

Министерство образования и науки 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)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К практической работе «Контакты ContactsApp»

Студент гр. 588-2:

Швоев В.С.

Проверил:

Горяинов А.Е.

Дата: 18.07.2020

Содержание

Назначение приложения.....	3
Стек технологий разработки.....	3
Поведенческие диаграммы.....	4
Пользовательский интерфейс	4
Диаграммы пакетов приложения.....	8
Диаграммы классов приложения.....	9
Описание тестирования приложения.....	10
Описание сборки установщика.....	10
Описание модели ветвления в репозитории проекта	12
Приложения	13

Назначение приложения

Пользовательское приложение ContactsApp, предназначено для ведения и хранения контактов. Приложение должно:

- 1) Обеспечивать стабильную работу приложения при порядке 200 контактов.
- 2) Обеспечивать поиск, навигацию по созданным контактам по фамилии и имени.
- 3) Предоставить инструменты для просмотра и редактирования контактов.
- 4) Сохранять и восстанавливать контакты между сессиями приложения.
- 5) Выполнять промежуточные сохранения контактов на машине пользователя на случай аварийного завершения программы, отключения компьютера и т.д. — для защиты от потери данных.

Приложение-референс: контакты мобильного телефона

Стек технологий разработки.

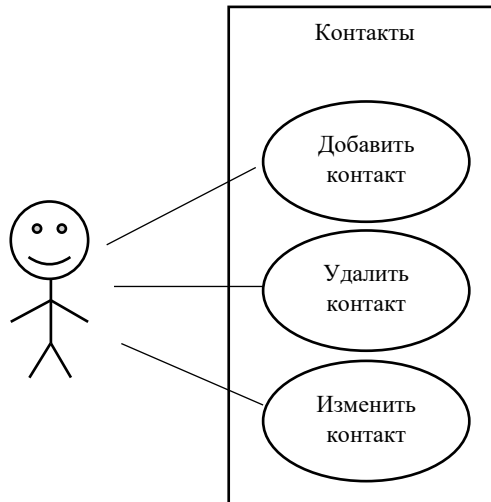
Для разработки приложения был выбран язык программирования C#

Для тестирования приложения была выбрана библиотека NUnit версии 3.12.0.

Выбрана версия .NET Framework 4.7.2.

Для работы со строками и файлами была выбрана библиотека JSON.NET версии 12.0.3.

Поведенческие диаграммы.



Пользовательский интерфейс

После запуска приложения перед пользователем появляется главное окно (рис. 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).

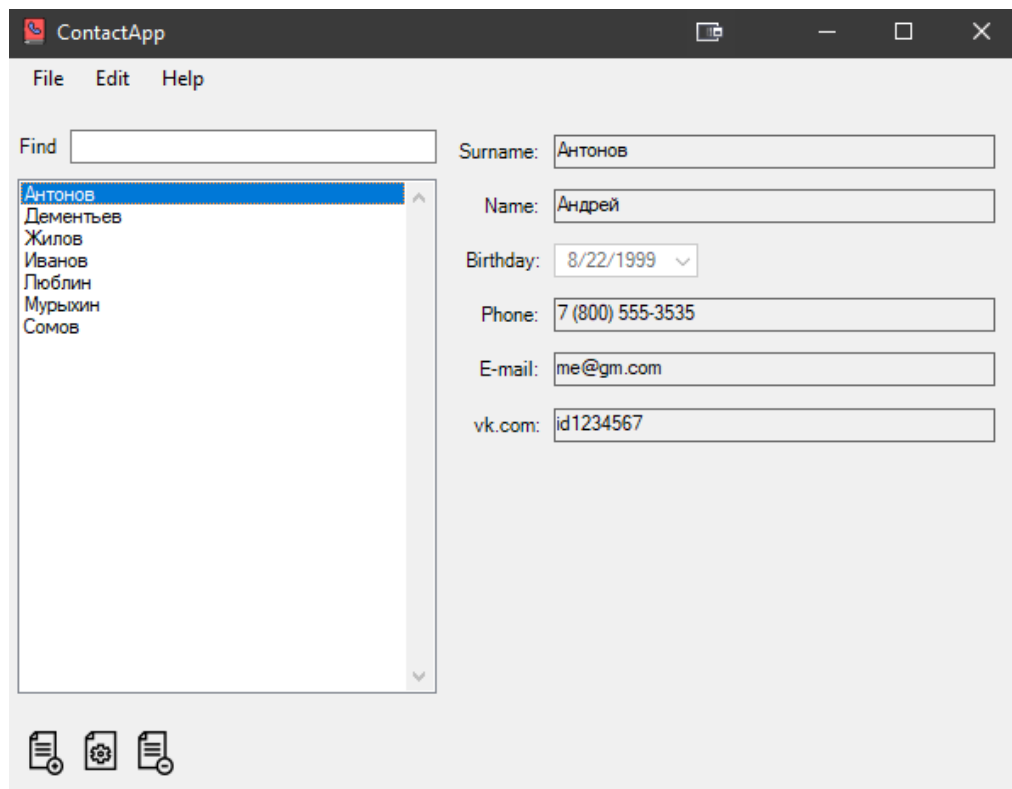


Рисунок 1 – Макет главного окна приложения ContactsApp

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 (см. рис. 1).

При нажатии на кнопку Add Contact и Edit Contact появляется окно создания/редактирования контакта в диалоговом режиме (рис. 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.

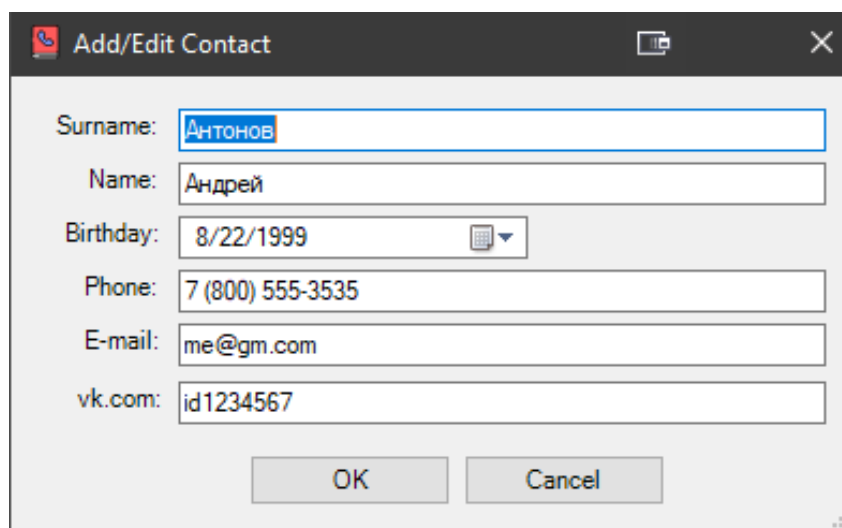


Рисунок 2 – Макет окна редактирования контакта в приложении ContactsApp

При нажатии на кнопку ОК окно создания контакта закрывается, в список контактов главного окна добавляется новый контакт. При редактировании текущей контакта, нажатие на кнопку ОК 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). Реализация передачи данных между двумя окнами см. п. п. 2.3.5 «Передача данных между формами».

В случае ввода пользователем некорректных данных (нарушение допустимой длины фамилии, имени, указание невозможной даты рождения или неправильного номера теле-фона), данная ситуация должна быть обработана соответствующим образом (см. п. 2.3.2 «Получение данных со стандартных элементов управления»).

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

Меню главного окна содержит следующие пункты:

1. File:
 - 1.1.Exit (Выйти из приложения – Alt+F4)
2. Edit:
 - 2.1.Add Contact (Создать новый контакт)
 - 2.2.Edit Contact (Редактировать текущий контакт)
 - 2.3.Remove Contact (Удалить текущий контакт)
3. Help:
 - 3.1.About (Вызвать окно «О программе» - F1) (см. рис. 3)

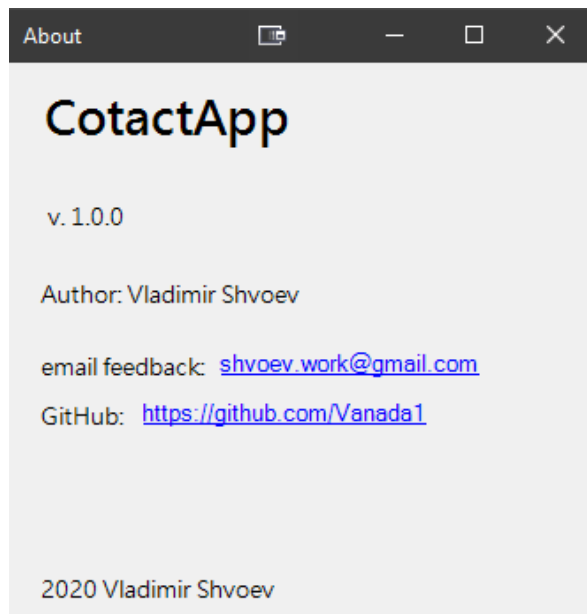


Рисунок 3 – Макет окна «About» приложения ContactsApp

Таким образом, в программе должно быть реализовано три окна:

- Главное окно.
- Окно создания/редактирования контакта.
- Окно «About».

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

Создание дополнительных элементов управления для уменьшения дублирования кода остается на усмотрение разработчика или руководителя.

Загрузка контактов осуществляется при запуске программы до вывода главного окна пользователю, сохранение контактов в файл должно выполняться в случаях: а) создания нового контакта; б) удаления контакта; в) закрытии приложения.

Диаграммы пакетов приложения

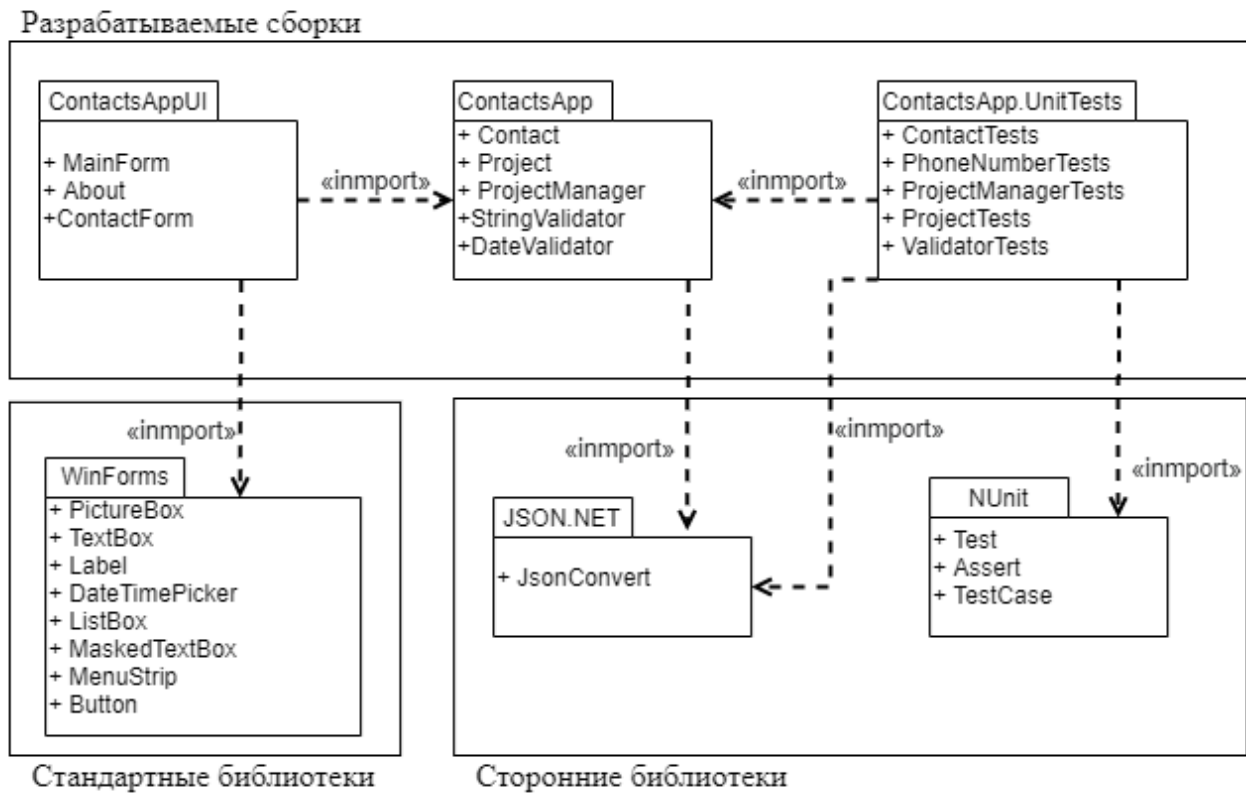


Рисунок 4 – Диаграмма пакетов приложения

Диаграммы классов приложения

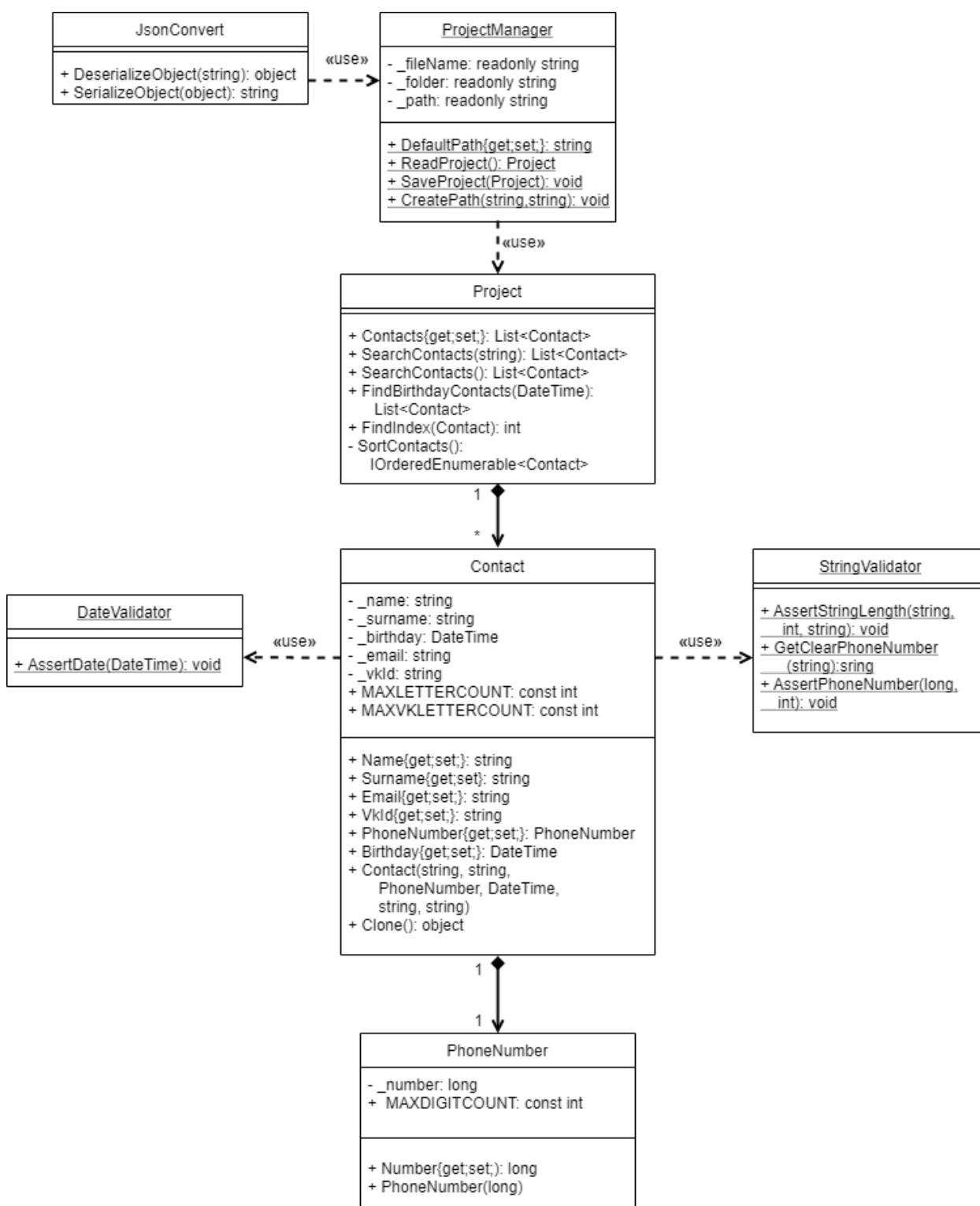


Рисунок 5 – Диаграмма классов приложения

Описание тестирования приложения

Вручную тестируются методы и классы интерфейсов.

Методы бизнес-логики тестируются с помощью Unit-тестов. Тестировка делится на 5 модулей: ContactTests, PhoneNumberTests, ProjectManagerTests, ProjectTests, ValidatorTests. Все эти модули тестируют свойства и методы классов в соответствии с названием.

Описание сборки установщика

Сборка проводилась с помощью автоматического сборщика Inno Setup. Также был создан дополнительный проект для компиляции .iss файла в установщик. Код .iss файла:

```
#define Name "ContactsApp"
#define Version "0.0.1"
#define Publisher "Vladimir Shvov"
#define URL "https://github.com/Vanada1"
#define ExeName "ContactsApp"

[Setup]
AppId = {{AC2F2855-FF95-4C0D-81EB-46E4052298FE}}
AppName={#Name}
AppVersion={#Version}
AppPublisher={#Publisher}
AppPublisherURL={#URL}
AppSupportURL={#URL}
AppUpdatesURL={#URL}
DefaultDirName = {pf}\{#Name}
DefaultGroupName = {#Name}
OutputDir=..\setup
OutputBaseFileName=setup
SetupIconFile=phonebook.ico
```

```

Compression=lzma
SolidCompression=yes
[Languages]
Name: "english"; MessagesFile: "compiler:Default.isl"
Name: "russian"; MessagesFile: "compiler:Languages\Russian.isl"
[Tasks]
; Создание иконки на рабочем столе
Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}";
GroupDescription: "{cm:AdditionalIcons}"; Flags: unchecked
[Files]
; Исполняемый файл
Source: "..\Release\ContactsApp.exe"; DestDir: "{app}"; Flags:
ignoreversion
; Прилагающиеся ресурсы
Source: "..\Release\*"; DestDir: "{app}"; Flags: ignoreversion
recursesubdirs createallsubdirs
[Icons]
Name: "{group}\{#Name}"; Filename: "{app}\{#ExeName}"
Name: "{commondesktop}\{#Name}"; Filename: "{app}\{#ExeName}";
Tasks: desktopicon

```

Скрипт для создания установщика:

```

md $(SolutionDir)InstallScripts\Release
md $(SolutionDir)InstallScripts\Installers
xcopy $(SolutionDir)ContactsAppUI\$(OutDir)*.dll
$(SolutionDir)InstallScripts\Release
xcopy $(SolutionDir)ContactsAppUI\$(OutDir)*.exe
$(SolutionDir)InstallScripts\Release
$(SolutionDir)packages\Tools.InnoSetup.6.0.5\tools\ISCC.exe
$(SolutionDir)InstallScripts\Installers\Installer.iss
rmdir /s /q $(SolutionDir)InstallScripts\Release

```

Описание модели ветвления в репозитории проекта

Главные ветви

Центральный репозиторий содержит две главные ветки, существующие всё время.

- master
- develop

Ветвь master создаётся при инициализации репозитория, что должно быть знакомо каждому пользователю Git. Параллельно ей также мы создаём ветку для разработки под названием develop.

Origin/master – главная ветка. Исходный код в ней должен находиться в состоянии production-ready в любой произвольный момент времени.

Ветвь origin/develop – главная ветвь для разработки. Хранящийся в ней код в любой момент времени должен содержать самые последние изданные изменения, необходимые для следующего релиза. Эту ветку также можно назвать «интеграционной». Она служит источником для сборки автоматических ночных билдов.

Когда исходный код в ветви разработки (develop) достигает стабильного состояния и готов к релизу, все изменения должны быть определённым способом влиты в главную ветвь (master) и помечены тегом с номером релиза. Ниже мы рассмотрим этот процесс в деталях.

Помимо главных ветвей master и develop, наша модель разработки содержит некоторое количество типов вспомогательных ветвей, которые используются для распараллеливания разработки между членами команды, для упрощения внедрения нового функционала (features), для подготовки релизов и для быстрого исправления проблем в производственной версии приложения. Эти ветви всегда имеют ограниченный срок жизни. Каждая из них в конечном итоге рано или поздно удаляется.

Используются следующие типы ветвей:

- Ветви функциональностей (Feature branches)
- Ветви релизов (Release branches)
- Ветви исправлений (Hotfix branches)

Приложения

Часовая смета проекта:

#	Задача	Сроки, ч
1	Макетирование	4
2	Проектирование архитектуры	6
3	Проектирование плана тестирования	6
4	Разработка валидаторов	3
5	Разработка класса PhoneNumber	1
6	Разработка класса Contact	3
7	Разработка класса Project	2
8	Разработка класса ProjectManager	2
9	Разработка класса MainForm	2
10	Разработка класса ContactForm	1
11	Разработка класса About	1
12	Разработка unit-тестов и тестирование (2 часа на каждый класс)	16
13	Код ревью	5
14	Сборка установщика	2
	Итого:	54