Министерство образования и науки

Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Пояснительная записка к приложению NoteApp

По дисциплине

«Новые технологии в программировании» (НТвП)

Выполнил: Студент гр. 587–2

Адамчук И. А.

«\_\_» 2020 г.

Принял: доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_Горяинов А. Е.

« » 2020 г.

Томск 2020

Содержание

[1 Назначение приложения 3](#_Toc533353550)

[2 Группы пользователей и их функциональные возможности о приложении 3](#_Toc533353551)

[3 Стек технологий разработки. Системные требования 4](#_Toc533353552)

[4 Пользовательский интерфейс 5](#_Toc533353553)

[5 Диаграммы пакетов приложения 8](#_Toc533353554)

[6 Диаграммы классов приложения 9](#_Toc533353555)

[7 Описание тестирования приложения 10](#_Toc533353556)

[8 Описание сборки установщика 11](#_Toc533353557)

[9 Описание модели ветвления в репозитории проекта 12](#_Toc533353557)

[Приложение А 1](#_Toc533353557)3

# 1 Назначение приложения

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

# 2. Группы пользователей и их функциональные возможности в приложении.

1. Обеспечивать стабильную работу приложения при порядке 200 заметок;
2. Обеспечивать категоризацию заметок, навигацию по созданным заметкам;
3. Предоставить инструменты для просмотра и редактирования заметок;
4. Сохранять и восстанавливать заметки между сессиями приложения;
5. Выполнять промежуточные сохранения заметок на машине пользователя на случай аварийного завершения программы, отключения компьютера и т.д. – для защиты от потери данных.

Пользователи системы не разделены на группы, и вся функциональность приложения является доступной.

# 3. Стек технологий разработки. Системные требования.

Для реализации проекта был задан язык программирования С# на платформе .NET 4.7.1, набор библиотек Windows Forms для создания десктоп-приложений для операционной системы Windowsи среда разработки Visual Studio 2017. Системные требования обусловлены требованиями к платформе .NET.

Для поддержки сериализации и десериализации как механизма файлового сохранения данных в проекте разработки была использована библиотека Newtonsoft JSON.NET.

Для проведения юнит-тестирования проекта логики приложения в среде Visual Studio была использована библиотека NUnit c ее пакетами NUnit.

Для создания сценария сборки установочного пакета и компиляции установщика используется программа Inno Setup.

# 4 Пользовательский интерфейс (на примере реальных данных).

Вся работа с приложением осуществляется через два окна пользовательского интерфейса – основное окно с полным списком заметок и окно редактирования информации выбранной или новой заметки. Для дополнительной информации о программе и ее разработчике предусмотрено специальное окно, вызываемое из основного.

При запуске приложения появляется основное окно (рисунок 4.1). Оно разделено на две части по вертикали: в левой части находится список всех заметок, которые представлены в виде их названий, а на правой панели отображается информация о выбранной записи.

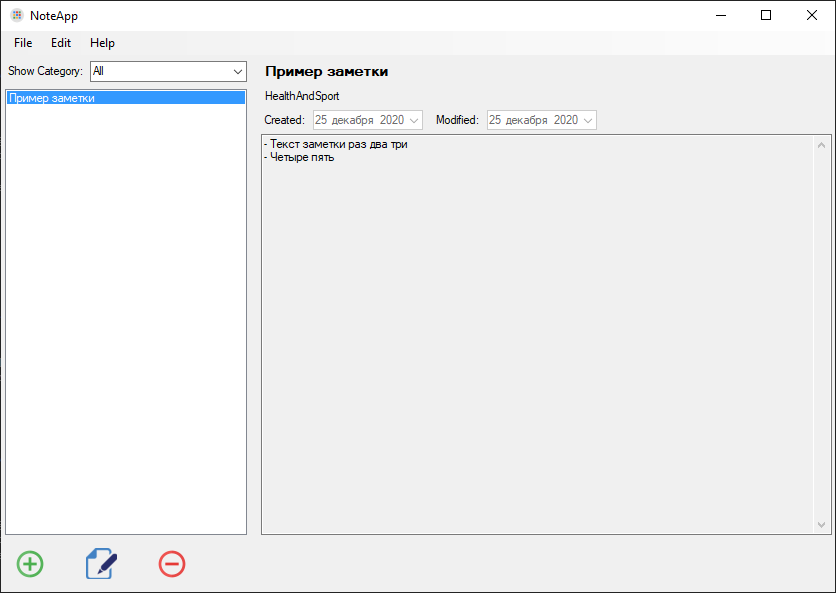


Рис.4.1 – Интерфейс главного окна программы

Для работы с заметками на форме есть элементы, вызывающие функции создания, редактирования, удаления записей и соответствующую форму. Они выполнены в виде кнопок с рисунками под списком заметок. Также все эти действия продублированы в верхнем меню главного окна во вкладке Edit.

В случае добавления новой заметки или изменения старой вызывается окно редактирования записи (рисунок 4.2). Если запись редактируется, то открывшееся окно содержит информацию этой записи. При создании новой записи открывается пустое окно, котором необходимо заполнить данными для новой заметки. Чтобы внесенные изменения были зафиксированы, необходимо выйти из формы по нажатию клавиши ОК, в противном случае все изменения будут отменены.

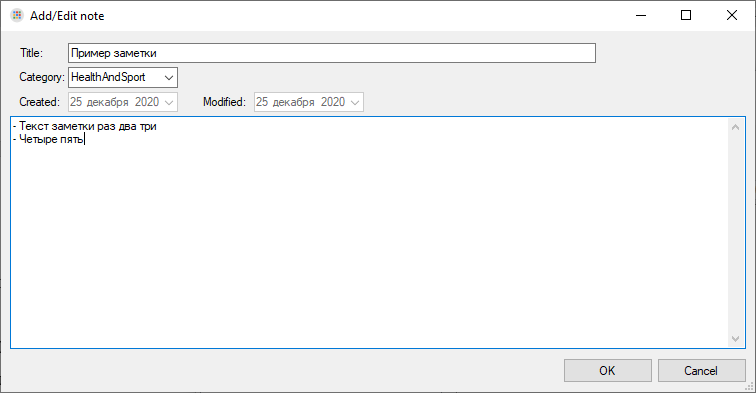


Рис.4.2 – Интерфейс окна для создания и редактирования заметки

При вызове окна изменения или добавления заметки происходит передача данных сначала из основной формы в вызываемую, а затем после подтверждения изменений осуществляется обратная передача данных и сохранение их в списке всех заметок. Удаление происходит с вызовом диалогового окна, на котором требуется подтвердить выбранное действие.

Из основного окна программы можно открыть еще одно окно (рисунок 4.3), содержащее информацию о приложении и его разработчике. Для этого необходимо в меню главного окна перейти в раздел About. В этой форме нет доступных для использования элементов.

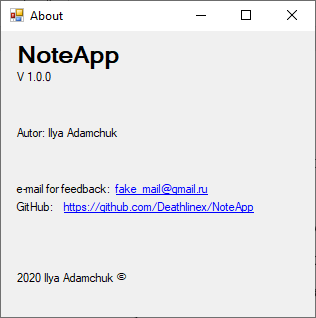


Рис.4.3 – Интерфейс окна дополнительной информации

Оставшийся пункт меню главного окна File реализует выход из программы.

# 5 Диаграммы пакетов приложения.

Диаграмма пакетов отображает архитектуру приложения, разделенную на отдельные пакеты – библиотеки. Внутри библиотек описываются доступные извне классы, между пакетами рисуются направленные линии, обозначающие связи между библиотеками. Диаграмма пакетов разработанного приложения представлена на рисунке 5.1.

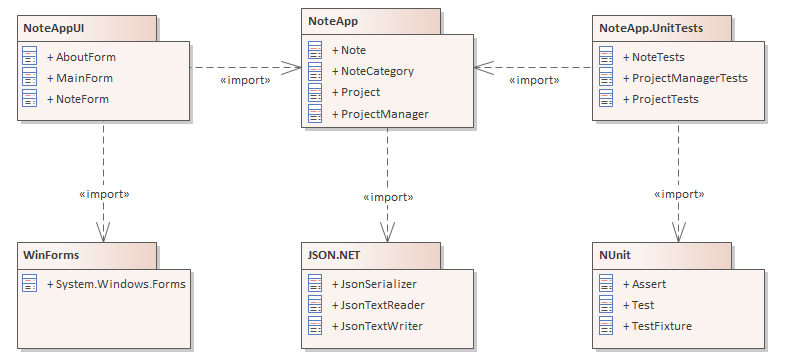


Рис.5.1 – Интерфейс окна дополнительной информации

# 6 Диаграммы классов приложения

Проект логики приложения состоит из следующих 4 классов:

1. Note – класс заметки;
2. NoteCategory – перечисление категорий заметки;
3. Project – класс списка заметок;
4. ProjectManager – класс для работы с файловым сохранением.

Диаграмма классов приложения представлена на рисунке 6.1.

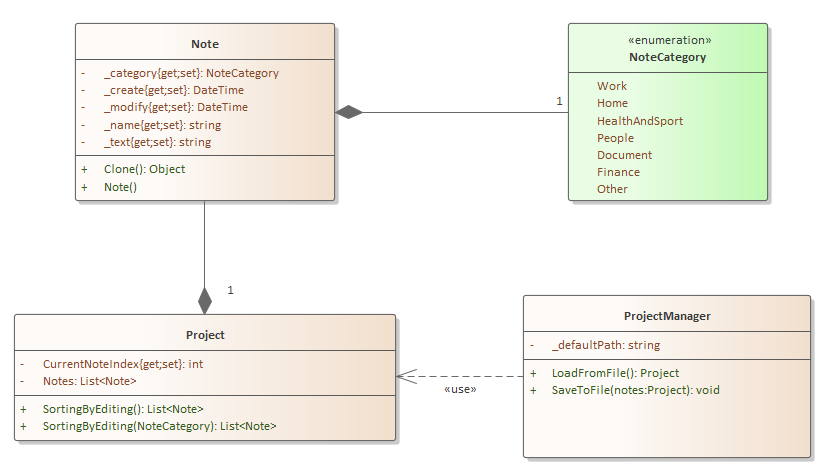


Рисунок 6.1 - Диаграмма проекта логики приложения

# 7 Описание тестирования приложения.

В данном проекте основным типом тестирования было юнит-тестирование. Это тестирование минимальных модулей архитектуры, максимально изолированных друг от друга. Так как минимальными модулями архитектуры, как правило, являются классы, в разработанном приложении тестировался проект бизнес-логики.

Были разработаны тесты, полностью покрывающие исходный код. На рисунке 7.1 представлена информация о проведении тестов. На рисунке 7.2 покрытие кода в процентах.

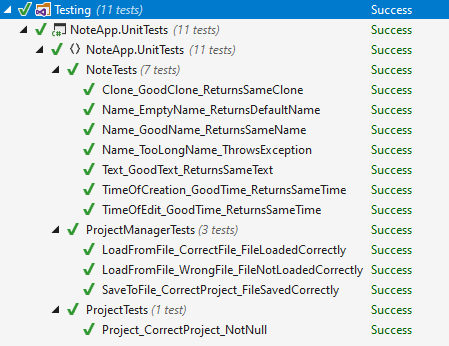


Рисунок 7.1 – Результаты тестирования.



Рисунок 7.2 – Покрытие кода юнит-тестами.

# 8 Описание сборки установщика.

Для сборки установочного пакета приложения используется программное обеспечение Inno Setup. С его помощью компилируется сценарий сборки (Приложение А) создающий установочный пакет.

В сценарии сборки указывается различная информация о приложении, а также стандартный путь установки, название установочного пакета и дополнительные особенности установщика. Необходимо указать файлы, требуемые для работы приложения, такие как \*.exe и \*.dll. Остальные файлы (\*.pdb, \*.config, \*.manifest, \*.xml и другие возможные файлы) для работы приложения не нужны и исключаются из установочного пакета.

# 9 Описание модели ветвления в репозитории проекта.

При разработке приложения использовалась система версионного контроля Git в локальной файловой системе проекта. Фиксации производились из среды разработки Visual Studio. Ссылка на репозиторий: https://github.com/Deathlinex/NoteApp

Работа над проектом велась в двух ветках репозитория:

1. Master - ветка, содержащая проверенную, протестированную, и готовую к включению в сборщик версию проекта
2. Develop – основная ветка разработки.

**Приложение А**

Сценарий установщика приложения

#define MyAppName "NoteApp"

#define MyAppVersion "1.0"

#define MyAppPublisher "KONAMI, Inc."

#define MyAppURL "https://github.com/Deathlinex/NoteApp"

#define MyAppExeName "NoteAppUI.exe"

[Setup]

AppId={{DCDA6598-A526-4AC3-B6DC-62E9173B1BC2}

AppName={#MyAppName}

AppVersion={#MyAppVersion}

AppPublisher={#MyAppPublisher}

AppPublisherURL={#MyAppURL}

AppSupportURL={#MyAppURL}

AppUpdatesURL={#MyAppURL}

DefaultDirName={autopf}\{#MyAppName}

ChangesAssociations=yes

DisableProgramGroupPage=yes

OutputDir=Installers

OutputBaseFilename=NoteAppSetup

Compression=lzma

SolidCompression=yes

WizardStyle=modern

[Languages]

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

Name: "russian"; MessagesFile: "compiler:Languages\Russian.isl"

[Tasks]

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

[Files]

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

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

[Registry]

Root: HKA; Subkey: "Software\Classes\{#MyAppAssocExt}\OpenWithProgids"; ValueType: string; ValueName: "{#MyAppAssocKey}"; ValueData: ""; Flags: uninsdeletevalue

Root: HKA; Subkey: "Software\Classes\{#MyAppAssocKey}"; ValueType: string; ValueName: ""; ValueData: "{#MyAppAssocName}"; Flags: uninsdeletekey

Root: HKA; Subkey: "Software\Classes\{#MyAppAssocKey}\DefaultIcon"; ValueType: string; ValueName: ""; ValueData: "{app}\{#MyAppExeName},0"

Root: HKA; Subkey: "Software\Classes\{#MyAppAssocKey}\shell\open\command"; ValueType: string; ValueName: ""; ValueData: """{app}\{#MyAppExeName}"" ""%1"""

Root: HKA; Subkey: "Software\Classes\Applications\{#MyAppExeName}\SupportedTypes"; ValueType: string; ValueName: ".myp"; ValueData: ""

[Icons]

Name: "{autoprograms}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"

Name: "{autodesktop}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"; Tasks: desktopicon

[Run]

Filename: "{app}\{#MyAppExeName}"; Description: "{cm:LaunchProgram,{#StringChange(MyAppName, '&', '&&')}}"; Flags: nowait postinstall skipifsilent