**Лабораторная работа №5.**

**Тестирование событий**

**Цель работы**

Приобретение практических навыков использования делегатов (delegate) и событий (event); тестирования операций, связанных с событием; практика использования тестового каркаса NUnit, практика использования изолирующего каркаса NSubstitute.

**Задание на лабораторную работу**

1. Подготовить учебный проект

2. Реализовать генерацию события

3. Реализовать подписку на событие

4. Реализовать тесты для проверки прослушивателя события, с имитацией события в ручной заглушке, созданный с помощью наследования.

5. Реализовать тесты для проверки прослушивателя события с генерацией имитацией события в с помощью NSubstitute

6. Тестирования факта генерации события с использованием анонимного делегата

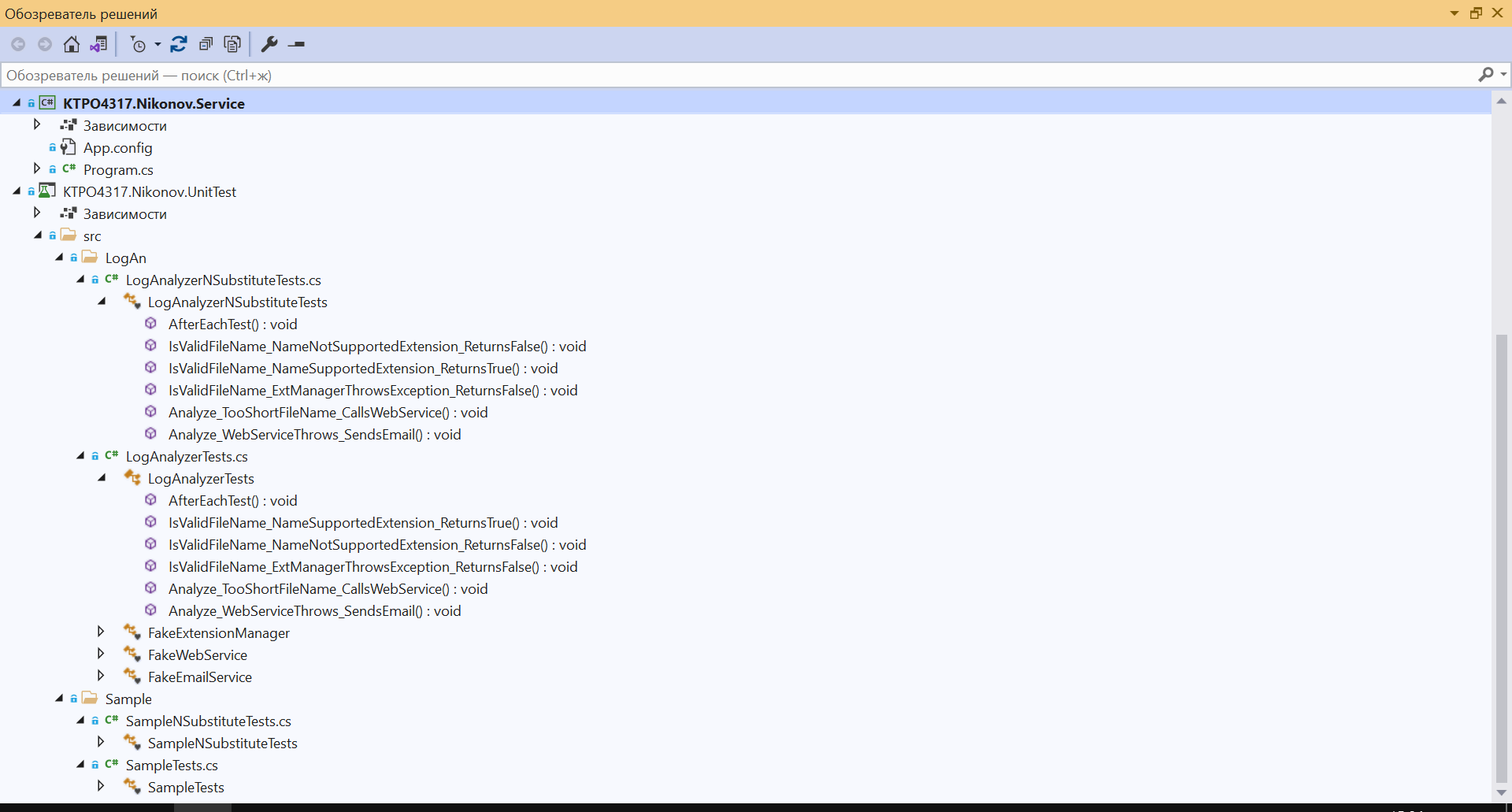
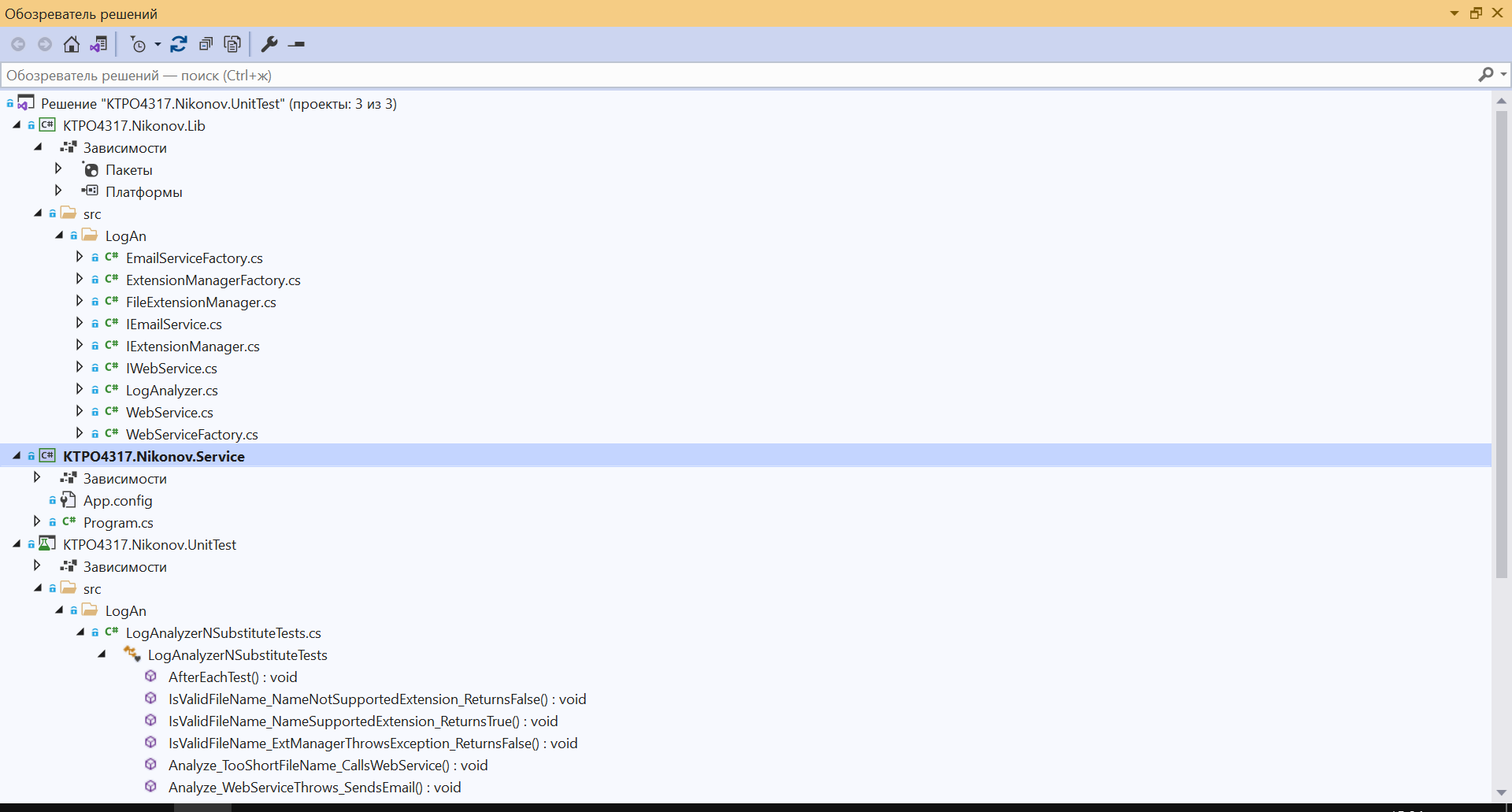
7. На каждом шаге делайте снимки исходного кода создаваемых или изменяемых классов и тестов, окна «Результаты тестов» и «Обозреватель решения» и сохраните в документе MS Word.

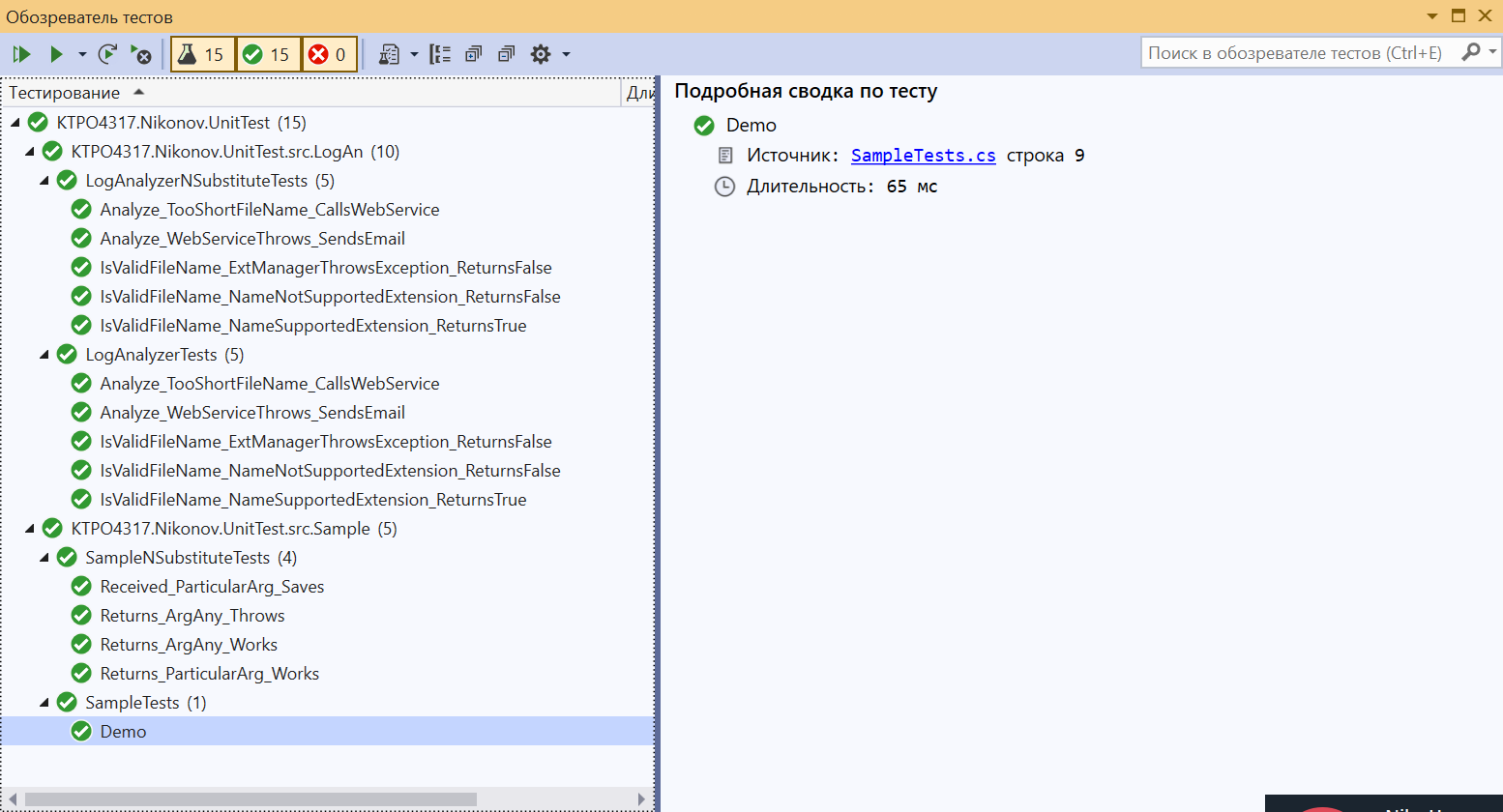
8. Оформить отчет

**Ход работы**

**1. Подготовка проекта**

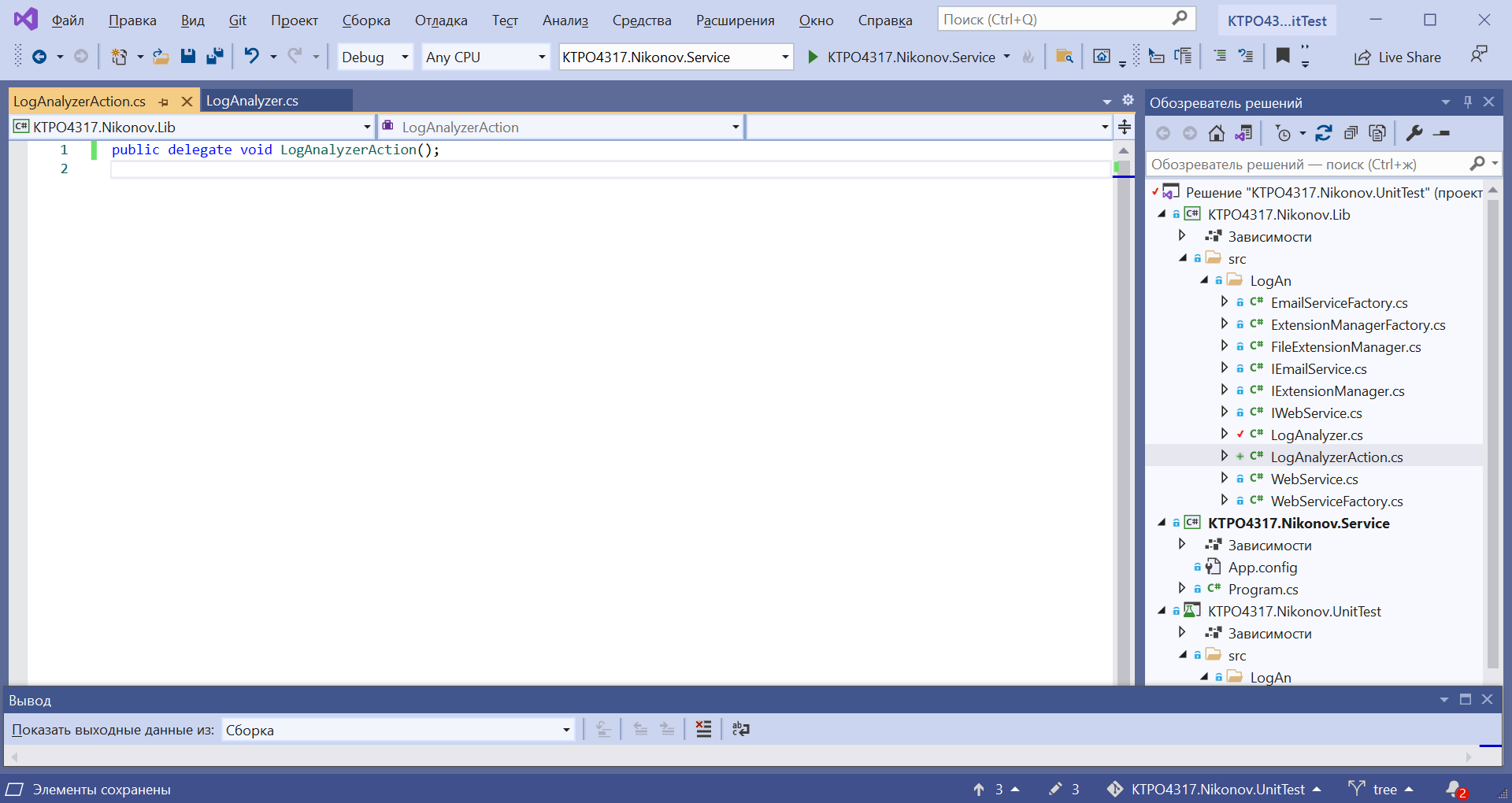
Для выполнения данной лабораторной работы возьмите решение, полученное в результате выполнения лабораторной работы №4. Выполните тесты.



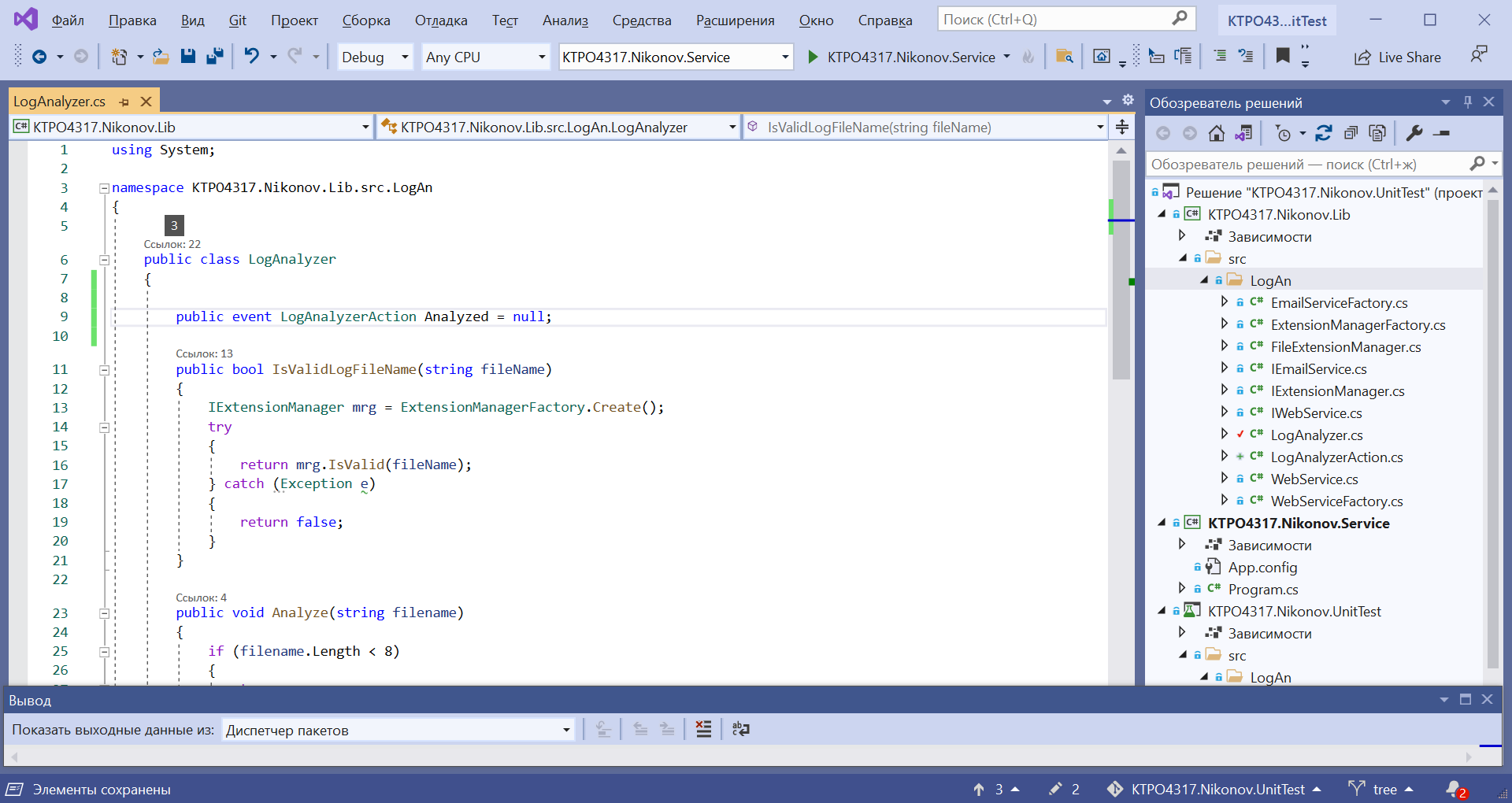


**2. Генерация события**

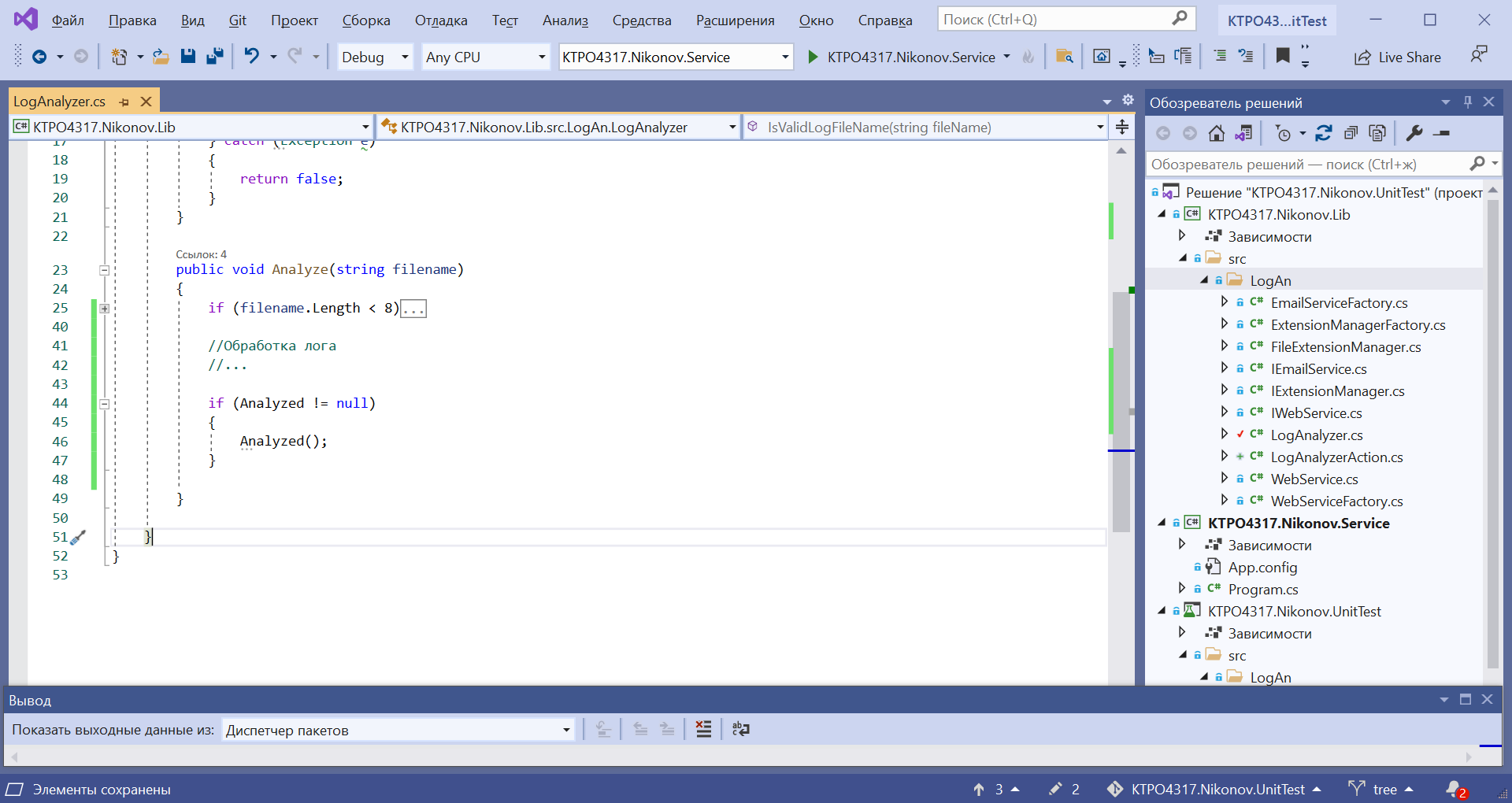
Объявите в проекте «.Lib» делегат LogAnalyzerAction. Разметите его в отдельном файле «.cs» файл в папке LogAn.



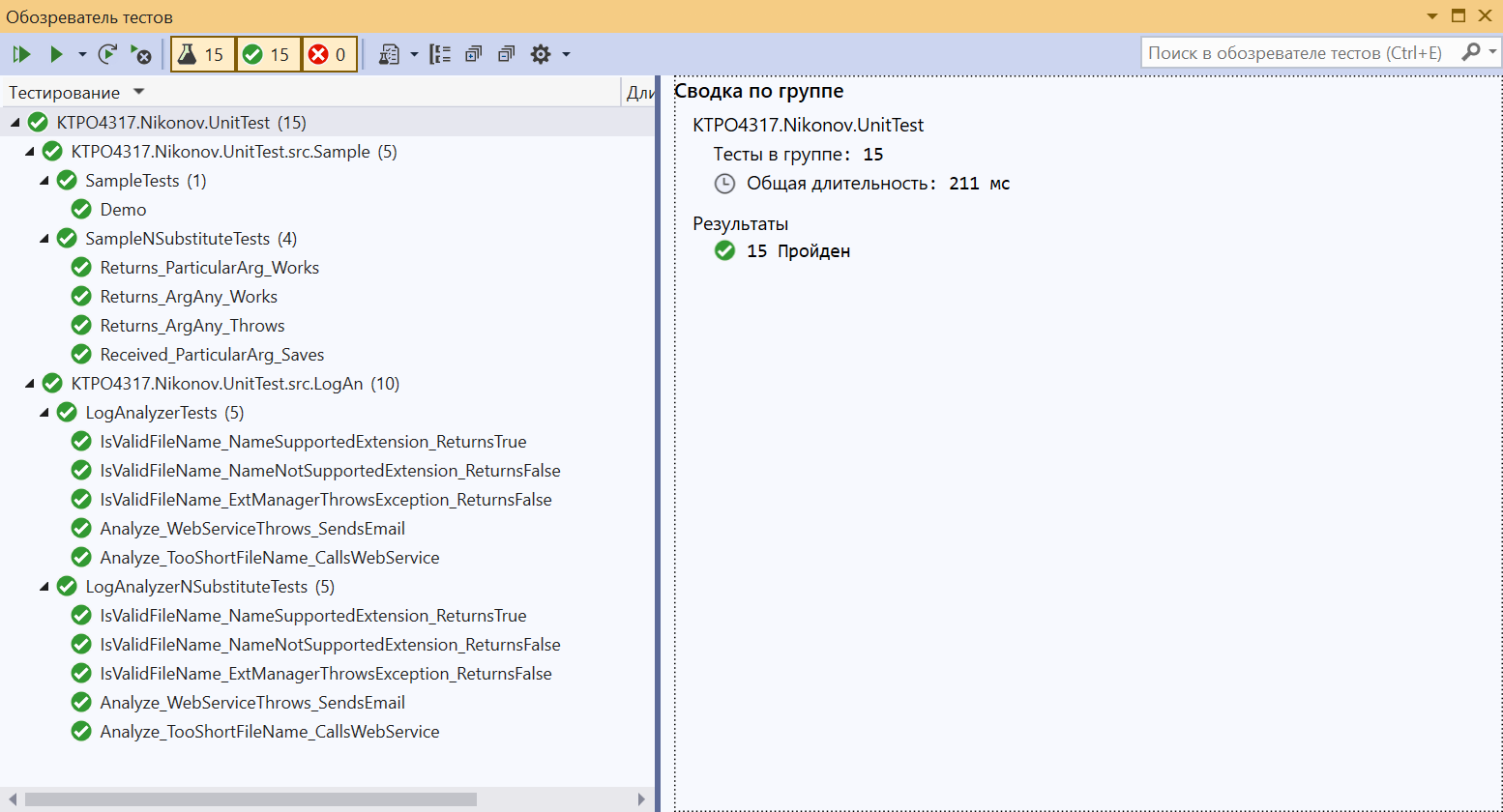
Объявите событие с помощью ключевого слова event типа LogAnalyzerAction.



Добавьте вызов события в метод LogAnalyzer. Analyze. Обратите внимание, вызов события необходимо выполнять только, если для него есть обработчик.

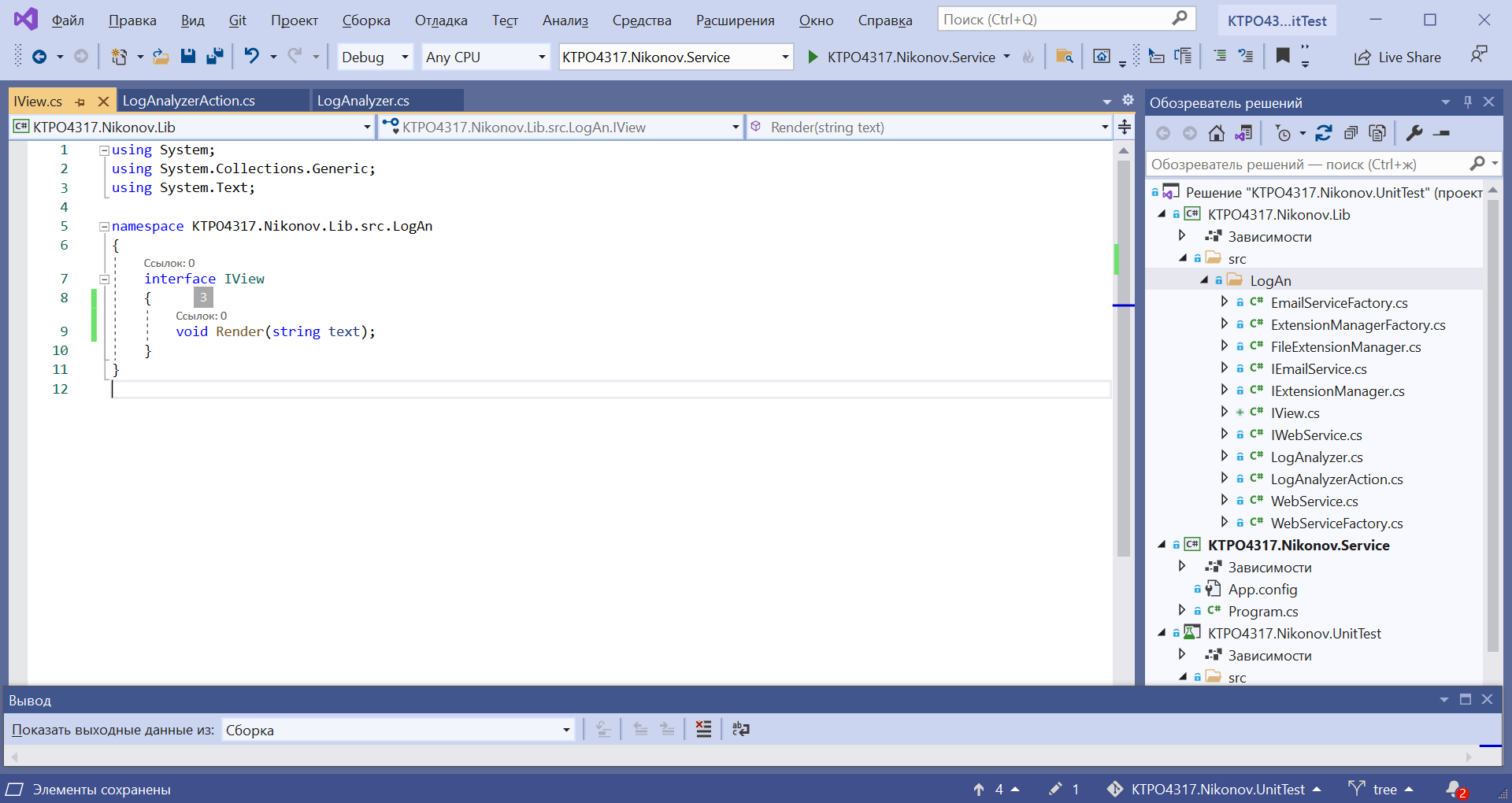


Постройте приложение и выполните тесты.

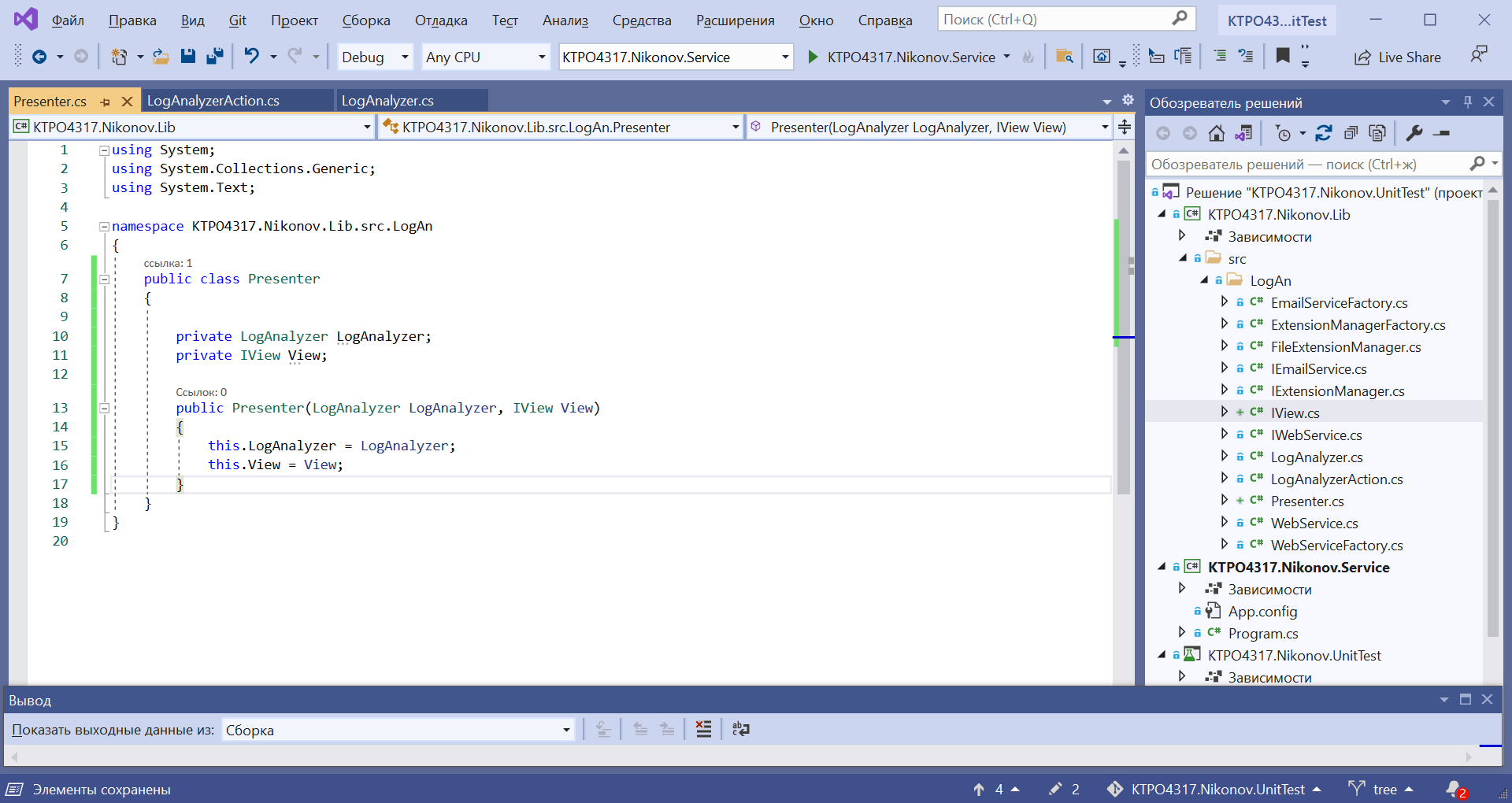


**3. Подписка объекта на событие**

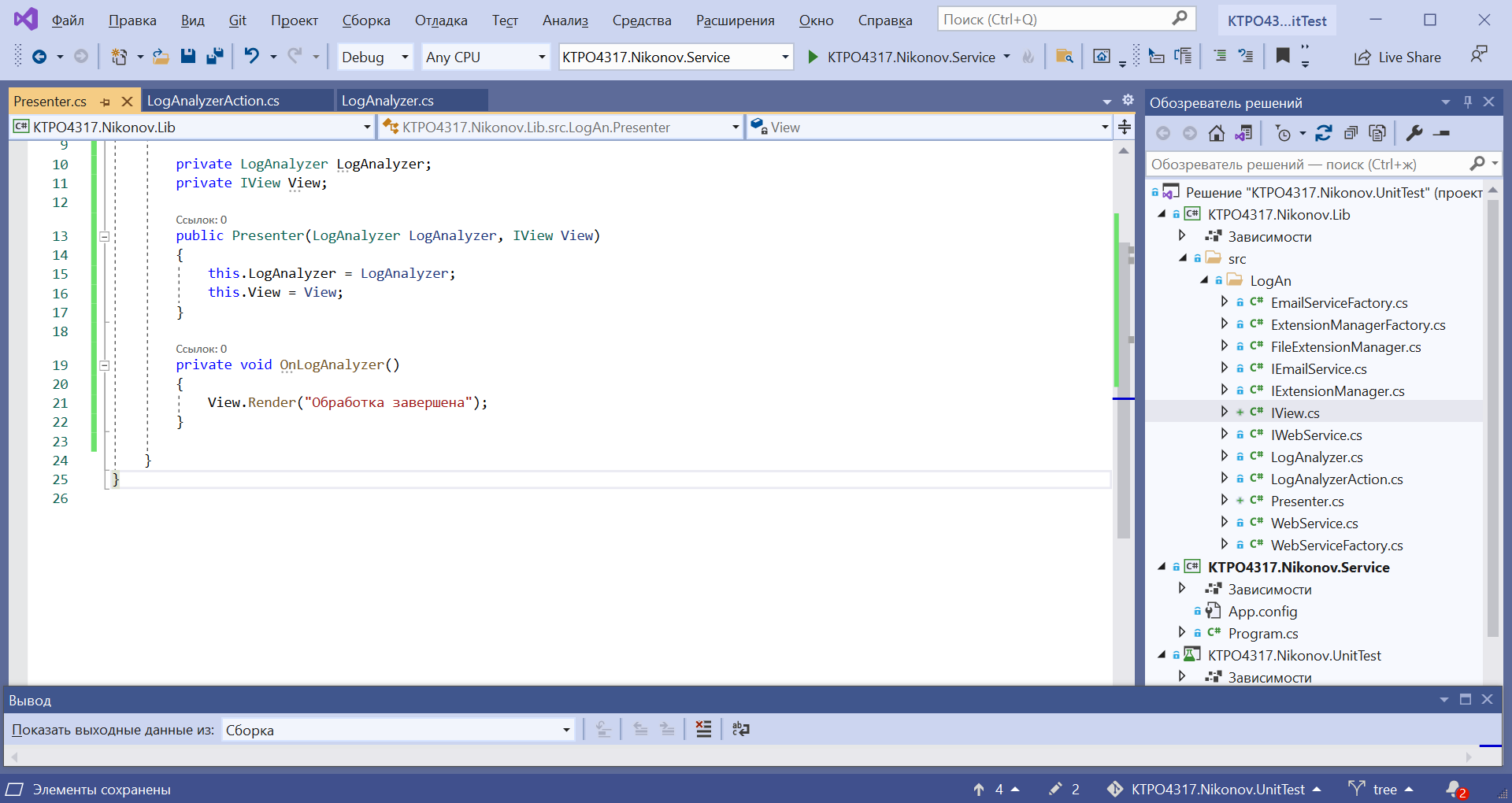
Добавьте в проекте «.Lib» в папке LogAn интерфейс IView для Представления . Включите в него один метод Render.



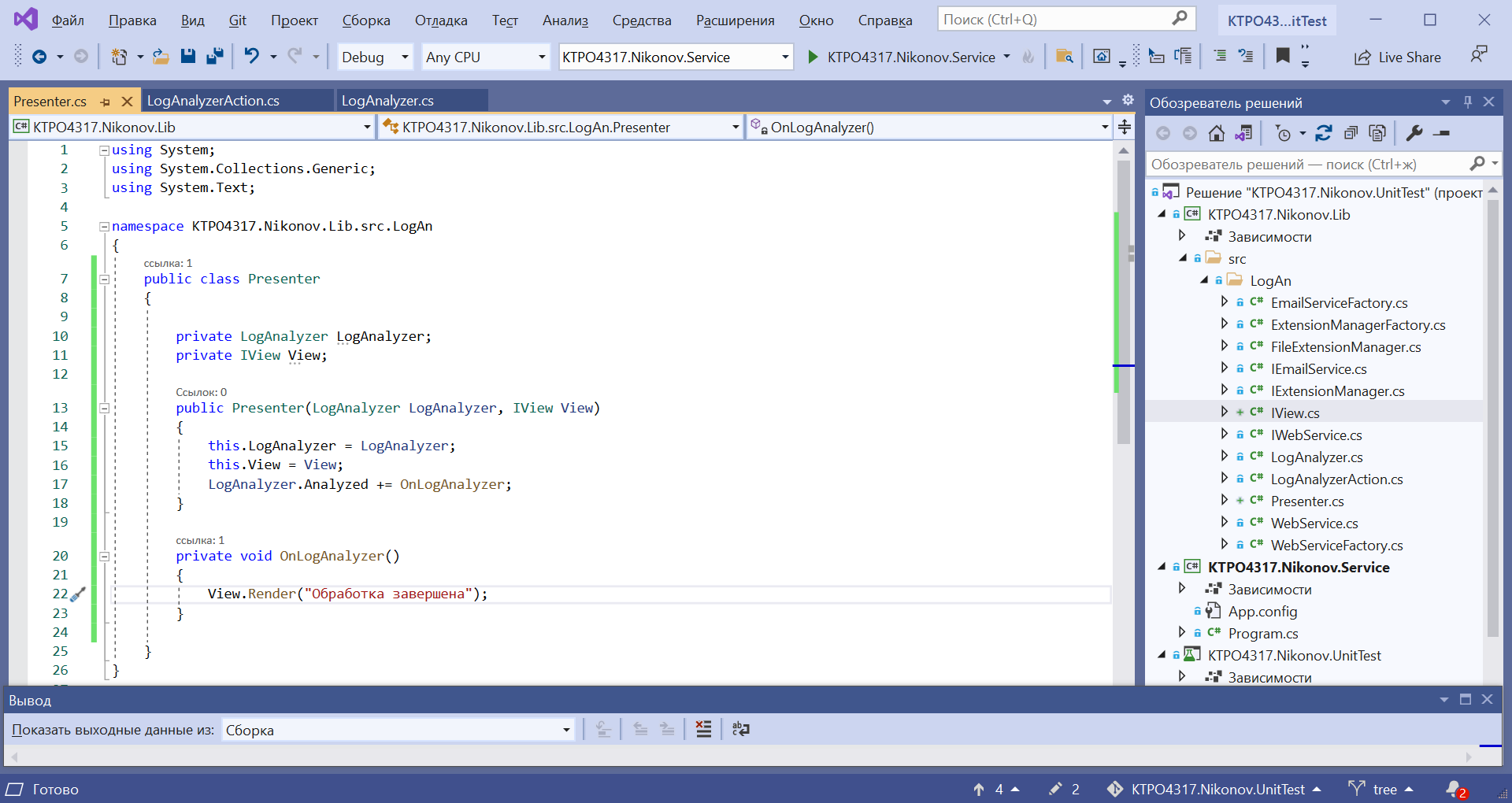
Добавьте в проекте «.Lib» в папке LogAn класс Presenter – Презентатор. В классе Presenter объявите конструктор с двумя параметрами: (LogAnalyzer logAnalyzer, IView view). Добавьте два поля для объектов LogAnalyzer и IView и инициализируйте их в конструкторе значениями из параметров.



Объявите в классе приватный метод OnLogAnalyzed() Это обработчик события. По сигнатуре он должен совпадать с сигнатурой делегата LogAnalyzerAction. Обратите внимание на соглашение о наименовании метода. Добавьте в него вызов обращение к Представлению.

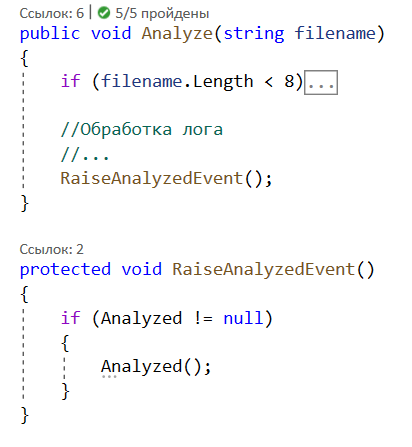


Добавьте в конструктор подписку на событие.

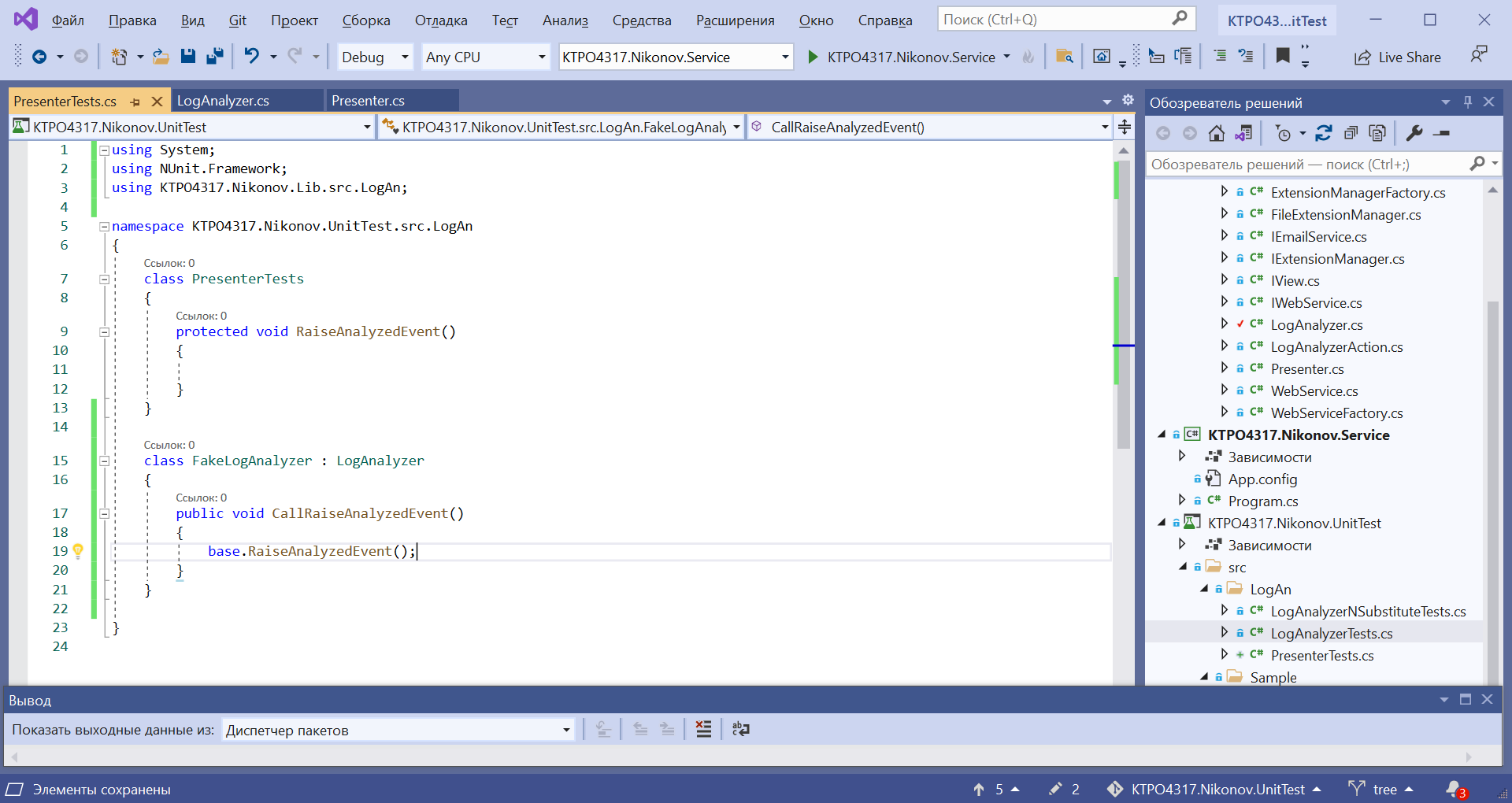


**4. Тестирование прослушивания события. Введение зазора с помощью наследования.**

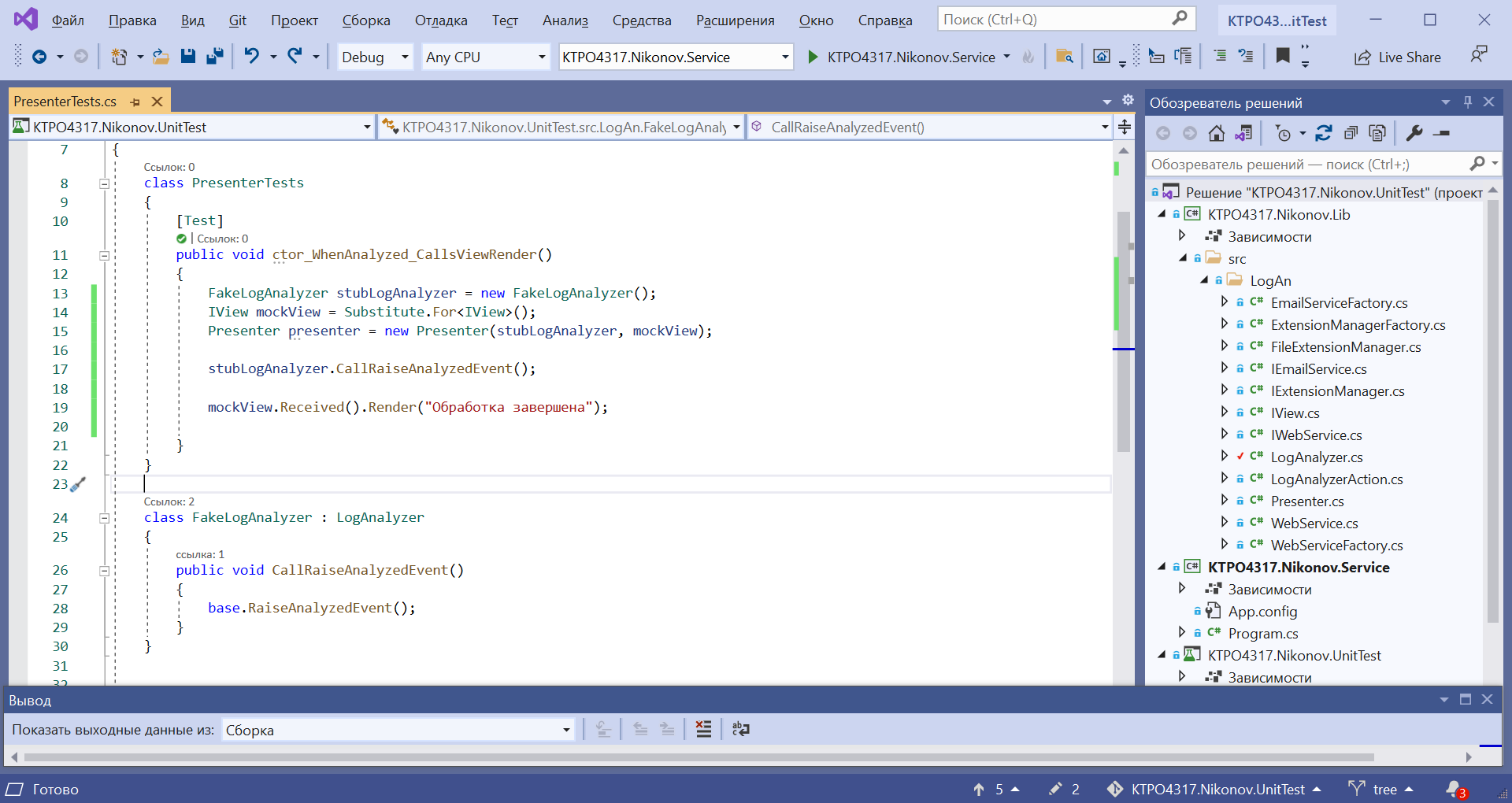
Добавьте в проект «.UnitTest» в папку LogAn тестовый класс PresenterTests. Выделите код вызова события в метод RaiseAnalyzedEvent(). Сделайте его защищенным.

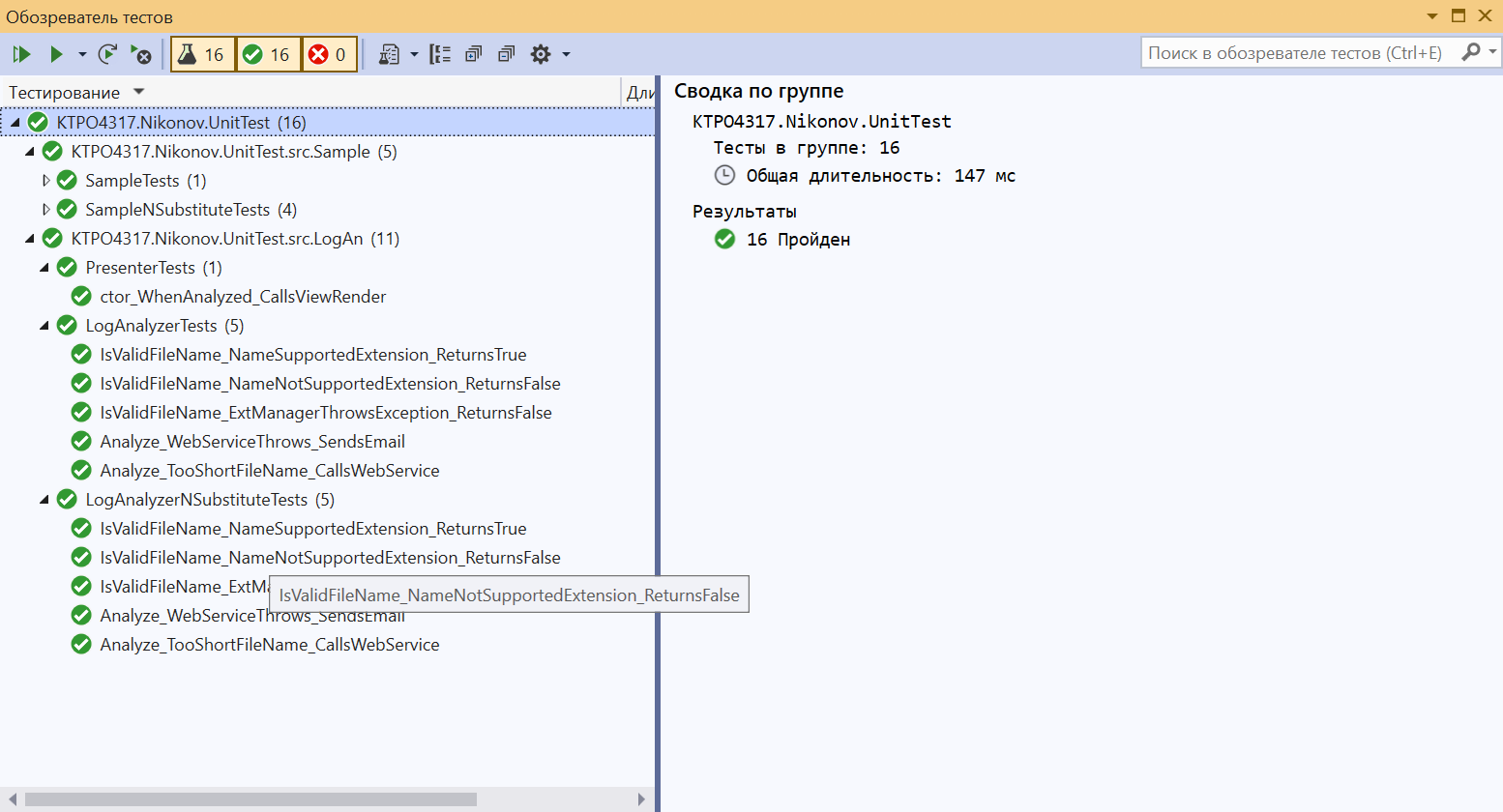


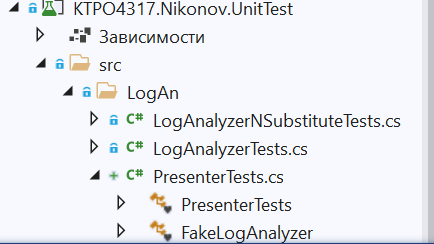
Добавьте в файл PresenterTests.cs поддельный объект, унаследовав его от LogAnalyzer. Добавьте в него открытый метод, в котором вызовите метод RaiseAnalyzedEvent() базового класса.



Добавьте тестовый метод для сценария «Если вызвано событие, то класс вызывает отображение»: ctor\_WhenAnalyzed\_CallsViewRender

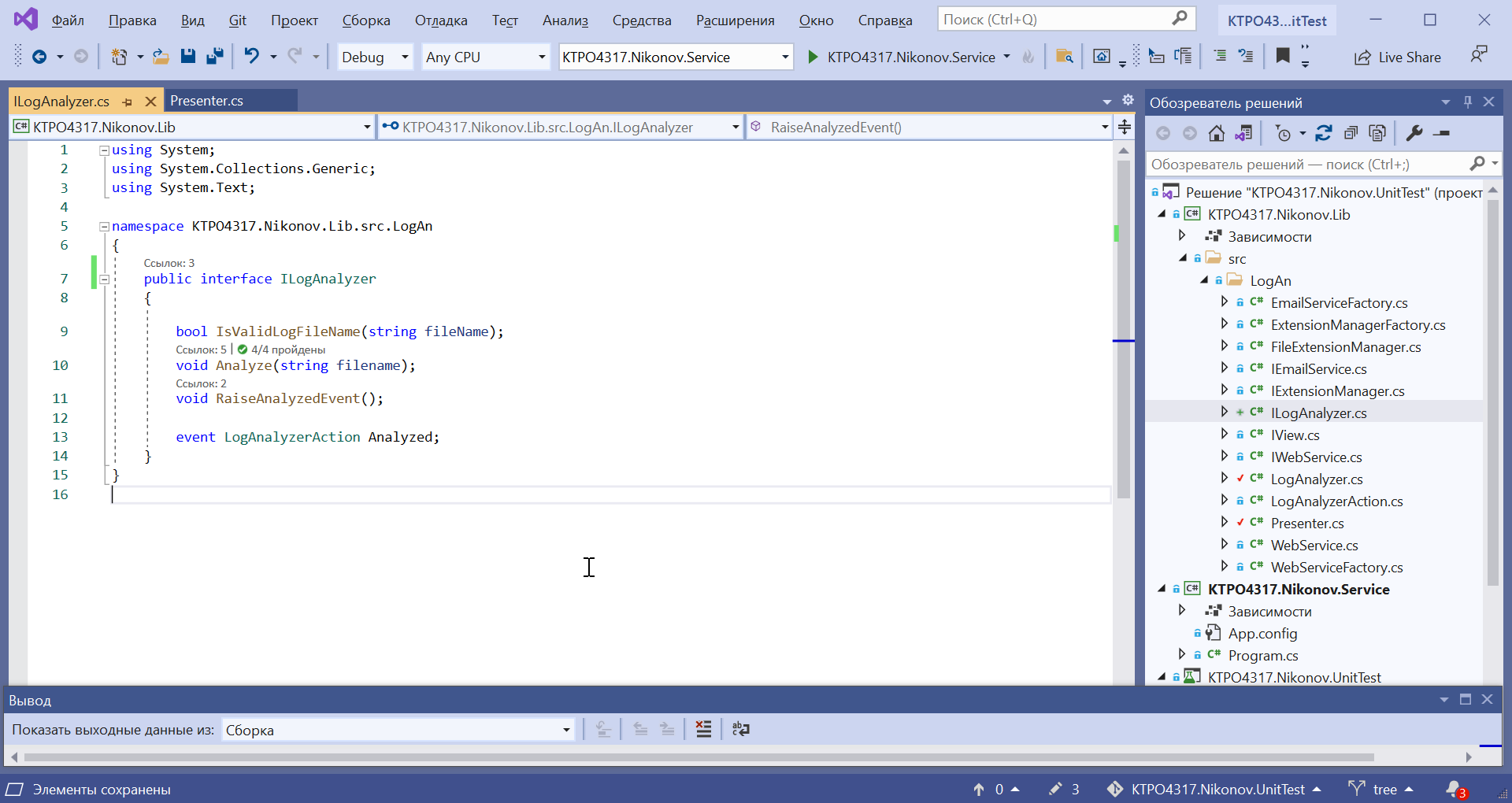




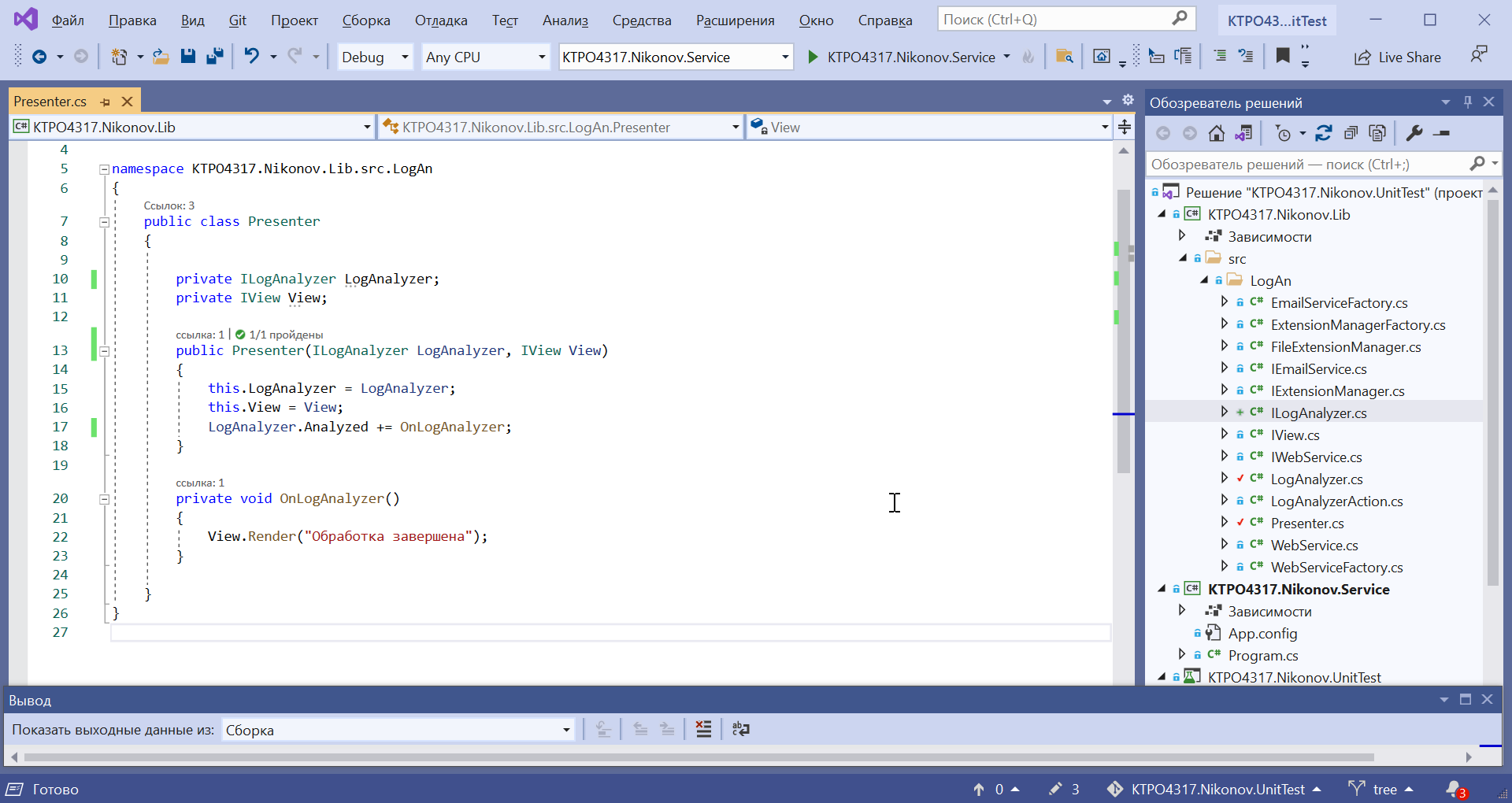


**5. Имитация события с помощью NSubstitute**

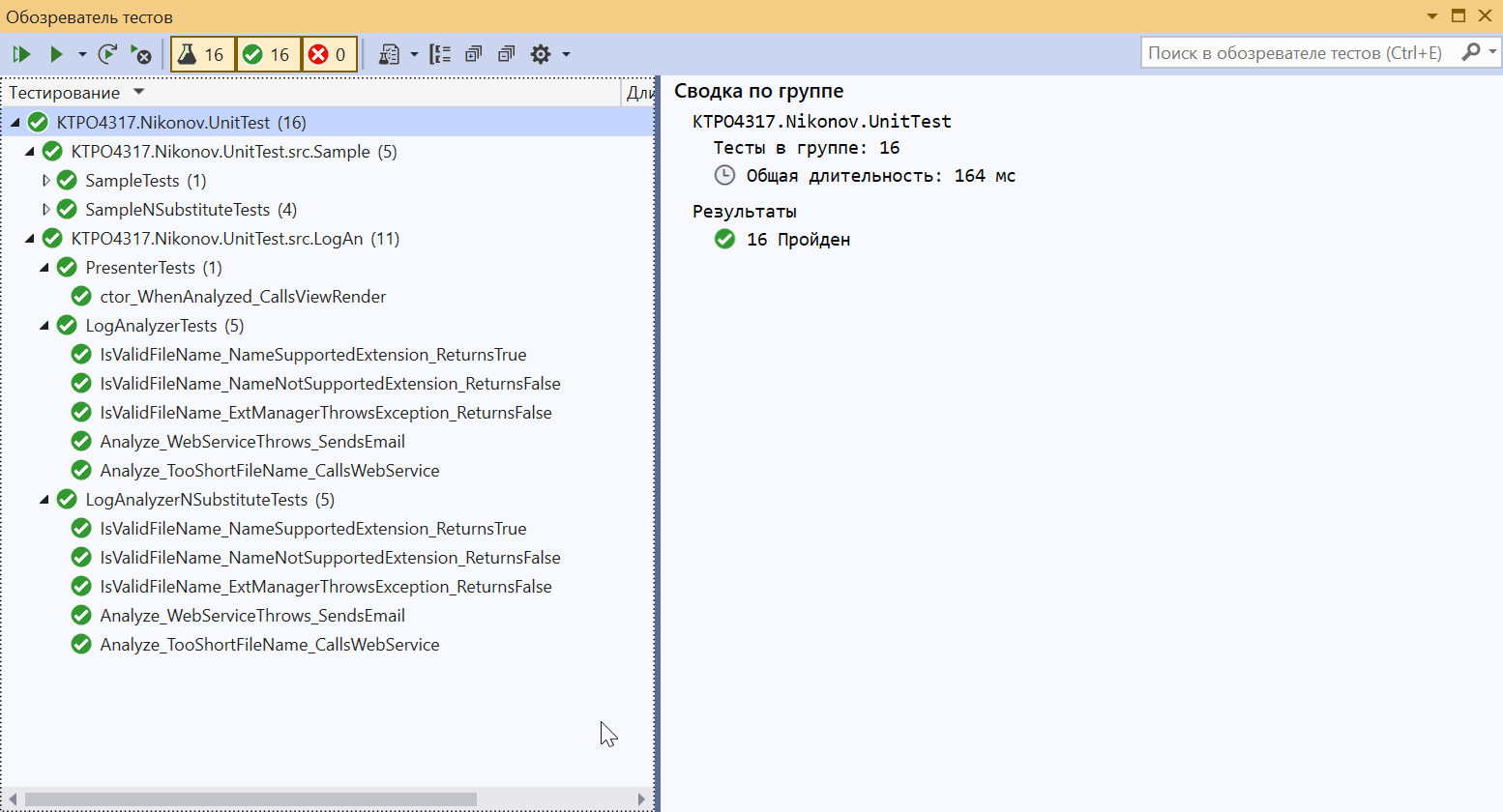
Выделите для класса LogAnalyzer интерфейс ILogAnalyze. Включите в него все методы и событие.



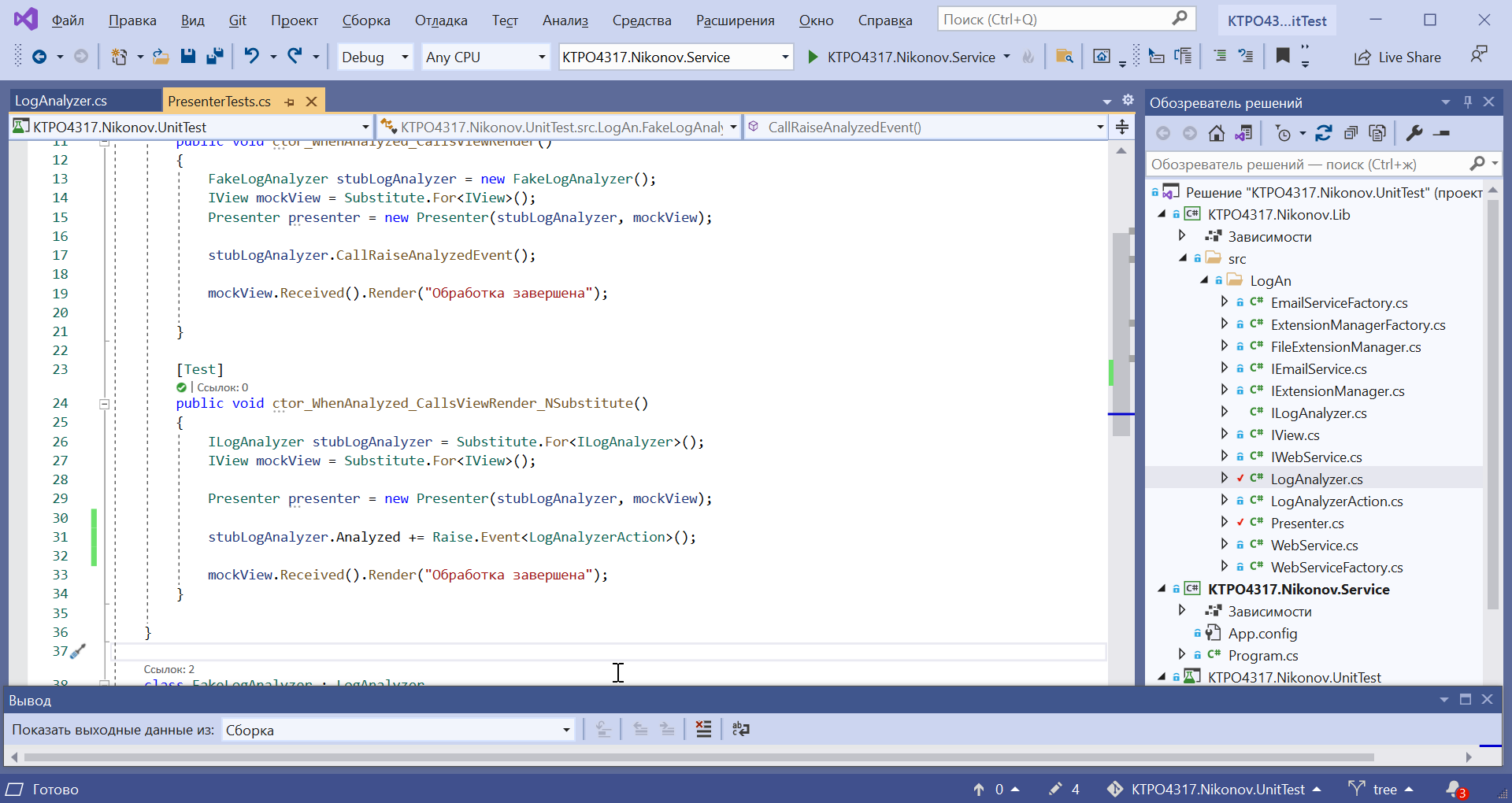
В классе Presenter замените использование класса LogAnalyzer на интерфейс.



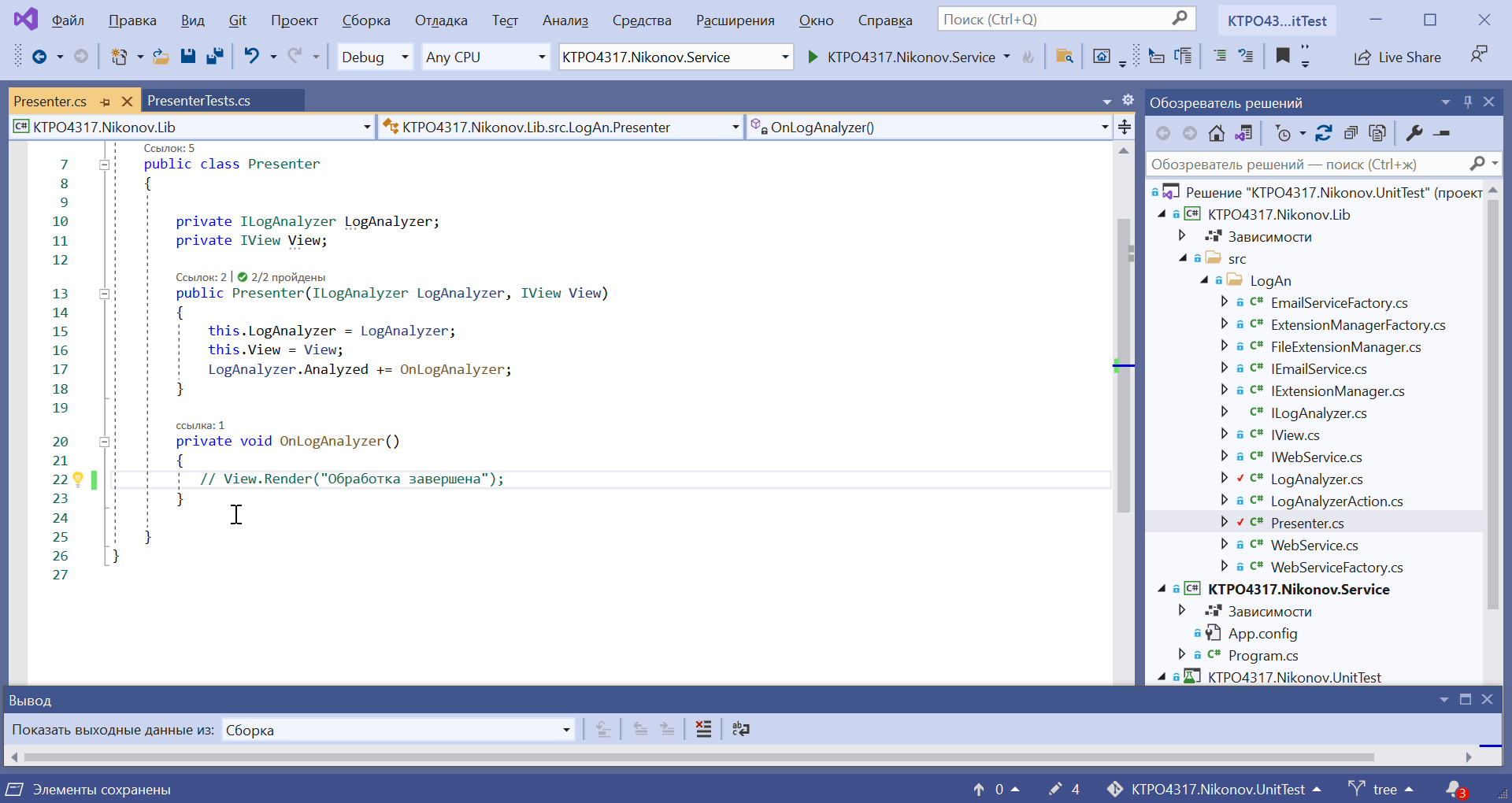
Постройте решение и выполните тесте. Зафиксируйте результат.

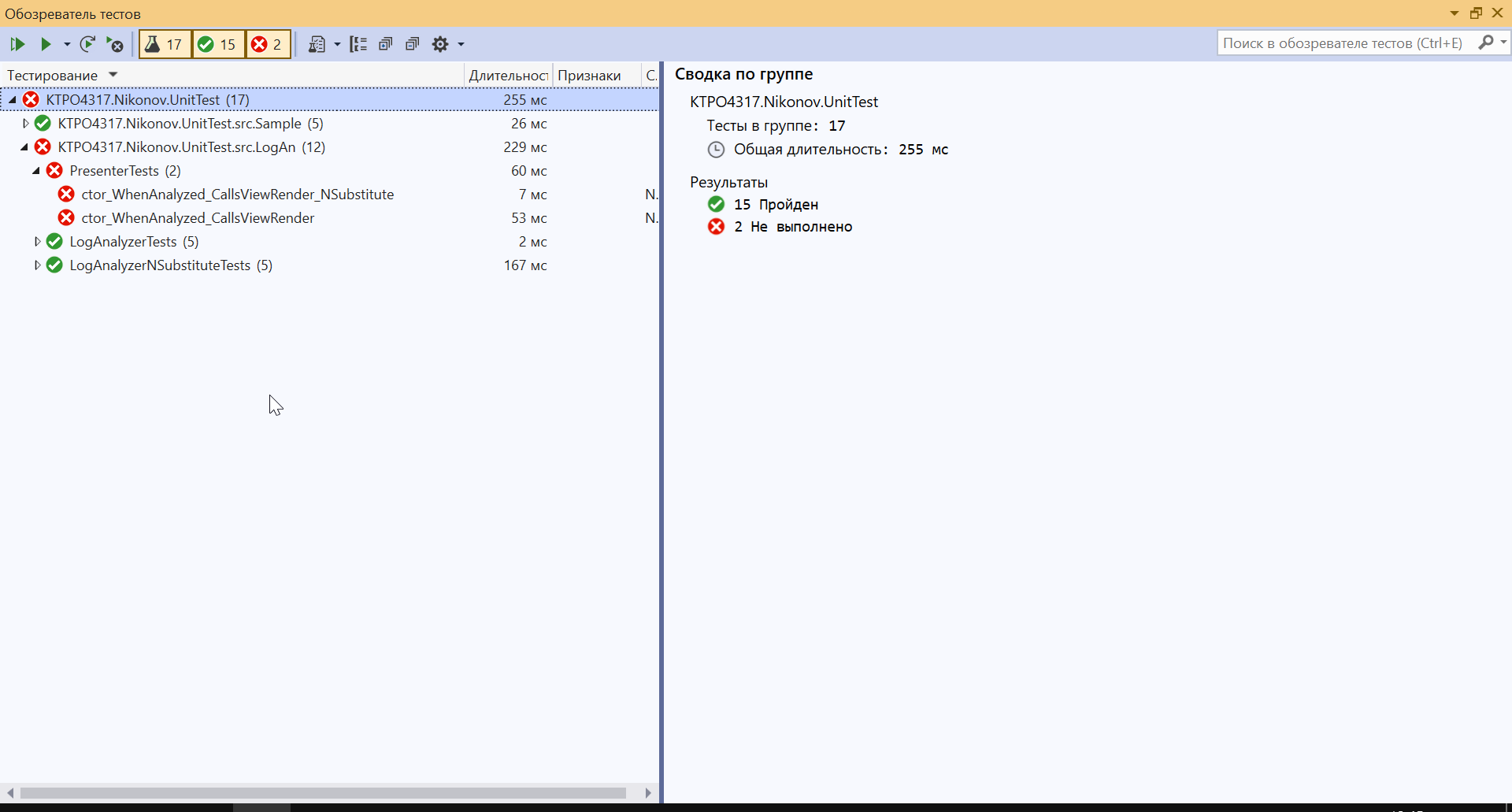


Создайте в PresenterTests тестовый метод ctor\_WhenAnalyzed\_CallsViewRender\_NSubstitute

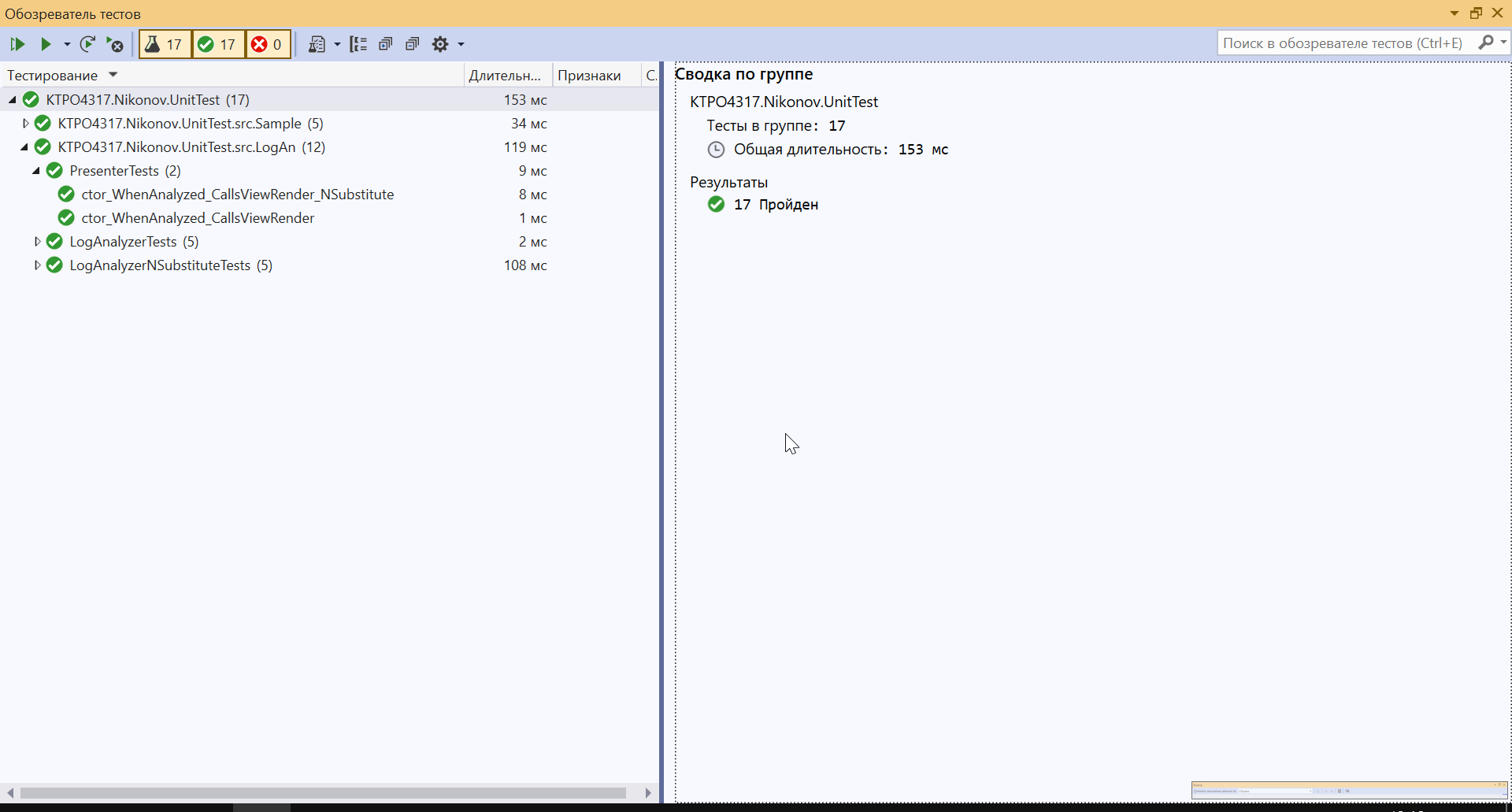


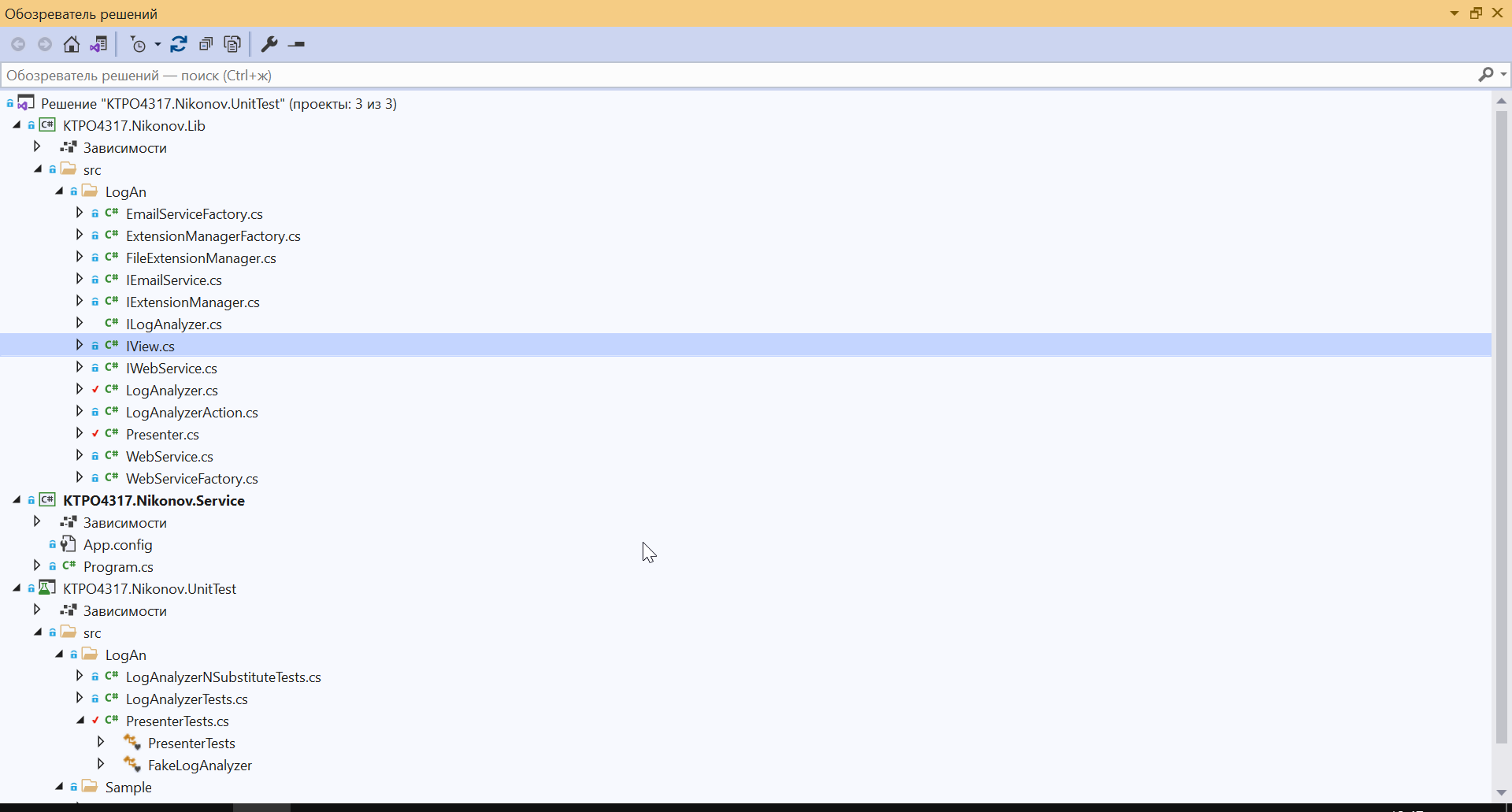
Проверьте правильность тестовых методов. Для этого внесите в тестируемый метод дефект, ошибку которую должен обнаружить тест. Выполните тест и зафиксируйте результат, в том числе и текст сообщения об ошибке. Убедитесь, что вариант теста и рукописной и динамической заглушкой работают одинаково.

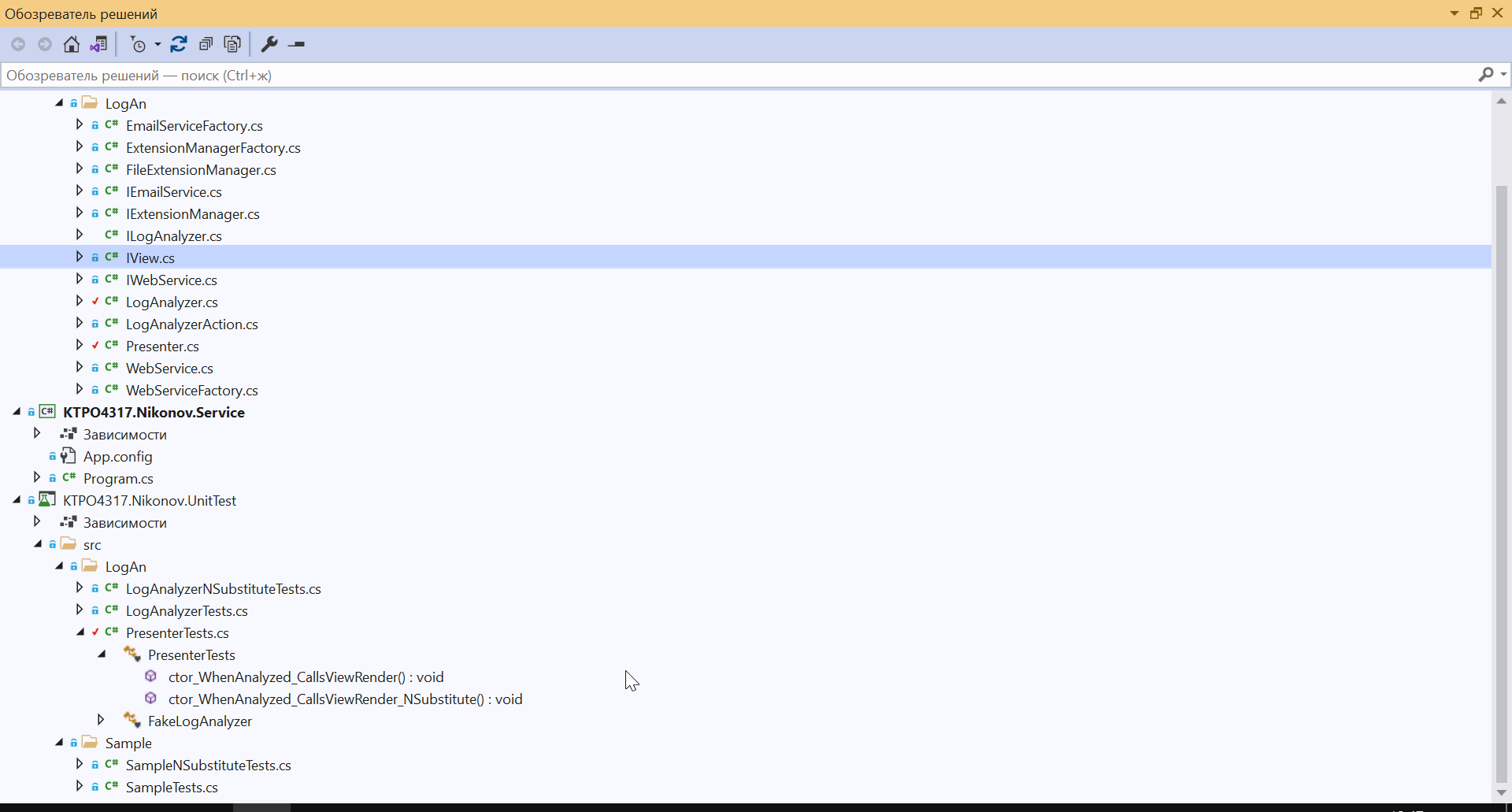




Восстановите правильный код.

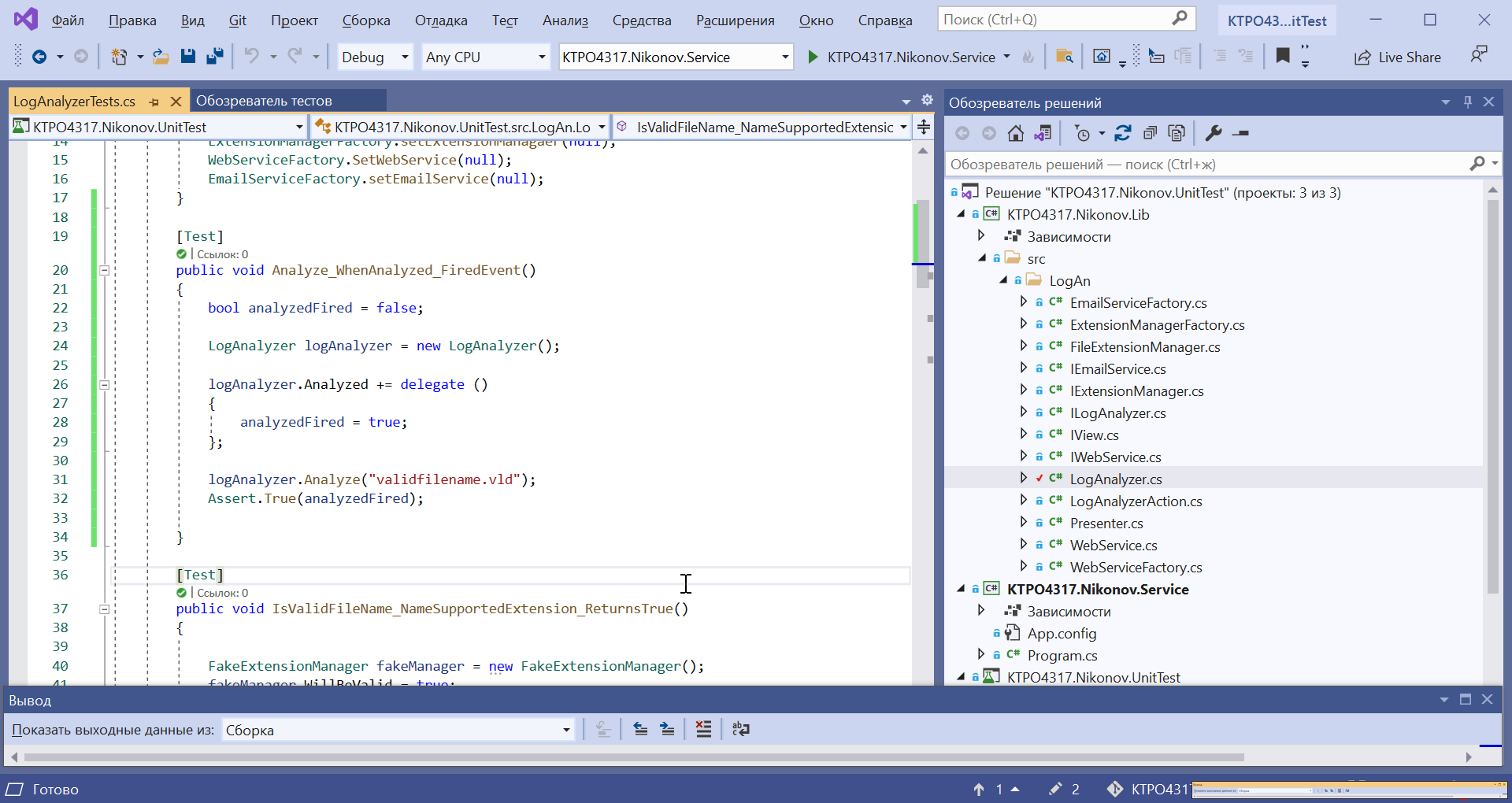




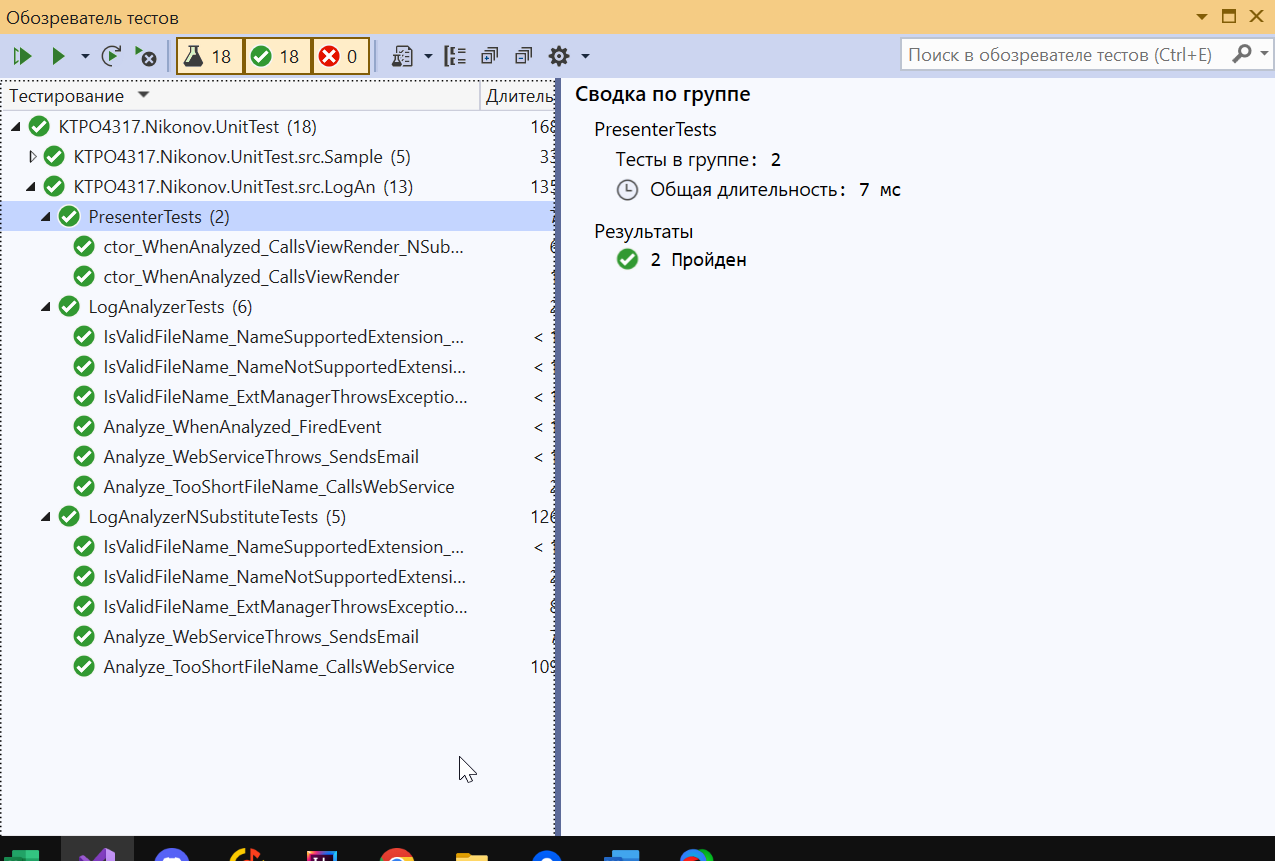


**6. Тестирования факта генерации события**

Создайте тестовый метод для сценария «Если файл проанализирован, то вызывается событие» Analyze\_WhenAnalyzed\_FiredEvent



Выполните тесты.



Проверьте правильность 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. Восстановите правильный код.

