Министерство науки и образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПОЛЬЗОВАТЕЛЬСКОМУ ПРИЛОЖЕНИЮ «NoteApp»

Лабораторная работа № 6

по дисциплине «Системное программное обеспечение»

Выполнил:

Студент группы 516

\_\_\_\_\_\_\_\_Климентьев Р.Е.

«\_\_\_»\_\_\_\_\_\_\_\_ 2020 г.

Принял:

Доцент кафедры КСУП

\_\_\_\_\_\_\_\_Горяинов А.Е.

«\_\_»\_\_\_\_\_\_\_\_2020 г.

Томск 2020

Содержание

[1 Назначение приложения 3](#_Toc34444528)

[2 Стек технологий разработки. Системные требования. 3](#_Toc34444529)

[3 Пользовательский интерфейс 4](#_Toc34444530)

[4 Диаграмма пакетов 6](#_Toc34444531)

[5 Диаграмма классов 7](#_Toc34444532)

[6 Описание тестирование приложения 7](#_Toc34444533)

[6.1 Модульное тестирование 7](#_Toc34444534)

[6.2 Функциональное тестирование 8](#_Toc34444535)

[7 Сборка установщика 9](#_Toc34444536)

[8 Описание модели ветвления 10](#_Toc34444537)

[Список литературы 10](#_Toc34444538)

# 1 Назначение приложения

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

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

# 2 Стек технологий разработки. Системные требования.

Для проекта был выбран язык C# на целевой платформе .NET Framework 4.8. Интерфейс программирования приложения (API) выбран Windows Form т.к. данный набор компонентов относительно прост в изучении и хорошо подходит для учебных задач.

Для тестирования приложения была выбрана библиотека NUnit версии 3.12.0.

Для сохранения списка заметок выбран сериализатор Newtonsoft.Json версии 12.0.3.

# 3 Пользовательский интерфейс

После запуска приложения перед пользователем появляется главное окно (рис. 1). В левой части приложения показан выпадающее меню выбора категории и список заметок, в правой части содержание выбранной заметки.

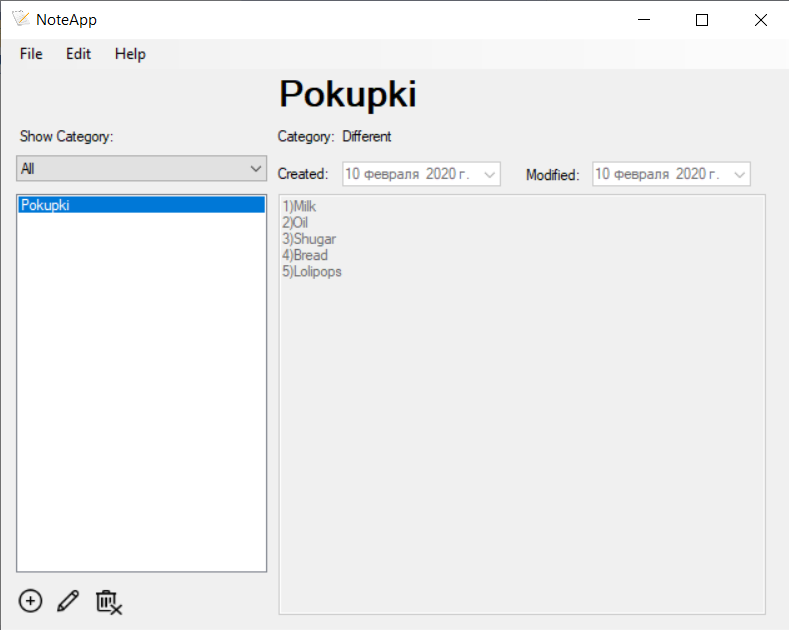
****

Рисунок 1 – Главное окно программы

На панели со списком заметок внизу располагаются три кнопки в виде иконок: Add («Создать новую заметку»), Edit («Редактировать текущую заметку»), Remove («Удалить текущую заметку»).

При выборе заметки в списке, выбранная заметка отображается в правой панели. Главное окно не позволяет редактировать содержимое заметки – только просмотр.

При нажатии на кнопку Add или Edit появляется окно создания/редактирования заметки в диалоговом режиме (рис. 3.2). Для новой заметки по умолчанию выбрана категория Home и имя «Без имени». Для редактирования уже существующей заметки все поля будут данными.

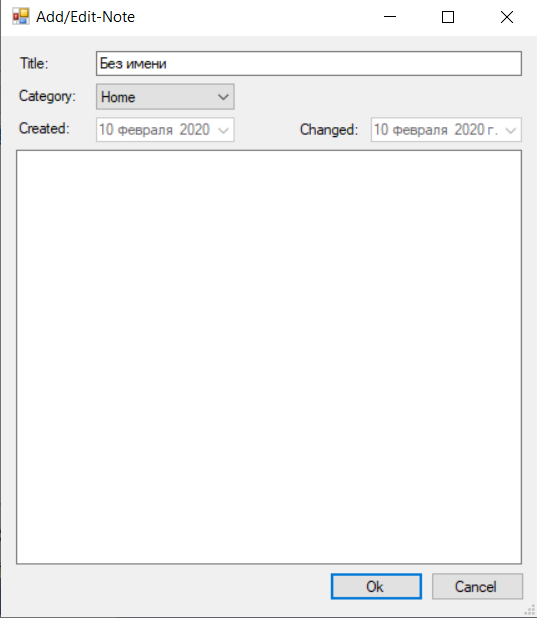


Рисунок 2 – Окно редактирования заметки

При нажатии на кнопку OK окно создания заметки закрывается, в список заметок главного окна добавляется новая заметка. После редактирования текущей заметки, при нажатии на кнопку OK, обновляется список заметок в главном окне. При нажатии кнопки Cancel создание/редактирование заметки отменяется (новая заметка не добавляется, исходная заметка остается без изменений).

Если нажать на кнопку ОК в разделе remove note,то заметка удаляется,а при нажатии на кнопку Cancel удаление отменяется.

Меню главного окна содержит следующие пункты:

* File:
  + Exit (Выйти из приложения – Alt+F4)
* Edit:
  + Add Note (Создать заметку)
  + Edit Note (Редактировать заметку)
  + Remove Note (Удалить заметку)
* Help
  + About (Вызвать окно «О программе») (рис. 3)

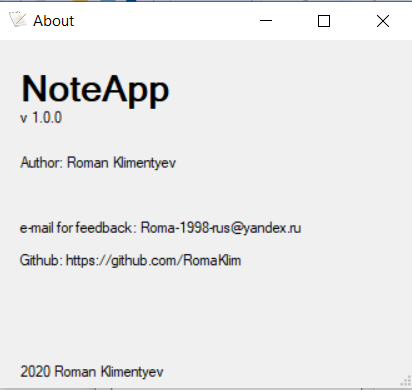


Рисунок 3 – Окно «About»

# 4 Диаграмма пакетов

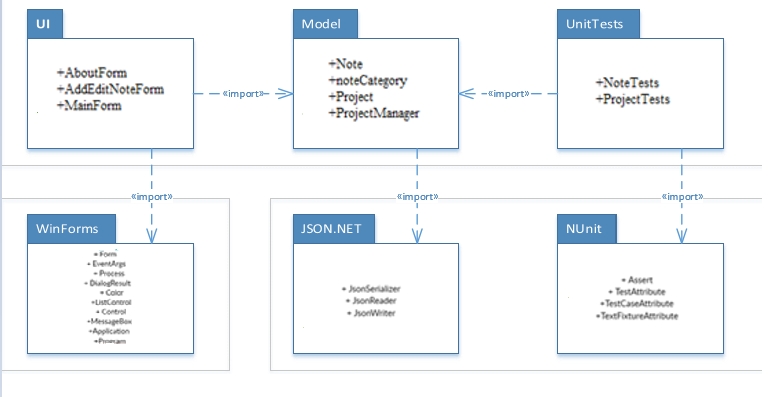
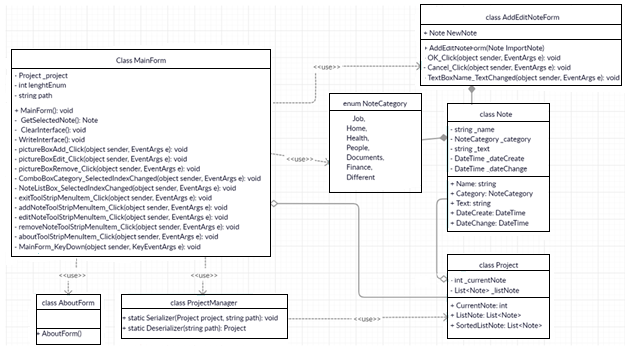


Рисунок 4 – Полный пакет диаграмм

# 

# 5 Диаграмма классов

****  
Рисунок 5 - Диаграмма классов

# 6 Описание тестирование приложения

Тестирование программы позволяет убедиться, что реальное поведение программы соответствует ожидаемому. Своевременное тестирование дает возможность выявлять ошибки на ранних стадиях разработки, что значительно снижает затраты на их устранение и в целом упрощает разработку в дальнейшую поддержку программы.

## **6.1 Модульное тестирование**

Модульное тестирование — процесс изолированного тестирования отдельных модулей программ для подтверждения корректности их работы. В данной программе для классов с использованием библиотеки модульного тестирования NUnit были написаны тесты для всех публичных свойств и методов. Были использованы как позитивные тесты, проверяющие правильность работы классов на корректных данных, так и негативные, проверяющие правильность типа выдаваемых классом исключений при подаче на вход некорректных данных. Общее количество написанных тестовых случаев — 9.Полный список юнит-тестов представлен на рисунке 6.1.

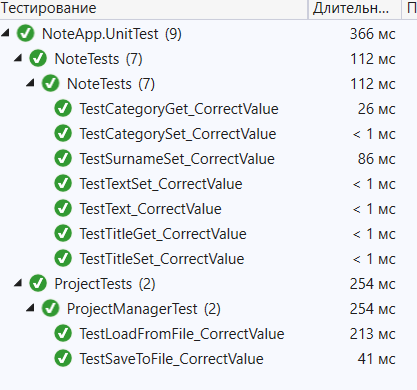
****

Рисунок 6.1 – Полный список Unit-тестов.

## **6.2 Функциональное тестирование**

Функциональное тестирование предназначено для проверки реализуемости в программе функциональных требований к ней, заданных ТЗ. Пример предупреждения удаления приведен на рисунке 6.2.

В случае ввода пользователем не допустимой длины вводимого имени выводится окно предупреждения на рисунке 6.3 и текущее имя заметки изменяется на: «Not true».

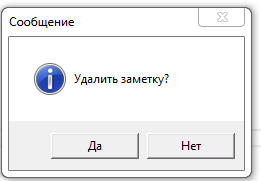
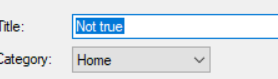


Рисунок 6.2 – Предупреждение при удалении заметки



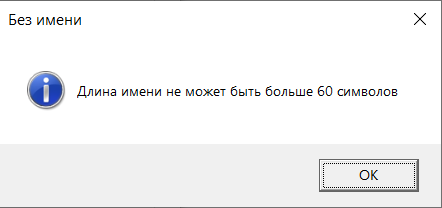


Рисунок 6.3 – Предупреждение при вводе имени

# 7 Сборка установщика

Для сборки установщика программы была использована свободно распространяемая утилита с открытым исходным кодом InnoSetup. С её помощью был создан установочный файл. Исполняемый файл программы, NoteApp.UI.exe, библиотека, необходимая для его работы библиотека NoteApp\InstallScripts\Release.

Сценарий сборки установочного пакета :

#define MyAppName "NoteAppEA"

#define MyAppVersion "1.2.0"

#define MyAppExeName "NoteApp.UI.exe"

[Setup]

AppId={{03F9C4BF-91B3-4CA5-AA49-8EDF948F8984}

AppName={#MyAppName}

AppVersion={#MyAppVersion}

DefaultDirName={autopf}\{#MyAppName}

DisableProgramGroupPage=yes

OutputDir=C:\Users\Администратор\Desktop\NoteApp.Model\InstallScripts\installers

OutputBaseFilename=SetupNoteApp

SetupIconFile=C:\Users\Администратор\Desktop\NoteApp.Model\NoteApp.UI\notepad\_bloc\_notes\_15548.ico

Compression=lzma

SolidCompression=yes

WizardStyle=modern

[Languages]

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

[Tasks]

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

[Files]

Source: "\InstallScripts\Release\NoteApp.UI.exe"; DestDir: "{app}"

Source: "\InstallScripts\Release\\*"; DestDir: "{app}"

[Icons]

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

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

[Run]

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

# 8 Описание модели ветвления

Для хранения исходного кода программы на сервисе GitHub был создан git репозиторий NoteApp. Ссылка на репозиторий –<https://github.com/RomaKlim/NoteApp>

Для разработки использовалось две ветки: master – главная ветвь, содержащая стабильную и отлаженную версию программы, а также developer – ветвь, для разработки и тестирования нового функционала.

# Список литературы

1. Команды командной строки [Электронный ресурс]. – URL: http://allmbs.ru/cmd-01.html (дата обращения 10.01.2020).

2. М. Фаулер. UML. Основы, 3-е издание. Книга по UML для начинающих [Электронный ресурс]. – URL: https://drive.google.com/drive/folders/1ckgoCtts8h3GswQygRxCb6VewGCzcWoM (дата обращения: 15.12.2019)

3. GitHub [Электронный ресурс]. – URL: https://github.com (дата обращения: 12.10.2019)

# 