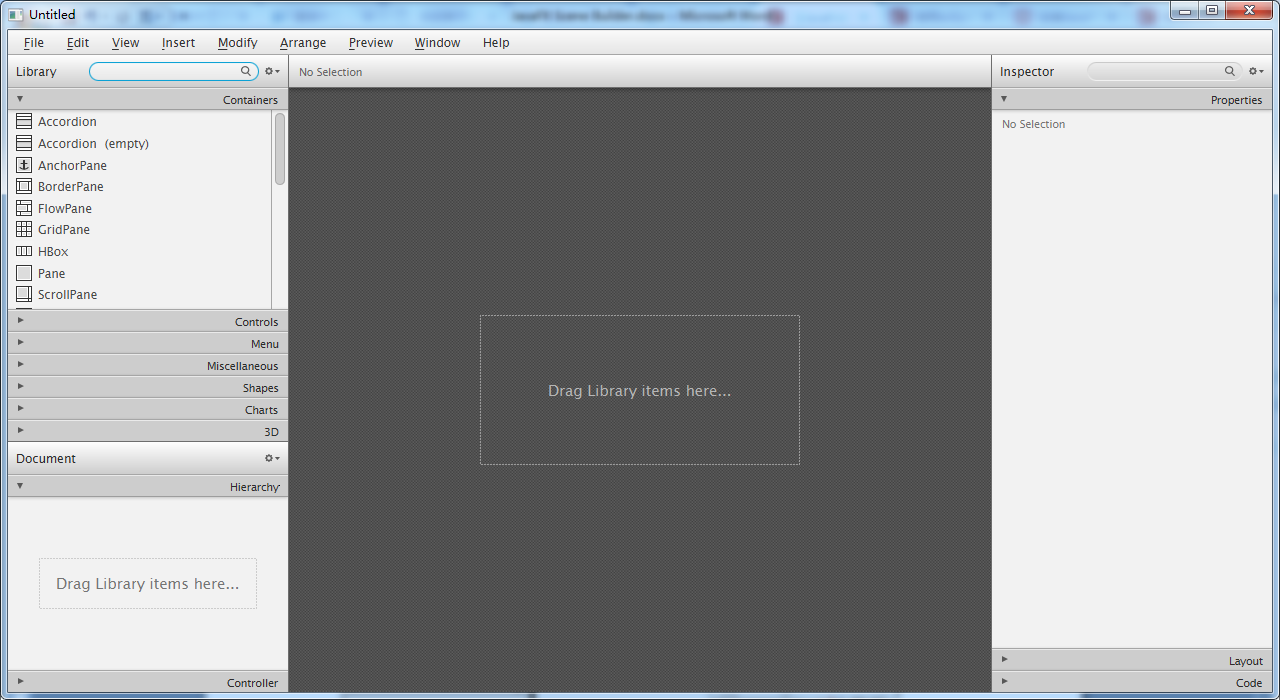
**1-JavaFX Scene Builder是什么？**

**JavaFX Scene Builder**是一种可视布局工具，允许用户快速设计**JavaFX**应用程序用户界面，而无需编码。用户可以将**UI**组件拖放到工作区，修改其属性，应用样式表，并且它们正在创建的布局的FXML代码将在后台自动生成。它的结果是一个FXML文件，然后可以通过绑定到应用程序的逻辑与Java项目组合。

**JavaFX Scene Builder**的开发界面如下 -



**JavaFX Scene Builder**可以集成到IDE(如[Eclipse](http://www.yiibai.com/eclipse/)和[Netbean](http://www.yiibai.com/netbean/))中。

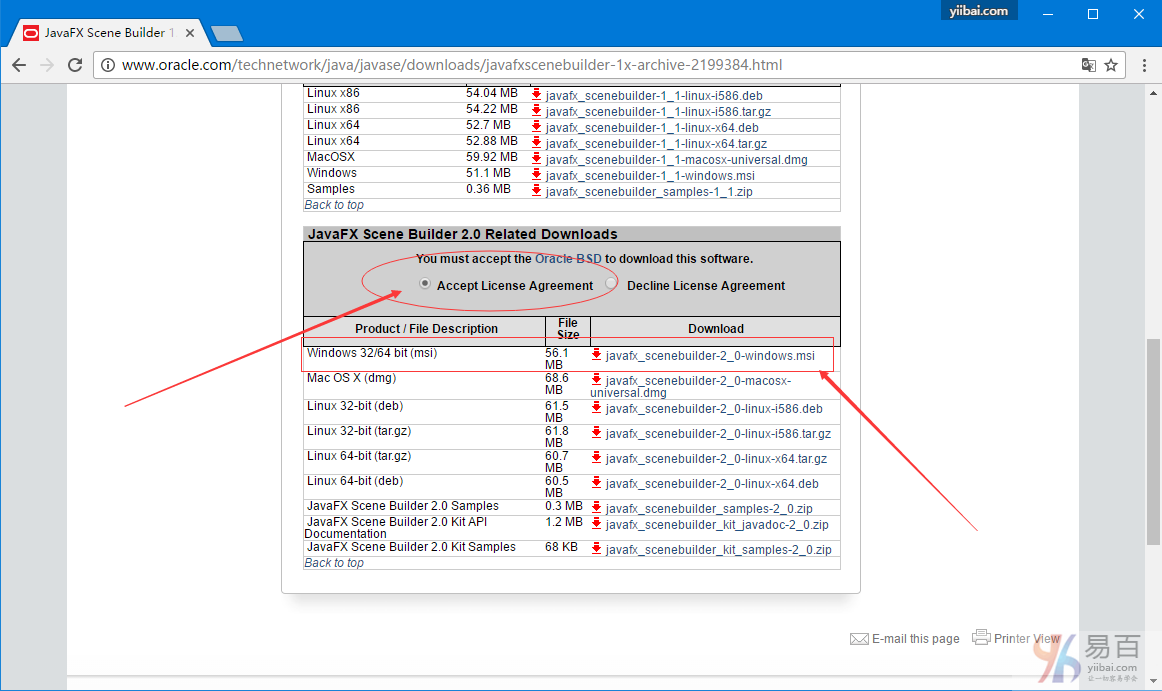
**2- 安装必须条件**

要将**Scene Builder**嵌入到**Eclipse**中，首先需要安装**e(fx)clipse**，这是**JavaFX**编程所需一组工具(库)。需要安装它作为**eclipse**插件。

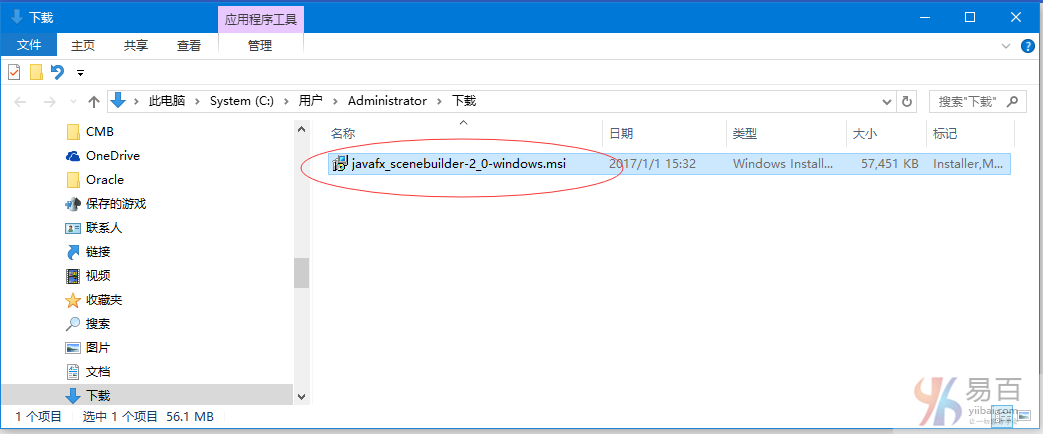
**3- 下载JavaFX Scene Builder**

下载**JavaFX Scene Builder**很简单，访问以下网址下载即可:

* <http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-1x-archive-2199384.html>



下载结果是一个名称为 javafx\_scenebuilder-2\_0-windows.msi 的文件，如下所示：



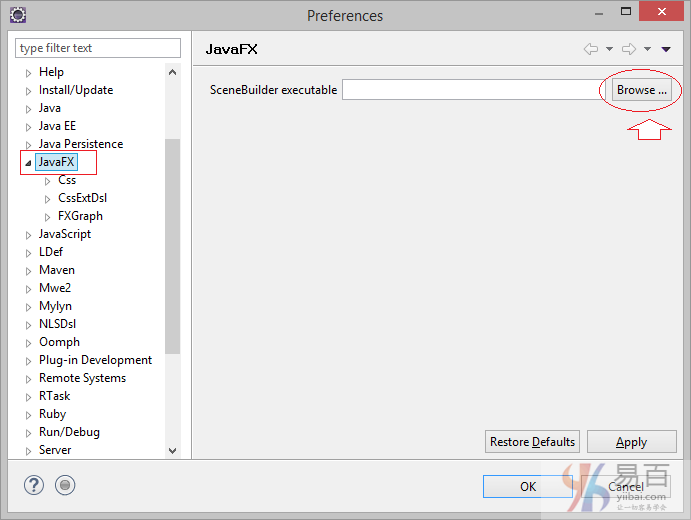
**4-安装JavaFX Scene Builder**

双击 javafx\_scenebuilder-2\_0-windows.msi 文件来安装 **JavaFX Scene Builder** 。-

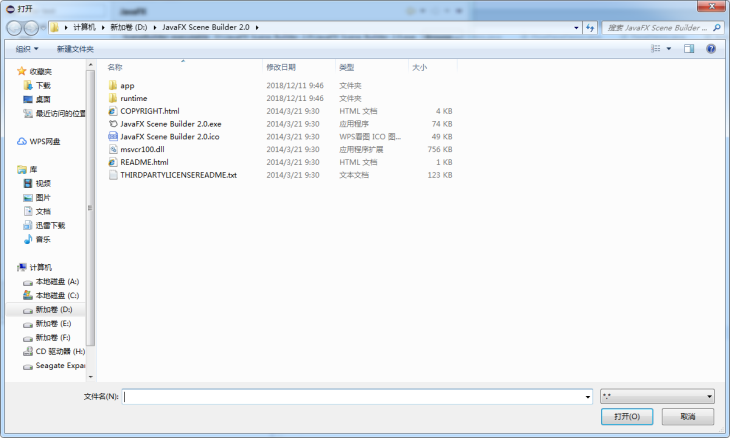
**5-配置Eclipse以使用Scene Builder**

启动**eclipse**，并选择：

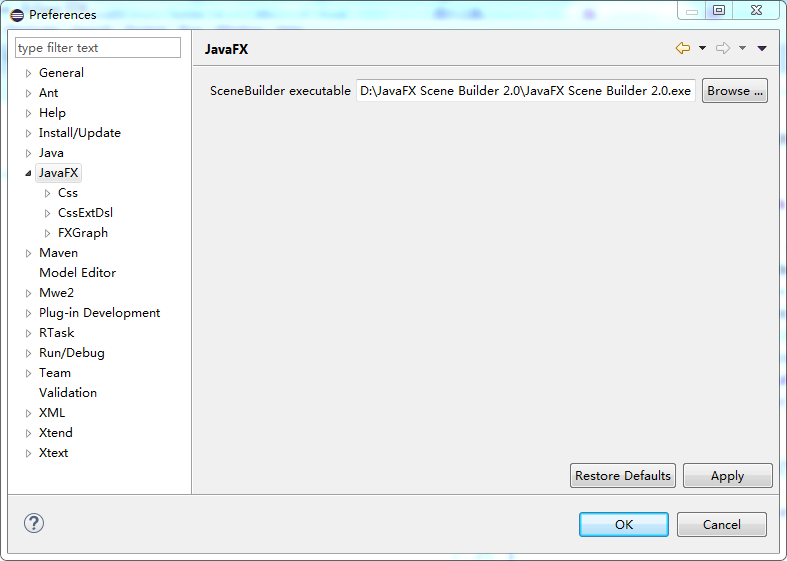
* **Window -> References**



指向**JavaFX Scene Builder**的exe可执行文件位置(也就是**JavaFX Scene Builder**的安装目录下)，在这个示例中安装的位置是： D:\JavaFX Scene Builder 2.0\JavaFX Scene Builder 2.0.exe 。



最后，点击 **OK** 完成！如下图所示 -



**6-创建一个新的 JavaFX 项目**

在 Eclipse（已安装 e(fx)clipse 的）中，点击 File | New | Other… 并选择 \*JavaFX Project\*。 指定这个项目的名字（e.g. \*AddressApp\*）并点击 \*Finish\*。

如果 application 包被自动创建，那么删除它和它的内容。

### 创建包

[**Model-View-Controller** (MVC)](http://zh.wikipedia.org/wiki/MVC)是一个非常重要的软件设计原则。按照MVC模式可以将我们的应用程序划分成3个部分，然后为这每一部分建立自己的包 (在源代码文件夹上右键， 选择 新建 | 包):

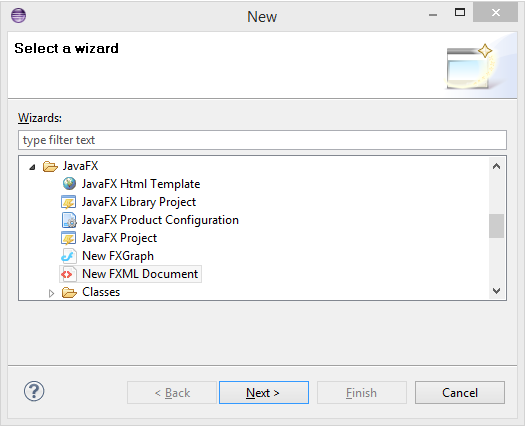
* ch.makery.address - 放置所有的控制器类(也就是应用程序的业务逻辑)
* ch.makery.address.model - 放置所有的模型类
* ch.makery.address.view - 放置所有界面和控件类

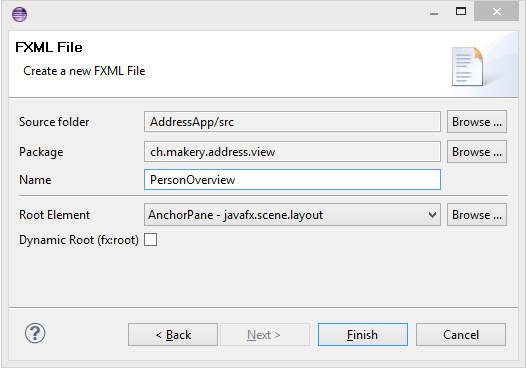
**注意:** view包里可能会包含一些控制器类，它可以直接被单个的view引用，我们叫它 **视图-控制器**。

**创建FXML布局文件**

有两种方式来创建用户界面，一种是通过XML文件来定义，另外一种则是直接通过java代码来创建. 这两种方式你都可以在网上搜到. 我们这里将使用XML的方式来创建大部分的界面。因为这种方式将会更好的将你的业务逻辑和你的界面开来，以保持代码的简洁。在接下来的内容里，我们将会介绍使用Scene Builder(所见即所得)来编辑我们的XML布局文件，它可以避免我们直接去修改XML文件。

在view包上右键创建一个新\*FXML Document\*，把它命名为PersonOverview。

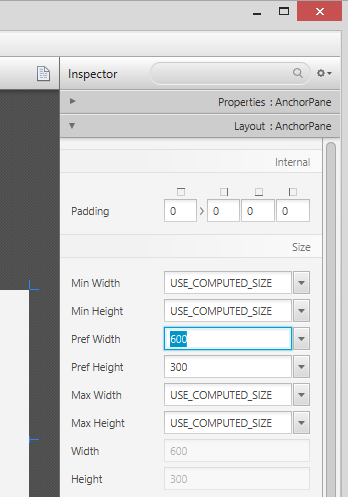
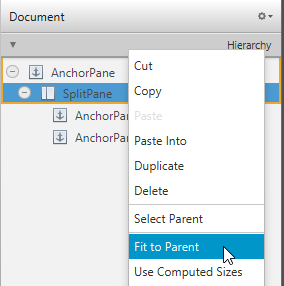
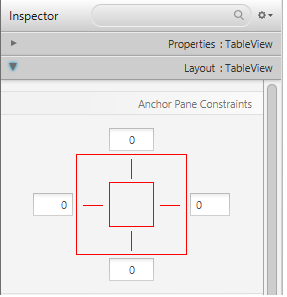
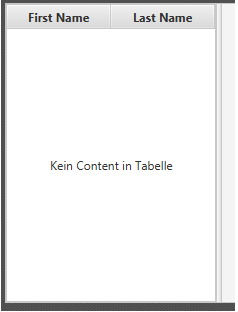
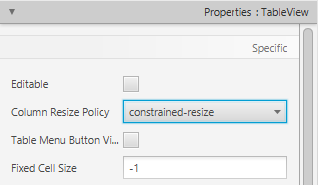
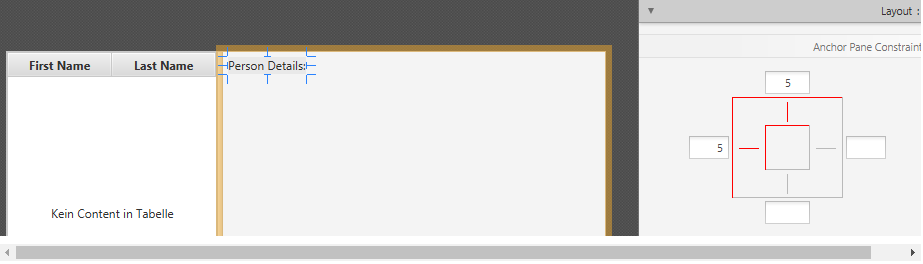
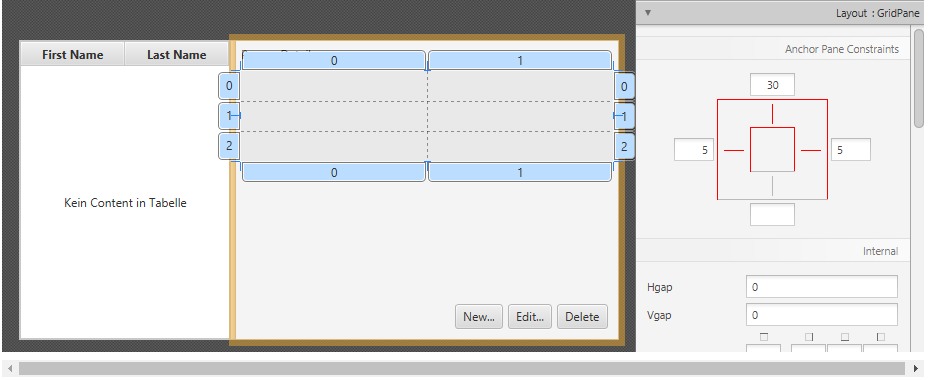
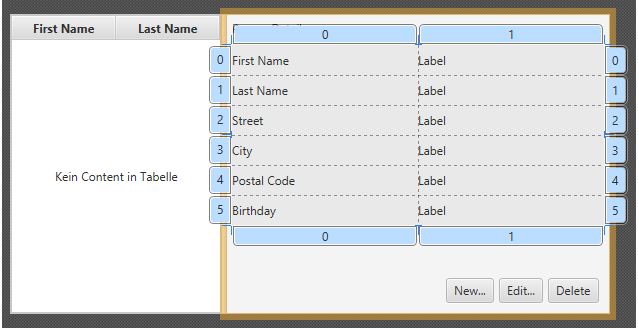
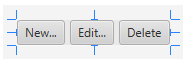
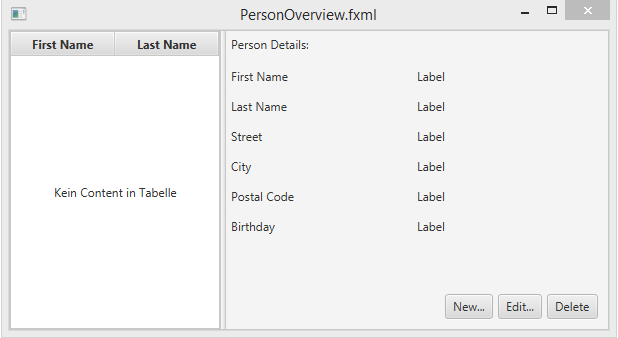




**用Scene Builder来设计你的界面**

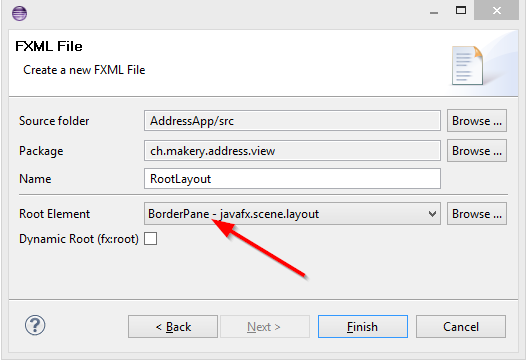
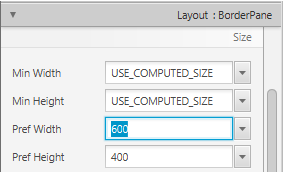
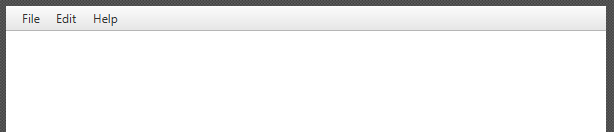
**注意:** 你可以下载这部分教程的源码，它里面已经包含了设计好的布局文件。

在PersonOverview.fxml 右键选择 \*Open with Scene Builder\*，那么你将会在打开的Scene Builder里面看到一个固定的界面设计区域(在整个界面的左边)。

1. 选中这个界面设计区域，你就可以在右边的属性设置栏中对它的尺寸进行修改:  
   
2. 从Scene Builder的左边控件栏中拖拽一个 Splite Pane(Horizontal Flow) 到界面设计区域，在Builder的右边视图结构中选择刚添加的Pane，在弹出的右键菜单中选择 Fit to Parent 。  
   
3. 同样从左边的控件栏中拖拽一个 TableView 到 SplitPane 的左边，选择这个TableView(而不是它的列)对它的布局进行设置，你可以在 AnchorPane 中对这个TableView四个边的外边距进行调节。([more information on Layouts](http://docs.oracle.com/javase/8/javafx/layout-tutorial/builtin_layouts.htm)).  
   
4. 点击菜单中的 Preview | Show Preview in Window 可以预览你设计好的界面，试着缩放预览的界面，你会发现TableView会随着窗口的缩放而变化。
5. 修改TableView中的列名字，”First Name” and “Last Name”，在右边面板中的属性设置项  
   
6. 选择这个 TableView ，在右边面板中将它的 Column Resize Policy 修改成 constrained-resize (同样是在属性设置项里面)。确保这个TableView的列能够铺满所有的可用空间。  
   
7. 添加一个 Label 到 \*SplitePane\*的右边部分，并设置它的显示文字为 “Person Details” (提示: 你可以通过搜索来找到 Label 这个控件)。 使用anchors来调节这个控件的布局位置。  
   
8. 再添加一个 GridPane \*SplitePane\*的右边部分, 使用anchors来调节这个控件的布局位置。  
   
9. 按照下面的图添加多个 \*Lables\*到表格中去。
10. 注意: 添加一个控件到已经存在的行里面去，你可在这行的行号上右键选择 “Add Row”。  
    
11. 添加3个按钮到这个 GridPane 的下面。 小提示: 选择这3个按钮，右键 \*Wrap In | HBox\*，那么它们会被放置到一个HBox里面。 你可能需要对这个HBox指定一个 spacing,同时也需要设置它们的右边和下边的anchors。  
    
12. 那么基本已经完成了界面的设计，你可以通过 Preview 来预览一下你设计的界面，同时缩放一下窗口来检验一下各个控件的位置是否正确。  
    

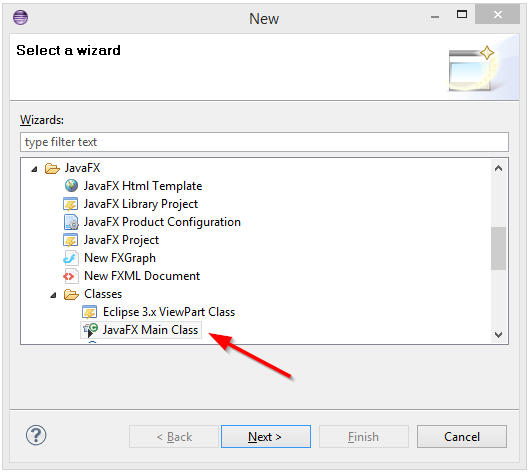
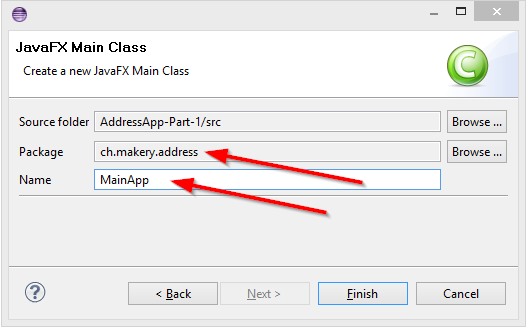
## 创建主应用程序

我们还需要新建一个\*FXML\*文件来做为主布局文件，它将包含菜单栏并存放我们之前创建的布局文件 PersonOverview.fxml 。

1. 在view包里面创建一个新的 FXML Document 叫做 RootLayout.fxml, 这一次，选择 BorderPane 做为它的根节点  
   
2. 在Scene Builder中打开 RootLayout.fxml。
3. 通过设置 Pref Width 为600和 Pref Height 为400来改变这个 \*BorderPane\*的尺寸。  
   
4. 在最顶上添加一个 \*MenuBar\*，先不去给这个菜单添加任何的功能。  
   

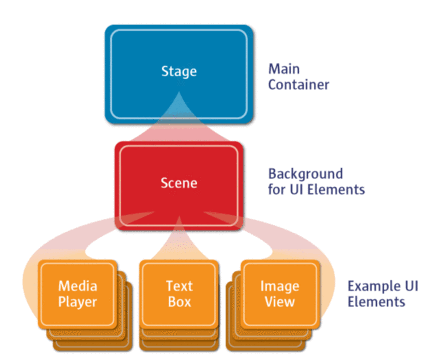
### The JavaFX Main Class

现在，我们需要创建一个 **main java class** 用来加载 RootLayout.fxml ，同时添加 PersonOverview.fxml 到\*RootLayout.fxml\*中去，这个main class将做为我们这个应用程序的入口。

1. 在工程上右键选择 \*New | Other…\*，然后选择 \*JavaFX Main Class\*。  
   
2. 将这个class命名为 MainApp，将它放置到controller包中，也就是上面建的 ch.makery.address (注意: 这个包下有两个子包，分别是view 和 model)。  
   

你可能注意到了IDE生成的 MainApp.java 继承自 Application 同时包含了两个方法， 这是一个JavaFX应用程序的最基本的代码结构，这里最重要的方法是 start(Stage primaryStage) ，它将会在应用程序运行时通过内部的 main 方法自动调用。

正如你所看到的，这个start(...) 方法会接收一个 Stage 类型的参数，下面的图向你展示了一个JavaFX应用程序的基本结构。

  
Image Source: [*http://www.oracle.com*](http://www.oracle.com/)

**一切看起来象是剧场里表演**: 这里的 Stage 是一个主容器，它就是我们通常所认为的窗口（有边，高和宽，还有关闭按钮）。在这个 Stage 里面，你可以放置一个 Scene，当然你可以切换别的 Scene，而在这个 Scene 里面，我们就可以放置各种各样的控件。

更详细的信息，你可以参考 [Working with the JavaFX Scene Graph](http://docs.oracle.com/javase/8/javafx/scene-graph-tutorial/scenegraph.htm).

打开 MainApp.java，将已有的代码替换成下面的代码:

package ch.makery.address;

import java.io.IOException;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

public class MainApp extends Application {

private Stage primaryStage;

private BorderPane rootLayout;

@Override

public void start(Stage primaryStage) {

this.primaryStage = primaryStage;

this.primaryStage.setTitle("AddressApp");

initRootLayout();

showPersonOverview();

}

/\*\*

\* Initializes the root layout.

\*/

public void initRootLayout() {

try {

// Load root layout from fxml file.

FXMLLoader loader = new FXMLLoader();

loader.setLocation(MainApp.class.getResource("view/RootLayout.fxml"));

rootLayout = (BorderPane) loader.load();

// Show the scene containing the root layout.

Scene scene = new Scene(rootLayout);

primaryStage.setScene(scene);

primaryStage.show();

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* Shows the person overview inside the root layout.

\*/

public void showPersonOverview() {

try {

// Load person overview.

FXMLLoader loader = new FXMLLoader();

loader.setLocation(MainApp.class.getResource("view/PersonOverview.fxml"));

AnchorPane personOverview = (AnchorPane) loader.load();

// Set person overview into the center of root layout.

rootLayout.setCenter(personOverview);

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* Returns the main stage.

\* @return

\*/

public Stage getPrimaryStage() {

return primaryStage;

}

public static void main(String[] args) {

launch(args);

}

}

代码中的注释会给你一些小提示，注明代码的含义。

如果你现在就运行这个程序，那么你将会看到和这篇文章开头所展示的图片那样的界面。

### 你有可能遇见的问题

如果你的应用程序找不到你所指定的 fxml 布局文件，那么系统会提示以下的错误：

java.lang.IllegalStateException: Location is not set.

你可以检查一下你的 fxml 文件名是否拼写错误