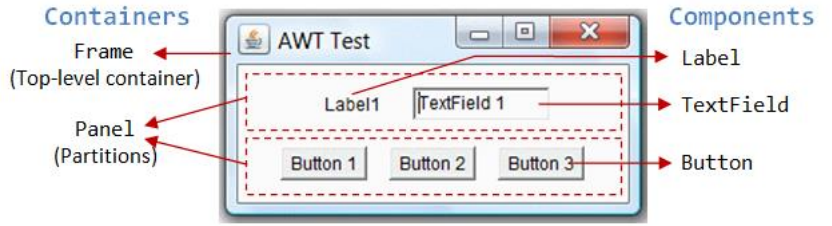
Programming Languages - II
Visual Programming-JavaFX

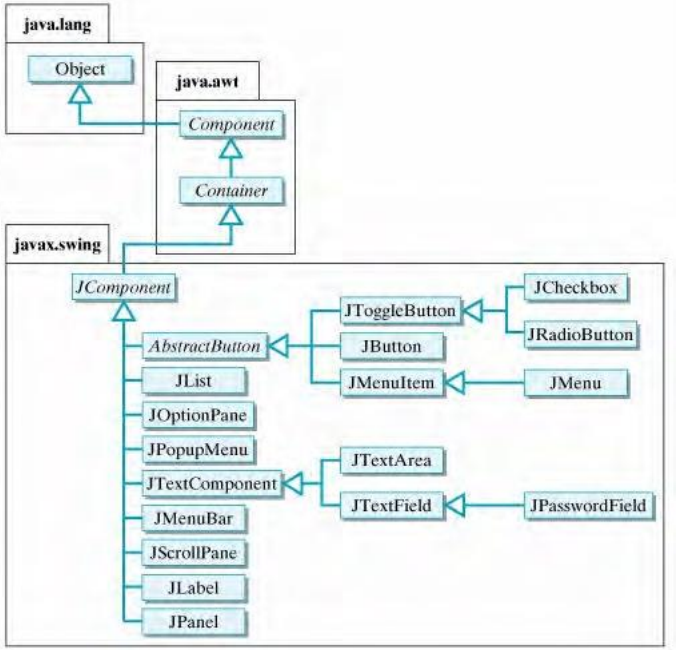
Özgür Koray ŞAHİNGÖZ
Prof.Dr.

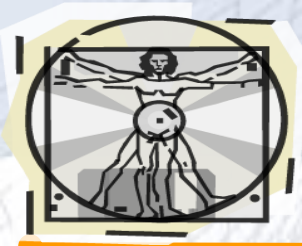
Biruni University
Computer Engineering Department

Visual Programming in Java

- Visual Programming Environment in Java
  - Abstract Windows Toolkit (AWT),
  - Swing,
  - JavaFX, Java 8'den sonra.

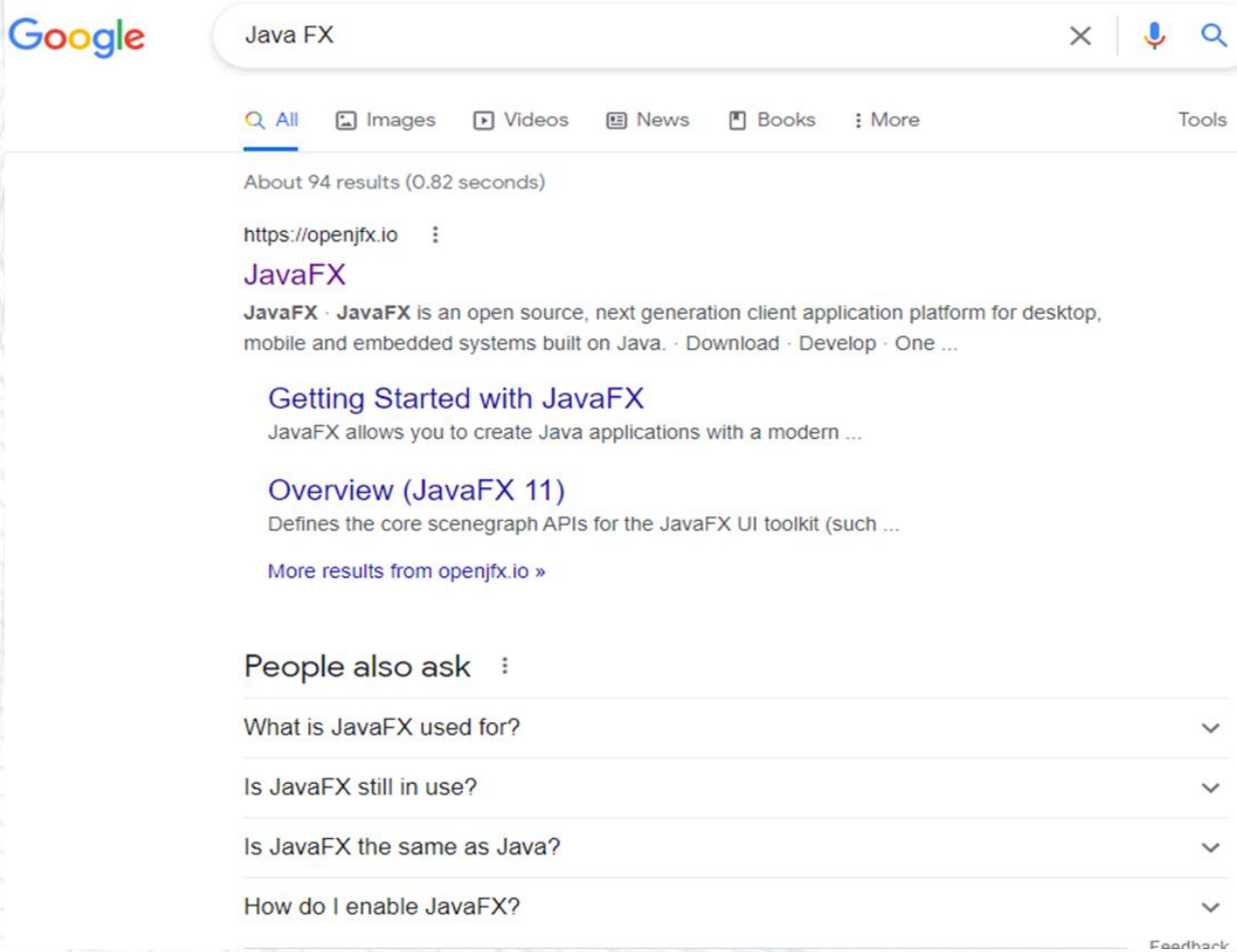


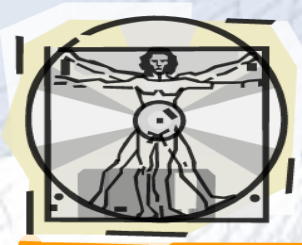




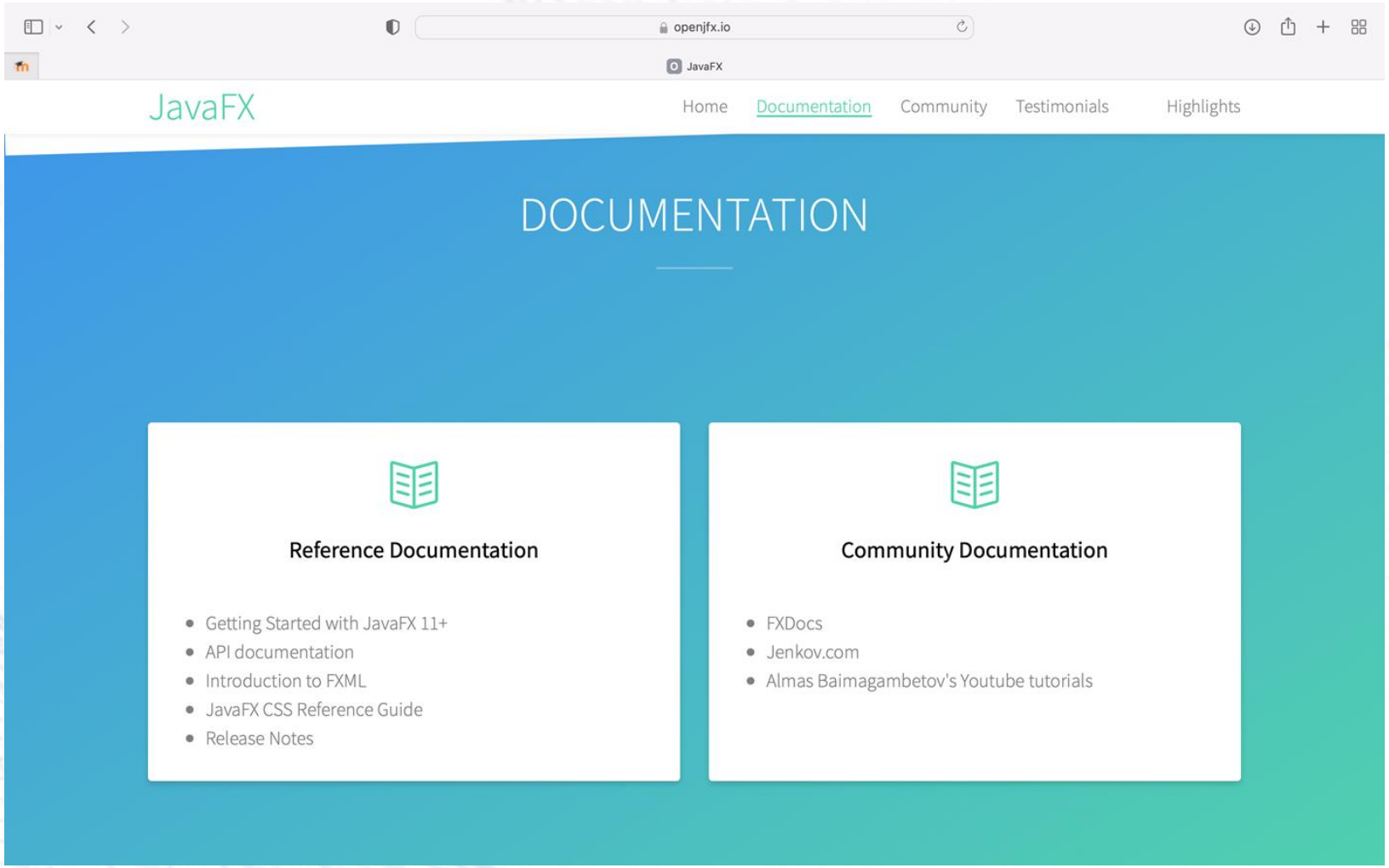
Visual Programming in Java

Setting Up JavaFX-Visit Web Page





JavaFXSetting Up JavaFX-Visit Web Page



GLUON

Products Developers Pricing Services Insights Contact

Linux	18.0.1	arm32	SDK	Download [SHA256]
Linux	18.0.1	x64	SDK	Download [SHA256]
Linux	18.0.1	x64	jmods	Download [SHA256]
macOS	18.0.1	aarch64	SDK	Download [SHA256]
macOS	18.0.1	aarch64	jmods	Download [SHA256]
macOS	18.0.1	x64	SDK	Download [SHA256]
macOS	18.0.1	x64	jmods	Download [SHA256]
Windows	18.0.1	x64	SDK	Download [SHA256]
Windows	18.0.1	x64	jmods	Download [SHA256]
Windows	18.0.1	x86	SDK	Download [SHA256]
Windows	18.0.1	x86	jmods	Download [SHA256]
Javadoc	18.0.1		Javadoc	Download [SHA256]

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.

Download

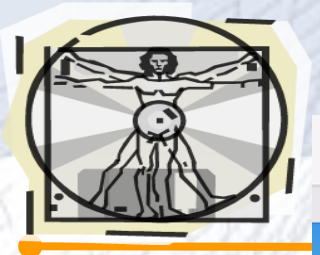
JavaFX runtime is available as a platform-specifc SDK, as a number of jmods, and as a set of artifacts in Maven Central.

DOWNLOAD

Develop

JavaFX, also known as OpenJFX, is free software; licensed under the GPL with the class path exception, just like the OpenJDK.

LET'S DO IT!



openjfx.io

JavaFX | Getting Started with JavaFX

JavaFX

Getting Started with JavaFX

- Introduction
- Install Java
- Run HelloWorld using JavaFX
- Run HelloWorld via Maven
- Run HelloWorld via Gradle
- Runtime images
- JavaFX and IntelliJ
- JavaFX and NetBeans
- JavaFX and Eclipse
  - Non-modular from IDE
  - Non-modular with Maven
  - Non-modular with Gradle
  - Modular from IDE
  - Modular with Maven
  - Modular with Gradle
- JavaFX and Visual Studio Code
- Next Steps

JavaFX and Eclipse

This section explains how to create a JavaFX application in Eclipse. JavaFX 15.0.1 and Eclipse 2020-12 (4.18.0) were used for the IDE screenshots.

Download an appropriate JDK for your operating system and set `JAVA_HOME` to the JDK directory. Refer to Install Java section for more information.

Include the new JDK as `Installed JREs` in `Eclipse -> Preferences -> Java -> Installed JREs -> Add`.

You can create a JavaFX modular or non-modular project and use the IDE tools, Maven or Gradle build tools.

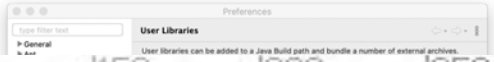
Non-modular projects

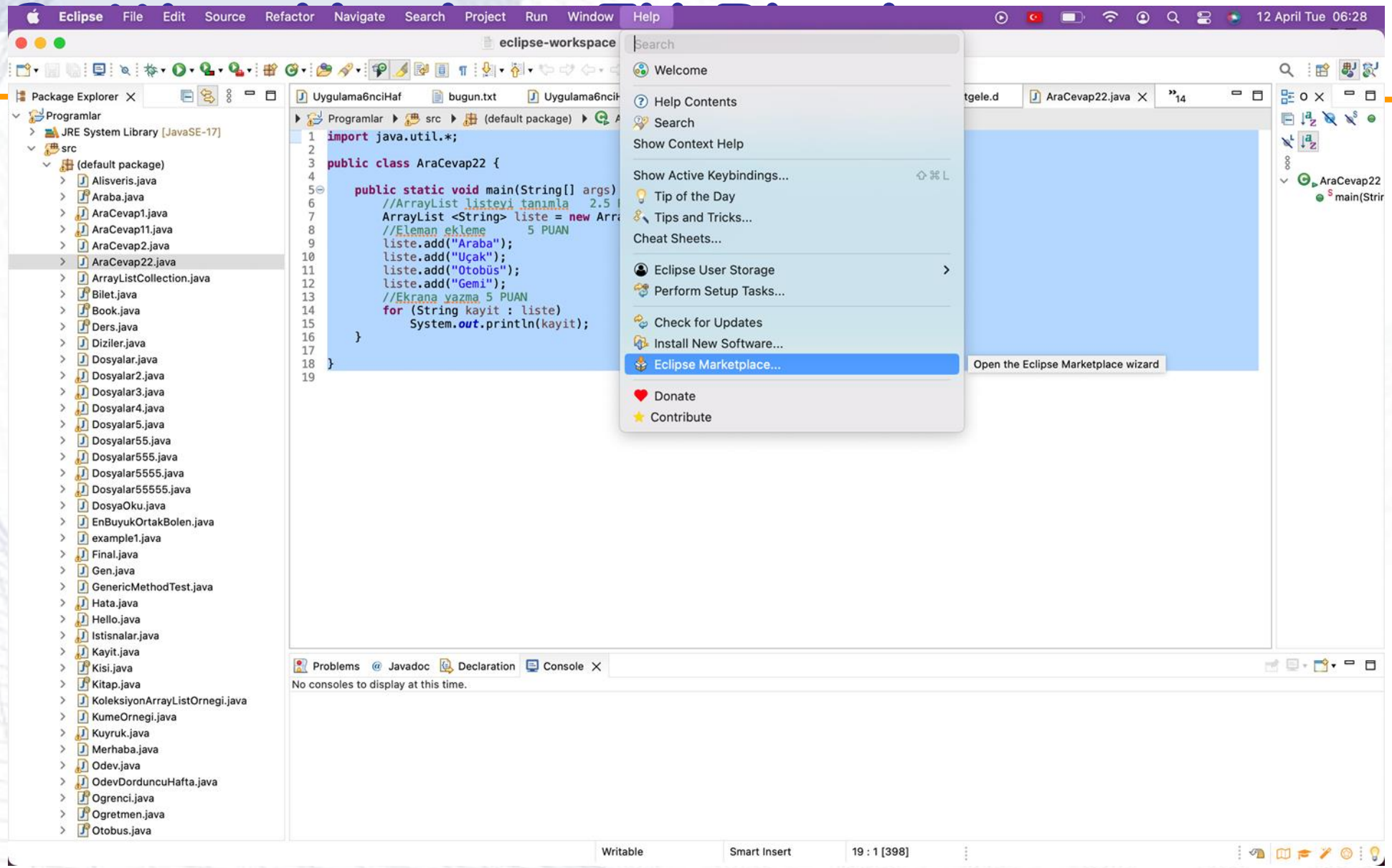
IDE

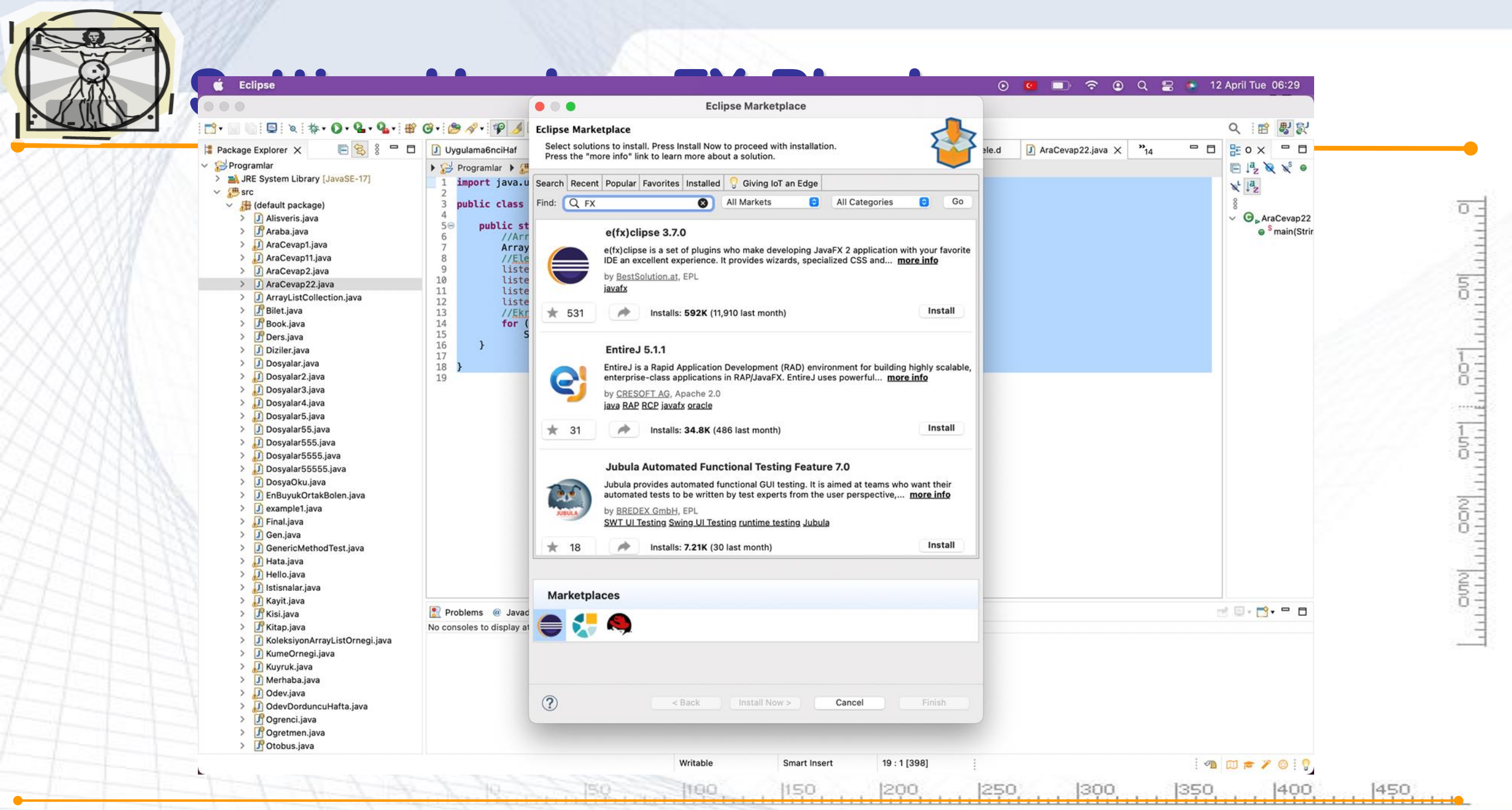
Follow these steps to create a JavaFX non-modular project and use the IDE tools to build it and run it. Alternatively, you can download a similar project from here.

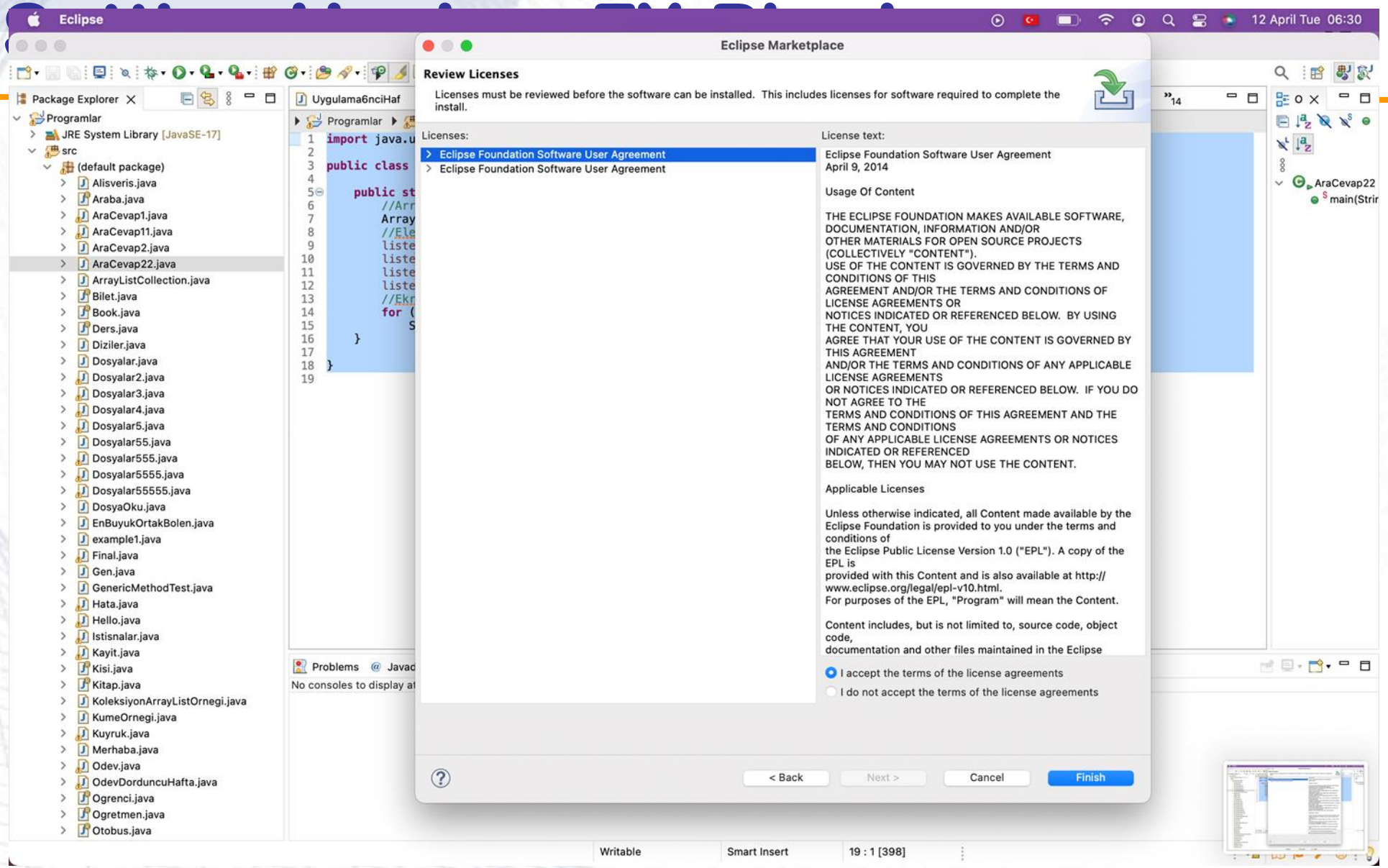
Download the appropriate JavaFX SDK for your operating system and unzip it to a desired location, for instance `/Users/your-user/Downloads/javafx-sdk-17.0.1`.

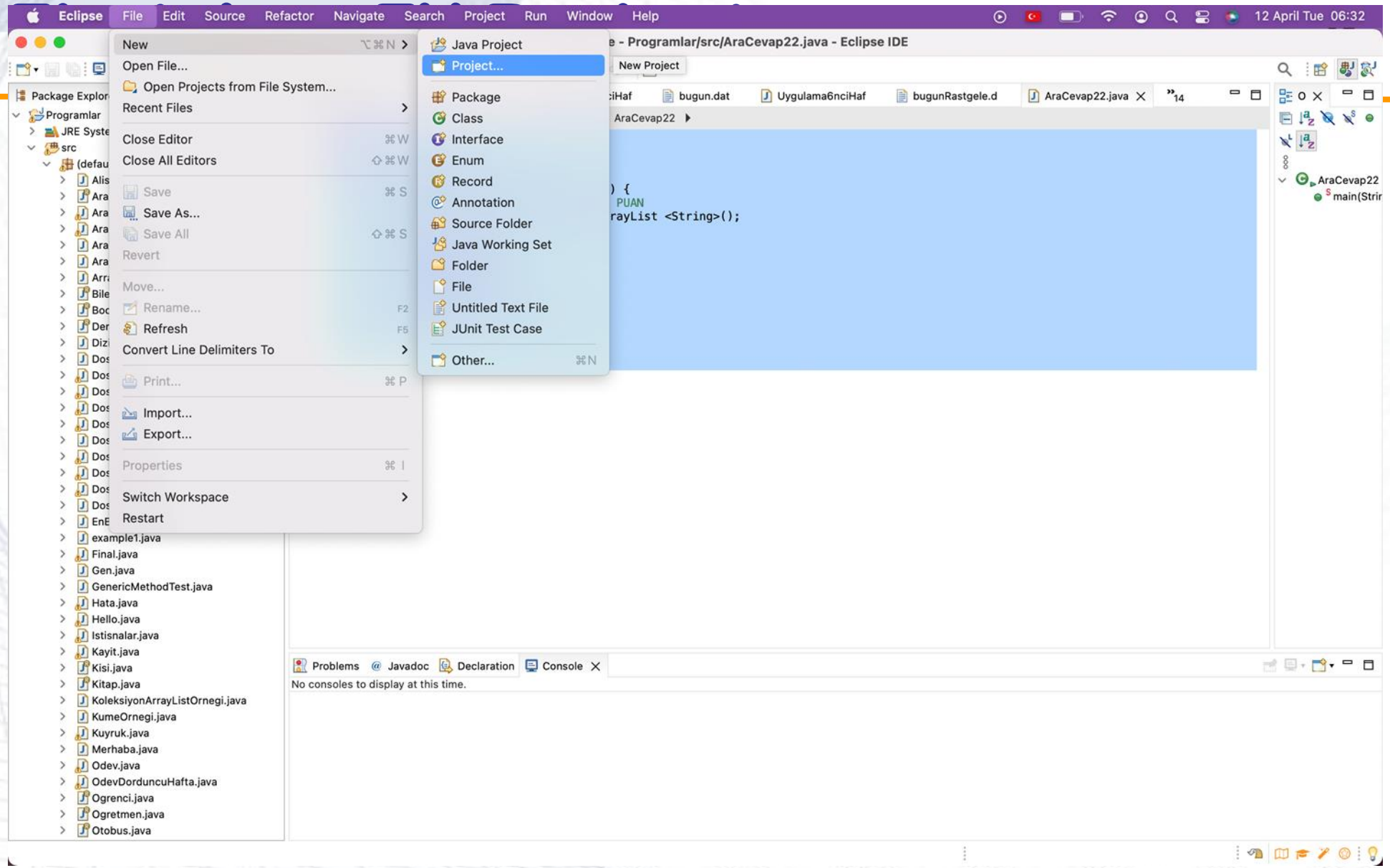
Create a new `User Library` under `Eclipse -> Window -> Preferences -> Java -> Build Path -> User Libraries -> New`.



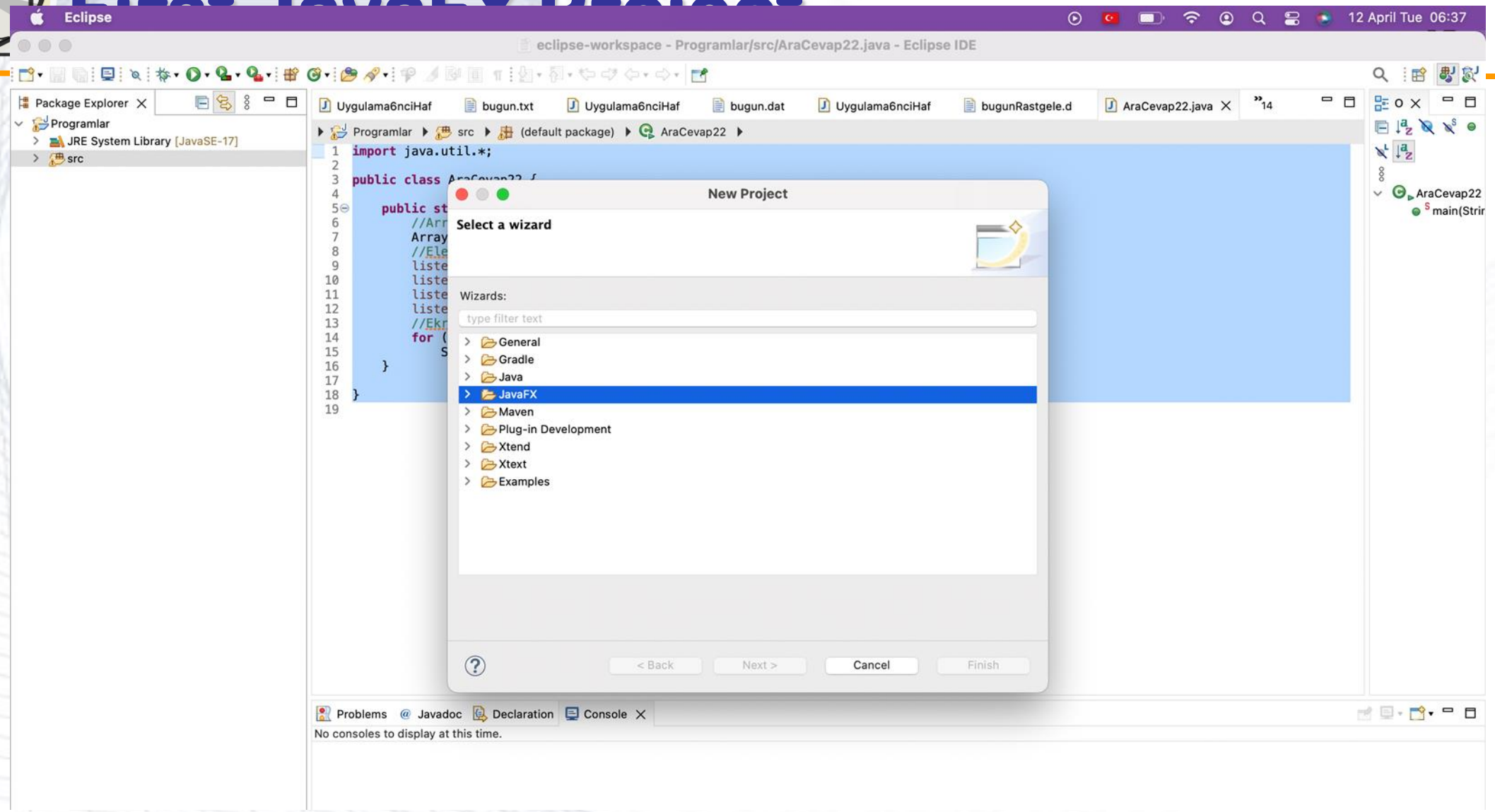

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
12 April Tue 06:28
eclipse-workspace
Package Explorer
Programlar
JRE System Library [JavaSE-17]
src
(default package)
Alisveris.java
Araba.java
AraCevap1.java
AraCevap11.java
AraCevap2.java
AraCevap22.java
ArrayListCollection.java
Bilet.java
Book.java
Ders.java
Diziler.java
Dosyalar.java
Dosyalar2.java
Dosyalar3.java
Dosyalar4.java
Dosyalar5.java
Dosyalar55.java
Dosyalar555.java
Dosyalar5555.java
Dosyalar55555.java
DosyaOku.java
EnBuyukOrtakBolen.java
example1.java
Final.java
Gen.java
GenericMethodTest.java
Hata.java
Hello.java
Istisnalar.java
Kayit.java
Kisi.java
Kitap.java
KoleksiyonArrayListOrnegi.java
KumeOrnegi.java
Kuyruk.java
Merhaba.java
Odev.java
OdevDorduncuHafta.java
Ogrenci.java
Ogretmen.java
Otobus.java
UygulamaGnciHaf
bugun.txt
AraCevap22.java
Programlar src (default package)
1 import java.util.*;
2
3 public class AraCevap22 {
4
5 public static void main(String[] args)
6 //ArrayList listeyi tanımla 2.5
7 ArrayList <String> liste = new Arra
8 //Eleman ekleme 5 PUAN
9 liste.add("Araba");
10 liste.add("Uçak");
11 liste.add("Otobüs");
12 liste.add("Gemi");
13 //Ekrana yazma 5 PUAN
14 for (String kayit : liste)
15 System.out.println(kayit);
16 }
17
18 }
19
Search
Welcome
Help Contents
Search
Show Context Help
Show Active Keybindings...
Tip of the Day
Tips and Tricks...
Cheat Sheets...
Eclipse User Storage
Perform Setup Tasks...
Check for Updates
Install New Software...
Eclipse Marketplace...
Open the Eclipse Marketplace wizard
Donate
Contribute
AraCevap22
main(Strir
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable
Smart Insert
19 : 1 [398]

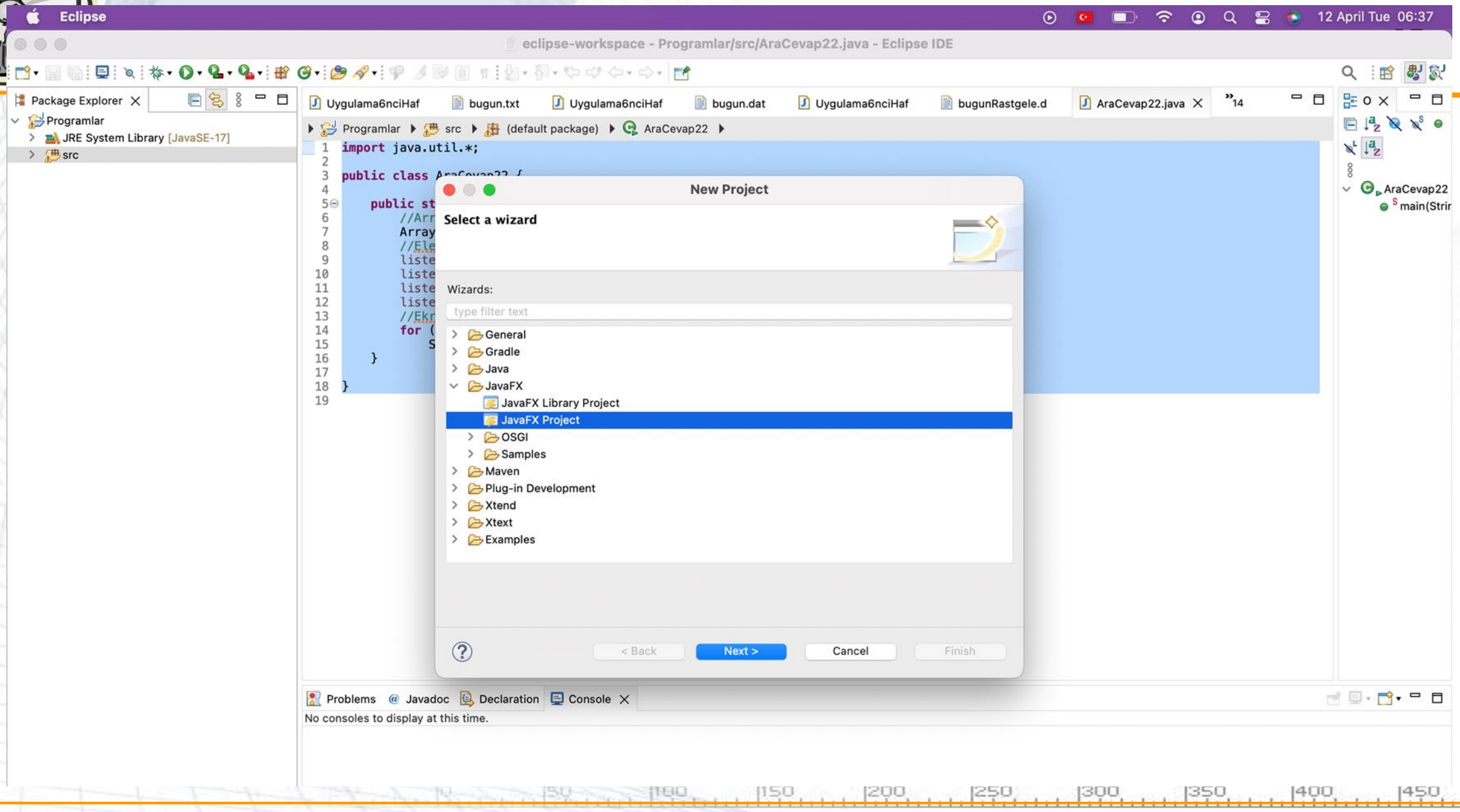

Eclipse
12 April Tue 06:29
Eclipse Marketplace
Eclipse Marketplace
Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.
Search
Recent
Popular
Favorites
Installed
Giving IoT an Edge
Find: FX
All Markets
All Categories
Go
e(fx)clipse 3.7.0
e(fx)clipse is a set of plugins who make developing JavaFX 2 application with your favorite IDE an excellent experience. It provides wizards, specialized CSS and... more info
by BestSolution.at, EPL
javafx
531
Installs: 592K (11,910 last month)
Install
EntireJ 5.1.1
EntireJ is a Rapid Application Development (RAD) environment for building highly scalable, enterprise-class applications in RAP/JavaFX. EntireJ uses powerful... more info
by CRESOFT AG, Apache 2.0
java RAP RCP javafx oracle
31
Installs: 34.8K (486 last month)
Install
Jubula Automated Functional Testing Feature 7.0
Jubula provides automated functional GUI testing. It is aimed at teams who want their automated tests to be written by test experts from the user perspective,... more info
by BREDEX GmbH, EPL
SWT UI Testing Swing UI Testing runtime testing Jubula
18
Installs: 7.21K (30 last month)
Install
Marketplaces
< Back
Install Now >
Cancel
Finish
Package Explorer
Programlar
JRE System Library [JavaSE-17]
src
(default package)
Alisveris.java
Araba.java
AraCevap1.java
AraCevap11.java
AraCevap2.java
AraCevap22.java
ArrayListCollection.java
Bilet.java
Book.java
Ders.java
Diziler.java
Dosyalar.java
Dosyalar2.java
Dosyalar3.java
Dosyalar4.java
Dosyalar5.java
Dosyalar55.java
Dosyalar555.java
Dosyalar5555.java
Dosyalar55555.java
DosyaOku.java
EnBuyukOrtakBolen.java
example1.java
Final.java
Gen.java
GenericMethodTest.java
Hata.java
Hello.java
Istisnalar.java
Kayit.java
Kisi.java
Kitap.java
KoleksiyonArrayListOrnegi.java
KumeOrnegi.java
Kuyruk.java
Merhaba.java
Odev.java
OdevDorduncuHafta.java
Ogrenci.java
Ogretmen.java
Otobus.java
UygulamaGnciHaf
AraCevap22.java
AraCevap22
main(Strir
Problems
No consoles to display at
Writable
Smart Insert
19 : 1 [398]


Eclipse
12 April Tue 06:30
Eclipse Marketplace
Review Licenses
Licenses must be reviewed before the software can be installed. This includes licenses for software required to complete the install.
Licenses:
Eclipse Foundation Software User Agreement
Eclipse Foundation Software User Agreement
License text:
Eclipse Foundation Software User Agreement
April 9, 2014
Usage Of Content
THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT").
USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT.
Applicable Licenses
Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is also available at http://www.eclipse.org/legal/epl-v10.html.
For purposes of the EPL, "Program" will mean the Content.
Content includes, but is not limited to, source code, object code, documentation and other files maintained in the Eclipse
I accept the terms of the license agreements
I do not accept the terms of the license agreements
< Back
Next >
Cancel
Finish
Package Explorer
Programlar
JRE System Library [JavaSE-17]
src
(default package)
Alisveris.java
Araba.java
AraCevap1.java
AraCevap11.java
AraCevap2.java
AraCevap22.java
ArrayListCollection.java
Bilet.java
Book.java
Ders.java
Diziler.java
Dosyalar.java
Dosyalar2.java
Dosyalar3.java
Dosyalar4.java
Dosyalar5.java
Dosyalar55.java
Dosyalar555.java
Dosyalar5555.java
Dosyalar55555.java
DosyaOku.java
EnBuyukOrtakBolen.java
example1.java
Final.java
Gen.java
GenericMethodTest.java
Hata.java
Hello.java
Istisnalar.java
Kayit.java
Kisi.java
Kitap.java
KoleksiyonArrayListOrnegi.java
KumeOrnegi.java
Kuyruk.java
Merhaba.java
Odev.java
OdevDorduncuHafta.java
Ogrenci.java
Ogretmen.java
Otobus.java
UygulamaGnciHaf
Problems
No consoles to display at
Writable
Smart Insert
19 : 1 [398]
AraCevap22
main(Strir

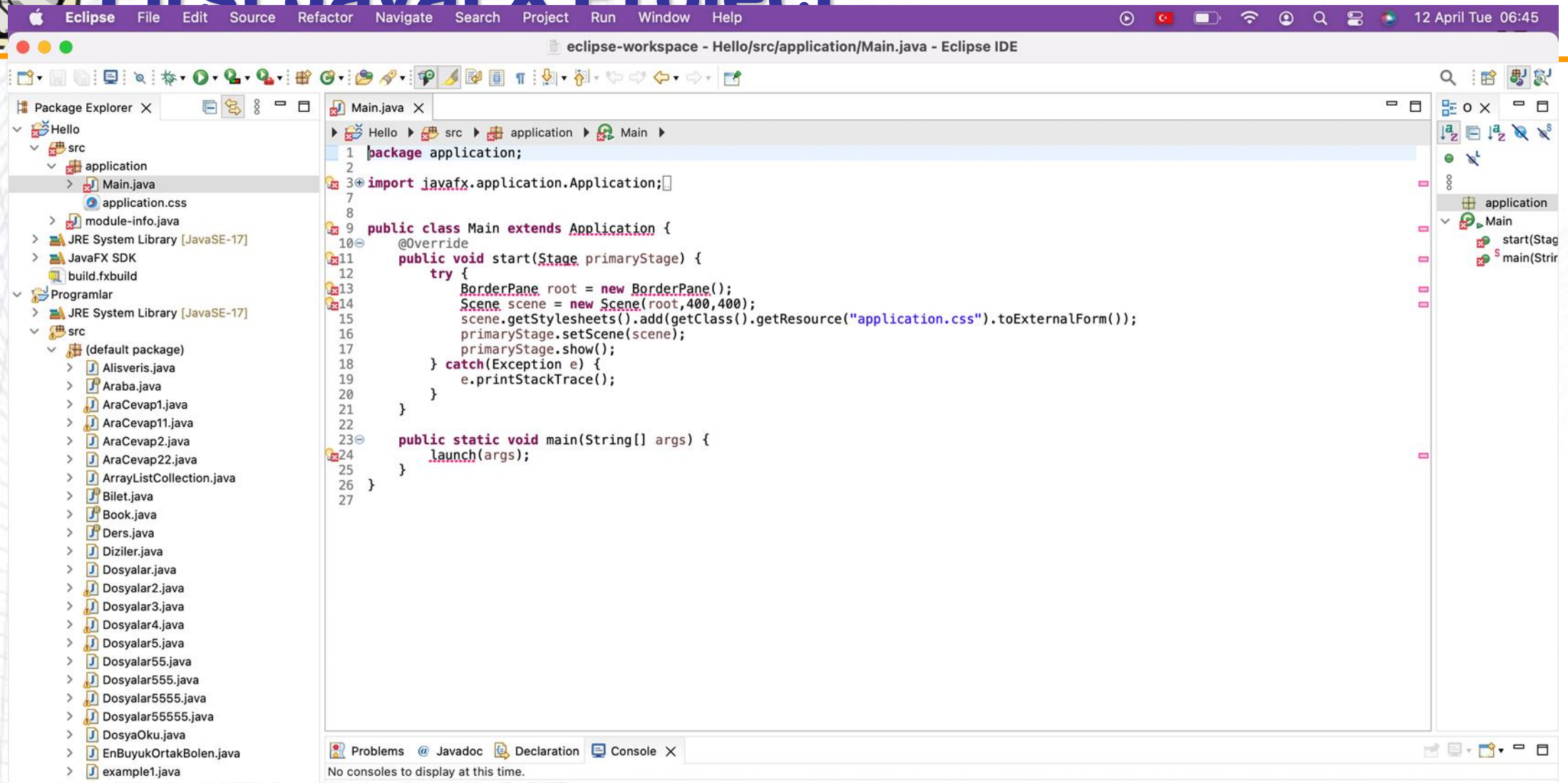

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
12 April Tue 06:32
New ⌥⌘N
Open File...
Open Projects from File System...
Recent Files
Close Editor ⌘W
Close All Editors ⇧⌘W
Save ⌘S
Save As...
Save All ⇧⌘S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... ⌘P
Import...
Export...
Properties ⌘I
Switch Workspace
Restart
Java Project
Project...
Package
Class
Interface
Enum
Record
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Other... ⌘N
New Project
bugun.dat
Uygulama6nciHaf
bugunRastgele.d
AraCevap22.java
AraCevap22
Problems Javadoc Declaration Console
No consoles to display at this time.
example1.java
Final.java
Gen.java
GenericMethodTest.java
Hata.java
Hello.java
Istisnalar.java
Kayit.java
Kisi.java
Kitap.java
KoleksiyonArrayListOrnegi.java
KumeOrnegi.java
Kuyruk.java
Merhaba.java
Odev.java
OdevDorduncuHafta.java
Ogrenci.java
Ogretmen.java
Otobus.java

First JavaFX Project




Eclipse
12 April Tue 06:37
eclipse-workspace - Programlar/src/AraCevap22.java - Eclipse IDE
Package Explorer
Programlar
JRE System Library [JavaSE-17]
src
UygulamaGnciHaf
bugun.txt
UygulamaGnciHaf
bugun.dat
UygulamaGnciHaf
bugunRastgele.d
AraCevap22.java
Programlar
src
(default package)
AraCevap22
import java.util.*;
public class
public st
Array
liste
liste
liste
liste
for (
}
}
New Project
Select a wizard
Wizards:
type filter text
General
Gradle
Java
JavaFX
JavaFX Library Project
JavaFX Project
OSGi
Samples
Maven
Plug-in Development
Xtend
Xtext
Examples
< Back
Next >
Cancel
Finish
AraCevap22
main(Strir
Problems
Javadoc
Declaration
Console
No consoles to display at this time.

First JavaFX Project



First JavaFX Project

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help — 12 April Tue 06:47

/src/application/Main.java - Eclipse IDE

Open Project
Close Project
Build All ⌘B
Build Project
Build Working Set
Clean...
✓ Build Automatically
Generate Javadoc...
Properties

Package Explorer

- Hello
  - src
    - application
      - Main.java
      - application.css
    - module-info.java
  - JRE System Library [JavaSE-17]
  - JavaFX SDK
  - build.fxbuild
- Programlar
  - JRE System Library [JavaSE-17]
  - src
    - (default package)
      - Alisveris.java
      - Araba.java
      - AraCevap1.java
      - AraCevap11.java
      - AraCevap2.java
      - AraCevap22.java
      - ArrayListCollection.java
      - Bilet.java
      - Book.java
      - Ders.java
      - Diziler.java
      - Dosyalar.java
      - Dosyalar2.java
      - Dosyalar3.java
      - Dosyalar4.java
      - Dosyalar5.java

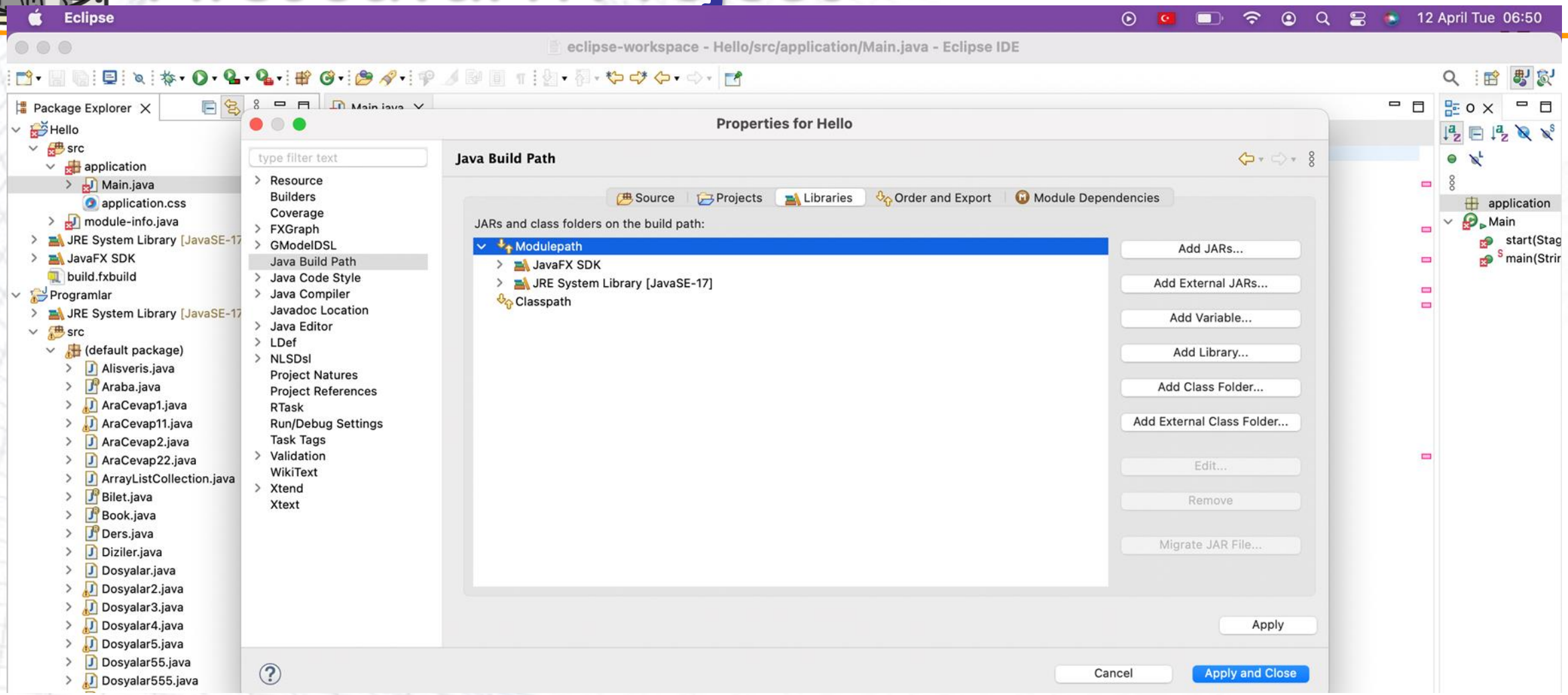
Main.java

Hello ▸ src ▸

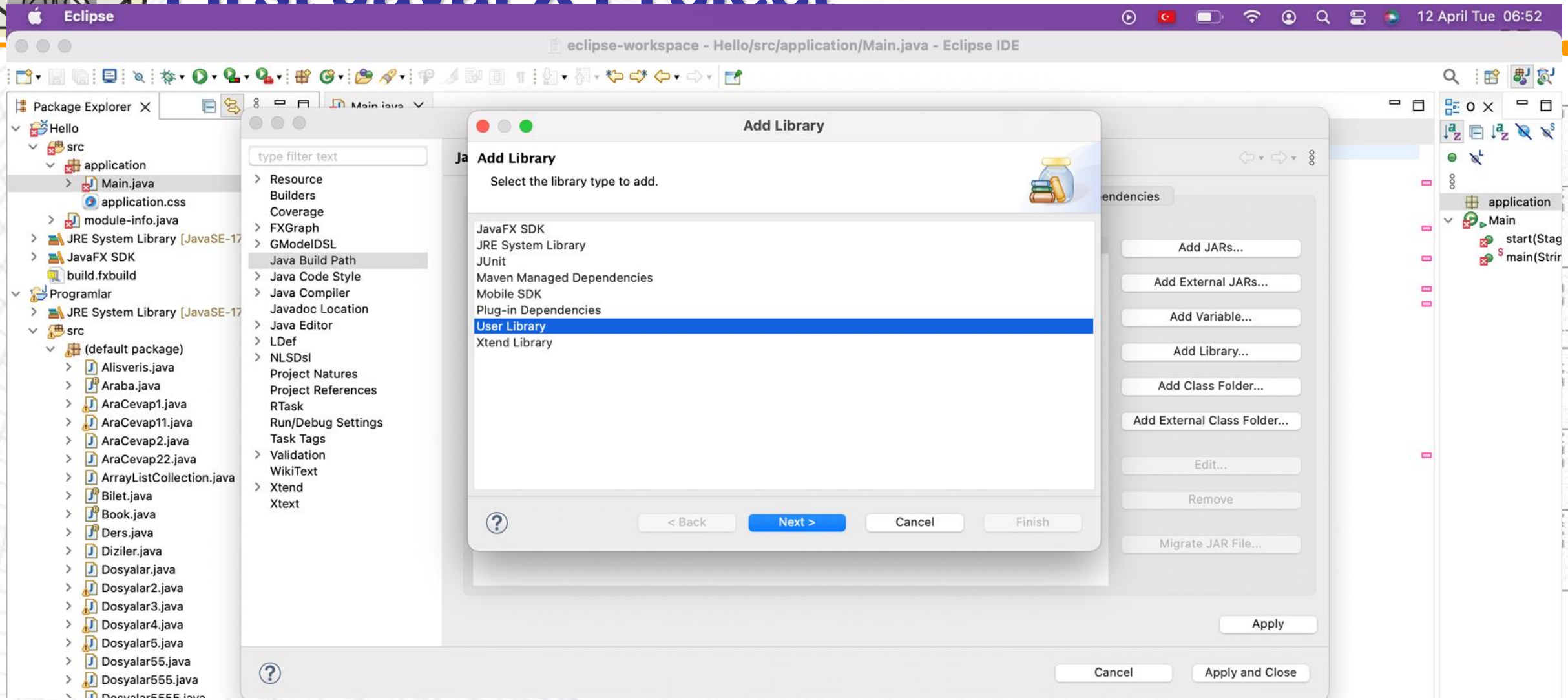
```
1   package appli
2
3   import javafx.app
7
8
9   public class Mair
10      @Override
11      public void s
12          try {
13              Borde
14              Scene scene = new Scene(root,400,400);
15              scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
16              primaryStage.setScene(scene);
17              primaryStage.show();
18          } catch(Exception e) {
19              e.printStackTrace();
20          }
21      }
22
23      public static void main(String[] args) {
24          launch(args);
25      }
26  }
27
```

application
Main
start(Stag
main(Strir

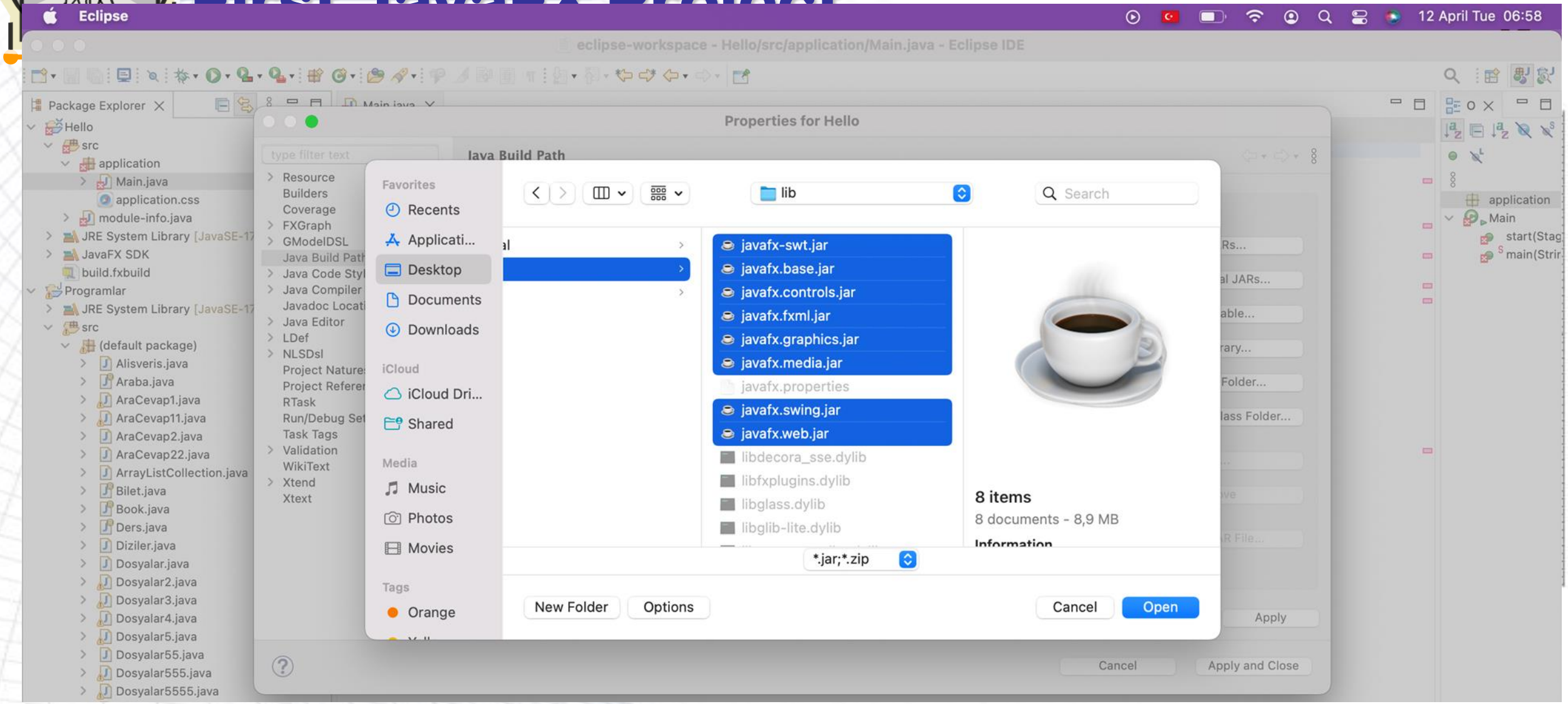
First JavaFX Project



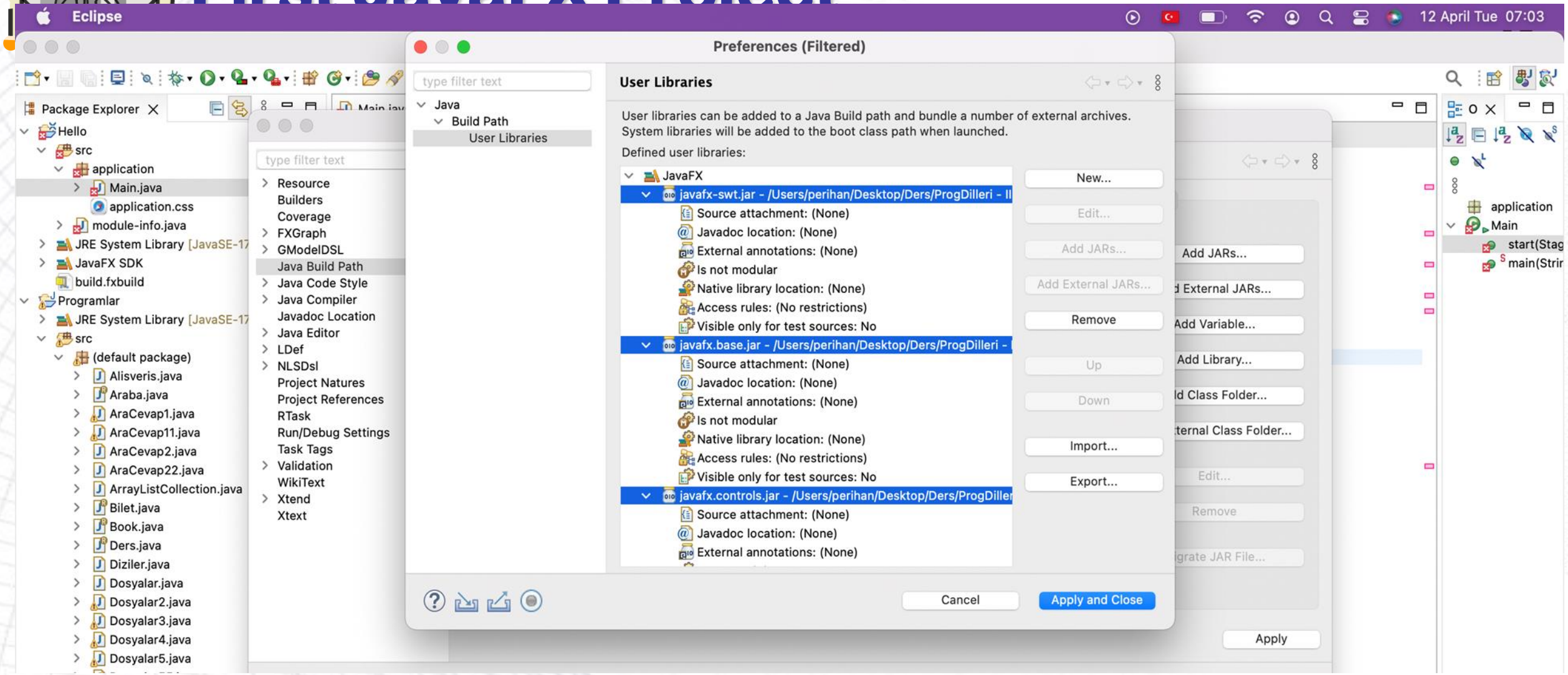
First JavaFX Project



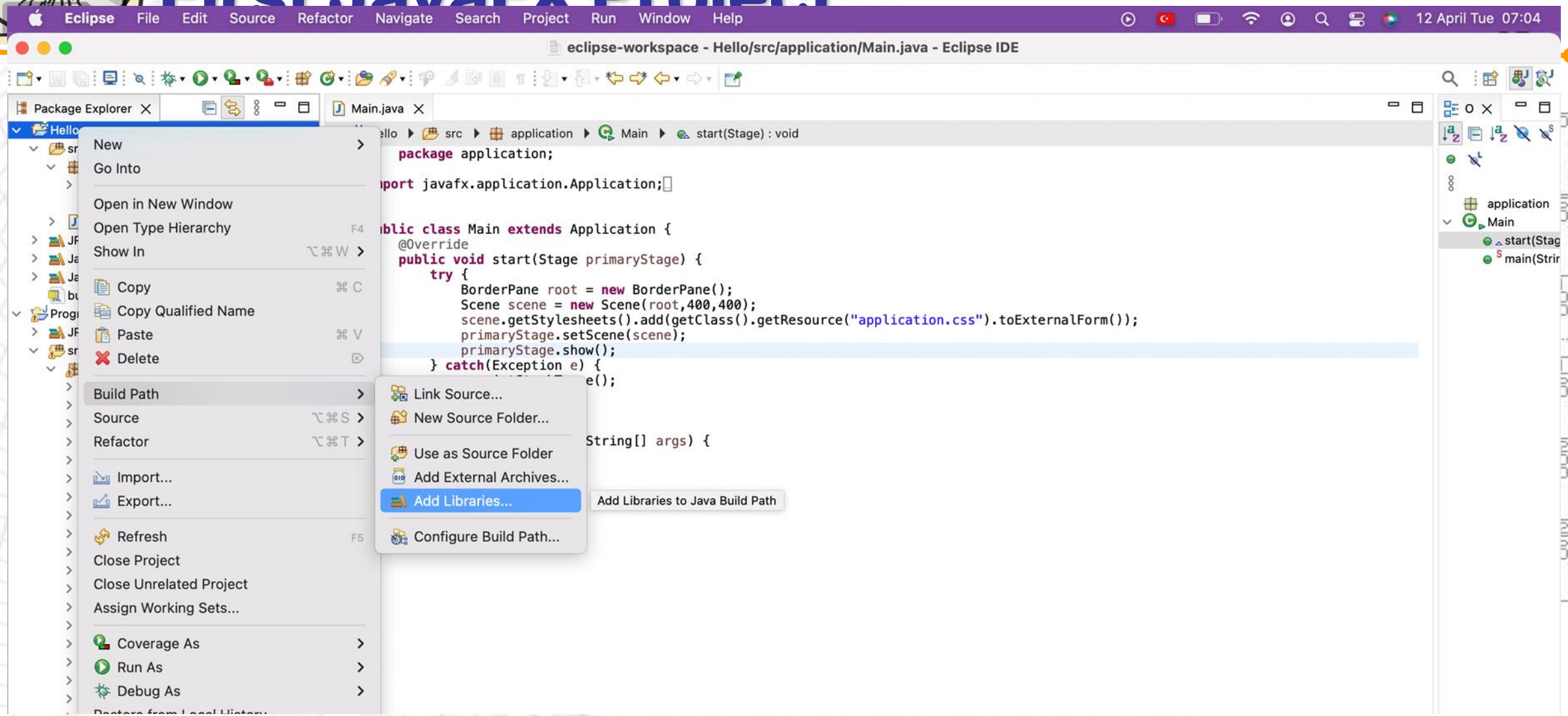
First JavaFX Project



First JavaFX Project



First JavaFX Project



First JavaFX Project

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help — 12 April Tue 07:09

eclipse-workspace - Hello/src/application/Main.java - Eclipse IDE

Package Explorer — Hello, src, JRE, Java, Java, build, Progra, JRE, src

Main.java — src ▸ application ▸ Main ▸ start(Stage) : void

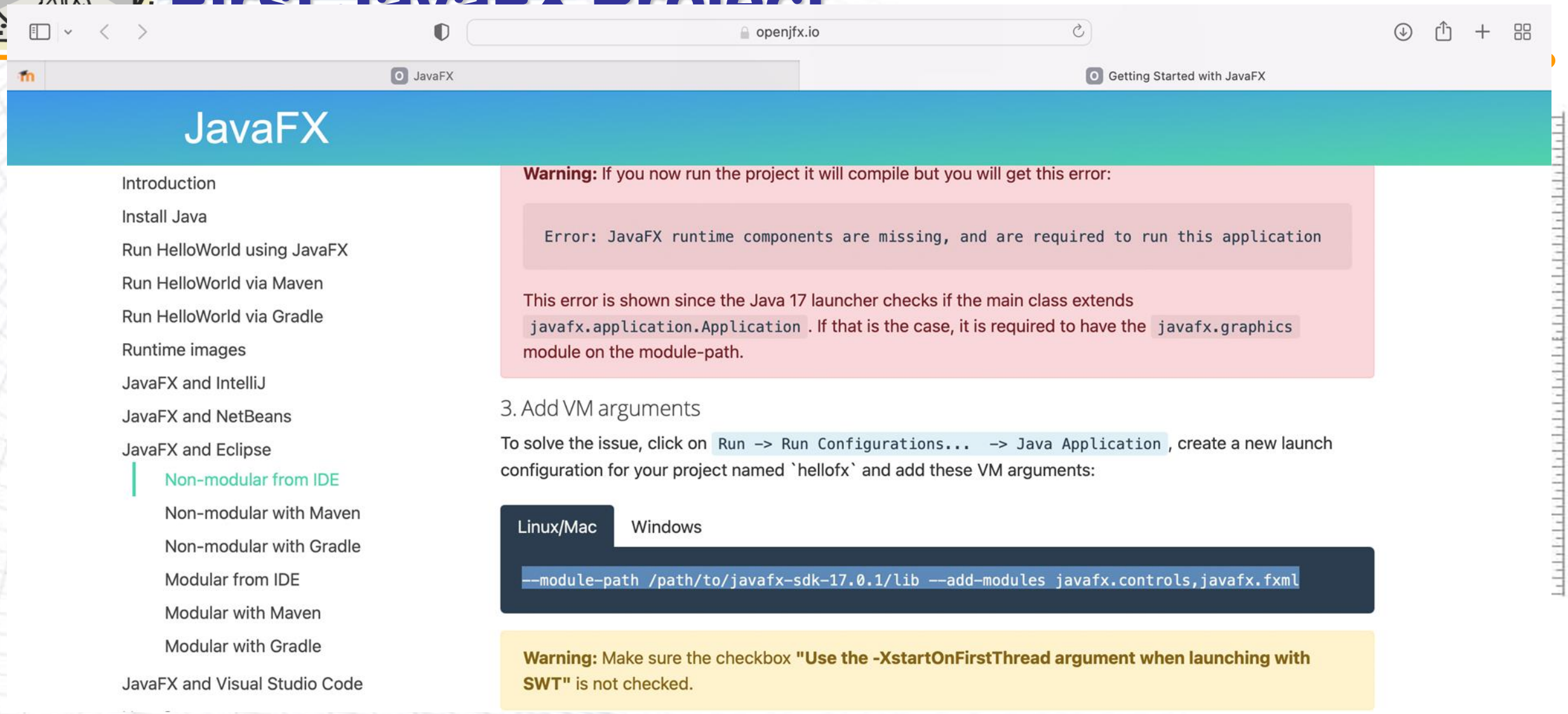
```
package application;

ort javafx.application.Application;


lic 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);
}
```

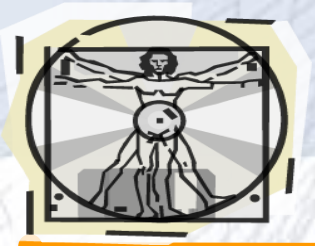
Outline — application, Main, start(Stag, main(Strir

Context menu:
- New
- Go Into
- Open in New Window
- Open Type Hierarchy F4
- Show In ⌥⌘W
- Copy ⌘C
- Copy Qualified Name
- Paste ⌘V
- Delete
- Build Path
- Source ⌥⌘S
- Refactor ⌥⌘T
- Import...
- Export...
- Refresh F5
- Close Project
- Close Unrelated Project
- Assign Working Sets...
- Coverage As
- Run As

Run As submenu: 1 Java Application ⌥⌘X J

First JavaFX Project

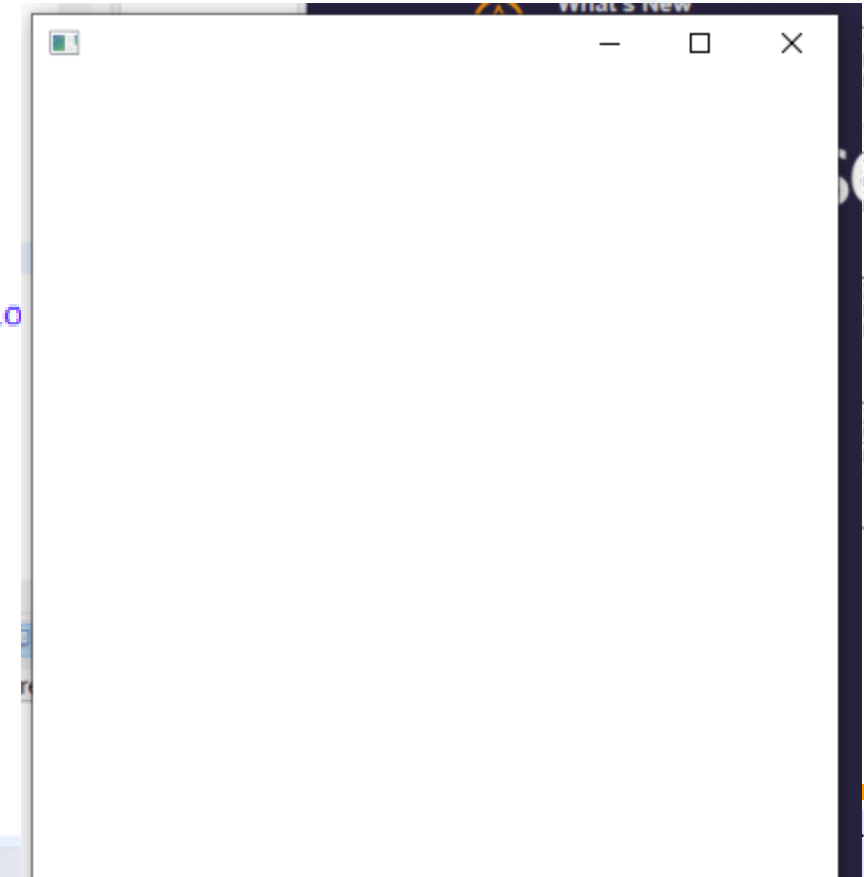


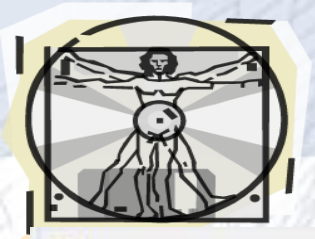


First Program

module-info.java | Lab005.java | Main.java

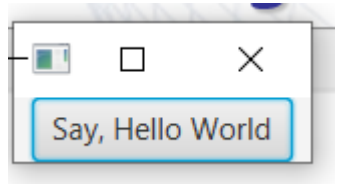
```
1  package application;
2
3  import javafx.application.Application;
7
8
9  public class Main extends Application {
10     @Override
11     public void start(Stage primaryStage) {
12         try {
13             BorderPane root = new BorderPane();
14             Scene scene = new Scene(root,400,400);
15             scene.getStylesheets().add(getClass().getResource("applicatio
16             primaryStage.setScene(scene);
17             primaryStage.show();
18         } catch(Exception e) {
19             e.printStackTrace();
20         }
21     }
22
23     public static void main(String[] args) {
24         launch(args);
25     }
26 }
```



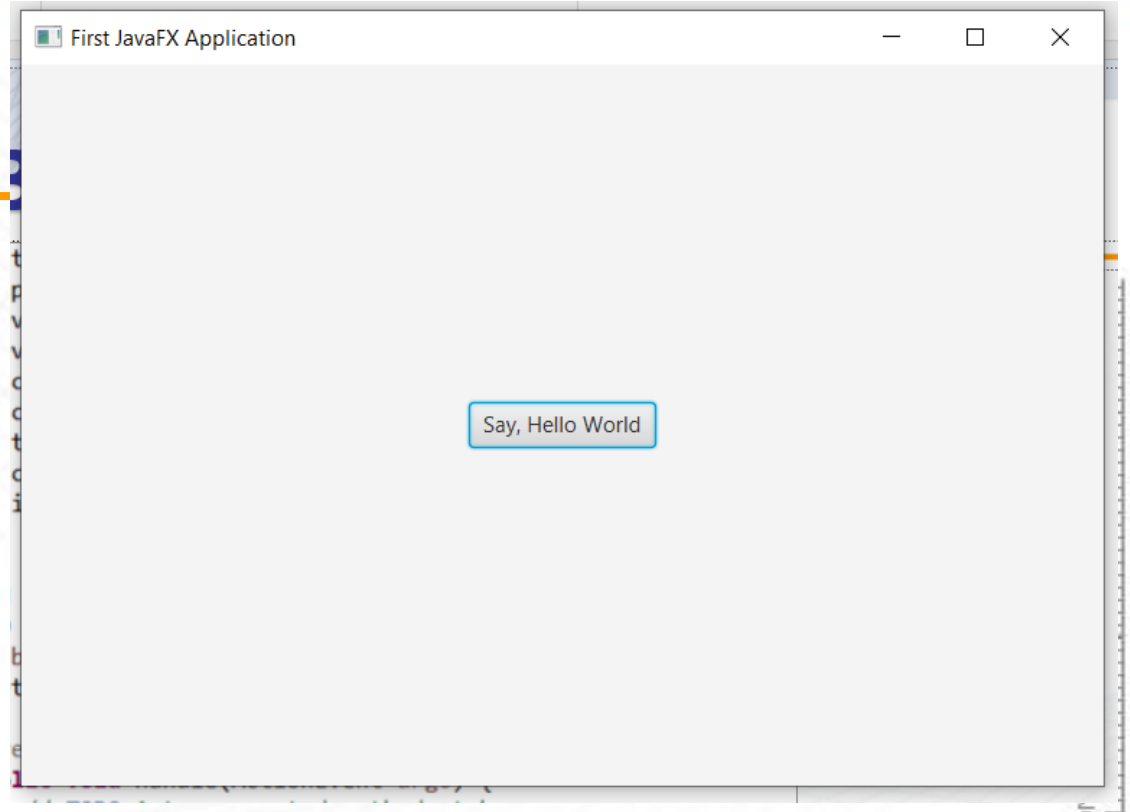


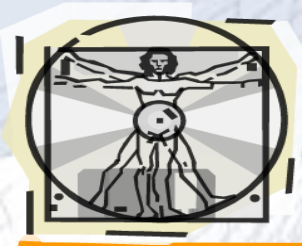
module-info.java | Lab005.java | *Main.java

```java
package application;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;
import javafx.scene.layout.StackPane;


public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            Button btn1=new Button("Say, Hello World");
            StackPane root=new StackPane();
            root.getChildren().add(btn1);
            Scene scene=new Scene(root);
            primaryStage.setScene(scene);
            primaryStage.setTitle("First JavaFX Application");
            primaryStage.show();
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

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



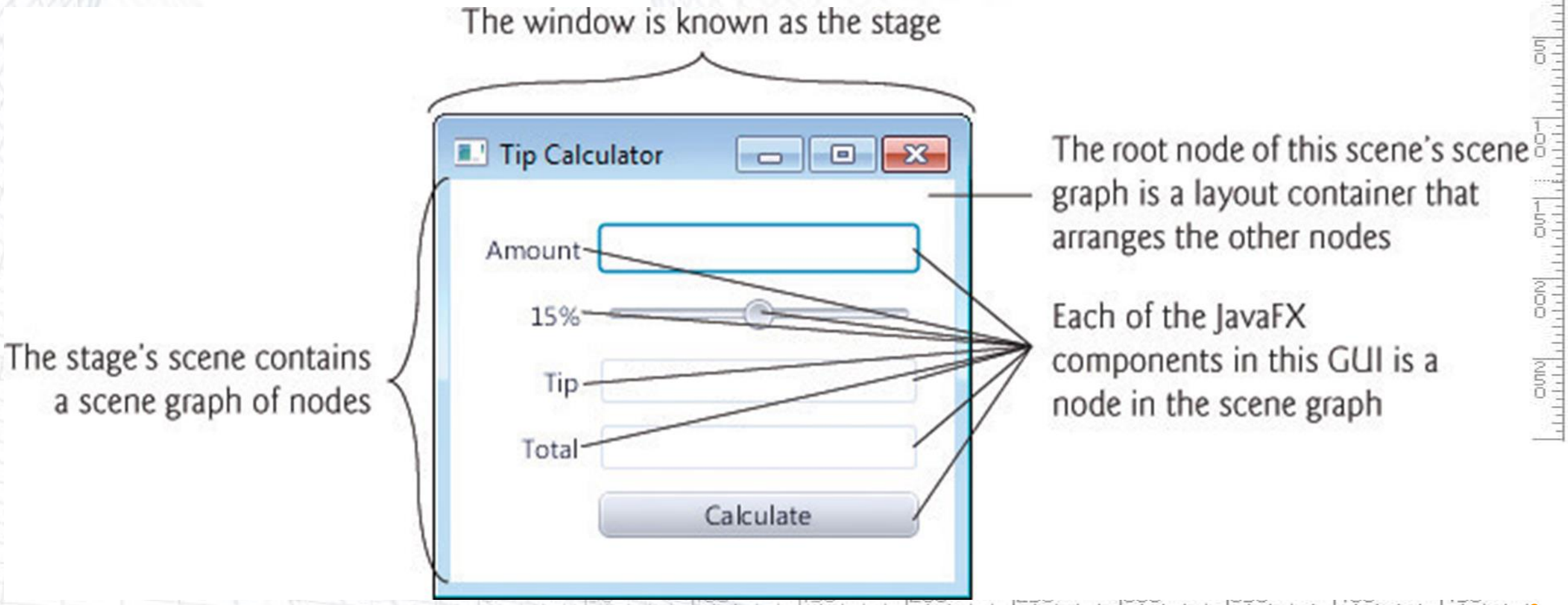
```java
package application;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;
import javafx.scene.layout.StackPane;
public class Main extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub
        Button btn1=new Button("Say, Hello World");
        btn1.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent arg0) {
                // TODO Auto-generated method stub
                System.out.println("hello world");
            }
        });
        StackPane root=new StackPane();
        root.getChildren().add(btn1);
        Scene scene=new Scene(root,600,400);
        primaryStage.setScene(scene);
        primaryStage.setTitle("First JavaFX Application");
        primaryStage.show();
    }

}
```

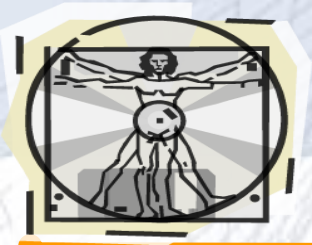




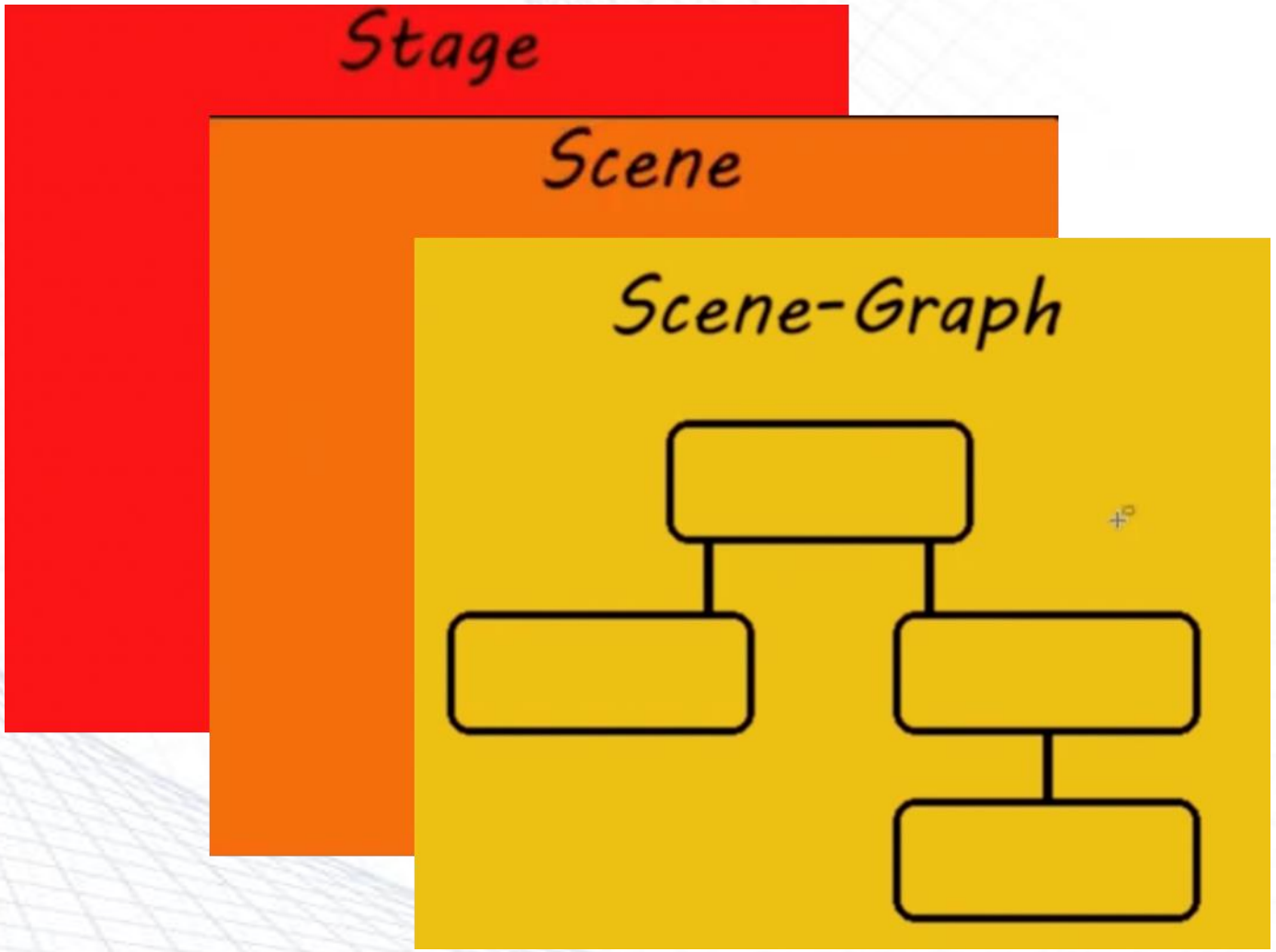
A JavaFX Code

- Consists of Stage, Scene and nodes

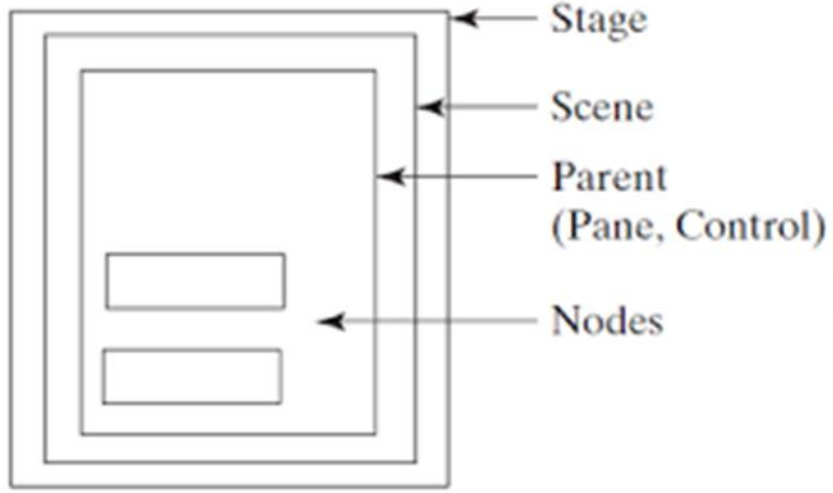


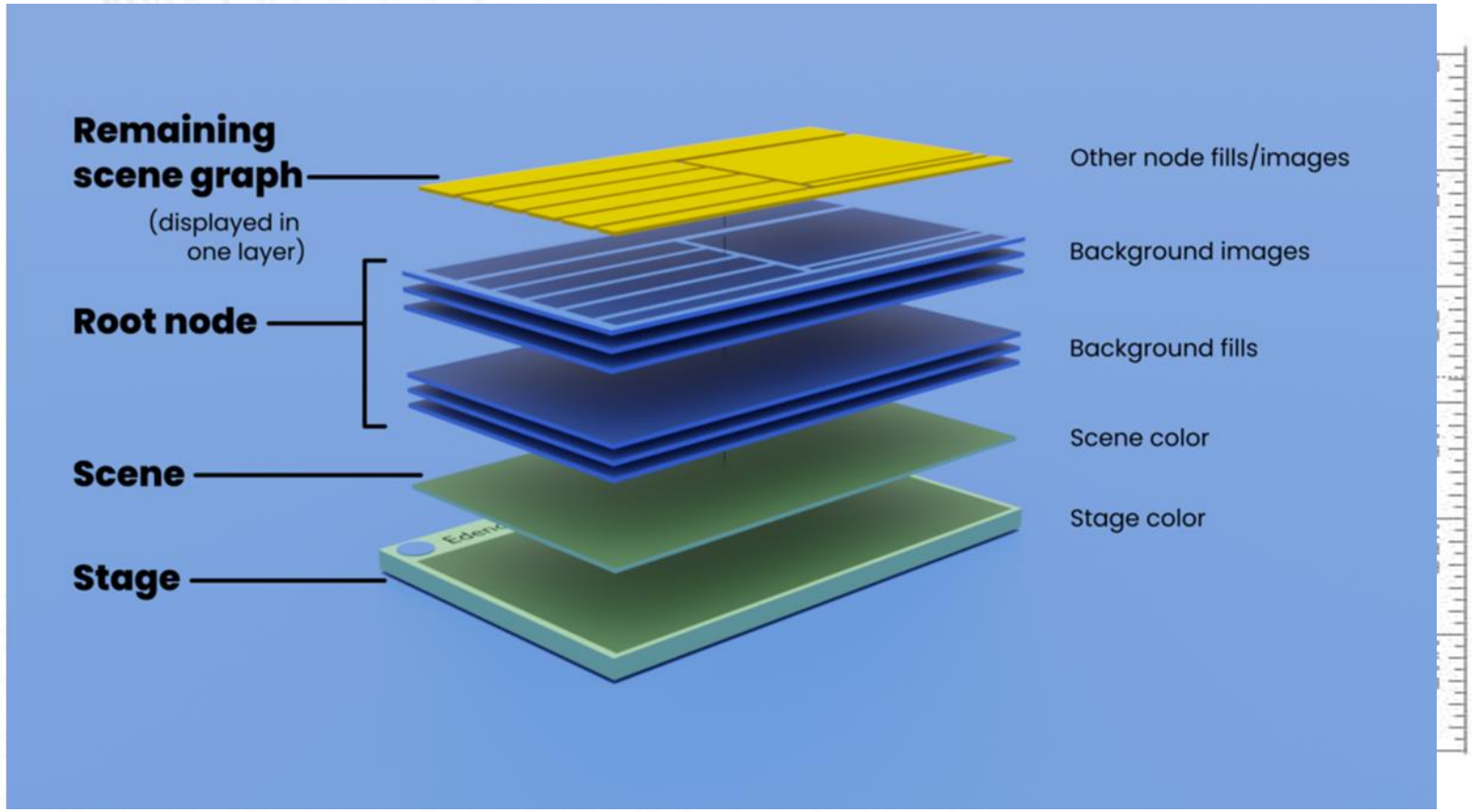


A JavaFX Code

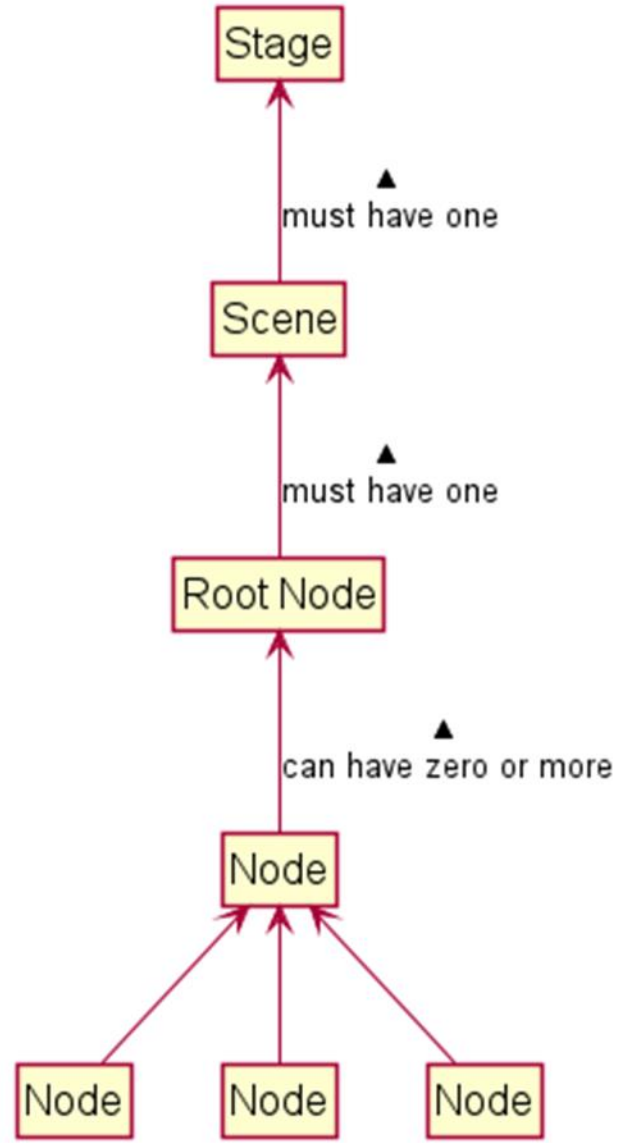


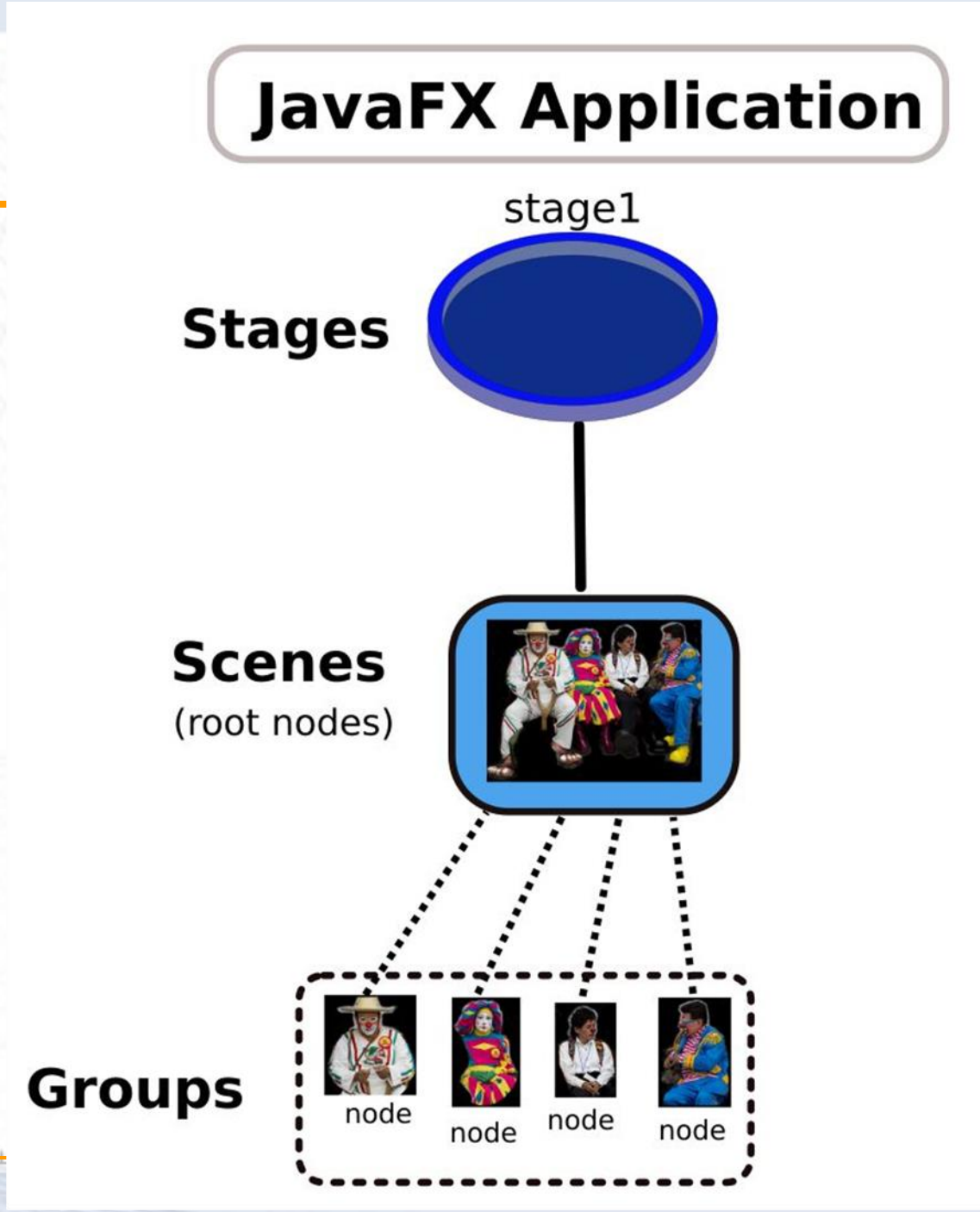
A JavaFX Code



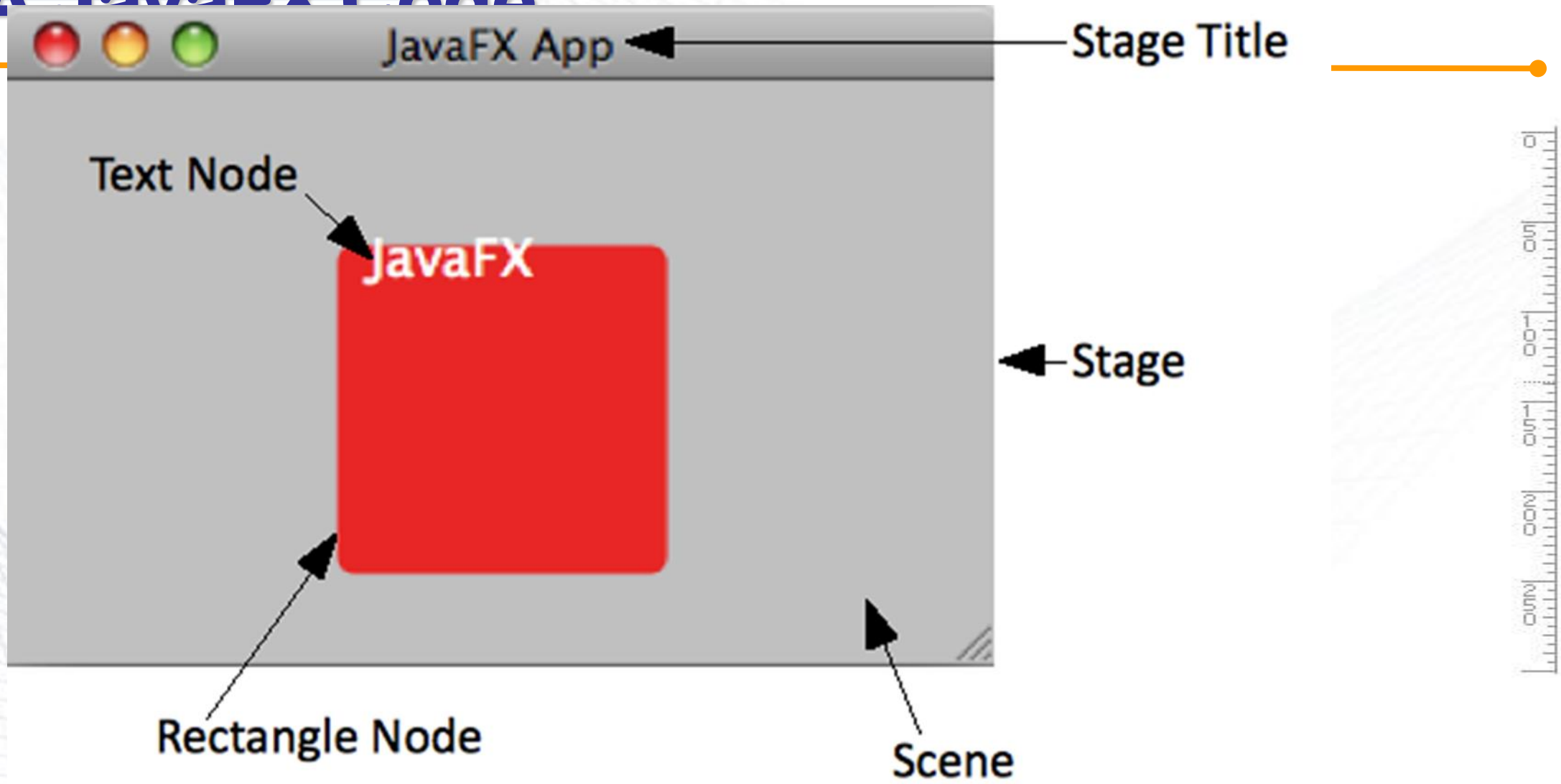


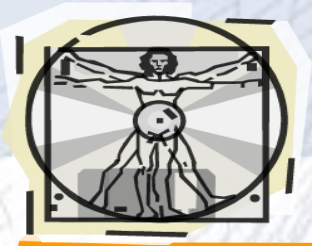
A JavaFX Code





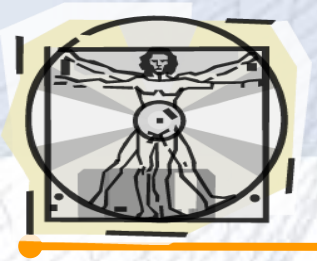
A JavaFX Code





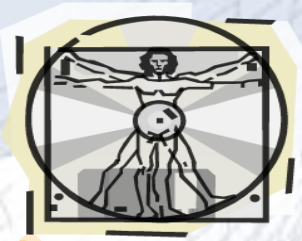
JavaFX App Window Structure (cont.)

- The **Stage** is the window in which a JavaFX app's GUI is displayed
  - It's an instance of class **`Stage`** (package `javafx.stage`).
- The **Stage** contains one active **Scene** that defines the GUI as a **scene graph**—a tree data structure of an app's visual elements, such as GUI controls, shapes, images, video, text and.
- The scene is an instance of class **`Scene`** (package `javafx.scene`).
- **Controls** are GUI components, such as
  - `Labels` that display text,
  - `TextFields` that enable a program to receive user input,
  - `Buttons` that users click to initiate actions, and more.



JavaFX Application Layout

- An application Window in JavaFX is known as a Stage.
  - package `javafx.stage`
- A Stage contains an active Scene which is set to a Layout container.
  - package `javafx.scene`
- The Scene may have other Layout containers for organizing Controllers in a Tree organization.
  - Nodes with children are layout containers.
  - Nodes without children are widgets.

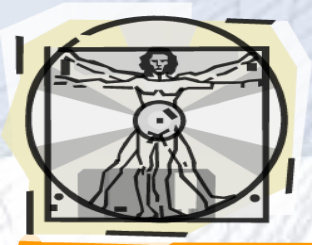


JavaFX App Window Structure (cont.)

- Each visual element in the scene graph is a **node**—an instance of a subclass of **`Node`** (package `javafx.scene`), which defines common attributes and behaviors for all nodes

- With the exception of the first node in the scene graph—the **root node**—each node in the scene graph has one parent.

- Nodes can have transforms (e.g., moving, rotating and scaling), opacity (whether a node is transparent, partially transparent or opaque), effects (e.g., drop shadows, blurs, reflection and lighting) and more.

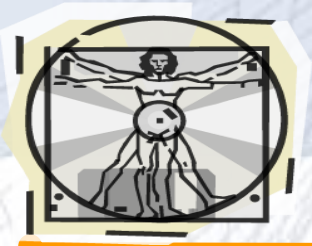


JavaFX controls

- Nodes with children are typically **layout containers** that arrange their child nodes in the scene.
  - Layout containers contain **controls** that accept inputs or other layout containers.

- When the user interacts with a control, the control generates an event.
- Programs can respond to these events—known as event handling—to specify what should happen when each user interaction occurs.

- An **event handler** is a method that responds to a user interaction. An FXML GUI's event handlers are defined in a so-called **controller class**.



Welcome App—Displaying Text and an Image

- In this section, *without writing any code* you'll build a GUI that displays text in a `Label` and an image in an `ImageView` (Fig. 12.2).
- You'll use only visual programming techniques to *drag-and-drop* JavaFX components onto Scene Builder's content panel—the design area.
- You'll use Scene Builder's **Inspector** to configure options, such as the `Label`'s text and font size, and the `ImageView`'s image.
- You'll view the completed GUI using Scene Builder's **Show Preview in Window** option.

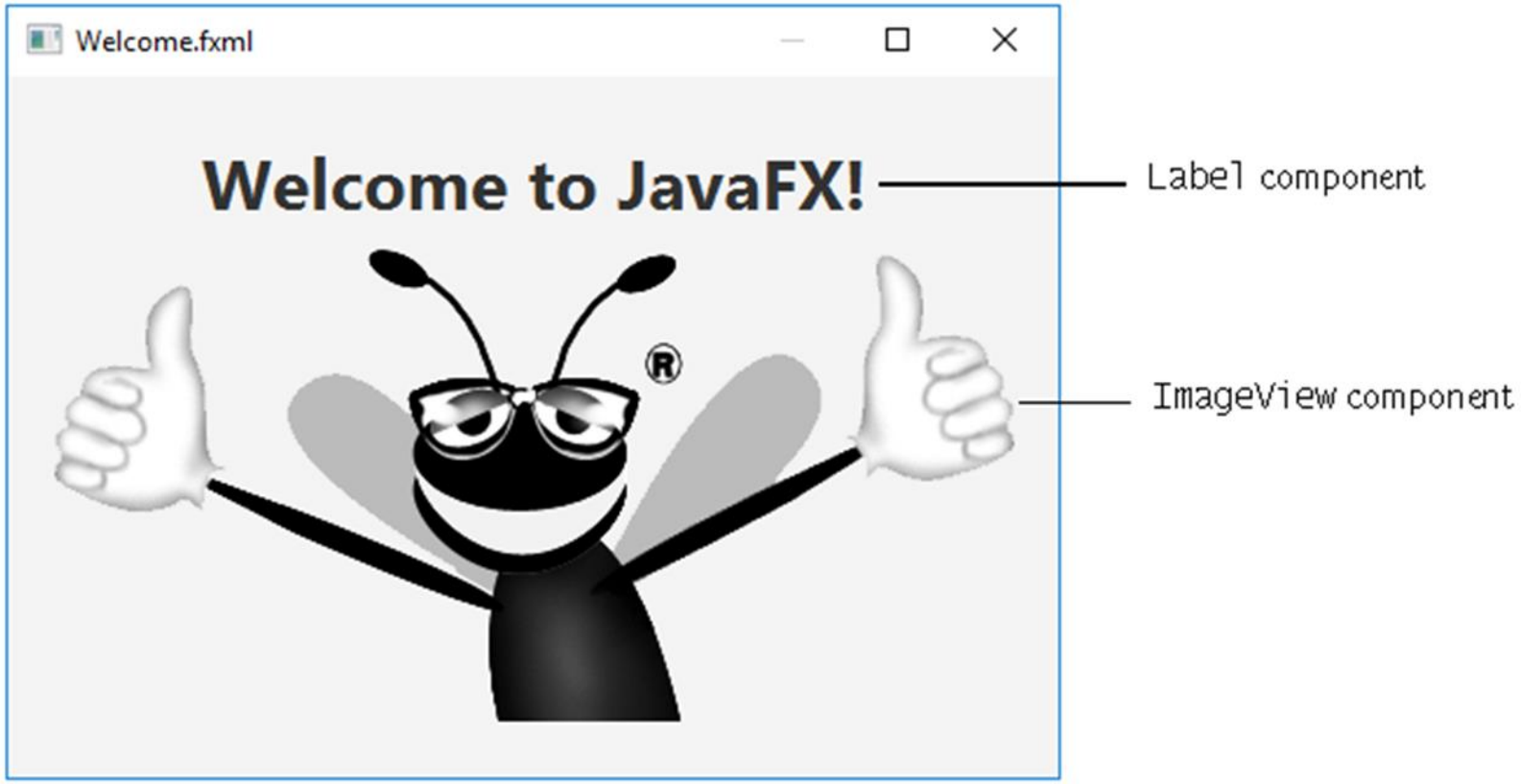
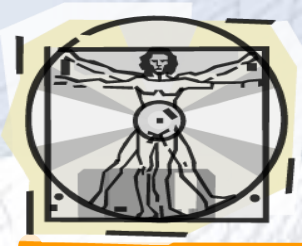
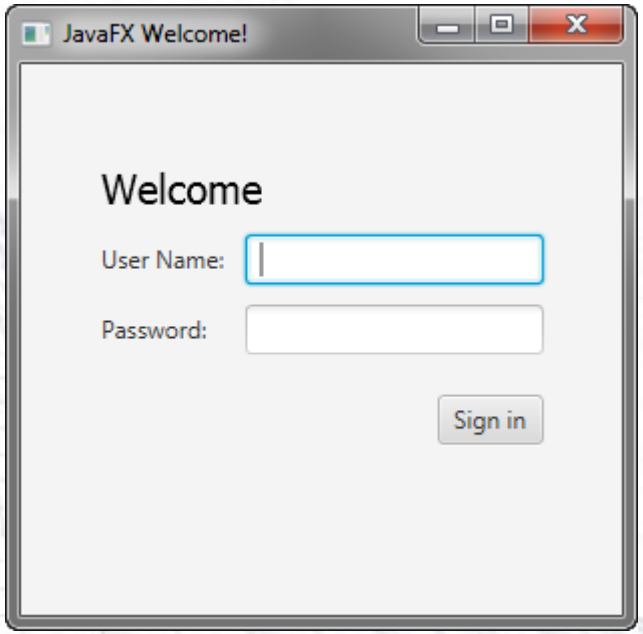


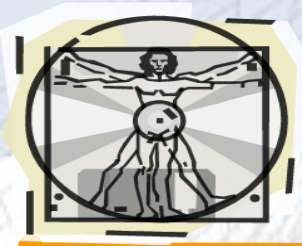
Fig. 12.2 | Final **Welcome** GUI in a preview window on Microsoft Windows 10.



Creating a Form in JavaFX

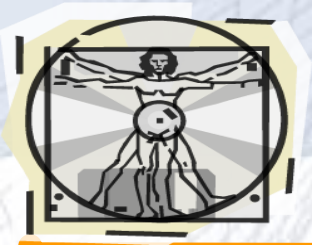
- Creating a form is a common activity when developing an application. This tutorial teaches you the basics of screen layout, how to add controls to a layout pane, and how to create input events.





Application Stage

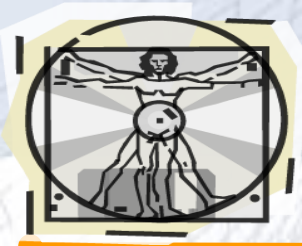
```
@Override
    public void start(Stage primaryStage) {
            primaryStage.setTitle("JavaFX Welcome");
            primaryStage.show();
    }
```



Create a GridPane Layout

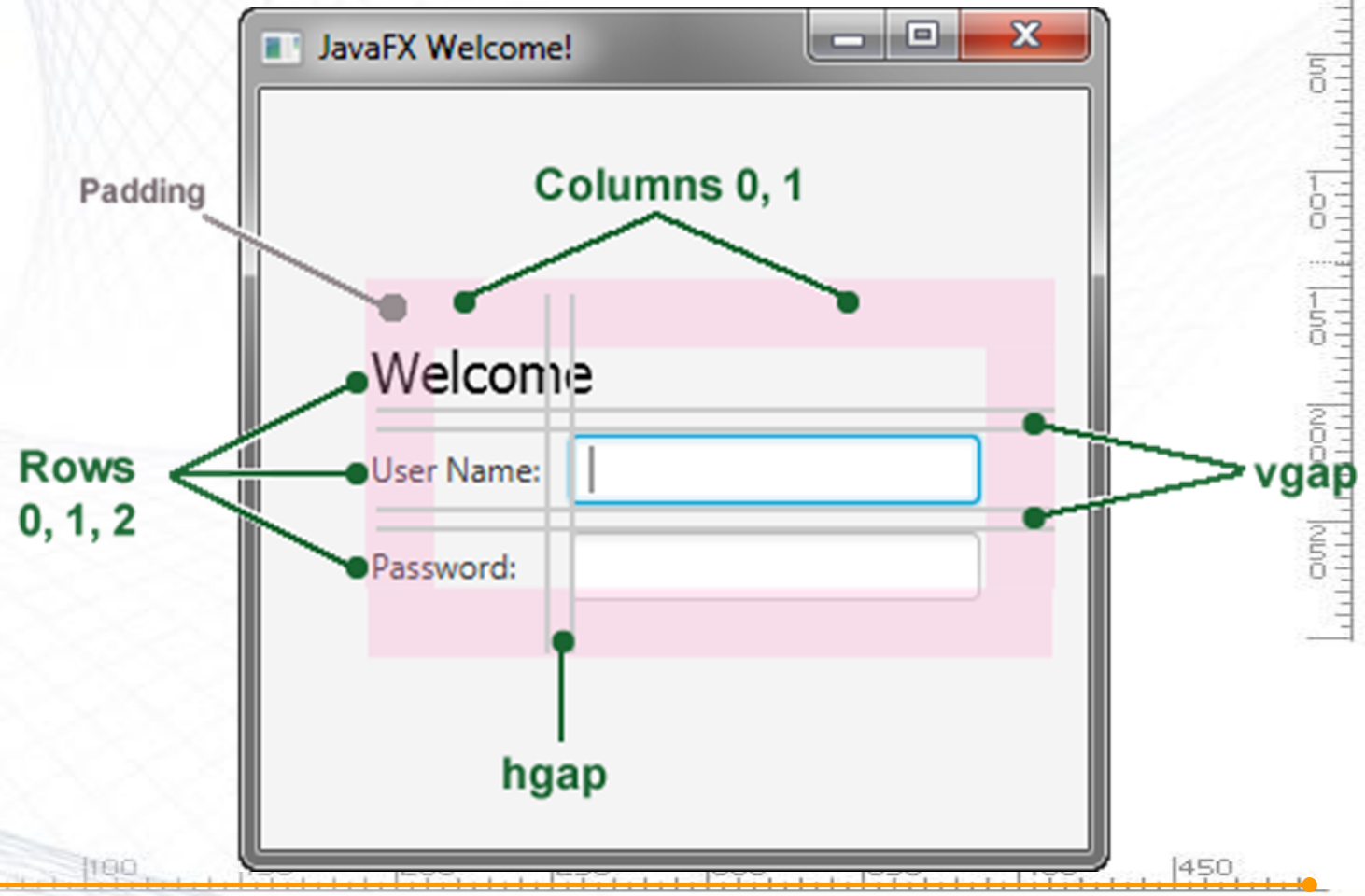
- For the login form, use a GridPane layout because it enables you to create a flexible grid of rows and columns in which to lay out controls. You can place controls in any cell in the grid, and you can make controls span cells as needed.

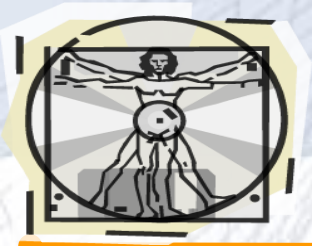
```
GridPane grid = new GridPane();
grid.setAlignment(Pos.CENTER);
grid.setHgap(10);
grid.setVgap(10);
grid.setPadding(new Insets(25, 25, 25, 25));
Scene scene = new Scene(grid, 300, 275);
primaryStage.setScene(scene);
```



Text Fields and Labels

```
Text scenetitle = new Text("Welcome");
scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
grid.add(scenetitle, 0, 0, 2, 1);

Label userName = new Label("User Name:");
grid.add(userName, 0, 1);

TextField userTextField = new TextField();
grid.add(userTextField, 1, 1);

Label pw = new Label("Password:");
grid.add(pw, 0, 2);

PasswordField pwBox = new PasswordField();
grid.add(pwBox, 1, 2);
```

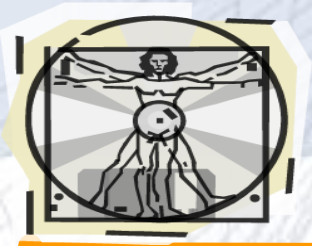




Buttons

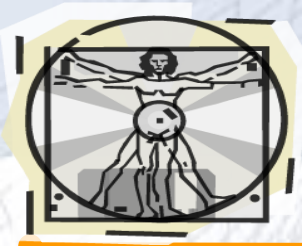
The first line creates a button named btn with the label Sign in, and the second line creates an HBox layout pane named hbBtn with spacing of 10 pixels. The HBox pane sets an alignment for the button that is different from the alignment applied to the other controls in the grid pane. The alignment property has a value of Pos.BOTTOM_RIGHT, which positions a node at the bottom of the space vertically and at the right edge of the space horizontally. The button is added as a child of the HBox pane, and the HBox pane is added to the grid in column 1, row 4.

```
Button btn = new Button("Sign in");
HBox hbBtn = new HBox(10);
hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
hbBtn.getChildren().add(btn);
grid.add(hbBtn, 1, 4);
```



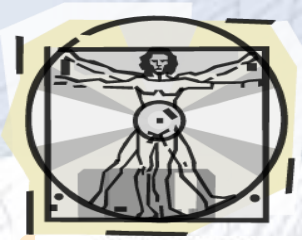
Actions

```
final Text actiontarget = new Text();
        grid.add(actiontarget, 1, 6);


btn.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent e) {
        actiontarget.setFill(Color.FIREBRICK);
        actiontarget.setText("Sign in button pressed");
    }
});
```



Header Libraries

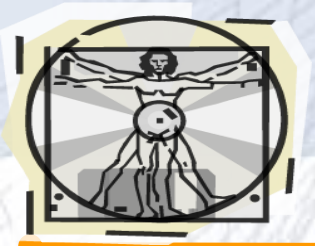
```
package application;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import static javafx.geometry.HPos.RIGHT;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class Main extends Application {
```



Main Code

```
@Override
public void start(Stage primaryStage) {
    primaryStage.setTitle("JavaFX Welcome");
    GridPane grid = new GridPane();
    grid.setAlignment(Pos.CENTER);
    grid.setHgap(10);
    grid.setVgap(10);
    grid.setPadding(new Insets(25, 25, 25, 25));

    Text scenetitle = new Text("Welcome");
    scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
    grid.add(scenetitle, 0, 0, 2, 1);

    Label userName = new Label("User Name:");
    grid.add(userName, 0, 1);

    TextField userTextField = new TextField();
    grid.add(userTextField, 1, 1);

    Label pw = new Label("Password:");
    grid.add(pw, 0, 2);

    PasswordField pwBox = new PasswordField();
    grid.add(pwBox, 1, 2);

    Button btn = new Button("Sign in");
    HBox hbBtn = new HBox(10);
    hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
    hbBtn.getChildren().add(btn);
    grid.add(hbBtn, 1, 4);

    final Text actiontarget = new Text();
    grid.add(actiontarget, 0, 6);
    grid.setColumnSpan(actiontarget, 2);
    grid.setHalignment(actiontarget, RIGHT);
    actiontarget.setId("actiontarget");

    btn.setOnAction(new EventHandler<ActionEvent>() {

        @Override
        public void handle(ActionEvent e) {
            actiontarget.setFill(Color.FIREBRICK);
            actiontarget.setText("Sign in button pressed");
        }
    });

    Scene scene = new Scene(grid, 300, 275);
    primaryStage.setScene(scene);
    primaryStage.show();
}

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

}
```

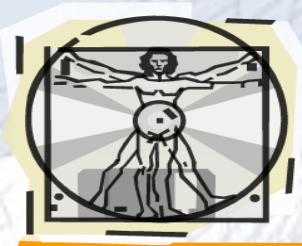
50 200 250 300 350 400 450



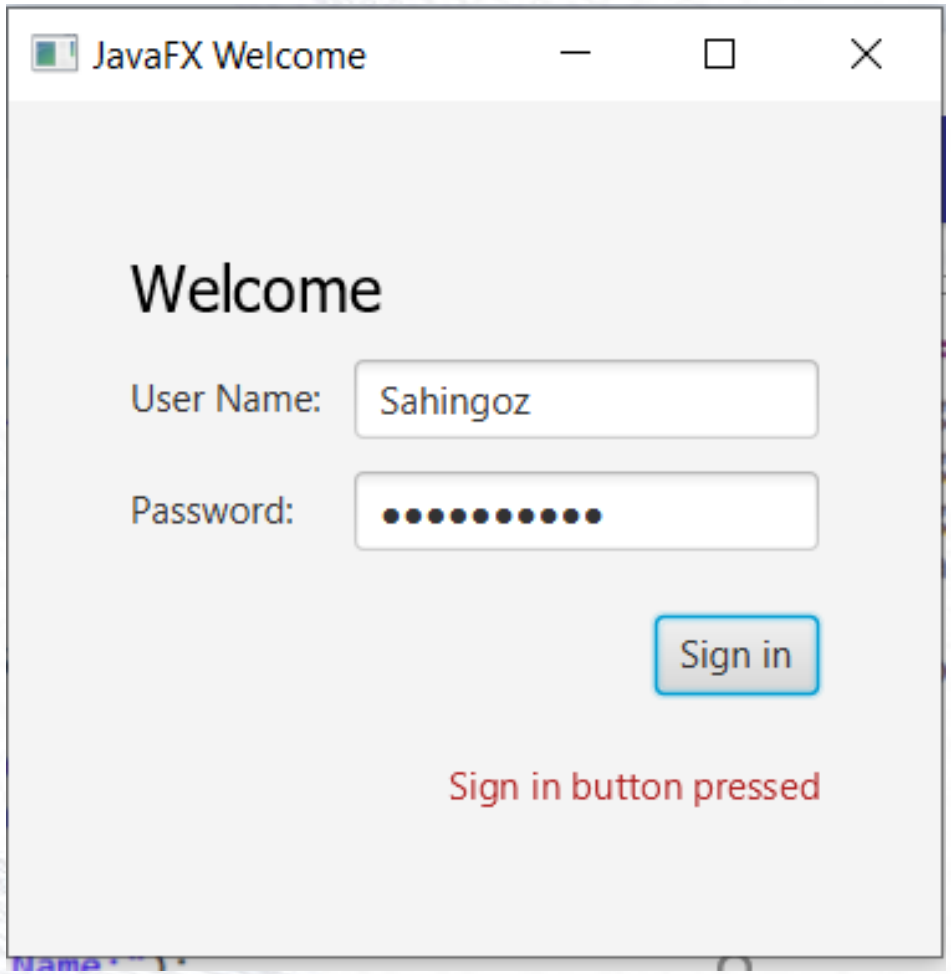
Full Code for copy

```
package application;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import static javafx.geometry.HPos.RIGHT;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("JavaFX Welcome");
        GridPane grid = new GridPane();
        grid.setAlignment(Pos.CENTER);
        grid.setHgap(10);
        grid.setVgap(10);
        grid.setPadding(new Insets(25, 25, 25, 25));

        Text scenetitle = new Text("Welcome");
        scenetitle.setFont(Font.font("Tahoma",
FontWeight.NORMAL, 20));
        grid.add(scenetitle, 0, 0, 2, 1);

        Label userName = new Label("User Name:");
        grid.add(userName, 0, 1);

        TextField userTextField = new TextField();
        grid.add(userTextField, 1, 1);

        Label pw = new Label("Password:");
        grid.add(pw, 0, 2);

        PasswordField pwBox = new PasswordField();
        grid.add(pwBox, 1, 2);

        Button btn = new Button("Sign in");
        HBox hbBtn = new HBox(10);
        hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
        hbBtn.getChildren().add(btn);
        grid.add(hbBtn, 1, 4);

        final Text actiontarget = new Text();
        grid.add(actiontarget, 0, 6);
        grid.setColumnSpan(actiontarget, 2);
        grid.setHalignment(actiontarget, RIGHT);
        actiontarget.setId("actiontarget");

        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent e) {
                actiontarget.setFill(Color.FIREBRICK);
                actiontarget.setText("Sign in button
pressed");
            }
        });

        Scene scene = new Scene(grid, 300, 275);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

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

}
```

https://docs.oracle.com/javase/8/javafx/sample-apps/Login.zip



Output


JavaFX Welcome
Welcome
User Name:
Sahingoz
Password:
Sign in
Sign in button pressed