



Programmazione III

Prof.ssa Liliana Ardissono
Dipartimento di Informatica
Università di Torino

Interfacce Utente Grafiche (GUI)
JavaFX – note tecniche (installazione etc.)



Questa presentazione è distribuita sotto licenza Creative Commons CC BY ND

JavaFX - II



- Tutorials:
https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
- Documentazione: <http://docs.oracle.com/javafx/2/>
- API: <https://openjfx.io/javadoc/13/>



JavaFX – Installazione e uso

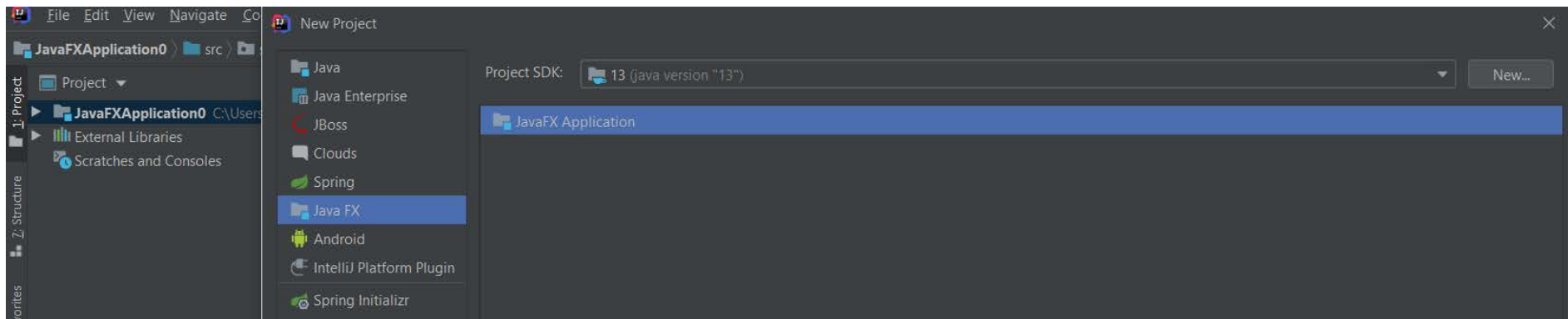
JavaFX deve essere installato come pacchetto di librerie esterno. Seguono le istruzioni:

- **Download:** <https://openjfx.io/index.html> - scaricate la cartella decompressa sotto IdeaProjects, per comodità.
 - **Installazione:**
 - Le librerie (file .jar) devono essere inserite nei progetti JavaFX che necessitano di utilizzarle.
 - Una volta caricate le librerie il progetto JavaFX compila correttamente ma potrebbero esserci problemi di esecuzione. In tal caso seguire le istruzioni fornite alla pagina <https://openjfx.io/openjfx-docs/>
- I prossimi lucidi mostrano queste istruzioni.

Passo 1: creare un progetto JavaFX - I



Con IntelliJ IDEA: File → new project → Java FX
Viene creato un progetto di esempio, dobbiamo far funzionare questo progetto per considerare l'ambiente funzionante.





Passo 1: creare un progetto JavaFX - II

Notate che nel codice del progetto ci sono errori di compilazione: mancano le librerie di JavaFX

The screenshot shows an IDE window for a project named 'JavaFXApplication00'. The main editor displays the 'Main.java' file. The code is as follows:

```
1 package sample;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Main extends Application {
10
11     @Override
12     public void start(Stage primaryStage) throws Exception{
13         Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
14         primaryStage.setTitle("Hello World");
15         primaryStage.setScene(new Scene(root, 300, 275));
16         primaryStage.show();
17     }
18
19
20 public static void main(String[] args) { launch(args); }
21
22 }
23
24
```

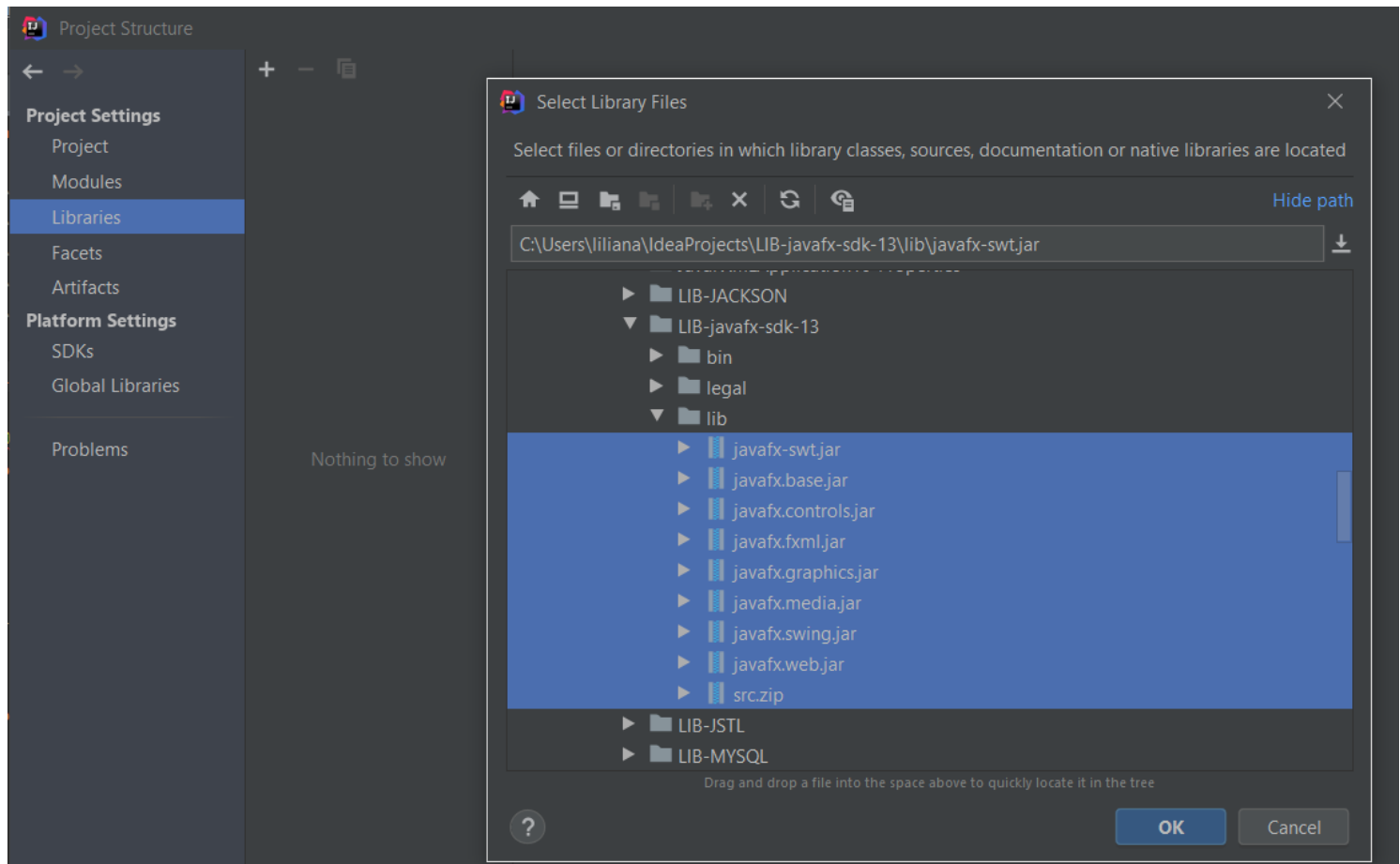
On the right side of the editor, there is a vertical list of red error markers corresponding to each line of code, indicating that the JavaFX classes (Application, FXMLLoader, Parent, Scene, Stage) cannot be found. The IDE interface includes a menu bar at the top, a project structure view on the left, and a terminal/console area at the bottom.



Passo 2: caricare le librerie di JavaFX - I

File → Project Structure → Libraries → + → Java → scendere nella cartella di JavaFX fino a lib e selezionare tutti i .jar:

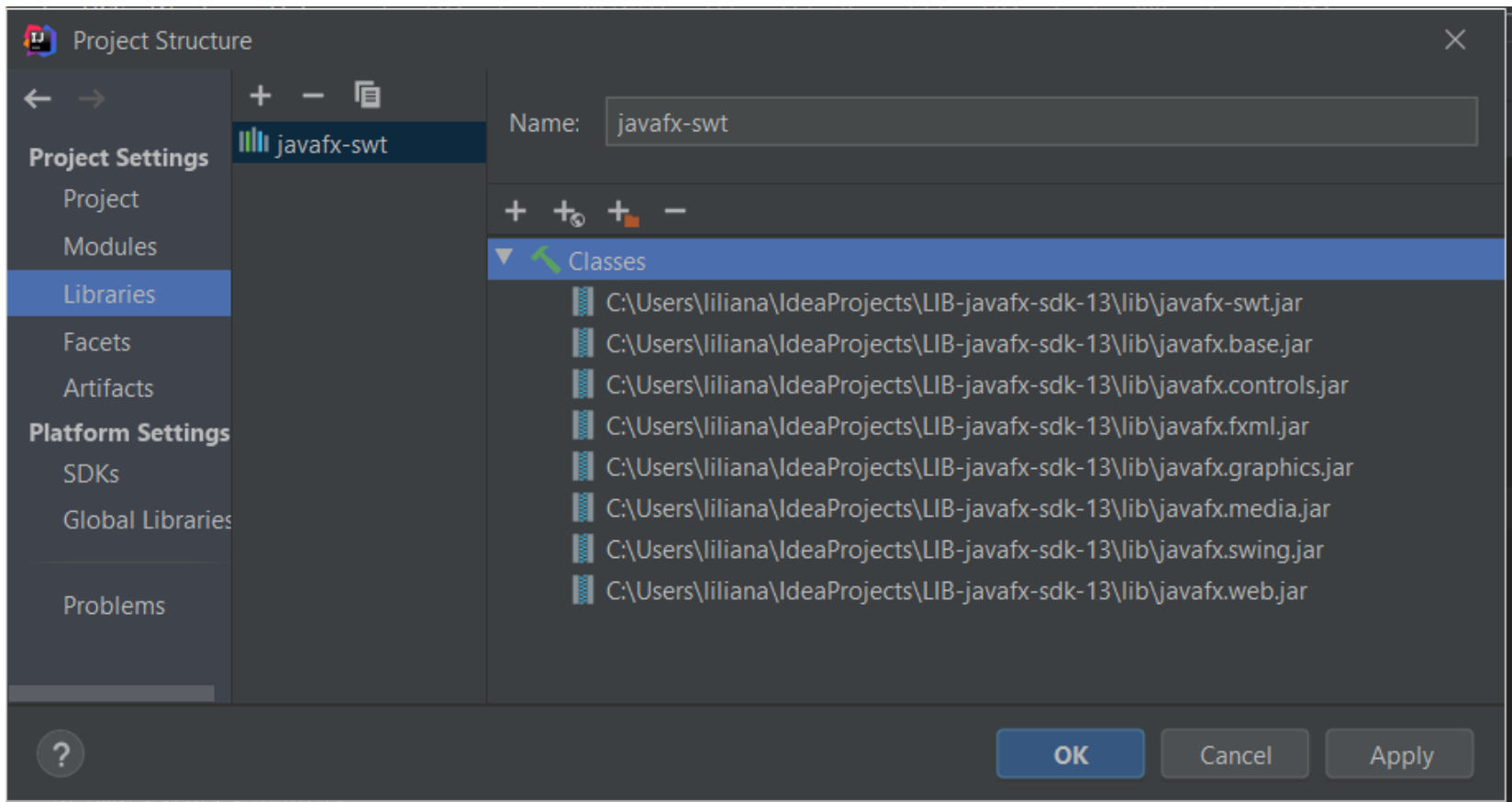
→ OK



Passo 2: caricare le librerie di JavaFX - II



Cliccare prima Apply (per applicare le librerie), poi OK
→ si risolvono i problemi di compilazione



Passo 2: caricare le librerie di JavaFX - III



Potrebbero esserci ancora problemi nell'esecuzione dell'applicazione

The screenshot shows the IntelliJ IDEA IDE with a project named 'JavaFXApplication00'. The main class 'Main.java' is open, showing the following code:

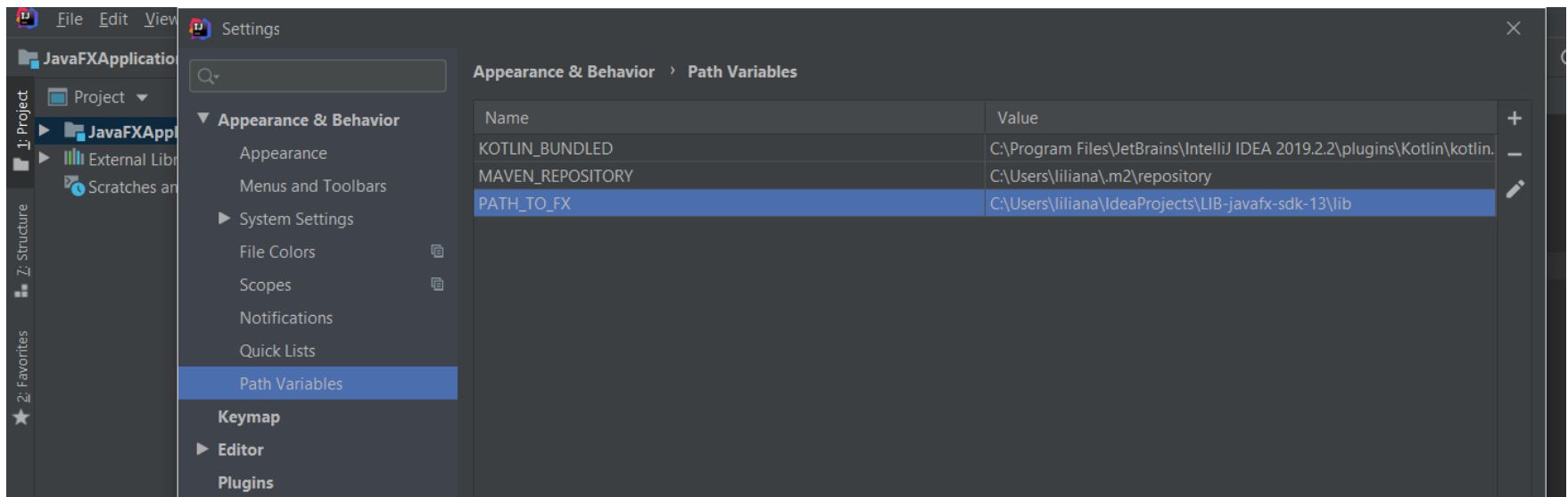
```
1 package sample;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Main extends Application {
10
11     @Override
12     public void start(Stage primaryStage) throws Exception{
13         Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
14         primaryStage.setTitle("Hello World");
15         primaryStage.setScene(new Scene(root, 300, 275));
16         primaryStage.show();
17     }
18 }
```

The Run window at the bottom shows the command executed: `"C:\Program Files\Java\jdk-13\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2019.2.2\lib\idea_rt.jar=50396:C:\Program Files\JetBrains\IntelliJ IDEA 2019.2.2\bin\idea_rt.jar" -jar sample.jar`. The output shows an error: `Error: JavaFX runtime components are missing, and are required to run this application`. The process finished with exit code 1.



Passo 3: settare la variable di ambiente PATH_TO_FX

Come consigliato in <https://openjfx.io/openjfx-docs/>, aprire File → Settings → Appearance & Behavior → Path variables e definire la variable PATH_TO_FX che punta alla cartella lib della libreria di JavaFX che avete scaricato, click Apply e OK



Passo 4: settare le VM options per usare la variabile

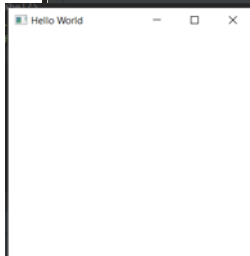
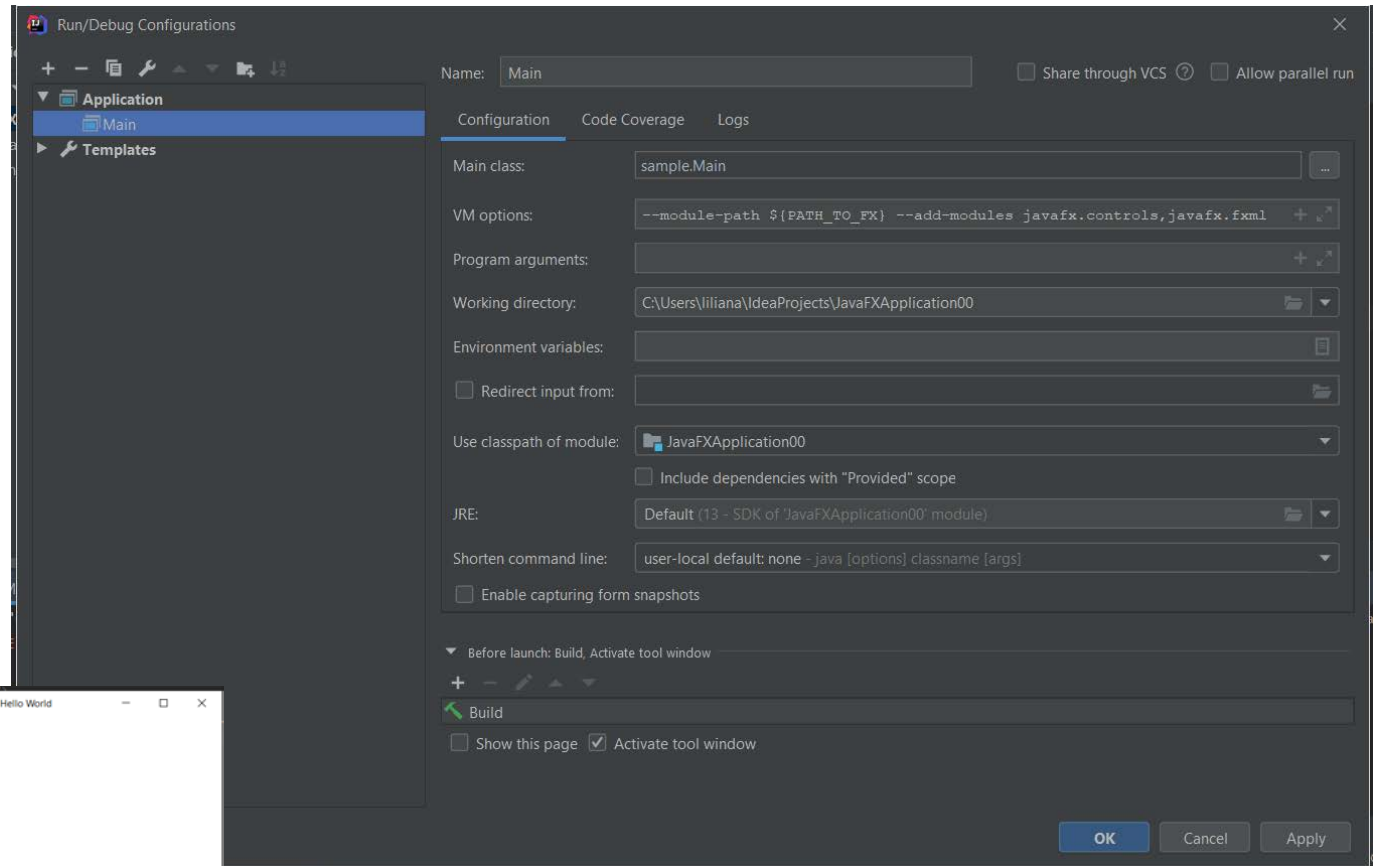
Run → Edit Configurations → settare **VM options** al path



**--module-path \${PATH_TO_FX} --add-modules
javafx.controls,javafx.fxml**

Apply
OK

A questo punto
l'applicazione
funziona e apre
una finestra
vuota



JavaFX – nuovi progetti



Per ogni nuovo progetto che utilizza JavaFX dovete:

- Aggiungere le librerie di JavaFX
- Settare il path alle librerie di JavaFX nella virtual machine tramite Run → Edit configurations → VM options:

`--module-path ${PATH_TO_FX}`

`--add-modules javafx.controls,javafx.fxml`

- NB: la variable `PATH_TO_FX` rimane disponibile → Non va ridefinita.