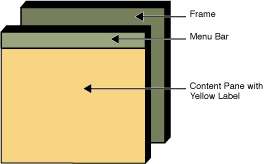
**Java图形界面**

组件和容器？

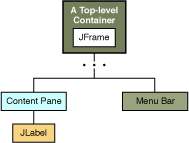
http://docs.oracle.com/javase/tutorial/uiswing/components/button.html

要掌握Java图形界面开发，首先需理解组件和容量两个概念。组件放置在容器中，容器包含组件。日常生活中，书包就是一个容器，而书包里的一本书、一支笔都是组件。当你把书包放到车里时，车就变成容器，而书包就成为组件。容器可以被放置到更大的容器中，因此容器可以包含组件，也可以包含其他的容器。容器之间的包含关系也反映容器之间层级关系，上一级的容器自然可以包含下一级的容器。组件如何放置到容器中？Swing中的各种Layout负责解决。

Java提供现成的用户界面程序开发工具包Swing。开发人员只需调用Swing中的组件和容器，就可以完成界面开发，十分便利。前面已经提过容器的层级问题，JFrame是Swing中常用的最高层级。图\*\*是一JFrame的例子，Frame内部放置一菜单栏和contentPane，contentPane中放置一黄色的标签。

上述各对象的层级关系如下：



注意JLabel没有直接放置，而是通过简接地放置在Content Pane实现放置在Frame中。具体的代码见文件夹MyFrame。注意放置JLabel部分代码：

frame.getContentPane().add(yellowLabel, BorderLayout.CENTER);

add方法中有BorderLayout.CENTER参数，就是设置Label在ContentPane中放置的方式与位置。当然还有很多其他的Layout方式，读者可以在应用时查阅官方API文档或者其他相关资料，此处不再赘述。

尝试在Content Pane中增加一个按钮组件，Swing中的按钮是类JButton，代码如下：

jb = new JButton("press me");

jb.setPreferredSize(new Dimension(50, 20));

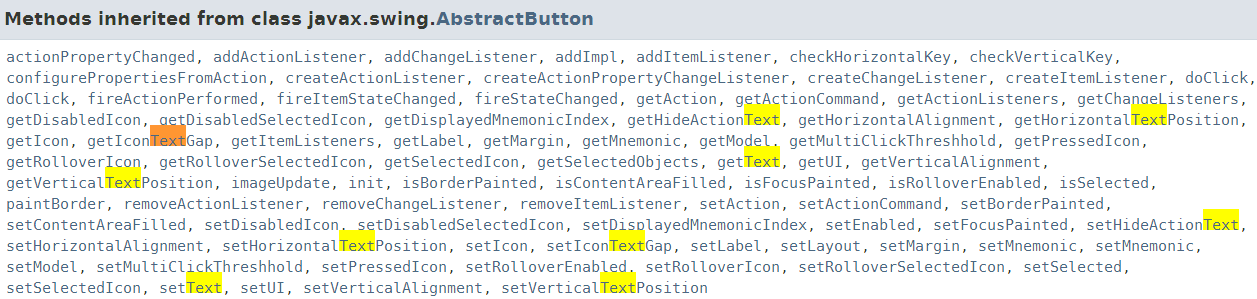
按钮被放置到content pane的下方。

日常经验显示当按钮被点击后，能看到界面对应的反馈，比如弹出新的对话框。整个过程可由如下的链条描述：点击动作->触发事件->捕捉事件->进行对应的处理。当鼠标点击按钮时，产生按钮点击事件，如何区分不同按钮的点击事件？对按钮而言，可以给每个按钮设置一个身份标签，不同的按钮拥有自己独一无二的标签区分，通过方法setActionCommand实现。对按钮事件的处理，则通过actionPerformed方法。

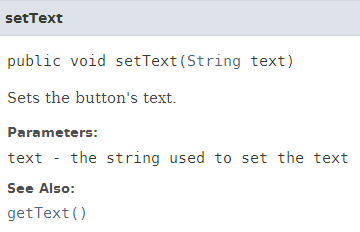
小目标：以MyFrame中的按钮为例，假设要实现如下的功能，每点击一次按钮，按钮上显示当前的点击次数。

按钮上的文字要求能够更新，这种需求不是现在才有，以前的程序员也有，因此可确定Java肯定会提供对应的API。打开Java API文档，定位到JButton类后，可在Method Summary下面看到8个方法，但没有一个符合我们的需求。再往下看，可发现JButton继承大量来自javax.swing.AbstractButton，javax.swing.JComponent，java.awt.Container和java.awt.Component的方法。即使只是从字面意思去揣测哪个是目标方法，面对这么多方法，人也有点不知所措。



此时应该停下来想一想，目标是在按钮上设置文字，自然和关键词text有关（为什么不是String？这依赖于经验，一般是text优先；如果text不行，再考虑String也不迟。），因此可以基于关键词text进行查找。 AbstractButton中有大量的Text相关的方法。因为是要设置按钮文字，所以凡是“get”前缀的方法，可以排除。

对比剩下的方法，setText的可能性最大。点击查看setText方法的说明，可确定就是我们需要的方法。

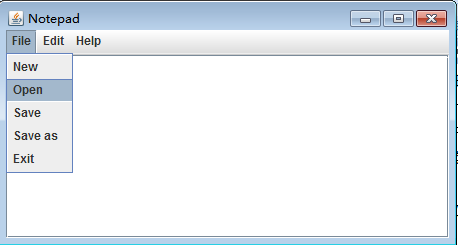


另外要注意的一点，actionPerformed方法由谁来实现是个很重要的问题。根据面向对象设计中“自己的事情自己负责”原则，每个按钮应该有权利指定：发生在自身的点击事件由谁来处理。具体则由方法addActionListener实现（作者怎么知道要用该方法？因为作者看过示例代码，反推得出这样的结论呀!!），查阅该方法，发现一个输入参数是一个接口ActionListener，其中有唯一的方法：void actionPerformed(ActionEvent e)。只要有一个实现ActionListener接口以及其中的actionPerfromed方法的类，再将该类与按钮绑定，就可以让发生在按钮上的事件由该类中的actionPerfromed方法来实现。

请读者参考官方Java API文档或者查找书籍，实现小目标；如果还不明白上述的文字，请查看官方文档中的示例代码。小目标的示例代码见文件夹MyFrame2。其他组件的事件处理与按钮的处理类似，应用时请参照官方文档或者其他书籍即可。学以致用，我们来看两个UI程序界面的案例。

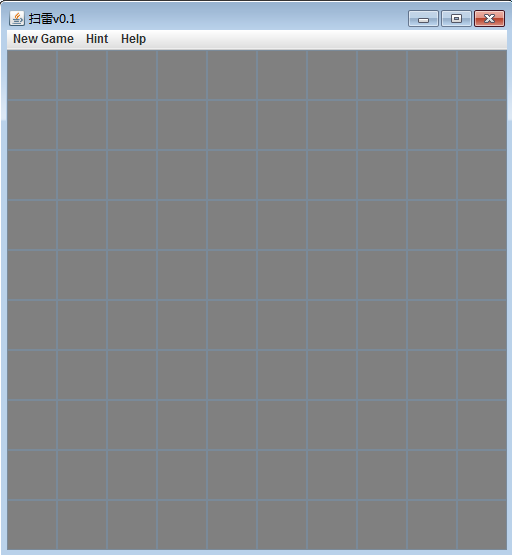
案例1:

第一个是简单的记事本界面，具体代码见Notepad文件夹，使用到的组件有JMenuBar，JMenu，JMenuItem，JScrolPane，JText等。

****

案例2:

如下的扫雷游戏的界面，主要是JButton的二维排列，并且每个按钮点击后改变颜色，具体代码见MineView文件夹。



Java图形界面的设计开发介绍到此，主要的方法是参考示例代码和查阅官方API文档。实际的开发过程中，采用可视化的开发工具，比如windowbuilder，将界面的开发过程转化为画图的过程，提高开发速率。