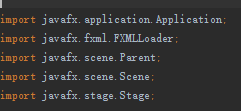
**创建一个基本窗口**

1.导入相关包



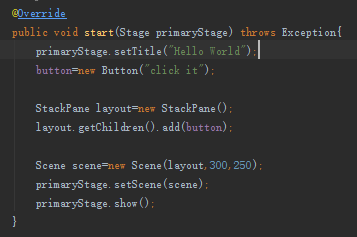
2.创建一个javafx程序的第一步就是extends为Application



3.在main程序中， 输入 lanch(args) 即可启动为javafx程序



4．创建一个主页面首先要重写start 函数



这里Stage是整个窗口，



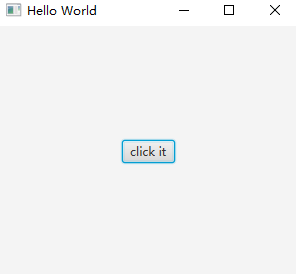
如此就设置了窗口的标题



窗口成员的关系如下:

Stage🡪Scene🡪layout🡪component

然后就得到了一个基本的窗口



**处理事件**

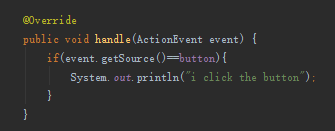
首先实现接口

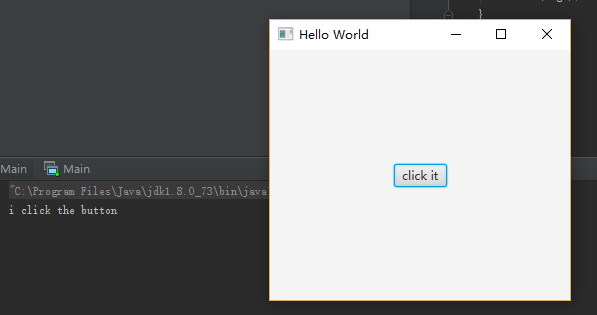


然后给按钮加上事件



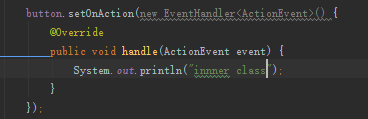
在事件处理器处识别该按钮，然后处理该事件





**匿名方法**

也可以选择不使用implements,而使用匿名函数代替



如果有多个按钮，就不需要判断source了

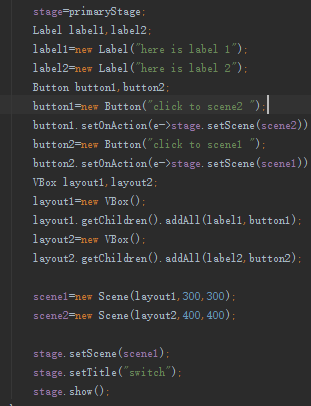
**Lambda表达式**

Lambda表达式是Java SE 8中一个重要的新特性。lambda表达式允许你通过表达式来代替功能接口。 lambda表达式就和方法一样,它提供了一个正常的参数列表和一个使用这些参数的主体(body,可以是一个表达式或一个代码块)。  
 Lambda表达式还增强了集合库。 Java SE 8添加了2个对集合数据进行批量操作的包: java.util.function 包以及java.util.stream 包。 流(stream)就如同迭代器(iterator),但附加了许多额外的功能。 总的来说,lambda表达式和 stream 是自Java语言添加泛型(Generics)和注解(annotation)以来最大的变化。

除了内部类，注册事件也可以使用lambda表达式，十分简洁。



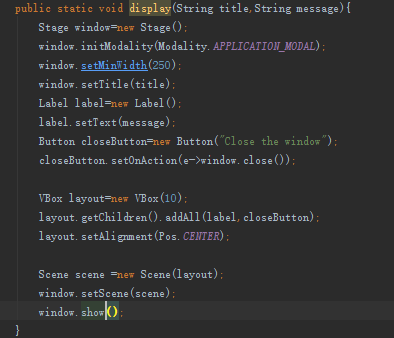
**切换scene**



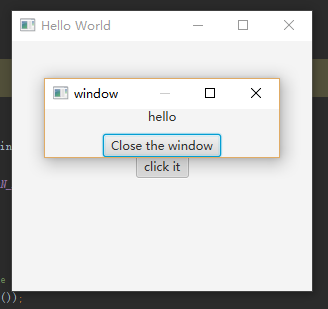
在每个scene内加入一个按钮，按钮可以切换stage的scene

**创建弹窗**

Modality.APPLICATION\_MIDAL禁止点击该窗口外



只有关闭该窗口才能点击其他

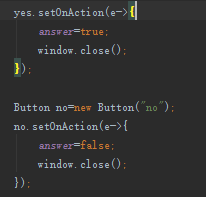


**窗口之间的交互**

主页面点击按钮触发confirm\_box

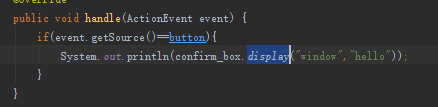
在confirm\_box处设置display函数返回值





按yes会返回true，按no返回false

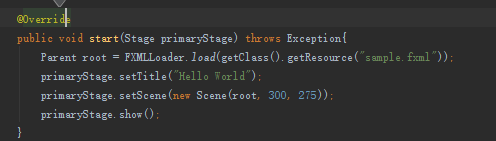
然后在主页面处打印返回值



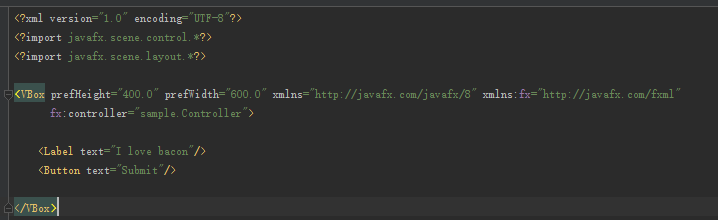
**创建FXML**

FXML是基于XML的语言，它提供与应用逻辑代码分离的构建用户界面的结构。展现层和应用逻辑的分离对web开发者更有吸引力：因为他们可以利用[Java](http://lib.csdn.net/base/17)组件组装用户界面而不需要掌握获取并填充数据的代码。

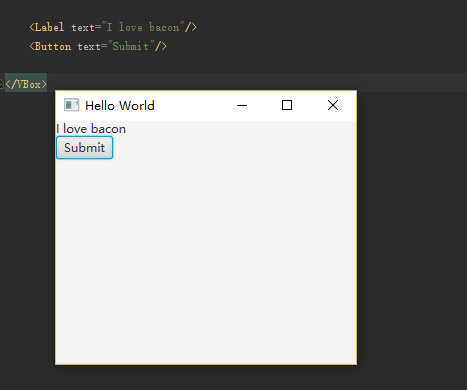
首先要在start处调用



然后写fxml

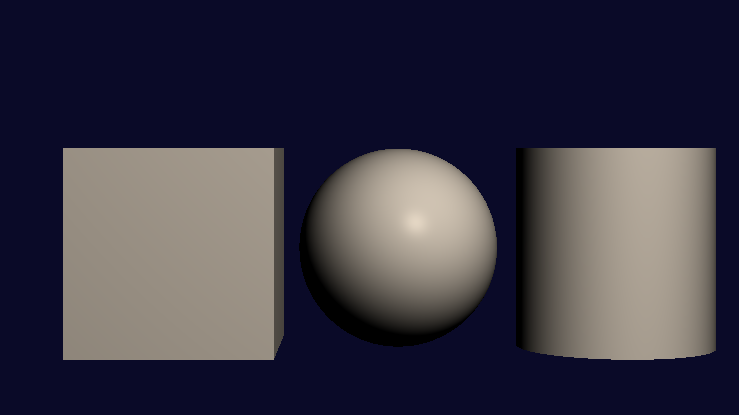


运行结果如图

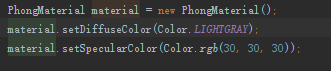


Fxml可以使用scene builder 轻松修改。不过官网已经不提供下载了，网上还有人维护。

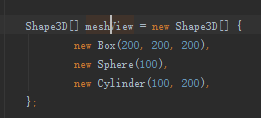
**创建3d模型**



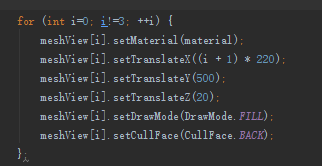
创建阴影材质



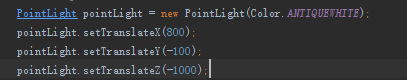
新建3d模型



设置属性



设置灯光



最后放到场景中

