

# Контакты ContactsApp

## 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

## Пользовательский интерфейс

(лабораторная работа «Разработка пользовательского интерфейса»)

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).

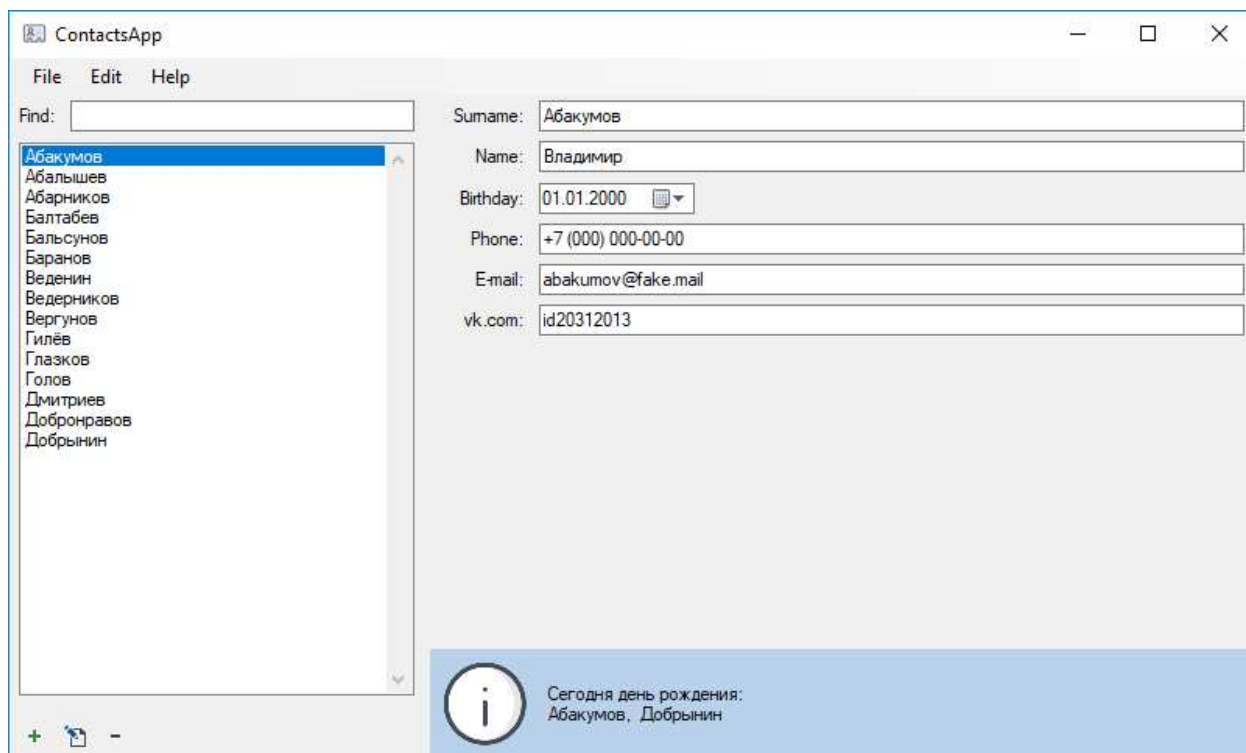


Рисунок 1 – Макет главного окна приложения ContactsApp

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.

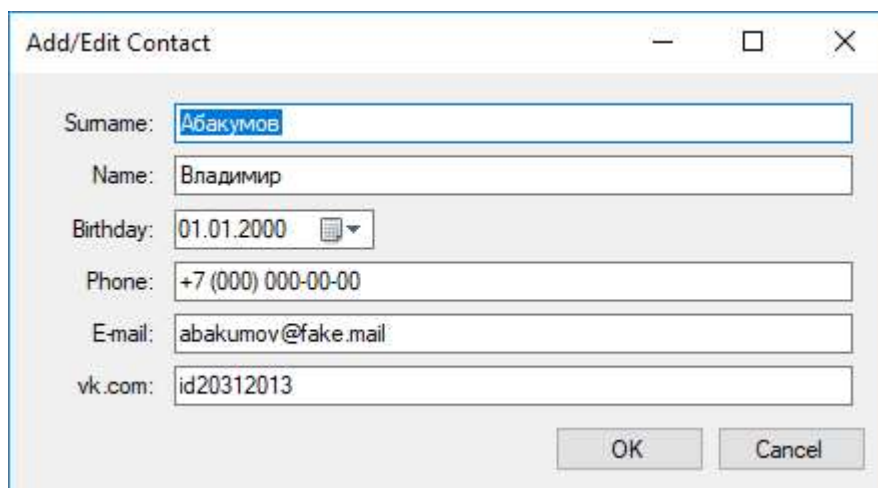


Рисунок 2 – Макет окна редактирования контакта в приложении ContactsApp

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).

В случае ввода пользователем некорректных данных (нарушение допустимой длины фамилии, имени, указание невозможной даты рождения или неправильного 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 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:

- *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) (см. рис. 3)

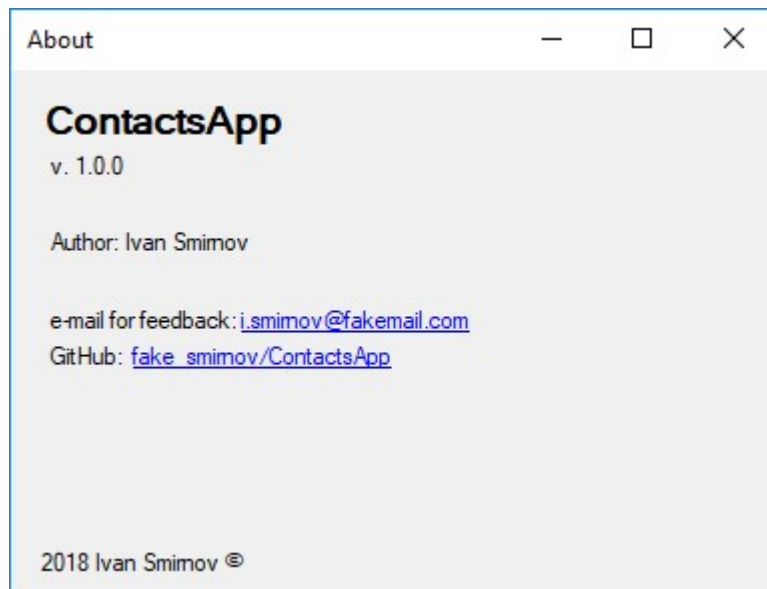


Рисунок 3 – Макет окна «About» приложения ContactsApp

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.

#### ***Допущения первой версии программы:***

*\*В первой версии программы контакты в списке отображаются в порядке их создания или ином произвольном порядке (сортировка по алфавиту в первой версии программы необязательна)*

*\*В первой версии программы необязательно реализовывать логику выпадающего списка с выбором категории заметок (поиск по имени/фамилии реализуется в дополнительной функциональности).*

*\*В первой версии программы может не реализовываться информационная панель с фамилиями именинников.*

*\*Все описанные допущения будут устранены в лабораторной работе «Расширение функциональности и релиз программы» (см. п. «Дополнительная функциональность»).*

## **Требования к бизнес-логике**

### **(лабораторная работа «Разработка бизнес-логики приложения»)**

Необходимо реализовать следующие типы данных:

- Класс «Номер телефона» с полем «Номер». Поле должно быть числовым и содержать ровно 11 цифр. Первая цифра должна быть '7' (только Российские телефонные номера)

- Класс «Контакт» с полями «Фамилия», «Имя», «Номер телефона», «Дата рождения», «e-mail», «ID ВКонтакте». Все поля доступны для изменений. Фамилия, имя и e-mail ограничены 50 символами каждое, ID Вконтакте ограничен 15 символами. Первая буква в фамилии и имени должна преобразовываться к верхнему регистру. Дата рождения не может быть более текущей даты и не может быть менее 1900 года. Допустимы контакты с одинаковыми фамилиями и именами. Реализует интерфейс ICloneable.

- Класс «Проект». Содержит список(или словарь) всех контактов, созданных в приложении.

- Класс «Менеджер проекта». Реализует метод для сохранения объекта «Проект» в файл и метод загрузки проекта из файла. Сохранение и загрузка осуществляются в один и тот же файл «...\My Documents\ContactsApp.notes», имя которого задано закрытой константой внутри класса. Формат данных – json, библиотека сериализации (преобразования данных в json-формат) – Newtonsoft JSON.NET.

\*Возможно, вы обратили внимание, что класс «Номер телефона» содержит только одно поле. Может возникнуть сомнения в целесообразности выделения класса с единственным полем. Однако очевидным расширением данного класса является разделение номера телефона на такие составляющие как «Код страны», «Код города», «Локальная часть номера», а также перечисления «Тип номера» («Мобильный», «Рабочий», «Домашний», «Основной» и т.д.). Таким образом, предполагая появление новых полей в классе, для простоты дальнейшего расширения программы разумно заранее вынести номер телефона в отдельный класс, даже если в первой версии программы предполагается только одно поле в классе.

## **Дополнительная функциональность**

### **(лабораторная работа «Расширение функциональности и релиз программы»)**

- В класс «Проект» добавить метод, который возвращает список контактов, отсортированных по фамилии. Добавить перегруженный метод сортировки контактов, принимающий на вход подстроку – метод возвращает отсортированный по алфавиту список контактов, фамилия или имя которых содержит указанную подстроку. В случае отсутствия контактов с данной подстрокой, метод возвращает пустой список.

- В левой панели главного окна обеспечить сортировку списка контактов по алфавиту.

- В левой панели главного окна над списком контактов реализовать текстовое поле для поиска контактов по подстроке. При вводе пользователем подстроки, в списке контактов должны отобразиться только контакты, фамилия и имя которых содержат данную подстроку. Если из текстового поля стереть подстроку, то в панели должны отобразиться все существующие контакты.

- В правой панели добавить информационную панель с именинниками. Для этого в классе «Проект» добавить метод, который принимает в качестве аргумента дату (экземпляр

стандартного класса DateTime), и возвращает список всех контактов, в которых дата рождения (день и месяц, но не год!) совпадает с датой во входном аргументе. Главное окно должно вызывать метод проекта с текущей датой, и отображать на панели список фамилий найденных именинников. Для упрощения реализации панель с именинниками должна инициализироваться только один раз в момент запуска приложения. Обновлять панель с именинниками при создании/удалении/редактировании контактов не требуется.

- Реализовать удаление текущего контакта по нажатию на клавишу Delete. Перед удалением должно появиться окно с запросом на разрешение удаление контакта.

### **Приёмочное тестирование**

Для приёмочного тестирования выполните следующую последовательность действий:

1. Установите приложение на компьютер с помощью собранного установочного пакета.
2. Запустите приложение. Окно программы должно быть пустым – в приложении не должно быть контактов.
3. Создайте три контакта в приложении с разными фамилиями.
4. Переключитесь между контактами, показав, что смена текущего контакта в правой панели происходит корректно.
5. Введите в поиск подстроку для поиска контактов – в списке контактов должны остаться только контакты, содержащие подстроку.
6. Введите в поиск подстроку, которой нет в фамилиях и именах контактов – список контактов должен быть пустым. Сотрите подстроку поиска – список контактов должен восстановиться.
7. Выберите любой контакт и нажмите кнопку редактирования. Должно открыться окно редактирования контакта.
8. Введите фамилию более 50 символов. Элемент управления с фамилией должен указать на некорректное значение.
9. Введите фамилию менее 50 символов. Элемент управления с фамилией должен стать корректным.
10. Покажите, что защита от некорректных значений также работает и для других полей.
11. Поменяйте фамилию контакта, отличную от исходной. Поменяйте номер телефона. Нажмите «ОК». Отредактированный контакт должен переместиться в списке контактов согласно алфавиту, отображаемые данные текущего контакта в правой панели также должны измениться.
12. Выберите любой контакт и нажмите кнопку редактирования. Должно открыться окно редактирования. Измените фамилию контакта, номер телефона и e-mail. Нажмите «Cancel». Исходный контакт должен остаться без изменений.
13. Удалите третий контакт.
14. Закройте приложение. Должно произойти сохранение контактов в целевой файл.
15. Запустите приложение. В программе должны восстановиться контакты, созданные в предыдущую сессию.
16. Дайте руководителю провести исследовательское тестирование программы.

**Варианты развития программы (по желанию студента, вне учебного курса дисциплины или для повышения оценки за пройденный курс)**

\* Создать элемент управления ContactControl для просмотра и редактирования заметок. Созданный элемент управления должен использоваться в правой панели главного окна и в окне создания/редактирования заметки. У элемента управления должно быть булево свойство ReadOnly, с помощью которого задается режим работы элемента управления (режим просмотра или режим редактирования контакта). Для удобства работы с элементом управления, элемент должен иметь свойство типа Contact (класс «Заметка»), через которое осуществляется передача данных.

\* Добавить поле «Аватар» в контакт, для которого пользователь сможет подгрузить фотографию пользователя в формате jpg и png. Загруженные фото контактов должны копироваться в локальную папку приложения (чтобы обеспечить сохранность фото в случае удаления фото из исходной папки), контакт хранит относительный путь к фото в локальной папке. Файл с фото перезаписывается в случае обновления фотографии.

\* Реализовать возможность добавления нефиксированного количества номеров телефонов с пометками «рабочий», «мобильный», «домашний», «основной» и т.д.. Реализовать возможность добавления нефиксированного количества социальных сетей.

\* Добавьте в программу функцию экспорта текущего контакта в формат vCard (\*.vcf – описание формата можно найти в Интернете) – найти библиотеку для работы с vcf-файлами предлагается самостоятельно. При отсутствии библиотеки реализовать собственный класс, занимающийся конвертированием контакта в требуемый формат.