1. **Постановка задачи**

Целью реализуемого проекта является создание телефонного справочника частных лиц. Телефонный справочник частных лиц содержит информацию о фамилиях и инициалах абонентов, их номерах телефонов, а также дополнительную информацию (адреса, электронную почту, ICQ и т.п.)

Основными задачами являются:

* Обеспечение быстрого поиска нужного телефона по фамилии или имени и в различных категориях: друзей, деловых партнеров, родственников и т.п.
* Наличие у одного лица нескольких номеров телефонов: рабочих, домашних, мобильных.
* Обеспечение обратного поиска: определение абонента по номеру телефона.

Дополнительными задачами являются:

* Реализация ввода и хранения дополнительной информации (адресов, фотографий и др.).
* Реализация поиска по дополнительной информации.
* Хранение истории смены номеров с возможностью поиска по старому номеру.

1. **Требования к системе**

Для работы программы требуется компьютер с операционной системой Windows 10, а также запущенный сервер PostgreSQL (user name=postgres, password=postgres) с хранящейся на нем базой данных.

1. **Входные и выходные документы**

Входные и выходные документы не предусмотрены.

1. **Входные и выходные документы**

В качестве системы управления базой данных выбрана СУБД PostgreSQL. Данная СУБД является объектно-реляционной, что позволяет ей поддерживать пользовательские объекты и их поведения, включая типы данных, функции, операции, домены и индексы. А также довольно хорошей производительностью при работе с большими объемами данных. Совместима со стандартом SQL`92.

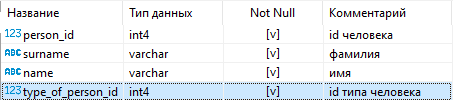
Для программной разработки использована среда IntelliJ IDEA 2021.3. Данная среда выбрана из-за наибольшего опыта работы и знания создания графических приложений именно на языке JavaFXML.

1. **Схема базы данных**

Базу данных образуют 4 таблицы: Person - таблица человека, Type of person - таблица типа человека, Info - таблица данных и Type of info - таблица типов данных.

Структура этих таблиц, следующая:

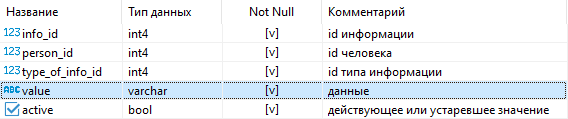
*Person: pk – person\_id*



*Type of person: pk – type\_of\_person\_id*



*Info: pk – info\_id*

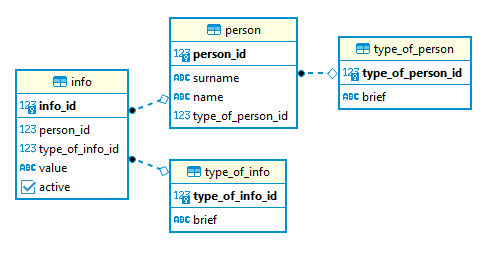


*Type of info: pk – type\_of\_info\_id*



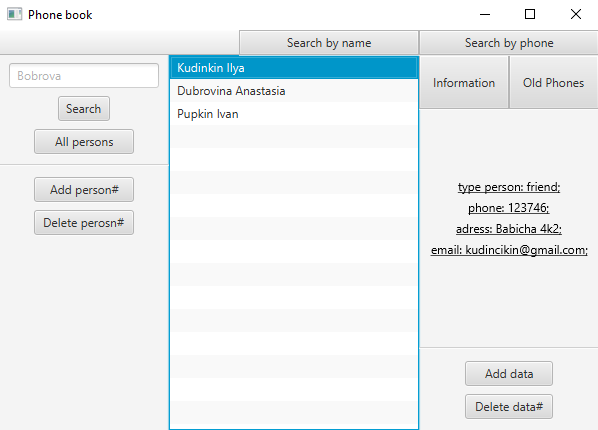
Внешние ключи:

* В таблице Info: поле person\_id ссылается на поле person\_id в таблице person, а так же type\_of\_info\_id на поле type\_of\_info\_id в таблице type\_of\_info.
* В таблице Person: поле type\_of\_person\_id ссылается на поле type\_of\_person\_id в таблице type\_of\_person\_id.



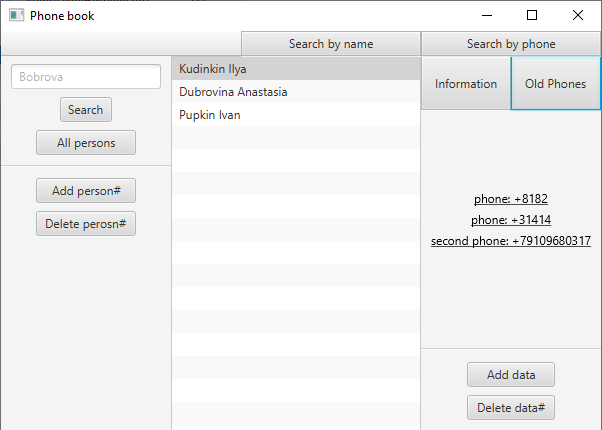
1. **Пользовательский интерфейс**

При запуске открывается основное окно (рис. 1). При его закрытии программа завершает работу. Так же, в саму программу будет добавлена кнопка выхода из программы.

*Рис. 1. Основное окно программы*

*(кнопки типа \*\*\*# не обязательны по ТЗ, их функционал будет реализован позже)*

В основном окне программы отображается список всех людей, которые были внесены в базу данных. Из данного списка можно выбрать интересующего вас человека и узнать всю хранящуюся о нем информацию в правом столбце основного окна. Также можно узнать старые номера телефонов, которые человек когда-либо использовал, нажав на кнопку “Old Phones” в правой верхней части экрана (рис. 2).



*Рис. 2. Основное окно программы (старые телефоны)*

В программе имеются 3 вида поиска информации:

1. *Поиск по телефону (Рис.3).* Кнопка поиска находится в правом верхнем углу главного окна. Запрашивается телефон или его фрагмент, а также возможен выбор телефона: любой, действующий или устаревший. При оставлении полей пустыми, любые данные будут являться верными.

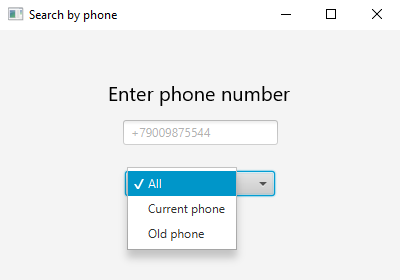
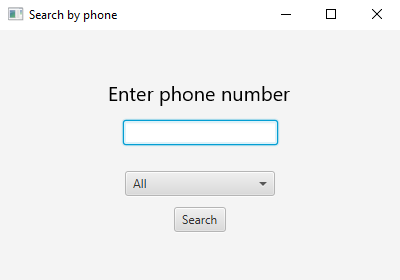


Рис. 3.

1. *Поиск по фамилии и/или имени (Рис. 2).* Кнопка поиска находится в правом верхнем углу главного окна. Запрашивается фамилия и/или имя, или фрагменты фамилии и/или имени, а также тип человека. Тип человека берется из базы данных и проявлении нового типа, также будет добавлен в выпадающий список. При оставлении полей пустыми, любые данные будут являться верными.

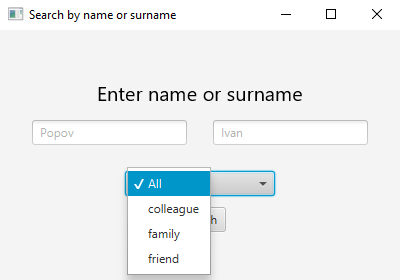
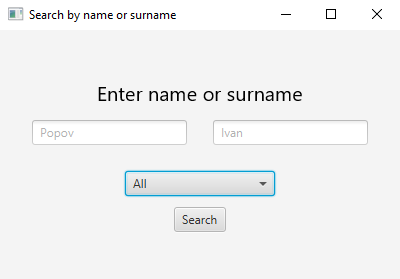


Рис. 2.

1. *Поиск по любой информации (Рис. 3).* Поле поиска находится в левой верхней части, чуть ниже кнопка поиска. С помощью данного поиска возможен поиск информации по всем типам данных, описывающих человека. Для поиска необходимо достаточно вписать часть той информации, что нужна (“Kudinkin” найдет по введённой строке “kud”). Данный поиск ищет совпадения с веденной строкой, так, например, введя в поле “adress”, программа выведет на экран всех людей, у которых существует данное поле, аналогично и для “friend”, программа выведет всех людей, которые записаны как “friend”.

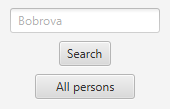


Рис. 3.

Так же чуть ниже поиска по любой информации, есть кнопка сбрасывающая любые фильтры (также данная кнопка обновляет всю информацию, хранящуюся в программе, обновляя ее на информацию из базы данных).

Кнопка “Add data”, находящаяся в правом низу основного окна, позволяет добавить новую информацию текущему выбранному человеку.

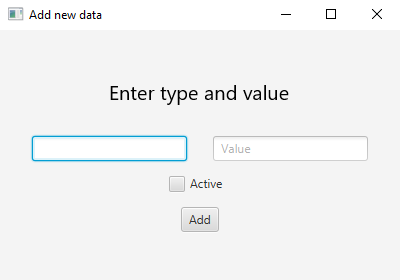


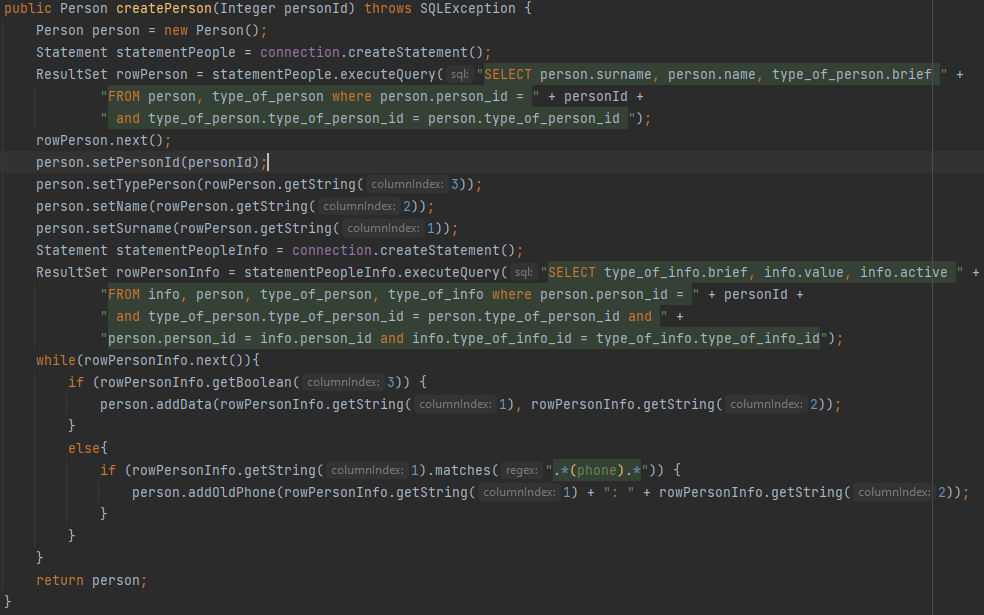
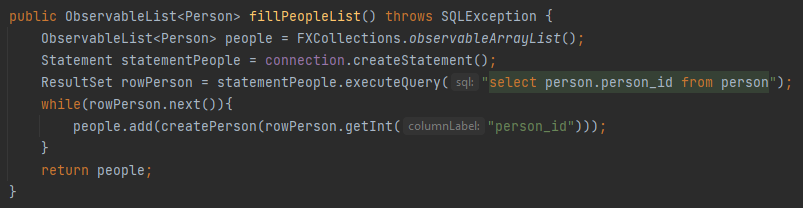
Рис. 4.

При нажатии на нее, открывается новое окно (Рис. 4), которое запрашивает ввод новых данных. Левое поле ввода отвечает за тип данных, если введенного не существует, то создает его. Правое же отвечает за значение, которое будет хранится. Оба поля должны быть заполнены, иначе программа не позволит вам нажать кнопку добавить. Кнопка “Active” отвечает за актуальность информации, если она не будет выбрана, то все данные типа “.\*phone.\*” будут записаны как старые телефоны, остальные проигнорированы и не добавлены в программу.

1. **Программная реализация**

Программа разработана на языке Java с использованием JavaFX. Работа с базой данных осуществлялась только с применением языка SQL. Для доступа к базе данных была использован интерфейс Connection, входящий в библиотеку java.sql.

В качестве примера ниже приведены 2 метода, которые позволяют инициализировать список людей для отображения в программе:



1. **Программная реализация**

Методика тестирования, а также тестовые наборы данных не разрабатывались.