

1 Intefejsy graficzne

JavaFX – początkowo zewnętrzna biblioteka języka Java obecnie jest zalecanym rozwiązaniem tworzenia interfejsów użytkownika w nowych programach.

1.1 Okienka

Klasa *Application* jest główną klasą aplikacji JavaFX i jest odpowiedzialna za wyświetlanie okien.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

public class FXTests extends Application {

    @Override
    public void start(Stage primaryStage) {

        StackPane root = new StackPane();
        Scene scene = new Scene(root, 300, 250);
        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

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

1.2 Komponenty

Istnieje szereg komponentów, które mogą zostać dodane do okien:

- Label
- Button
- ComboBox
- ProgressBar
- ...

```

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

public class FXTests extends Application {

    @Override
    public void start(Stage primaryStage) {
        StackPane root = new StackPane();

        Button btn = new Button();
        btn.setText("Hello");
        root.getChildren().add(btn);

        /*Label lbl = new Label("Hello");
        root.getChildren().add(lbl);*/

        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

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

```

1.3 Menadżery układu

Klasy menadżerów układu odpowiadają za prawidłowe rozłożenie komponentów w oknie:

- StackPane
- BorderPane
- HBox
- VBox
- GridPane
- ...

1.3.1 BorderPane

```

@Override
public void start(Stage primaryStage) {
    BorderPane root = new BorderPane();

    Button btn = new Button();
    Button btn2 = new Button("Button");
    btn.setText("Say 'Hello World'");
    Label lbl = new Label("Hello");

    root.setTop(btn);
    root.setRight(btn2);
    root.setLeft(lbl);

    Scene scene = new Scene(root, 300, 250);
    primaryStage.setTitle("Hello World!");
    primaryStage.setScene(scene);
    primaryStage.show();
}

```

1.3.2 HBox

```

@Override
public void start(Stage primaryStage) {
    HBox root = new HBox();

    Button btn = new Button();
    Button btn2 = new Button("Button");
    btn.setText("Say 'Hello World'");
    Label lbl = new Label("Hello");

    root.getChildren().add(btn);
    root.getChildren().add(btn2);
    root.getChildren().add(lbl);

    Scene scene = new Scene(root, 300, 250);
    primaryStage.setTitle("Hello World!");
    primaryStage.setScene(scene);
    primaryStage.show();
}

```

1.3.3 VBox

```

@Override
public void start(Stage primaryStage) {
    VBox root = new VBox();

    Button btn = new Button();
    Button btn2 = new Button("Button");
    btn.setText("Say 'Hello World'");
    Label lbl = new Label("Hello");

    root.getChildren().add(btn);
    root.getChildren().add(btn2);
    root.getChildren().add(lbl);

    Scene scene = new Scene(root, 300, 250);
    primaryStage.setTitle("Hello World!");
}

```

```
primaryStage.setScene(scene);
primaryStage.show();
}
```

2 Obsługa zdarzeń

```
Button btn = new Button();
btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {
        System.out.println("Clicked! :");
    }
});
```

3 Język FXML

Do deklaracji interfejsu użytkownika można także wykorzystać specjalny język FXML, dzięki któremu można oddzielić sposób działania aplikacji od jego wyglądu.

3.1 Plik FXML

```
<?xml version="1.0" encoding="UTF-8"?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<VBox id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns:fx="http://javafx.com/↔
    fxml/1" fx:controller="com.adeik.fxtests.FXMLController">

<Label text="Label" id="label" minHeight="16" minWidth="69" prefHeight="16" prefWidth="↔
    69" fx:id="label" />
<Button text="Button" onAction="#buttonHandler" />
<TextField text="TextField" fx:id="textField" />

</VBox>
```

3.2 Kontroler

Obiekt kontrolera odpowiada za obsługę interfejsu użytkownika.

```
package com.adeik.fxtests;

import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.Event;
```

```

import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TextField;

public class FXMLController implements Initializable {

    @FXML
    private TextField textField;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    protected void buttonHandler(Event e){
        textField.setText("Hello!");
    }

}

```

3.3 Okienko

```

package com.adeik.fxtests;

import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

public class FXMLTest extends Application {

    @Override
    public void start(Stage primaryStage) {
        try {
            Pane myPane = (Pane)FXMLLoader.load(FXMLTest.class.getResource("/com/adeik/fxtests↔
/FXML.fxml"));
            Scene myScene = new Scene(myPane);
            primaryStage.setScene(myScene);
            primaryStage.show();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

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

}

```

4 Zadania do wykonania

1. Zapoznać się z programem SceneBuilder

2. Zapoznać się z dokumentacją klas menadżerów układu (JavaFX Layout Managers)
3. Stworzyć okno z komponentami i rozmieszczeniem zaproponowanym przez prowadzącego
4. Obsłużyć zdarzenia za pomocą EventHandlerów
5. Wykorzystać lambdy do obsługi zdarzeń
6. Stworzyć rozkład komponentów za pomocą języka FXML