

GUI(图形用户界面)

GUI

Graphical User Interface(图形用户接口)。

用图形的方式，来显示计算机操作的界面，这样更方便更直观。

CLI

Command line User Interface (命令行用户接口)

就是常见的 Dos 命令行操作。

需要记忆一些常用的命令，操作不直观。

举例：

比如：创建文件夹，或者删除文件夹等

Java 为 GUI 提供的对象都存在 java.Awt 和 javax.Swing 两个包中。

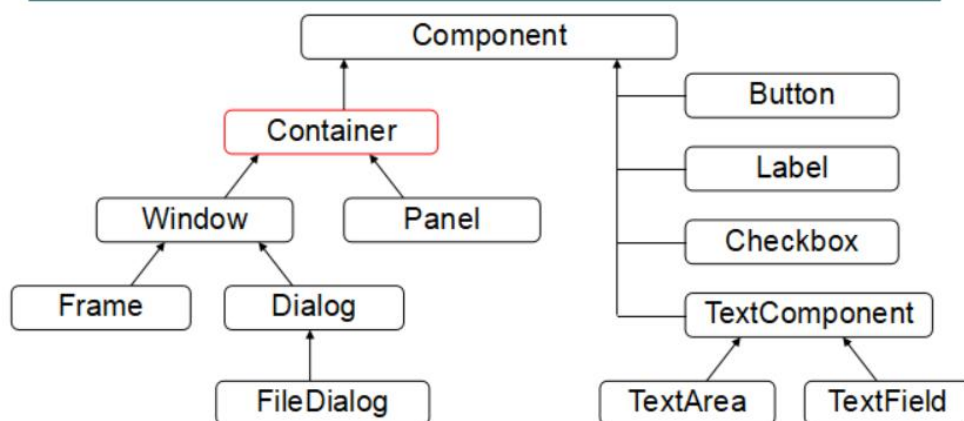
Awt 与 Swing

java.Awt: Abstract Window ToolKit (抽象窗口工具包)，需要调用本地系统方法实现功能。属重量级控件。

javax.Swing: 在 AWT 的基础上，建立的一套图形界面系统，其中提供了更多的组件，而且完全由 Java 实现。增强了移植性，属轻量级控件。

传智播客 — 高级软件人才实操培训专家!
www.itcast.cn

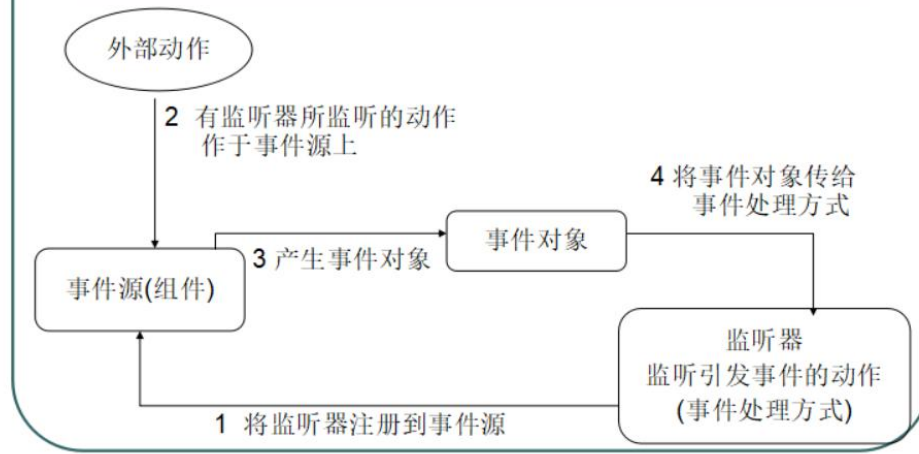
继承关系图



Container: 为容器，是一个特殊的组件，该组件中可以通过add方法添加其他组件进来。

北京传智播客教育 www.itcast.cn

事件监听机制流程图



/*

创建图形化界面：

- 1，创建 frame 窗体。
- 2，对窗体进行基本设置。
比如大小，位置，布局。
- 3，定义组件。
- 4，将组件通过窗体的 add 方法添加到窗体中。
- 5，让窗体显示，通过 setVisible(true)

事件监听机制的特点：

- 1，事件源。
- 2，事件。
- 3，监听器。
- 4，事件处理。

事件源：就是 awt 包或者 swing 包中的那些图形界面组件。

事件：每一个事件源都有自己特有的对应事件和共性事件。

监听器：将可以触发某一个事件的动作（不只一个动作）都已经封装到了监听器中。

以上三者，在 java 中都已经定义好了。

直接获取其对象来用就可以了。

我们要做的事情是，就是对产生的动作进行处理。

*/

```
f.addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        System.out.println("我关");
        System.exit(0);
    }
    public void windowActivated(WindowEvent e)
    {
        System.out.println("我活了。");
    }
    public void windowOpened(WindowEvent e)
    {
        System.out.println("我被打开了,hahahhahah");
    }
});
```

/*

按钮就是事件源。

那么选择哪个监听器呢？

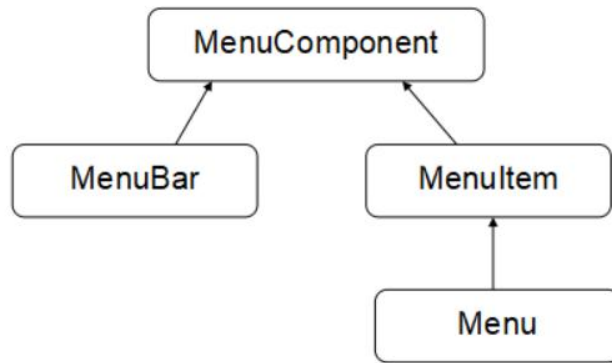
通过关闭窗体示例了解到，想要知道哪个组件具备什么样的特有监听器。

需要查看该组件对象的功能。

通过查阅 **button** 的描述。发现按钮支持一个特有监听 **addActionListener**。

*/

菜单继承体系



/*

如何制作可以双击执行的 jar 包呢？

1，将多个类封装到了一个包(package)中。

2，定义一个 jar 包的配置信息。

定义一个文件 a.txt 。文件内容内容为：

Main-Class:(空格)包名.类名(回车)

3，打 jar 包。

jar -cvfm my.jar a.txt 包名

4，通过 winrar 程序进行验证，查看该 jar 的配置文件中是否有自定义的配置信息。

5，通过工具--文件夹选项--文件类型--jar 类型文件，通过高级，定义该 jar 类型文件的打开动作的关联程序。

jdk\bin\javaw.exe -jar

6，双击试试！。哦了。

*/