

Introdução à construção de GUIs

Contribuição de Saleem Gul e Tomas Pavek, mantido por Ruth Kusterer e Patrick Keegan

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

Trabalharemos no layout e no design de uma GUI e adicionaremos alguns botões e campo 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. O aplicativo que criamos será uma calculadora simples mas funcional.

Para um guia mais abrangente dos recursos de desenho do Construtor de GUIs, incluindo demonstrações em vídeo dos vários recursos de desenho, consulte [Criando 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 funciona o manipulador de eventos](#)
- [Consulte também](#)



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

Software ou recurso	Versão necessária
NetBeans IDE	versão 6.5
Java Development Kit (JDK)	versão 6 ou versão 5

Exercício 1: Criando um projeto

A primeira etapa é criar um projeto do IDE para o aplicativo 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 Aplicativo Java. Clique em Próximo.
3. Digite `NumberAddition` no campo Nome do projeto e especifique um caminho, ou seja, em seu diretório base como a localização do projeto.
4. (Opcional) Selecione a caixa de verificação Usar 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. Verifique se a caixa de verificação Definir como projeto principal está selecionada.
6. Desmarque a caixa de verificação Criar classe principal caso esteja selecionada.
7. Clique em Terminar.

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

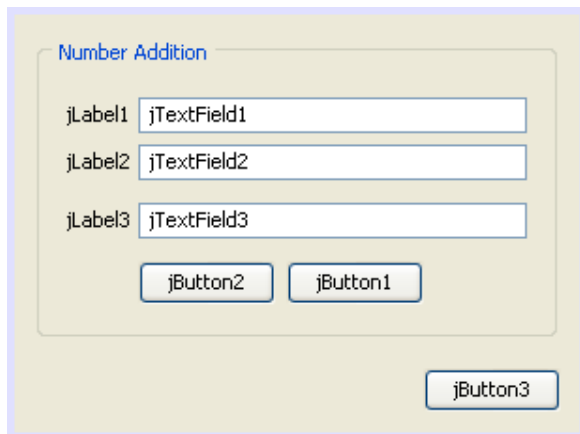
1. Na janela Projetos, clique com o botão direito do mouse no nó `NumberAddition` e selecione Novo > Formulário JFrame.
2. Indique `NumberAdditionUI` como o nome da classe.
3. Indique `my.numberaddition` como o pacote.
4. Clique em Terminar.

O IDE cria o formulário `NumberAdditionUI` e a classe `NumberAdditionUI` no aplicativo `NumberAddition`, e abre o formulário `NumberAdditionUI` no construtor de GUIs. O pacote `my.NumberAddition` substitui o pacote padrão.

Adicionando componentes: Criando o front end

Em seguida, usaremos a paleta para preencher o front end do nosso aplicativo com um `JPanel`. Em seguida, adicionaremos três `JLabels`, três `JTextField`s e três `JButtons`. Se você não usou o Construtor de GUIs antes, pode achar úteis as informações no tutorial [Desenhando 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 a da captura de tela seguinte.



Se você não vir a janela Paleta no canto direito superior do IDE, escolha Janelas > Paleta.

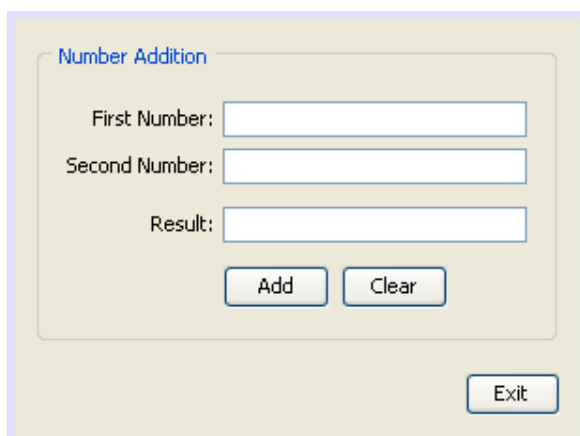
1. Comece selecionando um JPanel na paleta e solte-o no JFrame.
2. Enquanto o JPanel estiver realçado, vá para a janela Propriedades e clique no botão de reticências (...) ao lado da borda para escolher um estilo de borda.
3. Na caixa de diálogo Borda, selecione TitledBorder na lista, e digite Adição 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 captura de tela. Observe a captura de 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. Exclua o texto da amostra de jTextField1. Você pode tornar o texto de exibição editável, clicando no campo de texto, pausando e clicando no campo de texto novamente. 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 captura de 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 atribuir um manipulador 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 via teclado. Portanto, usaremos ActionListener respondendo a ActionEvent.

1. Clique com o botão direito do mouse no botão Sair. No menu pop-up, escolha Eventos > Ação > ActionPerformed. Observe que o menu

contém muito mais eventos aos quais você pode responder! Quando você seleciona o evento `actionPerformed`, o IDE adiciona automaticamente um `ActionListener` ao botão Sair e gera um método de manipulador para manipular o método `actionPerformed` do ouvinte.

2. O IDE abre a janela Código-fonte e rola para onde você implementa a ação que deseja que o botão faça quando é pressionado (pelo clique do mouse ou via 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 do que desejamos que o botão Sair faça. Você deve digitar `System.exit(0)`; no código acima, substituindo a linha `TODO`. Seu código do botão Sair concluído deve ter a seguinte aparência:

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

Fazendo o botão Limpar funcionar

1. Clique na aba Design na parte superior da sua área de trabalho para voltar para o design do formulário
2. Clique com o botão direito do mouse no botão Limpar (`jButton1`). No menu pop-up, selecione `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 altera o texto em todos os nossos três `JTextFields` para nada, na essência, ele 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. Em seguida, ele fará a adição dos dois números e, finalmente,
3. converterá a soma para um tipo `String` e o colocará em `jTextField3`.

Vamos começar!

1. Clique na aba Design na parte superior da sua área de trabalho para voltar para o design do formulário.
2. Clique com o botão direito do mouse no botão Adicionar (`jButton2`). No menu pop-up, selecione `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`
2. 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` do aplicativo.
2. Usando o explorador de arquivos ou o gerenciador de arquivos de seu sistema, navegue até o diretório `NumberAddition/dist`.

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

Depois de alguns segundos, o aplicativo deve ser iniciado.

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

Você também pode iniciar o aplicativo a partir da linha de comando.

Para iniciar o aplicativo a partir da linha de comando:

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 comando, insira a seguinte declaração:

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

Como a manipulação de eventos funciona

Este tutorial mostrou como responder a um simples evento de botão. Existe muito mais eventos aos quais você pode fazer seu aplicativo 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 aba Design para ver o layout da GUI no Construtor de GUIs.
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 aba Eventos. Na aba Eventos, você pode ver e editar os manipuladores de eventos associados ao componente de GUI ativos no momento.
4. Você pode fazer seu aplicativo 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 manipuladores de eventos para todos eles no menu Eventos. O evento mais comum usará um evento Ação. (Aprenda [melhores práticas para manipulação de eventos](#) no [Tutorial de eventos Java](#) da Sun.)

Como a manipulação de eventos funciona? Toda vez que você seleciona um evento no menu Evento, o IDE cria automaticamente um ouvinte de evento para você e o adiciona ao seu componente. Percorra as etapas seguintes para ver como a manipulação de eventos funciona.

1. Volte para o arquivo `NumberAdditionUI.java` no editor. Clique na aba 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 manipuladores 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()` recolhido.
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 formulário. Esse código é gerado e atualizado automaticamente enquanto você coloca e edita componentes na visualização 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 ouvinte 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 manipulador de eventos `jButton3ActionPerformed`. O botão agora está ouvindo eventos de ação. Toda vez que é pressionado, um `ActionEvent` é gerado e passado para o método `actionPerformed` do ouvinte que por sua vez executa o código que você forneceu no manipulador de eventos desse evento.

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

[Envie-nos seus comentários](#)

Veja também

- [Criando um GUI Swing no NetBeans IDE](#)
- [Criando um aplicativo de banco de dados de área de trabalho Java.](#)
- [Usando o Hibernar em um aplicativo Java Swing.](#)
- [Trilha do aprendizado das aplicações de GUI em Java](#)
- [Trilha de aprendizado da programação em Java e IDE básica.](#)
- [Construtor de GUIs - Perguntas frequentes](#)

- [Melhores práticas da manipulação de eventos](#) do [Tutorial de eventos Java](#) da Sun.
- As amostras de código podem ser encontradas em [Java Almanac: Java Events](#)

Se você tem perguntas ou precisa de suporte, e deseja manter-se informado sobre os desenvolvimentos mais recentes sobre os recursos do NetBeans IDE, inscreva-se na lista de usuários nbusers@netbeans.org, enviando uma mensagem vazia para nbusers-digest-subscribe@netbeans.org.