Министерство науки и высшего образования 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)

РАЗРАБОТКА ПРИЛОЖЕНИЯ CONTACTSAPP

Пояснительная записка к лабораторной работе

по дисциплине

«Новые технологии в программировании»

Выполнила:

студентка гр. 587-2

\_\_\_\_\_\_\_\_\_\_ Маркина А.Р.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021г.

Проверил:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_ Горяинов А.Е.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Содержание

[1 Назначение приложения 3](#_Toc62620170)

[2 Группы пользователей и их функциональные возможности в приложении 3](#_Toc62620171)

[3 Стек технологий и системные требования 3](#_Toc62620172)

[4 Пользовательский интерфейс 4](#_Toc62620173)

[5 Диаграммы пакетов приложения 7](#_Toc62620174)

[6 Диаграмма классов приложения 8](#_Toc62620175)

[7 Тестирование приложения 8](#_Toc62620176)

[8 Сборка установщика 11](#_Toc62620177)

[9 Описание модели ветвления 13](#_Toc62620178)

# 1 Назначение приложения

Пользовательское приложение ContactsApp предназначено для ведения и хранения контактов.

# 2 Группы пользователей и их функциональные возможности в приложении

У данного приложения один тип пользователей: пользователь ПК.

Приложение должно:

1. Обеспечивать стабильную работу приложения при порядке 200 контактов.

2. Обеспечивать поиск, навигацию по созданным контактам по фамилии и имени.

3. Обеспечивать возможность создания и редактирования контактов.

4. Сохранять и восстанавливать контакты между сессиями приложения.

5. Выполнять промежуточные сохранения контактов на жесткий диск, чтобы предотвратить потерю данных при аварийном завершении приложения.

Приложение-референс: контакты мобильного телефона.

# 3 Стек технологий и системные требования

Для реализации проекта был выбран язык программирования C# на платформе .NET Framework 4.6.1 и среда разработки Visual Studio 2019. Системные требования к приложению ограничены требованиями к .NET Framework.

Для выполнения сериализации и десериализации использовалась библиотека Newtonsoft.Json версии 12.0.3.

Для проведения юнит-тестирования приложения использовалась библиотека NUnit версии 3.12.0.

Для создания сценария сборки установочного пакета и компиляции установщика использовалась программа Inno Setup версии 6.1.2.

# 4 Пользовательский интерфейс

После запуска приложения перед пользователем появляется главное окно (рис. 4.1). Двухколоночная верстка главного окна содержит список всех контактов в левой панели и отображает текущий выбранный контакт в правой панели. В списке контактов показаны фамилии контактов, в один момент времени может быть выбран только один контакт (далее – текущий контакт).

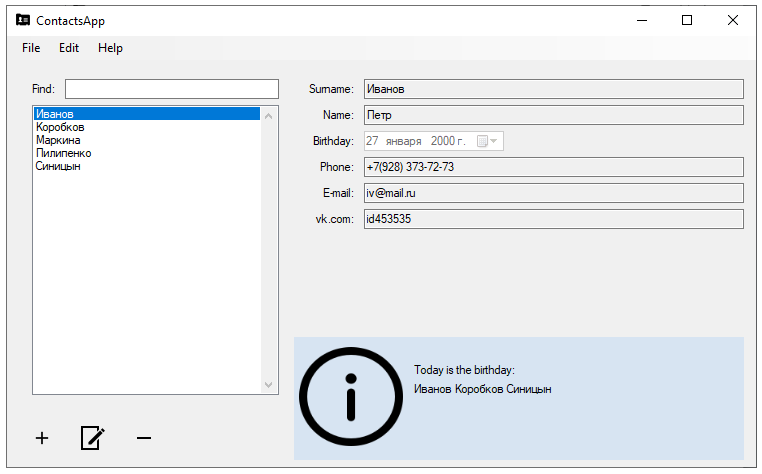


Рисунок 4.1 – Главное окно программы

На панели со списком контактов внизу располагаются три кнопки в виде пиктограмм: Add Contact («Создать новый контакт»), Edit Contact («Редактировать текущий контакт»), Remove Contact («Удалить текущий контакт»).

При выборе контакта в списке, выбранный контакт отображается в правой панели. Главное окно не позволяет редактировать содержимое контакта – только просмотр. Также в правой панели под текущим контактом отображается информационная панель с сегодняшними именинниками.

При нажатии на кнопку Add Contact и Edit Contact появляется окно создания/редактирования контакта в диалоговом режиме (рис. 4.2). Для нового контакта окно изначально не заполнено (установлена лишь дата рождения по умолчанию). Для редактирования уже существующего контакта все поля должны быть предзаполнены данными текущего контакта.

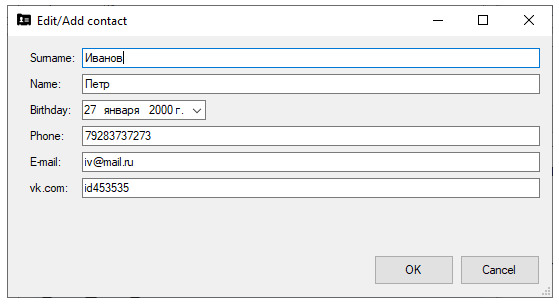


Рисунок 4.2 – Окно создания/редактирования контакта

При создании нового контакта нажатие на кнопку OK осуществляет закрытие окна создания, в список контактов главного окна добавляется новый контакт. При редактировании текущего контакта нажатие на кнопку OK должно обновить фамилию контакта в списке контактов (если фамилия текущего контакта была изменена/исправлена), и обновить отображаемый контакт в правой панели приложения. При нажатии кнопки Cancel создание/редактирование контакта отменяется (новый контакт не добавляется, исходный контакт остается без изменений).

В случае ввода пользователем некорректных данных (нарушение допустимой длины фамилии, имени, указание невозможной даты рождения или некорректного номера телефона), данная ситуация должна быть обработана соответствующим образом.

При нажатии на кнопку Remove Contact главного окна текущий контакт удаляется. Перед удалением должно появиться окно с запросом на разрешение удаления записи: «Do you really want to remove this contact?». При нажатии на кнопку OK происходит удаление, при нажатии на кнопку Cancel удаление отменяется.

Меню главного окна содержит следующие пункты:

* + File:
  + Exit (Выйти из приложения – Alt+F4)
* Edit:
  + Add Contact (Создать новый контакт)
  + Edit Contact (Редактировать текущий контакт)
  + Remove Contact (Удалить текущий контакт)
* Help
* About (Вызвать окно «О программе» – F1) (рис. 4.3)

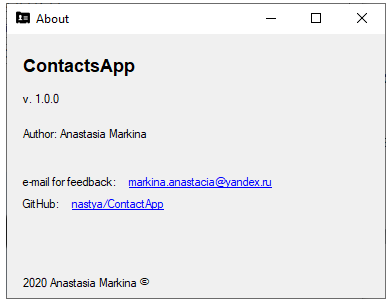
****

Рисунок 4.3 – Окно содержащее информацию о программе

Таким образом, в программе должно быть реализовано три окна:

1) Главное окно.

2) Окно создания/редактирования контакта.

3) Окно «About».

Верстка главного окна и окна создания/редактирования контакта должна быть адаптивной. Окно «About» имеет фиксированный размер.

Загрузка контактов осуществляется при запуске программы до вывода главного окна пользователю, сохранение контактов в файл должно выполняться следующих случаях:

а) создание нового контакта;

б) удаление контакта;

в) закрытие приложения.

# 5 Диаграммы пакетов приложения

Диаграмма пакетов приложения представлена на рисунке 5.1.

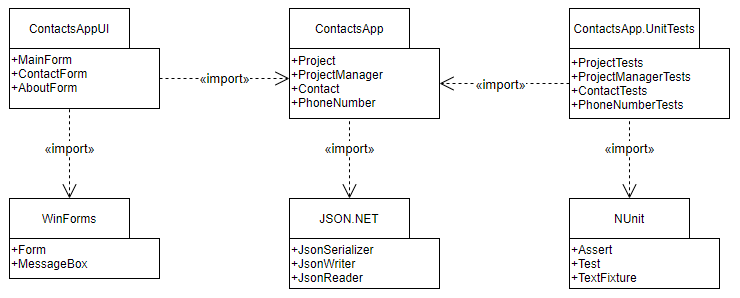


Рисунок 5.1 – Диаграмма пакетов приложения.

# 6 Диаграмма классов приложения

Диаграмма классов приложения представлена на рисунке 6.1.

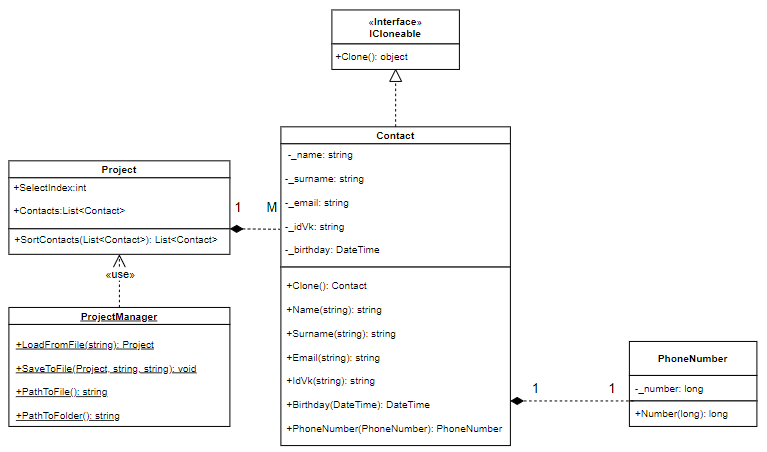


Рисунок 6.1 – Диаграмма классов приложения.

# 7 Тестирование приложения

Тестирование приложения проводилось при помощи юнит-тестов.

Для написания юнит-тестов использовалась библиотека NUnit. Полный список юнит-тестов представлен на рисунке 7.1. Покрытие тестов представлено на рисунке 7.2.

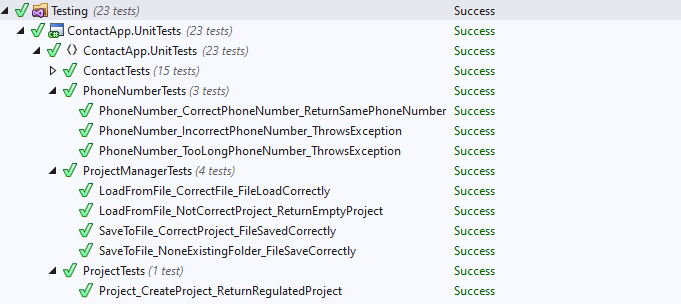


Рисунок 7.1 – Список юнит-тестов

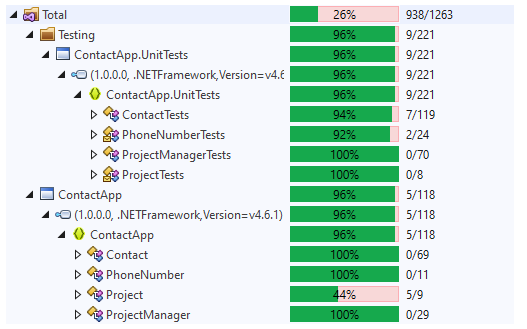


Рисунок 7.2 – Покрытие юнит-тестов

Помимо юнит-тестов также было приведено приёмочное тестирование приложения. Для успешного прохождения приёмочного тестирования, необходимо:

1. Установить приложение на компьютер с помощью собранного установочного пакета.

2. Запустить приложение. Окно программы должно быть пустым – приложение не содержит контактов.

3. Создать три контакта в приложении.

4. Переключиться между контактами, показав, что смена текущего контакта происходит корректно.

5. Воспользоваться навигацией по контактам – ввести в строку поиска имя или фамилию одного из контактов и убедиться, что приложение работает верно.

6. Выбрать один из контактов и нажать кнопку редактирования. Должно открыться окно редактирования контакта.

7. Ввести в поле Name или Surname более 50 символов. Элемент управления с названием поля должен указать на некорректное значение.

8. Ввести в поле Name или Surname менее 50 символов. Элемент управления с названием поля должен стать корректным.

9. Отредактировать один из контактов. Нажать «OK». Отредактированный контакт должен измениться и корректно отобразиться в главном окне программы.

10. Выбрать другой контакт и попытаться изменить его значения на некорректные. Нажать «ОК». Программа должна сообщить, что некоторые поля введены неверно.

11. Удалить один из контактов.

12. Закрыть приложение. Должно произойти сохранение контактов в целевой файл.

13. Запустить приложение. В программе должны восстановиться ранее созданные контакты.

# 8 Сборка установщика

Для сборки установщика приложения используется программа Inno Setup. С ее помощью создается установочный сценарий, который создает установщик. Он располагается в директории с исходным проектом.

Исходный код сценария для автоматизации сборки представлен ниже. В нем указывается название установочного пакета, название программы, имя автора, пути установки и др. Необходимо указать файлы, требуемые для работы приложения: \*exe и \*dll.

Скрипт, выполняющийся перед сборкой ContactAppUI:

rd /s /q "$(SolutionDir)Installer\InstallScripts\Release"

rd /s /q "$(SolutionDir)Installer\InstallScripts\Installer"

Скрипт, выполняющийся после сборки ContactAppUI:

md "$(SolutionDir)Installer\InstallScripts"

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

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

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

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

Скрипт, выполняющийся перед сборкой Installer:

"$(SolutionDir)packages\Tools.InnoSetup.6.1.2\tools\ISCC.exe" "$(SolutionDir)Installer\InstallScripts\Installer.iss"

где rd /s /q – удаление каталога по указанному пути,

md – создание директория,

xcopy – копирование файлов.

Код скрипта установщика:

; Script generated by the Inno Setup Script Wizard.

; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

#define MyAppName "ContactApp"

#define MyAppVersion "1.0.1"

#define MyAppPublisher "Markina Anastasia"

#define MyAppExeName "ContactAppUI.exe"

[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={{670B784F-D7B9-4518-B85A-96232E27E910}

AppName={#MyAppName}

AppVersion={#MyAppVersion}

;AppVerName={#MyAppName} {#MyAppVersion}

AppPublisher={#MyAppPublisher}

DefaultDirName={autopf}\{#MyAppName}

DisableProgramGroupPage=yes

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

;PrivilegesRequired=lowest

OutputBaseFilename=ContactAppSetup

OutputDir="Installer"

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

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

[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

# 9 Описание модели ветвления

Во время разработки программы для обеспечения версионного контроля использовалась распределенная система контроля версий Git в совокупности с сервисом GitHub. Ссылка на репозиторий – https://github.com/Nastt/ContactApp.

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.

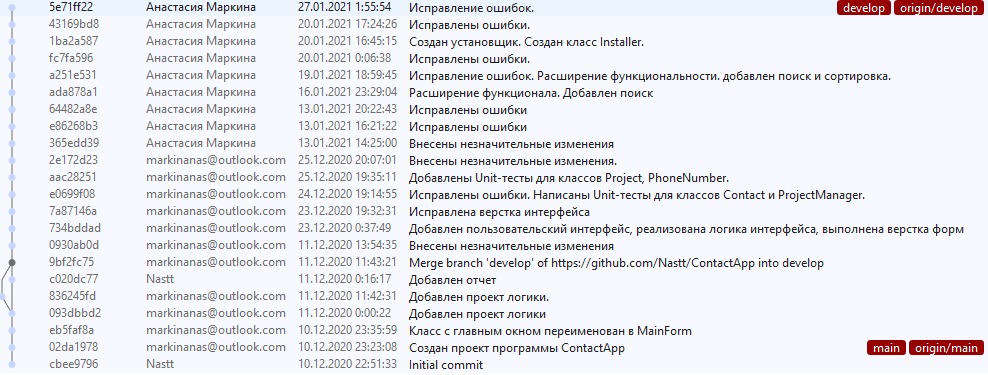


Рисунок 9.1 – История фиксаций ветки develop