



Royal University of Bhutan



Unit VIII

JavaFX and Creating JAR File

Tutor: Pema Galey

Learning Outcomes

In this session, you will learn about:

- What is JavaFX?
- JavaFX Features
- JavaFX Classes

What is JavaFX?

- JavaFX is an open source, next generation client application platform for desktop, mobile and embedded systems built on Java.
- It is a collaborative effort by many individuals and companies with the goal of producing a modern, efficient, and fully featured toolkit for developing rich client applications.

JavaFX Features

- Java Library
- FXML
- Web view
- CSS like styling
- Swing interoperability
- Canvas API
- Rich Set of APIs

JavaFX Classes

- 2D Shapes
- Text
- Effects
- Transformation
- Animation
- 3D Shapes
- Layouts
- UI
- Charts
- CSS
- Media
- Event Handling

JavaFX Application Structure

- JavaFX application is divided hierarchically into three main components known as Stage, Scene and nodes. We need to import `javafx.application.Application` class in every JavaFX application. This provides the following life cycle methods for JavaFX application.
 - `public void init()`
 - `public abstract void start(Stage primaryStage)`
 - `public void stop()`

JavaFX Application Structure

In order to create a basic JavaFX application, we need to:

1. Import **javafx.application.Application** into our code.
2. Inherit **Application** into our class.
3. Override **start()** method of Application class.

Sample Code

```
package application;  
import javafx.application.Application;  
import javafx.stage.Stage;  
public class Hello_World extends Application{  
  
    @Override  
    public void start(Stage primaryStage) throws Exception {  
        // TODO Auto-generated method stub  
  
    }  
  
}
```

References for JavaFX

- <https://openjfx.io>
- <https://www.javatpoint.com/javafx-tutorial>

JAR File Creation

1. Ensure that all necessary files are within the directory, you have opened a terminal/Command Prompt and have navigated to that directory.
2. Compile the .java class, for example HelloWorld.java with:

`javac HelloWorld.java`

3. This will produce a .class file needed for the JAR file.
4. Next create a manifest file (saved using the extension .txt) using the text editor and input the following:

`Main-Class: HelloWorld`

or whatever your class file name is. *Keep newline after end of your class name inside the manifest file*

JAR File Creation

5) Next create the JAR file using this code:

```
jar cfm HelloWorld.jar Manifest.txt HelloWorld.class
```

6) Run the file:

```
java -jar HelloWorld.jar
```

OR

Double click on JAR file to run the application if your system supports

Thank you!