



Programmazione III

Prof.ssa Liliana Ardissono
Dipartimento di Informatica
Università di Torino

Interfacce Utente Grafiche (GUI)
Overview di JavaFX – parte 2
JavaFXML



Questa presentazione è distribuita sotto licenza Creative Commons CC BY ND

E' possibile semplificare il codice delle GUI



La semplificazione si basa su due concetti:

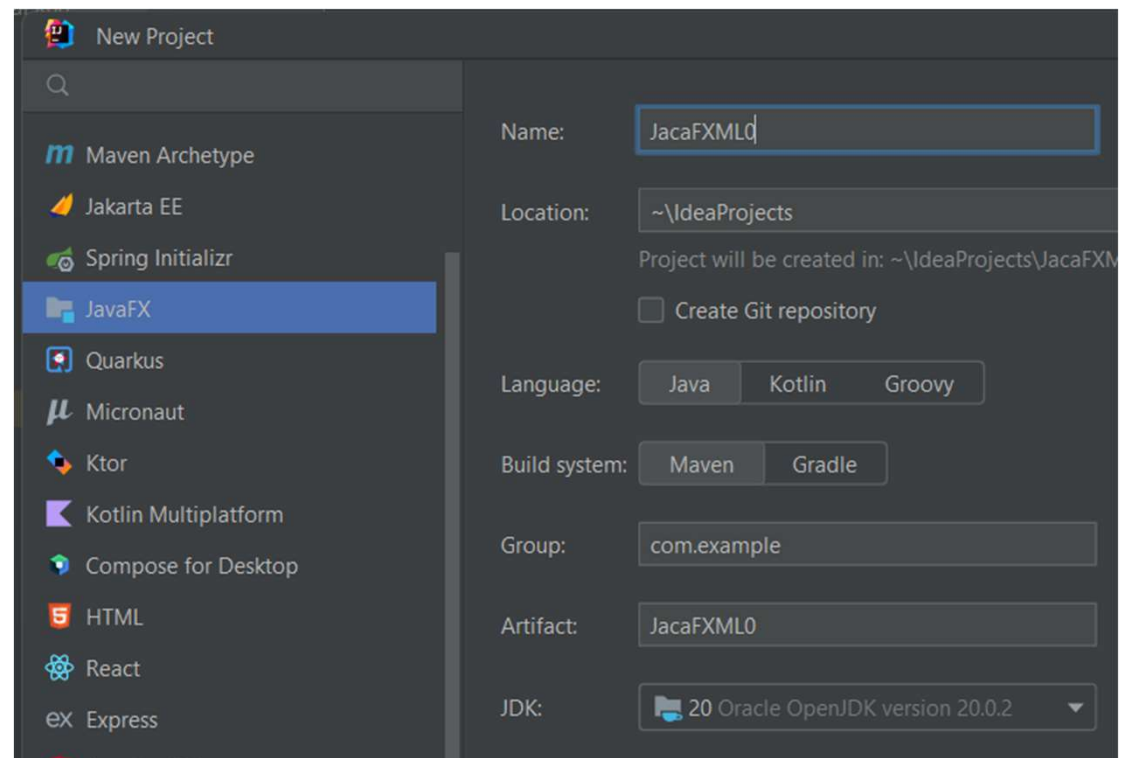
- **Definizione dichiarativa dell'interfaccia grafica, attraverso il linguaggio XML (JavaFXML)** – Evita di sviluppare le componenti grafiche pezzo per pezzo!
- **Uso di java beans e properties** per
 1. Caratterizzare i dati del model in modo standard
 2. Legare i dati del model tramite binding alle componenti dell'interfaccia grafica (o a componenti di osservatori, in generale) che devono reagire ai cambiamenti di valore in automatico

PASSO 1 - XML



Impariamo a definire le interfacce grafiche in modo dichiarativo, utilizzando il linguaggio XML per specificare i componenti delle GUI:

JavaFXML

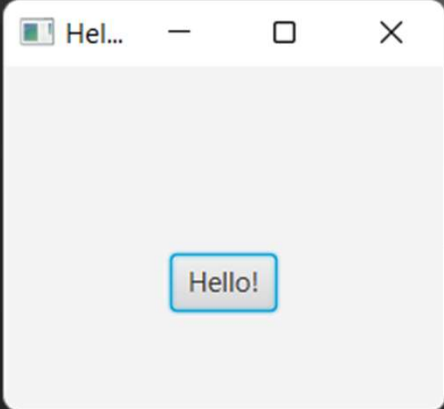


Esempio – applicazione JavaFXML0



La creiamo come new Project JavaFX, nasce già organizzata con architettura MVC (ma il Model non c'è)

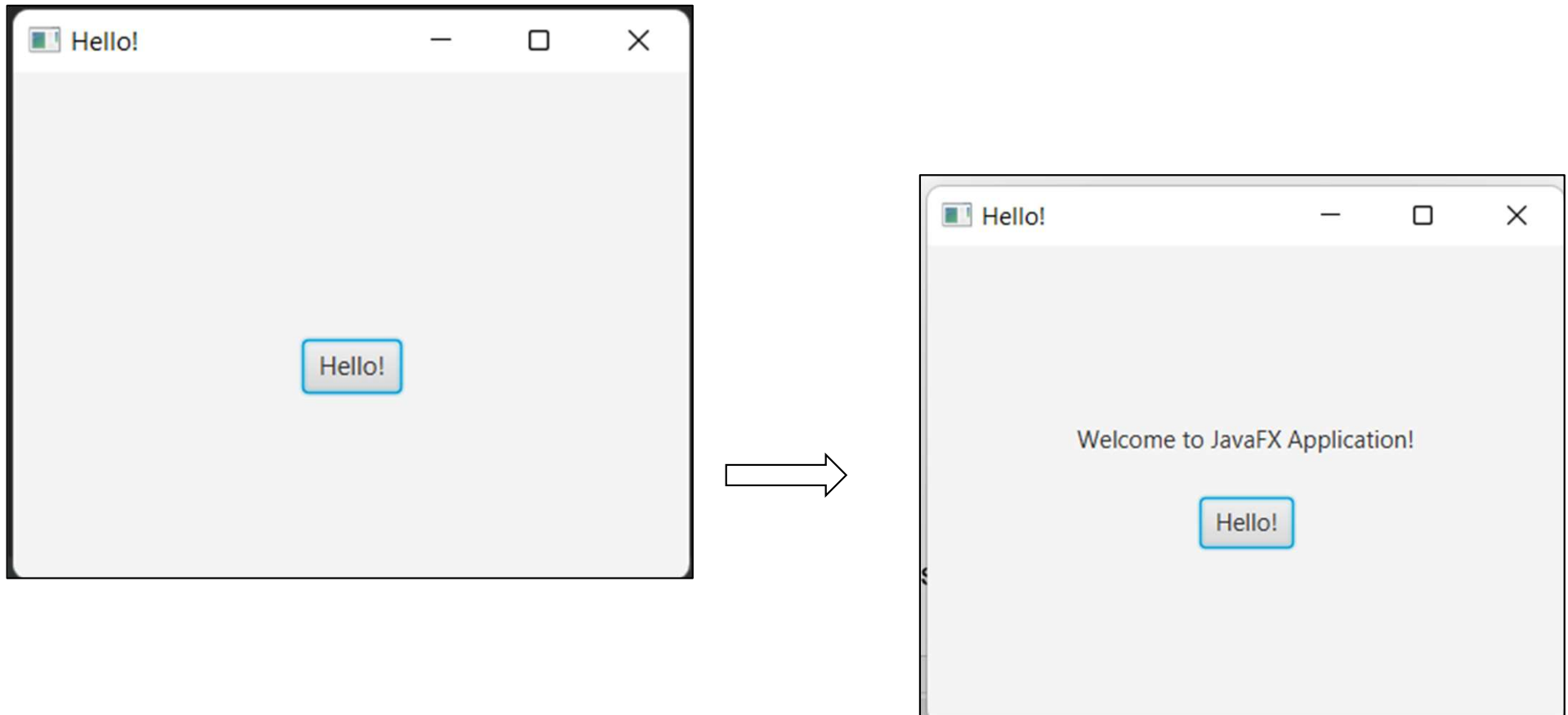
```
public class HelloApplication extends Application {  
    @Override  
    public void start(Stage stage) throws IOException {  
        FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource  
        Scene scene = new Scene(fxmlLoader.load(), 320, 240);  
        stage.setTitle("Hello!");  
        stage.setScene(scene);  
        stage.show();  
    }  
  
    public static void main(String[] args) { launch(); }  
}
```



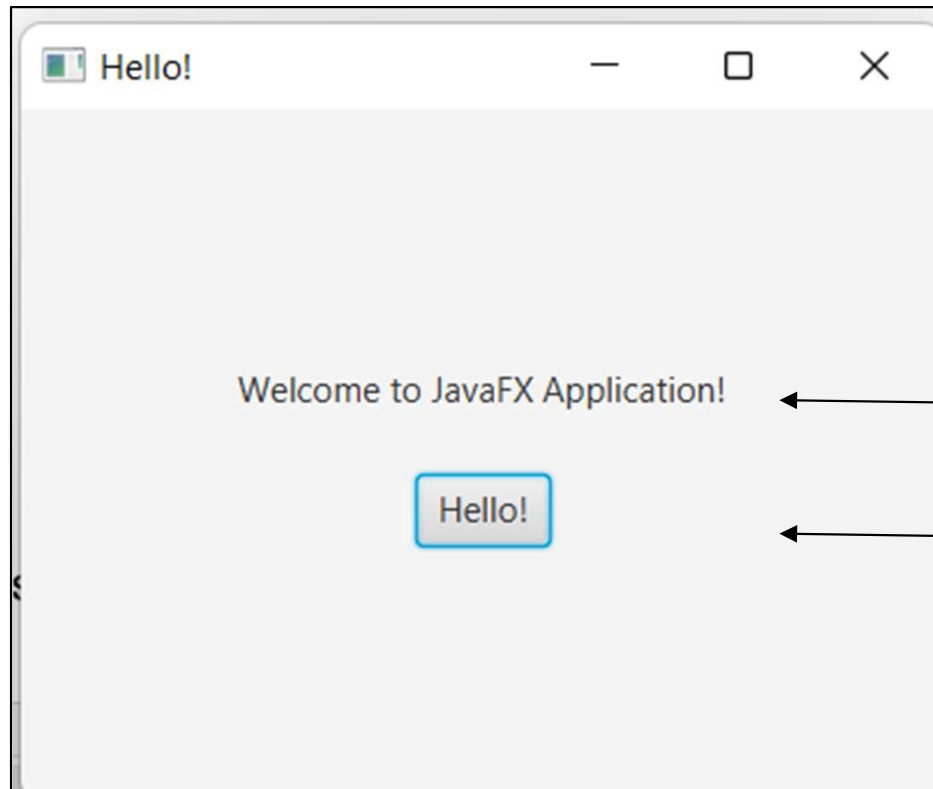
JavaFXML0 - funzionamento



Cliccando il bottone appare la scritta “Welcome...”



JavaFXML0 - ID dei componenti della GUI



Label welcomeText

Button «Hello!»



JavaFXML0 – hello-view.fxml

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

```
<?import javafx.geometry.Insets?>
```

... altre import per far funzionare i componenti grafici della view

```
<VBox alignment="CENTER" spacing="20.0"  
xmlns:fx="http://javafx.com/fxml"
```

```
    fx:controller="com.example.javafxml0.HelloController">
```

```
    <padding>
```

```
        <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"/>
```

```
    </padding>
```

```
    <Label fx:id="welcomeText"/>
```

```
    <Button text="Hello!" onAction="#onHelloButtonClick"/>
```

```
</VBox>
```

JavaFXML0 – Il controller



// il Controller usa le Java annotations per inizializzare le variabili in automatico
// il Controller si basa sugli ID dati alle componenti grafiche (welcomeText)

```
public class HelloController {  
    @FXML  
    private Label welcomeText;  
  
    @FXML  
    protected void onHelloButtonClick() {  
        welcomeText.setText("Welcome to JavaFX Application!");  
    }  
}
```


JavaFXML0 – Applicazione (main)



```
public class HelloApplication extends Application {  
    @Override  
    public void start(Stage stage) throws IOException {  
        FXMLLoader fxmlLoader =  
            new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));  
        Scene scene = new Scene(fxmlLoader.load(), 320, 240);  
        stage.setTitle("Hello!");  
        stage.setScene(scene);  
        stage.show();  
    }  
    public static void main(String[] args) {  
        launch();  
    }  
}
```

JavaFX – GUI XML con SceneBuilder



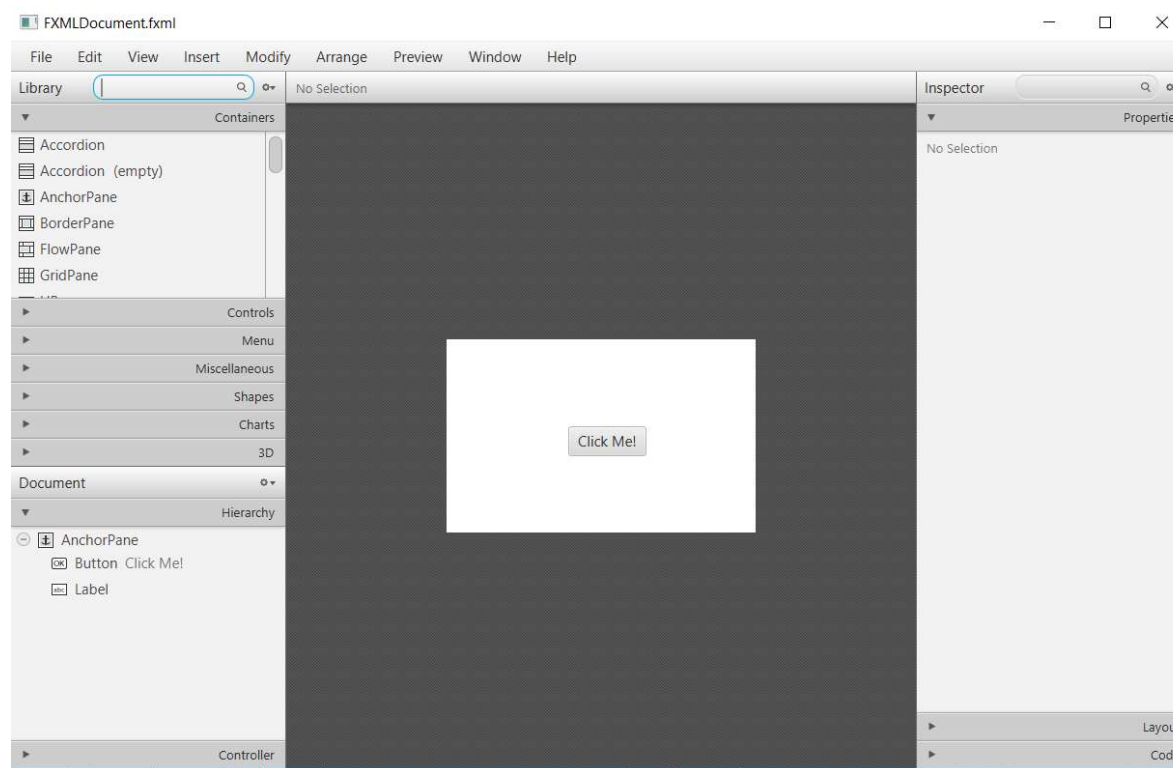
Per evitare di scrivere i file XML a mano:

- Utilizzare Scene Builder come strumento grafico per il design dell'interfaccia utente → Scene Builder genera automaticamente i file XML che specificano le componenti dell'interfaccia e la loro posizione nella finestra dell'applicazione
- Scene Builder: scaricare l'installazione e installare il software sul proprio computer.
- Aprire i documenti FXML con SceneBuilder e modificarli nelle loro parti grafiche

Scene Builder – modifica della GUI



- Aprire il file XML che definisce la GUI su cui si deve lavorare



- Poi, drag and drop (e ridimensionamento) delle componenti grafiche
→ SceneBuilder modifica il documento XML che descrive la GUI (FXMLDocument.fxml) per includere le nuove componenti secondo il layout specificato graficamente (salvare!)

Come aggiornare il codice dell'applicazione alla nuova GUI XML? – I



Il codice generato da SceneBuilder descrive i componenti grafici ma **manca la logica di gestione dell'interfaccia** (per es., l'aggancio al controller)

→ bisogna completare il codice con gli ID delle componenti grafiche e i gestori degli eventi. Per esempio:

```
<Label fx:id="welcomeText"/>
```

```
<Button text="Hello!" onAction="#onHelloButtonClick"/>
```

dove onHelloButtonClick è il metodo definito nel Controller.

Per apportare modifiche al codice XML bisogna modificare direttamente il codice XML da IDE (o da SceneBuilder attraverso il pannello di destra).

NB: gli id permettono anche di agganciare le regole di un foglio stile ai singoli componenti dell'interfaccia grafica

Come aggiornare il codice dell'applicazione alla nuova GUI XML? - II



Nel Controller, inserite le dichiarazioni delle variabili globali da utilizzare, con le java annotation. I nomi delle variabili devono corrispondere agli ID specificati nella GUI (inoltre il controller può definire variabili locali). **Il controller inizializza automaticamente le variabili globali dichiarate** → esse possono essere usate nel metodo di gestione degli eventi o in altri metodi della classe. Es:

@FXML

```
private Label welcomeText;
```

@FXML

```
protected void onHelloButtonClick() {  
    welcomeText.setText("Welcome to JavaFX Application!");  
}
```