Національний університет «Львівська Політехніка»

Кафедра Систем Штучного Інтелекту



Лабораторна робота №8

Виконав: студент КН-111

Зомчак Богдан

Перевірив:

Грабовська Н. Р.

**Львів-2019**

Тема:

Розробка графічного інтерфейсу користувача

Мета:

Отримання навичок використання засобів клієнтських технологій (Client Technologies) платформи Java SE.

package lab08;

import java.io.File;

import java.io.IOException;

import java.util.prefs.Preferences;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.Marshaller;

import javax.xml.bind.Unmarshaller;

import javafx.application.Application;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.BorderPane;

import javafx.stage.Modality;

import javafx.stage.Stage;

import lab08.model.Route;

import lab08.model.RouteList;

import lab08.view.RootLayoutController;

import lab08.view.RouteEditDialogController;

import lab08.view.RouteOverviewController;

public class MainApp extends Application {

private Stage primaryStage;

private BorderPane rootLayout;

private ObservableList<Route> personData = FXCollections.observableArrayList();

public ObservableList<Route> getPersonData() {

return personData;

}

@Override

public void start(Stage primaryStage) {

this.primaryStage = primaryStage;

this.primaryStage.setTitle("BusStation");

initRootLayout();

showPersonOverview();

}

public void initRootLayout() {

try {

FXMLLoader loader = new FXMLLoader();

loader.setLocation(MainApp.class.getResource("view/RootLayout.fxml"));

rootLayout = (BorderPane) loader.load();

Scene scene = new Scene(rootLayout);

primaryStage.setScene(scene);

RootLayoutController controller = loader.getController();

controller.setMainApp(this);

primaryStage.show();

} catch (IOException e) {

e.printStackTrace();

}

}

public File getPersonFilePath() {

Preferences prefs = Preferences.userNodeForPackage(MainApp.class);

String filePath = prefs.get("filePath", null);

if (filePath != null) {

return new File(filePath);

} else {

return null;

}

}

public void setPersonFilePath(File file) {

Preferences prefs = Preferences.userNodeForPackage(MainApp.class);

if (file != null) {

prefs.put("filePath", file.getPath());

primaryStage.setTitle("BusStation - " + file.getName());

} else {

prefs.remove("filePath");

primaryStage.setTitle("BusStation");

}

}

public boolean showPersonEditDialog(Route person) {

try {

FXMLLoader loader = new FXMLLoader();

loader.setLocation(MainApp.class.getResource("view/RouteEditDialog.fxml"));

AnchorPane page = (AnchorPane) loader.load();

// Create the dialog Stage.

Stage dialogStage = new Stage();

dialogStage.setTitle("Edit route");

dialogStage.initModality(Modality.WINDOW\_MODAL);

dialogStage.initOwner(primaryStage);

Scene scene = new Scene(page);

dialogStage.setScene(scene);

RouteEditDialogController controller = loader.getController();

controller.setDialogStage(dialogStage);

controller.setRoute(person);

dialogStage.showAndWait();

return controller.isOkClicked();

} catch (IOException e) {

e.printStackTrace();

return false;

}

}

public void showPersonOverview() {

try {

FXMLLoader loader = new FXMLLoader();

loader.setLocation(MainApp.class.getResource("view/Overview.fxml"));

AnchorPane personOverview = (AnchorPane) loader.load();

rootLayout.setCenter(personOverview);

RouteOverviewController controller = loader.getController();

controller.setMainApp(this);

} catch (IOException e) {

e.printStackTrace();

}

}

public void loadRouteDataFromFile(File file) {

try {

JAXBContext context = JAXBContext.newInstance(RouteList.class);

Unmarshaller um = context.createUnmarshaller();

RouteList wrapper = (RouteList) um.unmarshal(file);

personData.clear();

personData.addAll(wrapper.getRoutes());

setPersonFilePath(file);

} catch (Exception e) {

Alert alert = new Alert(AlertType.ERROR);

alert.setTitle("Error");

alert.setHeaderText("Could not load data");

alert.setContentText("Could not load data from file:\n" + file.getPath());

alert.showAndWait();

}

}

public void saveRouteDataToFile(File file) {

try {

JAXBContext context = JAXBContext.newInstance(RouteList.class);

Marshaller m = context.createMarshaller();

m.setProperty(Marshaller.JAXB\_FORMATTED\_OUTPUT, true);

RouteList wrapper = new RouteList();

wrapper.setRoutes(personData);

m.marshal(wrapper, file);

setPersonFilePath(file);

} catch (Exception e) {

Alert alert = new Alert(AlertType.ERROR);

alert.setTitle("Error");

alert.setHeaderText("Could not save data");

alert.setContentText("Could not save data to file:\n" + file.getPath());

alert.showAndWait();

}

}

public Stage getPrimaryStage() {

return primaryStage;

}

public static void main(String[] args) {

launch(args);

}

} package lab08.model;

//import javafx.beans.property.IntegerProperty;

//import javafx.beans.property.ObjectProperty;

//import javafx.beans.property.SimpleIntegerProperty;

//import javafx.beans.property.SimpleObjectProperty;

import javafx.beans.property.SimpleStringProperty;

import javafx.beans.property.StringProperty;

public class Route {

private final StringProperty name;

private final StringProperty departure;

private final StringProperty arrival;

private final StringProperty placeFree;

private final StringProperty day;

private final StringProperty number;

public Route() {

this(null, null);

}

public Route(String name, String number) {

this.name = new SimpleStringProperty(name);

this.number = new SimpleStringProperty(number);

this.placeFree = new SimpleStringProperty("12");

this.departure = new SimpleStringProperty("12:00");

this.arrival = new SimpleStringProperty("13:00");

this.day = new SimpleStringProperty("23");

}

public String getNazva() {

return name.get();

}

public String getChasPr() {

return departure.get();

}

public String getChasVd() {

return arrival.get();

}

public String getPlaceFree() {

return placeFree.get();

}

public String getDay() {

return day.get();

}

public String getNomer() {

return number.get();

}

public void setNazva(String name) {

this.name.set(name);

}

public void setChasPr(String departure) {

this.departure.set(departure);

}

public void setChasVd(String arrival) {

this.arrival.set(arrival);

}

public void setPlaceFree(String placeFree) {

this.placeFree.set(placeFree);

}

public void setDay(String day) {

this.day.set(day);

}

public void setNomer(String number) {

this.number.set(number);

}

public StringProperty nazvaProperty() {

return name;

}

public StringProperty chasPrProperty() {

return departure;

}

public StringProperty chasVdProperty() {

return arrival;

}

public StringProperty placeFreeProperty() {

return placeFree;

}

public StringProperty dayProperty() {

return day;

}

public StringProperty nomerProperty() {

return number;

}

} package lab08.model;

import java.util.List;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "routes")

public class RouteList {

private List<Route> routes;

@XmlElement(name = "routes")

public List<Route> getRoutes() {

return routes;

}

public void setRoutes(List<Route> routes) {

this.routes = routes;

}

} package lab08.view;

import java.io.File;

import javafx.fxml.FXML;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.stage.FileChooser;

import lab08.MainApp;

public class RootLayoutController {

private MainApp mainApp;

public void setMainApp(MainApp mainApp) {

this.mainApp = mainApp;

}

@FXML

private void handleNew() {

mainApp.getPersonData().clear();

mainApp.setPersonFilePath(null);

}

@FXML

private void handleOpen() {

FileChooser fileChooser = new FileChooser();

FileChooser.ExtensionFilter extFilter = new FileChooser.ExtensionFilter(

"XML files (\*.xml)", "\*.xml");

fileChooser.getExtensionFilters().add(extFilter);

File file = fileChooser.showOpenDialog(mainApp.getPrimaryStage());

if (file != null) {

mainApp.loadRouteDataFromFile(file);

}

}

@FXML

private void handleSave() {

File personFile = mainApp.getPersonFilePath();

if (personFile != null) {

mainApp.saveRouteDataToFile(personFile);

} else {

handleSaveAs();

}

}

@FXML

private void handleSaveAs() {

FileChooser fileChooser = new FileChooser();

FileChooser.ExtensionFilter extFilter = new FileChooser.ExtensionFilter(

"XML files (\*.xml)", "\*.xml");

fileChooser.getExtensionFilters().add(extFilter);

File file = fileChooser.showSaveDialog(mainApp.getPrimaryStage());

if (file != null) {

if (!file.getPath().endsWith(".xml")) {

file = new File(file.getPath() + ".xml");

}

mainApp.saveRouteDataToFile(file);

}

}

@FXML

private void handleAbout() {

Alert alert = new Alert(AlertType.INFORMATION);

alert.setTitle("");

alert.setHeaderText("About");

alert.setContentText("");

alert.showAndWait();

}

@FXML

private void handleExit() {

System.exit(0);

}

} package lab08.view;

import javafx.fxml.FXML;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.TextField;

import javafx.stage.Stage;

import lab08.model.Route;

public class RouteEditDialogController {

@FXML

private TextField nameField;

@FXML

private TextField chasPrField;

@FXML

private TextField chasVdField;

@FXML

private TextField freePlaceField;

@FXML

private TextField dayField;

@FXML

private TextField nomerField;

private Stage dialogStage;

private Route route;

private boolean okClicked = false;

@FXML

private void initialize() {

}

public void setDialogStage(Stage dialogStage) {

this.dialogStage = dialogStage;

}

public void setRoute(Route route) {

this.route = route;

nameField.setText(route.getNazva());

chasPrField.setText(route.getChasPr());

chasVdField.setText(route.getChasVd());

freePlaceField.setText(route.getPlaceFree());

dayField.setText(route.getDay());

nomerField.setText(route.getNomer());

}

public boolean isOkClicked() {

return okClicked;

}

@FXML

private void handleOk() {

if (isInputValid()) {

route.setNazva(nameField.getText());

route.setChasPr(chasPrField.getText());

route.setChasVd(chasVdField.getText());

route.setPlaceFree(freePlaceField.getText());

route.setDay(dayField.getText());

route.setNomer(nomerField.getText());

okClicked = true;

dialogStage.close();

}

}

@FXML

private void handleCancel() {

dialogStage.close();

}

private boolean isInputValid() {

String errorMessage = "";

if (nameField.getText() == null || nameField.getText().length() == 0) {

errorMessage += "No valid name!\n";

}

if (chasPrField.getText() == null || chasPrField.getText().length() == 0) {

errorMessage += "No valid time!\n";

}

if (chasVdField.getText() == null || chasVdField.getText().length() == 0) {

errorMessage += "No valid time!\n";

}

if (freePlaceField.getText() == null || freePlaceField.getText().length() == 0) {

errorMessage += "No valid number of places!\n";

}

if (dayField.getText() == null || dayField.getText().length() == 0) {

errorMessage += "No valid day!\n";

}

if (nomerField.getText() == null || nomerField.getText().length() == 0) {

errorMessage += "No valid number!\n";

}

if (errorMessage.length() == 0) {

return true;

} else {

Alert alert = new Alert(AlertType.ERROR);

alert.initOwner(dialogStage);

alert.setTitle("Invalid Fields");

alert.setHeaderText("Please correct invalid fields");

alert.setContentText(errorMessage);

alert.showAndWait();

return false;

}

}

} package lab08.view;

import javafx.fxml.FXML;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import lab08.MainApp;

import lab08.model.Route;

public class RouteOverviewController {

@FXML

private TableView<Route> routeTable;

@FXML

private TableColumn<Route, String> firstNameColumn;

@FXML

private TableColumn<Route, String> lastNameColumn;

//Ticket

@FXML

private Label nazvaLabel;

@FXML

private Label chasPrLabel;

@FXML

private Label chasVdLabel;

@FXML

private Label placeFreeLabel;

@FXML

private Label dayLabel;

@FXML

private Label nomerLabel;

private MainApp mainApp;

public RouteOverviewController() {

}

@FXML

private void initialize() {

firstNameColumn.setCellValueFactory(cellData -> cellData.getValue().nazvaProperty());

lastNameColumn.setCellValueFactory(cellData -> cellData.getValue().nomerProperty());

showRouteDetails(null);

routeTable.getSelectionModel().selectedItemProperty().addListener(

(observable, oldValue, newValue) -> showRouteDetails(newValue));

}

public void setMainApp(MainApp mainApp) {

this.mainApp = mainApp;

routeTable.setItems(mainApp.getPersonData());

}

private void showRouteDetails(Route route) {

if (route != null) {

nazvaLabel.setText(route.getNazva());

chasPrLabel.setText(route.getChasPr());

chasVdLabel.setText(route.getChasVd());

placeFreeLabel.setText(route.getPlaceFree());

dayLabel.setText(route.getDay());

nomerLabel.setText(route.getNomer());

} else {

nazvaLabel.setText("");

chasPrLabel.setText("");

chasVdLabel.setText("");

placeFreeLabel.setText("");

dayLabel.setText("");

nomerLabel.setText("");

}

}

@FXML

private void handleDeleteRoute() {

int selectedIndex = routeTable.getSelectionModel().getSelectedIndex();

if(selectedIndex>=0) {

routeTable.getItems().remove(selectedIndex);

}

else {

Alert alert = new Alert(AlertType.WARNING);

alert.initOwner(mainApp.getPrimaryStage());

alert.setTitle("No Selection");

alert.setHeaderText("No Person Selected");

alert.setContentText("Please select a person in the table.");

alert.showAndWait();

}

}

@FXML

private void handleNewPerson() {

Route tempPerson = new Route();

boolean okClicked = mainApp.showPersonEditDialog(tempPerson);

if (okClicked) {

mainApp.getPersonData().add(tempPerson);

}

}

@FXML

private void handleEditPerson() {

Route selectedPerson = routeTable.getSelectionModel().getSelectedItem();

if (selectedPerson != null) {

boolean okClicked = mainApp.showPersonEditDialog(selectedPerson);

if (okClicked) {

showRouteDetails(selectedPerson);

}

} else {

Alert alert = new Alert(AlertType.WARNING);

alert.initOwner(mainApp.getPrimaryStage());

alert.setTitle("No Selection");

alert.setHeaderText("No Person Selected");

alert.setContentText("Please select a person in the table.");

alert.showAndWait();

}

}

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

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

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

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

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

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

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

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

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

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

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

<AnchorPane prefHeight=*"300.0"* prefWidth=*"600.0"* xmlns=*"http://javafx.com/javafx/8.0.201"* xmlns:fx=*"http://javafx.com/fxml/1"* fx:controller=*"lab08.view.RouteOverviewController"*>

<children>

<SplitPane dividerPositions=*"0.29797979797979796"* prefHeight=*"300.0"* prefWidth=*"600.0"* AnchorPane.bottomAnchor=*"0.0"* AnchorPane.leftAnchor=*"0.0"* AnchorPane.rightAnchor=*"0.0"* AnchorPane.topAnchor=*"0.0"*>

<items>

<AnchorPane minHeight=*"0.0"* minWidth=*"0.0"* prefHeight=*"160.0"* prefWidth=*"100.0"*>

<children>

<TableView fx:id=*"routeTable"* prefHeight=*"200.0"* prefWidth=*"200.0"* AnchorPane.bottomAnchor=*"0.0"* AnchorPane.leftAnchor=*"0.0"* AnchorPane.rightAnchor=*"0.0"* AnchorPane.topAnchor=*"0.0"*>

<columns>

<TableColumn fx:id=*"firstNameColumn"* prefWidth=*"75.0"* text=*"Route"* />

<TableColumn fx:id=*"lastNameColumn"* prefWidth=*"75.0"* text=*"Number"* />

</columns>

<columnResizePolicy>

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

</columnResizePolicy>

</TableView>

</children>

</AnchorPane>

<AnchorPane minHeight=*"0.0"* minWidth=*"0.0"* prefHeight=*"160.0"* prefWidth=*"100.0"*>

<children>

<Label layoutX=*"28.0"* layoutY=*"14.0"* text=*"Menu:"* AnchorPane.leftAnchor=*"5.0"* AnchorPane.topAnchor=*"5.0"* />

<GridPane AnchorPane.leftAnchor=*"5.0"* AnchorPane.rightAnchor=*"5.0"* AnchorPane.topAnchor=*"30.0"*>

<columnConstraints>

<ColumnConstraints hgrow=*"SOMETIMES"* minWidth=*"10.0"* prefWidth=*"100.0"* />

<ColumnConstraints hgrow=*"SOMETIMES"* minWidth=*"10.0"* prefWidth=*"100.0"* />

</columnConstraints>

<rowConstraints>

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

</rowConstraints>

<children>

<Label fx:id=*"nazvaLabel"* text=*"Label"* GridPane.columnIndex=*"1"* />

<Label fx:id=*"chasPrLabel"* text=*"Label"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"1"* />

<Label fx:id=*"chasVdLabel"* text=*"Label"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"2"* />

<Label fx:id=*"placeFreeLabel"* text=*"Label"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"3"* />

<Label text=*"Name"* />

<Label text=*"Arriving time"* GridPane.rowIndex=*"1"* />

<Label text=*"Departure time"* GridPane.rowIndex=*"2"* />

<Label text=*"Free places"* GridPane.rowIndex=*"3"* />

<Label text=*"Day"* GridPane.rowIndex=*"4"* />

<Label text=*"Number"* GridPane.rowIndex=*"5"* />

<Label fx:id=*"dayLabel"* text=*"Label"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"4"* />

<Label fx:id=*"nomerLabel"* text=*"Label"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"5"* />

</children>

</GridPane>

<ButtonBar layoutX=*"138.0"* layoutY=*"238.0"* prefHeight=*"46.0"* prefWidth=*"265.0"* AnchorPane.bottomAnchor=*"14.0"* AnchorPane.leftAnchor=*"138.0"* AnchorPane.rightAnchor=*"14.0"* AnchorPane.topAnchor=*"238.0"*>

<buttons>

<Button mnemonicParsing=*"false"* onAction=*"#handleNewPerson"* text=*"New..."* />

<Button mnemonicParsing=*"false"* onAction=*"#handleEditPerson"* text=*"Edit..."* />

<Button mnemonicParsing=*"false"* onAction=*"#handleDeleteRoute"* text=*"Delete"* />

</buttons>

</ButtonBar>

</children>

</AnchorPane>

</items>

</SplitPane>

</children>

</AnchorPane>

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

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

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

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

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

<BorderPane prefHeight=*"400.0"* prefWidth=*"600.0"* xmlns=*"http://javafx.com/javafx/8.0.201"* xmlns:fx=*"http://javafx.com/fxml/1"* fx:controller=*"lab08.view.RootLayoutController"*>

<top>

<MenuBar BorderPane.alignment=*"CENTER"*>

<menus>

<Menu mnemonicParsing=*"false"* text=*"File"*>

<items>

<MenuItem mnemonicParsing=*"false"* onAction=*"#handleNew"* text=*"New"* />

<MenuItem mnemonicParsing=*"false"* onAction=*"#handleOpen"* text=*"Open.."* />

<MenuItem mnemonicParsing=*"false"* onAction=*"#handleSave"* text=*"Save"* />

<MenuItem mnemonicParsing=*"false"* onAction=*"#handleSaveAs"* text=*"Save as..."* />

<MenuItem mnemonicParsing=*"false"* onAction=*"#handleExit"* text=*"Exit"* />

</items>

</Menu>

<Menu mnemonicParsing=*"false"* text=*"Edit"*>

<items>

<MenuItem mnemonicParsing=*"false"* text=*"Delete"* />

</items>

</Menu>

<Menu mnemonicParsing=*"false"* text=*"Help"*>

<items>

<MenuItem mnemonicParsing=*"false"* text=*"About"* />

</items>

</Menu>

</menus>

</MenuBar>

</top>

</BorderPane>

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

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

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

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

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

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

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

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

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

<AnchorPane prefHeight=*"250.0"* prefWidth=*"400.0"* xmlns=*"http://javafx.com/javafx/8.0.201"* xmlns:fx=*"http://javafx.com/fxml/1"* fx:controller=*"lab08.view.RouteEditDialogController"*>

<children>

<GridPane prefHeight=*"195.0"* prefWidth=*"400.0"* AnchorPane.bottomAnchor=*"55.0"* AnchorPane.leftAnchor=*"10.0"* AnchorPane.rightAnchor=*"10.0"* AnchorPane.topAnchor=*"0.0"*>

<columnConstraints>

<ColumnConstraints hgrow=*"SOMETIMES"* minWidth=*"10.0"* prefWidth=*"100.0"* />

<ColumnConstraints hgrow=*"SOMETIMES"* minWidth=*"10.0"* prefWidth=*"100.0"* />

</columnConstraints>

<rowConstraints>

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

<RowConstraints minHeight=*"10.0"* prefHeight=*"30.0"* vgrow=*"SOMETIMES"* />

</rowConstraints>

<children>

<Label text=*"Name"* />

<Label text=*"Arriving time"* GridPane.rowIndex=*"1"* />

<Label text=*"Departure time"* GridPane.rowIndex=*"2"* />

<Label text=*"Free places"* GridPane.rowIndex=*"3"* />

<Label text=*"Day"* GridPane.rowIndex=*"4"* />

<Label text=*"Number"* GridPane.rowIndex=*"5"* />

<TextField fx:id=*"nameField"* GridPane.columnIndex=*"1"* />

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

<TextField fx:id=*"chasVdField"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"2"* />

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

<TextField fx:id=*"dayField"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"4"* />

<TextField fx:id=*"nomerField"* GridPane.columnIndex=*"1"* GridPane.rowIndex=*"5"* />

</children>

</GridPane>

<ButtonBar layoutX=*"186.0"* layoutY=*"195.0"* prefHeight=*"40.0"* prefWidth=*"200.0"* AnchorPane.bottomAnchor=*"15.0"* AnchorPane.leftAnchor=*"186.0"* AnchorPane.rightAnchor=*"14.0"* AnchorPane.topAnchor=*"195.0"*>

<buttons>

<Button mnemonicParsing=*"false"* onAction=*"#handleOk"* text=*"Ok"* />

<Button mnemonicParsing=*"false"* onAction=*"#handleCancel"* text=*"Cancel"* />

</buttons>

</ButtonBar>

</children>

</AnchorPane>

Висновок: Я набув навичок використання засобів клієнтських технологій (Client Technologies) платформи Java SE.