Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании

(КСУП)

ОТЧЁТ

Пояснительная записка к программе, разработанной по дисциплине

«Новые технологии в программировании»

Выполнил: студент гр. 588-2

Лапардин А.С.

«01» июня 2021

Проверил: к.т.н., доцент кафедры КСУП

Горяинов А.Е.

« » \_\_\_\_\_\_\_ 2021

Томск 2021

# Назначение приложения

Пользовательское приложение NoteApp, предназначено для ведения персональных записей и заметок. Приложение должно:

1. Обеспечивать категоризацию заметок, навигацию по созданным заметкам.
2. Предоставить инструменты для просмотра и редактирования заметок.
3. Сохранять и восстанавливать заметки между сессиями приложения.
4. Выполнять промежуточные сохранения заметок на машине пользователя на случай аварийного завершения программы, отключения компьютера и т.д. – для защиты от потери данных.

# 2 Группы пользователей и их функциональные возможности в приложении

Пользователь имеет только одну роль в приложении. Возможности пользователя:

1. Просмотр заметок;
2. Создание и редактирование заметок;
3. Сортировка по категориям заметок, навигация по созданным заметкам;
4. Загрузка уже существующих заметок и сохранение новых.

# 3 Стек технологий разработки. Системные требования.

Для проекта был выбран язык C#, и фреймворк Windows Forms.

Для сериализации и десериализации была выбрана библиотека JSON.NET от компании Newtonsoft версии 13.0.1.

Для юнит-тестов была выбрана библиотека NUnit версии 3.13.1.

# 4 Пользовательский интерфейс

На рисунке 4.1 показано главное окно приложения (NoteApp). В этом окне пользователь в левой части окна видит список заметок, созданных им, в правой части он видит информацию о выбранной заметке.

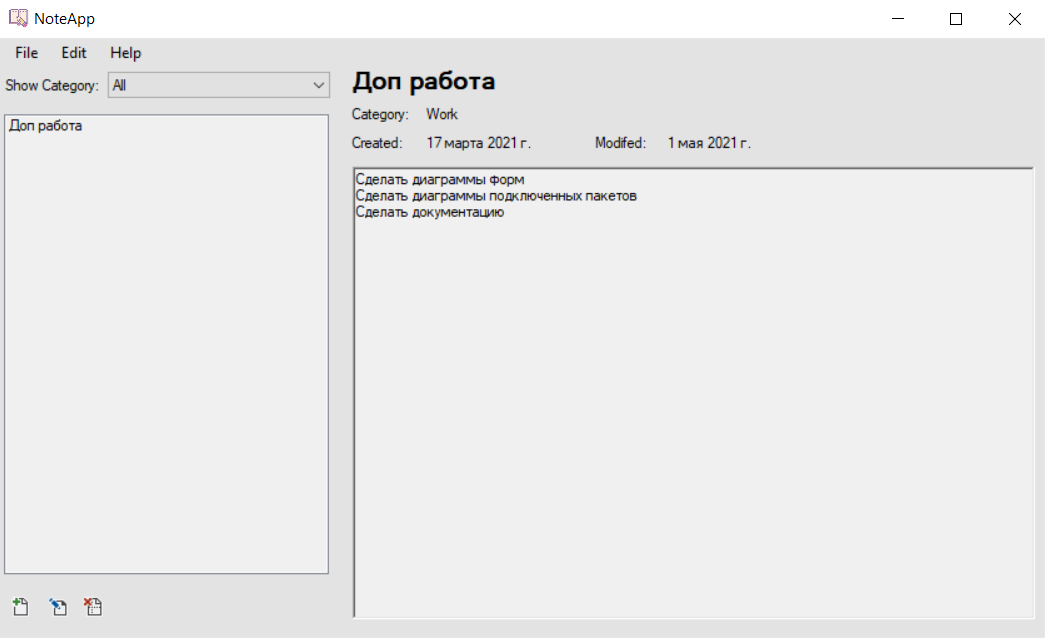


Рисунок 4.1 – Главное окно программы.

На рисунке 4.2 показано окно создания и редактирования заметки (Add/Edit Note). В этом окне пользователь может создать или отредактировать уже существующую заметку.

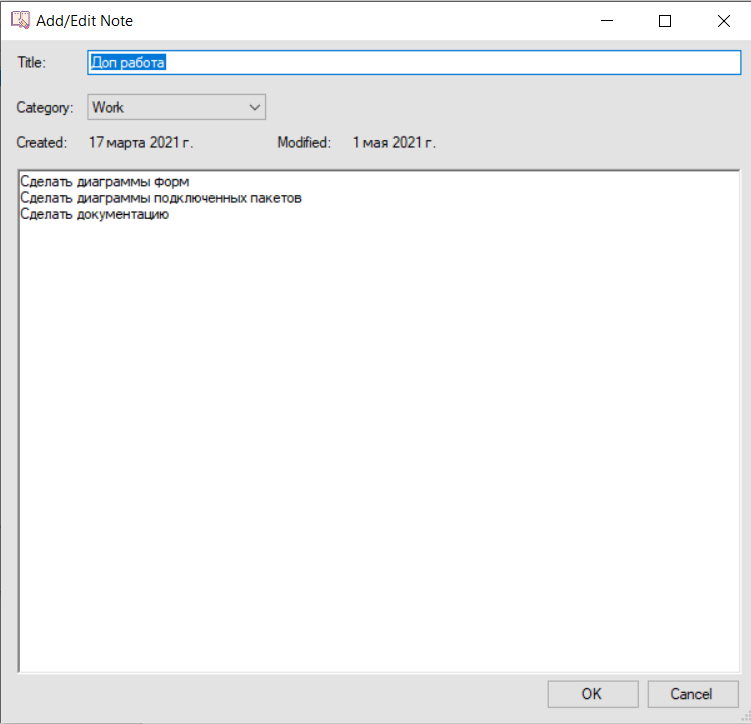


Рисунок 4.2 – Окно создания/редактирования заметки.

На рисунке 4.3 показано окно о приложении (About). В этом окне пользователь видит название приложения, версию приложения, автора и как с ним связаться.

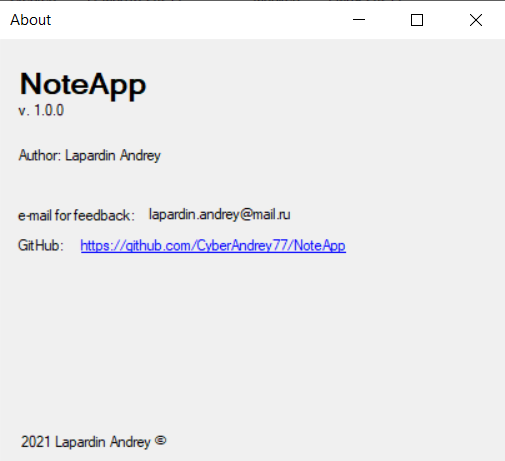


Рисунок 4.3 – Окно о программе.

# 5 Диаграмма пакетов

На рисунке 5.1 показана диаграмма пакетов.

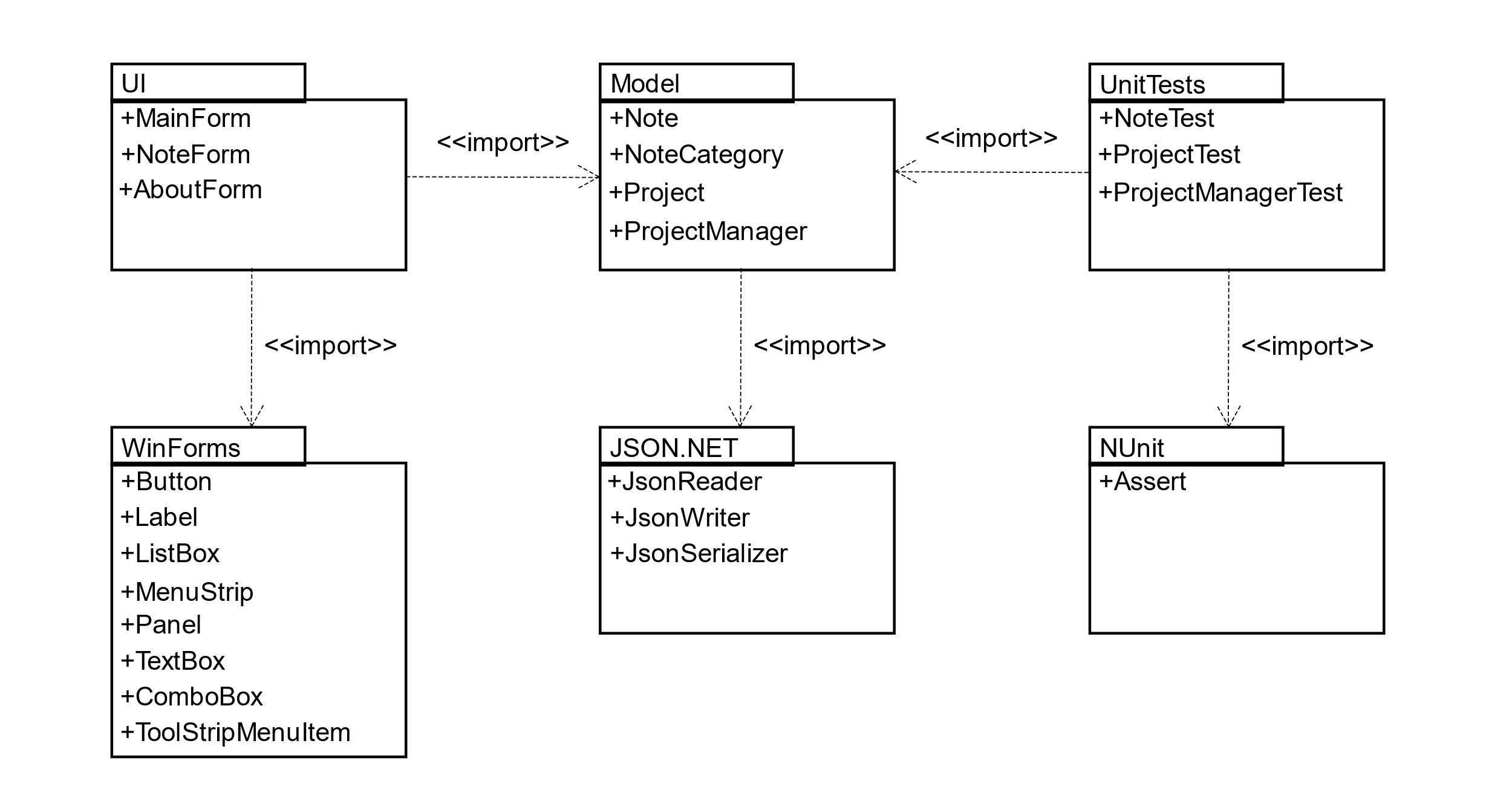


Рисунок 5.1 – Диаграмма пакетов

# 6 Диаграмма классов

На рисунке 6.1 показана диаграмма классов бизнес-логики приложения.

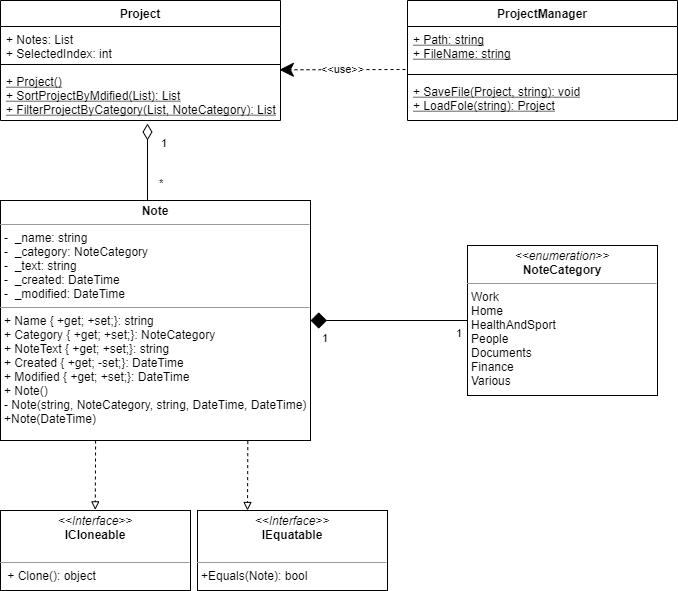


Рисунок 6.1 – Диаграмма классов бизнес-логики приложения

# 7 Описание тестирования приложения.

Для тестирования приложения была использована библиотека NUnit. С помощью неё были написаны тесты для тестирования бизнес-логики приложения. На рисунке 7.1 показано степень покрытия кода юнит-тестами.

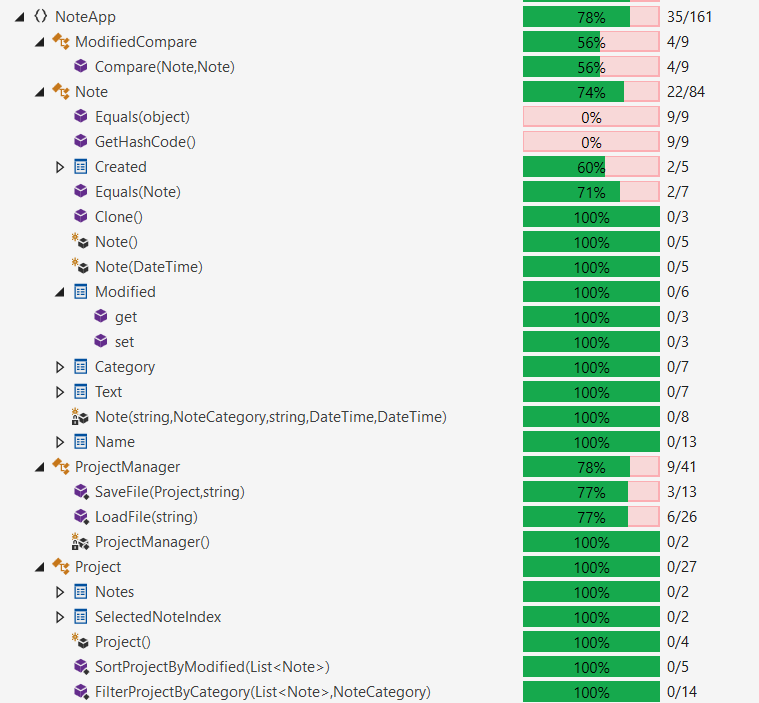


Рисунок 7.1 – Покрытие кода юнит-тестами

# 8 Сборка установщика

Сборка проекта осуществляется в автоматическом режиме. Для создания сценария сборки инсталлятора, использовалась программа InnoSetup.

В свойстве проекта NoteAppUI прописываем сценарий, который будет исполнен после сборки проекта.

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"

В свойстве проекта Intaller прописываем сценарий, который также исполнятся после сборки проекта.

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"

Ниже указан код, с помощью которого InnoSetup соберет установщик.

В данных строках указываются ссылка на репозиторий с проектом, название файла, который будет запускать приложение после установки, название приложения и его версию.

#define URL "https://github.com/CyberAndrey77/NoteApp"

#define ExeName "NoteAppUI.exe"

#define Name "NoteApp"

#define Version "0.0.1"

[Setup]

Это id приложения, которое однозначно идентифицирует приложение и записывается в реестре операционной системы.

AppId={{59F628AE-79AD-4A60-8757-84560EB97CA8}

AppName={#Name}

AppVersion={#Version}

AppPublisherURL={#URL}

AppSupportURL={#URL}

AppUpdatesURL={#URL}

Это папку для установки приложения по умолчанию, pf это program files соответствующей разрядности.

DefaultDirName={pf}\{#Name}

Тут указывается как называется папка в меню пуск.

DefaultGroupName={#Name}

DisableProgramGroupPage=yes

OutputBaseFilename=NoteAppSetup

Тут указывается путь до иконки приложения.

SetupIconFile="..\..\icon.ico"

UninstallDisplayName=uninslall

Тут указываем путь куда положить скомпилированный установщик.

OutputDir = "Installers"

Compression=lzma

SolidCompression=yes

Указываем на каком языке будет производиться установка приложения.

[Languages]

Name: "english"; MessagesFile: "compiler:Default.isl"

Указываем какие задачи будут выполнены вовремя установки. В данном случае спрашиваем у пользователя, нужно ли ставить ярлык на рабочий стол.

[Tasks]

Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDescription: "{cm:AdditionalIcons}"; Flags: unchecked

Здесь указываем какие файлы используются установщиком для сборки.

[Files]

Source: "Release\NoteAppUI.exe"; DestDir: "{app}"

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

Source: "..\..\icon.ico"; DestDir: "{app}"

Здесь мы указываем какие иконки должны быть у ярлыка на рабочем столе и в меню пуск.

[Icons]

Name: "{group}\{#Name}"; Filename: "{app}\{#ExeName}"; WorkingDir: "{app}"; IconFilename: {app}\icon.ico

Name: "{group}\uninslall"; Filename: "{uninstallexe}"; IconFilename: {app}\icon.ico

Name: "{commondesktop}\{#Name}"; Filename: "{app}\{#ExeName}"; Tasks: desktopicon; IconFilename: {app}\icon.ico

# 9 Модели ветвления

В рамках разработки приложения использовалась программа для версионного контроля GitHub. Для правильной работы создавалось две ветви main и develop. В первой ветви должна находиться работоспособная и стабильная версия программы, а во второй - программа на стадии разработки и тестирования.

Ссылка на GitHub <https://github.com/CyberAndrey77/NoteAppNTVP.git>