



Introdução à Construção de GUIs

PRINTABLE VERSION

Contribuição de Saleem Gul e Tomas Pavek

Este tutorial de iniciante ensina como criar uma interface de usuário gráfica simples e adicionar funcionalidade de back-end simples. Em particular, mostraremos como codificar o comportamento dos botões e dos campos em um form de Swing.

Nós iremos trabalhar no layout e design de uma GUI e adicionar alguns botões e campos de texto. Os campos de texto serão usados para receber entrada do usuário e também para exibir a saída do programa. O botão iniciará a funcionalidade interna do front-end. A aplicação que criamos será uma calculadora simples mas funcional.

Para uma guia mais abrangente das funcionalidades de design do GUI Builder, incluindo demonstrações em vídeo das várias funcionalidades de design, consulte [Projetando uma GUI Swing no NetBeans IDE](#).

Duração esperada: 15 minutos

Conteúdo

- [Exercício 1: Criando um Projeto](#)
- [Exercício 2: Construindo o Front End](#)
- [Exercício 3: Adicionando Funcionalidade](#)
- [Exercício 4: Executando o Programa](#)
- [Como o Tratamento de Eventos Funciona](#)
- [Consulte Também](#)

Para concluir este tutorial, você precisa dos seguintes recursos e softwares.

Software ou Recurso	Versão Necessária
NetBeans IDE com Java SE	versão 6.9 ou posterior
JDK (Java Development Kit)	versão 6 ou 7



Exercício 1: Criando um Projeto

A primeira etapa é criar um projeto do IDE para a aplicação que vamos desenvolver. Chamaremos nosso projeto de `NumberAddition`.

1. Escolha **Arquivo > Novo Projeto**. Como alternativa, você pode clicar no ícone **Novo Projeto** na barra de ferramentas do IDE.
2. No painel **Categorias**, selecione o nó **Java**. No painel **Projetos**, escolha **Aplicação Java**. Clique em **Próximo**.
3. Digite `NumberAddition` no campo **Nome do Projeto** e especifique um caminho; por exemplo: em seu diretório home, como a localização do projeto.
4. (Opcional) Marque a caixa de seleção **Utilizar Pasta Dedicada para Armazenamento de Bibliotecas** e especifique o local da pasta de bibliotecas. Consulte [Compartilhando Bibliotecas do Projeto](#) para obter mais informações sobre esta opção.
5. Desmarque a caixa de seleção **Criar Classe Principal** caso esteja selecionada.
6. Clique em **Finalizar**.

Exercício 2: Construindo o Front End

Para prosseguir com a construção da nossa interface, precisamos criar um contêiner Java no qual colocaremos os outros componentes requeridos da GUI. Nesta etapa criaremos um contêiner usando o componente `JFrame`. Colocaremos o contêiner em um novo pacote, que aparecerá no nó **Pacotes de Códigos-fonte**.

Crie um contêiner JFrame

1. Na janela **Projetos**, clique com o botão direito do mouse no nó `NumberAddition` e selecione **Novo > Outro**.
2. Na caixa de diálogo **Novo Arquivo**, escolha a categoria **Swing GUI Forms** e o tipo de arquivo **JFrame Form**. Clique em **Próximo**.
3. Insira `NumberAdditionUI` como o nome da classe.
4. Indique `my.numberaddition` como o pacote.
5. Clique em **Finalizar**.

O IDE cria o form `NumberAdditionUI` e a classe `NumberAdditionUI` na aplicação `NumberAddition` e abre o form `NumberAdditionUI` no GUI Builder. O pacote `my.NumberAddition` substitui o pacote default.

Adicionando componentes: Criando o Front End

Em seguida, usaremos a **Paleta** para preencher o front end da nossa aplicação com um `JPanel`. Em seguida, adicionaremos três `JLabels`, três `JTextFields` e três `JButtons`. Se você não usou o GUI Builder antes, pode achar úteis as informações no tutorial [Projetando uma GUI Swing no NetBeans IDE](#) sobre posicionamento de componentes.

Depois de terminar de arrastar e posicionar os componentes mencionados anteriormente, o `JFrame` deve ter uma aparência como o da tela seguinte.

[Download NetBeans IDE](#)

Training

Java Programming
Language



Support

Oracle Development Tools
Support Offering for
NetBeans IDE

Documentation

General Java Development

External Tools and Services

Java GUI Applications

Java EE & Java Web Development

Web Services Applications

NetBeans Platform (RCP) and Module Development

PHP and HTML5 Applications

C/C++ Applications

Mobile Applications

Sample Applications

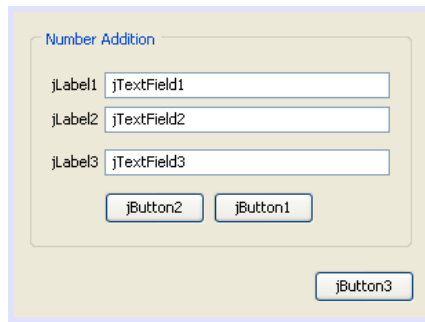
Demos and Screencasts

More

[FAQs](#)

[Contribute Documentation!](#)

[Docs for Earlier Releases](#)



Caso não veja a janela Paleta no canto superior direito do IDE, escolha Janela > Paleta.

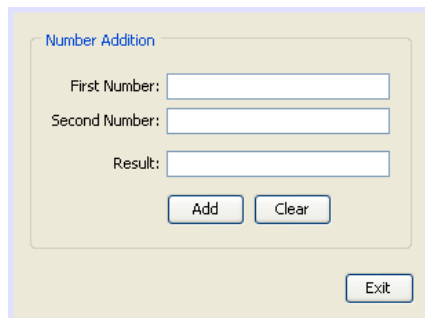
1. Comece selecionando um Pannel da categoria Contêineres Swing na Paleta e solte-o no JFrame.
2. Enquanto o JPanel estiver realçado, vá para a janela Propriedades e clique no botão de reticências (...) ao lado da borda para escolher um estilo de borda.
3. Na caixa de diálogo Borda, selecione TitledBorder na lista, e digite Adição de Número no campo Título. Clique em OK para salvar as alterações de e saia da caixa de diálogo.
4. Você agora deve ver um JFrame intitulado vazio que diz Adição de Número como na tela. Observe a tela e os três JLabels, três JTextFields e três JButtons como você vê acima.

Renomeando os Componentes

Nesta etapa, vamos renomear o texto de exibição dos componentes que acabamos de adicionar ao JFrame.

1. Clique duas vezes em jLabel1 e altere a propriedade do texto para Primeiro Número
2. Clique duas vezes em jLabel2 e altere o texto para Segundo Número
3. Clique duas vezes em jLabel3 e altere o texto para Resultado
4. Delete o texto da amostra de jTextField1. Você pode tornar o texto editável ao clicar com o botão direito do mouse no campo de texto e escolher Editar Texto no menu suspenso. Talvez você tenha que redimensionar o jTextField1 para seu tamanho original. Repita esta etapa para jTextField2 e jTextField3.
5. Renomeie o texto de exibição de jButton1 para Limpar. (Você pode editar o texto de um botão, clicando com o botão direito do mouse no botão e escolhendo Editar Texto. Ou você pode clicar no botão, pausar e clicar novamente.)
6. Renomeie o texto de exibição de jButton2 para Adicionar.
7. Renomeie o texto de exibição de jButton3 para Sair.

Sua GUI finalizada agora deve ser semelhante à tela seguinte:



Exercício 3: Adicionando Funcionalidade

Neste exercício, vamos dar funcionalidade aos botões Adicionar, Limpar e Sair. As caixas jTextField1 e jTextField2 serão usadas para entrada do usuário e jTextField3 para a saída do programa - o que estamos criando é uma calculadora muito simples. Vamos começar.

Fazendo o Botão Sair Funcionar

Para dar função aos botões, temos que designar um handler de eventos a cada um deles para responder aos eventos. Em nosso caso, queremos saber quando o botão é pressionado, por clique do mouse ou por meio do teclado. Portanto, usaremos ActionListener respondendo a ActionEvent.

1. Clique com o botão direito do mouse no botão Sair. No menu suspenso, escolha Eventos > Ação> actionPerformed. Observe que o menu contém muito mais eventos aos quais você pode responder! Quando você selecionar o evento actionPerformed, o IDE adicionará automaticamente um ActionListener ao botão Sair e gerará um método de handler para o método actionPerformed do listener.
2. O IDE abrirá a janela Código-fonte e rolará para onde você implementa a ação que deseja que o botão faça quando for pressionado (pelo clique do mouse ou por meio do teclado). Sua janela Código-fonte deve conter as seguintes linhas:

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

```
}
```

3. Agora vamos adicionar o código da ação que desejamos que o botão Sair execute. Substitua a linha `TODO` por `System.exit(0);`. Seu código do botão Sair finalizado deve ter a seguinte aparência:

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

Fazendo o Botão Limpar Funcionar

1. Clique na guia Design na parte superior da sua área de trabalho para voltar para o Design do Form.
2. Clique com o botão direito do mouse no botão Limpar (`jButton1`). No menu pop-up, escolha Eventos > Ação > `actionPerformed`.
3. Agora faremos o botão Limpar apagar todo o texto dos `JTextFields`. Para fazer isso, adicionaremos algum código como acima. Seu código-fonte finalizado deve ter a seguinte aparência:

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

O código acima quase não altera o texto em todos os nossos três `JTextFields`, basicamente está sobrescrevendo o Texto existente por um espaço em branco.

Fazendo o Botão Adicionar Funcionar

O botão Adicionar realizará três ações.

1. Ele vai aceitar a entrada do usuário de `jTextField1` e `jTextField2` e converter a entrada de um tipo `String` para uma flutuação.
2. Ele irá então executar a adição dos dois números.
3. E, finalmente, ele converterá a soma para um tipo `string` e irá colocá-lo no `TextField3`.

Vamos começar!

1. Clique na guia Design na parte superior da sua área de trabalho para voltar para o Design do Form.
2. Clique com o botão direito do mouse no botão Adicionar (`jButton2`). No menu pop-up, escolha Eventos > Ação > `actionPerformed`.
3. Agora vamos adicionar algum código para fazer nosso botão Adicionar funcionar. O código-fonte finalizado terá a seguinte aparência:

```
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));  
}
```

Agora nosso programa está completo e podemos construí-lo e executá-lo para vê-lo em ação.

Exercício 4: Executando o Programa

Para executar o programa no IDE:

1. Escolha Executar > Executar Projeto Principal (se preferir, pressione F6).
Observação: Se aparecer uma entrada informando que o `Project NumberAddition` não possui uma classe principal definida, então você deve selecionar `my.NumberAddition.NumberAdditionUI` como a classe principal na mesma janela e clicar no botão OK.

Para executar o programa fora do IDE:

1. Escolha Executar > Limpar e Construir Projeto Principal (Shift-F11) para construir o arquivo JAR da aplicação.
2. Usando o explorador de arquivos ou o gerenciador de arquivos de seu sistema, navegue até o diretório `NumberAddition/dist`.
Observação: a localização do diretório do projeto `NumberAddition` depende o caminho que você especificou ao criar o projeto na etapa 3 da seção [Exercício 1: Criando um Projeto](#).

3. Clique duas vezes no arquivo `NumberAddition.jar`.

Depois de alguns segundos, a aplicação deve ser iniciada.

Observação: Se clicar duas vezes no arquivo JAR e a aplicação não for iniciada, consulte [este artigo](#) para obter informações sobre a definição de associações do arquivo JAR em seu sistema operacional.

Você também pode iniciar a aplicação da linha de comandos.

Para iniciar a aplicação partir da linha de comandos:

1. em seu sistema, abra um prompt de comando ou uma janela de terminal.
2. No prompt de comando, altere os diretórios para o diretório `NumberAddition/dist`.
3. Na linha de comandos, digite a seguinte instrução:

```
java -jar NumberAddition.jar
```

Observação: Certifique-se de `my.numberaddition.numberadditionui` é definido como a classe principal antes de executar a aplicação. É possível verificar isso clicando com o botão direito no nó do projeto `numberaddition` no painel projetos, escolhendo Propriedades no menu pop-up, e selecionando Executar categoria na caixa de diálogo Propriedades do Projeto. O campo Classe Principal deve exibir `my.numberaddition.numberadditionui`.

Como o Tratamento de Eventos Funciona

Este tutorial mostrou como responder a um simples evento de botão. Existe muito mais eventos aos quais você pode fazer sua aplicação responder. O IDE pode ajudá-lo a encontrar a lista de eventos disponíveis que os componentes de GUI podem manipular:

1. Volte para o arquivo `NumberAdditionUI.java` no Editor. Clique na guia Design para ver o layout da GUI no GUI Builder.
2. Clique com o botão direito do mouse em qualquer componente de GUI, e selecione Eventos no menu pop-up. Por enquanto, apenas explore o menu para ver o que há nele, você não precisa selecionar nada.
3. Como alternativa, você pode selecionar Propriedades no menu Janela. Na janela Propriedades, clique na guia Eventos. Na guia Eventos, você pode exibir e editar os handlers de eventos associados ao componente de GUI ativos no momento.
4. Você pode fazer sua aplicação responder a pressionamentos de tecla, cliques únicos, duplos e triplos do mouse, movimentação do mouse, alterações do tamanho e do foco da janela. Você pode gerar handlers de eventos para todos eles no menu Eventos. O evento mais comum usará um evento Ação. (Aprenda [melhores práticas para Tratamento de eventos](#) no [Tutorial de Eventos Java](#) da Sun.)

Como o tratamento de eventos funciona? Cada vez que você seleciona um evento no menu Evento, o IDE automaticamente cria um listener de evento por você e o incorpora em seu componente. Percorra as etapas seguintes para ver como o tratamento de eventos funciona.

1. Volte para o arquivo `NumberAdditionUI.java` no Editor. Clique na guia Código-fonte para ver o código-fonte da GUI.
2. Role para baixo e observe os métodos `jButton1ActionPerformed()`, `jButton2ActionPerformed()` e `jButton3ActionPerformed()` que acabou de implementar. Esses métodos são chamados de handlers de eventos.
3. Agora role para um método chamado `initComponents()`. Se você não vir esse método, procure por uma linha que diz `Generated Code`; clique no sinal + ao lado dela para expandir o método `initComponents()` contraído.
4. Primeiro, observe o bloco azul ao lado do método `initComponents()`. Esse código foi gerado automaticamente pelo IDE e você não pode editá-lo.
5. Agora, explore o método `initComponents()`. Entre outras coisas, ele contém o código que inicializa e coloca seus componentes de GUI no form. Esse código é gerado e atualizado automaticamente enquanto você coloca e edita componentes na view Design.
6. Em `initComponents()`, role para baixo onde diz

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

Esse é o local em que um objeto de listener de evento é adicionado ao componente de GUI; nesse caso, você registra um `ActionListener` ao `jButton3`. A interface `ActionListener` possui um método `actionPerformed` usando o objeto `ActionEvent` que é implementado simplesmente pela chamada do handler de eventos `jButton3ActionPerformed`. O botão agora está fazendo listening dos eventos de ação. Toda vez que é pressionado, um `ActionEvent` é gerado e passado para o método `actionPerformed` do listener que por sua vez executa o código que você forneceu no handler de eventos desse evento.

Generalizando, para poder responder, cada componente de GUI interativo precisa registrar um listener de evento e precisa implementar um handler de eventos. Como você pode ver, o NetBeans IDE manipula a anexação do listener de evento; portanto, você pode se concentrar na implementação da lógica comercial real que deve ser adicionada pelo evento.

[Enviar Feedback neste Tutorial](#)

Consulte Também

- [Implementando GUIs de Java](#) em *Desenvolvendo Aplicações com o NetBeans IDE*

- [Encapsulando e Distribuindo Aplicações Java Desktop](#)
- [Projetando uma GUI Swing no NetBeans IDE](#)
- [Tratando imagens em uma Aplicação da GUI](#)
- [Usando Hibernate em uma Aplicação Java Swing](#)
- [Trilha de Aprendizado das Aplicações de GUI do Java](#)
- [Trilha de Aprendizado do Desenvolvimento Geral em Java](#)
- [GUI Builder - Perguntas Frequentes](#)
- [Lição: Gravando Listeners de Evento](#) do [The Java\(tm\) Tutorials](#) .
- As Amostras de Código podem ser encontradas em [Java Almanac: Java Events](#)

[SiteMap](#) [About Us](#) [Contact](#) [Legal & Licences](#)



By use of this website, you agree to the [NetBeans Policies](#) and [Terms of Use](#). © 2013, Oracle Corporation and/or its affiliates. Sponsored by **ORACLE**