

Домашнее задание #5

Телефонная книга

Содержание

Телефонная книга	1
Задание	2
Модель	2
Интерфейс справочника	2
Основное окно	3
Редактирование контакта	3
Меню	5
Требования	6
Результат работы	8
Структура проекта	8
Тесты	8
Архив	8

Телефонная книга

Маленькой, но гордой компании поручили создать desktop-приложение телефонной книги.

Консольный интерфейс уже не подойдёт, так как Марья Васильевна из бухгалтерии и Игорь Петрович из отдела кадров ненавидят консоль(в том числе XBox и даже PlayStation).

Вам не повезло и вы работаете как раз в отделе, который ответственен за разработку.

Задание

Необходимо разработать телефонную книгу-справочник и интерфейс взаимодействия с ней.

Модель

Основной сущностью приложения является **Контакт**, который содержит поля: **Фамилия**, **Имя**, **Отчество**, **Мобильный** и **Домашний Телефон**, **Адрес**, **День Рождения**, **Заметки**, но ими не ограничена и по желанию вы можете добавить что-то еще.



Контактов, у которых не указана **Фамилия**, **Имя** и хотя бы один из телефонов быть не должно.

Интерфейс справочника

Дизайнер уволился, но, пока ждал трудовую книжку, набросал на салфетке эскиз приложения.

Вам удалось отыскать эту скомканную салфетку, развернув которую вы увидели:

Файл		Настройки		Справка		
Фамилия	Имя	Отчество	Мобильный/ ДомашнийТелефон	Адрес	День Рождения	Комментарий
Кучук	Александр		89851881480	Адрес	20.12.1990	Твиттер!
Не Кучук	Иван		89851881485	Не Адрес		
Заведеев	Денис		89851881486			
Лектор	Хочет	ДЗ	89851881489			Java
Удалить		Редактировать		Добавить		Поиск

Связавшись с дизайнером, вы выяснили некоторые детали, однако после этого он заблокировал вас во всех социальных сетях, попросив перестать преследовать его.

Основное окно

Основное окно приложения содержит меню, отображение контактов и панель взаимодействия с ними.

Отображение контактов предлагается сделать в виде таблицы.

Поиск контактов осуществляется через **Строку Поиска** в нижней части приложения. После нажатия <ENTER> или кнопки **Поиск** должна быть осуществлена фильтрация контактов на совпадение по введенной строке. Контакты, не удовлетворяющие введенному значению для поиска, отображаться не должны.

Поиск должен вестись как по фамилии, так и по имени или отчеству.

Необходимо иметь возможность не только добавить новый контакт, но и удалить выбранный, а также отредактировать его.

Редактирование контакта

Окно редактирования контакта открывается после того, как при выделенном контакте нажимается кнопка **Редактировать**.

Редактирование предлагается сделать в виде перечисления полей с возможностью внести изменения.

Редактирование контакта	
Имя	<input type="text" value="Александр"/>
Фамилия	<input type="text" value="Кучук"/>
Отчество	<input type="text"/>
Мобильный телефон	<input type="text" value="89851881480"/>
Домашний телефон	<input type="text"/>
Адрес	<input type="text" value="Адрес"/>
День рождения	<input type="text" value="20.12.1990"/>
<input type="text" value="Твиттер!"/>	
<div><input type="button" value="Отмена"/> <input type="button" value="Сохранить"/></div>	



Окно редактирования должно быть *модальным*.

Изменения должны отображаться в основном окне *только* после завершения редактирования и сохранения изменений.



Если вы в окне **Редактирование** изменили данные, но не нажали кнопку **Сохранить**, изменения контакта быть не должно.

Редактирование и указание **Даты рождения** должно быть с помощью [JavaFX DatePicker](#), либо подобной реализации в **Swing**, например, [JDatePicker](#).

Поля, в которые ввели некорректные данные, должны быть выделены предупреждающей красной рамкой.



За отображение подсказок причины непрохождения данных валидации будут начислены дополнительные баллы. Например, при попытке пользователя сохранить **Контакт**, у которого отсутствует **Фамилия**, вывести подсказку, о том, что это поле является обязательным для заполнения.

При наличии некорректных данных в окне **Редактирование контакта** пользователь должен или отменить(сбросить) все изменения **Контакта** нажатием кнопки **Отмена**, или исправить данные. Пока данные не исправлены сохранение **Контакта** с невалидными данными запрещается.

Меню

Обязательными элементами меню являются: **Файл**, **Настройки** и **Справка**.

Файл

По нажатию на **Файл**, открывается список пунктов меню:

- Добавить новый контакт

При нажатии происходит либо открытие окна добавления нового **Контакта**(его дизайн остаётся на ваше усмотрение, но если у вас нет идей, то сделайте его похожим на окно **Редактирования Контакта**, либо добавляется строка в таблицу, где и происходит заполнение данными.

- Редактирование

При нажатии происходит открытие окна **Редактирования Контакта** для выделенного в таблице контакта, если ни один из контактов не выделен, то пункт меню должен быть не активен.

- Удалить контакт

При нажатии происходит удаление контакта из телефонной книги.

- Выход

При нажатии выполняется выход из приложения.

Настройки

По нажатию на **Настройки**, открывается список пунктов меню:

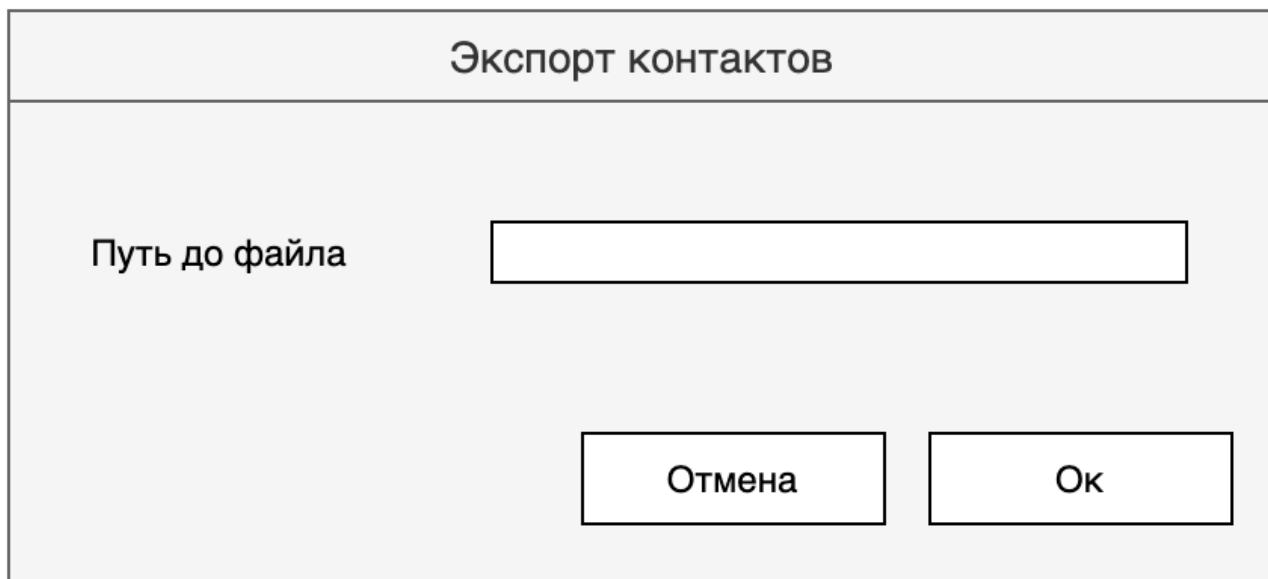
- Импортировать
- Экспортировать



Окна импорта и экспорта должны быть *модальными*.

Проходя мимо и сжавившись над вами, ваш начальник предложил сделать эти окна

максимально простыми, например:



Экспорт контактов

Путь до файла

Отмена

Ок

Где вы указываете путь до файла для импорта контактов, или путь для экспорта контактов.



Формат хранения данных остается на ваше усмотрение.

Файл выбирается с помощью *диалогового окна*, воспользоваться можно [Swing FileChooser](#), либо [JavaFx FileChooser](#), либо самописным, с подобным функционалом.

При повторном импорте одного и того же файла не должно быть дублирования контактов, текущие(добавленные до импорта) контакты не должны перезаписываться, при возникновении конфликтов(когда импортируется уже существующий и измененный контакт) должно показываться предупреждение. Уникальным идентификатором **Контакта** является комбинация полей **Имя**, **Фамилия** и **Отчество**.



Либо окно, позволяющее решить конфликт и за это будут начислены дополнительные баллы.

Справка

По нажатию на **Справка**, открывается окно, отображающее данные о том, кто разработал данный шедевр: номер группы, ваша фамилия, имя и послание потомкам, кто будет вынужден пользоваться этим приложением.

Требования

- Необходимо и обязательно чтобы данные телефонной книги не терялись при закрытии приложения. Это значит, что закрытие приложения не должно терять уже введенные и сохраненные данные.
- UI должен отвечать всем требованиям, описанным выше.
- UI не должен 'расползаться', при расширении/уменьшении окна.

- Проект должен включать в себя тесты, покрывающие весь слой логики поиска/сохранения контактов.
- Не смешивать UI-слой и логику поиска/сохранения контактов.
- Проект должен отвечать java code style guide.
- Валидация добавления данных. Ни при каких условиях не должно быть возможности сохранить контакт с некорректно введёнными данными.
- Постараться свести количество I/O операций к минимуму (не обращаться к файлу на диске по каждому чиху).

Результат работы

Структура проекта

Проект должен иметь 2 дерева исходных файлов:

- **основное**, которое содержит:
 - реализацию телефонной книги
- **для тестов**. В нём:
 - JUnit 5 `@Test` 'ы

Тесты

- Тесты должны **покрывать** код библиотеки: логику поиска, хранения и изменения данных.

Архив

Содержимое

Архив должен содержать:

1. Проект, созданный в IntelliJ IDEA
2. Исходные файлы приложения
3. **Исполняемый** `jar`-файл, включая файлы *конфигурации*, необходимые для его создания
 - Имя — в точности, что и у [архива](#), за исключением *расширения* файла
 - Для запуска может быть приложен файл *сценария командной строки* (*скрипт*)

Имя архива

Имя архива составляется по формуле `HW4_zzz_Фамилия_Имя.zip`, где:

- `zzz` - номер группы
- `Фамилия` - Ваша фамилия латиницей
- `Имя` - Ваше имя латиницей



Единственный допустимый формат архива - `.zip`.

Make Desktop Great Again!