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   
по дисциплине «Новые технологии в программировании»

Вариант №1

Студент гр. 588-2

\_\_\_\_\_\_\_\_\_ А.Д. Целицкий

« » 20\_\_

Руководитель

доцент кафедры КСУП

\_\_\_\_\_\_\_\_\_ А.Е. Горяинов  
 « » 20\_\_

Томск 2022

# ЛАБОРАТОРНАЯ РАБОТА №5

Функциональное расширение и релиз проекта

# Цель работы

Целью данной лабораторной работы является проведение второй итерации разработки приложения с расширением его дополнительными функциональными возможностями, а также получение умений сборки установочных пакетов.

# Постановка задачи

1. Изучить процессы сборки установочного пакета и сдачи проекта по окончании работ.
2. Разработать дополнительную функциональность в проекте согласно техническому заданию.
3. Собрать установочный пакет приложения с использованием программы InnoSetup.
4. Провести внутреннее приемочное тестирование установленного приложения.

# UML-диаграмма классов

Актуальная UML-диаграмма бизнес-логики разрабатываемого приложения NoteApp представлена на рисунке 1.

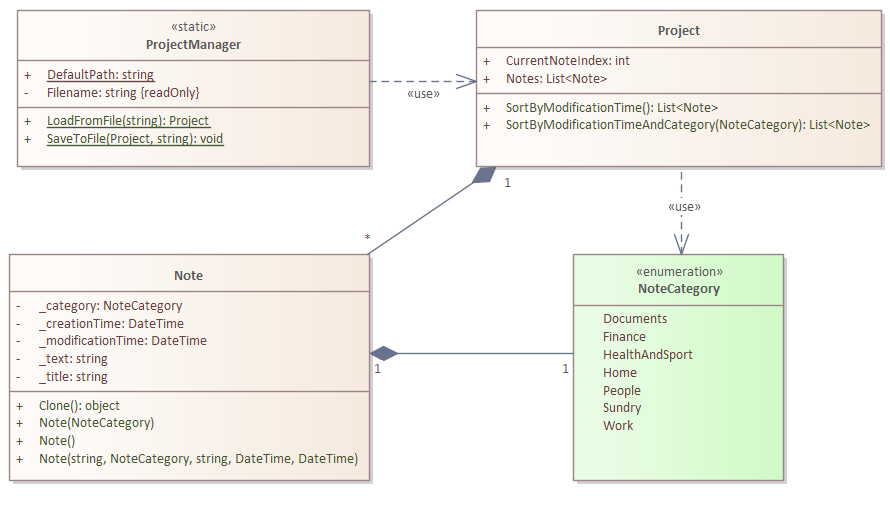


Рисунок 1 – Актуальная UML-диаграмма классов бизнес-логики приложения

# Описание сценария сборки установочного пакета и его состава

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

На рисунке 2 приведен сценарий сборки установочного пакета Installer.iss.

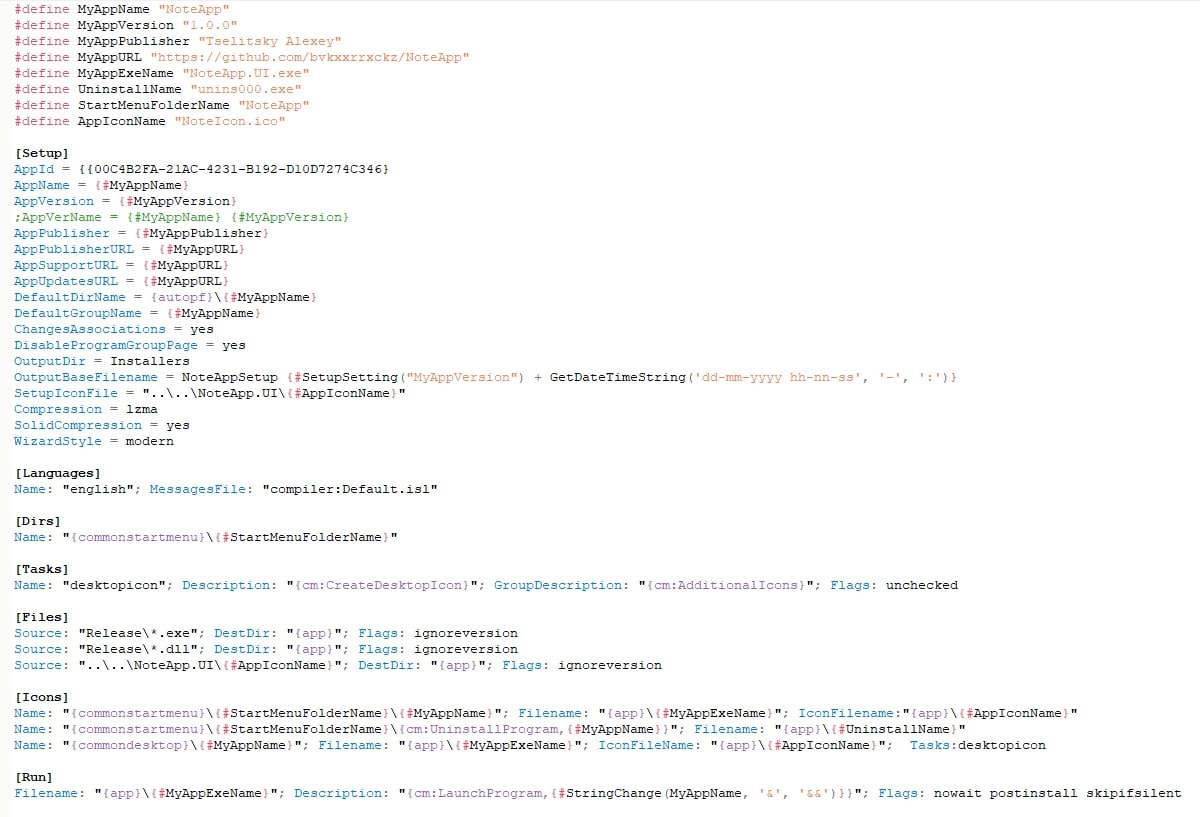


Рисунок 2 – Сценарий установщика Installer.iss, созданный в программе InnoSetup для установки разрабатываемого приложения NoteApp

В установочный пакет помещаются файлы, необходимые для работы вашего приложения. Также могут помещаться иконки, руководства пользователя, вспомогательные файлы примеров и т.д. – любые файлы, которые разработчик считает нужными для работы приложения у пользователя. Разумеется, в установочный пакет не стоит помещать файлы исходного кода – только скомпилированное приложение.

После компиляции программы в среде Visual Studio в конфигурации Release в соответствующей выходной папке появляются файлы скомпилированного приложения. Среди них есть скомпилированный исполняемый файл проекта, библиотека бизнес-логики приложения, а также дополнительные библиотеки, побочные файлы компиляции, файлы конфигурации и пр.

Для работы приложения необходимы только исполняемые файлы (\*.exe) и файлы библиотек (\*.dll). Данные файлы можно указать с помощью стандартных масок имен файлов в секции [File] сценария установщика:

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

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

Таким образом, для работы приложения NoteApp в установочный пакет помещены следующие файлы:

1. Newtonsoft.Json.dll – библиотека, необходимая для работы сериализации в бизнес-логике;
2. NoteApp.Model.dll – скомпилированная библиотека проекта бизнес-логики;
3. NoteApp.UI.exe – скомпилированный исполняемый файл проекта NoteApp.UI.
4. NoteIcon.ico – иконка главного окна приложения.

# Текущая история коммитов ветки develop

По результатам выполнения данной лабораторной работы текущая история изменений ветки develop в удаленном репозитории на сервисе GitHub выглядит следующим образом (рисунок 3).

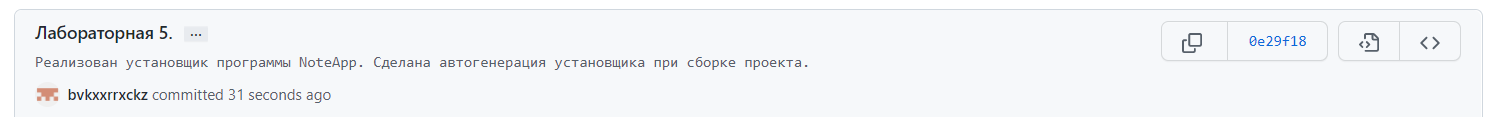


Рисунок 3 – Текущая история коммитов ветки develop

# Вывод

В данной лабораторной работе проведена вторая итерация разработки приложения с расширением его дополнительными функциональными возможностями, а также получены умения сборки установочных пакетов.