

Programmazione 3 e Laboratorio di Programmazione 3

Interfacce grafiche JavaFX

Proff. Angelo Ciaramella – Emanuel Di Nardo

GUI

- Graphical User interface (GUI)
 - Abstract Window Toolkit (AWT)
 - Swing
 - **JavaFX**



■ Diviso in tre sezioni:

■ Stage:

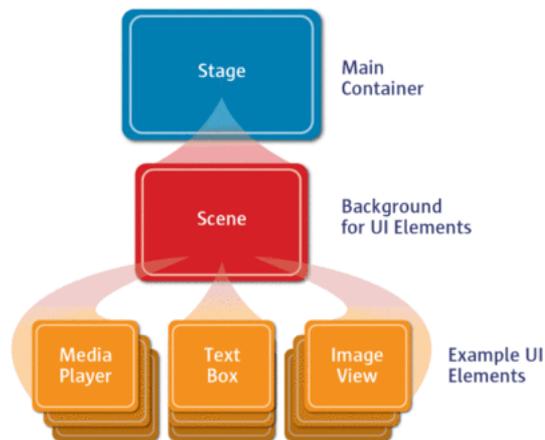
- Simili a JFrame, come la finestra e fa da super contenitore

■ Scene:

- Raggruppa tutti gli elementi grafici (nodi)

■ Grafo della scena (nodi)

- Singolo elemento all'interno della scena

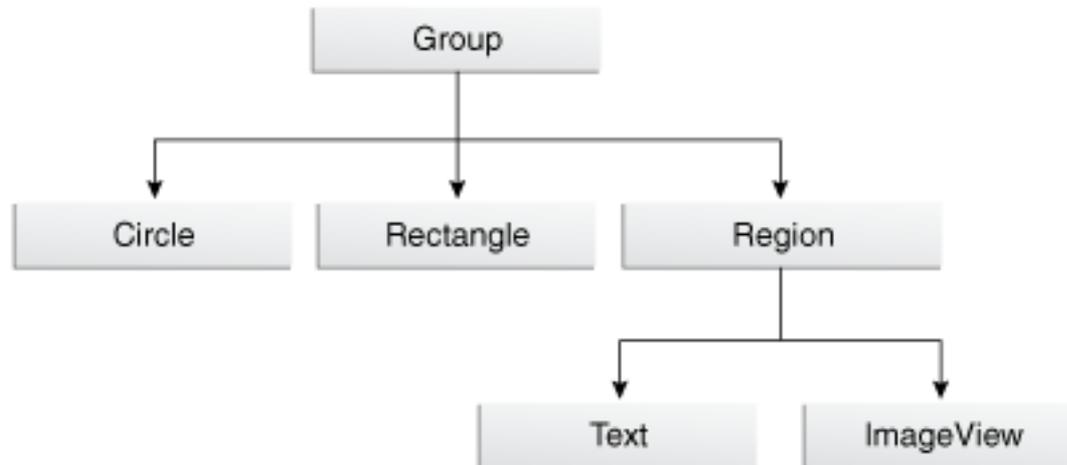


- Ogni applicazione è definita come **Application**
 - Bisogna estendere la classe Application
- In tale classe bisogna fare *override* del metodo ***start(Stage stage)***
- Lo *Stage* rappresenta la finestra
 - Gli elementi al suo interno sono le *Scene* con all'interno i componenti grafici



JavaFX - Rappresentazione ad albero

- Ogni elemento si comporta come un nodo di un albero
 - Per inserire nuovi elementi è necessario accedere alla lista dei nodi figlio (foglie) (`getChildren`) ed aggiungerne uno nuovo



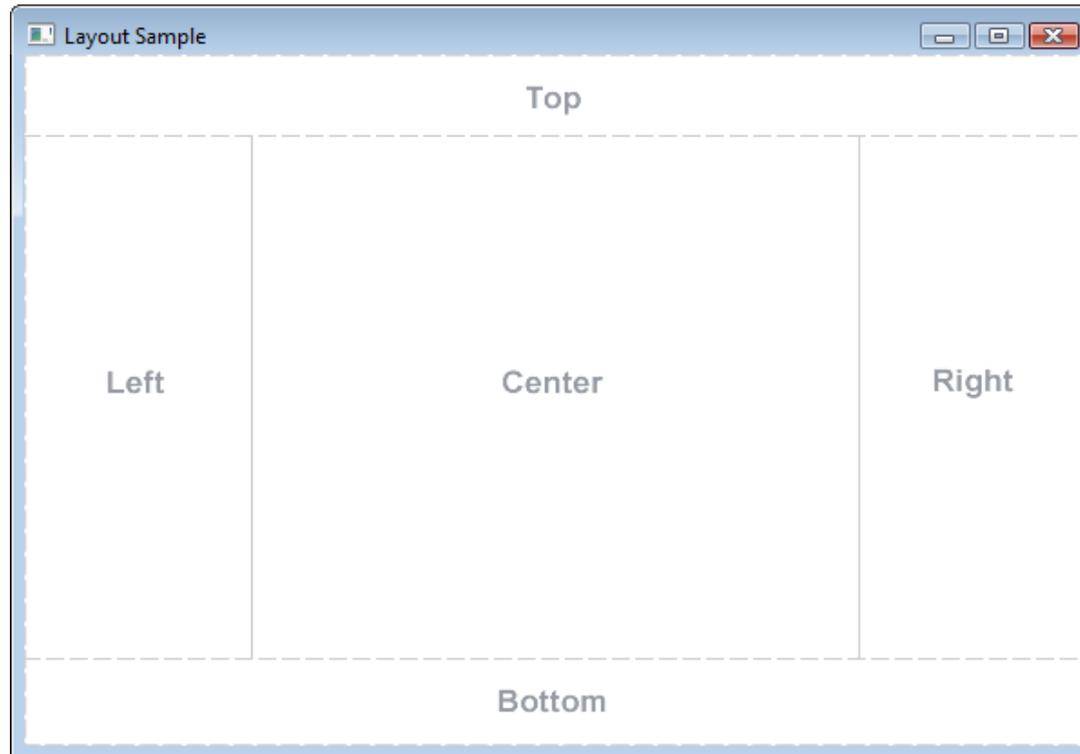
Layout Pane

- I layout permettono di avere una diversa distribuzione degli elementi grafici nel container
- I layout vengono chiamati *Pane*
- Alcuni Pane di default:
 - BorderLayout (default)
 - HBox
 - VBox
 - StackBox
 - GridBox
 - FlowPane
 - ...



BorderPane

- Gli elementi sono inseriti tramite posizionamento in 5 zone:
 - Left, Right, Top, Bottom, Center



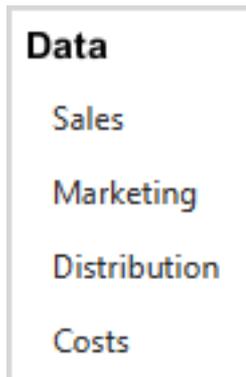
HBox, VBox

- Permettono di inserire gli elementi in varie direzioni:

- orizzontale (HBox)



- in verticale (VBox)



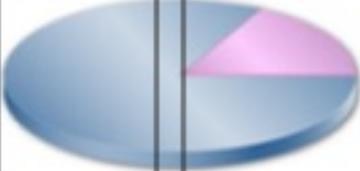
FlowPane

- Gli elementi sono inseriti da sinistra a destra in base all'ordine in cui vengono aggiunti
 - Al termine dello spazio su una riga, ne viene automaticamente aggiunta una seconda



GridPane

- Gli elementi sono inseriti come in una griglia (matrice):
 - ordinata da sinistra a destra e dall'alto verso il basso

	Sales:	Current Year	
	Goods and	Services	
Goods 80%			Services 20%



Creazione di GUI

- Le interfacce possono essere create in due modi:
 - Utilizzando un file xml di configurazione chiamato **FXML**
 - Deve essere caricato prima di creare la scena
 - Tramite codice



FXML

- Formato XML con una sintassi propria
- E' possibile importare librerie tramite il tag `<? ?>`
- Ogni tag è un elemento grafico o un suo modificatore
- Gli elementi grafici in fxml possono essere collegati a classi java che fanno da **Controller**
- Come accade per le classi, si possono collegare anche specifici metodi per determinare le *azioni* da fare (es. **Button**)



- Può essere modificato tramite SceneBuilder
- Il Parent principale ha una classe collegata chiamata Controller
 - attributo `fx:controller` (usare il package completo)
- Ogni Nodo può essere collegato con un oggetto JavaFX nel controller
 - Dichiarare la variabile ed aggiungere l'annotation `@FXML`
 - Nel FXML inserire `fx:id=nome_variabile`
- Si possono collegare metodi come azioni
 - Utilizzare l'annotation `@FXML`
 - Nel FXML inserire `onAction="#Azione»`
 - Il simbolo `#` è obbligatorio



- E' possibile stilizzare gli elementi utilizzando CSS
 - Creare un file css con all'interno le classi CSS
 - Assegnare il file alla scena

```
Scene scene = new Scene(borderPane) ;  
scene.getStylesheets().add("css/style.css") ;
```

- Aggiungere la classe CSS ad un componente

```
HBox hbox = new HBox() ;  
hbox.getStyleClass().add("hbox") ;
```

- Esempio di css:

```
.hbox {  
    padding-top: 15;  
    padding-bottom: 15;  
    background-color: #212121;  
    color: #ffffff;  
}
```



Esercizio - base

- Creiamo una calcolatrice come nell'immagine
 - Prova programmaticamente e poi con SceneBuilder



IDE Designer

- E' possibile disegnare interfacce in JavaFX usando un unico tool:
 - SceneBuilder
(<https://gluonhq.com/products/scene-builder/>)

