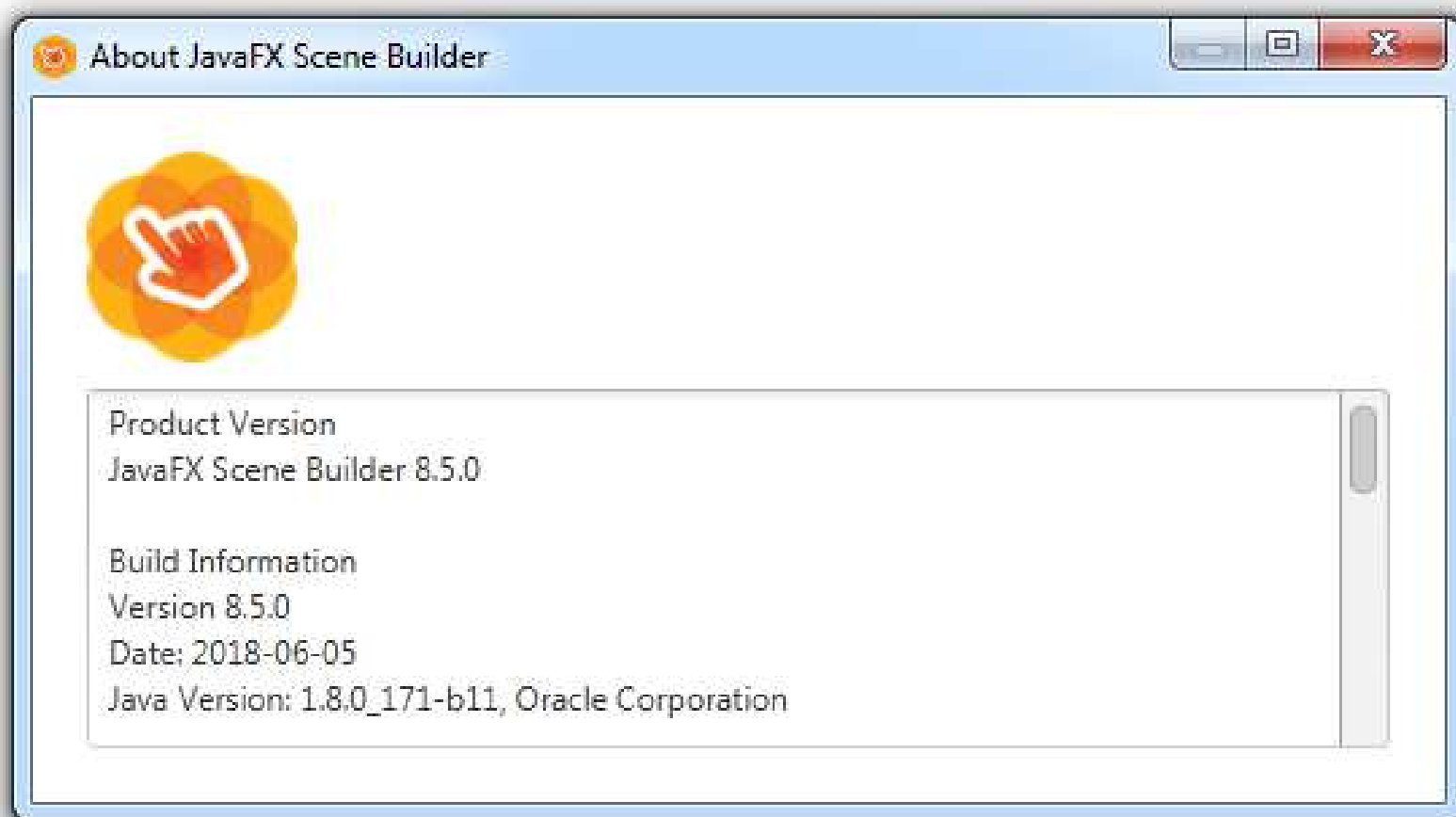


Tworzymy projekt JavaFX, Eclipse

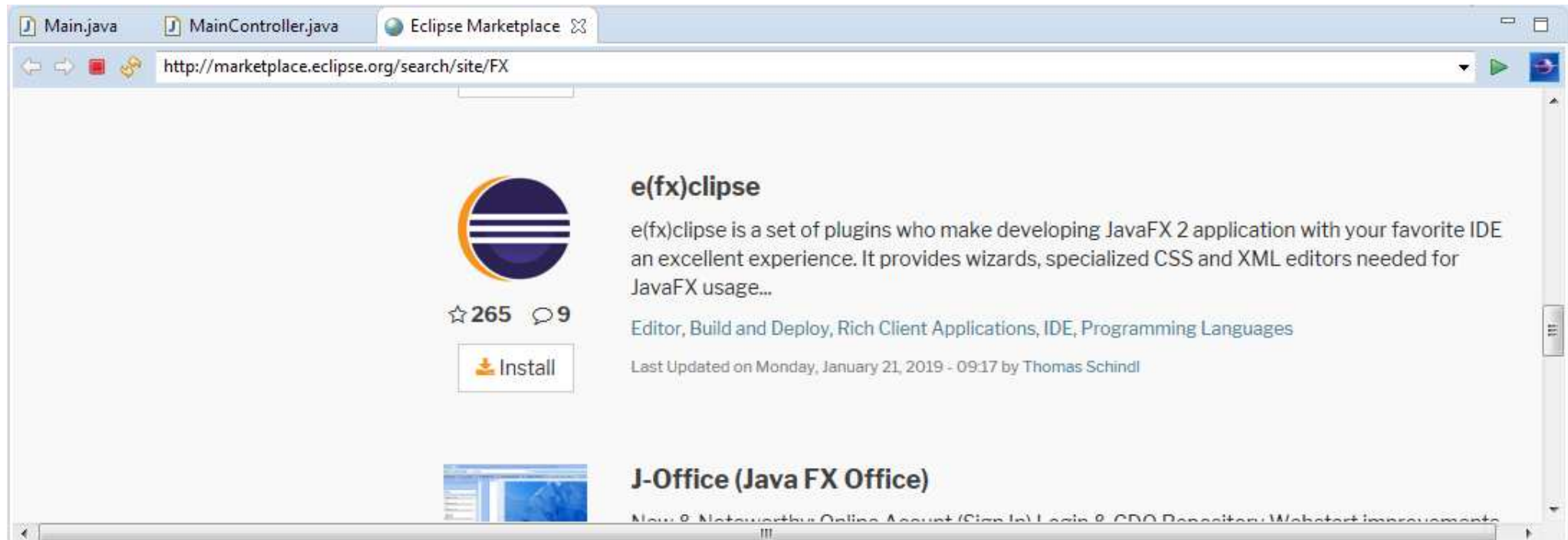
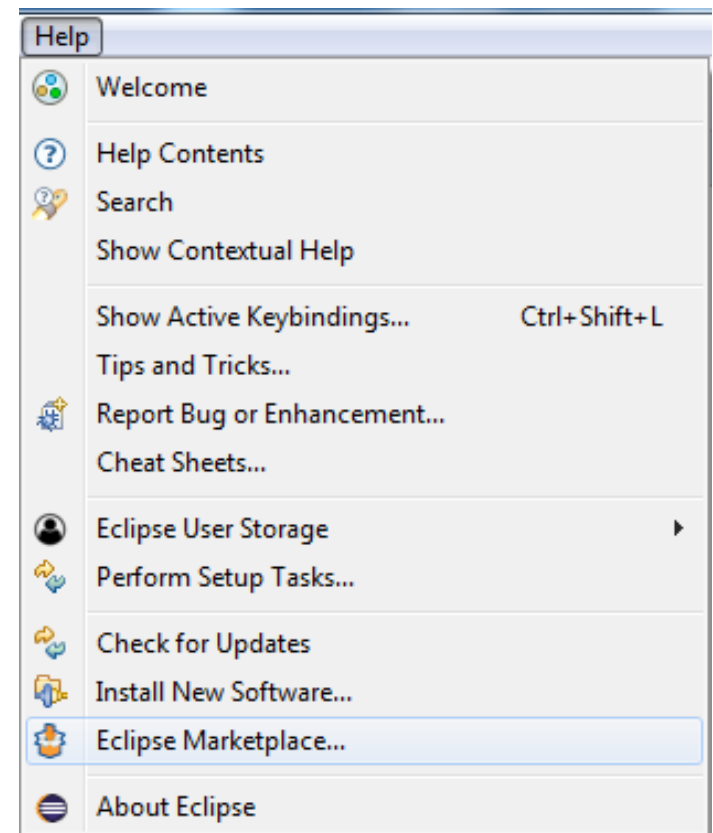


Tworzymy projekt JavaFX, Scene Builder

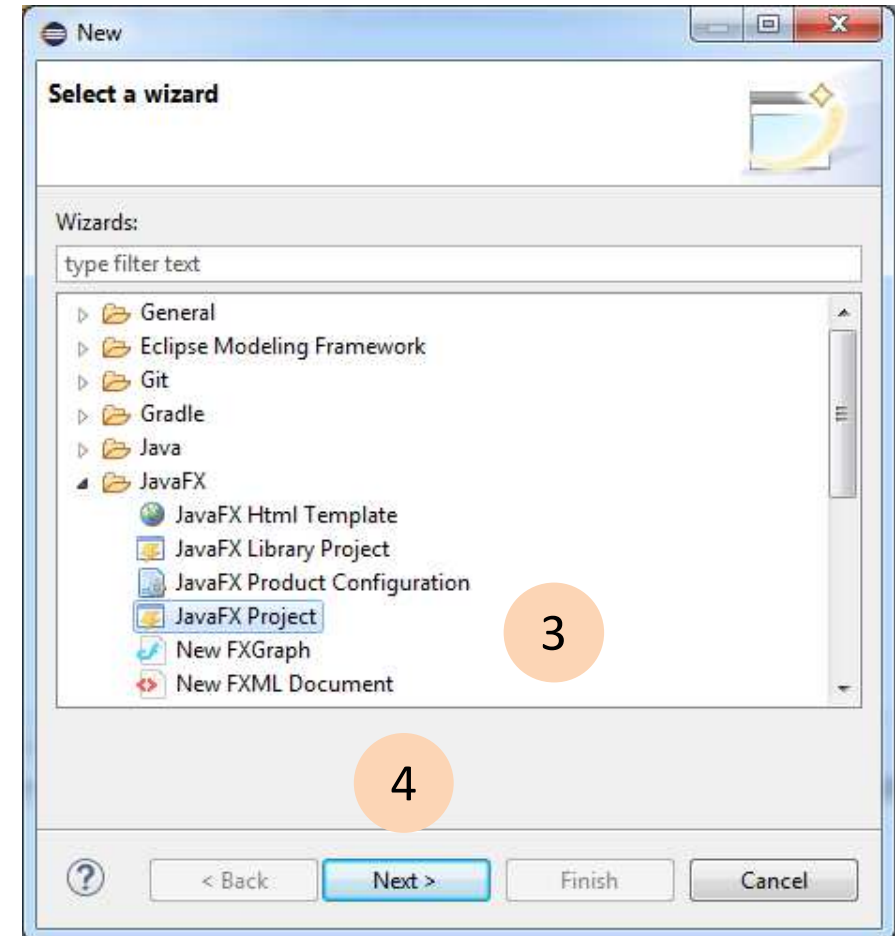
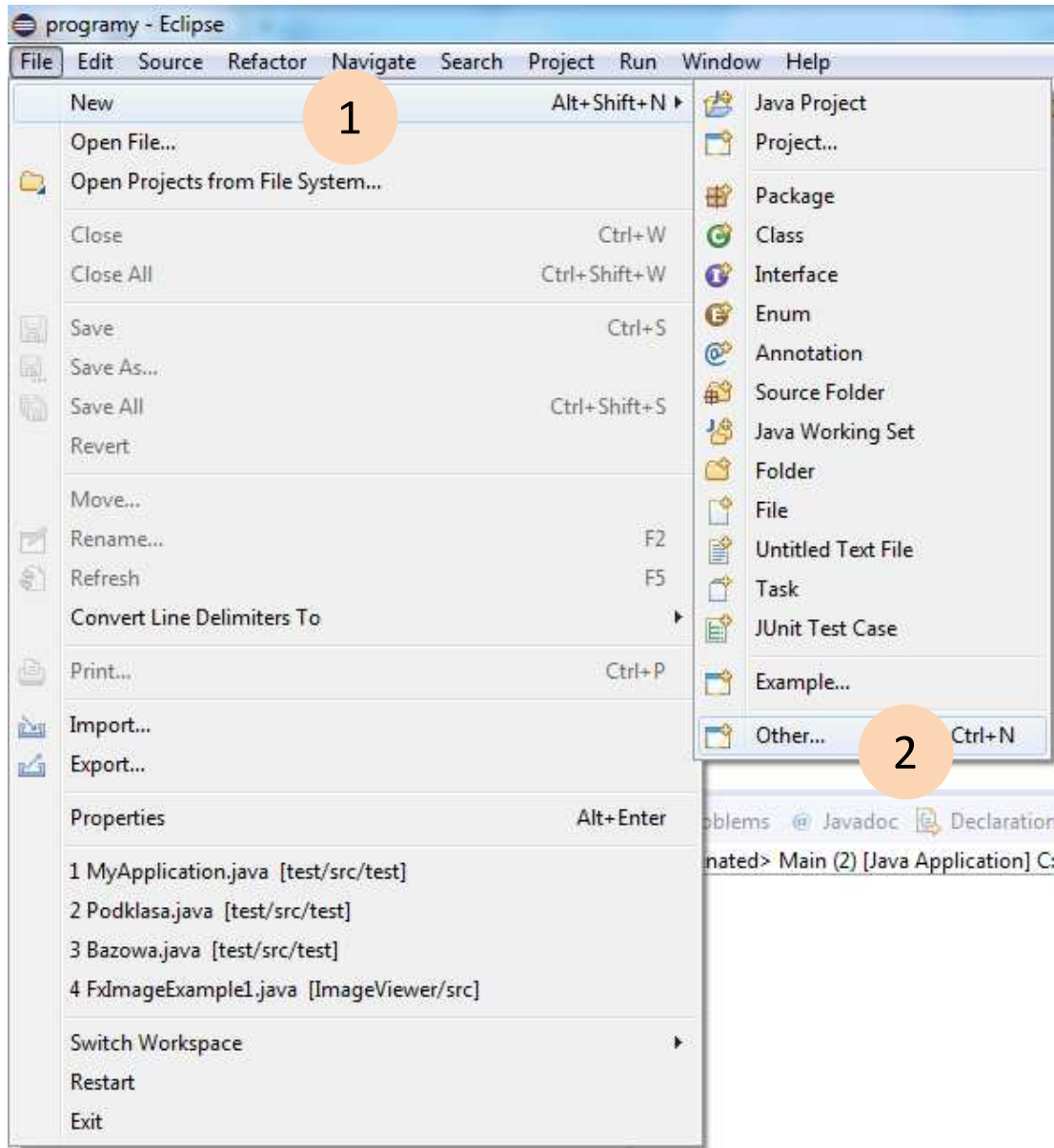


Jeśli korzystasz z wersji Javy 10 lub 11 użyj
Scene Builder'a odpowiednio w wersji 10 lub 11.

Tworzymy projekt JavaFX, plugin



Tworzymy projekt JavaFX, 1/2



Tworzymy projekt JavaFX, 2/2

New Java Project

Create a Java Project
Create a Java project in the workspace or in an external location.

Project name: **5**

☒ Use default location

Location: [Browse...](#)

JRE

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE (currently 'jre1.8.0_201') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

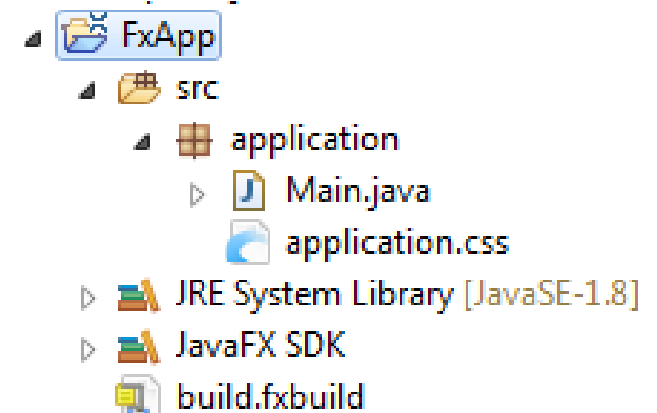
☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

6

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

! Utworzony projekt



Aplikacja JavaFX



```
public class Main extends Application {

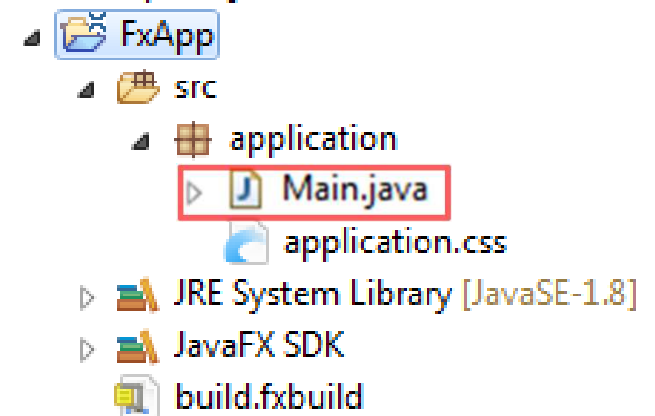
    @Override
    public void start(Stage primaryStage) {

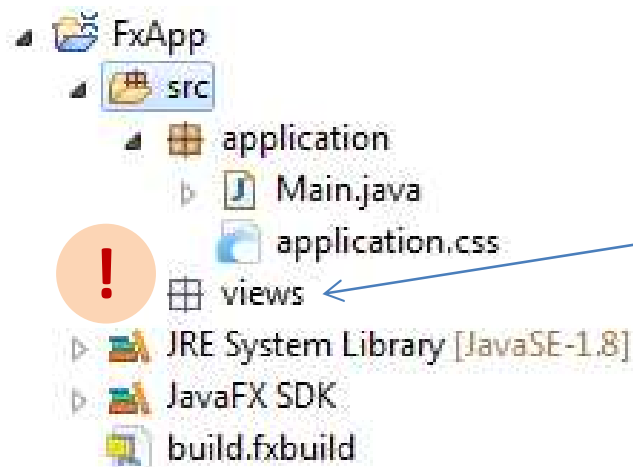
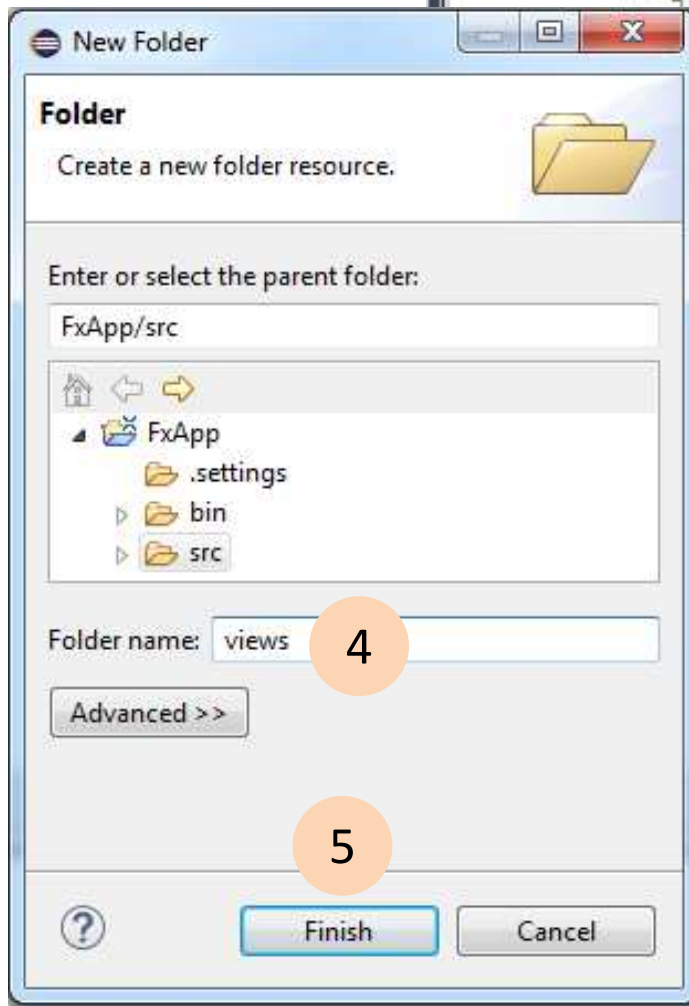
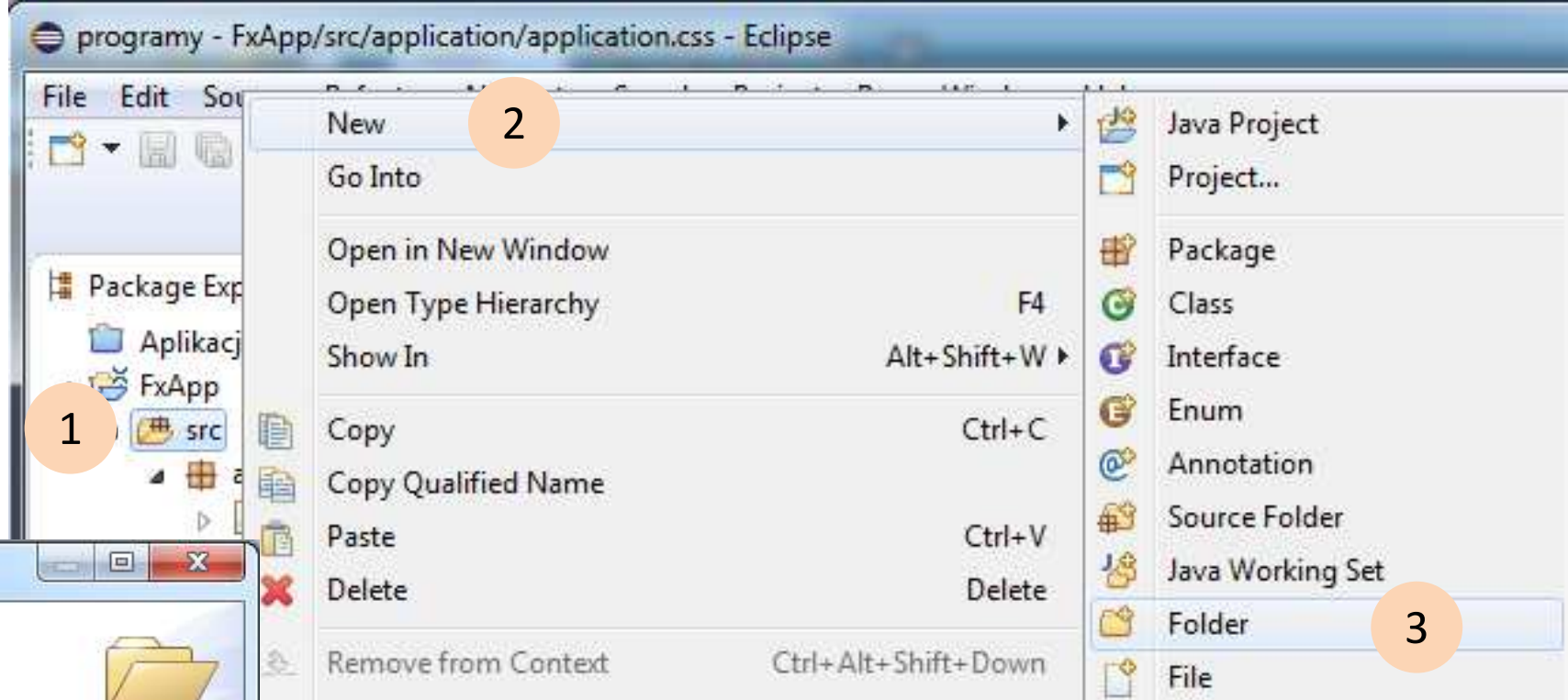
        try {
            BorderPane root = new BorderPane();
            Scene scene = new Scene(root,400,400);
            scene.getStylesheets().add(getClass().getResource("application.css")
                                                                    .toExternalForm());

            primaryStage.setScene(scene);
            primaryStage.show();
        } catch (Exception e) {e.printStackTrace();}

    }

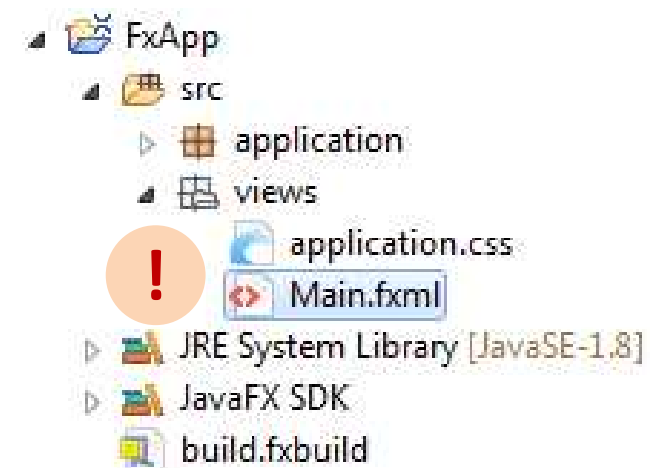
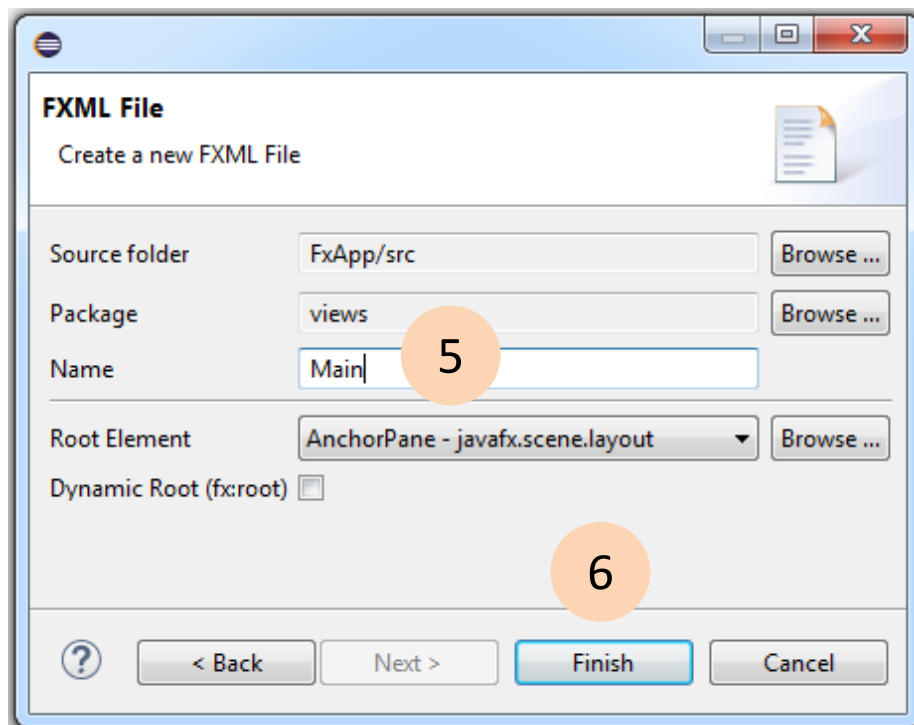
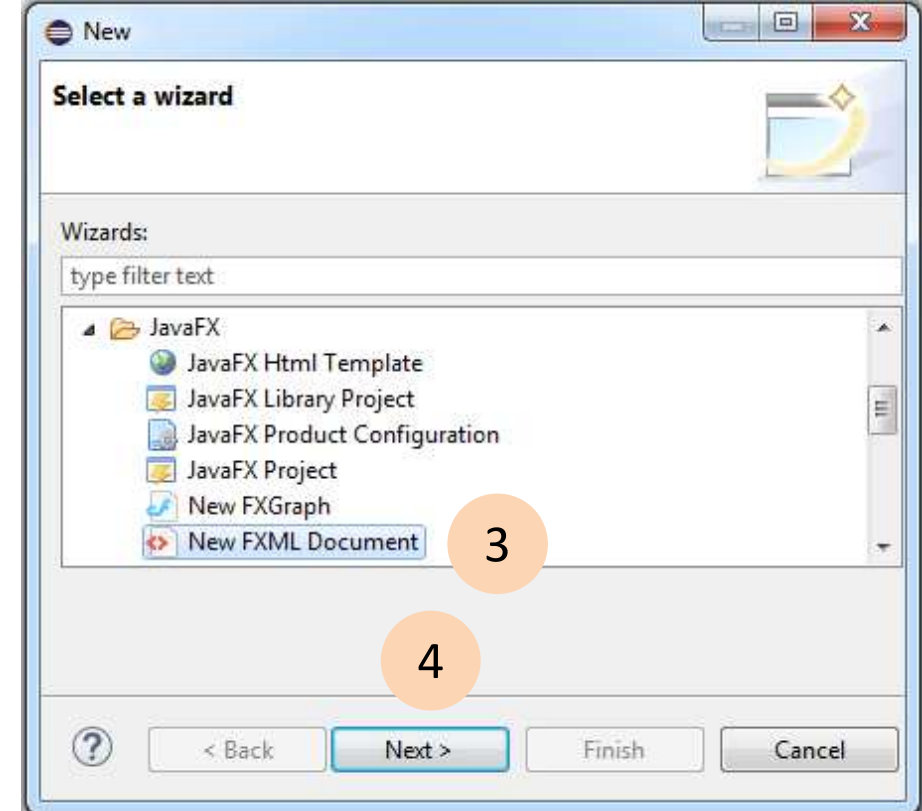
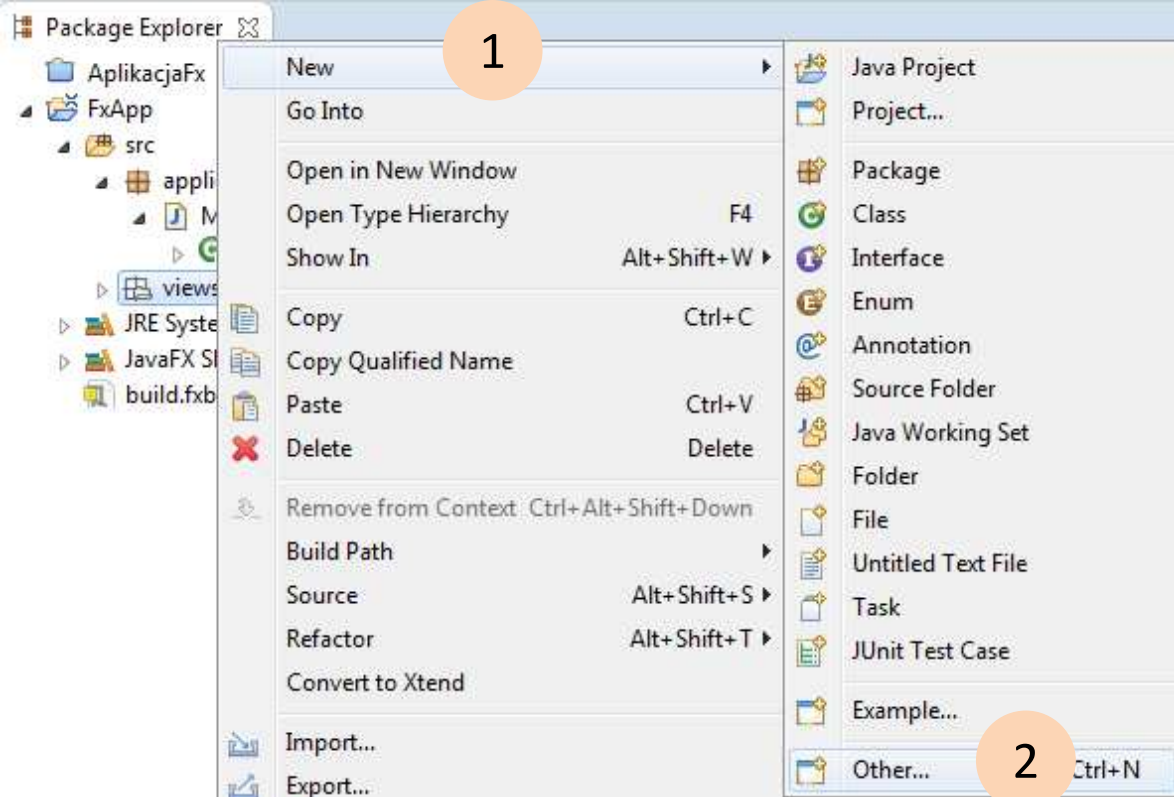
    public static void main(String[] args) {
        launch(args);
    }
}
```





Ponieważ plików *.fxml może być wiele w projekcie, dobrym pomysłem jest utworzenie katalogu, który będzie przechowywał formatki.

Tworzymy folder na pliki fxml



Tworzymy formatkę fxml

Aplikacja JavaFX



```
public class Main extends Application {
```

```
@Override
```

```
public void start(Stage primaryStage) {
```

```
try {
```

```
    BorderPane root = new BorderPane();
```

```
    Scene scene = new Scene(root, 400, 400);
```

```
    scene.getStylesheets().add(getClass().getResource("application.css")  
                                     .toExternalForm());
```

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

```
    primaryStage.show();
```

```
} catch (Exception e) {e.printStackTrace();}
```

```
}
```

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

```
    launch(args);
```

```
}
```

```
}
```

Zmienimy tę wygenerowaną linię kodu, aby wczytać naszą formatkę fxml

Ponieważ przenieśliśmy nasz plik CSS do katalogu views należałoby Zmienić ścieżkę na:

`"/views/application.css"`

W dalszej części nie będziemy korzystać z tego pliku, więc usuniemy tę linię kodu.

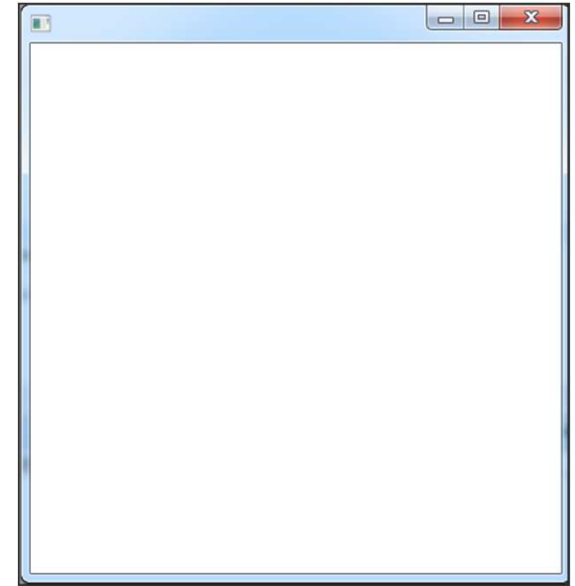
Aplikacja JavaFX



```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;

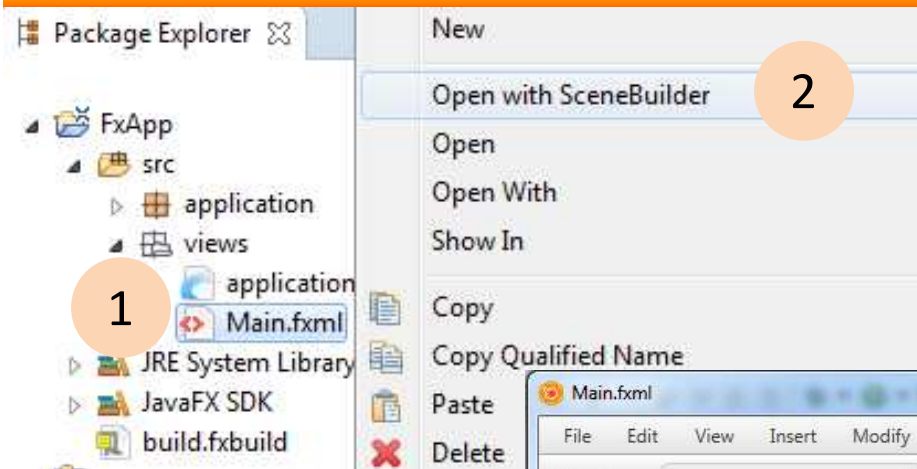
public class MyApplication extends Application {
    @Override
    public void start(Stage primaryStage){
        Parent root = FXMLLoader.load(getClass().getResource("/views/Main.fxml"));
        Scene scene = new Scene(root, 400, 400);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        Launch(MyApplication.class, args);
    }
}
```

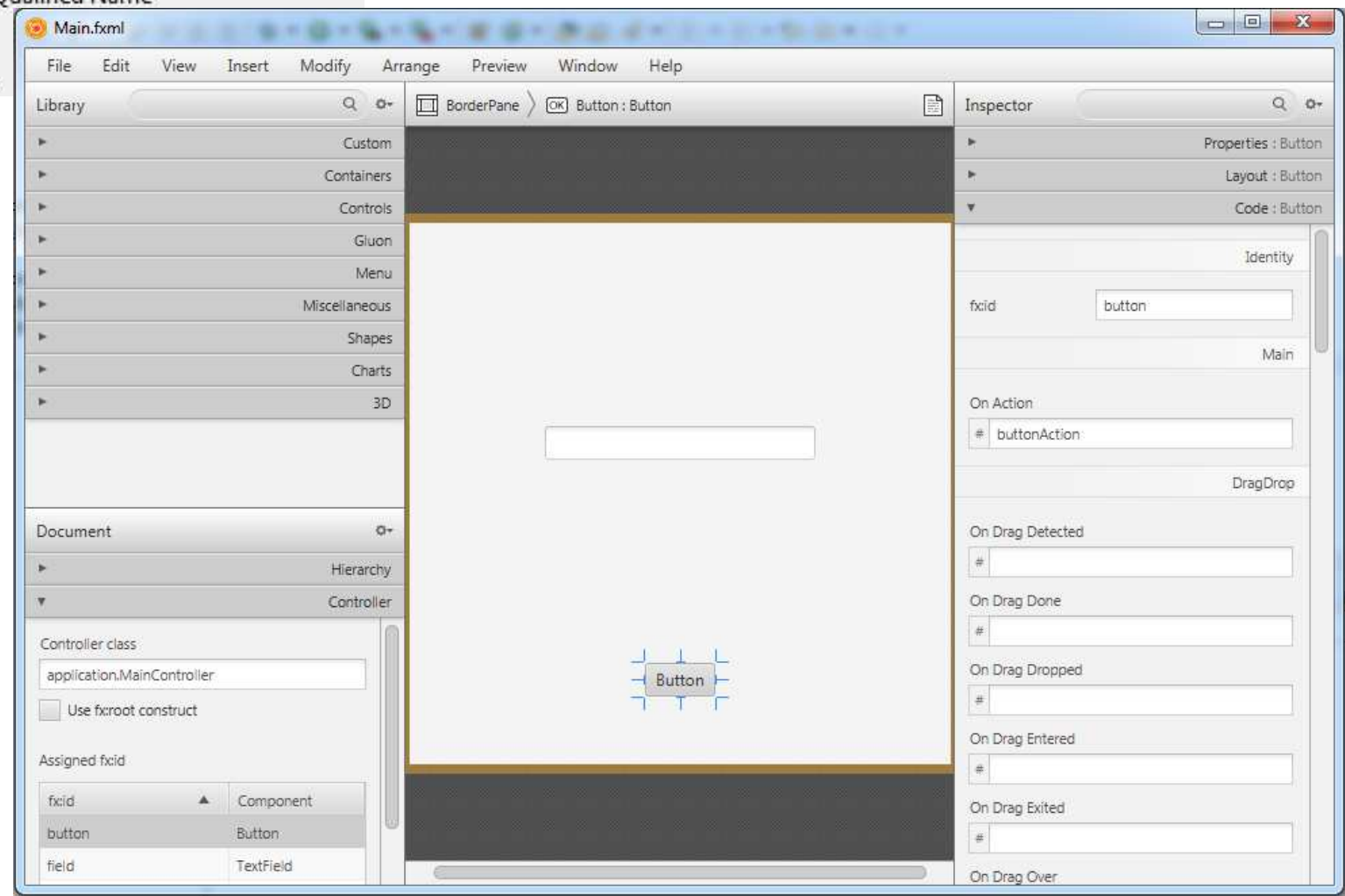


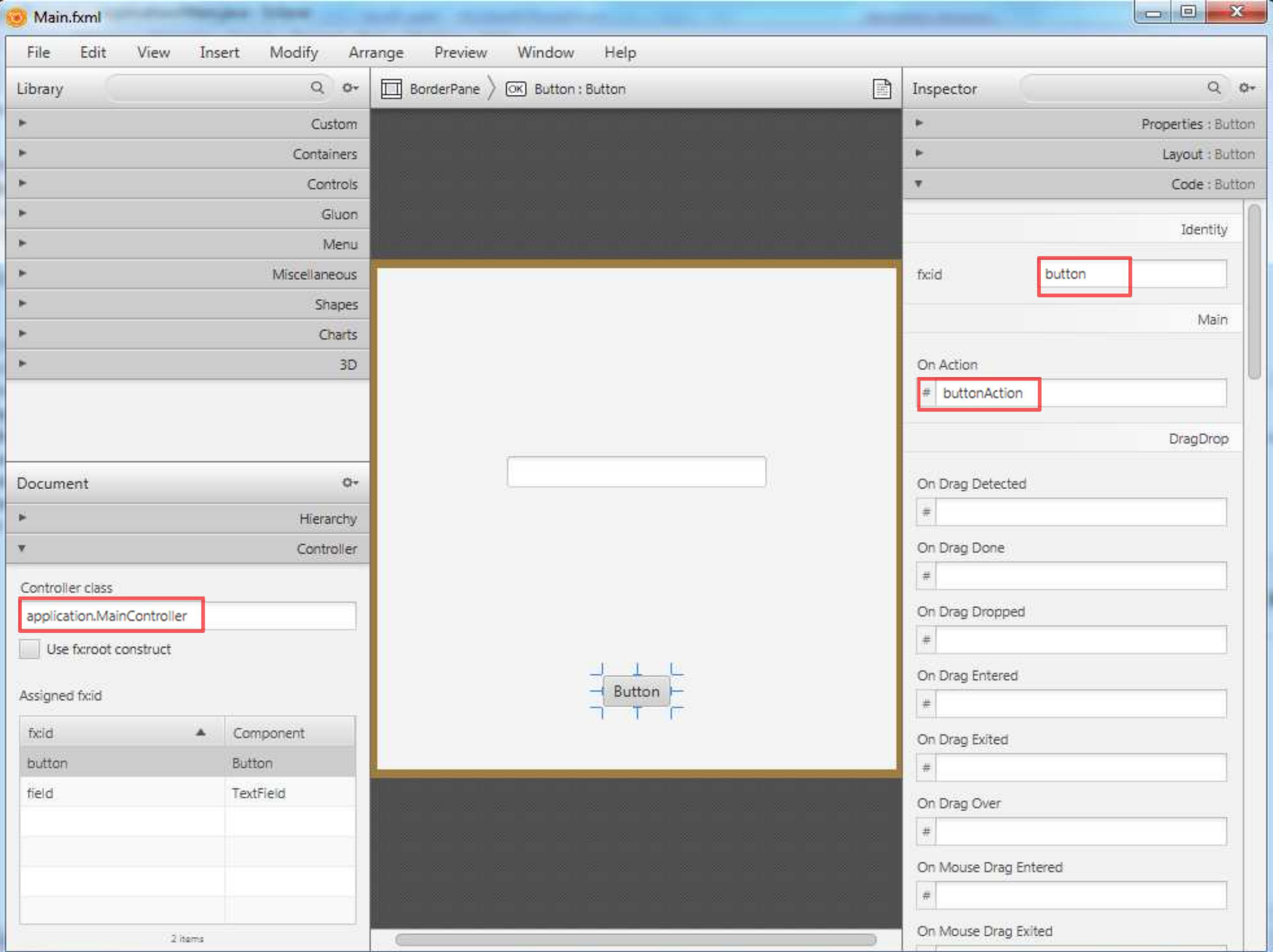
Wczytujemy formatkę fxml

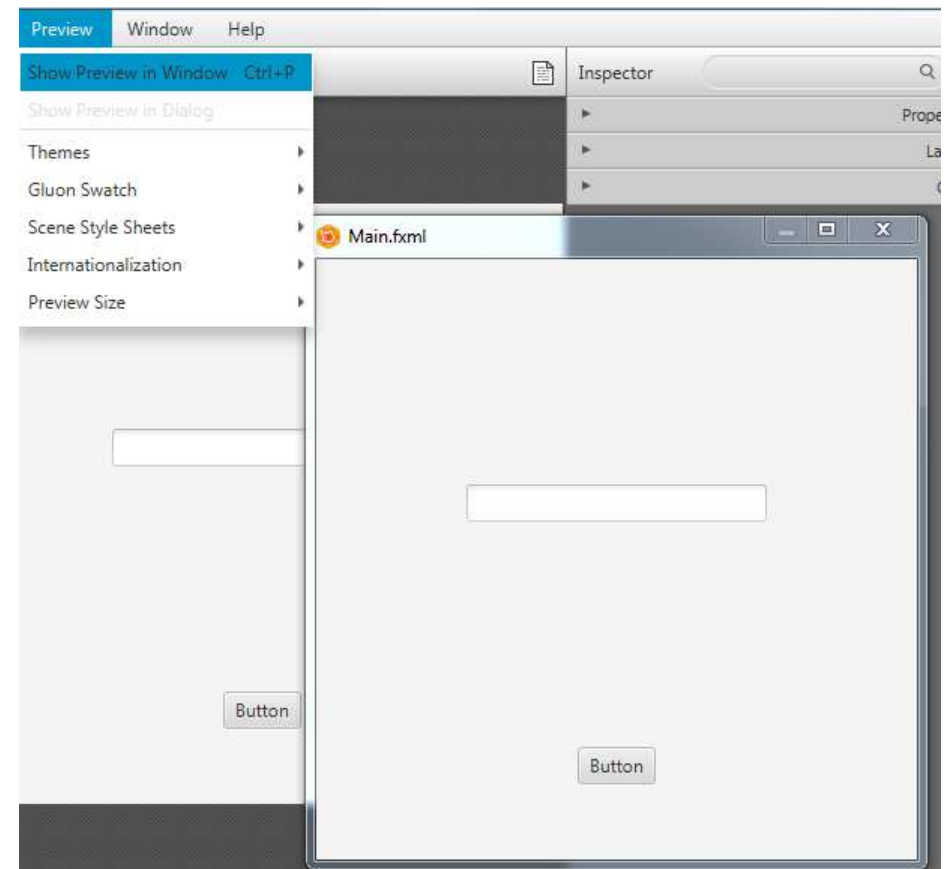
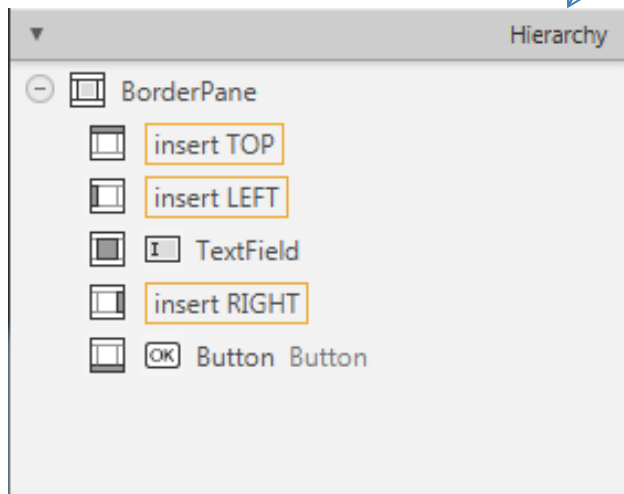
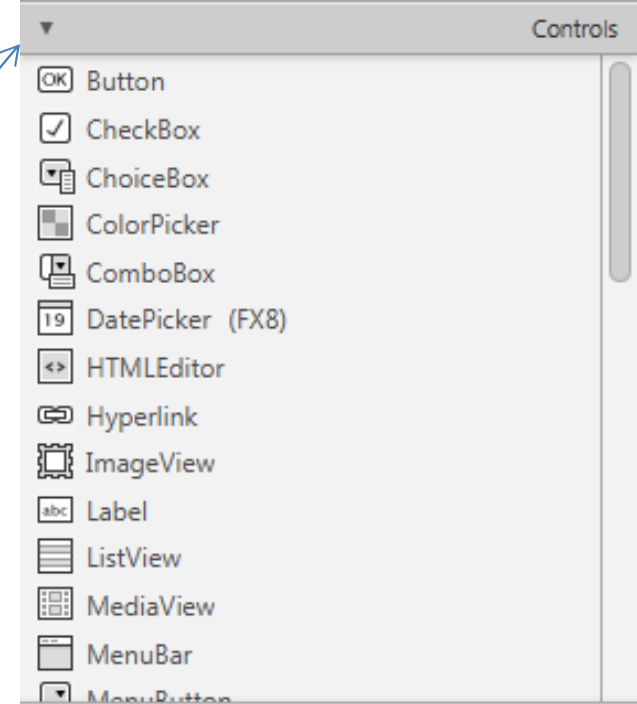
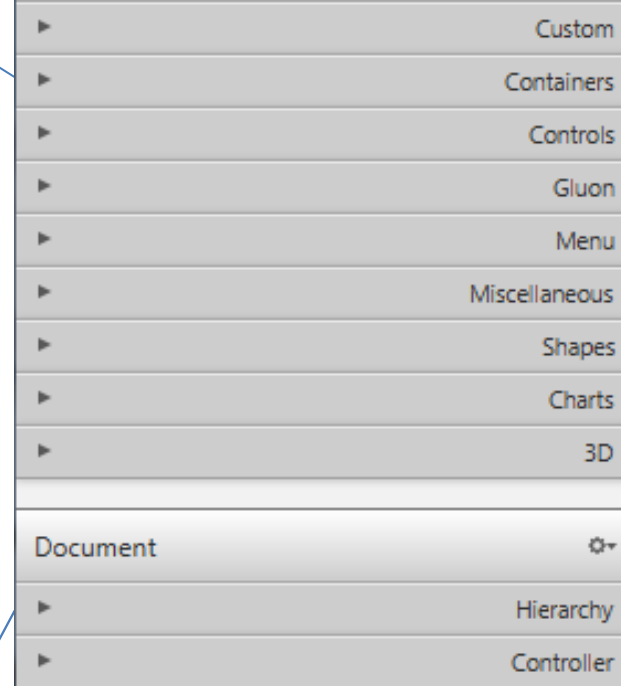
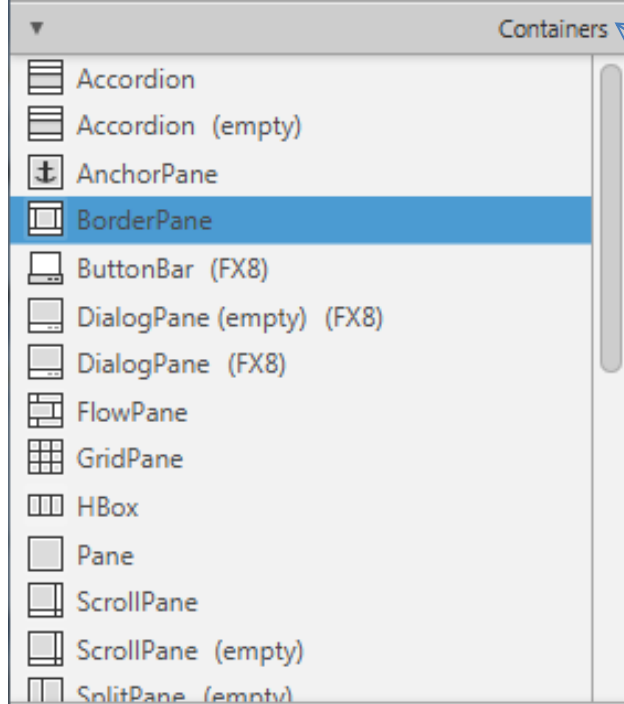
Otwieramy formatkę w Scene Builder



! Za pomocą Scene Builder'a możemy zdefiniować wygląd naszej aplikacji, także wygenerować szkielet kontrolera.





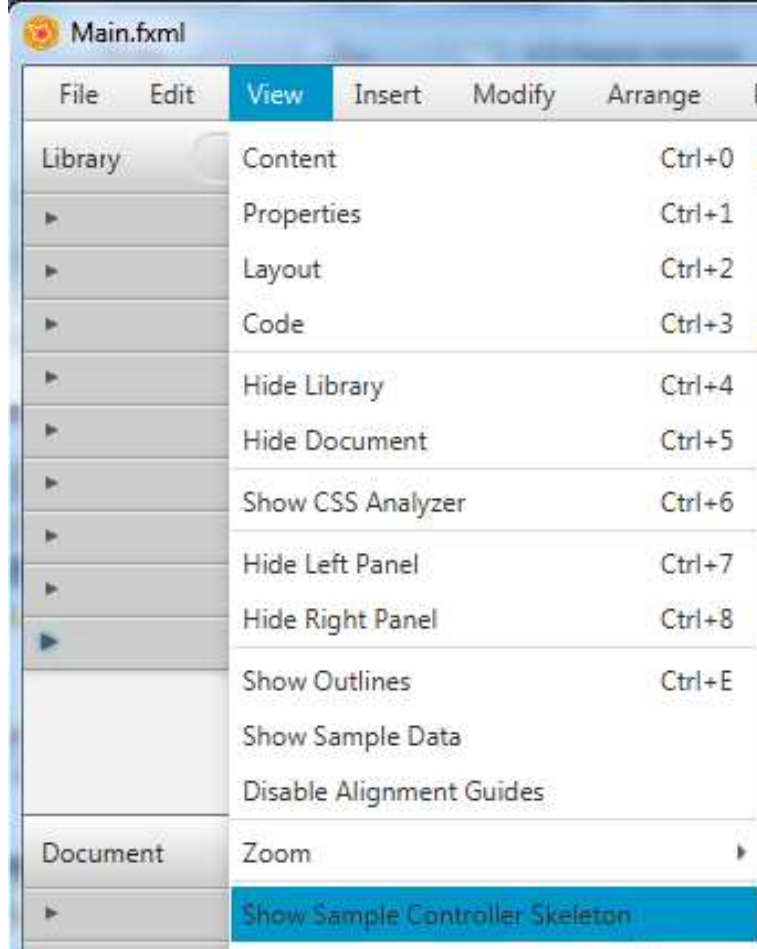


Wygenerowana Formatka FXML

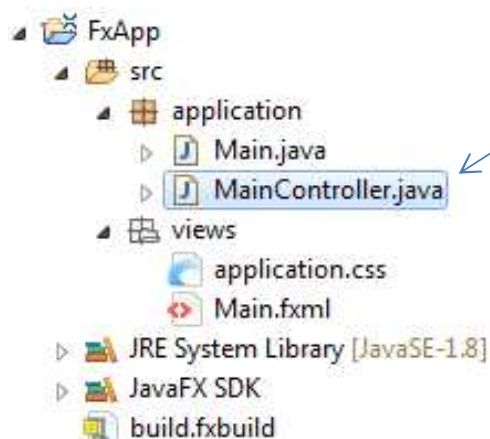
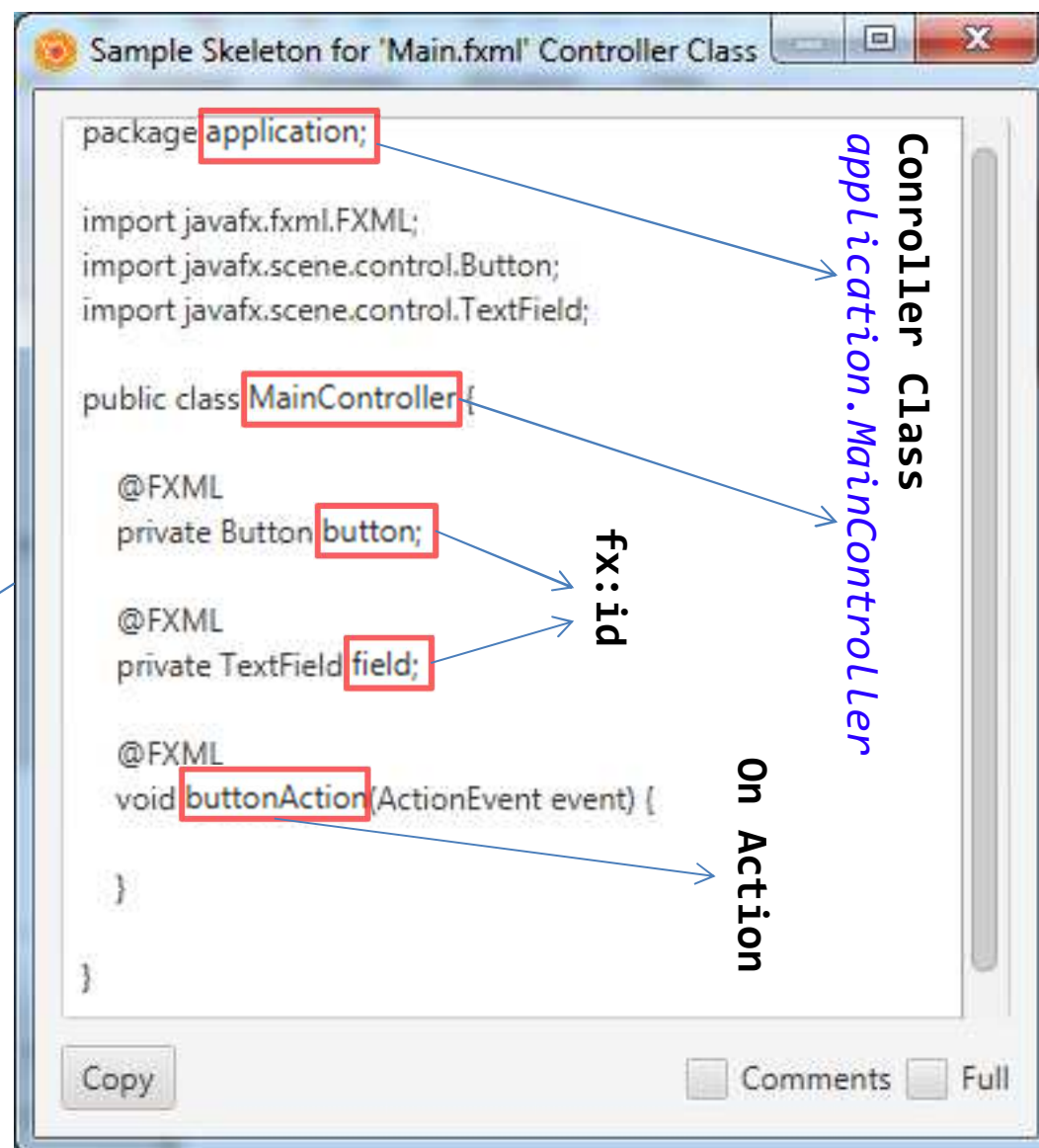


```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>

<BorderPane prefHeight="400.0" prefWidth="400.0,, ...
    xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1
        fx:controller="application.MainController">
    <bottom>
        <Button fx:id="button" mnemonicParsing="false" onAction="#buttonAction"
            text="Button" BorderPane.alignment="CENTER">
            <BorderPane.margin>
                <Insets bottom="50.0" />
            </BorderPane.margin>
        </Button>
    </bottom>
    <center>
        <TextField fx:id="field" maxWidth="200.0" BorderPane.alignment="CENTER">
            <BorderPane.margin>
                <Insets />
            </BorderPane.margin>
        </TextField>
    </center>
</BorderPane>
```



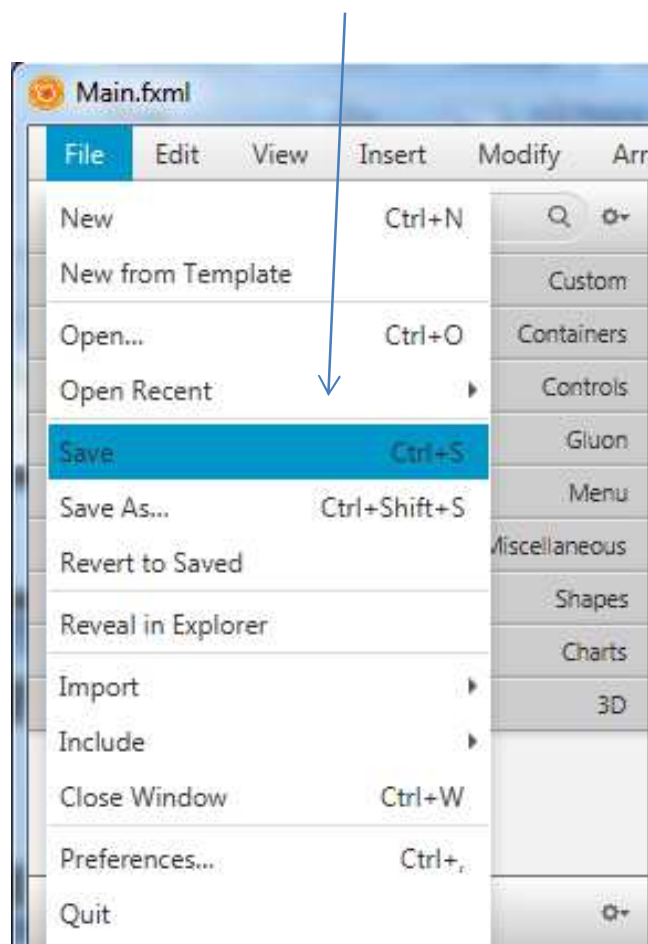
Generowanie szkieletu kontrolera aplikacji



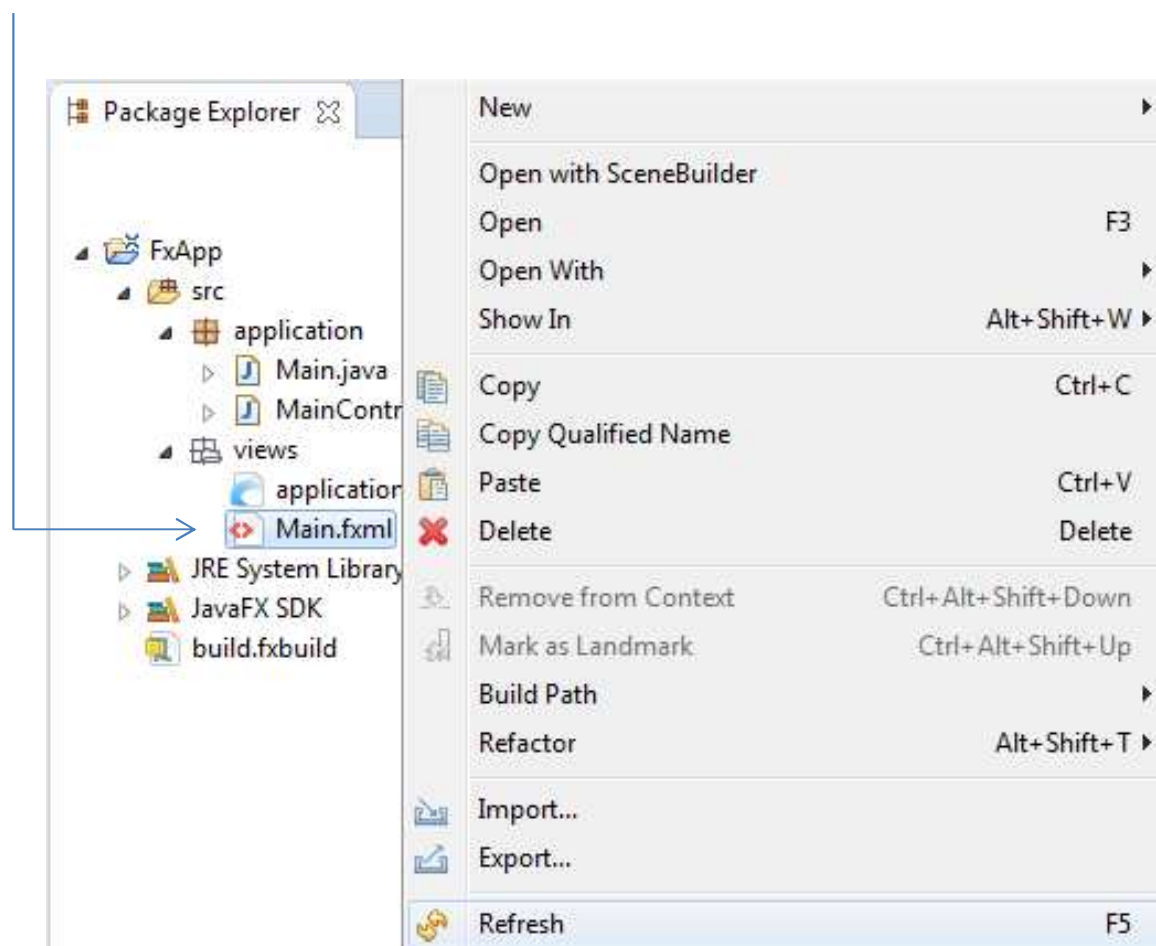
Zanim uruchomimy aplikację...



Należy pamiętać o zapisaniu zmian w formacie .fxml



Należy również odświeżyć plik w projekcie programu Eclipse



Uruchamiamy Aplikację



The image shows an IDE window with two tabs: `Main.java` and `MainController.java`. The `MainController.java` tab is active, displaying the following code:

```
1 package application;
2
3 import javafx.event.ActionEvent;
4 import javafx.fxml.FXML;
5 import javafx.scene.control.Button;
6 import javafx.scene.control.TextField;
7
8 public class MainController {
9
10     @FXML
11     private Button button;
12
13     @FXML
14     private TextField field;
15
16     @FXML
17     void buttonAction(ActionEvent event) {
18         System.out.println(field.getText());
19         field.clear();
20     }
21
22 }
```

Below the code editor, the `Console` tab is active, showing the output:

```
Main (8) [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\j
Hello
```

Overlaid on the right side of the IDE is a window representing the running JavaFX application. The window has a title bar with standard OS controls. Inside, there is a text input field containing the text "Hello" and a button labeled "Button" at the bottom right.