Звіт

Автор: Пумня О., КІТ118Б

Дата: 02.03.2020

Лабораторна робота №16

РОЗРОБКА ГРАФІЧНОГО ІНТЕРФЕЙСУ КОРИСТУВАЧА

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

Вимоги:

Розробити графічний інтерфейс користувача для програми рішення попередньої лабораторної роботи з використанням засобів JavaFX.

ОПИС ПРОГРАМИ

Ієрархія та структура класів

Pumnya16 – Точка входу в програму

EventSheduler – Клас, що реалізує захід

Util – Утилітарний клас

FXMLController – Абстрактний клас-контролер

GreetingPageController – Клас-контролер сторінки-вітання

MainPageController – Клас контролер головної сторінки

AboutPageController – Клас контролер сторінки «About»

ТЕКСТ ПРОГРАМИ

Текст файлу Pumnya16.java

```
package labs.pumnya16;

import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.stage.Modality;
import javafx.stage.Stage;
import labs.pumnya16.Model.EventScheduler;
import labs.pumnya16.Util.Util;
import labs.pumnya16.View.AboutPageController;
import labs.pumnya16.View.GreetingPageController;
import labs.pumnya16.View.MainPageController;
import java.io.IOException;
import java.util.ArrayList;

public class Pumnya16 extends Application {

    private Stage primaryStage;
    private ObservableList<EventScheduler> eventData = FXCollections.observableArrayList();
    public ObservableList<EventScheduler> getEventData() {
        return this.eventData;
    }

    public void setEventData(ArrayList<EventScheduler> list) {
        this.eventData.clear();
        this.eventData.addAll(list);
    }

    public Stage getPrimaryStage() {
        return this.primaryStage;
    }

    public Pumnya16() {
        eventData.add(new EventScheduler("01.01.1111", "01:01", 1, "Первое место.", "Первое
описание!", "А. Бук"));
        eventData.add(new EventScheduler("02.02.2222", "02:02", 2, "Второе место!", "Второе
описание?", "Б. Вук"));
        eventData.add(new EventScheduler("03.03.3333", "03:03", 3, "Третье место?", "Третье
описание.", "В. Гук"));
        Util.setMain(this);
    }

    @Override
    public void start(Stage stage) throws Exception {
        primaryStage = stage;
        primaryStage.setTitle("Event Scheduler");
        primaryStage.setResizable(false);
        primaryStage.getIcons().add(new
Image(Pumnya16.class.getResourceAsStream("Content/EventScheduler.png")));
        initGreetingPage();
    }

    public static void main(String[] args) {
        launch();
    }

    public void initGreetingPage() {
        try {
            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(Pumnya16.class.getResource("View/GreetingPage.fxml"));
            primaryStage.setScene(new Scene(loader.load()));
            GreetingPageController controller = loader.getController();
            controller.setMain(this);
            primaryStage.show();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void showMainPage() {
        try {
            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(Pumnya16.class.getResource("View/MainPage.fxml"));
            primaryStage.setScene(new Scene(loader.load()));
            MainPageController controller = loader.getController();
            controller.setMain(this);
            controller.setTableItems(this.getEventData());
            primaryStage.show();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void showAboutPage() {
        try {
            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(Pumnya16.class.getResource("View/AboutPage.fxml"));
            Stage dialogStage = new Stage();
            dialogStage.setTitle("About");
            dialogStage.setScene(new Scene(loader.load()));
            dialogStage.initModality(Modality.WINDOW_MODAL);
            dialogStage.initOwner(primaryStage);
            dialogStage.setResizable(false);
            dialogStage.getIcons().add(new
Image(Pumnya16.class.getResourceAsStream("Content/about.png")));
            AboutPageController controller = loader.getController();
            controller.setMain(this);
            controller.setDialogStage(dialogStage);

            dialogStage.show();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Текст файлу EventScheduler.java

```
package labs.pumnya16.Model;

import javafx.beans.property.*;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class EventScheduler implements Serializable {

    private static final long serialVersionUID = -4232419163919693876L;

    private transient StringProperty date;
    private transient StringProperty time;
    private transient StringProperty venue;
    private transient StringProperty description;
    private transient FloatProperty duration;
    private transient StringProperty organizer;

    public EventScheduler(String date, String time, float dur, String venue, String descr,
String organizer) {
        this.date = new SimpleStringProperty(date);
        this.time = new SimpleStringProperty(time);
        this.venue = new SimpleStringProperty(venue);
        this.description = new SimpleStringProperty(descr);
        this.organizer = new SimpleStringProperty(organizer);
        this.duration = new SimpleFloatProperty(dur);
    }

    public EventScheduler() {
    }

    public final StringProperty getDateProperty() {
        return this.date;
    }

    public final String getDate() {
        return this.date.get();
    }

    public final void setDate(final String date) {
        if (checkDate(date)) {
            this.date.setValue(date);
        }
    }
```

```
public final StringProperty getTimeProperty() {
    return this.time;
}

public final String getTime() {
    return time.get();
}

public final void setTime(final String time) {
    if (checkTime(time)) {
        this.time.setValue(time);
    }
}

public final StringProperty getVenueProperty() {
    return this.venue;
}

public final String getVenue() {
    return this.venue.get();
}

public final void setVenue(final String venue) {
    if (checkVenue(venue)) {
        this.venue.setValue(venue);
    }
}

public final StringProperty getDescriptionProperty() {
    return this.description;
}

public final String getDescription() {
    return this.description.get();
}

public final void setDescription(final String descr) {
    if (checkDescription(descr)) {
        this.description.setValue(descr);
    }
}

public final FloatProperty getDurationProperty() {
    return this.duration;
}

public final float getDuration() {
    return this.duration.get();
}

public final void setDuration(final Float duration) {
    if (checkDuration(duration)) {
        this.duration.setValue(duration);
    }
}

public final StringProperty getOrganizerProperty() {
    return this.organizer;
}

public final String getOrganizer() {
    return this.organizer.get();
}

public final void setOrganizer(final String organizer) {
    if (checkName(organizer)) {
        this.organizer.setValue(organizer);
    }
```

```java
}
private void writeObject(ObjectOutputStream s) throws IOException {
    s.defaultWriteObject();
    s.writeUTF(getDate());
    s.writeUTF(getTime());
    s.writeUTF(getDescription());
    s.writeUTF(getOrganizer());
    s.writeUTF(getVenue());
    s.writeFloat(getDuration());
}

private void readObject(ObjectInputStream s) throws IOException, ClassNotFoundException {
    date = new SimpleStringProperty(s.readUTF());
    time = new SimpleStringProperty(s.readUTF());
    description = new SimpleStringProperty(s.readUTF());
    organizer = new SimpleStringProperty(s.readUTF());
    venue = new SimpleStringProperty(s.readUTF());
    duration = new SimpleFloatProperty(s.readFloat());
}

public static boolean checkDate(final String d) {
    if (d == null) {
        return false;
    }
    final String df = "dd.MM.yyyy";
    try {
        SimpleDateFormat sdf = new SimpleDateFormat(df);
        sdf.setLenient(false);
        sdf.parse(d);
        return true;
    } catch (ParseException e) {
        return false;
    }
}

public static boolean checkTime(final String time) {
    if (time == null) {
        return false;
    }
    final String tf = "HH:mm";
    try {
        SimpleDateFormat sdf = new SimpleDateFormat(tf);
        sdf.setLenient(false);
        sdf.parse(time);
        return true;
    } catch (ParseException e) {
        return false;
    }
}

public static boolean checkVenue(final String venue) {
    if (venue == null) {
        return false;
    }
    Pattern pattern = Pattern.compile(
            "^[А-Я][а-я]+(\\s[\"<>№]?[А-Я]*[а-я]+[\"<>№]?)*");
    Matcher matcher = pattern.matcher(venue);
    return matcher.matches();
}

public static boolean checkName(final String name) {
    if (name == null) {
        return false;
    }
    Pattern pattern = Pattern.compile("^[А-Я а-я ,.'-]+$");
    Matcher matcher = pattern.matcher(name);
    return matcher.matches();
}
```

```
    public static boolean checkDescription(final String desc) {
        if (desc == null) {
            return false;
        }
        Pattern pattern = Pattern.compile(
                "^[А-Я][а-я]+(\\s[А-Я]*[а-я]+)*");
        Matcher matcher = pattern.matcher(desc);
        return matcher.matches();
    }

    public static boolean checkDuration(final Float dur) {
        if(dur <= 0) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        StringBuilder builder = new StringBuilder();
        builder.append("Дата: ").append(this.getDate()).append("\n");
        builder.append("Время начала: ").append(
                this.getTime()).append("\n");
        builder.append("Длительность (часы): ").append(
                this.getDuration()).append("\n");
        builder.append("Место проведения: ").append(
                this.getVenue()).append("\n");
        builder.append("Описание: ").append(
                this.getDescription()).append("\n");
        builder.append("Организатор: ").append(
                this.getOrganizer()).append("\n");
        return builder.toString();
    }
}
```

Текст файлу Util.java

```
package labs.pumnya16.Util;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.stage.FileChooser;
import labs.pumnya16.Model.EventScheduler;
import labs.pumnya16.Pumnya16;
import java.io.*;
import java.util.ArrayList;

public class Util {
    private Util() {
        // empty body
    }

    private static Pumnya16 main;
    public static void setMain(Pumnya16 m) {
        main = m;
    }

    public static void saveToFile() {
        FileChooser fileChooser = new FileChooser();
        fileChooser.setTitle("Save Resource File");
        fileChooser.getExtensionFilters().add(
                new FileChooser.ExtensionFilter("Event files", "*.event"));
        fileChooser.setInitialDirectory(new File("D:/Programming Projects/Java
Projects/pumnya-alex"));
        File file = fileChooser.showSaveDialog(main.getPrimaryStage());
        if (file != null) {
            try {
                ObjectOutputStream oos = new ObjectOutputStream(
                        new FileOutputStream(file.getName()));
                oos.writeObject(new ArrayList<>(main.getEventData()));
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public static void loadFromFile() {
        FileChooser fileChooser = new FileChooser();
        fileChooser.setTitle("Open Resource File");
        fileChooser.getExtensionFilters().add(
                new FileChooser.ExtensionFilter("Event files", "*.event"));
        fileChooser.setInitialDirectory(new File("D:/Programming Projects/Java
Projects/pumnya-alex"));
        File file = fileChooser.showOpenDialog(main.getPrimaryStage());
        if (file != null) {
            try {
                ObjectInputStream ois = new ObjectInputStream(
                        new FileInputStream(file.getName()));
                main.setEventData((ArrayList<EventScheduler>) ois.readObject());
            } catch (IOException | ClassNotFoundException e) {
                e.printStackTrace();
            }
        }
    }

    public static ObservableList<EventScheduler> search(String str) {
        ObservableList<EventScheduler> temp = FXCollections.observableArrayList();
        for(int i = 0; i < main.getEventData().size(); i++) {
            if(searchInItem(main.getEventData().get(i), str)) {
                temp.add(main.getEventData().get(i));
            }
        }
        return temp;
    }

    private static boolean searchInItem(EventScheduler item, String str) {
        if(item.getDate().contains(str)) {
            return true;
        }
        if(item.getTime().contains(str)) {
            return true;
        }
        if(item.getVenue().contains(str)) {
            return true;
        }
        if(item.getDescription().contains(str)) {
            return true;
        }
        if(item.getOrganizer().contains(str)) {
            return true;
        }
        if(String.valueOf(item.getDuration()).contains(str)) {
            return true;
        }
        return false;
    }
}
```

Текст файлу AboutPageController.java

```
package labs.pumnya16.View;

import javafx.fxml.FXML;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import java.awt.*;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;

public class AboutPageController extends FXMLController{
    private Stage dialogStage;

    public void setDialogStage(Stage stage) {
        this.dialogStage = stage;
    }
    @FXML
    public void handleClose() {
        this.dialogStage.close();
    }
    @FXML
    public void handleDiscord() {
        try {
            Desktop.getDesktop().browse(new
URI("https://discord.com/users/527049968176070656"));
        } catch (IOException | URISyntaxException e1) {
            e1.printStackTrace();
        }
    }
    @FXML
    public void handleTelegram() {
        try {
            Desktop.getDesktop().browse(new URI("https://t.me/SiriusQuase"));
        } catch (IOException | URISyntaxException e1) {
            e1.printStackTrace();
        }
    }
    @FXML
    public void handleSkype() {
        try {
            Desktop.getDesktop().browse(new
URI("https://join.skype.com/invite/i6li7fhK7Yqw"));
        } catch (IOException | URISyntaxException e1) {
            e1.printStackTrace();
        }
    }
    @FXML
    public void handleYoutube() {
        try {
            Desktop.getDesktop().browse(new
URI("https://www.youtube.com/channel/UCfrAsAoknyVJtOuWge0ojIg"));
        } catch (IOException | URISyntaxException e1) {
            e1.printStackTrace();
        }
    }
    @FXML
    public void handleExit(KeyEvent keyEvent) {
        if(keyEvent.getCode() == KeyCode.ESCAPE) {
            this.dialogStage.close();
        }
    }
}
```

Текст файлу AboutPage.fxml

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

<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.image.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>

<AnchorPane fx:id="anchor_pane" onKeyReleased="#handleExit" prefHeight="323.0"
prefWidth="640.0" stylesheets="@../Content/sceneStyle.css"
```

```
xmlns="http://javafx.com/javafx/10.0.2-internal" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="labs.pumnya16.View.AboutPageController">
   <children>
      <ImageView fitHeight="301.0" fitWidth="339.0" layoutX="14.0" layoutY="14.0"
pickOnBounds="true" preserveRatio="true">
         <image>
            <Image url="@../Content/about_photo.jpg" />
         </image>
      </ImageView>
      <ImageView fitHeight="56.0" fitWidth="62.0" layoutX="344.0" layoutY="262.0"
onMouseClicked="#handleDiscord" pickOnBounds="true" preserveRatio="true">
         <image>
            <Image url="@../Content/discord.png" />
         </image>
         <cursor>
            <Cursor fx:constant="HAND" />
         </cursor></ImageView>
      <ImageView fitHeight="56.0" fitWidth="69.0" layoutX="414.0" layoutY="262.0"
onMouseClicked="#handleTelegram" pickOnBounds="true" preserveRatio="true">
         <image>
            <Image url="@../Content/telegram.png" />
         </image>
         <cursor>
            <Cursor fx:constant="HAND" />
         </cursor>
      </ImageView>
      <ImageView fitHeight="56.0" fitWidth="63.0" layoutX="491.0" layoutY="262.0"
onMouseClicked="#handleSkype" pickOnBounds="true" preserveRatio="true">
         <image>
            <Image url="@../Content/skype.png" />
         </image>
         <cursor>
            <Cursor fx:constant="HAND" />
         </cursor>
      </ImageView>
      <ImageView fitHeight="56.0" fitWidth="74.0" layoutX="562.0" layoutY="262.0"
onMouseClicked="#handleYoutube" pickOnBounds="true" preserveRatio="true">
         <image>
            <Image url="@../Content/youtube.png" />
         </image>
         <cursor>
            <Cursor fx:constant="HAND" />
         </cursor>
      </ImageView>
            <Label layoutX="325.0" layoutY="14.0" prefHeight="63.0" prefWidth="301.0"
text="This program was created using Java 13 &amp; JavaFX 14" textAlignment="CENTER"
wrapText="true">
         <font>
            <Font name="Times New Roman Bold" size="24.0" />
         </font>
      </Label>
      <Label layoutX="344.0" layoutY="144.0" text="Platform: Windows">
         <font>
            <Font name="Times New Roman Bold" size="18.0" />
         </font>
      </Label>
      <Label layoutX="344.0" layoutY="175.5" text="Version: 1.0">
         <font>
            <Font name="Times New Roman Bold" size="18.0" />
         </font>
      </Label>
      <Label layoutX="344.0" layoutY="207.0" text="Author: Pumnya Alex">
         <font>
            <Font name="Times New Roman Bold" size="18.0" />
         </font>
      </Label>
      <Label layoutX="422.0" layoutY="234.0" text="Contact me">
         <font>
            <Font name="Times New Roman Bold" size="24.0" />
         </font>
      </Label>
      <Button layoutX="427.0" layoutY="87.0" mnemonicParsing="false" onAction="#handleClose"
prefHeight="38.0" prefWidth="85.0" text="Close">
         <font>
            <Font name="Comic Sans MS Bold" size="18.0" />
         </font>
      </Button>
   </children>
</AnchorPane>
```

Текст файлу GreetingPageController.java

```
package labs.pumnya16.View;

import javafx.fxml.FXML;
import javafx.scene.control.Button;

public class GreetingPageController extends FXMLController {
    @FXML
    private Button startButton;

    @FXML
    private void handleStart() {
        getMain().showMainPage();
    }
}
```

Текст файлу GreetingPage.fxml

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

<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>

<AnchorPane fx:id="anchor_pane" prefHeight="500" prefWidth="800"
stylesheets="@../Content/sceneStyle.css" xmlns="http://javafx.com/javafx/10.0.2-internal"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="labs.pumnya16.View.GreetingPageController">
   <children>
      <Label ellipsisString="" layoutX="66.0" layoutY="70.0" text="Welcome to the Event
Scheduler">
         <font>
            <Font name="Times New Roman Bold" size="48.0" />
         </font>
      </Label>
      <Button fx:id="startButton" layoutX="342.0" layoutY="397.0" mnemonicParsing="false"
onAction="#handleStart" text="START">
         <font>
            <Font name="Comic Sans MS" size="24.0" />
         </font>
      </Button>
      <Label layoutX="103.0" layoutY="136.0" prefHeight="94.0" prefWidth="593.0" text="Here
you can manipulate your events by the following actions:" textAlignment="CENTER"
wrapText="true">
         <font>
            <Font name="Times New Roman Bold" size="36.0" />
         </font>
      </Label>
      <Label layoutX="103.0" layoutY="249.0" text="○ Create new events">
         <font>
            <Font name="Times New Roman Bold" size="28.0" />
         </font>
      </Label>
      <Label layoutX="103.0" layoutY="297.0" text="○ Delete events">
         <font>
            <Font name="Times New Roman Bold" size="28.0" />
         </font>
      </Label>
      <Label layoutX="470.0" layoutY="297.0" text="○ View events data">
```

```
         <font>
            <Font name="Times New Roman Bold" size="28.0" />
         </font>
      </Label>
      <Label layoutX="470.0" layoutY="249.0" text="○ Sort events">
         <font>
            <Font name="Times New Roman Bold" size="28.0" />
         </font>
      </Label>
      <Label layoutX="329.0" layoutY="340.0" text="And more...">
         <font>
            <Font name="Times New Roman Bold" size="28.0" />
         </font>
      </Label>
   </children>
</AnchorPane>
```

Текст файлу MainPageController.java

```
package labs.pumnya16.View;

import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.scene.control.Alert;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.control.cell.TextFieldTableCell;
import javafx.scene.input.MouseButton;
import javafx.scene.input.MouseEvent;
import javafx.stage.Modality;
import javafx.util.StringConverter;
import labs.pumnya16.Model.EventScheduler;
import labs.pumnya16.Util.Util;

public class MainPageController extends FXMLController {
    @FXML
    public TextField searchField;
    @FXML
    private TextField dateField;
    @FXML
    private TextField timeField;
    @FXML
    private TextField durationField;
    @FXML
    private TextField venueField;
    @FXML
    private TextField descrField;
    @FXML
    private TextField organizerField;
    @FXML
    private TableView<EventScheduler> tableView;
    @FXML
    private TableColumn<EventScheduler, String> dateCol;
    @FXML
    private TableColumn<EventScheduler, String> timeCol;
    @FXML
    private TableColumn<EventScheduler, Float> durationCol;
    @FXML
    private TableColumn<EventScheduler, String> venueCol;
    @FXML
    private TableColumn<EventScheduler, String> descrCol;
    @FXML
    private TableColumn<EventScheduler, String> organizerCol;

    private EventScheduler selectedColumn;

    @FXML
    public void handleOpen() {
        Util.loadFromFile();
        setTableItems(getMain().getEventData());
    }
    @FXML
    public void handleSave() {
        Util.saveToFile();
    }
    @FXML
    public void handleExit() {
        getMain().getPrimaryStage().close();
    }
    @FXML
    public void handleExitButton() {
        getMain().getPrimaryStage().close();
    }
    @FXML
    public void handleAbout() {
        getMain().showAboutPage();
    }
    @FXML
    public void handleAdd() {
        if(isInputValid()) {
            getMain().getEventData().add(new EventScheduler(
                    dateField.getText(), timeField.getText(),
Float.parseFloat(durationField.getText()),
                    venueField.getText(), descrField.getText(), organizerField.getText()));
        }
        dateField.clear();
        timeField.clear();
        venueField.clear();
        descrField.clear();
        durationField.clear();
        organizerField.clear();
        setTableItems(getMain().getEventData());
    }
    @FXML
    public void handleClear() {
        getMain().getEventData().clear();
        setTableItems(getMain().getEventData());
    }
    @FXML
    public void handleDelete() {
        getMain().getEventData().remove(getSelectedColumn());
        setTableItems(getMain().getEventData());
    }
    @FXML
    public void handleTextInput() {
        if(searchField.getText().length() == 0) {
            setTableItems(getMain().getEventData());
        } else {
            ObservableList<EventScheduler> temp = Util.search(searchField.getText());
            setTableItems(temp);
        }
    }
    @FXML
    public void handleSaveButton() {
        Util.saveToFile();
    }
    @FXML
    public void handleLoadButton() {
        Util.loadFromFile();
        setTableItems(getMain().getEventData());
    }
    @FXML
    public void handleDateChange(TableColumn.CellEditEvent<EventScheduler, String> t) {

t.getTableView().getItems().get(t.getTablePosition().getRow()).setDate(t.getNewValue());
    }
    @FXML
    public void handleTimeChange(TableColumn.CellEditEvent<EventScheduler, String> t) {

t.getTableView().getItems().get(t.getTablePosition().getRow()).setTime(t.getNewValue());
    }
    @FXML
    public void handleVenueChange(TableColumn.CellEditEvent<EventScheduler, String> t) {

t.getTableView().getItems().get(t.getTablePosition().getRow()).setVenue(t.getNewValue());
    }
    @FXML
    public void handleDescrChange(TableColumn.CellEditEvent<EventScheduler, String> t) {

t.getTableView().getItems().get(t.getTablePosition().getRow()).setDescription(t.getNewValue())
;
    }
    @FXML
    public void handleOrganizerChange(TableColumn.CellEditEvent<EventScheduler, String> t) {

t.getTableView().getItems().get(t.getTablePosition().getRow()).setOrganizer(t.getNewValue());
    }
    @FXML
    public void handleDurationChange(TableColumn.CellEditEvent<EventScheduler, Float> t) {

t.getTableView().getItems().get(t.getTablePosition().getRow()).setDuration(t.getNewValue());
    }

    @FXML
    void initialize() {
        dateCol.setCellValueFactory(new PropertyValueFactory<>("date"));
        dateCol.setCellFactory(TextFieldTableCell.forTableColumn());
        timeCol.setCellValueFactory(new PropertyValueFactory<>("time"));
        timeCol.setCellFactory(TextFieldTableCell.forTableColumn());
        durationCol.setCellValueFactory(new PropertyValueFactory<>("duration"));
        durationCol.setCellFactory(TextFieldTableCell.forTableColumn(new StringConverter<>() {
            @Override
            public String toString(Float value) {
                return Float.toString(value);
            }
            @Override
            public Float fromString(String s) {
                return Float.parseFloat(s);
            }
        }));
        venueCol.setCellValueFactory(new PropertyValueFactory<>("venue"));
        venueCol.setCellFactory(TextFieldTableCell.forTableColumn());
        descrCol.setCellValueFactory(new PropertyValueFactory<>("description"));
        descrCol.setCellFactory(TextFieldTableCell.forTableColumn());
        organizerCol.setCellValueFactory(new PropertyValueFactory<>("organizer"));
        organizerCol.setCellFactory(TextFieldTableCell.forTableColumn());
        tableView.setOnMouseClicked((MouseEvent event) -> {
            if(event.getButton().equals(MouseButton.PRIMARY)) {
                setSelectedColumn(tableView.getSelectionModel().getSelectedItem());
            }
        });
    }
    public void setTableItems(ObservableList<EventScheduler> data) {
        tableView.setItems(data);
    }
    private void setSelectedColumn(EventScheduler event) {
        selectedColumn = event;
    }
    private EventScheduler getSelectedColumn() {
        return selectedColumn;
    }
    private boolean isInputValid() {
        StringBuilder error = new StringBuilder();
        if (!EventScheduler.checkDate(dateField.getText())) {
            error.append("\"Date\" ");
        }
        if (!EventScheduler.checkTime(timeField.getText())) {
            error.append("\"Time\" ");
        }
        if (!EventScheduler.checkVenue(venueField.getText())) {
            error.append("\"Venue\" ");
        }
        if (!EventScheduler.checkDescription(descrField.getText())) {
            error.append("\"Description\" ");
        }
        if (!EventScheduler.checkName(organizerField.getText())) {
            error.append("\"Organizer\" ");
        }
        if (durationField.getText().equals("") ||
                !EventScheduler.checkDuration(Float.parseFloat(durationField.getText()))) {
            error.append("\"Duration\" ");
        }
        if (error.length() == 0) {
            return true;
        } else {
            error.append("- поле(я) с ошибками");

            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setHeight(600);
            alert.setWidth(800);
            alert.initOwner(getMain().getPrimaryStage());
            alert.initModality(Modality.WINDOW_MODAL);
            alert.setTitle("Invalid Fields");
            alert.setHeaderText("Некоторым полям требуются правки!");
            alert.setContentText(error.toString());
            alert.showAndWait();

            return false;
        }
    }
}
```

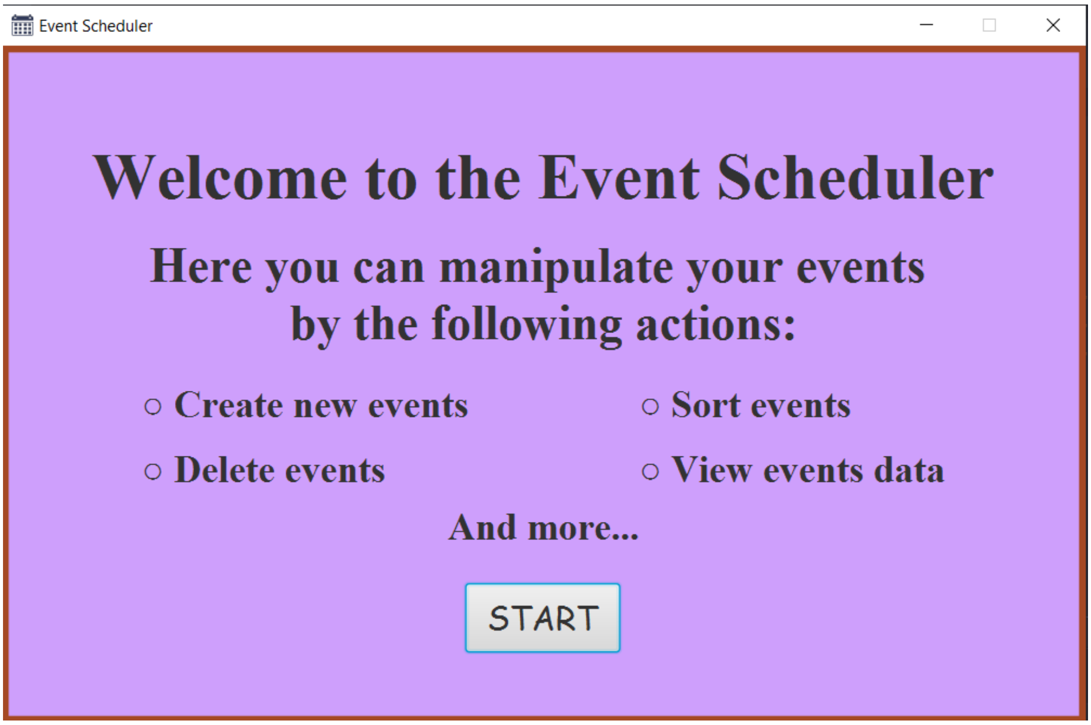
Текст файлу MainPage.fxml

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

<?import javafx.scene.control.*?>
<?import javafx.scene.image.*?>
<?import javafx.scene.input.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>

<AnchorPane prefHeight="500.0" prefWidth="800.0" xmlns="http://javafx.com/javafx/10.0.2-
internal" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="labs.pumnya16.View.MainPageController">
   <children>
      <AnchorPane fx:id="anchor_pane" layoutY="26.0" prefHeight="474.0" prefWidth="800.0"
stylesheets="@../Content/sceneStyle.css">
         <children>
            <TableView fx:id="tableView" editable="true" layoutX="10.0" layoutY="59.0"
prefHeight="356.0" prefWidth="698.0">
              <columns>
                <TableColumn fx:id="dateCol" onEditCommit="#handleDateChange" prefWidth="84.0"
style="-fx-alignment: center;" text="Date" />
                <TableColumn fx:id="timeCol" onEditCommit="#handleTimeChange" prefWidth="78.0"
style="-fx-alignment: center;" text="Time" />
                <TableColumn fx:id="durationCol" onEditCommit="#handleDurationChange"
prefWidth="56.0" style="-fx-alignment: center;" text="Duration" />
                <TableColumn fx:id="venueCol" onEditCommit="#handleVenueChange"
prefWidth="144.59999084472656" style="-fx-alignment: center;" text="Venue" />
                <TableColumn fx:id="descrCol" onEditCommit="#handleDescrChange"
prefWidth="222.4000244140625" style="-fx-alignment: center;" text="Description" />
                <TableColumn fx:id="organizerCol" onEditCommit="#handleOrganizerChange"
prefWidth="111.19998779296873" style="-fx-alignment: center;" text="Organizer" />
```

```
              </columns>
            </TableView>
             <Label layoutX="86.0" layoutY="14.0" text="Events Book">
                 <font>
                     <Font name="Times New Roman Bold" size="36.0" />
                 </font>
             </Label>
            <TextField fx:id="dateField" layoutX="12.0" layoutY="427.0" prefHeight="26.0"
prefWidth="73.0" promptText="Date" />
            <TextField fx:id="timeField" layoutX="90.0" layoutY="427.0" prefHeight="26.0"
prefWidth="73.0" promptText="Time" />
            <TextField fx:id="durationField" layoutX="168.0" layoutY="427.0" prefHeight="26.0"
prefWidth="62.0" promptText="Duration" />
            <TextField fx:id="venueField" layoutX="235.0" layoutY="427.0" prefHeight="26.0"
prefWidth="131.0" promptText="Venue" />
            <TextField fx:id="descrField" layoutX="371.0" layoutY="427.0" prefHeight="26.0"
prefWidth="170.0" promptText="Description" />
            <TextField fx:id="organizerField" layoutX="546.0" layoutY="427.0"
prefHeight="26.0" prefWidth="96.0" promptText="Organizer" />
             <TextField fx:id="searchField" layoutX="555.0" layoutY="21.0"
onKeyReleased="#handleTextInput" promptText="Search field" />
             <Label layoutX="456.0" layoutY="16.0" text="Search:">
                 <font>
                     <Font name="Comic Sans MS Bold" size="24.0" />
                 </font>
             </Label>
             <ImageView fitHeight="53.0" fitWidth="58.0" layoutX="22.0" layoutY="8.0"
pickOnBounds="true" preserveRatio="true">
                 <image>
                     <Image url="@../Content/book.png" />
                 </image>
             </ImageView>
             <Button layoutX="646.0" layoutY="427.0" mnemonicParsing="false"
onAction="#handleAdd" prefHeight="26.0" prefWidth="58.0" text="Add">
               <font>
                  <Font name="Comic Sans MS Bold" size="12.0" />
               </font></Button>
            <Button layoutX="722.0" layoutY="83.0" mnemonicParsing="false"
onAction="#handleClear" prefHeight="40.0" prefWidth="62.0" text="Clear">
               <font>
                  <Font name="Comic Sans MS Bold" size="14.0" />
               </font>
            </Button>
            <Button layoutX="719.0" layoutY="158.0" mnemonicParsing="false"
onAction="#handleDelete" prefHeight="40.0" prefWidth="68.0" text="Delete">
               <font>
                  <Font name="Comic Sans MS Bold" size="14.0" />
               </font>
            </Button>
            <Button layoutX="719.0" layoutY="355.0" mnemonicParsing="false"
onAction="#handleExitButton" prefHeight="40.0" prefWidth="68.0" text="Exit">
               <font>
                  <Font name="Comic Sans MS Bold" size="18.0" />
               </font>
            </Button>
            <Button layoutX="720.0" layoutY="217.0" mnemonicParsing="false"
onAction="#handleSaveButton" prefHeight="40.0" prefWidth="68.0" text="Save">
               <font>
                  <Font name="Comic Sans MS Bold" size="18.0" />
               </font>
            </Button>
            <Button layoutX="720.0" layoutY="275.0" mnemonicParsing="false"
onAction="#handleLoadButton" prefHeight="40.0" prefWidth="68.0" text="Load">
               <font>
                  <Font name="Comic Sans MS Bold" size="18.0" />
               </font>
            </Button>
            <ImageView fitHeight="53.0" fitWidth="96.0" layoutX="726.0" layoutY="8.0"
pickOnBounds="true" preserveRatio="true">
```

```
               <image>
                  <Image url="@../Content/search.png" />
               </image>
            </ImageView>
            <ImageView fitHeight="26.0" fitWidth="31.0" layoutX="754.0" layoutY="427.0"
pickOnBounds="true" preserveRatio="true">
               <image>
                  <Image url="@../Content/copyright-symbol.png" />
               </image>
            </ImageView>
         </children></AnchorPane>
      <MenuBar prefHeight="26.0" prefWidth="800.0">
        <menus>
          <Menu mnemonicParsing="false" text="File">
               <items>
                  <MenuItem mnemonicParsing="false" onAction="#handleOpen" text="Open">
                     <accelerator>
                        <KeyCodeCombination alt="UP" code="O" control="DOWN" meta="UP"
shift="UP" shortcut="UP" />
                     </accelerator>
                  </MenuItem>
                  <MenuItem mnemonicParsing="false" onAction="#handleSave" text="Save">
                     <accelerator>
                        <KeyCodeCombination alt="UP" code="S" control="DOWN" meta="UP"
shift="UP" shortcut="UP" />
                     </accelerator>
                  </MenuItem>
                  <MenuItem mnemonicParsing="false" onAction="#handleExit" text="Exit" />
               </items>
          </Menu>
          <Menu mnemonicParsing="false" text="Help">
            <items>
              <MenuItem mnemonicParsing="false" onAction="#handleAbout" text="About" />
            </items>
          </Menu>
        </menus>
      </MenuBar>
   </children>
</AnchorPane>
```

ВАРІАНТИ ВИКОРИСТАННЯ



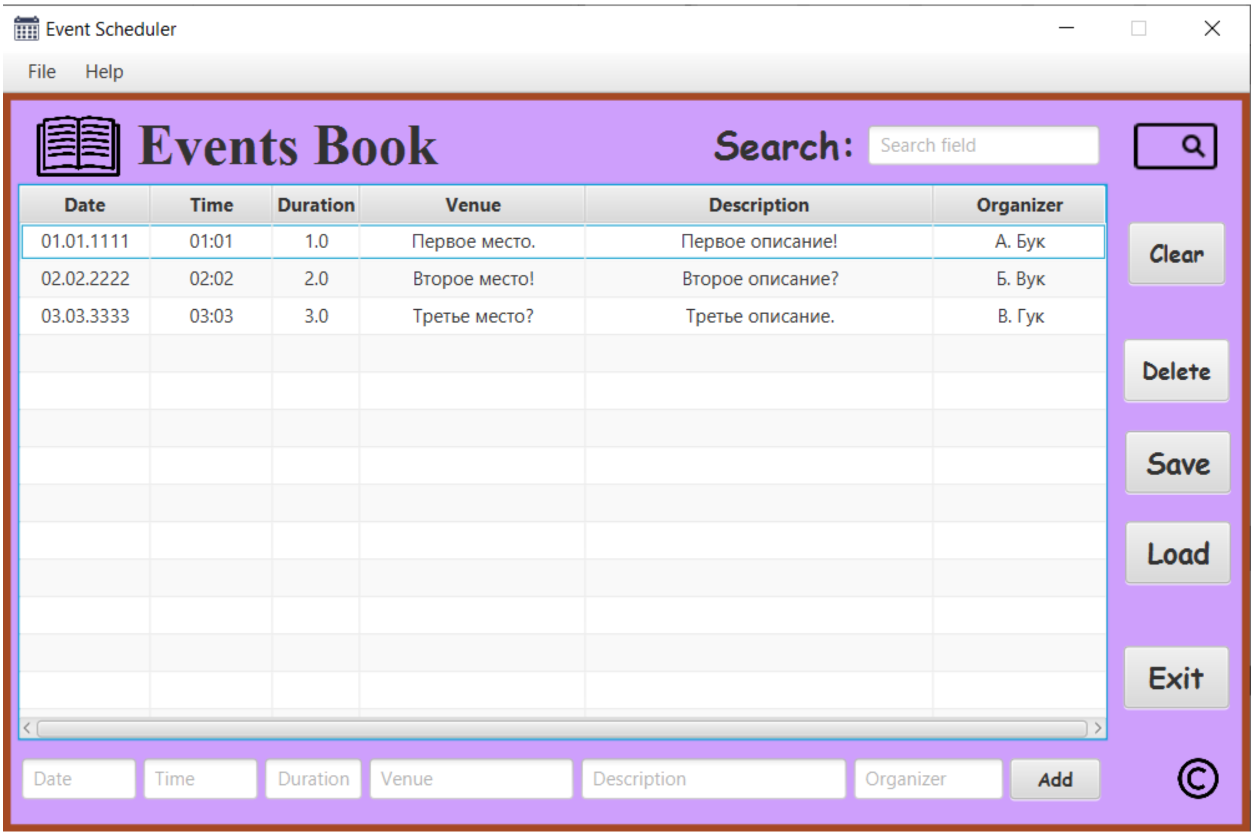
Welcome to the Event Scheduler

Here you can manipulate your events
by the following actions:

- ○ Create new events
- ○ Sort events
- ○ Delete events
- ○ View events data

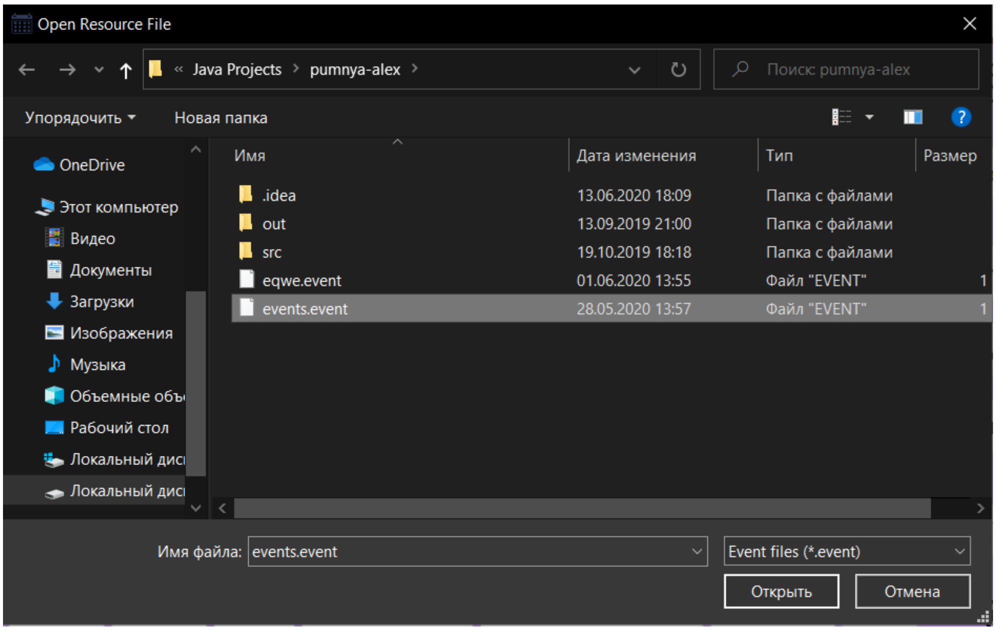
And more...

Рисунок 1 – Сторінка-вітання



Date	Time	Duration	Venue	Description	Organizer
01.01.1111	01:01	1.0	Первое место.	Первое описание!	А. Бук
02.02.2222	02:02	2.0	Второе место!	Второе описание?	Б. Вук
03.03.3333	03:03	3.0	Третье место?	Третье описание.	В. Гук

Рисунок 2 – Головна сторінка

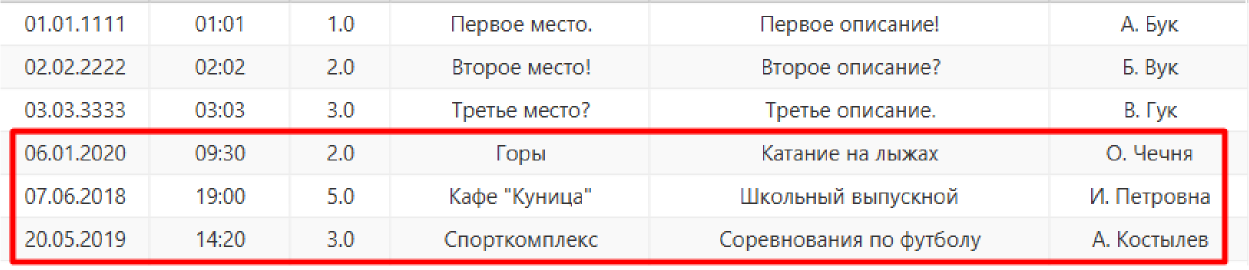


Имя	Дата изменения	Тип	Размер
.idea	13.06.2020 18:09	Папка с файлами	
out	13.09.2019 21:00	Папка с файлами	
src	19.10.2019 18:18	Папка с файлами	
eqwe.event	01.06.2020 13:55	Файл "EVENT"	1
events.event	28.05.2020 13:57	Файл "EVENT"	1

Имя файла: events.event

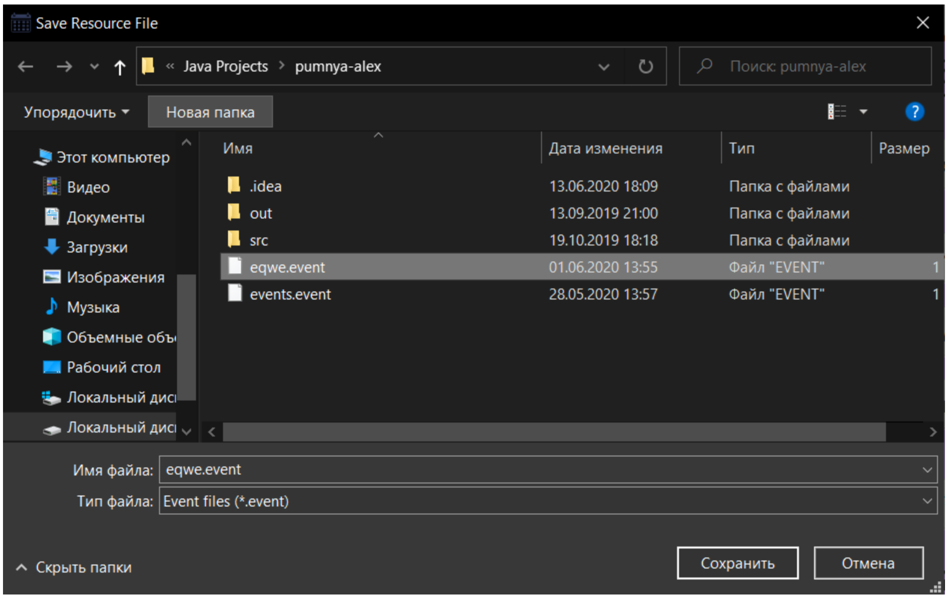
Event files (*.event)

Рисунок 3 – Вікно «Загрузити дані з файлу»



01.01.1111	01:01	1.0	Первое место.	Первое описание!	А. Бук
02.02.2222	02:02	2.0	Второе место!	Второе описание?	Б. Вук
03.03.3333	03:03	3.0	Третье место?	Третье описание.	В. Гук
06.01.2020	09:30	2.0	Горы	Катание на лыжах	О. Чечня
07.06.2018	19:00	5.0	Кафе "Куница"	Школьный выпускной	И. Петровна
20.05.2019	14:20	3.0	Спорткомплекс	Соревнования по футболу	А. Костылев

Рисунок 4 – Результат

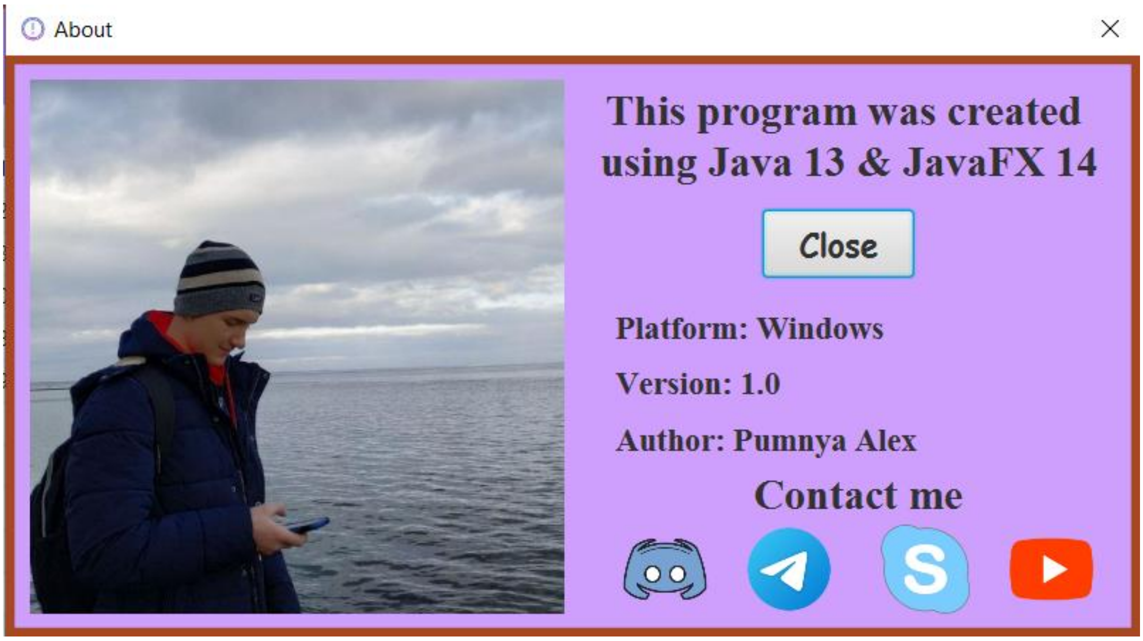


Имя	Дата изменения	Тип	Размер
.idea	13.06.2020 18:09	Папка с файлами	
out	13.09.2019 21:00	Папка с файлами	
src	19.10.2019 18:18	Папка с файлами	
eqwe.event	01.06.2020 13:55	Файл "EVENT"	1
events.event	28.05.2020 13:57	Файл "EVENT"	1

Имя файла: eqwe.event

Тип файла: Event files (*.event)

Рисунок 5 – Вікно «Зберегти у файл»



This program was created using Java 13 & JavaFX 14

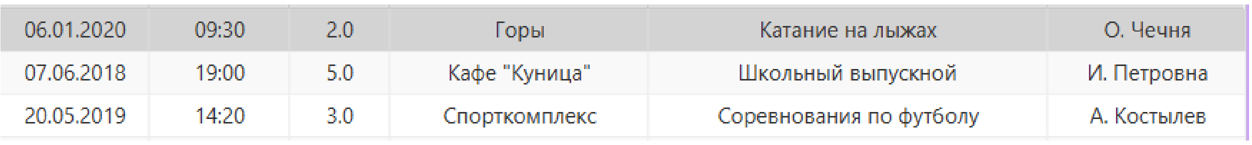
Platform: Windows

Version: 1.0

Author: Pumnya Alex

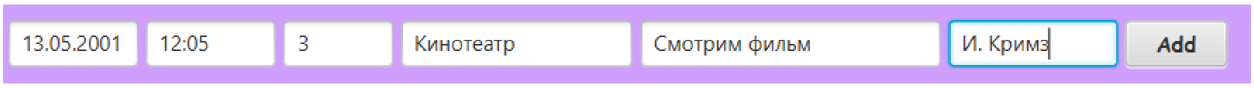
Contact me

Рисунок 6 – Сторінка «About»



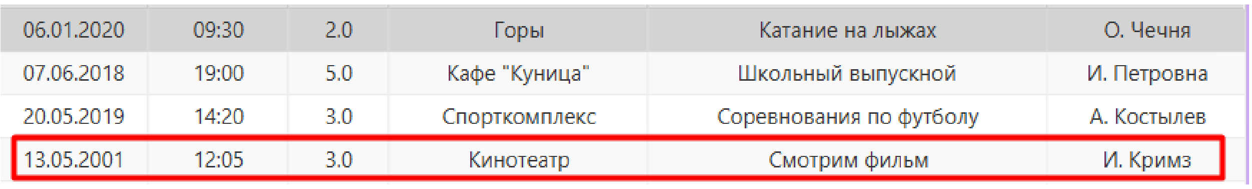
06.01.2020	09:30	2.0	Горы	Катание на лыжах	О. Чечня
07.06.2018	19:00	5.0	Кафе "Куница"	Школьный выпускной	И. Петровна
20.05.2019	14:20	3.0	Спорткомплекс	Соревнования по футболу	А. Костылев

Рисунок 7 – Результат видалення



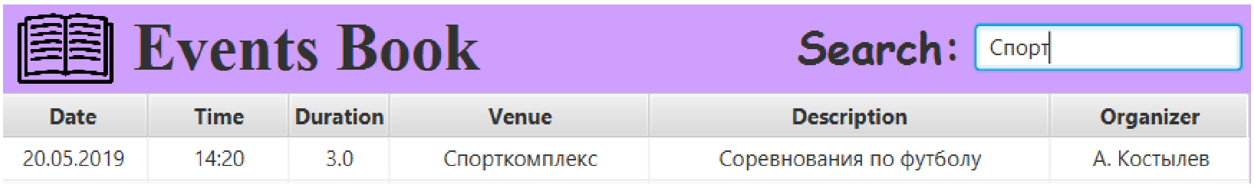
13.05.2001 | 12:05 | 3 | Кинотеатр | Смотрим фильм | И. Кримз

Рисунок 8 – Додавання заходу



06.01.2020	09:30	2.0	Горы	Катание на лыжах	О. Чечня
07.06.2018	19:00	5.0	Кафе "Куница"	Школьный выпускной	И. Петровна
20.05.2019	14:20	3.0	Спорткомплекс	Соревнования по футболу	А. Костылев
13.05.2001	12:05	3.0	Кинотеатр	Смотрим фильм	И. Кримз

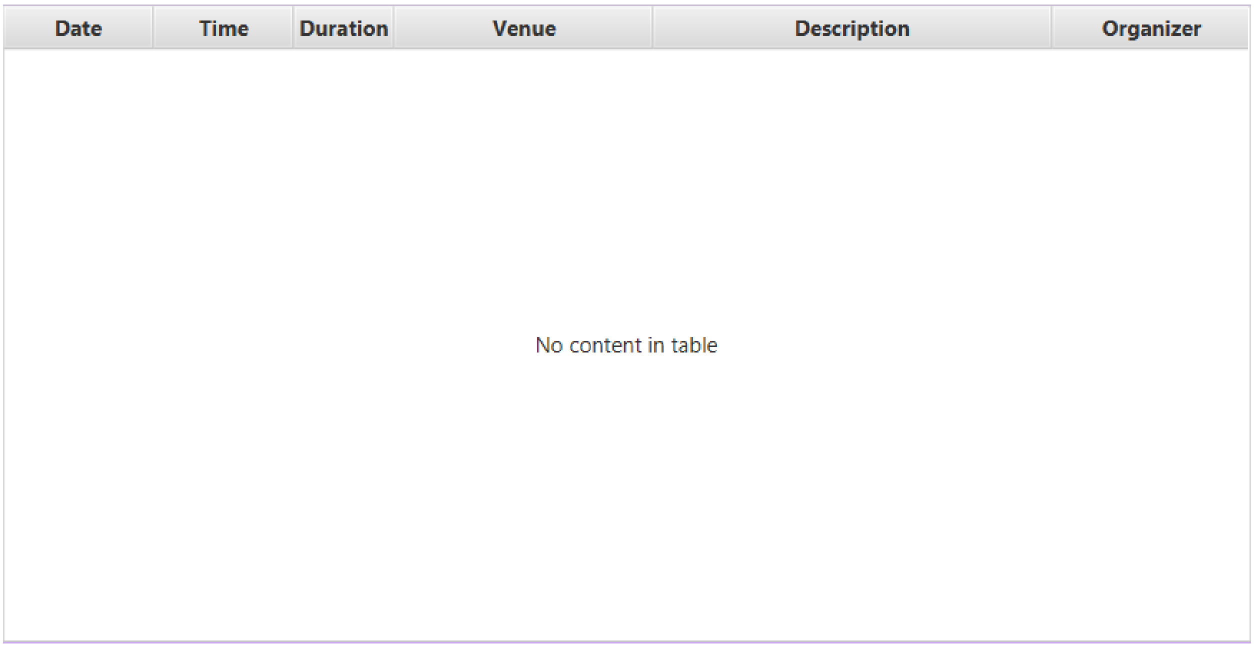
Рисунок 9 – Результат додавання



Search: Спорт

Date	Time	Duration	Venue	Description	Organizer
20.05.2019	14:20	3.0	Спорткомплекс	Соревнования по футболу	А. Костылев

Рисунок 10 – Результат пошуку



Date	Time	Duration	Venue	Description	Organizer

No content in table

Рисунок 11 – Результат «Clear»

ВИСНОВКИ

При виконанні лабораторної набуто практичних навичок роботи з JavaFX. Створено програму-планувальник заходами. Розроблений функціонал: створення заходів, видалення заходів, пошук заходів по всім характеристикам, збереження у файл, зчитування з файлу. Також розроблено додаткову сторінку «About», де коротко описано програму, та наведено інформацію про автора.