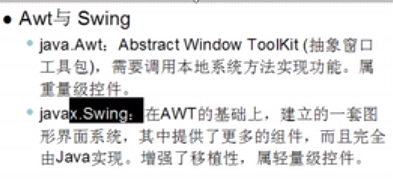
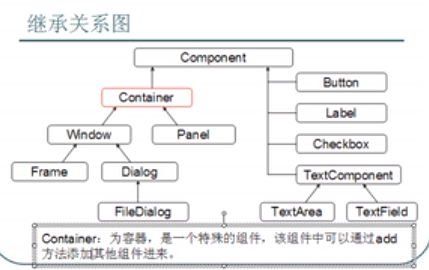
**P278 GUI (概述）**

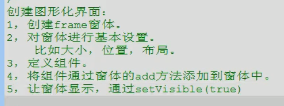




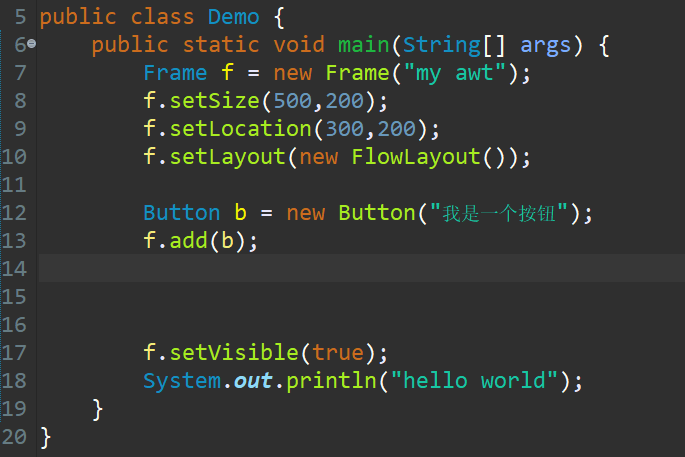
**P279 GUI (布局）**

//

**P280 GUI (Frame）**



举个例子，下面代码能生成这样一个窗口





**P281 GUI (事件监听机制）**

//

**P282 GUI (窗体事件）**

需求：关闭窗口

此时事件源是窗体，所以要定义一个监听器到窗体上。

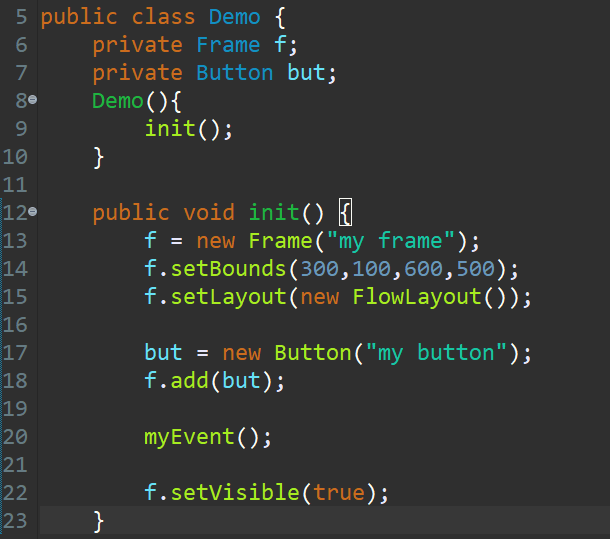
窗体接收的监听器WindowListener是一个接口，我们如果要实现该窗口，必须覆盖全部的7个方法。而WindowListener的子类WindowAdapter已经实现了WindowListener接口并覆盖了其中的所有方法，那么我们只需要继承WindowAdapter并覆盖我们需要的方法即可。



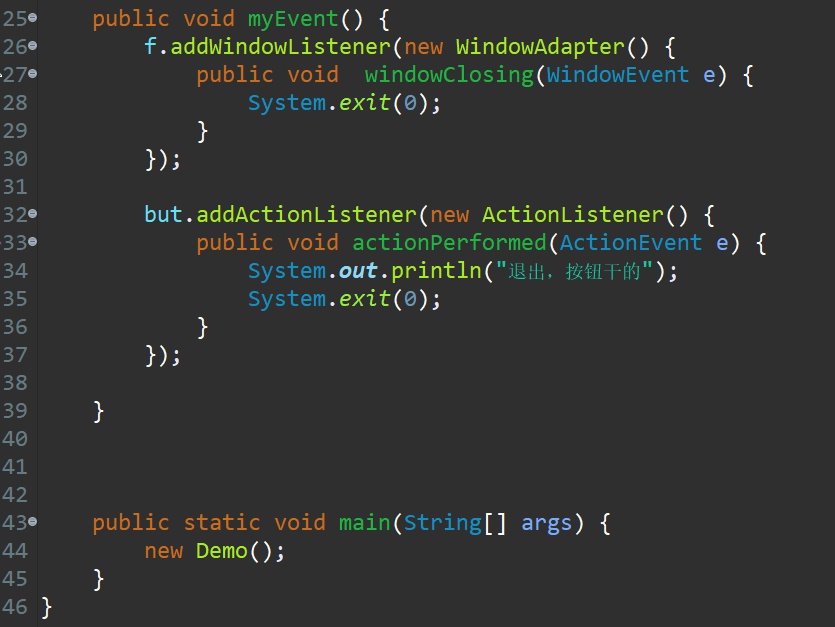
14-25行定义了匿名内部类。如何理解覆盖了三个方法中，传入的WindowEvent e？

覆盖了三个方法：关闭，激活和打开。然后一旦有对应的关闭，激活或打开行为，就执行对应的关闭，激活或打开的方法体。

**P283 GUI (Action事件）**



将初始化时要干的封装成一个init方法，通过构造函数去调用。

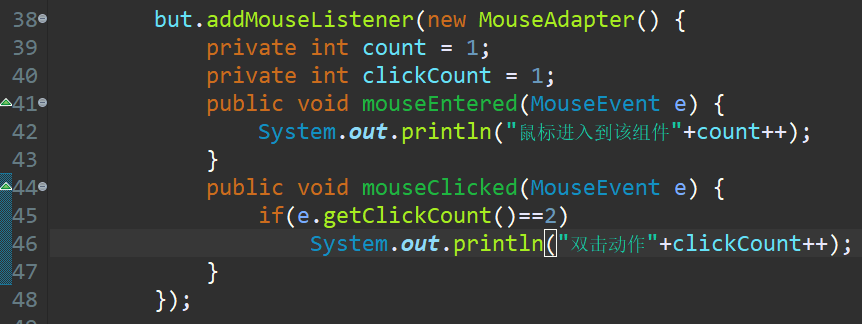


将事件监听和布局分开，更加条理清晰。

通过26行的窗口监听和32行的按钮监听，对于监听机制有了更好的理解：在布局时定义了这些按钮，但是这些按钮并没有实际的功能。窗口适配器WindowAdapter（WindowListener的子类）有7个方法，意味着它监听7中行为，一旦有符合7中行为的行为发生时，就执行对应行为的方法体。而button的监听器只有actionPerformed一个方法，可以理解为只监听按下按钮这一行为。一旦按下按钮这一行为发生，就执行其方法体。至于传入方法的参数WindowEvent e 和ActionEvent e有点像工具人。

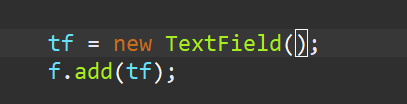
**P284 GUI (鼠标事件）**

为上面的button按钮添加鼠标事件：在myEvent方法中添加即可。

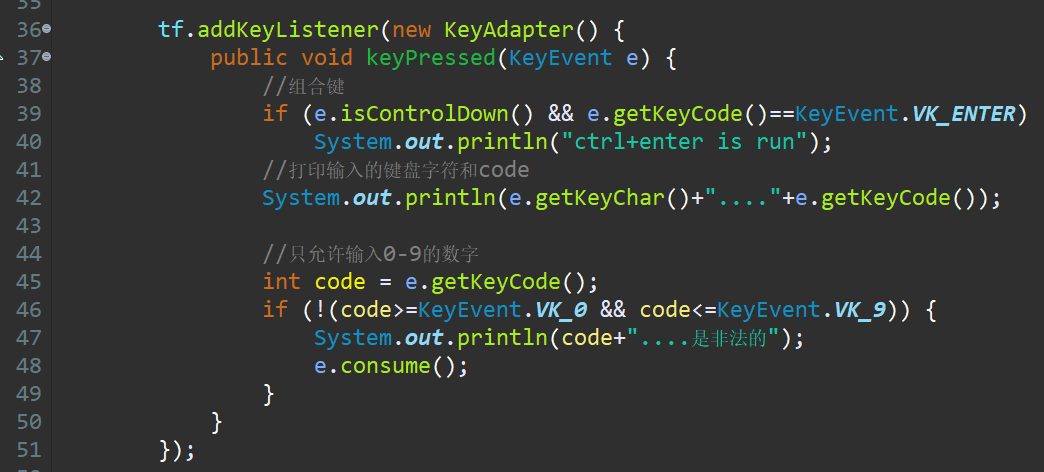


**P285 GUI (键盘事件）**

在Frame中添加新的文本框

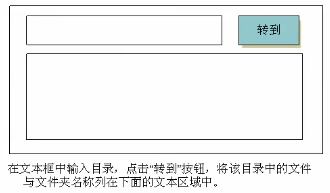


可以往构造函数中传入整数来控制文本框的长度

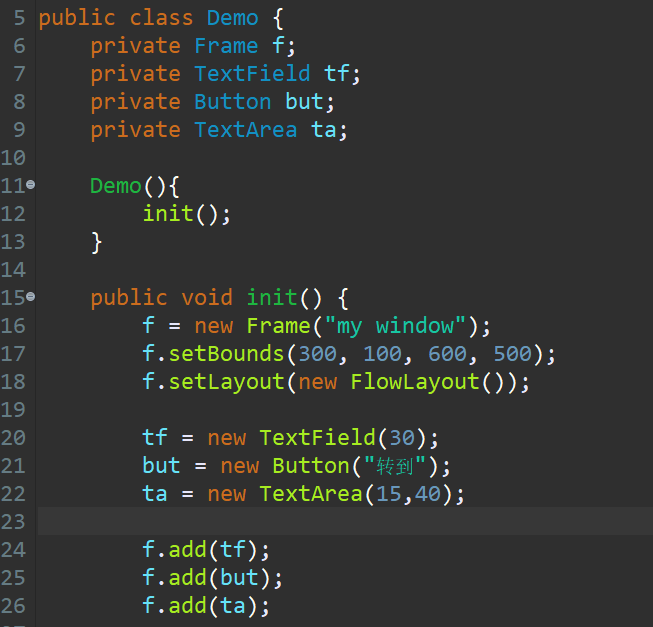


**P286 GUI (练习-列出指定目录内容）**

需求：

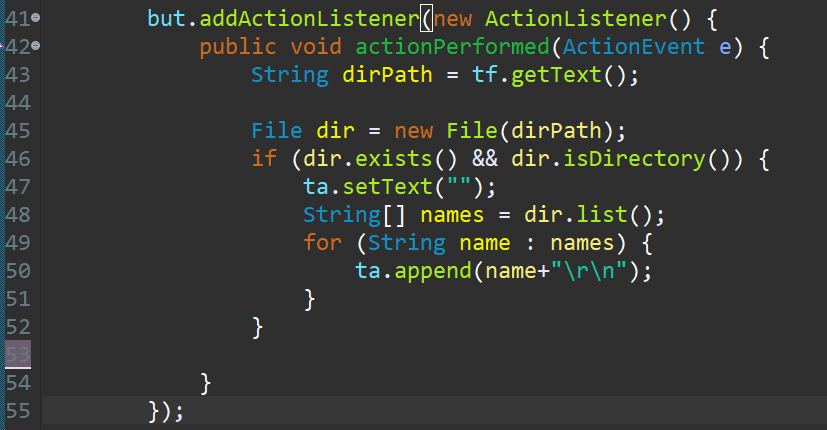


首先，将界面弄出来





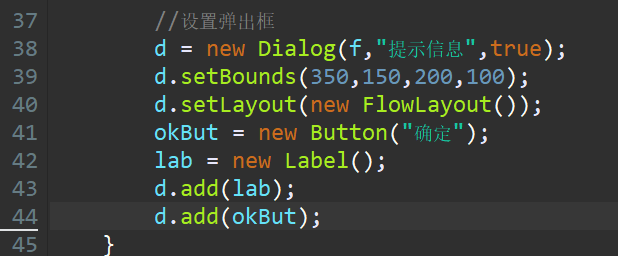
事件源是button



添加到37-38行之间

**P287 GUI (对话框Dialog）**





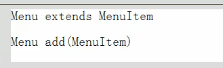




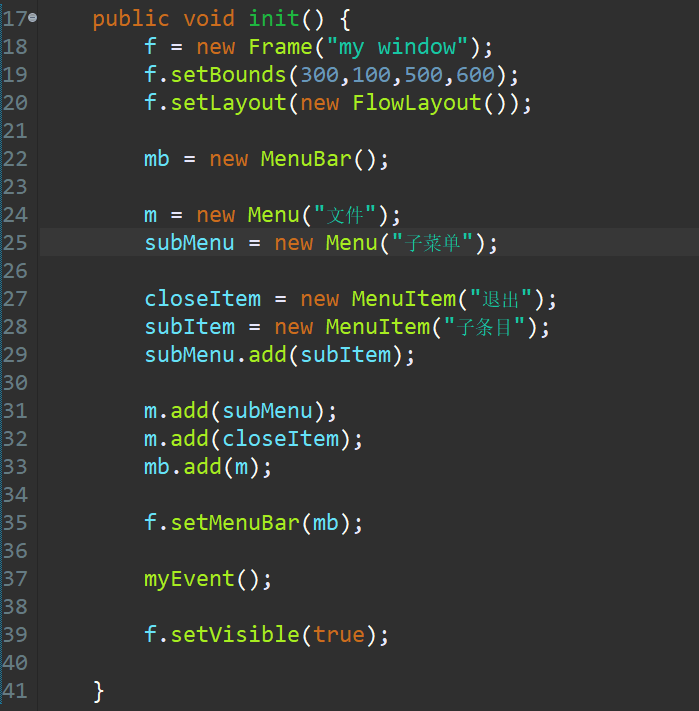


**P288 GUI (菜单）**

要想有子菜单，创建Menu，没有子菜单，创建MenuItem



看下初始化过程就完事。监听跟前面的一样，MenuItem支持addActionListener

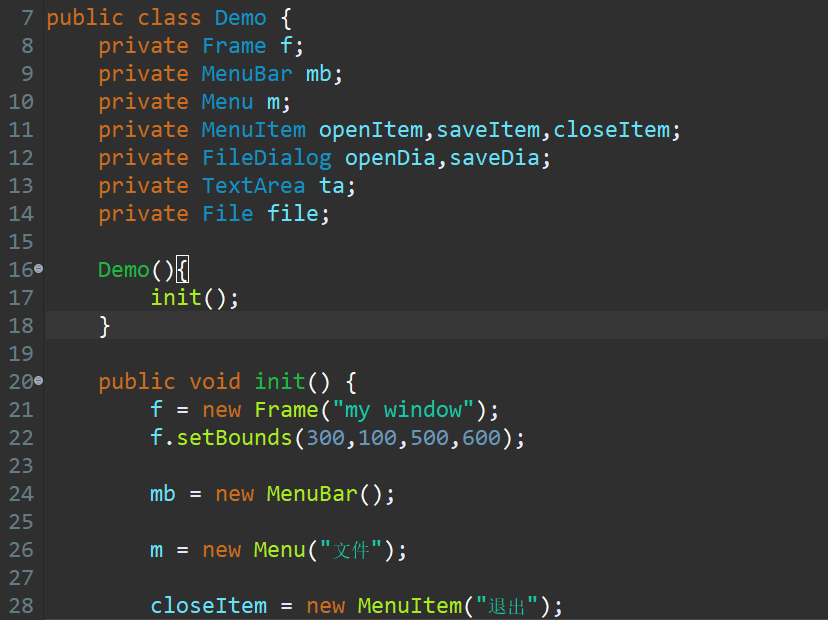


**P289 GUI (练习-打开文件）**

//

**P290 GUI (练习-保存文件）**

设计窗口

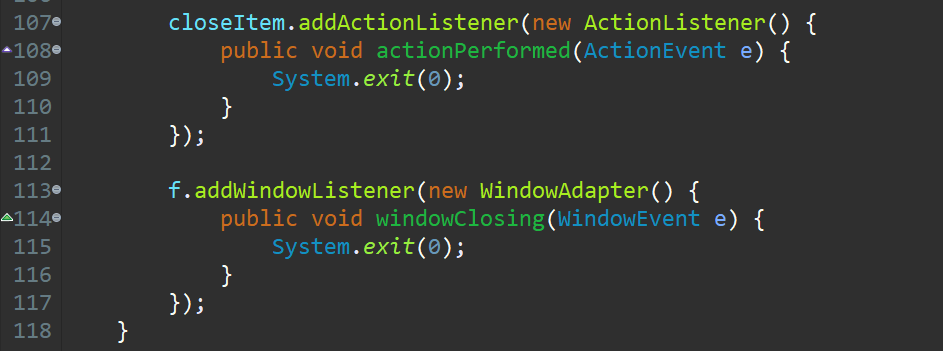




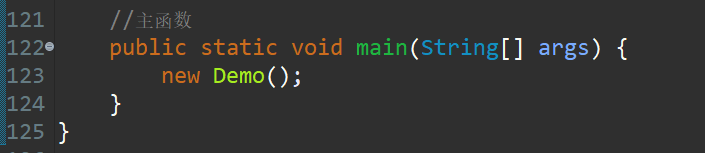
监听事件，包括saveItem，openItem，closeItem和窗口本身







主函数



**P291 GUI (jar包双击执行）**

1：到java所在目录下编译（因为你一般还导入了其他包）



2：你获得了带有很多class文件的test文件夹，但是jar包并不知道主函数在哪，所以要写个配置信息



3.生成jar包

