

Лабораторна робота №8.

Мета:

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

1. Вимоги

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

1.1 Розробник

Левицький Богдан, КН-108

1.2 Задача

Захід: дата, час початку і тривалість; місце проведення; опис; учасники (кількість не обмежена)

2 Опис програми

Програма містить інформацію про заходи. Можна додавати, чистити, виписувати, серіалізувати та десеріалізувати ваші заходи у файл вибраний користувачем

2.1 Засоби ООП

Були використані різні класи та методи, структури даних та модифікатори доступу також пакет JavaFX

2.2 Ієрархія та структура класів

1. Клас `MainApp` який запускає графічний інтерфейс
2. Клас `Event` domain – об'єкт, який перероблений для JavaFX
3. Fxml file `Event Overview` слугує головною функцією виклику графічного інтерфейсу та дизайном для нього
4. Клас `EventOverviewController` контроллер у якому зберігаються всі методи, слугує back – end об'єктом
5. Клас `DateUtil` для редагування дати

2.3 Важливі фрагменти програми.

MainLauncher

```
public class MainApp extends Application
{
    private Stage primaryStage;
    private BorderPane rootLayout;

    private ObservableList<Event> eventData = FXCollections.observableArrayList();

    public ObservableList<Event> getEventData()
    {
        return eventData;
    }

    @Override
    public void start(Stage primaryStage)
    {
        this.primaryStage = primaryStage;
        this.primaryStage.setTitle("Lab8");
        initRootLayout();
        showEventOverview();
    }

    public boolean showEventEditDialog(Event event) {
        try
        {
            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(MainApp.class.getResource("view/EventEditDialog.fxml"));
            AnchorPane page = (AnchorPane) loader.load();

            Stage dialogStage = new Stage();
            dialogStage.setTitle("Edit Event");
            dialogStage.initModality(Modality.WINDOW_MODAL);
            dialogStage.initOwner(primaryStage);
            Scene scene = new Scene(page);
            dialogStage.setScene(scene);

            EventEditDialogController controller = loader.getController();
            controller.setDialogStage(dialogStage);
            controller.setEvent(event);

            dialogStage.showAndWait();

            return controller.isOkClicked();
        }
        catch (IOException e)
        {
            e.printStackTrace();
            return false;
        }
    }
}
```

```

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);
        primaryStage.show();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
}

```

```

public void showEventOverview() {
    try
    {
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(MainApp.class.getResource("view/EventOverview.fxml"));
        AnchorPane eventOverview = (AnchorPane) loader.load();

        rootLayout.setCenter(eventOverview);

        EventOverviewController controller = loader.getController();
        controller.setMainApp(this);
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
}

public Stage getPrimaryStage()
{
    return primaryStage;
}

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

```

MainController

```

package ch.makery.adress.view;

import javafx.fxml.FXML;

public class EventOverviewController {

    @FXML
    private TableView<Event> eventTable;

    @FXML
    private TableColumn<Event, String> nameColumn;

    @FXML
    private Label dateLabel;

    @FXML
    private Label timeStartLabel;

    @FXML
    private Label durationLabel;

    @FXML
    private Label placeLabel;

    @FXML
    private Label descriptionLabel;

    private MainApp mainApp;

    public EventOverviewController()
    {

    }

    @FXML
    private void initialize()
    {
        nameColumn.setCellValueFactory(cellData -> cellData.getValue().getNameProperty());

        showEventDetails(null);

        eventTable.getSelectionModel().selectedItemProperty().addListener((observable, oldValue, newValue) -> showEventDetails(newValue));
    }

    public void setMainApp(MainApp mainApp)
    {
        this.mainApp = mainApp;
        eventTable.setItems(mainApp.getEventData());
    }

    private void showEventDetails(Event event)
    {
        if (event != null)
        {
            dateLabel.setText(DateUtil.format(event.getDate()));
            timeStartLabel.setText(event.getTimeStart());
            durationLabel.setText(event.getDuration());
            placeLabel.setText(event.getPlace());
            descriptionLabel.setText(event.getDescription());
        }
        else
        {
            dateLabel.setText("");
            timeStartLabel.setText("");
            durationLabel.setText("");
            placeLabel.setText("");
            descriptionLabel.setText("");
        }
    }
}

```

```

private void handleDeleteEvent()
{
    int selectedIndex = eventTable.getSelectionModel().getSelectedIndex();
    if (selectedIndex >= 0)
    {
        eventTable.getItems().remove(selectedIndex);
    }
    else
    {
        Alert alert = new Alert(AlertType.WARNING);
        alert.initOwner(mainApp.getPrimaryStage());
        alert.setTitle("No Selection");
        alert.setHeaderText("No Event Selected");
        alert.setContentText("Please select an Event in the table.");
        alert.showAndWait();
    }
}

@FXML
private void handleNewEvent()
{
    Event tempEvent = new Event();
    boolean okClicked = mainApp.showEventEditDialog(tempEvent);
    if (okClicked)
        mainApp.getEventData().add(tempEvent);
}

```

```

@FXML
private void handleEditEvent()
{
    Event selectedEvent = eventTable.getSelectionModel().getSelectedItem();
    if (selectedEvent != null)
    {
        boolean okClicked = mainApp.showEventEditDialog(selectedEvent);
        if (okClicked)
            showEventDetails(selectedEvent);
    }
    else
    {
        Alert alert = new Alert(AlertType.WARNING);
        alert.initOwner(mainApp.getPrimaryStage());
        alert.setTitle("No Selection");
        alert.setHeaderText("No Event Selected");
        alert.setContentText("Please select an event in the table.");
        alert.showAndWait();
    }
}

```

6. Варіанти використання

Можна використовувати для зберігання важливих заходів

ВИСНОВКИ

Ми отримали навички використання засобів клієнтських технологій (Client Technologies) платформи Java SE