Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЕТ

по лабораторной работе №4

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ РАЗРАБОТКИ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ПЛАТФОРМЫ JAVAFX 2.

Выполнил:

ст. гр. ИС/б-21-о

Куркчи А. Э.

Проверил:

Кузнецов С. А.

Севастополь

2016

1. ЦЕЛЬ РАБОТЫ

В ходе выполнения данной лабораторной работы необходимо ознакомиться с особенностями платформы JavaFX 2 и приобрести практические навыки создания насыщенных пользовательских интерфейсов Java-программ.

1. ПОСТАНОВКА ЗАДАЧИ

С использованием компонентов JavaFX 2 необходимо создать Java приложение реализующее добавление, редактирование и удаление данных заданного по варианту типа информации T (см. табл. 4.1). Данные отображать в виде таблицы. Реализовать поля ввода для добавления новых записей. Редактирование записей реализовать в таблице (использовать CellValueFactory). Предусмотреть возможность загрузки информации из текстового файла и сохранения в текстовый файл. Данные столбца N отображать в виде автоматически обновляющегося графика/диаграммы G (см. табл. 4.1).

|  |  |  |
| --- | --- | --- |
| A | 4 | StackedBarChart |

Вариант № 6

A: Компакт диск (Название альбома, Исполнитель, Количество треков, Длительность звучания);

1. ТЕКСТ ПРОГРАММЫ

Файл Main.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab05**;**

**import** javafx**.**application**.**Application**;**

**import** javafx**.**collections**.**FXCollections**;**

**import** javafx**.**collections**.**ObservableList**;**

**import** javafx**.**fxml**.**FXMLLoader**;**

**import** javafx**.**scene**.**Parent**;**

**import** javafx**.**scene**.**Scene**;**

**import** javafx**.**stage**.**Stage**;**

**import** java**.**io**.**IOException**;**

**import** java**.**io**.**RandomAccessFile**;**

**import** java**.**util**.**Random**;**

public class Main **extends** Application **{**

private Stage primaryStage**;**

private ObservableList**<**CD**>** cdData **=** FXCollections**.**observableArrayList**();**

@Override

public void start**(**Stage stage**)** **throws** Exception **{**

primaryStage **=** stage**;**

stage**.**setTitle**(**"SevSU semester.04 Java Lab.05"**);**

Parent root **=** FXMLLoader**.**load**(**getClass**().**getResource**(**"main.fxml"**));**

stage**.**setScene**(new** Scene**(**root**));**

initMainLayout**();**

**}**

public void initMainLayout**()** **{**

**try** **{**

FXMLLoader loader **=** **new** FXMLLoader**();**

loader**.**setLocation**(**Main**.**class**.**getResource**(**"main.fxml"**));**

Scene scene **=** **new** Scene**(**loader**.**load**());**

MainController mainController **=** loader**.**getController**();**

mainController**.**setMain**(this);**

primaryStage**.**setScene**(**scene**);**

primaryStage**.**show**();**

**}** **catch** **(**IOException e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

public Stage getPrimaryStage**()** **{**

**return** primaryStage**;**

**}**

public ObservableList**<**CD**>** getCDData**()** **{**

**return** cdData**;**

**}**

public static void main**(**String**[]** args**)** **{**

launch**(**args**);**

**}**

**}**

Файл CD.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab05**;**

**import** javafx**.**beans**.**property**.**IntegerProperty**;**

**import** javafx**.**beans**.**property**.**SimpleIntegerProperty**;**

**import** javafx**.**beans**.**property**.**SimpleStringProperty**;**

**import** javafx**.**beans**.**property**.**StringProperty**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**io**.**IOException**;**

**import** java**.**util**.**Comparator**;**

public class CD **{**

protected final StringProperty title**;**

protected final StringProperty author**;**

protected final IntegerProperty trackCount**;**

protected final IntegerProperty duration**;**

public CD**(**String title**,** String author**,** int trackCount**,** int duration**)** **{**

**this.**title **=** **new** SimpleStringProperty**(**title**);**

**this.**author **=** **new** SimpleStringProperty**(**author**);**

**this.**trackCount **=** **new** SimpleIntegerProperty**(**trackCount**);**

**this.**duration **=** **new** SimpleIntegerProperty**(**duration**);**

**}**

public String getTitle**()** **{**

**return** title**.**get**();**

**}**

public String getAuthor**()** **{**

**return** author**.**get**();**

**}**

public int getTrackCount**()** **{**

**return** trackCount**.**get**();**

**}**

public int getDuration**()** **{**

**return** duration**.**get**();**

**}**

public void setTitle**(**String title**)** **{**

**this.**title**.**set**(**title**);**

**}**

public void setAuthor**(**String author**)** **{**

**this.**author**.**set**(**author**);**

**}**

public void setTrackCount**(**int trackCount**)** **{**

**this.**trackCount**.**set**(**trackCount**);**

**}**

public void setDuration**(**int duration**)** **{**

**this.**duration**.**set**(**duration**);**

**}**

public StringProperty titleProperty**()** **{**

**return** title**;**

**}**

public StringProperty authorProperty**()** **{**

**return** author**;**

**}**

public IntegerProperty trackCountProperty**()** **{**

**return** trackCount**;**

**}**

public IntegerProperty durationProperty**()** **{**

**return** duration**;**

**}**

public void write**(**DataOutputStream dos**)** **throws** IOException **{**

dos**.**writeUTF**(**getTitle**());**

dos**.**writeUTF**(**getAuthor**());**

dos**.**writeInt**(**getTrackCount**());**

dos**.**writeInt**(**getDuration**());**

**}**

public static CD read**(**DataInputStream dis**)** **throws** IOException **{**

**return** **new** CD**(**dis**.**readUTF**(),** dis**.**readUTF**(),** dis**.**readInt**(),** dis**.**readInt**());**

**}**

**}**

Файл MainController.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab05**;**

**import** javafx**.**collections**.**FXCollections**;**

**import** javafx**.**collections**.**ListChangeListener**;**

**import** javafx**.**collections**.**ObservableList**;**

**import** javafx**.**event**.**EventType**;**

**import** javafx**.**fxml**.**FXML**;**

**import** javafx**.**scene**.**chart**.**StackedBarChart**;**

**import** javafx**.**scene**.**chart**.**XYChart**;**

**import** javafx**.**scene**.**control**.\*;**

**import** javafx**.**scene**.**control**.**cell**.**PropertyValueFactory**;**

**import** javafx**.**scene**.**control**.**cell**.**TextFieldTableCell**;**

**import** javafx**.**stage**.**FileChooser**;**

**import** javafx**.**util**.**StringConverter**;**

**import** javafx**.**util**.**converter**.**IntegerStringConverter**;**

**import** java**.**io**.\*;**

**import** java**.**util**.**IllegalFormatException**;**

**import** java**.**util**.**Optional**;**

public class MainController **{**

@FXML

private TableView**<**CD**>** table**;**

@FXML

private TableColumn**<**CD**,** String**>** tableTitleC**;**

@FXML

private TableColumn**<**CD**,** String**>** tableAuthorC**;**

@FXML

private TableColumn**<**CD**,** Integer**>** tableTrackCountC**;**

@FXML

private TableColumn**<**CD**,** Integer**>** tableDurationC**;**

@FXML

private TextField titleF**;**

@FXML

private TextField authorF**;**

@FXML

private TextField trackCountF**;**

@FXML

private TextField durationF**;**

@FXML

private Button loadB**;**

@FXML

private Button saveB**;**

@FXML

private Button deleteB**;**

@FXML

private Button addB**;**

@FXML

private StackedBarChart**<**String**,** Integer**>** chart**;**

private FileChooser fileChooser **=** **new** FileChooser**();**

private Main main**;**

public MainController**()** **{**

**}**

@FXML

private void initialize**()** **{**

fileChooser**.**setInitialDirectory**(new** File**(**"."**));**

chart**.**getXAxis**().**setLabel**(**"Author"**);**

chart**.**getYAxis**().**setLabel**(**"Duration"**);**

chart**.**setAnimated**(false);**

table**.**setEditable**(true);**

tableTitleC**.**setCellValueFactory**(**cellData **->** cellData**.**getValue**().**titleProperty**());**

tableAuthorC**.**setCellValueFactory**(**cellData **->** cellData**.**getValue**().**authorProperty**());**

tableTrackCountC**.**setCellValueFactory**(**cellData **->** cellData**.**getValue**().**trackCountProperty**().**asObject**());**

tableDurationC**.**setCellValueFactory**(**cellData **->** cellData**.**getValue**().**durationProperty**().**asObject**());**

tableTitleC**.**setCellValueFactory**(new** PropertyValueFactory**<>(**"title"**));**

tableAuthorC**.**setCellValueFactory**(new** PropertyValueFactory**<>(**"author"**));**

tableTrackCountC**.**setCellValueFactory**(new** PropertyValueFactory**<>(**"trackCount"**));**

tableDurationC**.**setCellValueFactory**(new** PropertyValueFactory**<>(**"duration"**));**

tableTitleC**.**setCellFactory**(**TextFieldTableCell**.**forTableColumn**());**

tableTitleC**.**setOnEditCommit**(**event **->** **{**

String val **=** event**.**getNewValue**();**

**if(**val**.**isEmpty**())** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields can`t be empty"**,** "Title field can`t be empty"**,** **true);**

**throw** **new** RuntimeException**(**"Title field can`t be empty"**);**

**}**

event**.**getRowValue**().**setTitle**(**val**);**

updateChart**();**

**});**

tableAuthorC**.**setCellFactory**(**TextFieldTableCell**.**forTableColumn**());**

tableAuthorC**.**setOnEditCommit**(**event **->** **{**

String val **=** event**.**getNewValue**();**

**if(**val**.**isEmpty**())** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields can`t be empty"**,** "Author field can`t be empty"**,** **true);**

**throw** **new** RuntimeException**(**"Author field can`t be empty"**);**

**}**

event**.**getRowValue**().**setAuthor**(**val**);**

updateChart**();**

**});**

tableTrackCountC**.**setCellFactory**(**TextFieldTableCell**.**forTableColumn**(new** IntegerStringConverter**()));**

tableTrackCountC**.**setOnEditCommit**(**event **->** **{**

int val **=** event**.**getNewValue**();**

**if(**val **<** 1**)** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields has wrong format"**,** "Track count should be greater then 1"**,** **true);**

**throw** **new** RuntimeException**(**"Track count should be greater then 1"**);**

**}**

event**.**getRowValue**().**setTrackCount**(**val**);**

updateChart**();**

**});**

tableDurationC**.**setCellFactory**(**TextFieldTableCell**.**forTableColumn**(new** IntegerStringConverter**()));**

tableDurationC**.**setOnEditCommit**(**event **->** **{**

int val **=** event**.**getNewValue**();**

**if(**val **<** 1**)** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields can`t be empty"**,** "Duration should be greater then 1"**,** **true);**

**throw** **new** RuntimeException**(**"Duration should be greater then 1"**);**

**}**

event**.**getRowValue**().**setDuration**(**val**);**

updateChart**();**

**});**

**}**

public void setMain**(**Main main**)** **{**

**this.**main **=** main**;**

table**.**setItems**(**main**.**getCDData**());**

table**.**setEditable**(true);**

**}**

public void updateChart**()** **{**

ObservableList**<**StackedBarChart**.**Series**<**String**,** Integer**>>** data **=** FXCollections**.**observableArrayList**();**

Optional**<**XYChart**.**Series**<**String**,** Integer**>>** seriesOptional**;**

**for(**CD cd **:** main**.**getCDData**())** **{**

seriesOptional **=** data**.**stream**().**filter**(**p **->** p**.**getName**().**equals**(**cd**.**getTitle**())).**findFirst**();**

**if(**seriesOptional**.**isPresent**())** **{**

seriesOptional**.**get**().**getData**().**add**(new** XYChart**.**Data**<>(**cd**.**getAuthor**(),** cd**.**getDuration**()));**

**}** **else** **{**

ObservableList**<**XYChart**.**Data**<**String**,** Integer**>>** list **=** FXCollections**.**observableArrayList**();**

list**.**add**(new** XYChart**.**Data**<>(**cd**.**getAuthor**(),** cd**.**getDuration**()));**

data**.**add**(new** XYChart**.**Series**<>(**cd**.**getTitle**(),** list**));**

**}**

**}**

chart**.**getData**().**setAll**(**data**);**

**}**

@FXML

public void handleAddAction**()** **{**

String title **=** titleF**.**getText**();**

String author **=** authorF**.**getText**();**

String trackCountS **=** trackCountF**.**getText**();**

String durationS **=** durationF**.**getText**();**

**if(**title**.**isEmpty**())** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields can`t be empty"**,** "Title field can`t be empty"**,** **true);**

**return;**

**}**

**if(**author**.**isEmpty**())** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields can`t be empty"**,** "Author field can`t be empty"**,** **true);**

**return;**

**}**

**if(**trackCountS**.**isEmpty**())** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields can`t be empty"**,** "Track count field can`t be empty"**,** **true);**

**return;**

**}**

int trackCount**;**

**try** **{**

trackCount **=** Integer**.**valueOf**(**trackCountS**);**

**if(**trackCount **<** 1**)** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields has wrong format"**,** "Track count should be greater then 1"**,** **true);**

**return;**

**}**

**}** **catch(**NumberFormatException ex**)** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields has wrong format"**,** "Track count should be an integer"**,** **true);**

**return;**

**}**

**if(**durationS**.**isEmpty**())** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields can`t be empty"**,** "Duration field can`t be empty"**,** **true);**

**return;**

**}**

int duration**;**

**try** **{**

duration **=** Integer**.**valueOf**(**durationS**);**

**if(**duration **<** 1**)** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields has wrong format"**,** "Duration should be greater then 1"**,** **true);**

**return;**

**}**

**}** **catch(**NumberFormatException ex**)** **{**

showAlert**(**Alert**.**AlertType**.**WARNING**,** "Form warning"**,** "Fields has wrong format"**,** "Duration should be an integer"**,** **true);**

**return;**

**}**

main**.**getCDData**().**add**(new** CD**(**title**,** author**,** trackCount**,** duration**));**

updateChart**();**

**}**

@FXML

public void handleDeleteAction**()** **{**

CD cd **=** table**.**getSelectionModel**().**getSelectedItem**();**

**if(**cd **!=** **null)** **{**

main**.**getCDData**().**remove**(**cd**);**

updateChart**();**

**}**

**}**

@FXML

public void handleLoadAction**()** **{**

fileChooser**.**setTitle**(**"Load file"**);**

File file **=** fileChooser**.**showOpenDialog**(**main**.**getPrimaryStage**());**

**if(**file **!=** **null)** **{**

**try(**DataInputStream dis **=** **new** DataInputStream**(new** FileInputStream**(**file**)))** **{**

int n **=** dis**.**readInt**();**

ObservableList**<**CD**>** list **=** FXCollections**.**observableArrayList**();**

**for(**int i**=**0**;**i**<**n**;**i**++)** **{**

list**.**add**(**CD**.**read**(**dis**));**

**}**

main**.**getCDData**().**setAll**(**list**);**

**}** **catch** **(**IOException ex**)** **{**

showAlert**(**Alert**.**AlertType**.**ERROR**,** "Error"**,** "File load error"**,** "Could not load file. It has wrong format or access denied."**,** **true);**

**return;**

**}**

**}**

updateChart**();**

**}**

@FXML

public void handleSaveAction**()** **{**

fileChooser**.**setTitle**(**"Save file"**);**

File file **=** fileChooser**.**showSaveDialog**(**main**.**getPrimaryStage**());**

**if(**file **!=** **null)** **{**

**try(**DataOutputStream dos **=** **new** DataOutputStream**(new** FileOutputStream**(**file**)))** **{**

dos**.**writeInt**(**main**.**getCDData**().**size**());**

**for(**CD cd **:** main**.**getCDData**())** **{**

cd**.**write**(**dos**);**

**}**

**}** **catch** **(**IOException ex**)** **{**

showAlert**(**Alert**.**AlertType**.**ERROR**,** "Error"**,** "File save error"**,** "Could not save file. Access denied."**,** **true);**

**return;**

**}**

**}**

updateChart**();**

**}**

private void showAlert**(**Alert**.**AlertType type**,** String title**,** String header**,** String text**,** boolean wait**)** **{**

Alert alert **=** **new** Alert**(**type**);**

alert**.**initOwner**(**main**.**getPrimaryStage**());**

alert**.**setTitle**(**title**);**

alert**.**setHeaderText**(**header**);**

alert**.**setContentText**(**text**);**

**if(**wait**)** **{**

alert**.**showAndWait**();**

**}** **else** **{**

alert**.**show**();**

**}**

**}**

**}**

Файл main.fxml

<?xml version=**"1.0"** encoding=**"UTF-8"**?>

<?import javafx.scene.chart.\*?>

<?import javafx.scene.control.\*?>

<?import java.lang.\*?>

<?import javafx.scene.layout.\*?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.layout.GridPane?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<GridPane alignment=**"center"** hgap=**"10"** vgap=**"10"** xmlns=**"http://javafx.com/javafx/8"** xmlns:fx=**"http://javafx.com/fxml/1"** fx:controller=**"ru.justnero.study.sevsu.java.lab05.MainController"**>

<columnConstraints>

<ColumnConstraints hgrow=**"NEVER"** maxWidth=**"10.0"** minWidth=**"10.0"** prefWidth=**"10.0"** />

<ColumnConstraints />

<ColumnConstraints />

<ColumnConstraints />

<ColumnConstraints />

<ColumnConstraints />

<ColumnConstraints hgrow=**"NEVER"** maxWidth=**"10.0"** minWidth=**"10.0"** prefWidth=**"10.0"** />

</columnConstraints>

<rowConstraints>

<RowConstraints maxHeight=**"10.0"** minHeight=**"10.0"** prefHeight=**"10.0"** vgrow=**"NEVER"** />

<RowConstraints />

<RowConstraints />

<RowConstraints />

<RowConstraints />

<RowConstraints maxHeight=**"10.0"** minHeight=**"10.0"** prefHeight=**"10.0"** vgrow=**"NEVER"** />

</rowConstraints>

<children>

<TableView fx:id=**"table"** editable=**"true"** prefHeight=**"200.0"** prefWidth=**"200.0"** GridPane.columnIndex=**"1"** GridPane.columnSpan=**"4"** GridPane.rowIndex=**"1"**>

<columns>

<TableColumn fx:id=**"tableTitleC"** prefWidth=**"75.0"** text=**"Title"** />

<TableColumn fx:id=**"tableAuthorC"** prefWidth=**"75.0"** text=**"Author"** />

<TableColumn fx:id=**"tableTrackCountC"** prefWidth=**"75.0"** text=**"Track count"** />

<TableColumn fx:id=**"tableDurationC"** prefWidth=**"75.0"** text=**"Duration"** />

</columns>

<columnResizePolicy>

<TableView fx:constant=**"CONSTRAINED\_RESIZE\_POLICY"** />

</columnResizePolicy>

</TableView>

<TextField GridPane.columnIndex=**"1"** GridPane.rowIndex=**"3"** />

<TextField fx:id=**"titleF"** GridPane.columnIndex=**"1"** GridPane.rowIndex=**"3"** />

<TextField fx:id=**"authorF"** layoutX=**"10.0"** layoutY=**"220.0"** GridPane.columnIndex=**"2"** GridPane.rowIndex=**"3"** />

<Label text=**"Title"** GridPane.columnIndex=**"1"** GridPane.rowIndex=**"2"** />

<Label text=**"Author"** GridPane.columnIndex=**"2"** GridPane.rowIndex=**"2"** />

<TextField fx:id=**"trackCountF"** GridPane.columnIndex=**"3"** GridPane.rowIndex=**"3"** />

<TextField fx:id=**"durationF"** GridPane.columnIndex=**"4"** GridPane.rowIndex=**"3"** />

<Label text=**"Track count"** GridPane.columnIndex=**"3"** GridPane.rowIndex=**"2"** />

<Label text=**"Duration"** GridPane.columnIndex=**"4"** GridPane.rowIndex=**"2"** />

<Button fx:id=**"addB"** alignment=**"CENTER"** defaultButton=**"true"** mnemonicParsing=**"false"** onAction=**"#handleAddAction"** text=**"Add"** GridPane.columnIndex=**"4"** GridPane.hgrow=**"ALWAYS"** GridPane.rowIndex=**"4"** />

<Button fx:id=**"saveB"** mnemonicParsing=**"false"** onAction=**"#handleSaveAction"** text=**"Save"** GridPane.columnIndex=**"2"** GridPane.rowIndex=**"4"** />

<Button fx:id=**"loadB"** mnemonicParsing=**"false"** onAction=**"#handleLoadAction"** text=**"Load"** GridPane.columnIndex=**"1"** GridPane.rowIndex=**"4"** />

<StackedBarChart fx:id=**"chart"** GridPane.columnIndex=**"5"** GridPane.rowIndex=**"1"**>

<xAxis>

<CategoryAxis side=**"BOTTOM"** />

</xAxis>

<yAxis>

<NumberAxis side=**"LEFT"** />

</yAxis>

</StackedBarChart>

<Button fx:id=**"deleteB"** mnemonicParsing=**"false"** onAction=**"#handleDeleteAction"** text=**"Delete"** GridPane.columnIndex=**"3"** GridPane.rowIndex=**"4"** />

</children>

</GridPane>

1. ИНТЕРФЕЙС ПРОГРАММЫ

Графический интерфейс программы состоит из одного окна. В нём присутствуют кнопки загрузки из файла и сохранения в файл, удаления выделенной записи и добавления новой. Редактирование записей происходит прямо в ячейках таблицы. Окно представлено на рисунке 1.

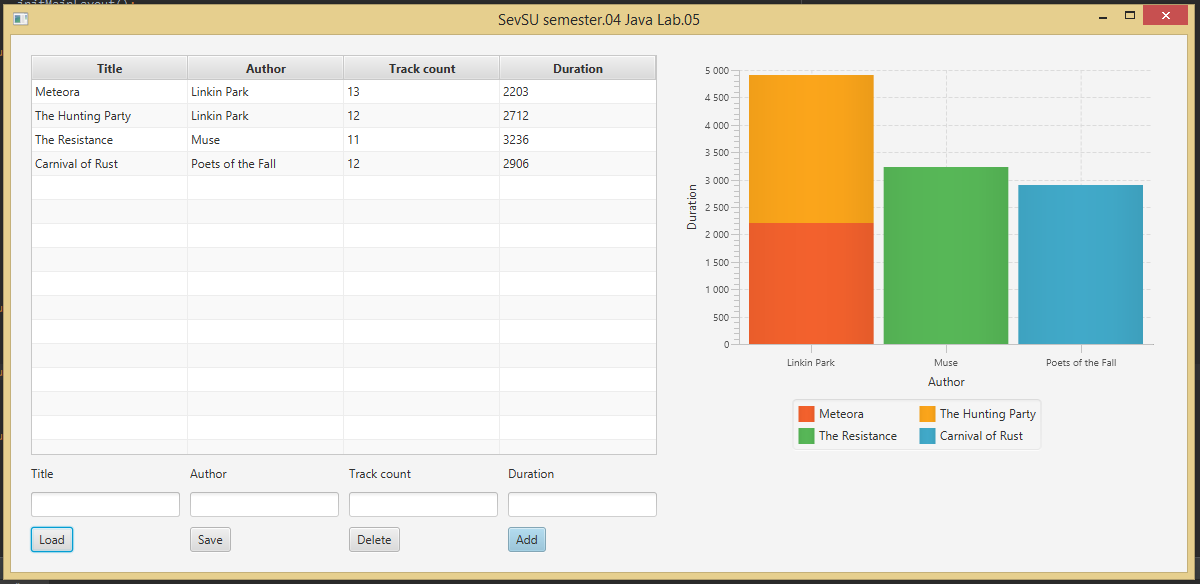


Рисунок 1 – Графический интерфейс программы

Вывод

В ходе выполнения данной лабораторной работы изучены особенности платформы JavaFX 2, приобретены практические навыки создания насыщенных пользовательских интерфейсов Java-программ. Изучено средство верстки представлений JavaFX 2 Scene Builder. В ходе работы было отмечено, что использование JavaFX 2 является отличным решением с точки зрения красоты и удобства интерфейса, однако потребляет достаточно много оперативной памяти, а именно в районе 100 МБ при старте.