МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

Язык программирования Java. Проектирование пользовательского интерфейса с использованием библиотеки Java FX.

Дисциплина: «Объектно-ориентированное программирование»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

|  |  |
| --- | --- |
| Руководитель | Сорокин А.Н. |
| Выполнили студенты | Пчелкина О.С. |
| Группа, курс | ВМ-41 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

2022 г.

**Цель работы**: научиться создавать приложения на языке программирования Java, освоить базовые элементы графической библиотеки Java FX.

# Вариант 1

Создайте приложение «Заметки», позволяющее пользователю сохранять и просматривать заметки. Внешний вид выберите сами. Должна быть кнопка сохранения, удаления и выведения на форму списка контактов. Данные сохраняются в список. Для этого ознакомьтесь с материалом по ссылке https://metanit.com/java/javafx/4.8.php . В ней описывается работа с ListView, которая может помочь вам выполнить задание.

Варианты заданий хранятся в файле или в базе данных.

Код приложения:

Файл NoteApplication.java

package com.example.lab5;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class NoteApplication extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(NoteApplication.class.getResource("note-view.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 750, 700);  
 stage.setTitle("Notes");  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 public static void main(String[] args) {  
 *launch*();  
 }  
}

Файл NoteControlle.java

package com.example.lab5;  
  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.\*;  
  
import java.nio.charset.StandardCharsets;  
import java.nio.file.\*;  
  
import java.io.\*;  
import java.net.URL;  
import java.util.\*;  
  
//реализовываем наш класс с помощью интерфейса Initializable, которые инициализирует наш контроллер  
public class NoteController implements Initializable {  
  
 public Label noteLabel;  
 public Button addButton;  
 public Button deleteButton;  
 public Label enterNoteLabel;  
 public Button addContactButton;  
 public Label contactLabel;  
 public Button deleteContactButton;  
 public Button viewContactButton;  
 public Label addContactLabel;  
 public Label nameContactLabel;  
 public Label numberContactLabel;  
 //специальный интерфейс, в котором есть специальные методы, которые могут ловить полезные исключения  
 private ObservableList<String> notes;  
 private ObservableList<String> contacts;  
 @FXML  
 private ListView<String> noteListView;  
 @FXML  
 private ListView<String> contactListView;  
 @FXML  
 private TextField nameContactTextField;  
 @FXML  
 private TextField numberContactTextField;  
 @FXML  
 private TextField noteTextField;  
 final String PATH\_CONTACT\_FILE = "D:\\lab5\\src\\main\\java\\com\\example\\lab5\\contact.txt";  
 final String PATH\_NOTES\_FILE = "D:\\lab5\\src\\main\\java\\com\\example\\lab5\\notes.txt";  
 @FXML  
 void btnAddNoteClicked() {  
 try {  
 File file = new File(PATH\_NOTES\_FILE);  
  
 if (!file.exists())  
 file.createNewFile();  
 String text = noteTextField.getText() + "\n";  
 //записываем в файл строку из TextField  
 Files.*write*(Paths.*get*(PATH\_NOTES\_FILE), text.getBytes(), StandardOpenOption.*APPEND*);  
 //добавление строки в массив, которая была введена в TextField  
 notes.add(noteTextField.getText());  
 //выводим содержимое массива в ListView  
 noteListView.setItems(notes);  
 //очищаем содержимое TextField  
 noteTextField.clear();  
 } catch (IOException e) {  
 System.*out*.println("Произошла ошибка: " + e);  
 }  
 }  
  
 @FXML  
 void btnAddContactClicked(ActionEvent event) throws FileNotFoundException {  
 try {  
 File file = new File(PATH\_CONTACT\_FILE);  
 if (!file.exists())  
 file.createNewFile();  
 String text = "Имя: " + nameContactTextField.getText() + ", номер телефона: " + numberContactTextField.getText() + "\n";  
 Files.*write*(Paths.*get*(PATH\_CONTACT\_FILE), text.getBytes(), StandardOpenOption.*APPEND*);  
 nameContactTextField.clear();  
 numberContactTextField.clear();  
 } catch (IOException e) {  
 System.*out*.println("Произошла ошибка: " + e);  
 }  
 }  
  
 @FXML  
 void btnDeleteNoteClicked(ActionEvent event) throws IOException{  
 Path notesFilePath = Paths.*get*("D:\\lab5\\src\\main\\java\\com\\example\\lab5\\notes.txt");  
 List<String> fileContent;  
 //создаем отдельный список с содержимым файла notes.txt  
 fileContent = new ArrayList<>(Files.*readAllLines*(notesFilePath, StandardCharsets.*UTF\_8*));  
 //объявляем переменную, которая содержит в себе выбранную строку для удаления  
 String selected = noteListView.getSelectionModel().getSelectedItem();  
 //пробегаемся по текстовому файлу  
 for (int i = 0; i < fileContent.size(); i++) {  
 //если мы в файле находим строку, которая равна selected  
 if (fileContent.get(i).equals(selected)) {  
 //то удаляем ее  
 fileContent.remove(selected);  
 }  
 }  
 //перезаписываем оставшиеся строки в файл  
 Files.*write*(notesFilePath, fileContent, StandardCharsets.*UTF\_8*);  
 //удаляем из ListView выбранную строку  
 noteListView.getItems().removeAll(noteListView.getSelectionModel().getSelectedItems());  
 }  
  
 @FXML  
 void btnDeleteContactClicked(ActionEvent event) throws IOException {  
 Path contactFilePath = Paths.*get*("D:\\lab5\\src\\main\\java\\com\\example\\lab5\\contact.txt");  
 List<String> fileContent;  
 fileContent = new ArrayList<>(Files.*readAllLines*(contactFilePath, StandardCharsets.*UTF\_8*));  
 String selected = contactListView.getSelectionModel().getSelectedItem();  
 for (int i = 0; i < fileContent.size(); i++) {  
 if (fileContent.get(i).equals(selected)) {  
 //то удаляем ее  
 fileContent.remove(selected);  
 }  
 }  
 Files.*write*(contactFilePath, fileContent, StandardCharsets.*UTF\_8*);  
 contactListView.getItems().removeAll(contactListView.getSelectionModel().getSelectedItems());  
 }  
  
 @FXML  
 void btnViewContactClicked(ActionEvent event) throws FileNotFoundException {  
 BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(PATH\_CONTACT\_FILE)));  
 try {  
 //очищаем ListView для того, чтобы заполнить его обновленной инфой  
 contactListView.getItems().clear();  
 File file = new File(PATH\_CONTACT\_FILE);  
 if (!file.exists())  
 file.createNewFile();  
 //чтение первой строки в файле  
 String line = br.readLine();  
 //добавление первой строки в listView  
 contacts.add(line);  
 //чтение и добавление остальных строк  
 while ((line = br.readLine()) != null) {  
 contacts.add(line);  
 contactListView.setItems(contacts);  
 }  
 } catch (IOException e) {  
 System.*out*.println("Произошла ошибка: " + e);  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 System.*out*.println("Произошла ошибка: " + e);  
 }  
 }  
 }  
  
 //Перегруженный метод инициализации, вызывается, чтобы инициализировать контроллер после того,  
 // как его корневой элемент был полностью обработан. В нем мы просто заполняем наш ListView с  
 // заметками после запуска приложения  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 //массивы данных в двух ListView  
 notes = FXCollections.*observableArrayList*();  
 contacts = FXCollections.*observableArrayList*();  
  
 BufferedReader br;  
 try {  
 br = new BufferedReader(new InputStreamReader(new FileInputStream(PATH\_NOTES\_FILE)));  
 } catch (FileNotFoundException e) {  
 throw new RuntimeException(e);  
 }  
 try {  
 contactListView.getItems().clear();  
 //чтение первой строки в файле  
 String line = br.readLine();  
 //добавление первой строки в listView  
 notes.add(line);  
 //чтение и добавление остальных строк  
 while ((line = br.readLine()) != null) {  
 notes.add(line);  
 noteListView.setItems(notes);  
 }  
 } catch (IOException e) {  
 System.*out*.println("Произошла ошибка: " + e);  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 System.*out*.println("Произошла ошибка: " + e);  
 }  
 }  
 }  
}

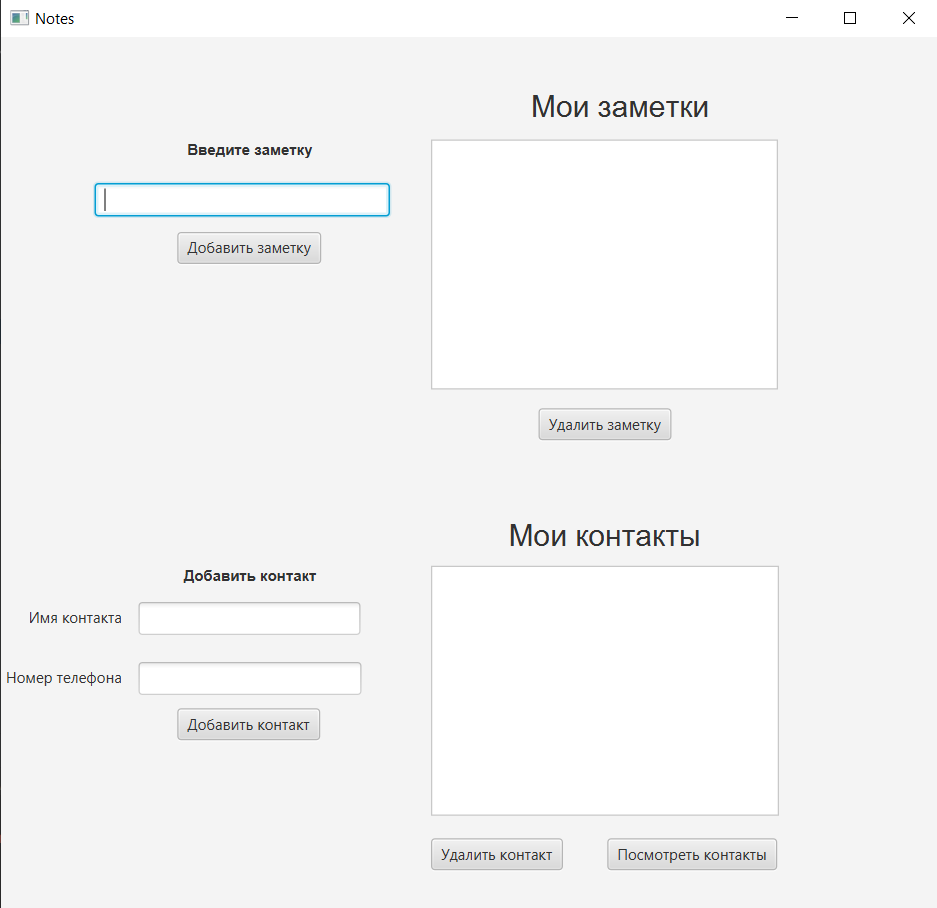
Файл module-info.java

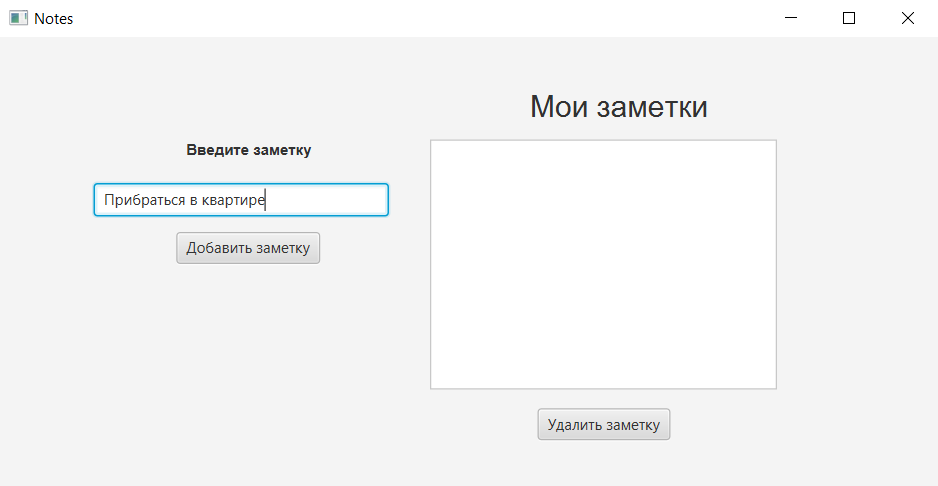
module com.example.lab5 {  
 requires javafx.controls;  
 requires javafx.fxml;  
  
  
 opens com.example.lab5 to javafx.fxml;  
 exports com.example.lab5;  
}

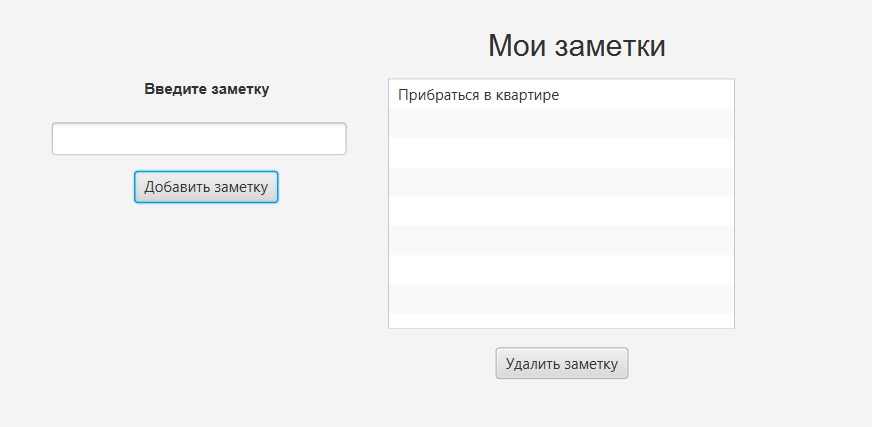
Файл note-view.fxml

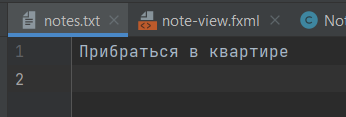
<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.ListView?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="713.0" prefWidth="698.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.lab5.NoteController">  
 <children>  
 <Label fx:id="noteLabel" layoutX="424.0" layoutY="42.0" text="Мои заметки">  
 <font>  
 <Font name="Arial" size="24.0" />  
 </font>  
 </Label>  
 <ListView fx:id="noteListView" layoutX="344.0" layoutY="82.0" prefHeight="200.0" prefWidth="277.0" />  
 <Button fx:id="addButton" layoutX="141.0" layoutY="156.0" mnemonicParsing="false" onAction="#btnAddNoteClicked" text="Добавить заметку" />  
 <Button fx:id="deleteButton" layoutX="430.0" layoutY="297.0" mnemonicParsing="false" onAction="#btnDeleteNoteClicked" text="Удалить заметку" />  
 <TextField fx:id="noteTextField" layoutX="75.0" layoutY="117.0" prefHeight="26.0" prefWidth="236.0" />  
 <Label fx:id="enterNoteLabel" layoutX="149.0" layoutY="82.0" text="Введите заметку">  
 <font>  
 <Font name="Arial Bold" size="12.0" />  
 </font>  
 </Label>  
 <Button fx:id="addContactButton" layoutX="141.0" layoutY="537.0" mnemonicParsing="false" onAction="#btnAddContactClicked" text="Добавить контакт" />  
 <ListView fx:id="contactListView" layoutX="344.0" layoutY="423.0" prefHeight="200.0" prefWidth="278.0" />  
 <Label fx:id="contactLabel" layoutX="406.0" layoutY="385.0" text="Мои контакты">  
 <font>  
 <Font name="Arial" size="24.0" />  
 </font>  
 </Label>  
 <Button fx:id="deleteContactButton" layoutX="344.0" layoutY="641.0" mnemonicParsing="false" onAction="#btnDeleteContactClicked" text="Удалить контакт" />  
 <Button fx:id="viewContactButton" layoutX="485.0" layoutY="641.0" mnemonicParsing="false" onAction="#btnViewContactClicked" text="Посмотреть контакты" />  
 <Label fx:id="addContactLabel" layoutX="146.0" layoutY="423.0" text="Добавить контакт">  
 <font>  
 <Font name="Arial Bold" size="12.0" />  
 </font>  
 </Label>  
 <TextField fx:id="nameContactTextField" layoutX="110.0" layoutY="452.0" prefHeight="26.0" prefWidth="177.0" />  
 <TextField fx:id="numberContactTextField" layoutX="110.0" layoutY="500.0" prefHeight="26.0" prefWidth="178.0" />  
 <Label fx:id="nameContactLabel" layoutX="22.0" layoutY="456.0" text="Имя контакта" />  
 <Label fx:id="numberContactLabel" layoutX="4.0" layoutY="504.0" text="Номер телефона" />  
 </children>  
</AnchorPane>

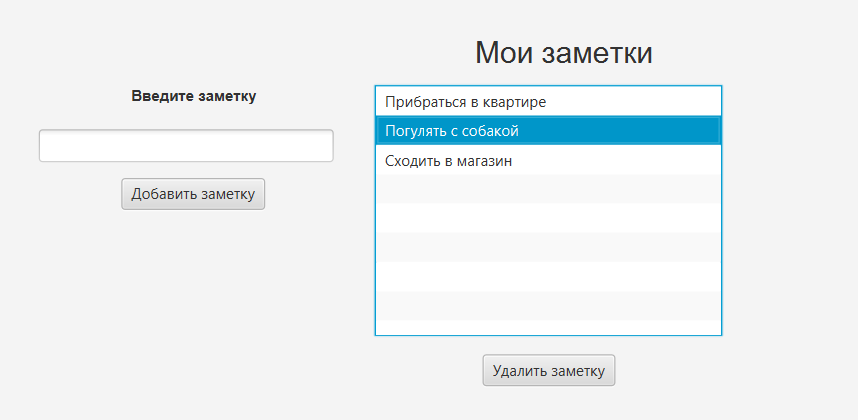
Тестирование:

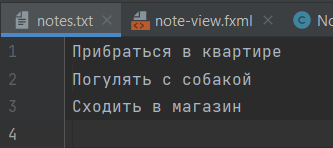


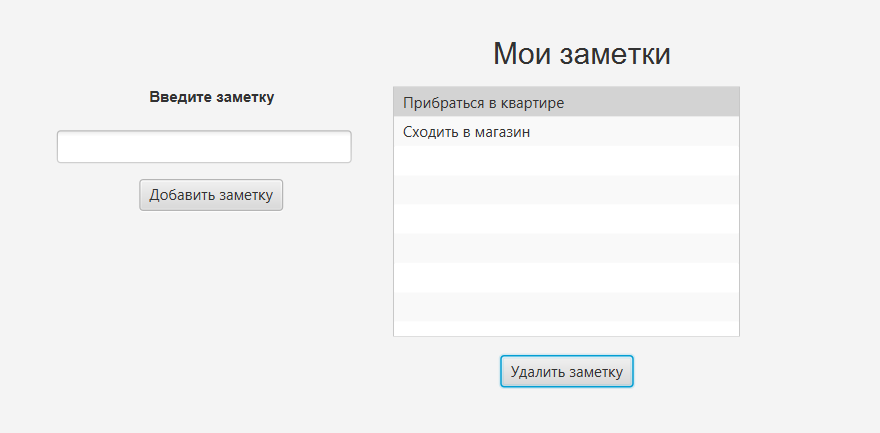


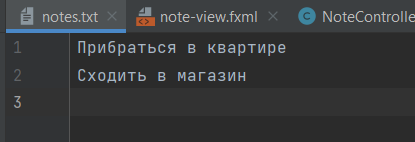


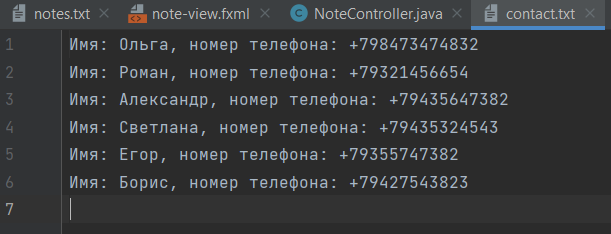


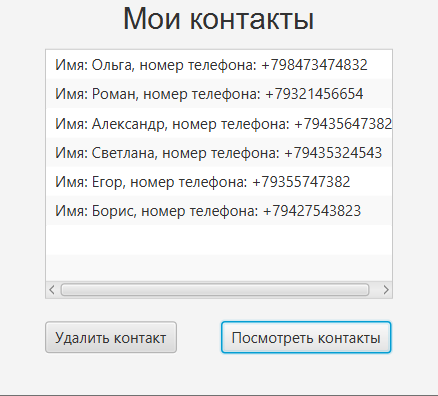


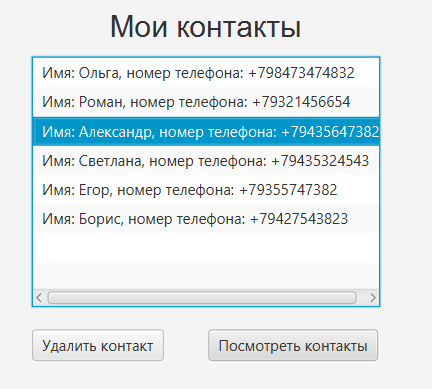


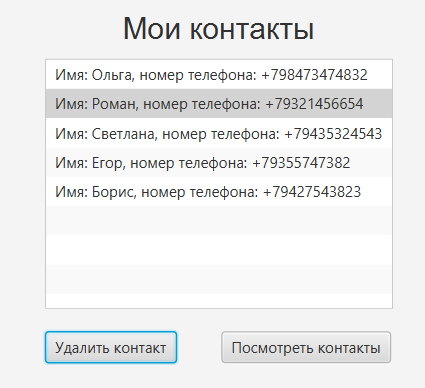


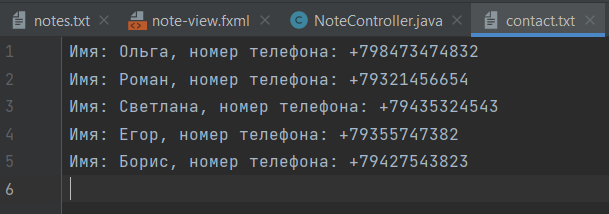


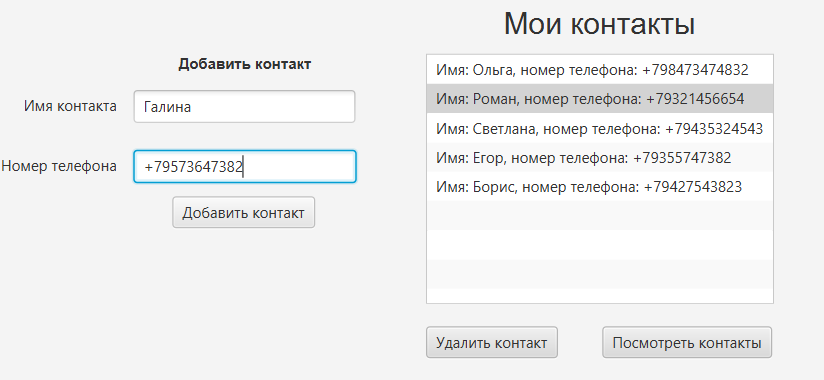


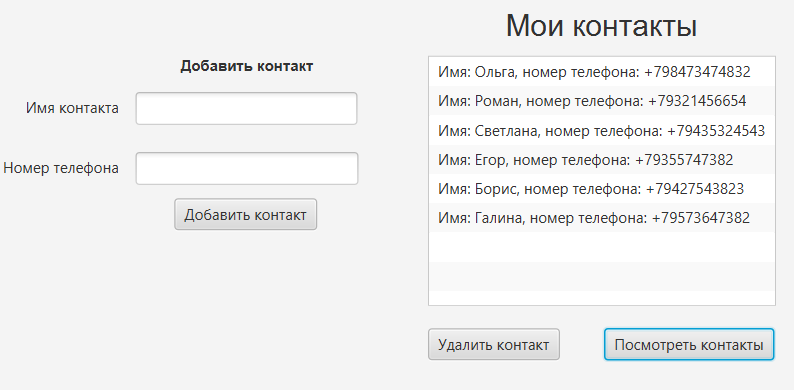


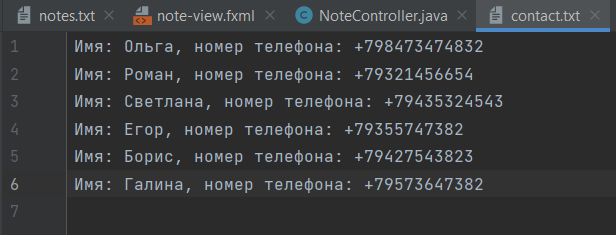












# Контрольные вопросы

1. Опишите основные возможности библиотеки JavaFX.

*JavaFX представляет инструментарий для создания кроссплатформенных графических приложений на платформе Java.*

*JavaFX позволяет создавать приложения с богатой насыщенной графикой благодаря использованию аппаратного ускорения графики и возможностей GPU.*

*С помощью JavaFX можно создавать программы для различных операционных систем: Windows, MacOS, Linux, Android, iOS и для самых различных устройств: десктопы, смартфоны, планшеты, встроенные устройства, ТВ. Приложение на JavaFX будет работать везде, где установлена исполняемая среда Java (JRE).*

*JavaFX предоставляет большие возможности по сравнению с рядом других подобных платформ, в частности, по сравнению со Swing. Это и большой набор элементов управления, и возможности по работе с мультимедиа, двухмерной и трехмерной графикой, декларативный способ описания интерфейса с помощью языка разметки FXML, возможность стилизации интерфейса с помощью CSS, интеграция со Swing*

2. Опишите основные элементы языка разметки FXML.

*В FXML элемент XML представляет одно из следующего:*

* *Экземпляр class*
* *Свойство экземпляра class*
* *"Статическое" свойство*
* *"Определить" блок*

*Блок кода сценария*

*Экземпляры класса, свойства экземпляра, статические свойства, и определяют блоки, обсуждаются в этом разделе ниже. Сценарии обсуждаются в более позднем разделе.*

*Экземпляр класса*

*Экземпляры класса могут быть созданы в FXML несколькими способами. Наиболее распространенное через элементы объявления экземпляра, которые просто создают новый экземпляр class по имени. Другие способы создать экземпляры class включают ссылающиеся существующие значения, копируя существующие значения, и включая внешние файлы FXML.*