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)

**ФУНКЦИОНАЛЬНОЕ РАСШИРЕНИЕ И РЕЛИЗ ПРОЕКТА**

Отчет по лабораторной работе №5

по дисциплине «Новые технологии в программировании»

Вариант №1

Студент гр. 588-2

\_\_\_\_\_\_\_\_\_\_\_\_ Робканов.К.Д

(подпись)

« 21 » июня 2021г.

Руководитель старший научный сотрудник, доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_ Горяинов А.Е.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_2021 г.

Томск 2021

**Содержание**

1 Введение……………………………………………………………………………3

2 Основная часть……………………………………………………………………..4

2.1 UML-диаграмма классов…………………………………………………4

2.2 Описание сценария сборки установочного пакета и его состава……...5

2.3 История изменений……………………………………………………….8

Заключение…………………………………………………………………………...9

**1 Введение**

**Цель работы:** провести вторую итерацию разработки приложения, расширив её дополнительными функциональными возможностями, а также получить умения сборки установочных пакетов.

**Задачи:**

1) Изучить процессы сборки установочного пакета и сдачи проекта по окончанию работ.

2) Разработать дополнительную функциональность в проекте согласно техническому заданию.

3) Собрать установочный пакет приложения с использованием программы InnoSetup.

4) Провести внутреннее приёмочное тестирование установленного приложения.

5) Провести приёмочное тестирование с руководителем для сдачи завершенной программы.

**2 Основная часть**

**2.1 UML-диаграмма классов**

На рисунке 2.1 представлена UML-диаграмма классов логики.

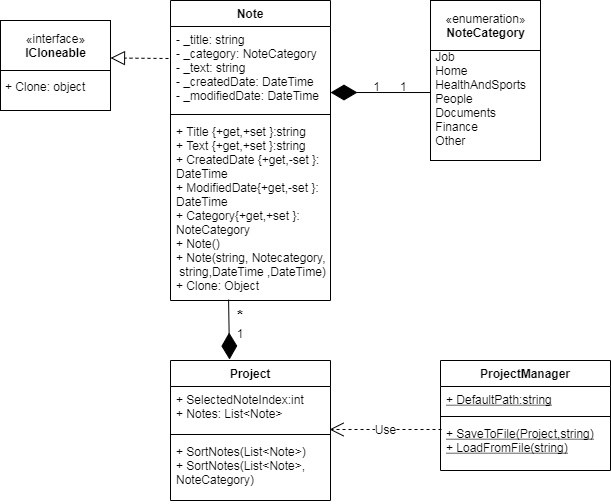
****

Рисунок 2.1 – UML-диаграмма классов логики

**2.2 Описание сценария сборки установочного пакета и его состава**

Обычно сборка и компиляция установщика выполняется в специальной программе на основе сценариев. Для десктоп-приложений такой программой является InnoSetup. Ее сценарии сохраняются в формате \*.iss и имеют специальный синтаксис (рисунок 2.2).

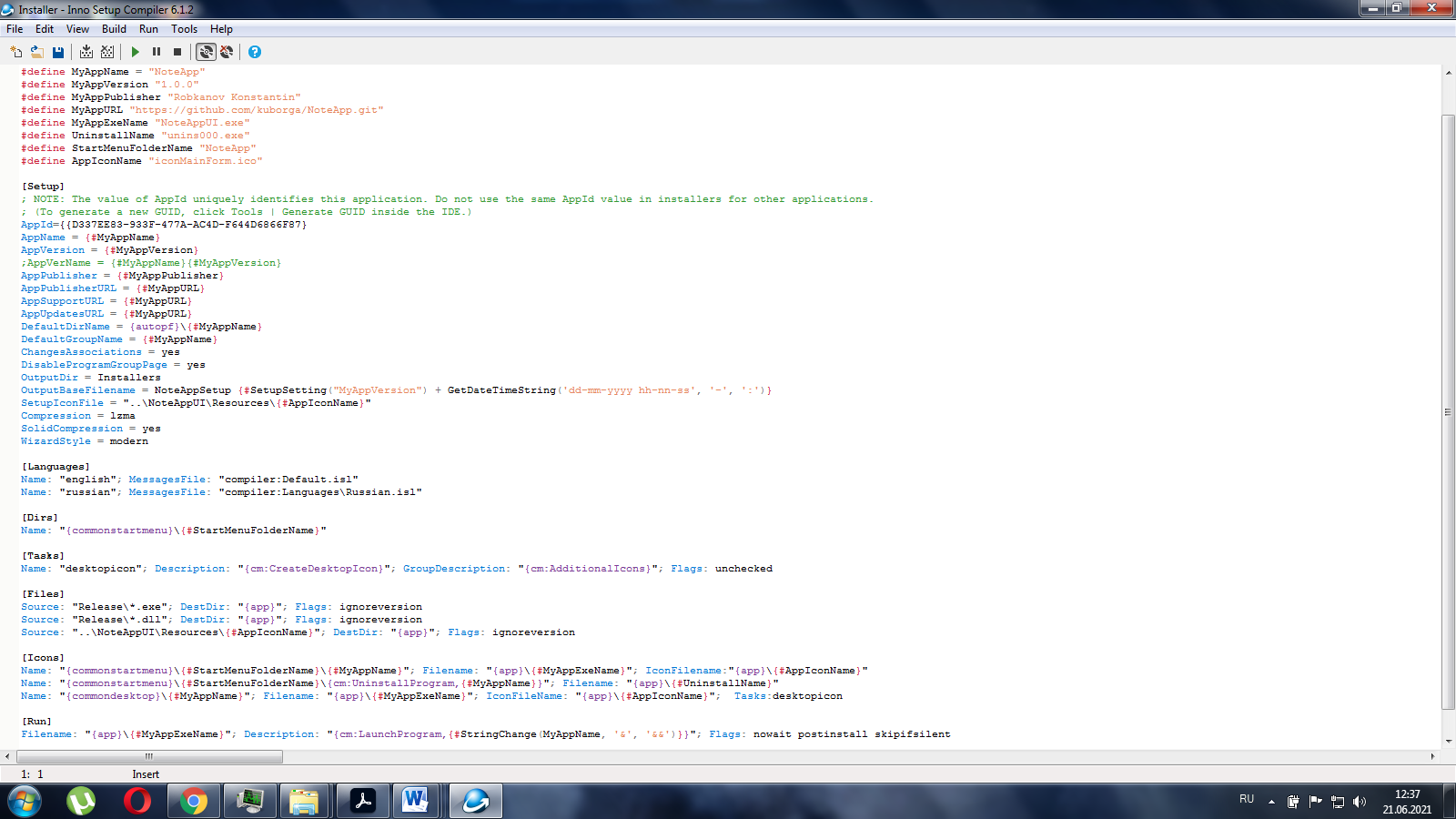


Рисунок 2.2 – Сценарий установщика

Для работы приложения необходимы только исполняемые файлы и библиотеки, соответственно, в установочный пакет помещаются только файлы с расширением \*.exe и \*.dll. Их можно указать с помощью масок имен файлов в секции [File] установочного сценария:

Source: "..\Release\\*.exe"; DestDir: "{app}"; Flags: ignoreversion

Source: "..\Release\\*.dll"; DestDir: "{app}"; Flags: ignoreversion

Таким образом, для разрабатываемого приложения в установочный пакет будут помещены следующие файлы:

1. NewtonsoftJson.dll – библиотека, необходимая для сериализации;
2. NoteApp.dll - скомпилированная библиотека проекта логики;
3. NoteAppUI.exe - скомпилированный исполняемый файл.

Так же необходимо добавить в свойстве проекта NoteAppUI сценарий(его код представлен на рисунке 2.3), который будет исполнен после сборки проекта.

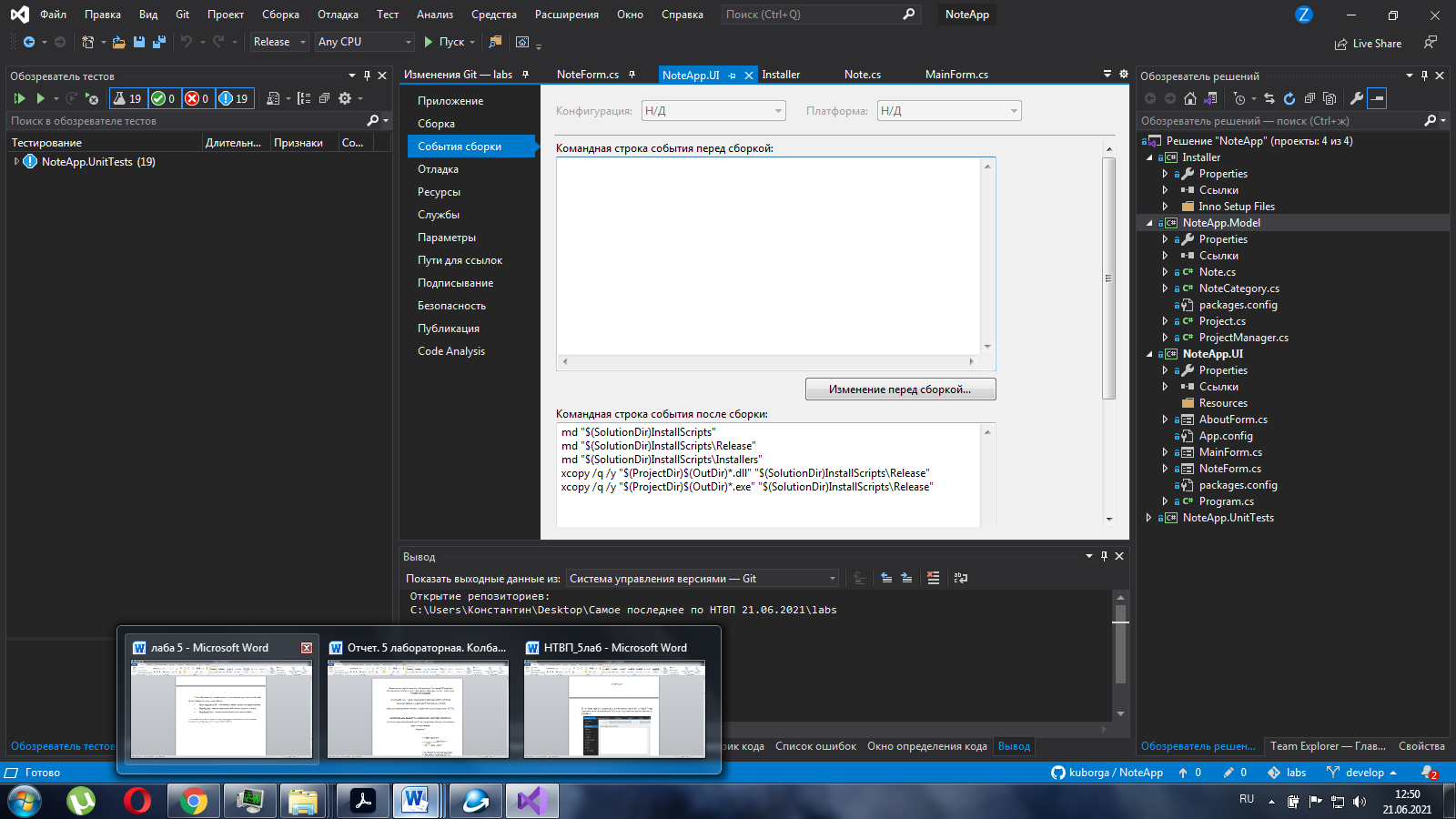


Рисунок 2.3 – Сценарий сборки

Данная строка создает папку InstallScripts в директории, где располагается файл NoteApp.sln.

md "$(SolutionDir)InstallScripts"

Данная строка создает папку Release в папке InstallScripts. Сюда будут помещены файлы с расширением exe и dll.

md "$(SolutionDir)InstallScripts\Release"

Данная команда создает папку Installers в папке InstallScripts. Сюда в дальнейшем помещен файл установщик.

md "$(SolutionDir)InstallScripts\Installers"

Данные команды копируют все файлы с расширением exe и dll из папки bin, куда собирается решение.

xcopy"$(ProjectDir)$(OutDir)\*.dll" "$(SolutionDir)InstallScripts\Release"

xcopy"$(ProjectDir)$(OutDir)\*.exe" "$(SolutionDir)InstallScripts\Release"

Команды выполняются после сборки проекта Installer, показаны на рисунке 2. 4.

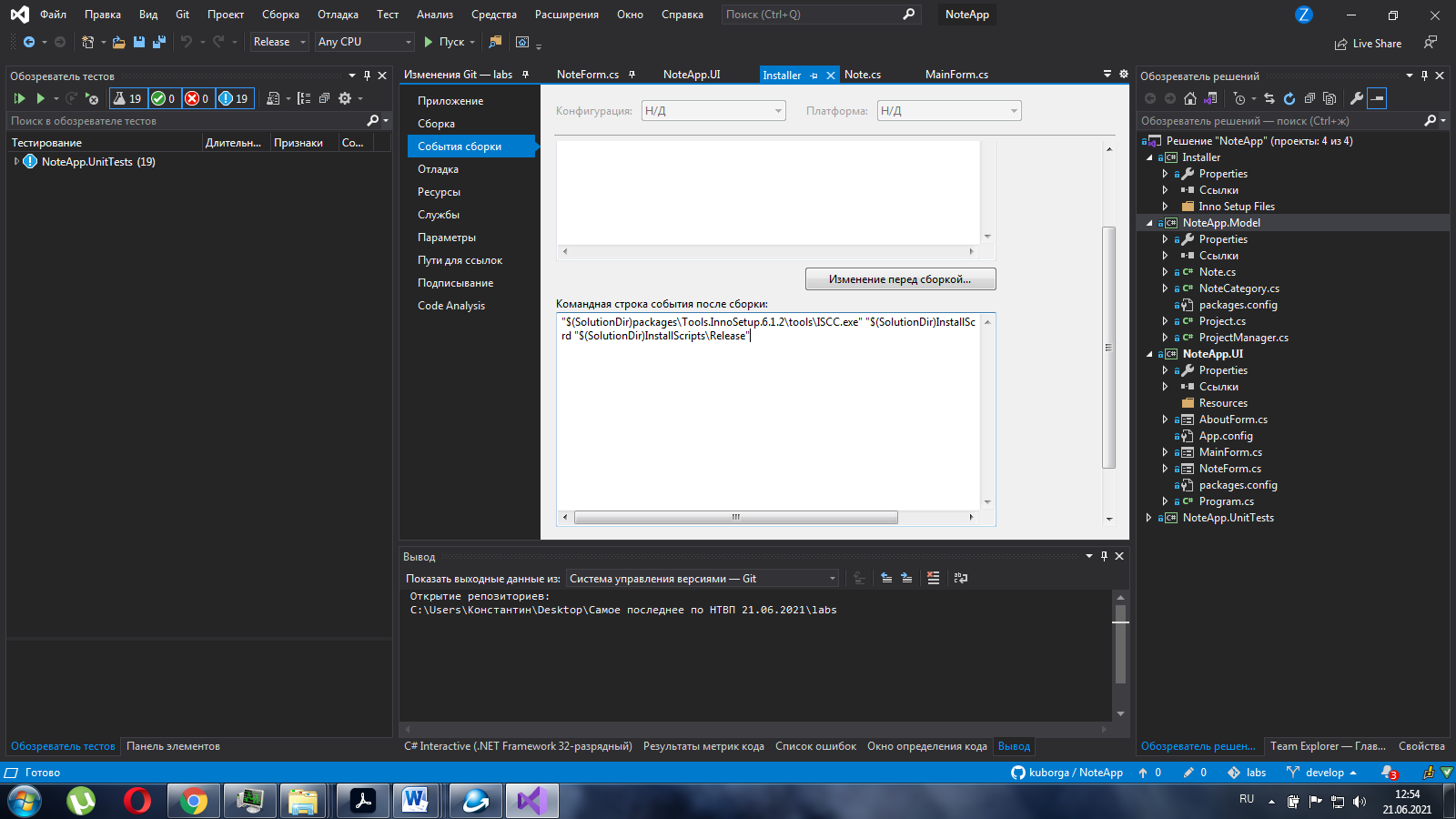


Рисунок 2.4 – Команды запуска установщика

Данная команда запускает файл установщика.

"$(SolutionDir)packages\Tools.InnoSetup.6.1.2\tools\ISCC.exe" "$(SolutionDir)InstallScripts\installer.iss"

Данная команда удаляет папку Realese со всеми файлами внутри.

rd /s/q "$(SolutionDir)InstallScripts\Release"

**2.3 История изменений**

По итогу выполнения лабораторной работы история изменений в репозитории на сервисе GitHub стала выглядеть следующим образом (рисунок 2.5).

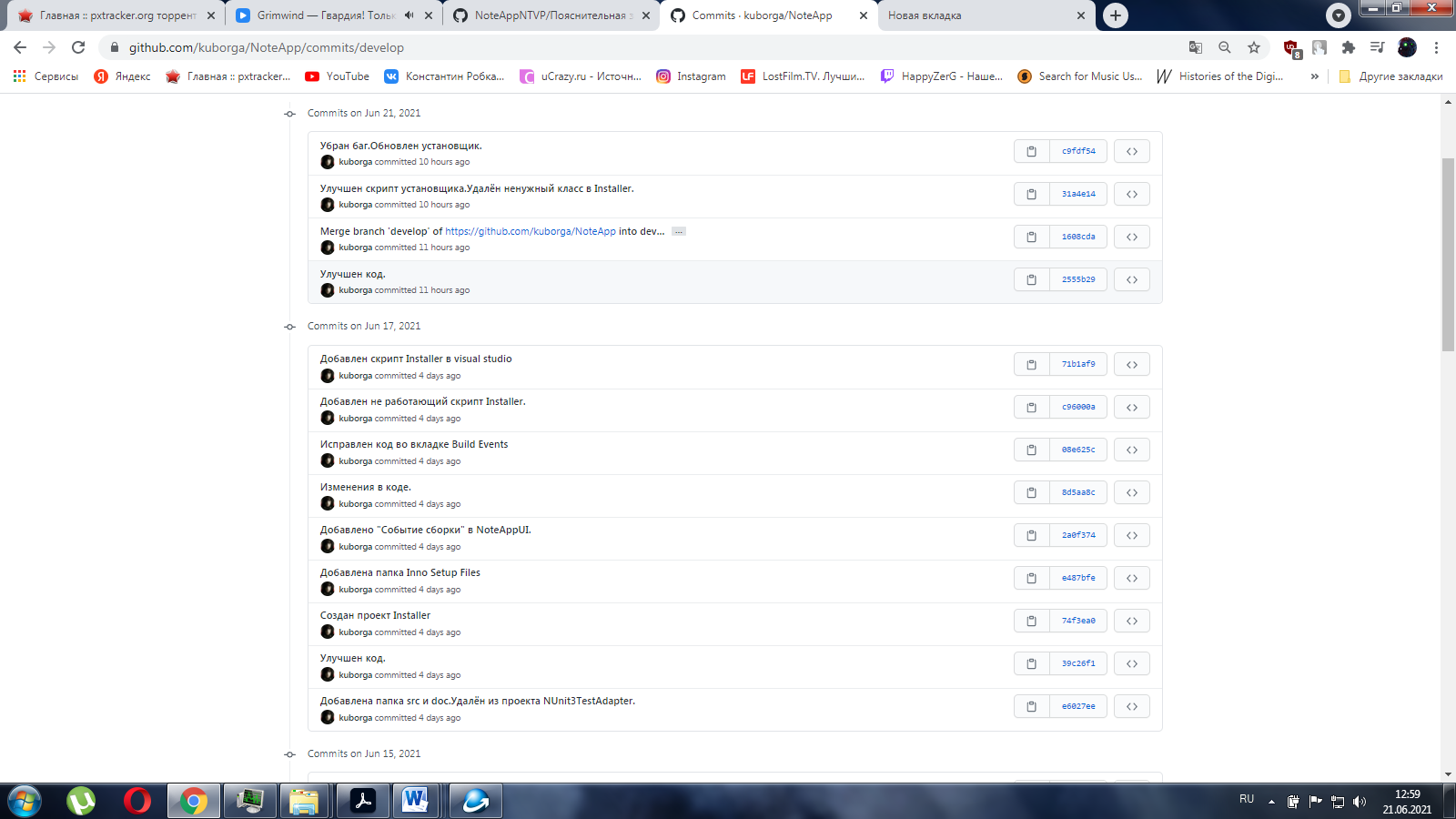


Рисунок 2.5 – История изменений в репозитории

**Заключение**

В ходе лабораторной работы проведена вторая итерация разработки приложения, расширив её дополнительными функциональными возможностями, а также получены умения сборки установочных пакетов.