به نام خدا

# دستور کار کارگاه برنامه‌نویسی پیشرفته

## جلسه نهم

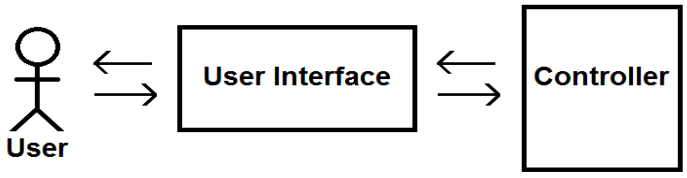
## آشنایی با طراحی رابط کاربری گرافیکی در جاوا (JavaFX)

### مقدمه

در این جلسه با مولفه‌های رابط کاربری گرافیکی در جاوا آشنا می‌شویم. روش‌های مختلفی برای ایجاد واسط‌کاربری[1] وجود دارد. در روش‌های سنتی از کدنویسی برای تولید واسط‌کاربری استفاده می‌شد. این روش دشوار بود و زمان زیادی را به خود اختصاص می‌داد. در روش‌های جدید سعی می‌شود با استفاده از ابزارها برای تولید واسط‌کاربری استفاده شود. هدف از این ابزارها افزایش سادگی نرم‌افزار و تسریع تولید واسط‌کاربری می‌باشد. در این نوشتار از ابزار JavaFX برای ایجاد واسط‌کاربری استفاده خواهیم کرد.

### آشنایی با JavaFX

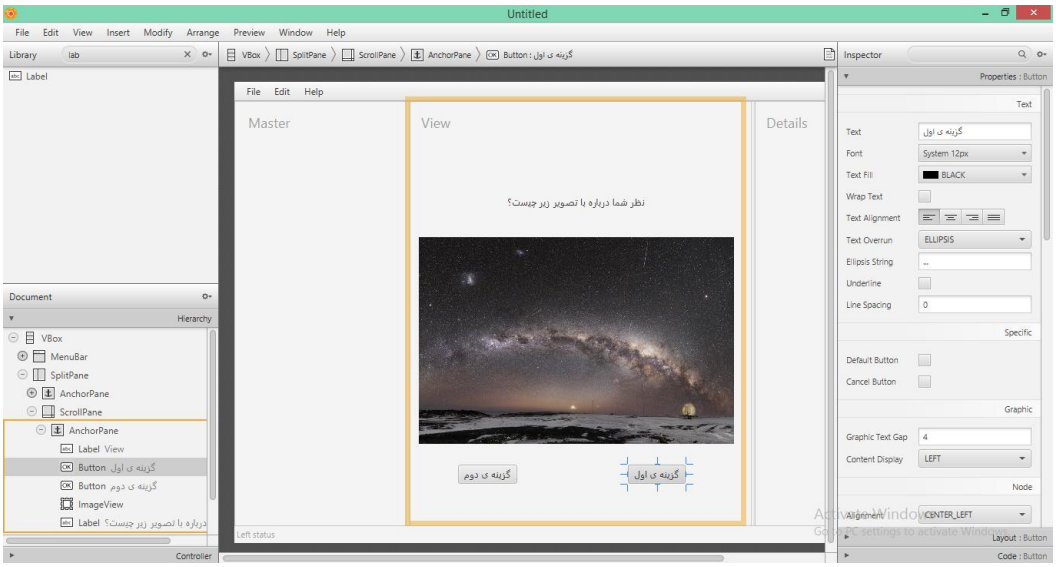
در این بخش به معرفی و تشریح مفاهیم لازم برای کار با ابزار JavaFX می‌پردازیم. به طورکلی، یک برنامه شامل دو بخش می‌باشد. بخش اول که واسط‌کاربری نرم‌افزار را شامل می‌شود، شامل ابزارهایی است که کاربر با استفاده از آن می‌تواند با نرم‌افزار تعامل داشته باشد. به عنوان مثال، Button، TextField و... ازجمله این ابزارها می‌باشند. بخش دوم شامل پردازش‌هایی است که معمولا در پس‌زمینه‌ی نرم‌افزار انجام می‌شوند و اهداف نرم‌افزار را برآورده می‌سازند. به عنوان نمونه، ثبت مشخصات کاربر در حافظه می‌تواند در بخش دوم نرم‌افزار انجام شود. تفاوت بدیهی میان بخش اول و دوم در آن است که بخش اول توسط کاربر دیده می‌شود؛ درحالی‌که بخش دوم بدون آن که کاربر آن را مشاهده کند، به اجرای یک‌سری محاسبات می‌پردازد. در شکل ۱ این دو بخش ترسیم شده است.




شکل ۱، بخش‌های مختلف نرم‌افزار در تعامل با کاربر

[1] User Interface

در برنامه‌هایی که با استفاده از JavaFX تولید می‌شوند، بخش اول که شامل واسط‌کاربری می‌شود توسط ابزار SceneBuilder ایجاد می‌شود. شما می‌توانید با استفاده از ابزار SceneBuilder تنها با استفاده از ماوس و بدون انجام کدنویسی، یک واسط‌کاربری مناسبی را ایجاد کنید. در شکل ۲ بخشی از این ابزار را مشاهده می‌کنید.



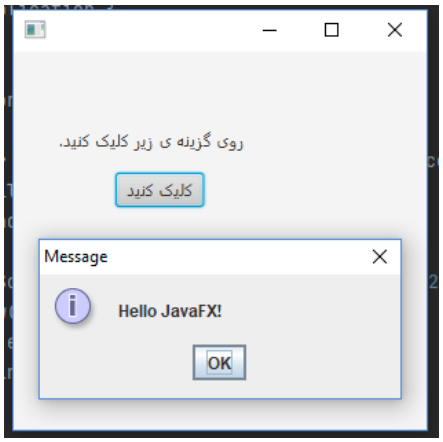

شکل ۲، محیط SceneBuilder برای ایجاد واسط‌کاربری

با استفاده از ابزار SceneBuilder قادر خواهید بود به راحتی و با عمل DragAndDrop محیطی زیبا و مناسب را برای تعامل با کاربر طراحی و پیاده‌سازی کنید. در نهایت، ابزار SceneBuilder کدی در قالب XML را تولید می‌کند. این کد که با پسوند FXML ایجاد می‌شود، برای جاوا قابل درک است.

بخش دوم که به پردازش‌های پس‌زمینه مربوط می‌شود، توسط کلاسی که معمولا Controller نامیده می‌شود، انجام می‌شود. ابزارهای واسط‌کاربری قادر هستند توابع این کلاس را فراخوانی کرده و پردازش‌های لازم را اجرا کنند.

## مراحل انجام کار

در این بخش، ما می‌خواهیم یک پروژه‌ی ساده را با استفاده از JavaFX ایجاد کنیم. در این پروژه می‌خواهیم یک برنامه بسازیم که یک کلید[2] داشته باشد. کاربر باید با کلیک روی این کلید، پیامی را در پنجره‌ای جدید مشاهده کند. این پروژه در شکل ۳ نمایش داده می‌شود. هدف از ایجاد این پروژه آن است که نحوه‌ی کار با این ابزار تشریح شود.

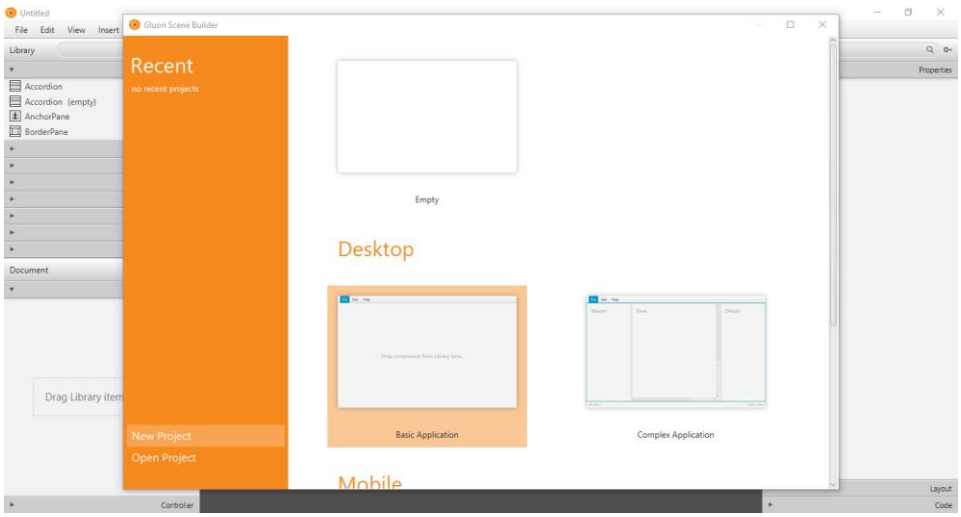



شکل ۳، یک برنامه‌ی ساده که با JavaFX تولید شده است.

در این بخش، ما با استفاده از ابزار Intellij نسخه‌ی Community اقدام به توسعه‌ی برنامه خواهیم کرد. درصورتی لزوم می‌توانید این ابزار را در سایت JetBrains دانلود کنید. لازم به ذکر است استفاده از JavaFX در سایر نرم‌افزارها مثل NetBeans و... از فرایند مشابهی تبعیت می‌کند.

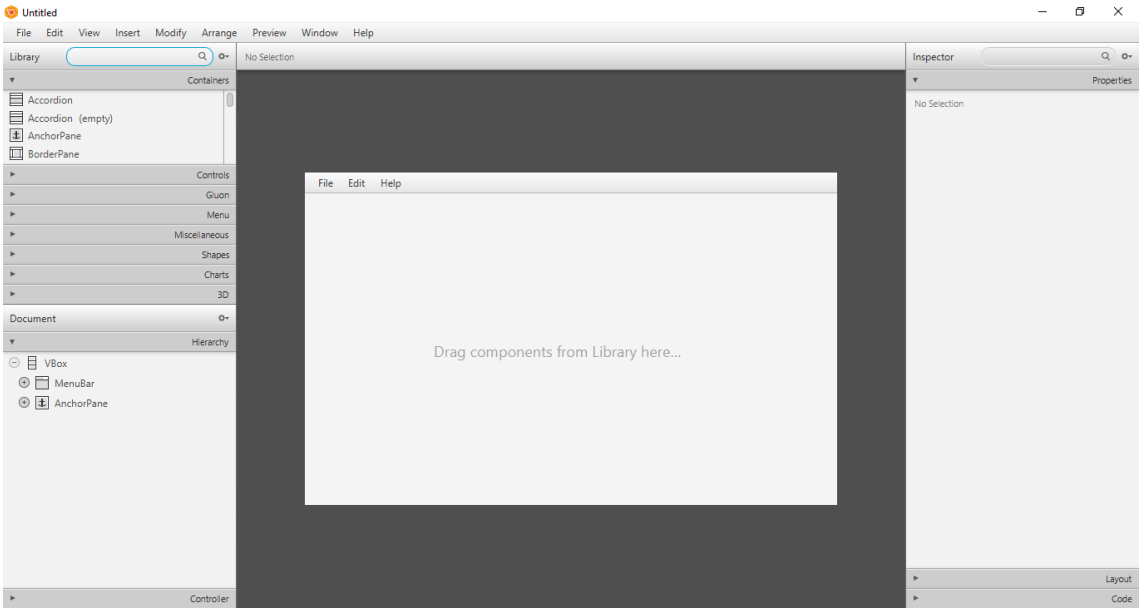
- مرحله‌ی اول: ابزار SceneBuilder را دانلود و نصب کنید. برای دریافت این نرم‌افزار می‌توانید به وب‌سایت Gluon مراجعه کنید.
- مرحله‌ی دوم: ابزار SceneBuilder را اجرا کنید. در شکل ۴ نمایی از اجرای این نرم‌افزار مشاهده نمایش داده می‌شود. در ابتدا نوع پروژه‌ای که قصد توسعه‌ی آن را دارید از شما پرسیده می‌شود. در این بخش، گزینه‌ی Basic Application را انتخاب کنید.

[2] Button

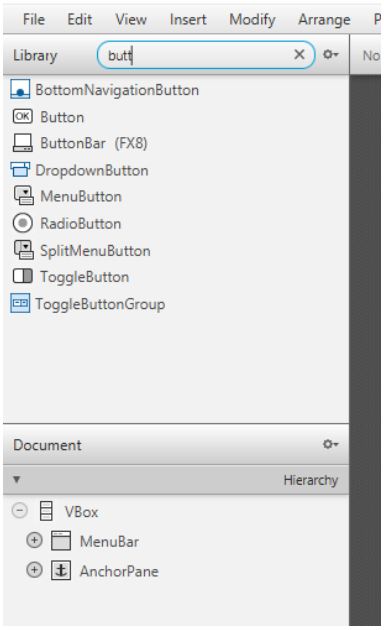



شکل ۴، نمایی از ابزار SceneBuilder

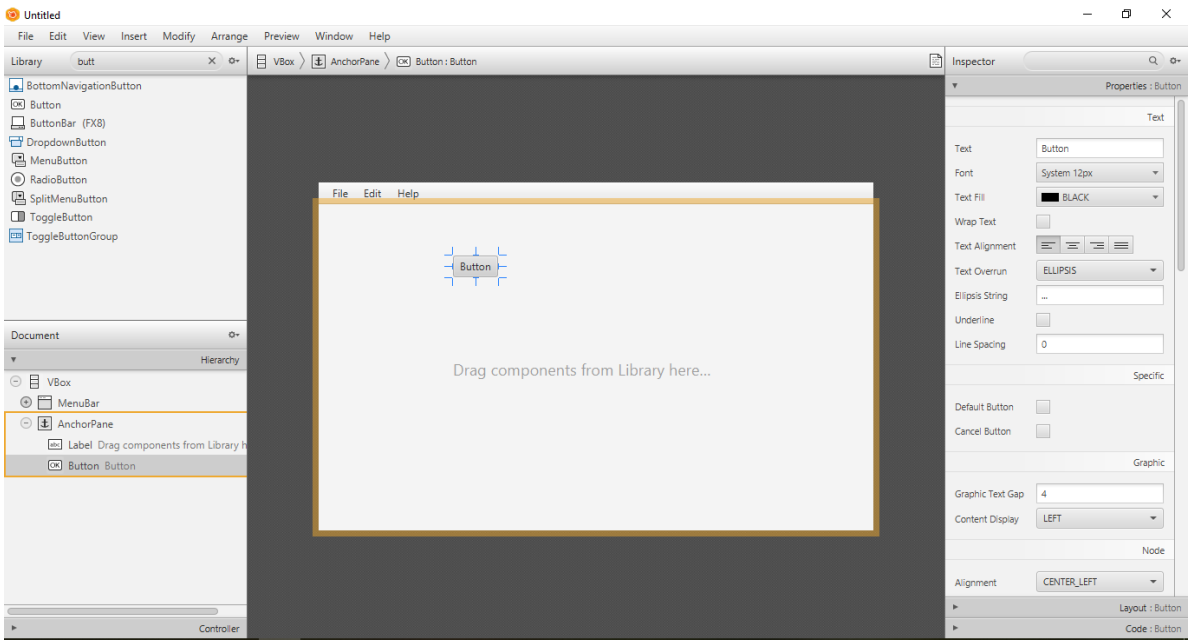
- مرحله‌ی سوم: بعد از انتخاب گزینه‌ی Basic Application وارد ابزار می‌شوید و در آن‌جا قادر به تنظیم و ویرایش واسط‌کاربری خواهید بود. همانطور که در شکل ۵ مشاهده می‌شود، در سمت چپ منویی وجود دارد که شامل ابزارهای گرافیکی مثل Buttonو... می‌شود. شما می‌توانید با انتخاب این ابزارها و کشیدن آن به واسط‌کاربری، آن ابزار را در واسط‌کاربری تعبیه کنید. افزودن ابزار Button در شکل های ۶ و ۷ نمایش داده شده است.




شکل ۵، محیط ویرایش واسط‌کاربری در ابزار SceneBuilder



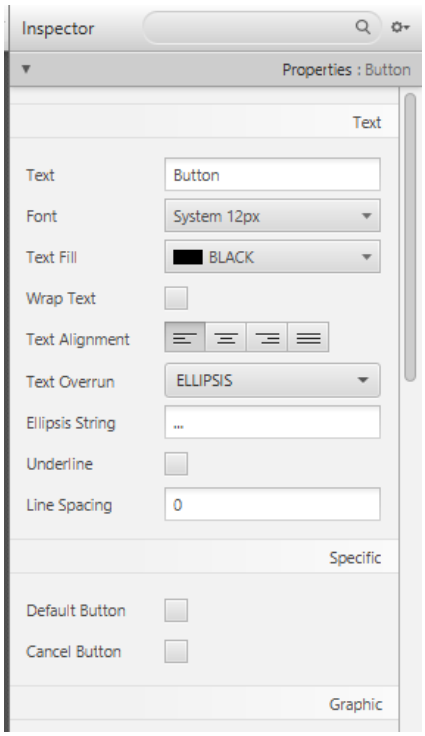

شکل ۶، جستجوی ابزار Button در محیط SceneBuilder




شکل ۷، ایجاد ابزار Button در واسط‌کاربری

همانطور که ملاحظه می‌شود، با انتخاب هر ابزار در واسط‌کاربری، مشخصات آن ابزار در منوی سمت راست نمایان می‌شود. شما می‌توانید به سادگی این صفات را تغییر داده و ویرایش کنید.

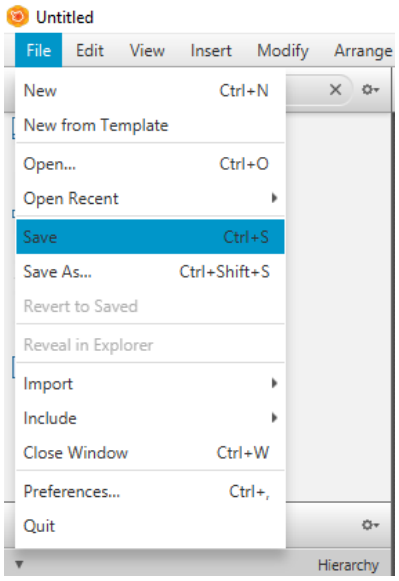
- مرحله‌ی چهارم: پس از انتخاب ابزار Button، با استفاده از منوی سمت راست (Inspector) متن درون Button را تغییر دهید. برای این کار کافیست محتوای Text را ویرایش کنید. در این نوشتار محتوای درون این ابزار را !Click here مقداردهی می‌کنیم. نحوه‌ی انجام این کار در شکل ۸ نمایش داده می‌شود.




شکل ۸، ویرایش مشخصات ابزار Button

عملیات افزودن و ویرایش واسط‌کاربری می‌تواند بنابر طراحی نرم‌افزار ادامه پیدا کند. در این نوشتار تنها به افزودن یک Button بسنده می‌کنیم. اکنون لازم است واسط‌کاربری طراحی شده را به IDE منتقل کنیم. بدین منظور، ابزار SceneBuilder فایلی متنی در قالب XML تولید می‌کند. این فایل به مفسر جاوا نحوه‌ی چینش ابزارهای گرافیکی را بیان می‌کند. پسوند این فایل FXML می‌باشد. برای ایجاد این فایل، به مرحله‌ی پنجم بروید.

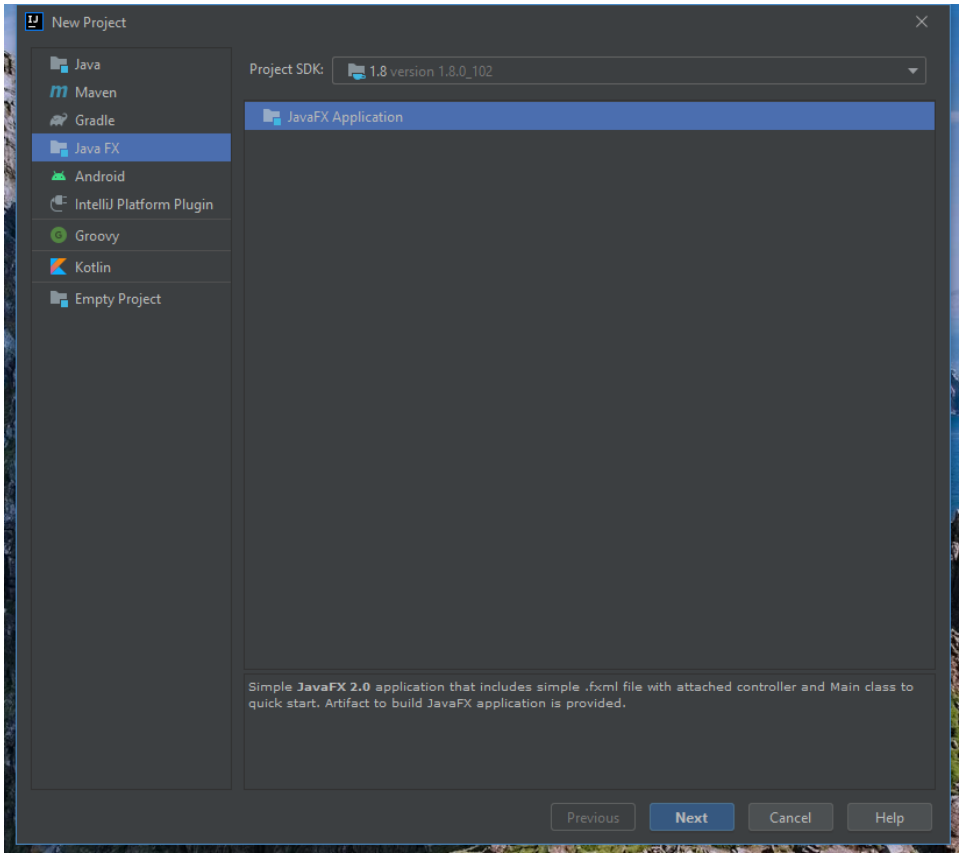
- مرحله‌ی پنجم: در این مرحله لازم است واسط‌کاربری طراحی شده را ذخیره کنید. بدین منظور از منوی File گزینه‌ی Save را انتخاب کنید. سپس مکانی را برای ذخیره‌ی فایل تعیین کنید. خواهید دید که فایل با پسوند fxml ایجاد خواهد شد.



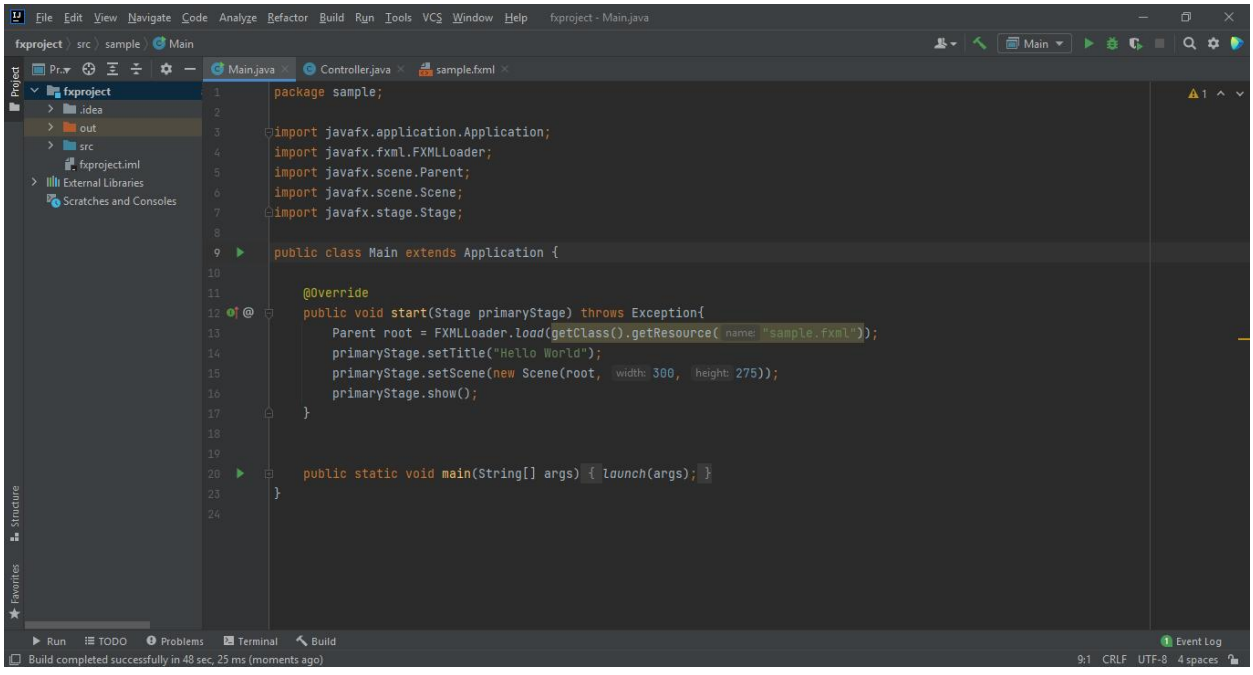

شکل ۹، ذخیره‌ی پروژه در ابزار SceneBuilder

اگر فایلی که در مرحله‌ی پنجم ذخیره می‌کنید را با ابزار Notepad باز کنید، متنی در قالب XML مشاهده خواهید کرد. این متن قادر است به مفسر جاوا، نحوه‌ی ترکیب‌بندی واسط‌کاربری را بیان کند. به بیان دیگر، این فایل متنی به زبانی که برای مفسرجاوا قابل درک است، طراحی انجام شده را به جاوا منتقل می‌کند. برای نمایش این واسط‌کاربری به مرحله‌ی بعدی بروید.

- مرحله‌ی ششم: برنامه‌ی Intellij را اجرا کنید و در آن، با استفاده از New Project پروژه‌ی جدیدی را ایجاد کنید. در فرایند ایجاد پروژه‌ی جدید، با انتخاب JavaFX از منوی سمت چپ، پروژه را از نوع JavaFX Application ایجاد کنید. این بخش در شکل ۱۰ نمایش داده می‌شود. مکان ذخیره‌ی پروژه می‌تواند آزادانه انتخاب شود. پس از ایجاد پروژه‌ی جدید، سه تب در پروژه ایجاد می‌شود. تب اول، شامل کلاس اصلی یا Main می‌باشد. همانطورکه می‌دانید این کلاس که شامل تابع main است، اولین کلاسی است که توسط مفسر جاوا برای اجرای برنامه مورد استفاده قرار می‌گیرد. تب دوم، یک کلاس Controller است. همانطور که در ابتدای نوشتار تشریح شد، هدف این کلاس اجرای محاسبات پس‌زمینه‌ی نرم‌افزار می‌باشد. تب سوم که sample.fxml نام دارد، حاوی متنی در قالب XML است که بیان‌گر چینش ابزارهای گرافیکی در واسط‌کاربری می‌باشد. در شکل ۱۱ نمایی از این مرحله نمایش داده می‌شود.




شکل ۱۰، پروژه‌ی جدید از نوع JavaFX Application انتخاب می‌شود.




شکل ۱۱، نمایی از پروژه‌ی ایجادشده در محیط Intellij

- مرحله‌ی هفتم: در کلاس Main تابعی به نام start وجود دارد. این تابع در ابتدای اجرای نرم‌افزار فراخوانی می‌شود. در این تابع می‌خواهیم محتویات فایل sample.fxml (در مرحله‌ی قبل درباره با این محتویات بحث شد.) را فراخوانی کرده و براساس آن، واسط‌کاربری را نمایش دهیم. بدین منظور کد زیر را بجای دستورات درون تابع start قرار می‌دهیم:

```
try {
  FXMLLoader loader = new FXMLLoader ( getClass().getResource( "FXML.fxml" ));
  loader.setController(new FXMLController());
  Parent root = loader.load();
  primaryStage.setScene(new Scene(root, 300, 275));
  primaryStage.show();
    } catch (IOException ex) {
          System.err.println(ex);
    }
```

در کد بالا، در خط اول، محتویات فایل FXML را دریافت کرده و براساس آن، یک شئ از نوع FXMLLoader ایجاد می‌شود. در خط دوم، یک شئ از کلاس Controller ساخته شده و به شئ loader الحاق می‌شود. یادآوری می‌شود پردازش‌های پس‌زمینه‌ی نرم‌افزار توسط کلاس Controller انجام می‌شوند. بنابراین لازم است شئ loader به شئ Controller دسترسی داشته باشد. در خطوط سوم، چهارم و پنجم محیط واسط‌کاربری آماده شده و نمایش داده می‌شود.

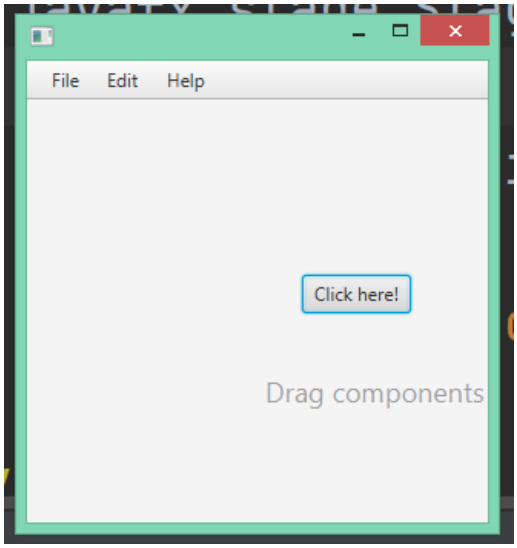
نکته‌ی اول: در صورتی که در پروژه‌ی شما، فایل محتویات fxml نام دیگری غیر از sample.fxml دارد، حتما آن را در خط اول کدبالا ویرایش کنید.

نکته‌ی دوم: در صورتی که کلاس کنترلر در پروژه‌ی شما، نامی جز FXMLController دارد، لازم است در خط دوم این تغییر اعمال شود. بدین معنا که نام تابع‌سازنده‌ی کلاس کنترلر به درستی فراخوانی شود.

نکته‌ی سوم: دراین کد، نیاز به افزودن برخی کتابخانه‌ها خواهید داشت. توجه داشته باشید استفاده از کتابخانه‌های javafx ارجحیت دارد.

- مرحله‌ی هشتم: در محیط Intellij به تب sample.fxml بروید. سپس، فایل fxml که در مرحله‌ی پنجم ذخیره کرده‌اید را با استفاده از Notepad باز کرده و تمامی محتویات آن را جایگزین محتویات تب sample.fxml در IDE کنید. (ابتدا تمام محتویات موجود در تب sample.fxml را حذف کنید؛ سپس با کپی کردن تمام متن موجود در Notepad آن را به تب sample.fxml در Intellij انتقال دهید.)

- مرحله‌ی نهم: برنامه را اجرا کنید. واسط‌کاربری که در SceneBuilder توسعه داده بودید نمایش داده می‌شود. این مرحله در شکل ۱۲ نشان داده شده است.

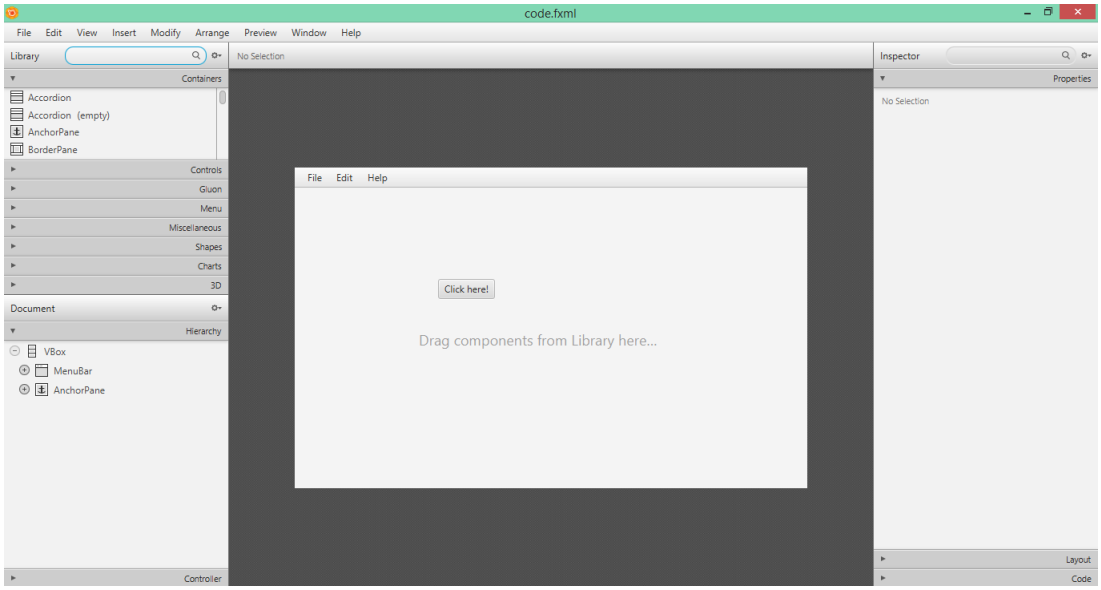



شکل ۱۲، اجرای برنامه و نمایش واسط‌کاربری آن

تا این مرحله، توانسته‌ایم واسط‌کاربری را با موفقیت توسعه دهیم. اما، کار هنوز تمام نـشده است! با کلیک روی Button هیچ اتفاقی نـمی‌افتد. در مراحل بعد، می‌خواهیم برنامه‌نویسی رویدادگرا[3] را به پروژه اضافه کنیم. بدین معنا که، با کلیک روی Button تابعی از کلاس کنترلر فراخوانی و اجرا شود. برای این امر، مراحل زیر را دنبال کنید:

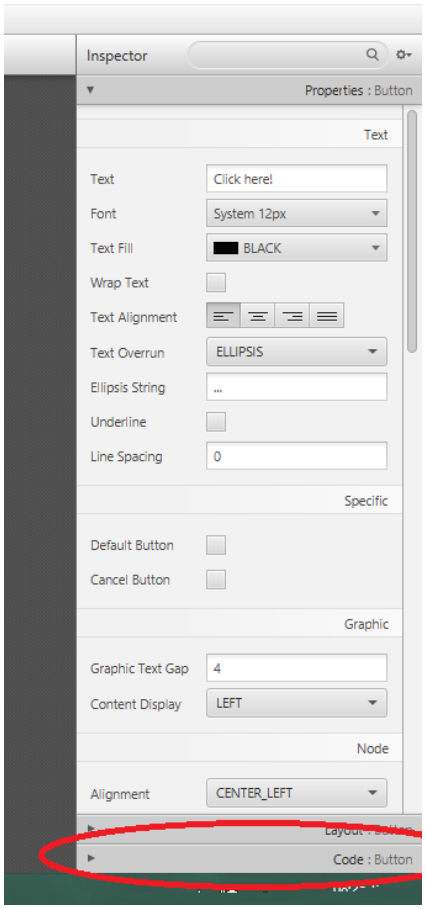
- مرحله‌ی دهم: ابزار SceneBuilder را مجددا اجرا کنید. با اجرای این برنامه و انتخاب گزینه‌ی Open Project می‌توانید واسط‌کاربری که قبلا ایجاد کرده‌اید را ویرایش کنید. در این بخش، ما فایلی که در مرحله‌ی پنجم ایجاد کرده‌ایم را انتخاب می‌کنیم. با این کار، واسط‌کاربری در برنامه نمایش داده می‌شود. این عمل را می‌توانید در شکل ۱۳ ملاحظه کنید.

[3] EventDriven

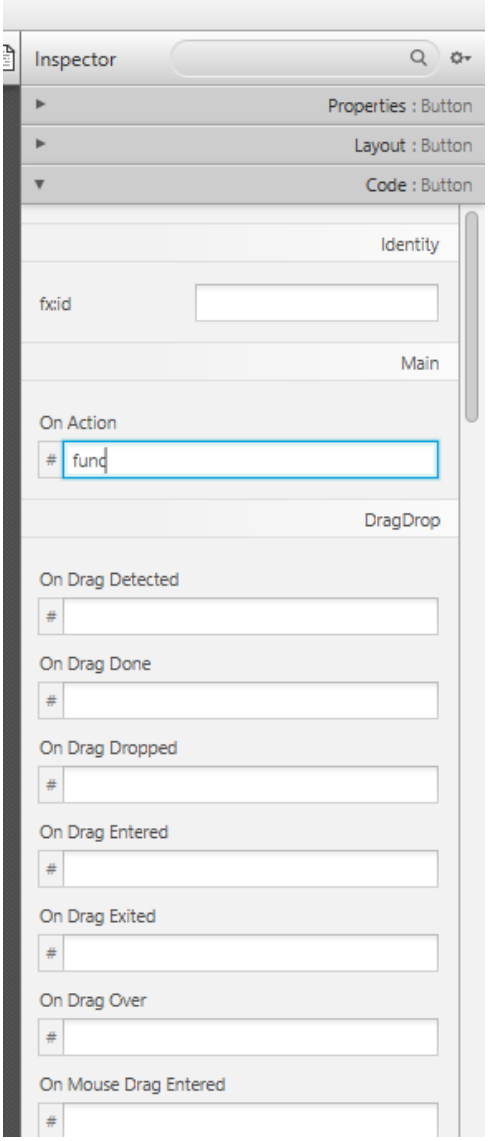



شکل ۱۳، نمایش واسط‌کاربری ذخیره‌شده در ابزار SceneBuilder

- مرحله‌ی یازدهم: اکنون، بر روی ابزار Button کلیک کنید تا انتخاب شود. با این عمل، منوی سمت راست مشخصات این ابزار را نمایش خواهد داد. این مشخصات را می‌توانید در شکل ۱۴ مشاهده کنید. این منو شامل سه بخش با نام‌های Properties و Layout و Code می‌باشد. بخش Code در شکل ۱۴ با یک بیضی قرمزرنگ نشان داده شده است. هدف ما در این بخش آن است که با کلیک‌کردن روی این Button تابعی به نام func فراخوانی شود. بنابراین باید از منوی Inspector وارد بخش Code شده و در آن، در قسمت On Action عبارت func را بنویسیم. این عمل در شکل ۱۵ نمایش داده می‌شود. با این عمل، به جاوا دستور می‌دهیم به محض کلیک کردن کاربر روی Button از کلاس کنترلر، تابعی به نام func را فراخوانی کند. (تعریف این تابع در کلاس کنترلر در مراحل بعدی تشریح خواهد شد.)




شکل ۱۴، منوی مشخصات برای ابزار Button




شکل ۱۵، مدیریت انواع رویدادها برای ابزار Button

- مرحله‌ی دوازدهم: اکنون، مشابه مرحله‌ی پنجم، پروژه‌ای را که در ابزار SceneBuilder تولید کرده‌اید، ذخیره کرده و سپس، مشابه مرحله‌ی هشتم، محتویات فایل ایجادشده را به محیط Intellij برده و بجای محتویات موجود در تب sample.fxml قرار دهید.

- مرحله‌ی سیزدهم: به کلاس Controller رفته و تابع func را به صورت زیر تعریف کنید:

```
@FXML
public void func(ActionEvent   e){
        JOptionPane.showMessageDialog(  null   ,   "Hello JavaFX!"    ) ;
}
```

نکته‌ی اول: کلاس ActionEvent که در ورودی تابع بیان شده است، از کلاس javafx.event استفاده می‌کند. توجه داشته باشید در کتابخانه‌ی java.awt نیز کلاسی با نام مشابه قرار دارد که **نـــباید** در این کلاس استفاده شود. (در غیر این صورت اجرای برنامه با مشکلاتی همراه خواهد بود!)

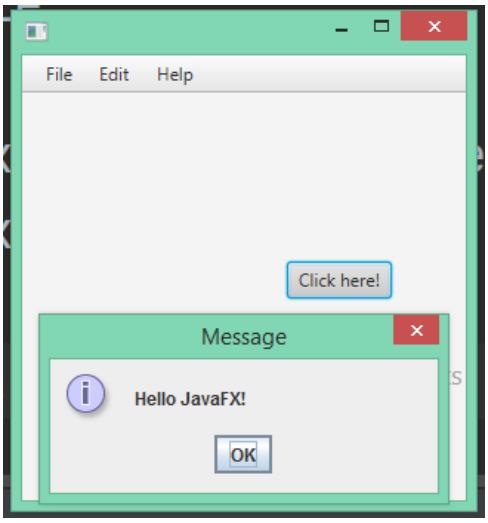
نکته‌ی دوم: از نوشتن برچسب @FXML قبل از تعریف تابع غافل نـــشوید. برای استفاده از این برچسب لازم است از کتابخانه‌ی javafx.fxml استفاده شود.

براساس نکات اول و دوم، در ابتدای تعریف کلاس کنترلر باید دستورات زیر حاضر باشند:

```
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
```

نکته‌ی سوم: دستور موجود در محتوای تابع func تنها برای نمایش یک پیام مورد استفاده قرار می‌گیرد. شما می‌توانید براساس اهداف و نیازمندی‌های برنامه، هرفرایند دیگری را درون این تابع کدنویسی کنید.

- مرحله‌ی چهاردهم: برنامه را اجرا کنید. با اجرای برنامه، خواهید دید درصورت کلیک روی گزینه‌ی Click here! پیامی با متن Hello JavaFX! همانطور که در تابع func تعریف شده است، نمایش داده می‌شود. این عمل در شکل ۱۶ مشاهده می‌شود.




شکل ۱۶، اجرای موفقیت‌آمیز برنامه

## چگونه به ابزارهای گرافیکی دسترسی داشته باشیم؟

شاید قصد داشته باشید در حین اجرای برنامه، برخی از ابزارهای گرافیکی را تغییر دهید. برای مثال، ممکن است بخواهید متن یک برچسب را در واسط‌گرافیکی ویرایش کنید. بدین منظور می‌توانید در کلاس کنترلر یک اشاره‌گر از جنس آن ابزار ایجاد کرده و نام آن را برابر با نامی که قبلا در ابزار SceneBuilder به عنوان fx:id تعیین کرده‌اید، نام‌گذاری کنید. (در شکل ۱۵ می‌توانید محل تعیین fx:id را برای ابزارهای گرافیکی در محیط SceneBuilder مشاهده کنید.) برنامه به صورت خودکار، این اشاره‌گر را به شئ مورد نظر شما نگاشت خواهد کرد. به عنوان مثال، فرض کنید در واسط‌کاربری، ابزاری از جنس Text را با fx:id برابر با mytext ایجاد کرده‌باشیم. برای دسترسی به این شئ کافیست در کلاس کنترلر، کد زیر را تعریف کنیم:

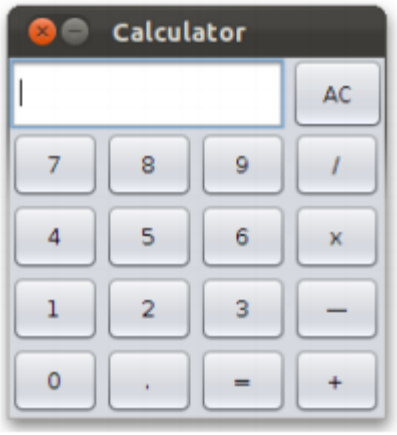
```
@FXML

public Text mytext ;
```

با این کار، می‌توانیم در توابع کلاس کنترلر، از این اشاره‌گر استفاده کرده و شئ Text که در واسط-کاربری موجود است را ویرایش کنیم.

## انجام دهید:

با استفاده از JavaFX یک ماشین حساب را طراحی و پیاده‌سازی کنید. ماشین‌حساب شما باید علاوه‌بر داشتن ظاهری زیبا و کاربرپسند، عملیات جمع، تفریق، ضرب، تقسیم، محاسبات توابع مثلثاتی مثل سینوس و کسینوس و توان را پشتیبانی کند. نمونه‌ای از این برنامه را در شکل ۱۷ مشاهده می‌کنید.

برای تولید این برنامه، لازم است از ابزار SceneBuilder برای توسعه‌ی واسط‌کاربری استفاده کنید. کدنویسی‌های مربوط به پردازش‌های پس‌زمینه نیز برعهده‌ی شماست.




شکل ۱۷، نمونه‌ای از واسط‌کاربری مربوط به برنامه‌ی ماشین‌حساب