# Домашнее задание #6

Телефонная книга. Эпизод 2

# Содержание

# Описание

Разработанное приложение телефонной книги оказалось полезным: его оценили как Марья Васильевна, так и Игорь Петрович.

Однако один из старших разработчиков предложил усовершенствовать приложение и хранить список контактов не в *файле*, а в **базе данных**.

# Задание

Необходимо изменить приложение телефонной книги так, чтобы **список контактов** хранился не в **файле**, а в **базе данных** — *Apache Derby*. Для взаимодействия с ней необходимо использовать **JDBC**.

Пользовательский интерфейс может быть:

**Графическим**

В этом случае можно оставить его таким, каким он был в решении задания 5.

**Консольным**

Им можно воспользоваться, если задание 5 не было выполнено или просто *при наличии желания*. Требования к нему описаны в разделе "Требования к консольному интерфейсу".

# Сущность "Контакт"

Основной сущностью по-прежнему остаётся "**контакт**", при этом у него добавляется новое поле - **Id**, всё остальное остаётся тем же самым:

*Таблица 1. Поля сущности контакт*

| **Поле** | **Обязательность** | **Описание** |
|---|---|---|
| Id | Да | Уникальный идентификатор, генерируемый автоматически. (см. Defining an identity column) |
| Фамилия | Да | - |
| Имя | Да | - |
| Отчество | Нет | |
| Мобильный телефон | Да, если не указан **домашний телефон** | |
| Домашний телефон | Да, если не указан **мобильный телефон** | |
| Адрес | Нет | |
| День рождения | Нет | |
| Заметки | Нет | Заметки или комментарии о контакте |

# Требования

## Общее

- База данных — **Apache Derby** в режиме встраивания. См. Embedded Derby.
- В базе данных ни при каких условиях не должны оказаться контакты, чьи данные введены *некорректно*.
- *Утечки ресурсов* недопустимы.

## Требования к консольному интерфейсу

В данном разделе содержатся требования к **консольному** интерфейсу. **Если выполнено ДЗ-5**, то можно оставить уже разработанный графический интерфейс.

Приложение в данном случае должно быть **интерактивным**, то есть программа должна постоянно ожидать от пользователя ввода команд.

Пример взаимодействия:

```
$ ./contacts
Телефонная книга. Автор - Фридрих Ницше.
Чтобы просмотреть список контактов введите 'list'.

> list
| # |    Имя    |  Фамилия | Дата рождения |     Моб.телефон    | Комментарий |
| - | --------- | -------- | ------------- | ------------------ | ----------- |
| 1 | Александр | Кучук    | 20.12.1990    | +7-(999)-123-45-67 | Twitter     |
| 2 | Денис     | Заведеев | 11.02.1996    | +7-(999)-123-45-68 | Read-only   |
| 3 | Елена     | Золотова | 30.12.1986    | +7-(999)-123-45-69 | Огоньки     |

> add
Введите имя:
> Katy
Введите фамилию:
> Perry
Введите отчество (или '-', если его нет)
> -
Введите дату рождения:
> 25.10.1984
Введите мобильный телефон (или '-', если его нет)
> +79172342542
Введите домашний телефон (или '-', если его нет)
> -
Введите адрес:
> US, California
Введите комментарий:
> Roar
Контакт успешно сохранён.
> list
| # |    Имя    |  Фамилия | Дата рождения |     Моб.телефон    | Заметки   |
| - | --------- | -------- | ------------- | ------------------ | --------- |
| 1 | Александр | Кучук    | 20.12.1990    | +7-(999)-123-45-67 | Twitter   |
| 2 | Денис     | Заведеев | 11.02.1996    | +7-(999)-123-45-68 | Read-only |
| 3 | Елена     | Золотова | 30.12.1986    | +7-(999)-123-45-69 | Огоньки   |
| 4 | Katy      | Perry    | 25.10.1984    | +7-(917)-234-25-42 | Roar      |
> exit
До встречи
```

- `$ ./contacts` запускает приложение.
- `>` — приглашение на ввод.

Вывод программы, формат ввода **не** должны *в точности* соответствовать *примеру* и в решении могут отличаться

## Минимальный список поддерживаемых команд

Необходимо поддержать следующий список команд:

- `list` - вывести список сохраненных контактов
- `add` - добавить новый контакт
- `delete` - удалить контакт
- `edit` - изменить контакт
- `about` - вывести текст привычного диалога "О программе", где указан её автор и послание потомкам
- `import` - импорт всех контактов из файла
- `export` - экспорт всех контактов в файл
- `exit` - выход

# Структура проекта

Проект должен иметь 2 дерева исходных файлов:

1. **Основное**, которое содержит:
   - реализацию телефонной книги
2. **Для тестов**. В нём:
   - JUnit 5 `@Test`' ы

# Тесты

Тесты должны **покрывать** как минимум логику работы с **данными**: *поиск, сохранение, обновление, удаление*.

В тестах можно создавать базу данных в памяти. См. Using in-memory databases.

# 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*).

Под файлами **конфигурации** для сборки `jar` в общем случае имеется в виду то, что должно быть понятно, как был собран `jar`. Если он собирается в **Intellij IDEA**, то ничего для этого делать специально **не нужно**: IDE создаёт эти файлы автоматически в директории `idea/artifacts`.

## Имя архива

Имя архива составляется по формуле `HW6_zzz_Фамилия_Имя.zip`, где:

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

Единственный допустимый формат архива - `.zip`.