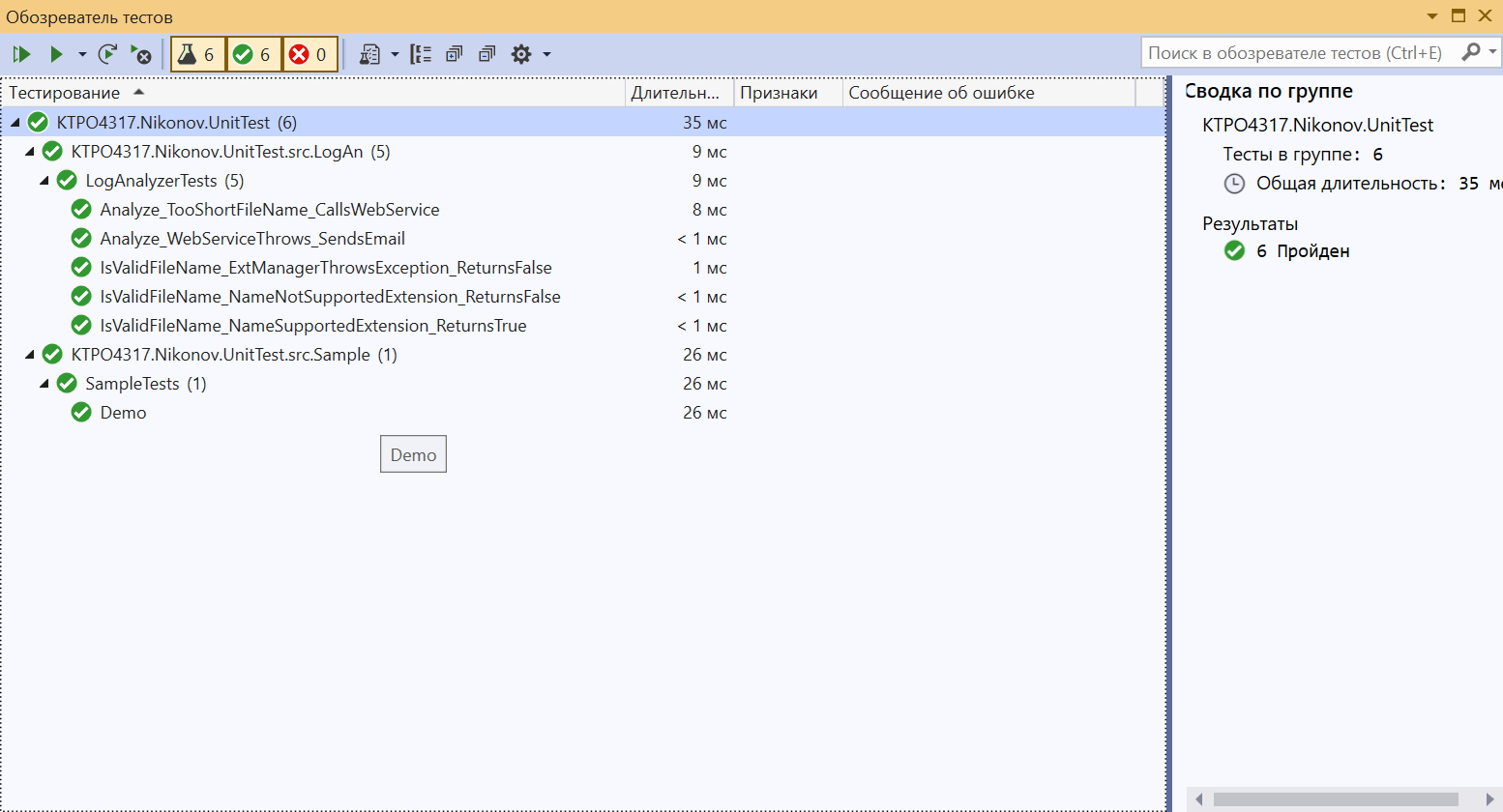
5. Создать реализацию тестов из лабораторной работы №3, заменив рукописные поддельные объекты на динамические.

6. 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.

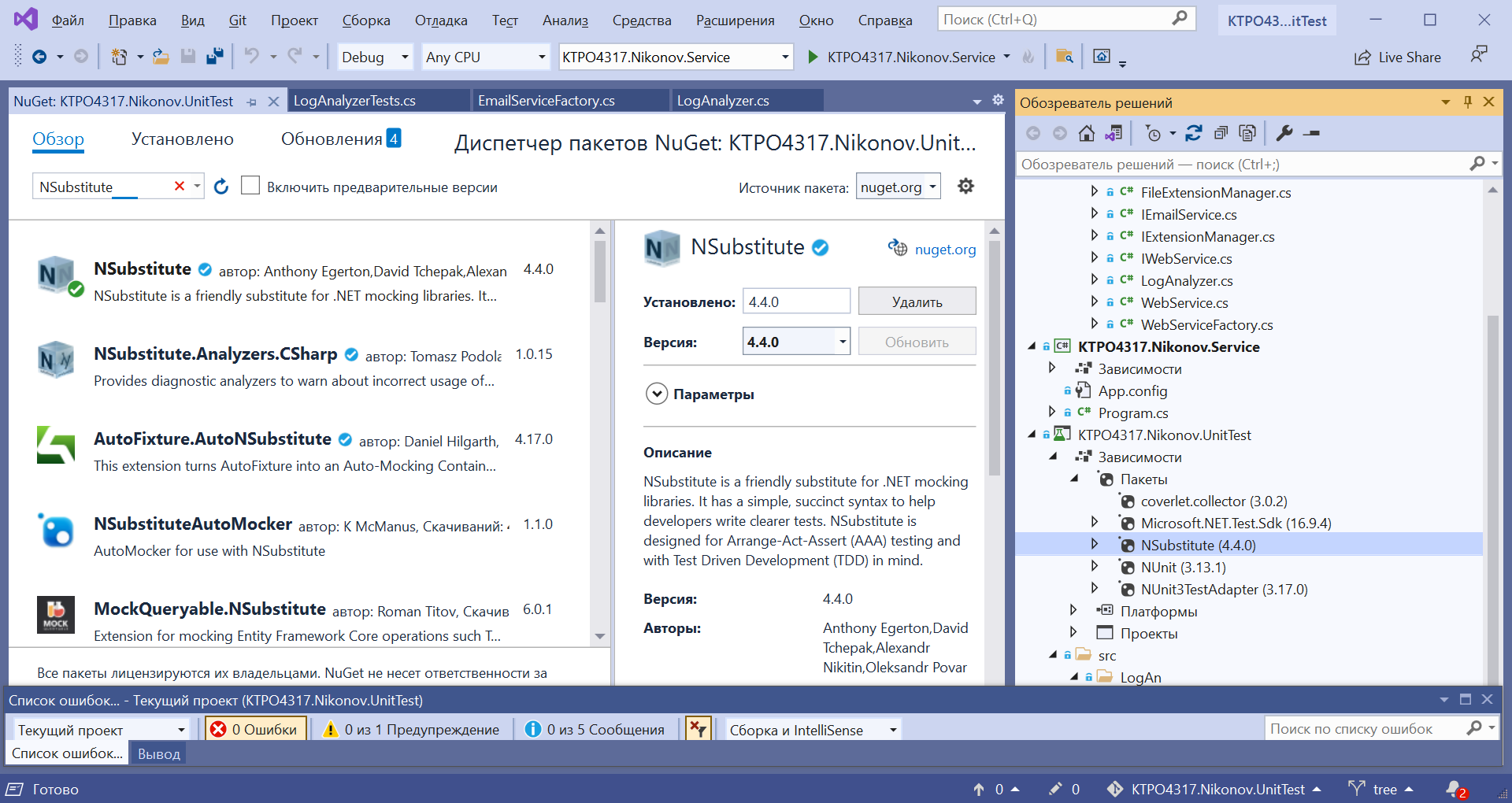
**Ход работы:**

**1. Подготовка проекта**

Для выполнения данной лабораторной работы возьмите решение, полученной в результате выполнения лабораторной работы №3. Выполните тесты.

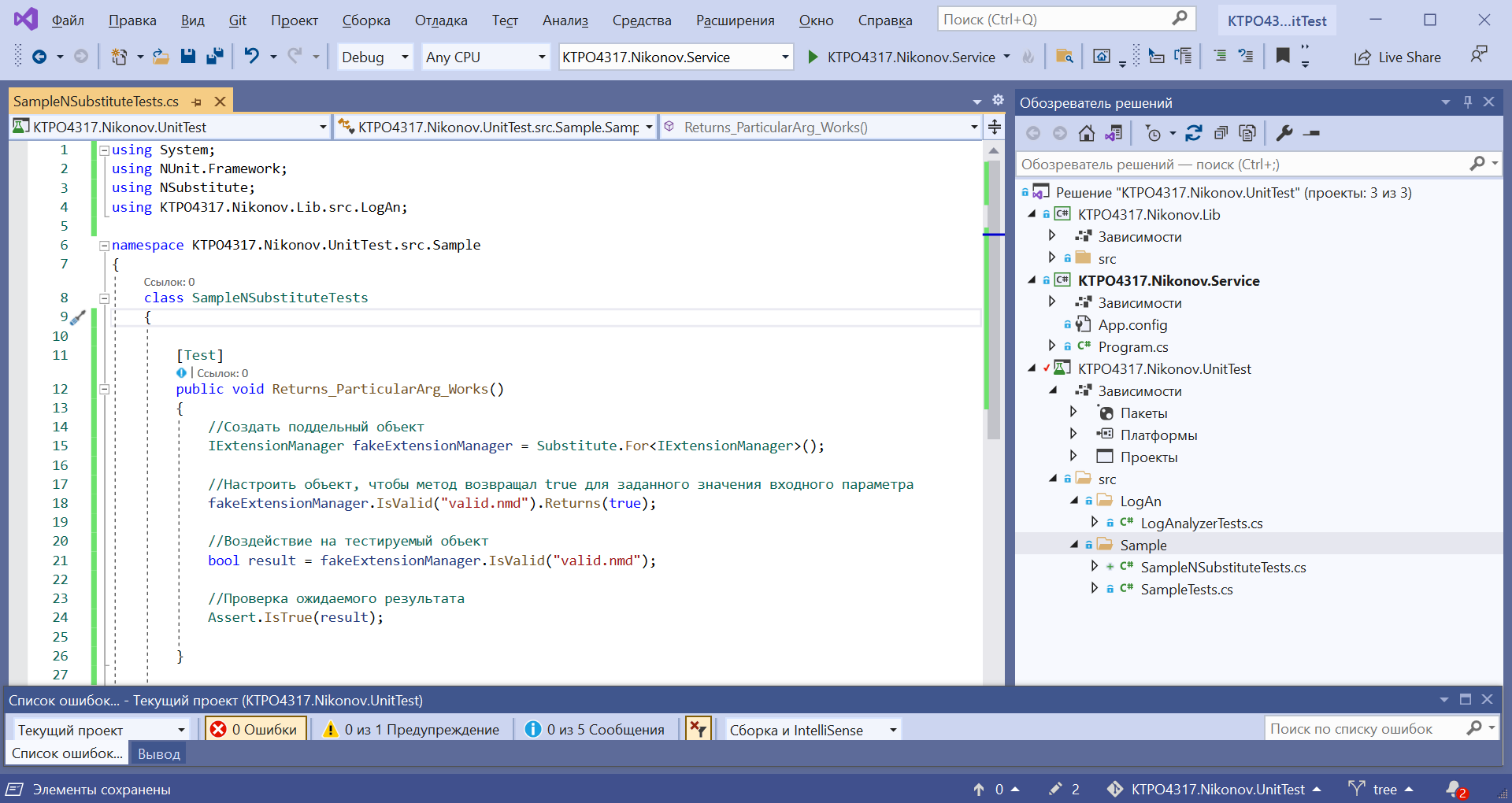


**2. Подключить в проект «.UnitTest» изолирующий каркас NSubstitute**

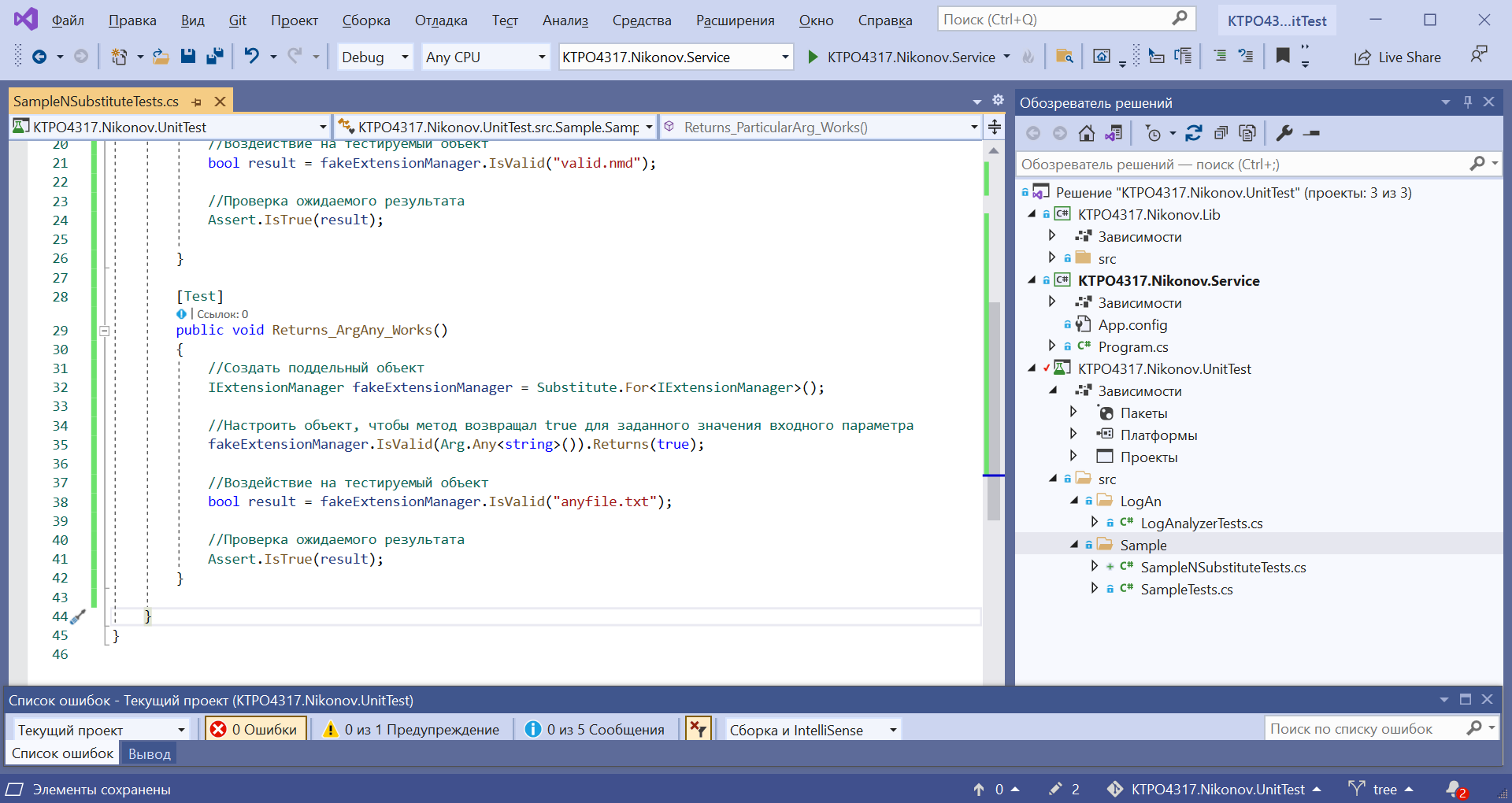


**3. Подделка значений, возвращаемых методом**

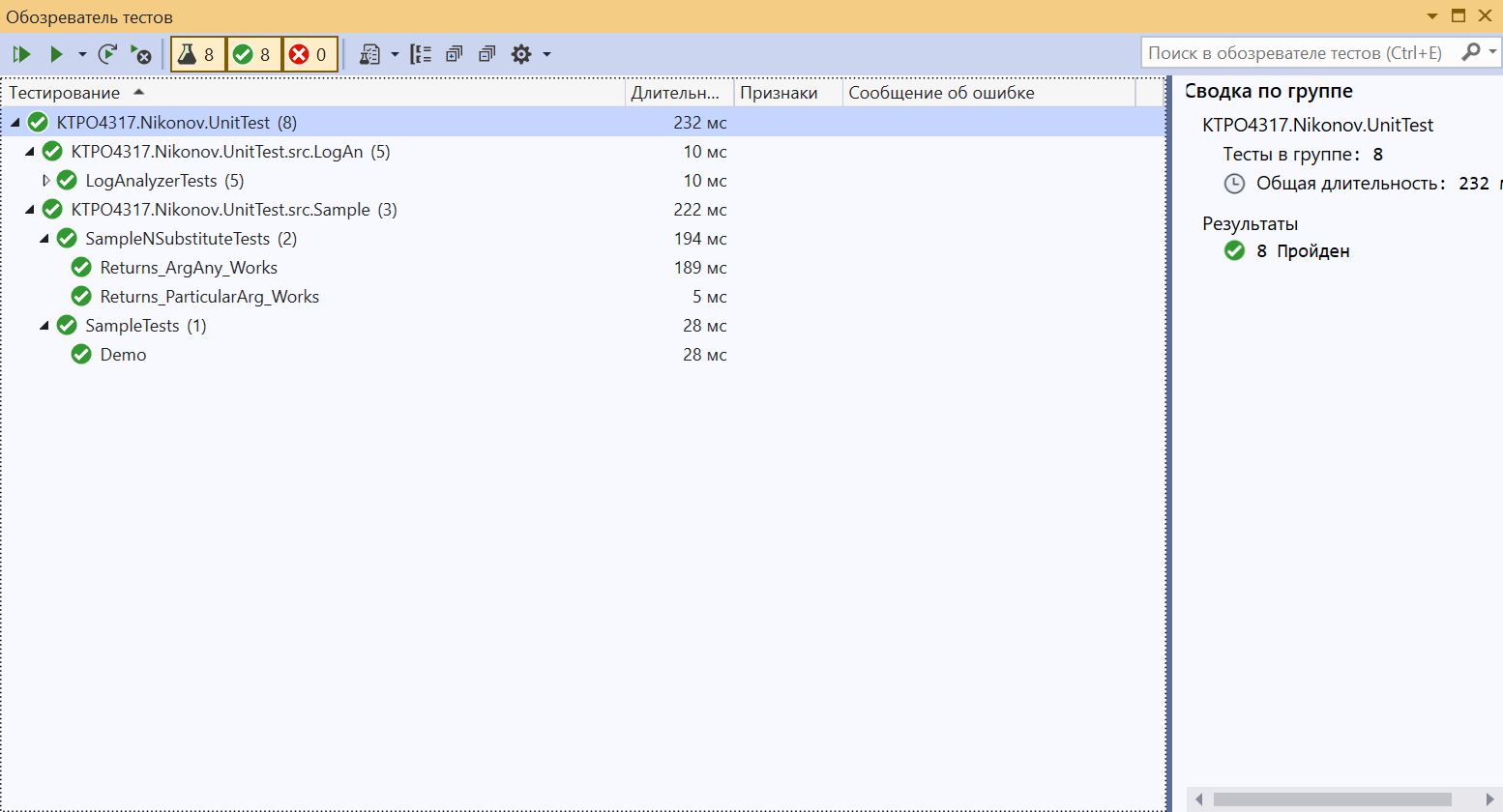
Добавьте в папку Sample тестовый класс SampleNSubstituteTests. Добавьте в класс SampleNSubstituteTests тестовый метод Returns\_ParticularArg\_Works(), который демонстрирует настройку возврата значения из поддельного объекта в зависимости от аргумента тестируемого метода.

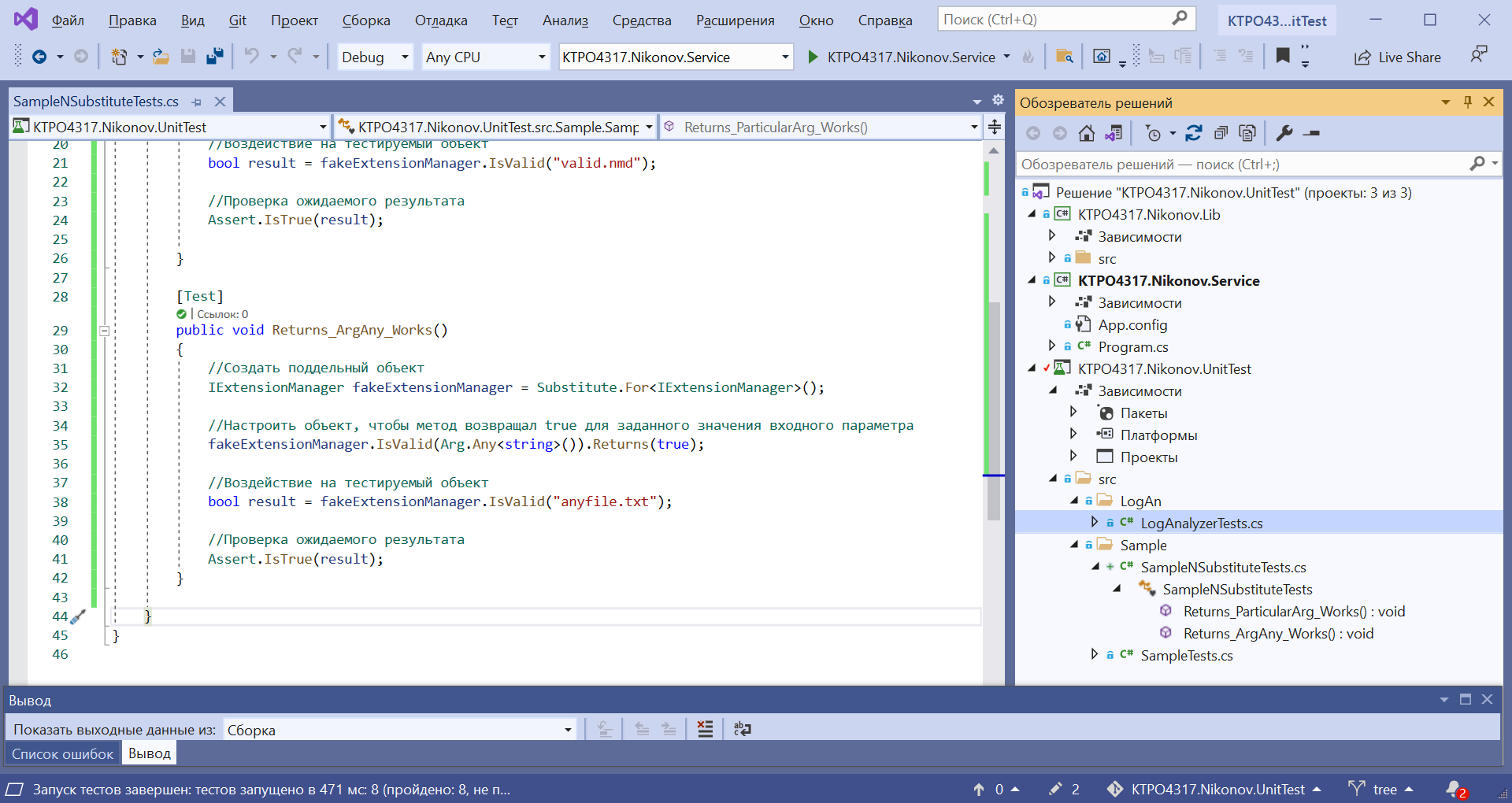


Добавьте в класс SampleNSubstituteTests тестовый метод Returns\_ArgAny\_Works(), который демонстрирует настройку возврата значения из поддельного объекта, когда возвращаемый результат не зависит от аргумента тестируемого метода.



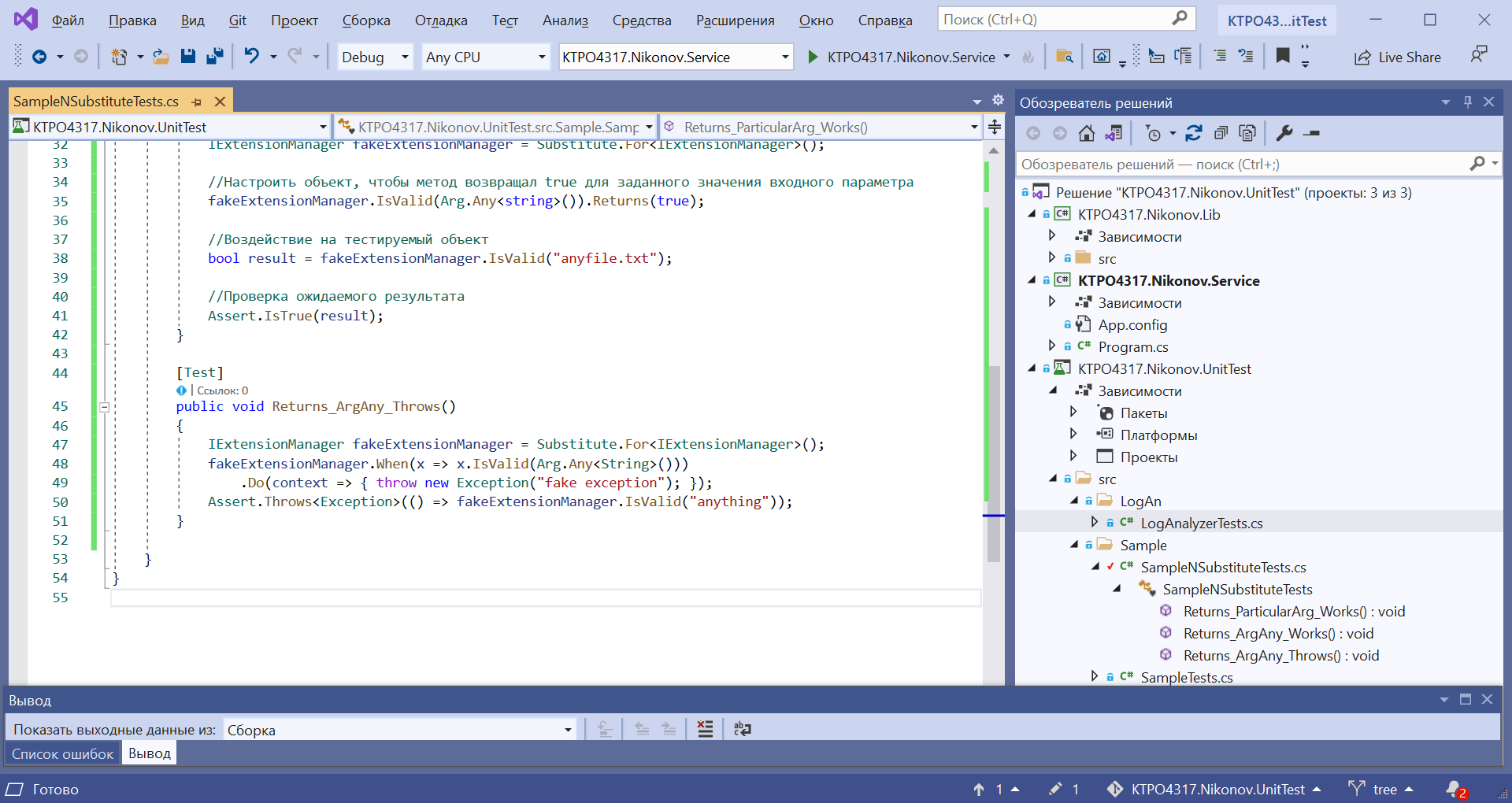
Выполните тесты.



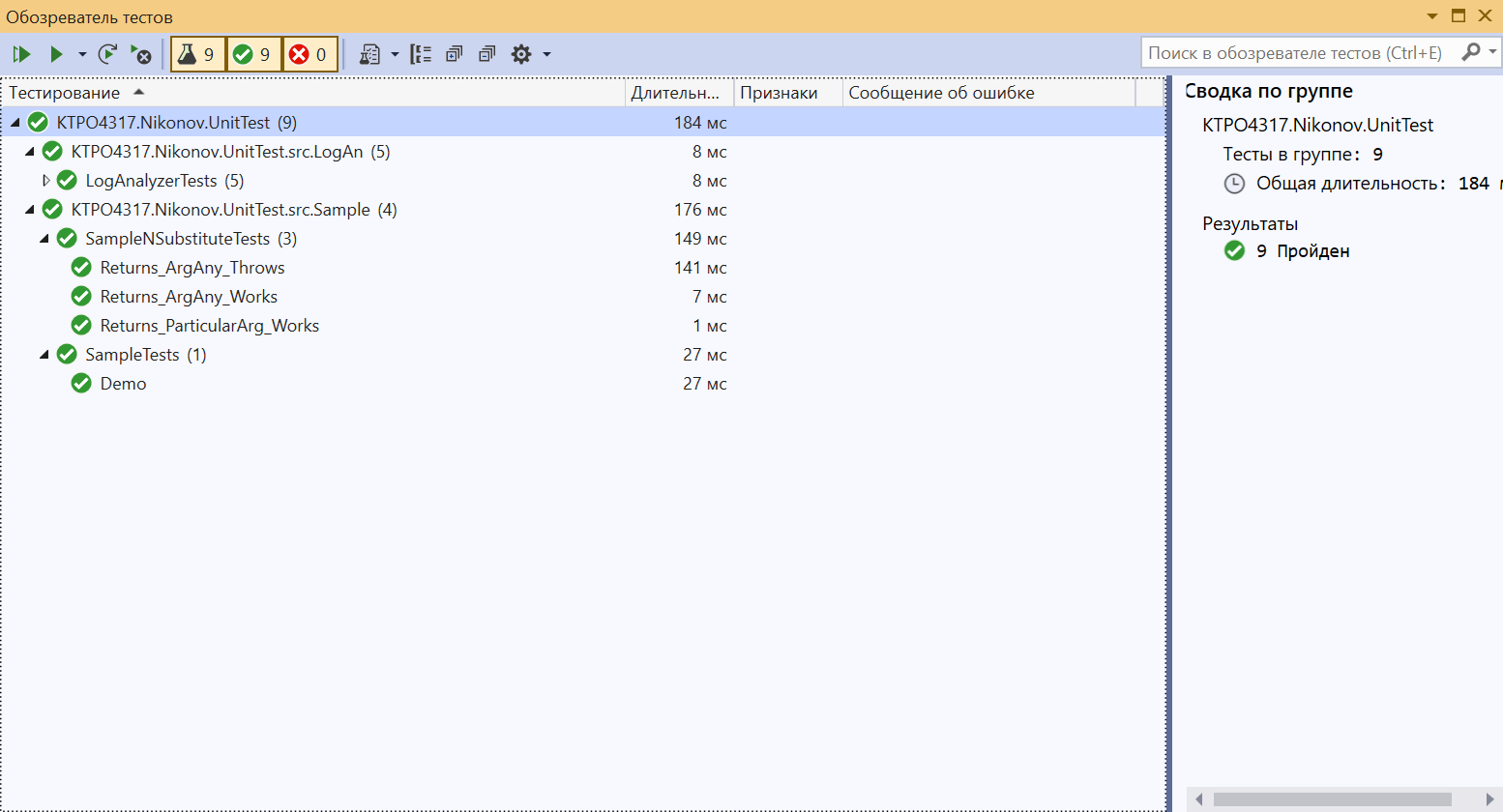


**4. Имитация вызова исключений**

Добавьте в класс SampleNSubstituteTests тестовый метод Returns\_ArgAny\_Throws(), который демонстрирует настройку вызова исключения в поддельном объекте. Рассмотрен случай, когда результат не зависит от аргумента тестируемого метода.

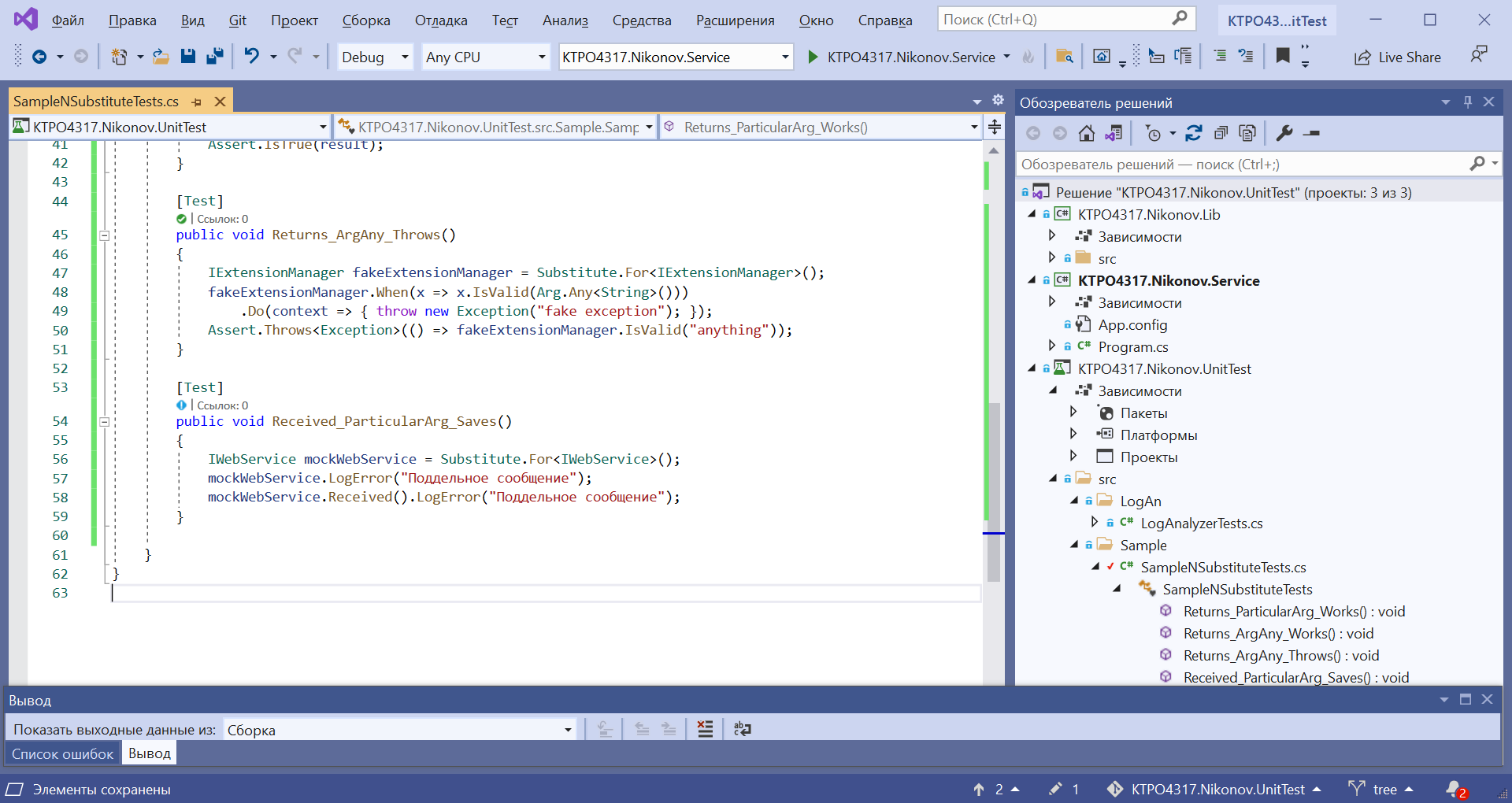


Выполните тесты.



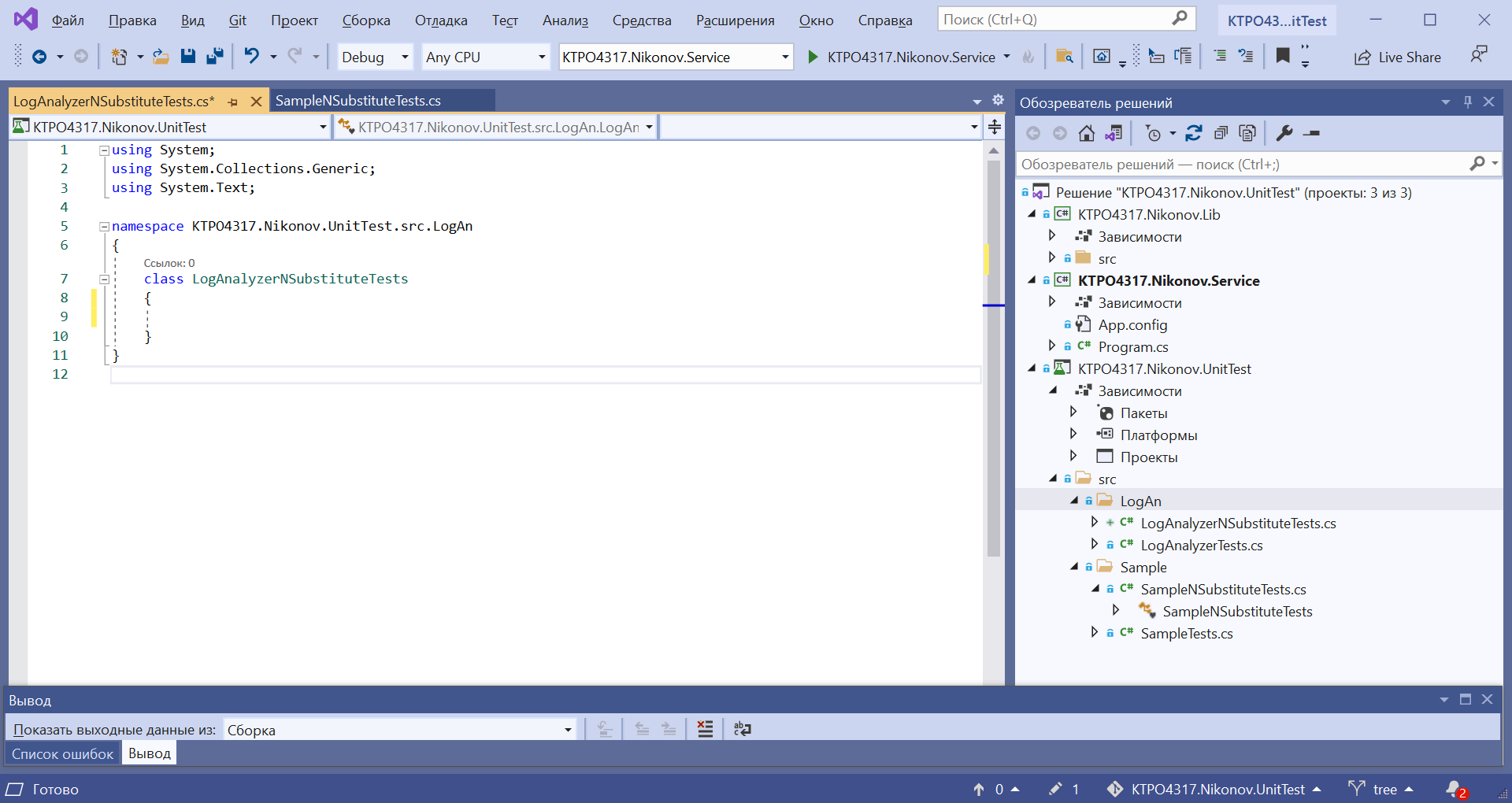
**5. Проверка вызова поддельного объекта**

Добавьте в класс SampleNSubstituteTests тестовый метод Received\_ParticularArg\_Saves(), который демонстрирует, как проверить, вызывался ли поддельный объект с указанными параметрами.

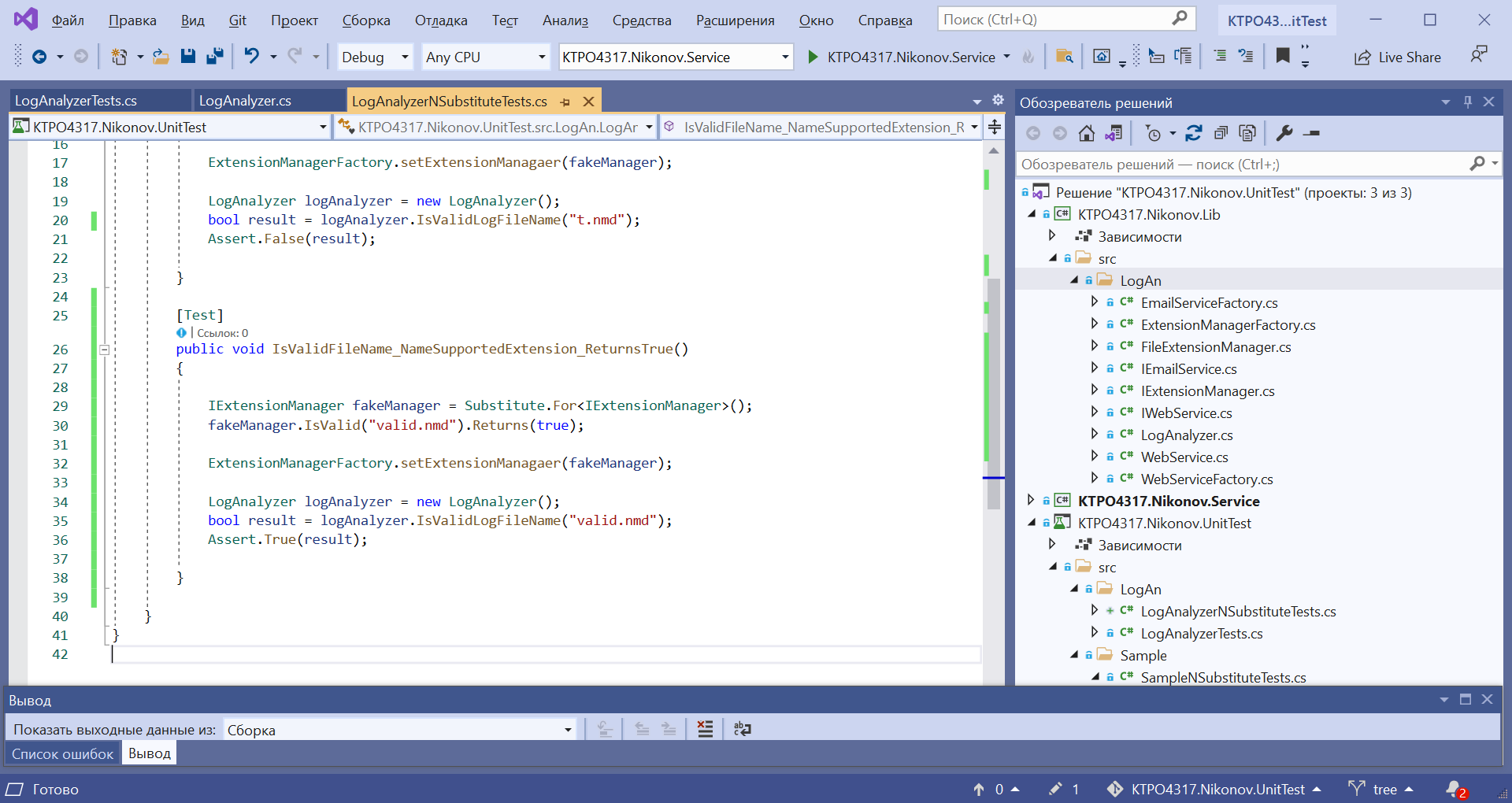




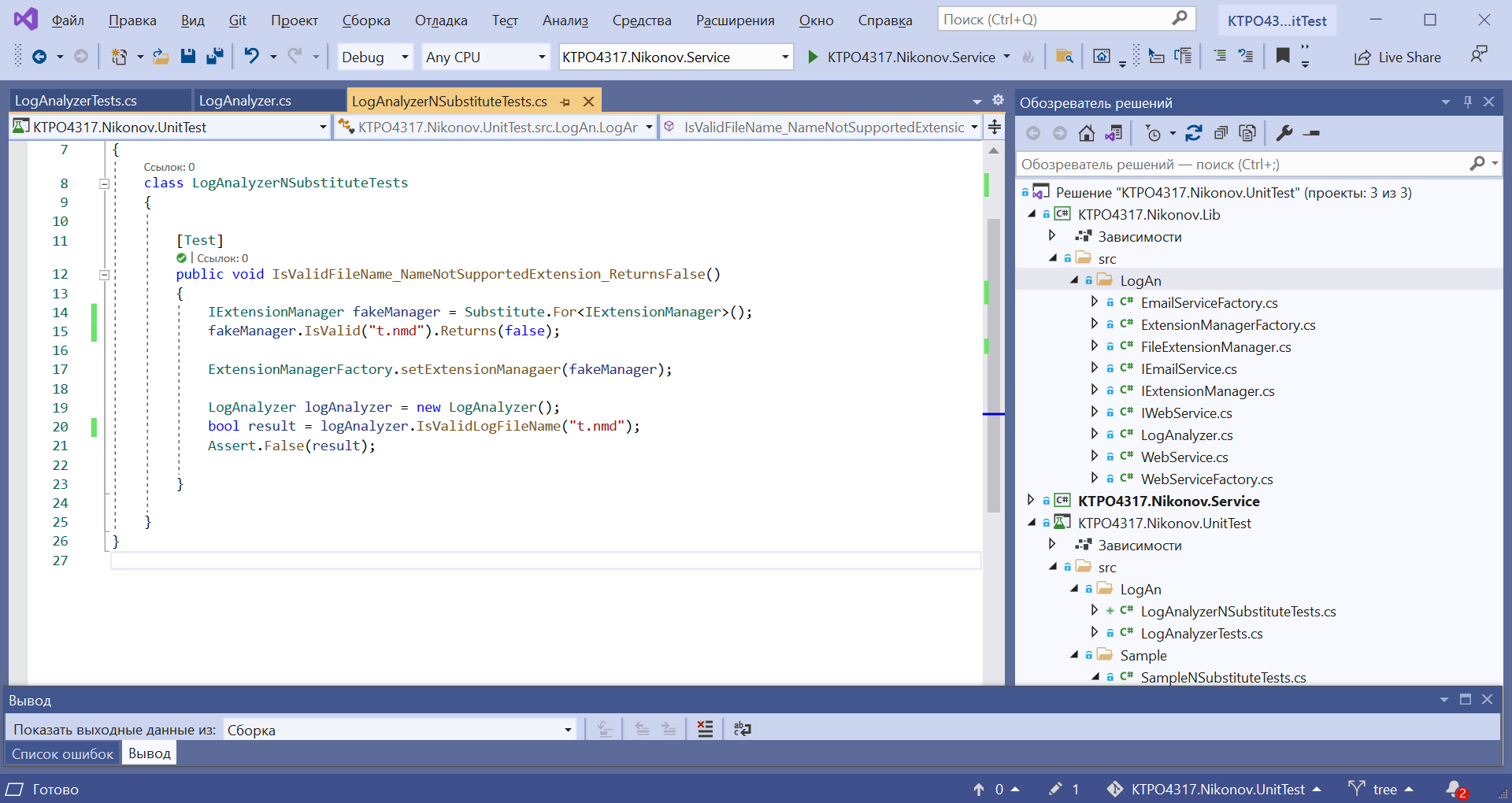
**6. Замена рукописной подделки динамической**

Добавьте в папку LogAn новый тестовый класс LogAnalyzerNSubstituteTests. В нем разместим тесты для класса LogAnalyzer реализованных с использованием изолирующего каркаса. 

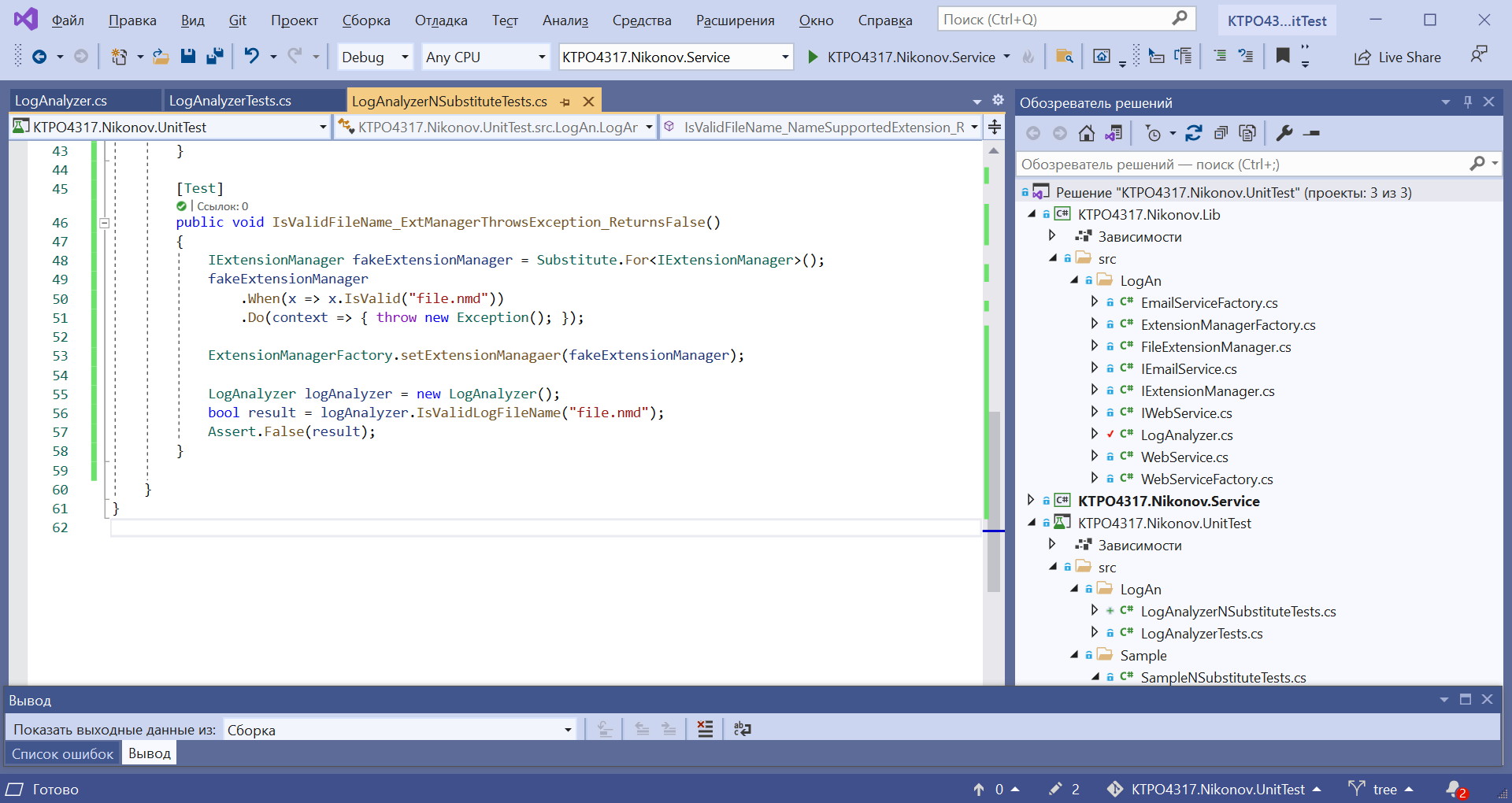
Реализуйте тестовый метод в классе LogAnalyzerNSubstituteTests тестовый метод IsValidFileName\_NameSupportedExtension\_ReturnsTrue (см. лабораторную работу №2).



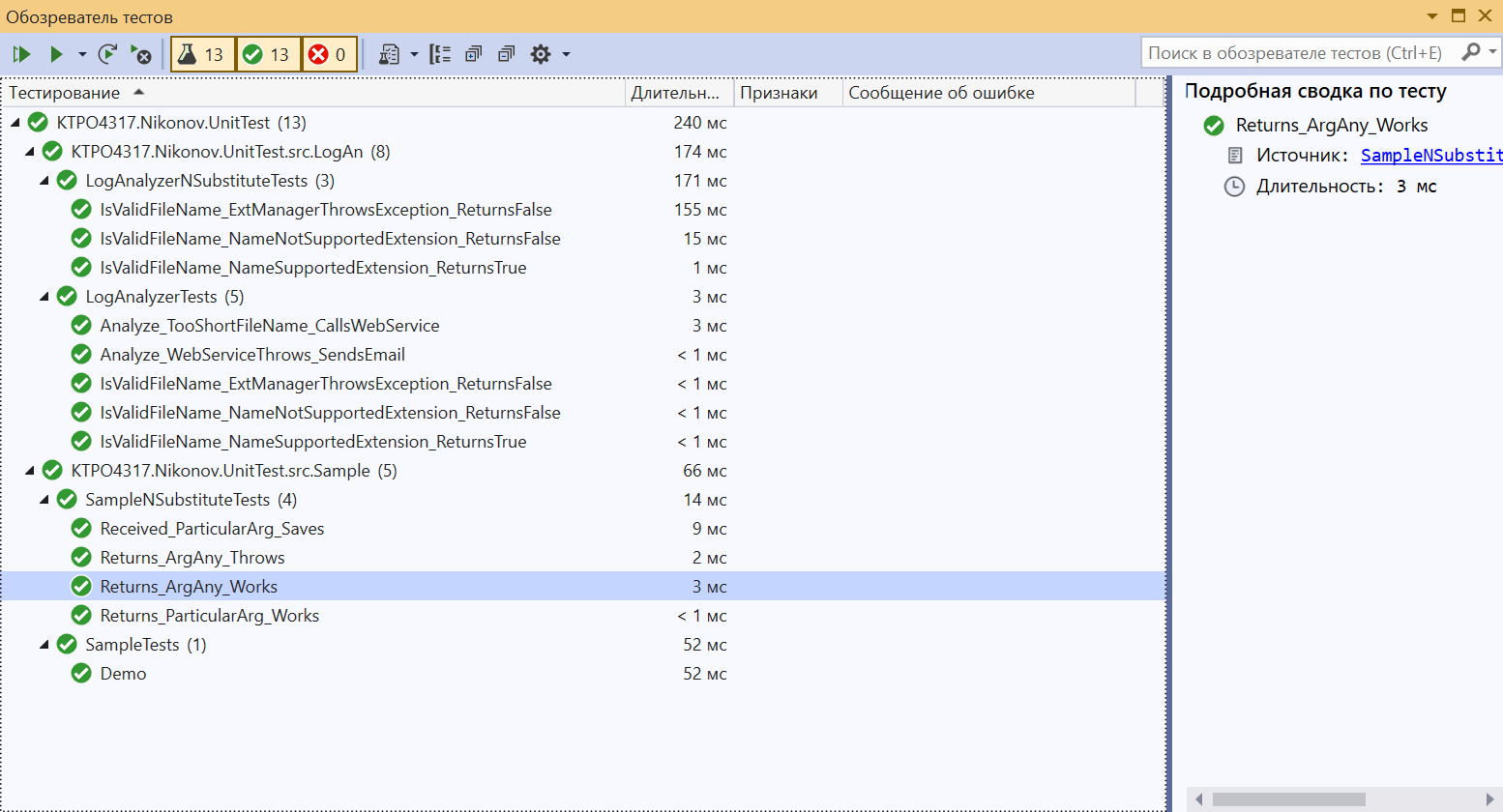
Реализуйте тестовый метод в классе LogAnalyzerNSubstituteTests тестовый метод IsValidFileName\_NoneSupportedExtension\_ReturnsFalse (см. лабораторную работу №2).



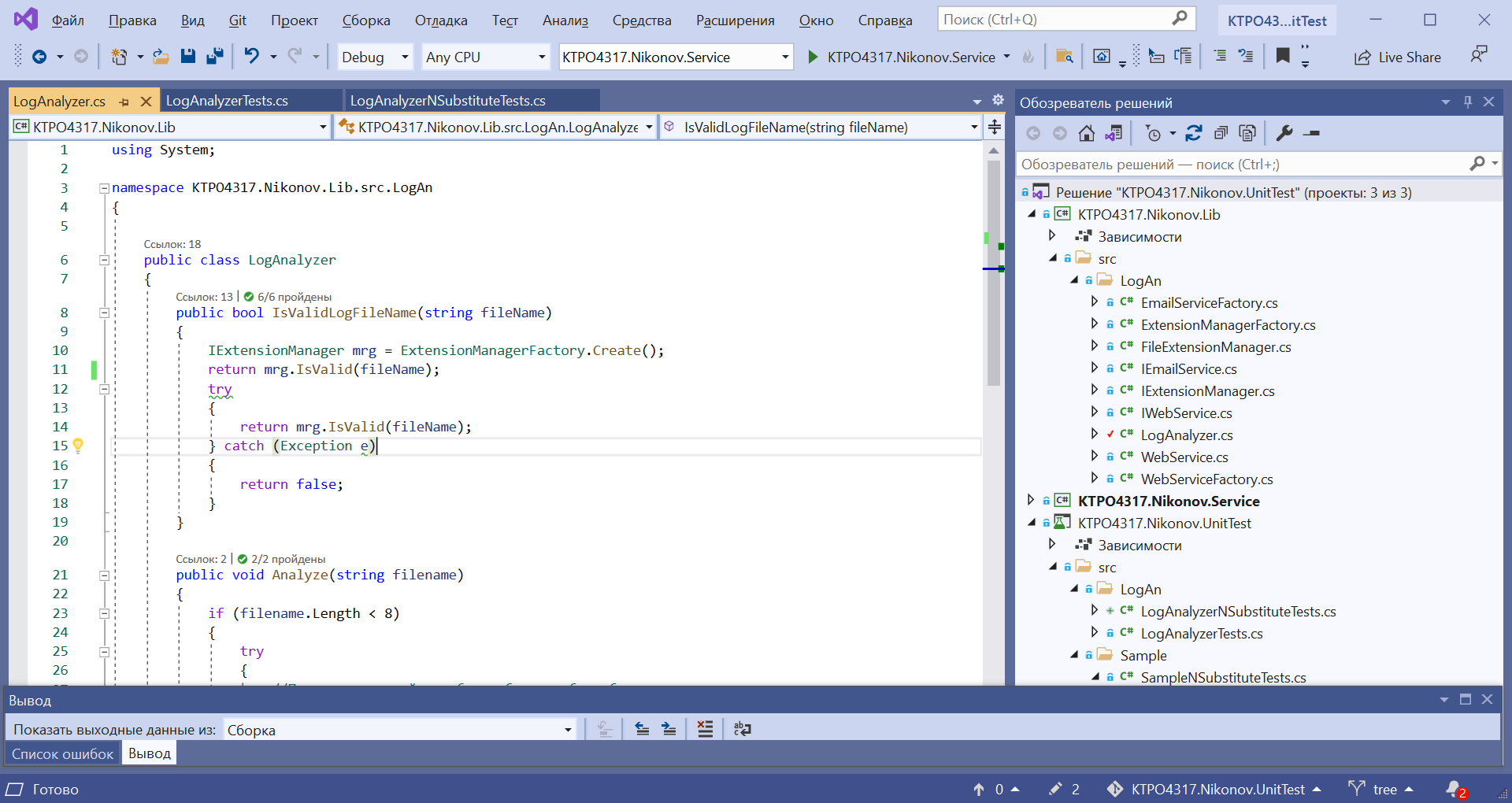
Реализуйте тестовый метод в классе LogAnalyzerNSubstituteTests тестовый метод IsValidFileName\_ExtManagerThrowsException\_ReturnsFalse (см. лабораторную работу №2).

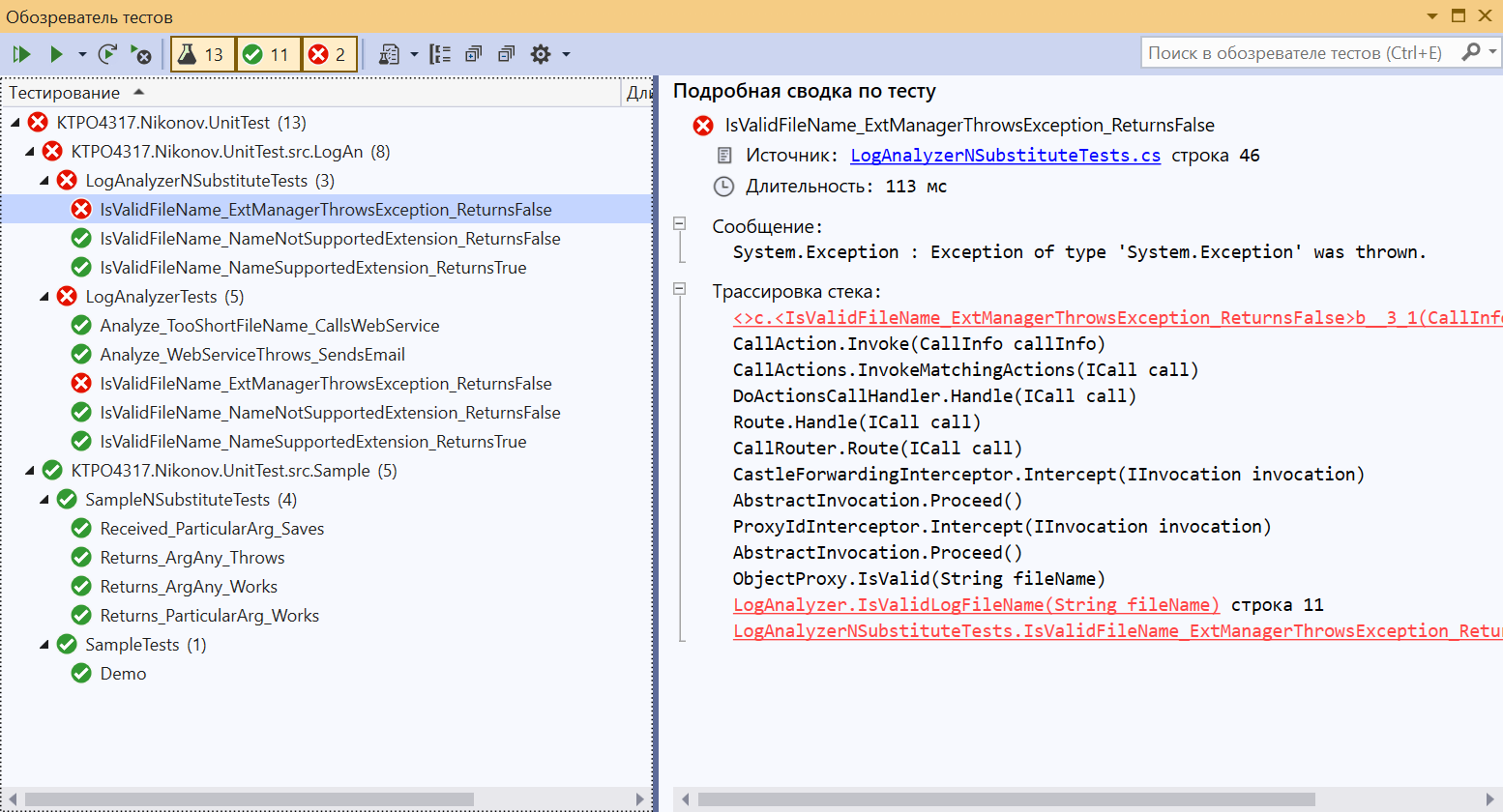


Выполните тесты.

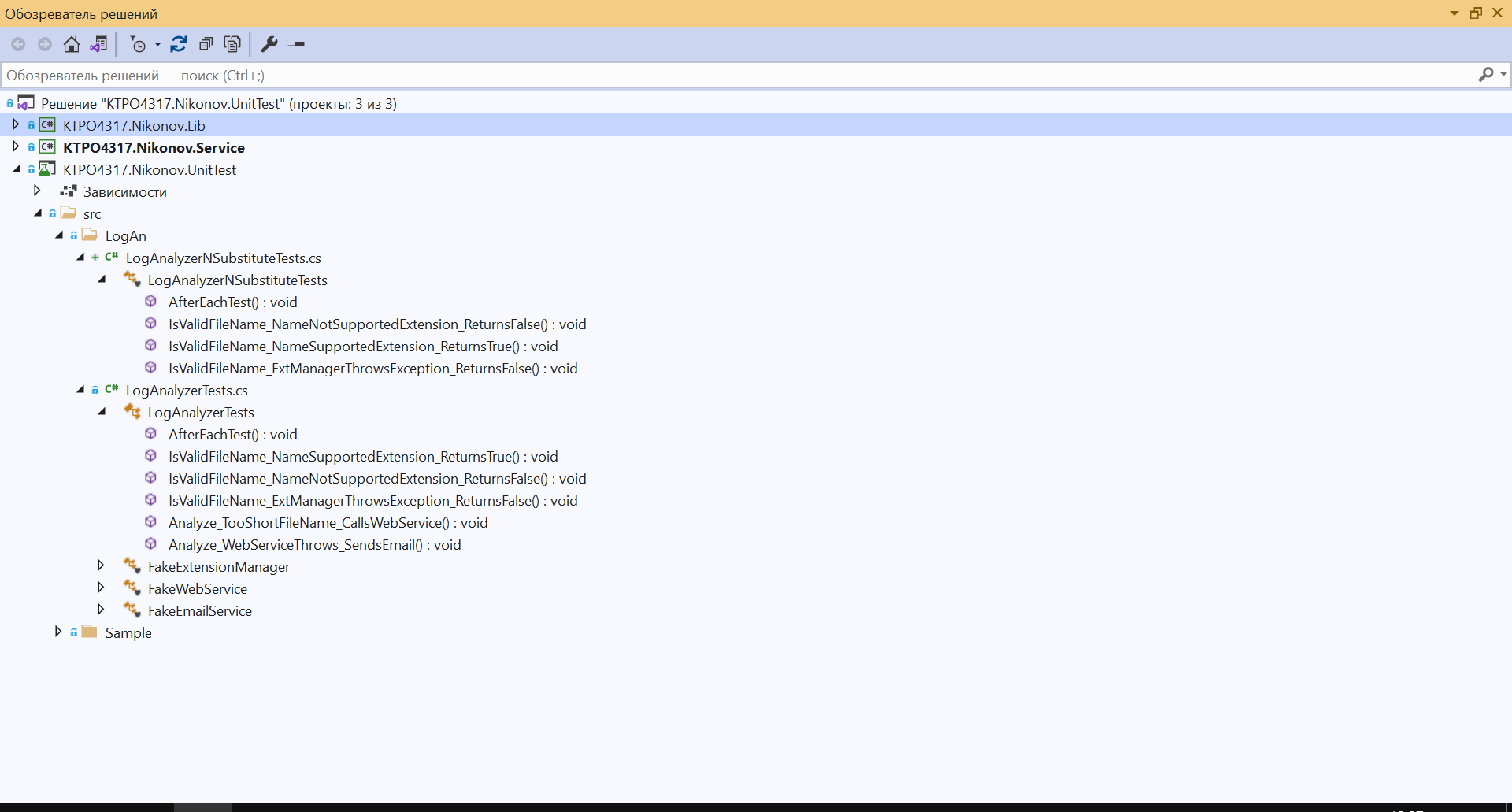


Проверьте правильность последнего тестового метода. Для этого внесите в тестируемый метод дефект, ошибку которую должен обнаружить тест. Выполните тест и зафиксируйте результат, в том числе и текст сообщения об ошибке. Убедитесь, что вариант теста и рукописной и динамической заглушкой работают одинаково. Восстановите правильный код.



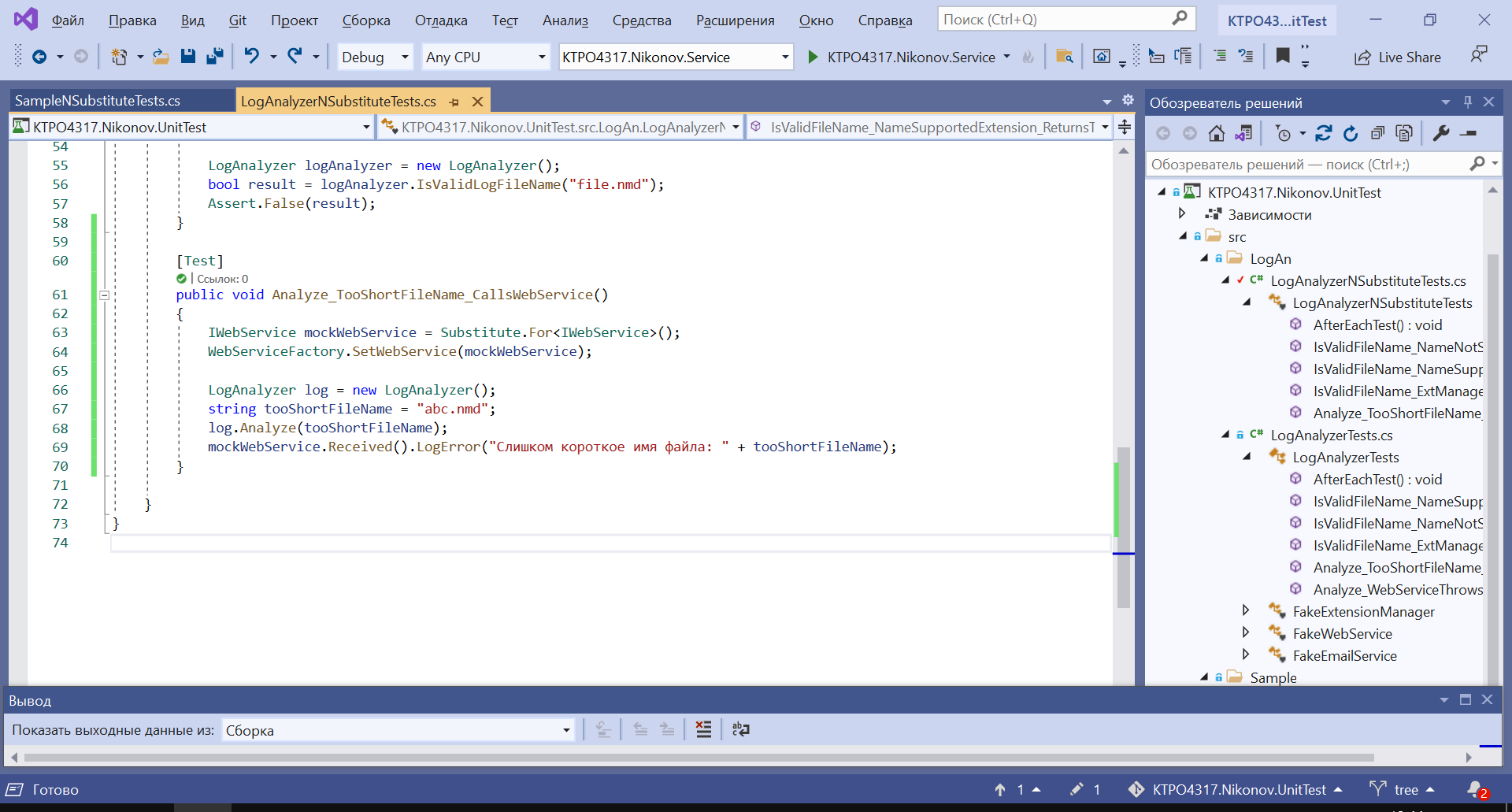


Структура проекта:



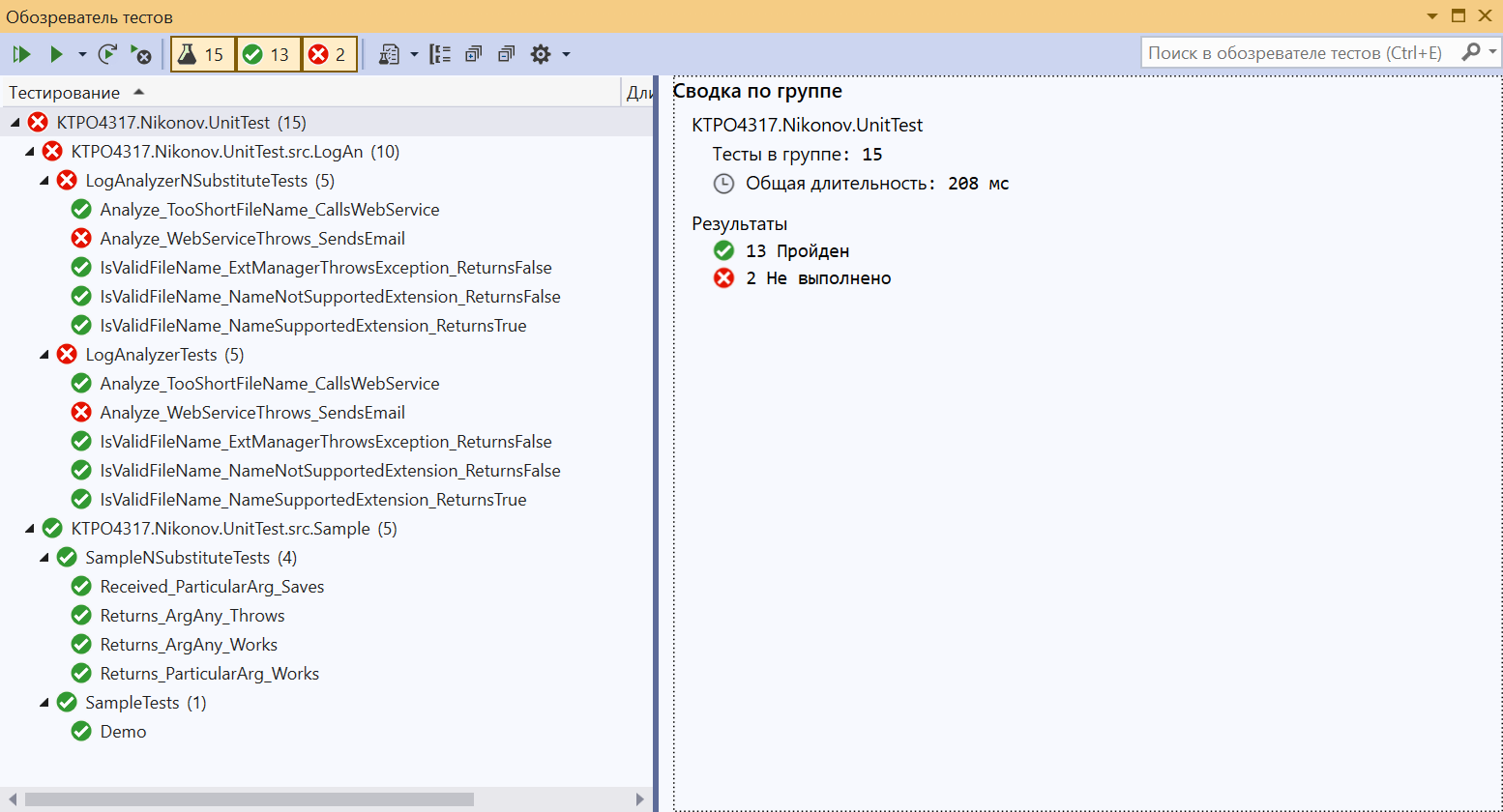
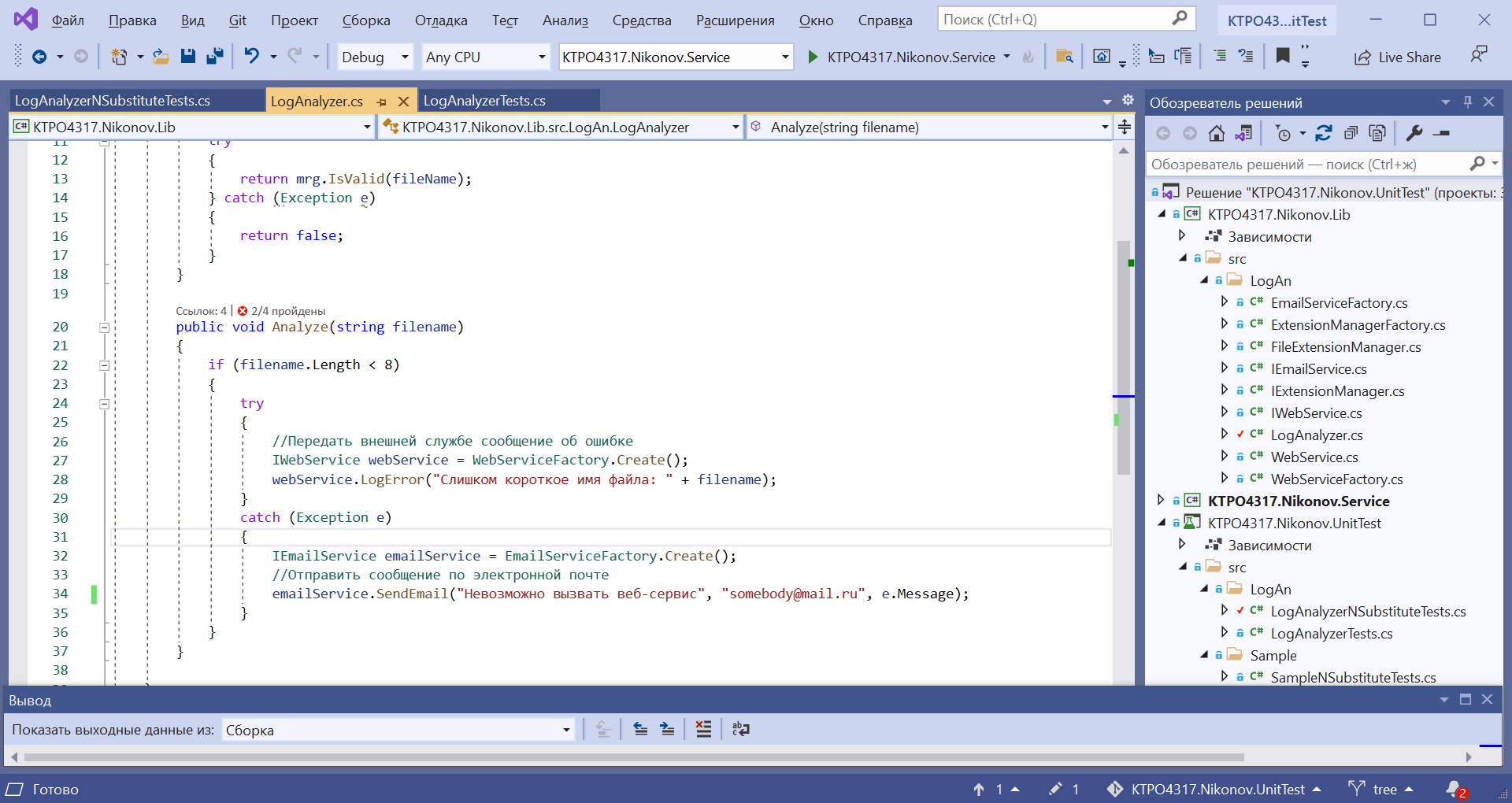
**7. Совместное использование заглушки и подставки**

Реализуйте тестовый метод в классе LogAnalyzerNSubstituteTests тестовый метод Analyze\_TooShortFileName\_CallsWebService() (см. лабораторную работу №3).



Реализуйте тестовый метод в классе LogAnalyzerNSubstituteTests тестовый метод Analyze\_WebServiceThrows\_SendsEmail() (см. лабораторную работу №3).



Проверьте правильность последнего тестового метода. Для этого внесите в тестируемый метод дефект, ошибку которую должен обнаружить тест: в тестируемом методе поменяйте местами значения параметров при вызове метода SendEmail. Выполните тест и зафиксируйте результат, в том числе и текст сообщения об ошибке. Убедитесь, что вариант теста и рукописными и динамическим поддельными объектами работают одинаково. Восстановите правильный код.  

Структура проекта:

