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)

РАЗРАБОТКА ПРИЛОЖЕНИЯ NoteApp

Календарный план и смета к лабораторной работе по дисциплине

«Системное программное обеспечение»

Выполнил:

Студент гр. 539-1

\_\_\_\_\_\_\_\_\_\_ Пантелеев В.Д.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Проверил:

доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_ Горяинов А.Е.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Томск 2022

Содержание

[1 Назначение приложения 3](#_Toc93962680)

[3 Стек технологий и системные требования 5](#_Toc93962681)

[4 Пользовательский интерфейс 5](#_Toc93962682)

[5 Диаграммы пакетов приложения 7](#_Toc93962683)

[6 Диаграмма классов приложения 7](#_Toc93962684)

[7 Тестирование приложения 7](#_Toc93962685)

[8 Сборка установщика 10](#_Toc93962686)

[9 Описание модели ветвления 11](#_Toc93962687)

[Список литературы 13](#_Toc93962688)

[Приложение А 14](#_Toc93962689)

# 1 Назначение приложения

Пользовательское приложение NoteApp, предназначено для ведения персональных записей и 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 Группы пользователей и их функциональные возможности в приложении**

Данное приложение разработано для пользователей ПК.

После запуска приложения перед пользователем появляется главное окно. Двухколоночная верстка главного окна содержит список всех созданных заметок в левой панели и отображает текущую выбранную заметку в правой панели. В списке заметок показаны названия заметок, в один момент времени может быть выбрана только одна заметка.

На панели со списком заметок внизу располагаются три кнопки в виде пиктограмм: Add Note («Создать новую заметку»), Edit Note («Редактировать текущую заметку»), Delete Note («Удалить текущую заметку»).  
При выборе заметки в списке, выбранная заметка отображается в правой панели. Главное окно не позволяет редактировать содержимое заметки, только просматривать.  
 При нажатии на кнопку Add Note и Edit Note появляется окно создания/редактирования заметки в диалоговом режиме (рис. 72). Для новой заметки окно изначально не заполнено (Установлены лишь название заметки по умолчанию, дата создания и дата редактирования). Для редактирования уже существующей заметки все поля должны быть предзаполнены данными текущей заметки.

При нажатии на кнопку OK окно создания заметки закрывается, в список заметок главного окна добавляется новая заметка. При редактировании текущей заметки, нажатие на кнопку OK должно обновить название заметки в списке заметок (если название текущей заметки было изменено), и обновить отображаемую заметку в правой панели приложения.  
При нажатии кнопки Cancel создание/редактирование заметки отменяется (новая заметка не добавляется, исходная заметка остается без изменений). В случае ввода пользователем некорректных данных (нарушение допустимой длины названия заметки), данная ситуация должна быть обработана соответствующим образом.   
 При нажатии на кнопку Remove Note главного окна текущая запись удаляется. Перед удалением должно появиться окно с запросом на разрешение записи: «Do you really want to remove this note: <Название текущей записи>». При нажатии на кнопку OK происходит удаление, при нажатии на кнопку Cancel удаление отменяется.

**Меню главного окна** содержит следующие пункты:  
1) *File*:  
• *Exit*   
2) *Edit*:  
• *Add Note*   
• *Edit Note*   
• *Remove Note*   
3) *Help*• *About*

# 3 Стек технологий и системные требования

Язык программирования C#, версия .NET Framework – 4.7.2. Для тестирования приложения использовалась библиотека NUnit. Для выполнения сериализации и десериализации использовалась библиотека Newtonsoft.Json.

Системные требования аналогичны требованиям ОС Windows 10.

# 4 Пользовательский интерфейс

На рисунке 4.1 показан снимок главного окна программы.

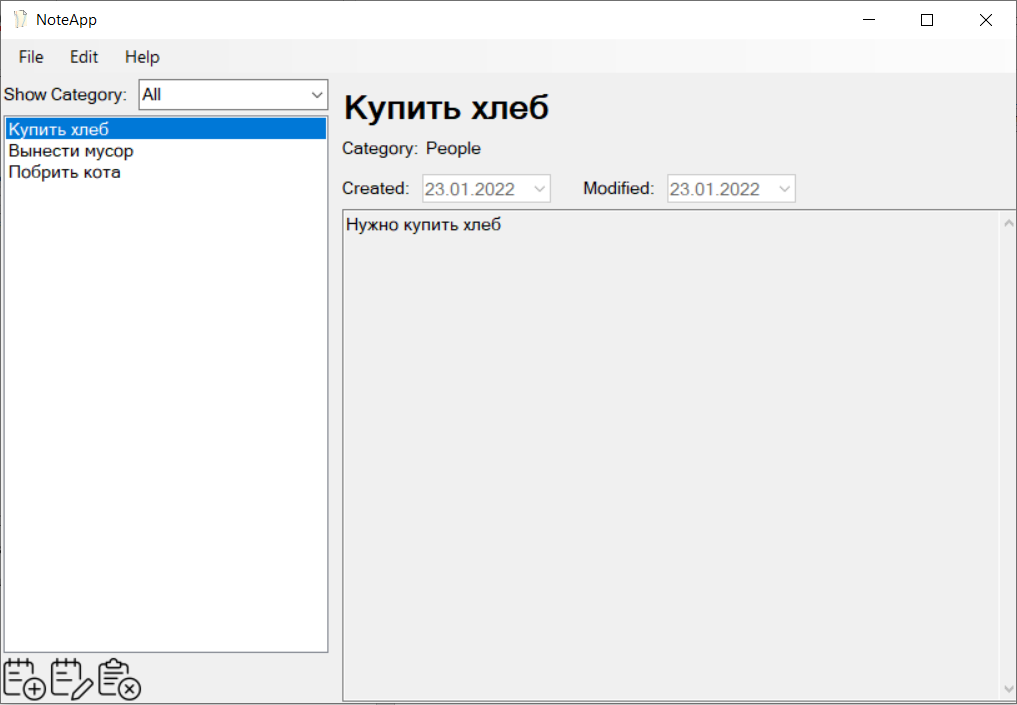


Рисунок 4.1 – Главное окно программы

На рисунке 4.2 показан снимок окна редактирования и добавления контакта.

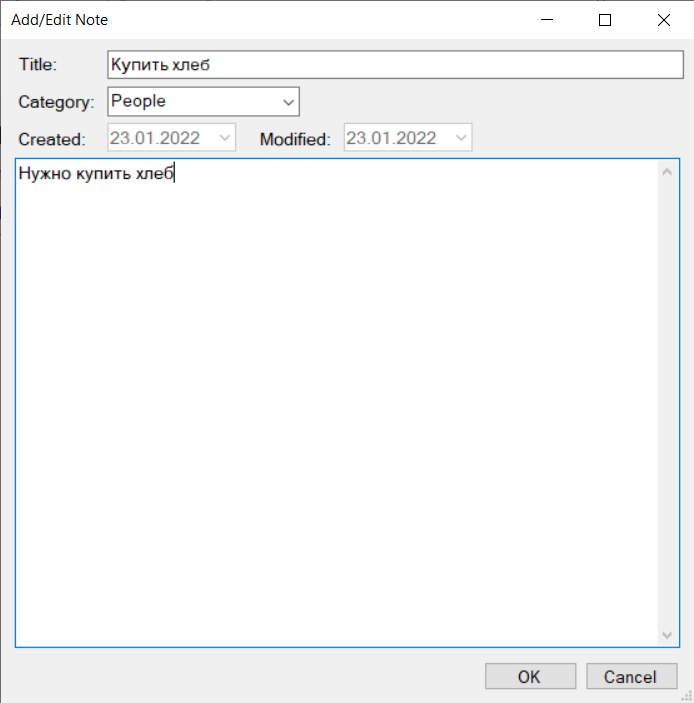


Рисунок 4.2 – Окно редактирования контакта

На рисунке 4.2 – показан снимок окна About.

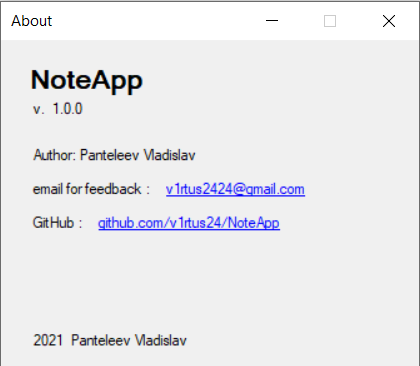


Рисунок 4.3 – Окно About

# 5 Диаграммы пакетов приложения

Диаграмма пакетов приложения представлена на рисунке 5.1.

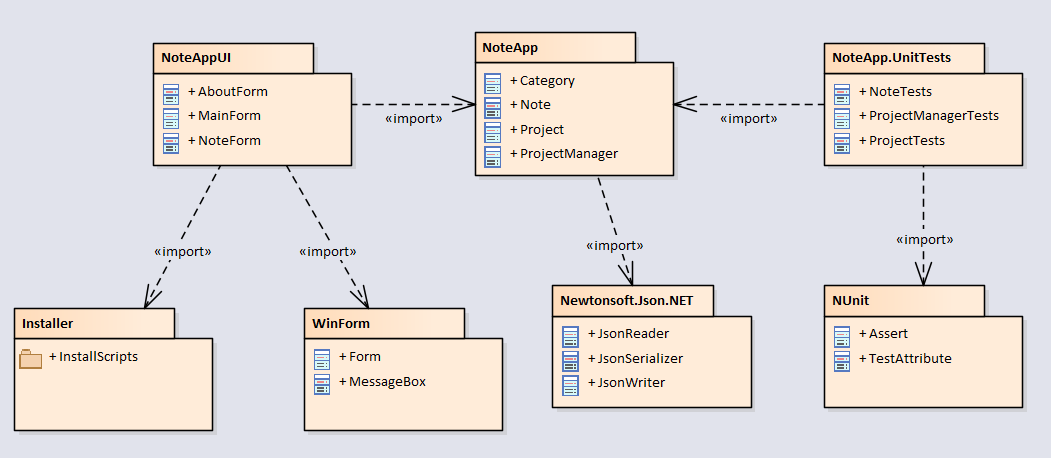


Рисунок 5.1 – Диаграмма пакетов приложения

# 6 Диаграмма классов приложения

Диаграмма классов приложения представлена в Приложении А

# 7 Тестирование приложения

Тестирование приложения было проведено как вручную, так и автоматически.

Основным типом тестирования было юнит-тестирование. С помощью юнит-тестов в разработанном приложении протестирован проект бизнес-логики. Для этого разработаны тесты, полностью покрывающие исходный код данного проекта.

Полный список юнит-тестов представлен на рисунке 7.1.

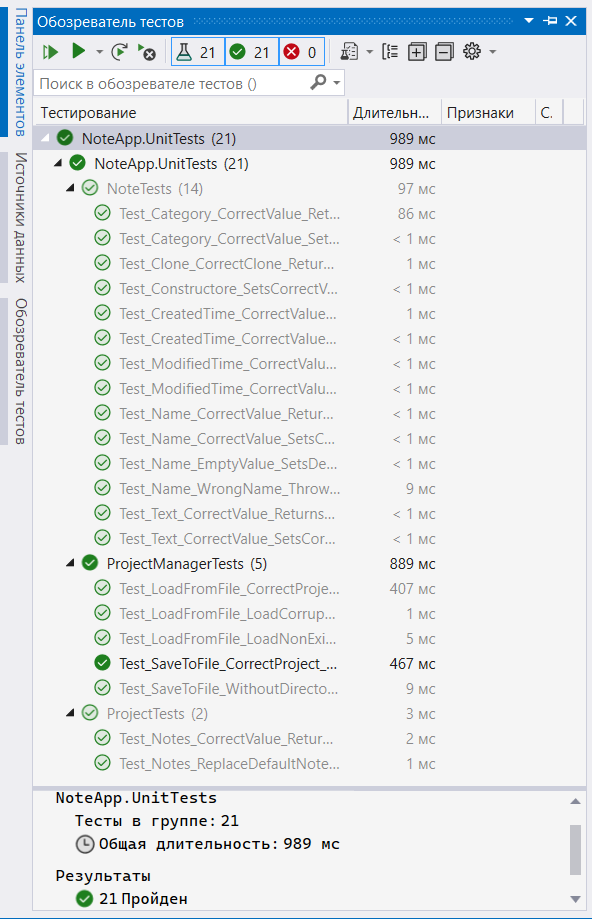


Рисунок 7.1 - Список юнит-тестов.

Тесты написаны с использованием библиотеки NUnit, предназначенной для написания юнит-тестов для платформы .NET. Написанные тесты проверяют как позитивные, так и негативные тестовые случаи.

Тестирование классов пользовательского интерфейса проводилось вручную.

В качестве общего системного теста, проверяющего работоспособность всего приложения, было проведено приемочное тестирование. Для приемочного тестирования необходимо выполнить следующую последовательность действий:

1. Установить приложение на компьютер с помощью собранного установочного пакета.
2. Запустить приложение. Окно программы должно быть пустым – в приложении не должно быть заметок.
3. Создать в приложении три заметки разных категорий.
4. Переключиться между заметками, чтобы убедиться, что смена текущей заметки происходит корректно.
5. Переключить отображаемую категорию заметок – в списке заметок должны остаться только заметки целевой категории. Снова отобразить все категории заметок – список заметок должен восстановиться.
6. Выбрать вторую заметку и нажать кнопку редактирования. Должно открыться окно редактирования заметки.
7. Ввести название заметки более 50 символов. Элемент управления с названием заметки должен указать на некорректное значение.
8. Ввести название заметки менее 50 символов. Элемент управления с названием заметки должен стать корректным.
9. Поменять текст заметки. Нажать «OK». Отредактированная заметка должна подняться в списке заметок на первую позицию, время изменения заметки должно поменяться, отображаемый текст заметки также должен измениться.
10. Выбрать вторую заметку и нажать кнопку редактирования. Должно открыться окно редактирования. Изменить название заметки, ее текст и категорию. Нажать «Cancel». Исходная заметка должна остаться без изменений.
11. Удалить третью заметку.
12. Закрыть приложение. Должно произойти сохранение заметок в целевой файл.
13. Запустить приложение. В программе должны восстановиться заметки, созданные в предыдущую сессию.

# 8 Сборка установщика

Сборка проекта осуществляется в автоматическом режиме. Для того, чтобы создания сценарий сборки инсталлятора, использовалась программа InnoSetup. Сборка установочного файла осуществляется автоматически во время сборки программы. Установщик располагается в директории с исходным проектом. Исходный сценарий для автоматизации сборки:

rd /S /Q "$(SolutionDir)\Installer\InstallScripts\Release"

rd /S /Q "$(SolutionDir)\Installer\InstallScripts\Installers"md"$(SolutionDir)Installer\InstallScripts"

md "$(SolutionDir)\Installer\InstallScripts"

md "$(SolutionDir)\Installer\InstallScripts\Release"

md "$(SolutionDir)\Installer\InstallScripts\Installers"

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

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

где:

1. «md» -создание папки;
2. «rd /S /Q» - удаление папки со всем содержимым;
3. «xcopy» - копирование из одной папки в другую.

Сценарий установщика:

#define MyAppName "NoteApp"

#define MyAppVersion "1.0"

#define MyAppPublisher "Panteleev Vlasislav"

#define MyAppURL "https://github.com/v1rtus24"

#define MyAppExeName "NoteAppUI.exe"

#define MyAppAssocName MyAppName + " File"

#define MyAppAssocExt ".myp"

#define MyAppAssocKey StringChange(MyAppAssocName, " ", "") + MyAppAssocExt

#define AppIconName "icon.ico"

[Setup]

; NOTE: The value of AppId uniquely identifies this application. Do not use the same AppId value in installers for other applications.

; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)

AppId={{AB31DE89-37CC-4547-859F-60B7597B7ACD}

AppName={#MyAppName}

AppVersion={#MyAppVersion}

;AppVerName={#MyAppName} {#MyAppVersion}

AppPublisher={#MyAppPublisher}

AppPublisherURL={#MyAppURL}

AppSupportURL={#MyAppURL}

AppUpdatesURL={#MyAppURL}

DefaultDirName={pf}\{#MyAppName}

ChangesAssociations=yes

DisableProgramGroupPage=yes

; Uncomment the following line to run in non administrative install mode (install for current user only.)

;PrivilegesRequired=lowest

OutputDir="Installers"

OutputBaseFilename=NoteAppSetup

SetupIconFile="..\..\NoteAppUI\{#AppIconName}"

Compression=lzma

SolidCompression=yes

WizardStyle=modern

[Languages]

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

Source: "..\..\NoteAppUI\{#AppIconName}"; DestDir: "{app}"; Flags: ignoreversion

; NOTE: Don't use "Flags: ignoreversion" on any shared system files ValueData: ""

[Icons]

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

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

[Run]

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

# 9 Описание модели ветвления

Во время разработки программы для обеспечения версионного контроля использовался сервис GitHub.

Ссылка на репозиторий: https://github.com/v1rtus24/NoteApp

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

На рисунке 9.1 – показана история фиксаций ветки develop.

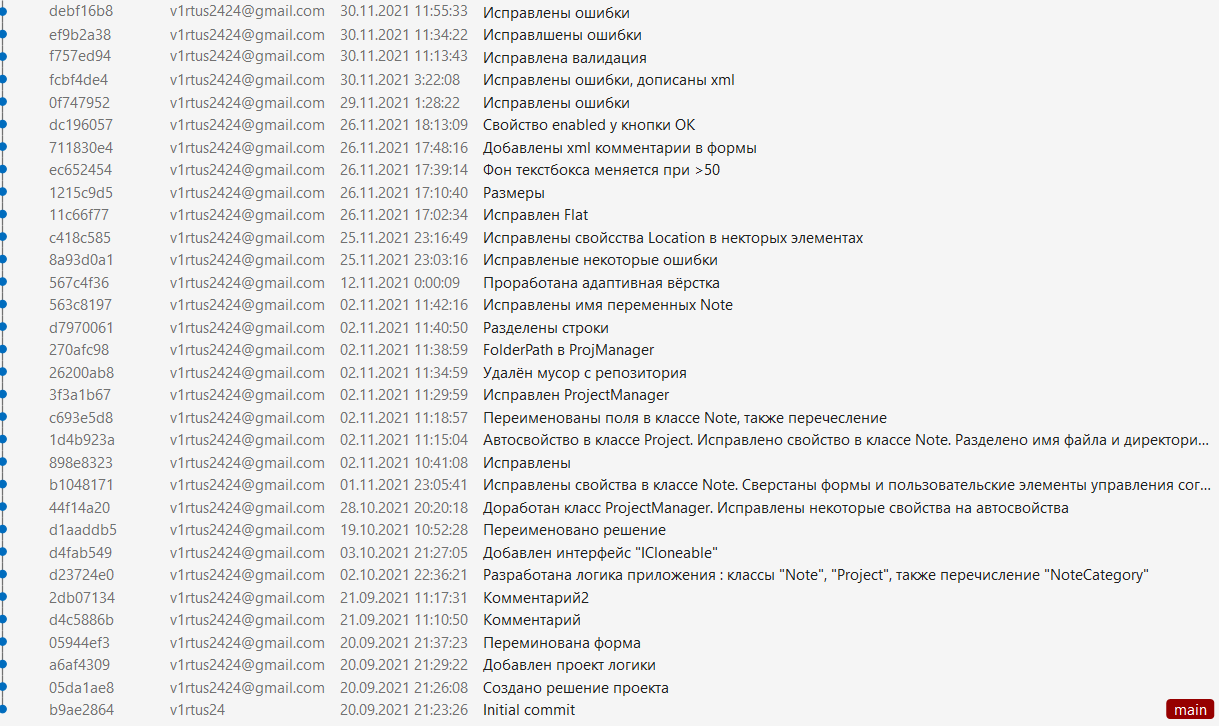


Рисунок 9.1.1 – История фиксаций ветки develop

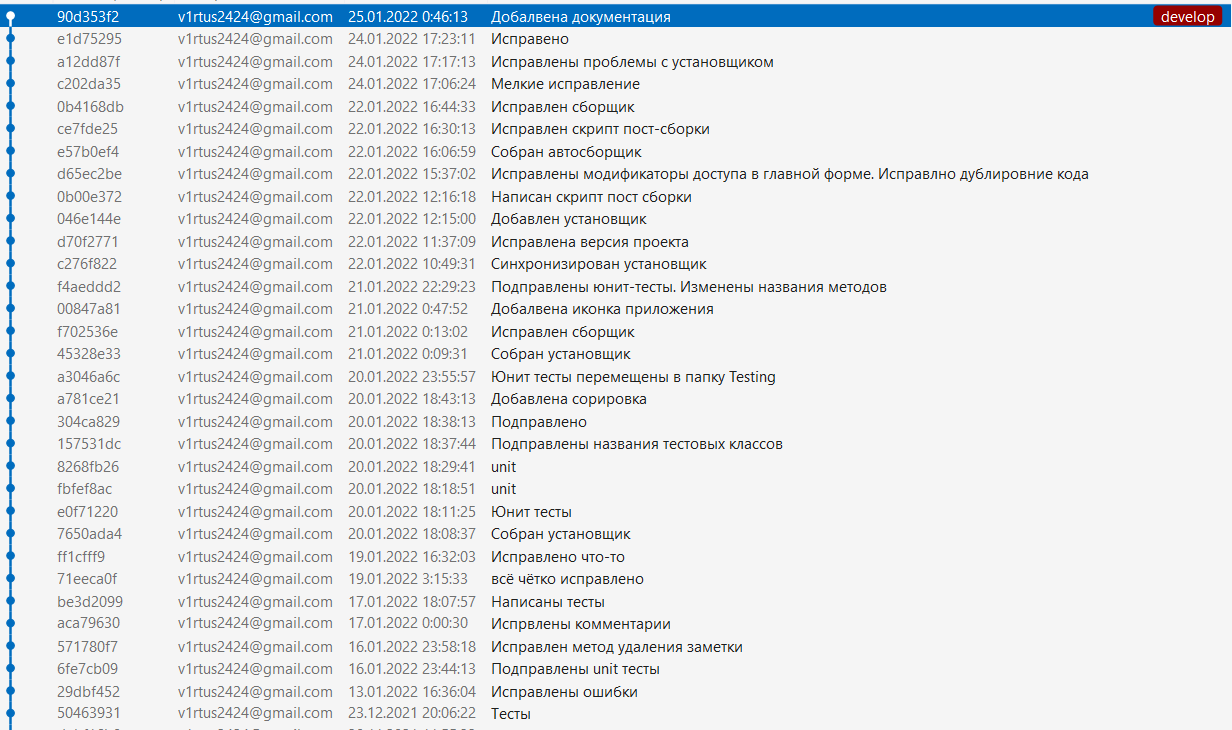


Рисунок 9.1.2 – История фиксаций ветки develop

# Список литературы

Команды Windows [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/windows-server/administration/windows-commands/windows-commands>

Введение в C#: язык C# и платформа .NET. [Электронный ресурс]. / Metanit.com: сайт о программировании. — URL: https://metanit.com/sharp/tutorial/1.1.php

Руководство по программированию в Windows Forms [Электронный ресурс]. matanit.com: сайт о программировании. — URL: https://metanit.com/sharp/windowsforms/

GitHub [Электронный ресурс]. – URL: <https://github.com>

# Приложение А

