如何构建 GUI 程序

贡献者： Saleem Gul 及 Tomas Pavek，

维护者：Ruth Kusterer 及 Patrick Keegan
2008 年 2 月

翻译： flying_space （NetBeans 中文社区成员）

这篇入门教程将教会您怎样创建一个简单的人机交互界面以及向其中添加简单的后台功能。 特别地，我们将向您展示如何按 Swing 规范编写控制按钮和域代码。

我们将会使用到布局管理、设计简单 GUI 界面以及添加一些“按钮(button)”和“文本字段(text field)”组件。 “文本字段”是用来接收输入和显示输出的，“按钮”在前端用来起动相应功能。 我们将创建的应用程序会是一个简单但实用的计算器。

获得更多GUI设计功能、视频及文档，请参见Desgning a Swing GUI in NetBeans IDE.

预计持续时间：15分钟

目录

本教程所需软件

请确保您的计算机已安装如下软件：

软件或者资源	需求版本
NetBeans IDE	版本 6.1 或者 版本 6.0
Java 开发工具包 (JDK)	版本 6 或者 版本 5

步骤1：创建项目

第一步，创建一个应用程序，并将其命名为 NumberAddition。

1. 选择“文件” -> “新建项目”。或者在“工具栏”单击“新建项目”图标。
2. 在弹出窗口的“类别”窗格中选择“Java”，在“项目”窗格中选择“Java应用程序”。单击“下一步”；
3. 在“项目名称”中键入 NumberAddition，在“项目位置”中键入本地文件目录以保存项目；
4. 确认复选框“设置为主项目”已被勾选。并确保复选框“创建主类”未被勾选。
5. 点击“完成”

步骤2：构建界面

继续创建我们的界面。我们需要一个 Java 容器来放置其他将被调用的 GUI 组件。 在此步骤中我们使用 JFrame 组件作为所需的容器。我们将该容器放置在一个新建包中， 该新建包位于“源包”中。

创建 JFrame 容器

1. 在“项目”窗口右击 NumberAddition，选择“新建” -> “JFrame 窗体”。

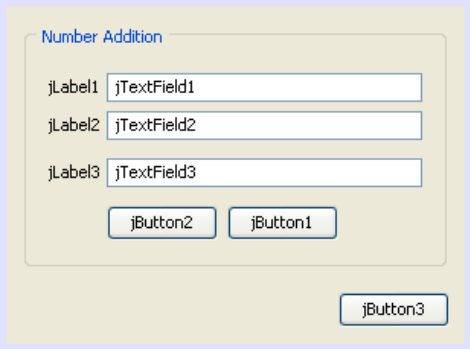
2. 在“类名”项键入 NumberAdditionUI。
3. 在“包”项键入 `my.numberaddition`。
4. 单击“完成”。

IDE 根据类 `NumberAdditionUI` 在 `NumberAddition` 应用程序中创建了 `NumberAdditionUI` 窗体，并在 GUI Builder 中打开 `NumberAdditionUI` 窗体。包 `my.NumberAddition` 被设置为默认包。

添加组件：构建界面

下一步我们将通过“组件面板”为界面获得一个 JPanel 组件。 而后将向其中添加三个 JLabels 组件，三个 JTextFields 组件，三个 JButtons 组件。 如果您从未使用过 GUI Builder，应该先通过 GUI Building in NetBeans IDE 教程获得相关信息。

一旦您通过拖拽的方式添加了上述组件，JFrame 应该显示如以下截图：



如果在您在 IDE 的右上角没有看到“组件面板”，请选择“窗口” -> “组件面板”调出。

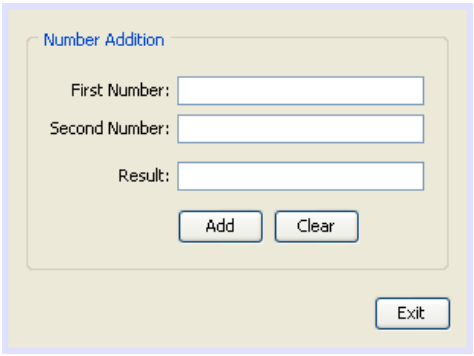
1. 首先在“组件面板”里选中 JPanel 组件并拖入 JFrame 中。
2. 当 JPanel 组件高亮时，在其“属性”窗口点击位于 `Border` 项后的省略号(...)按钮来选择组件风格。
3. 在 Border 对话框的列表里选择 TitleBorder 风格，并在“标题”项填入 `Number Addition`。单击“确定”按钮保存并退出。
4. 您现在将看到一个标有 Number Addition 的空的 JFrame 组件，如上述截图。请依据以上截图添加三个 JLabels，三个 JTextFields 及三个 JButtons。

重命名组件

在此步骤中我们将重命名刚添加进 JFrame 的组件的显示文本

1. 双击 `jLabel1` 并更改其文本内容为 `First Number`
2. 双击 `jLabel2` 并更改其内容为 `Second Number`
3. 双击 `jLabel3` 并更改其内容为 `Result`
4. 双击 `jTextField1`，删除其示例文本。您需要重新调整 `jTextField1` 的初始大小。同样设置 `jTextField2` 及 `jTextField3`。
5. 双击 `jButton1`，重命名为 `Clear`.
6. 双击 `jButton2`，重命名为 `Add`.
7. 双击 `jButton3`，重命名为 `Exit`.

实现 GUI 界面将如以下截图：



步骤3：添加功能

在此步骤中，我们将为 Add, Clear 及 Exit 按钮赋上相应功能。而 jTextField1 及 jTextField2 将用于用户输入，jTextField3 则用于程序输出－我们将创建一个非常简单的加法计算器。让我们开始吧。

让 Exit 按钮工作

为了将功能赋于按钮组件，我们必须分配一个事件处理器给每个需响应的事件。 这样我们可以知道用户是否“按下”了按钮，无伦是通过鼠标还是键盘操作。 因此，我们将使用事件监听器(ActionListener)来处理响应事件(ActionEvent)。

1. 右击 Exit 按钮。在弹出菜单中选择“事件” -> “Action” -> “actionPerformed”。 注意菜单中包含了许多您可以使用的事件处理！当您选择了 actionPerformed 事件处理，IDE 便会自动添加事件监听器(ActionListener)到 Exit 按钮中的并在监听器的 actionPerformed 方法内产生一个处理方法。
2. IDE 会转到“源”窗口并将光标调整到您欲添加功能（无伦是鼠标还是键盘操作按下按钮时）的方法内。如下所示：

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    //TODO: Add your handling code here:
                }
```

3. 现在我们将添加 Exit 按钮现实的代码。您需键入 System.exit(0); 来覆盖 TODO 所在行。如下代码：

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
                }
```

让 Clear 按钮工作

1. 单击位于工作区顶部的“设计”标签返回“设计”界面。
2. 右击 Clear 按钮(jButton1)。在下拉菜单中选择“事件” -> “Action” -> “actionPerformed”。
3. 我们需要 Clear 按钮来清空全部 jTextFields 上的文本内容。 接下来向上一步一样添加代码。完成的代码如下：

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt){
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
                }
```

上述代码改变了三个 JTextFields 的文本内容，将其中的文本置空。

让 Add 按钮工作

Add 按钮将实现三个功能。

1. 接收 jTextField1 及 jTextField2 中的用户输入并将输入的字符串转换为浮点型数据。
2. 将上述输入的两个数相加得到结果。

3. 将结果覆盖 jTextField3 中的文本以输出结果。

让我们开始吧！

1. 点击工作区上方的“设计”标签返回设计窗格。
2. 右击 Add 按钮(jButton3)。在弹出菜单中选择“事件” -> “Action” -> “actionPerformed”
3. 我们将添加一些代码使得 Add 按钮工作。代码如下:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt){
    // First we define float variables.
    float num1, num2, result;
    // We have to parse the text to a type float.
    num1 = Float.parseFloat(jTextField1.getText());
    num2 = Float.parseFloat(jTextField2.getText());
    // Now we can perform the addition.
    result = num1+num2;
    // We will now pass the value of result to jTextField3.
    // At the same time, we are going to
    // change the value of result from a float to a string.
    jTextField3.setText(String.valueOf(result));
                }
```

我们的项目已完成，现在可以生成并运行以查看其功能。

步骤4：运行项目

最后一步，生成及运行该项目。

1. 选择“生成” -> “生成主项目”。
2. 当“输出”窗格显示“成功生成”时， 选择“运行” -> “运行主项目”。
3. 如果您被提示项目 NumberAddition 没有设置主类时， 您应该选择 my.NumberAddition.NumberAdditionUI 作为主类， 并点击“确定”按钮。
4. 您所创建的项目现在已成功运行了。

在这篇教程中您学会了怎样通过 NetBeans GUI Builder 来关联各 GUI 组件。

事件处理的工作原理

这篇教程展示了如何响应一个简单的按钮事件。当然还有更多的事件可让您的应用程序来响应。 IDE 能帮您以列表的形式让您方便的找到您的 GUI 组件可实现的事件处理：

1. 让我们返回到文件 NumberAdditionUI.java 编辑器。点击“设计”标签来回看 GUI Builder 中的 GUI 版面。
2. 右击任一 GUI 组件，选择弹出菜单中的“事件”。现在，浏览菜单并了解都有些什么功能，您不需要选择任何选项。
3. 或者，您可选择“窗口”菜单中的“属性”，在“属性”窗格中点击“事件”标签。 在“事件”标签中，您能预览及编辑事件处理器来关联当前的活动组件。
4. 您能使您的程序响应如回车，单双击，三次点击，鼠标活动，窗口大小及聚焦改变等操作。 通过“事件”菜单您能自动地生成相应的事件处理器。将有更多的事件处理会被您使用。 (参阅 best practices for Event handling 来自 Sun 网页 Java Events Tutorial.)

1.

那么，事件处理是怎样工作的呢？每次当您从“事件”菜单选择事件时， IDE 都自动帮您生成一个所谓的事件监听器，并关联您的组件。浏览以下几步来了解事件处理的工作原理。

1. 返回文件 NumberAdditionUI.java 编辑器，点击“源”标签来查看 GUI 源代码。
2. 滚动代码并留意方法 jButton1ActionPerformed(), jButton2ActionPerformed() 及 jButton3ActionPerformed()。这些方法便是刚才实现的，称为“事件处理器”(event handlers)。
3. 现在将代码滚动到 initComponents() 方法。如果您看不到这个方法，那么请查找一行标为 Generated Code 的

代码，点击此行前方的“+”号来展开 `initComponents()` 方法。

4. 首先，注意到蓝色的底色围绕着 `initComponents()` 方法。这表明代码是 IDE 自动生成并且不允许再编辑的。
5. 现在，浏览方法 `initComponents()`。在这些代码中，包含了用于初始化并设置您 GUI 组件位置的代码。这些代码是您在“设计”模式下设置和编辑组件时自动生成和配置的。
6. 在 `initComponents()` 中查找如下代码

```
jButton3.setText("Exit");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
           jButton3ActionPerformed(evt);
    }
            });
```

这就是 GUI 组件添加事件监听器的地方，在此您为 `jButton3` 注册了一个事件监听器(ActionListener)。 而在这个 ActionListener 接口中的方法 actionPerformed 通过简单调用您之前设置的事件处理器 `jButton3ActionPerformed` 来处理响应事件。现在这个按钮便能监听活动事件了。当一个事件产生时，系统便通知监听器来执行您事件处理器中的代码来响应事件。

一般而言，要使 GUI 组件实现事件响应需要对组件注册监听器且实现事件处理。如您所见，NetBeans IDE 能帮您自动关联事件监听，因此您只需专注于各事件间的逻辑关系和内部联系而忽略实现细节。

反馈

相关参考

- Designing a Swing GUI in NetBeans IDE
- GUI Builder - Frequently Asked Questions
- Best practices for Event handling 来自 Sun's Java Events Tutorial.
- Building a Java Desktop Database Application.
- 示例代码参阅 Java Almanac: Java Events

如果您有任何疑问或者需要帮助，且希望实时获得的 NetBeans IDE 的最新消息， 请加入 nbusers@netbeans.org 邮件列表。您可以通过发送空邮件到 nbusers-digest-subscribe@netbeans.org 以加入该邮件列表。