

Biblioteca Java Swing



Prof. Dr. João Paulo Lemos Escola
Copyright© 2025

Tópicos da aula

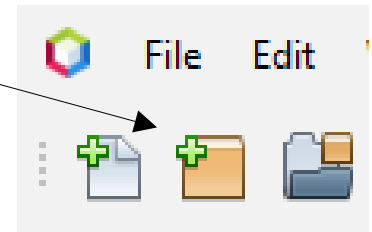
- Classes de componentes gráficos;
- Desenvolvimento de UI no NetBeans;
- Conceitos introdutórios sobre orientação a objetos.
- Eventos em componentes;

Introdução

- Swing é a biblioteca gráfica padrão da linguagem Java para criação de aplicações gráficas Desktop;
- A biblioteca está organizada dentro do pacote **javax.swing**;
 - Pacote é uma **pasta**, serve para organizar os arquivos de um projeto.
- Nesta aula vamos aprender seus principais componentes e começar o desenvolvimento de aplicações para PC.

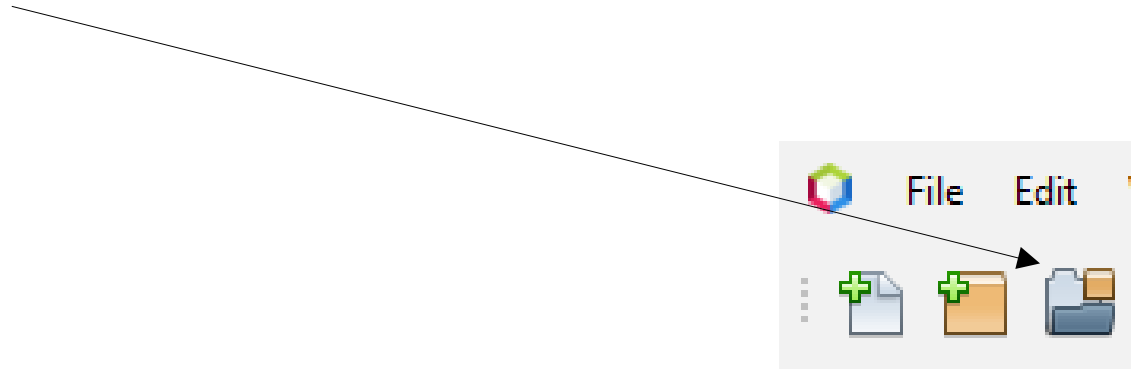
Criando um projeto

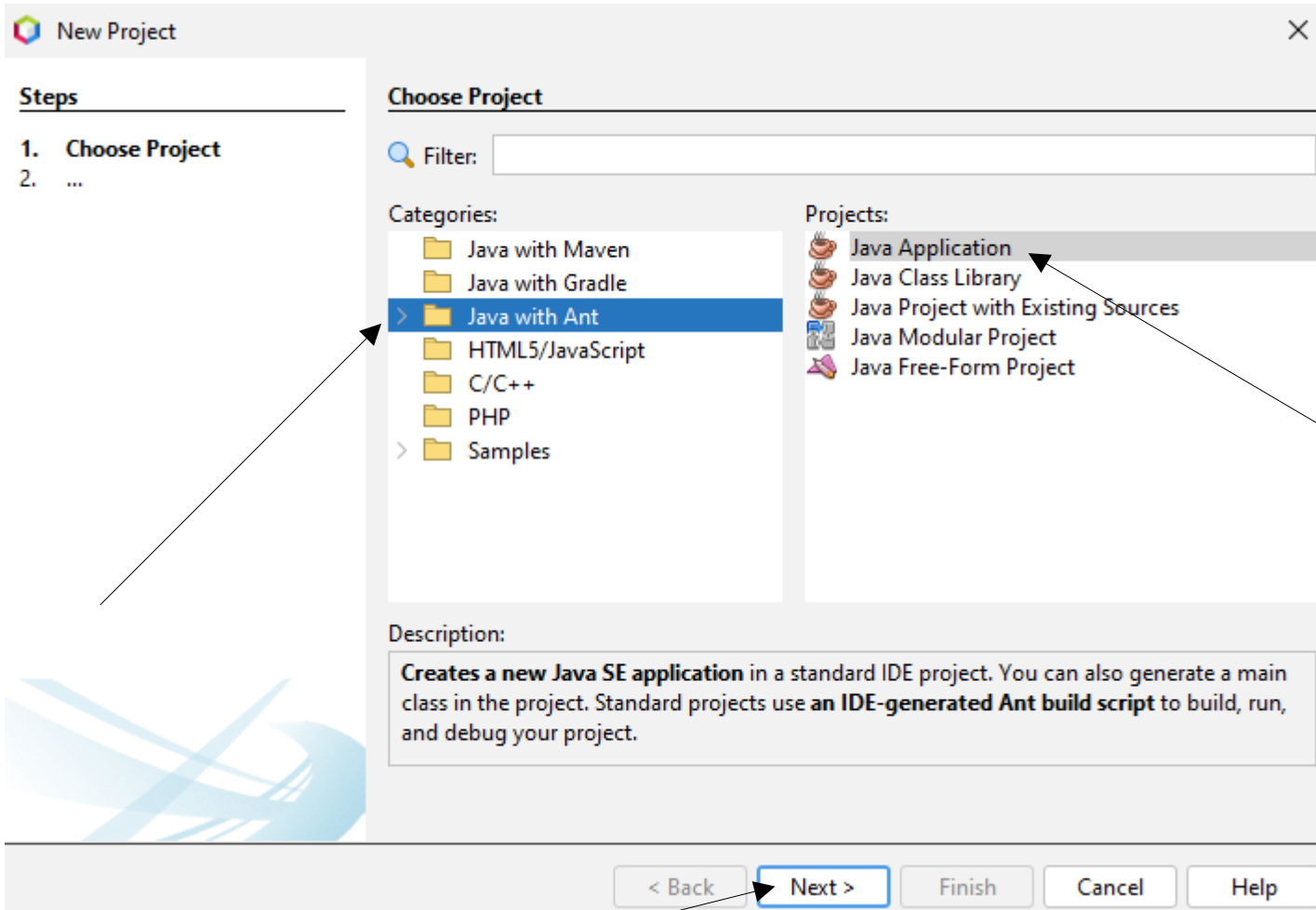
- Vamos criar um projeto para testar os componentes visuais;
- Em seguida, vamos criar uma janela gráfica e inserir componentes Swing para testá-los:
 - Clique no botão “New Project” (Ctrl+Shift+N) e crie um novo projeto com o nome TesteSwing.



Abrindo um projeto

- Nas próximas aulas, pode ser necessário abrir um projeto anteriormente criado;
- Para isso, utiliza-se o botão “Open Project” (Ctrl+Shift+O):





New Java Application

Steps

1. Choose Project

2. **Name and Location**

Name and Location

Project Name: TesteSwing

Project Location: C:\Users\JP\Documents\NetBeansProjects Browse...

Project Folder: C:\Users\JP\Documents\NetBeansProjects\TesteSwing

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class testeswing.TesteSwing

< Back

Next >

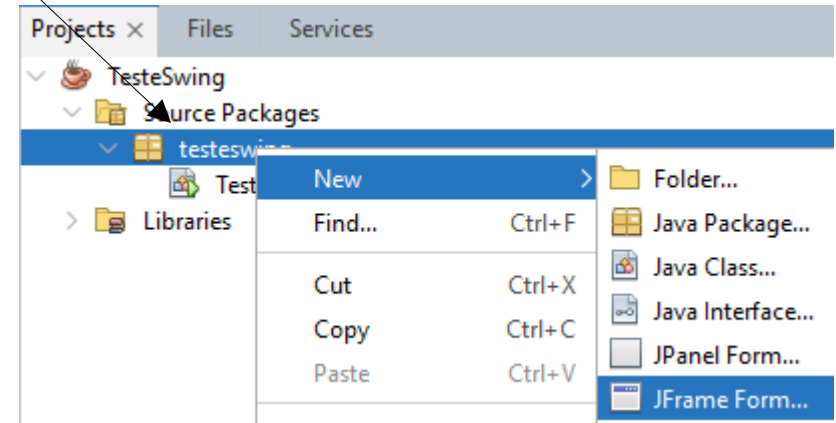
Finish

Cancel

Help

Classe JFrame

- Classe que implementa uma janela gráfica em Java.
- Clique com o botão direito no pacote e escolha New > JFrame



New JFrame Form

Steps

1. Choose File Type

2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

Superclass:

Browse...

Interfaces:

Browse...

< Back

Next >

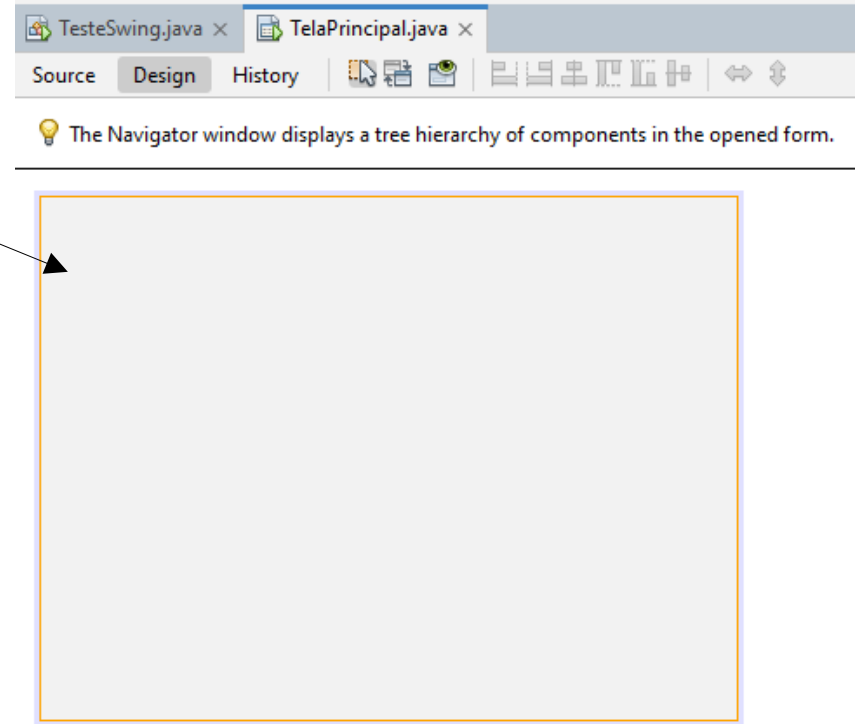
Finish

Cancel

Help

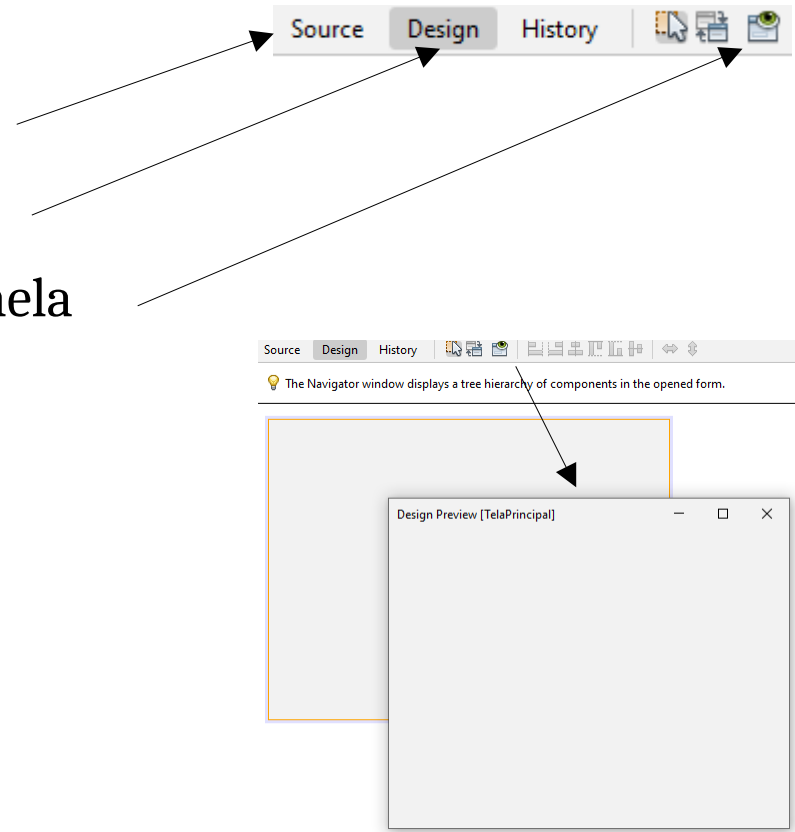
JFrame

- J = Java
- Frame = Quadro (Janela)
- Veja que uma janela foi criada;
- Podemos modelar ela e inserir outros componentes, como rótulos, botões, caixas de texto... que são outros componentes visuais (classes) da biblioteca Swing, conforme veremos a seguir.



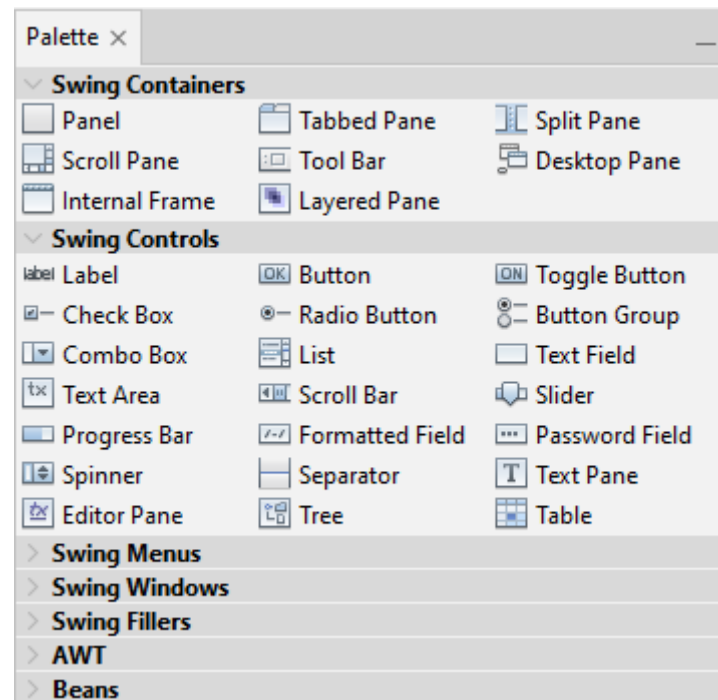
Barra superior do JFrame

- Nesta barra temos os botões:
 - Source: exibe o código fonte da Janela;
 - Design: retorna para o modo desenho;
 - Preview: mostra como está ficando a janela



Palette

- Paleta de componentes;
- Atalho: Ctrl+Shift+8
- Exibe todos os componentes da biblioteca Swing;
- Para inserir o componente, basta escolher o item da paleta e arrastar para o JFrame.



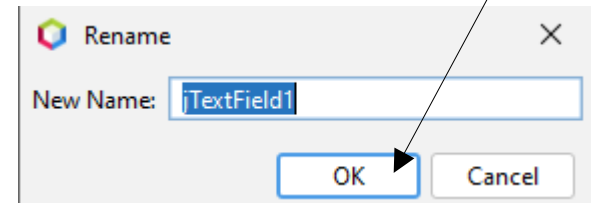
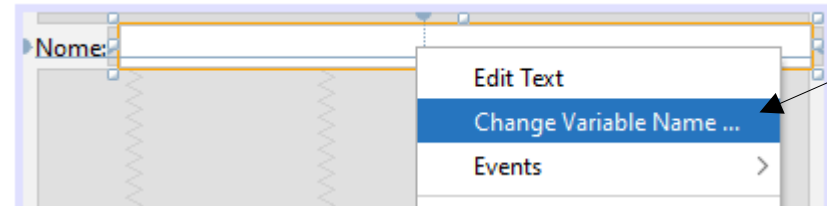
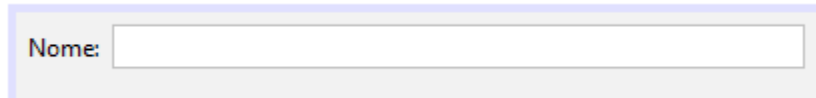
Componente JLabel

- Label = rótulo;
- Localize o componente Label na paleta;
- Arraste-o para o JFrame e posicione-o;
- Pressione F2 para renomeá-lo e Enter pra finalizar;



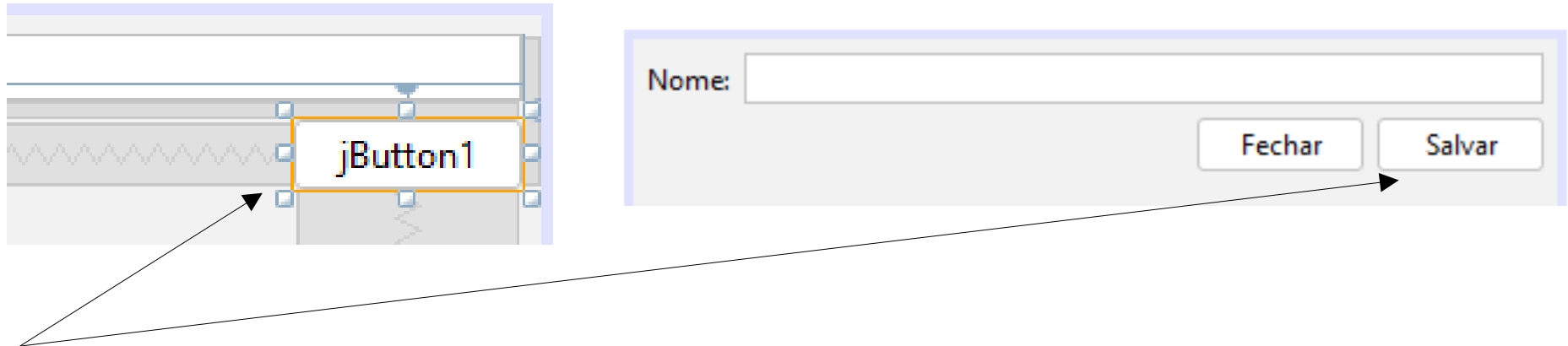
Componente JTextField

- Text Field = Campo de texto;
- Com a tecla F2 altere o conteúdo ou deixe sem texto;
- Ajuste sua largura com a janela;
- Clicando com o botão direito no componente, podemos alterar o **nome da variável** que representa o componente no código fonte.



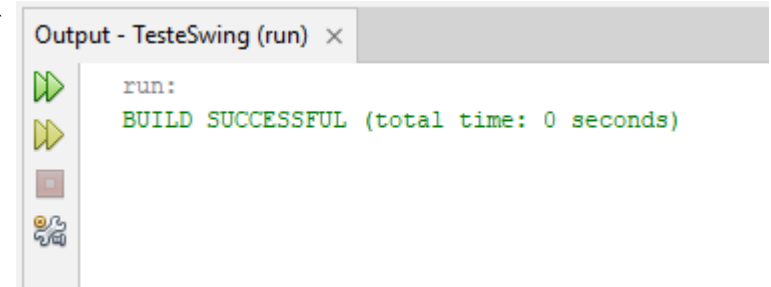
Componente JButton

- Button = Botão
- Componente que representa um botão;
- É possível ajustar sua largura, nome da variável e o texto do botão da mesma forma que os outros componentes.



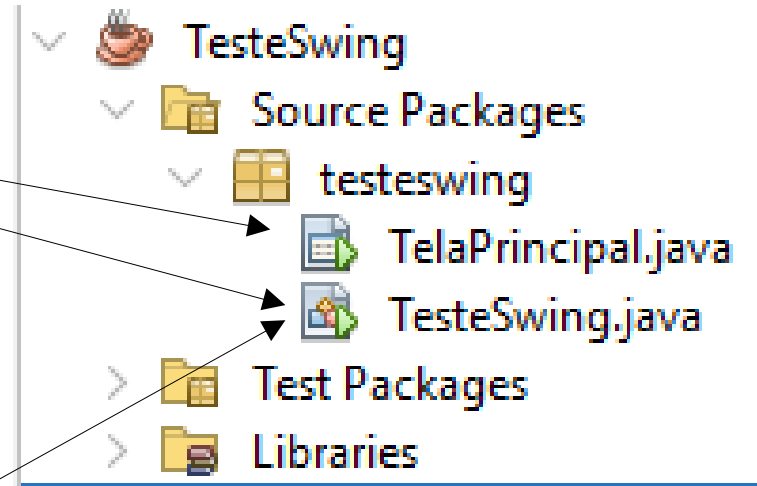
Executando o projeto

- Executar o projeto é o mesmo que executar o programa após compilado:
 - O que irá acontecer quando o usuário executar (rodar) o programa (software) que criamos?
- Pressione F6 ou o botão **play** (Run Project) abaixo do menu Tools;
 - Na parte inferior da janela surgirá a janela Output;
 - Veja que nada aconteceu.



Classe Principal

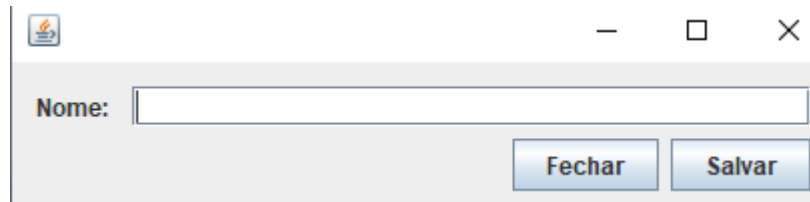
- Veja que nosso projeto tem duas classes;
- Qual delas o NetBeans deve executar primeiro?
- A classe principal é a primeira que foi criada:
 - Ao criar o projeto, o NetBeans cria a classe principal com o mesmo nome do projeto.



Mostrar a Janela JFrame

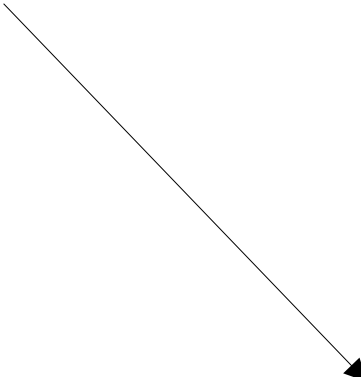
- Localize o método **main** na classe principal do seu projeto;
- Para exibir o JFrame, precisamos criar um objeto e, em seguida, chamar o método **setVisible**;
- Agora execute o projeto e veja o resultado;

```
public static void main(String[] args) {  
    TelaPrincipal tp = new TelaPrincipal();  
    tp.setVisible(true);  
}
```



Encadeamento (Chaining)

- Outra forma de criar o objeto e chamar o método setVisible:

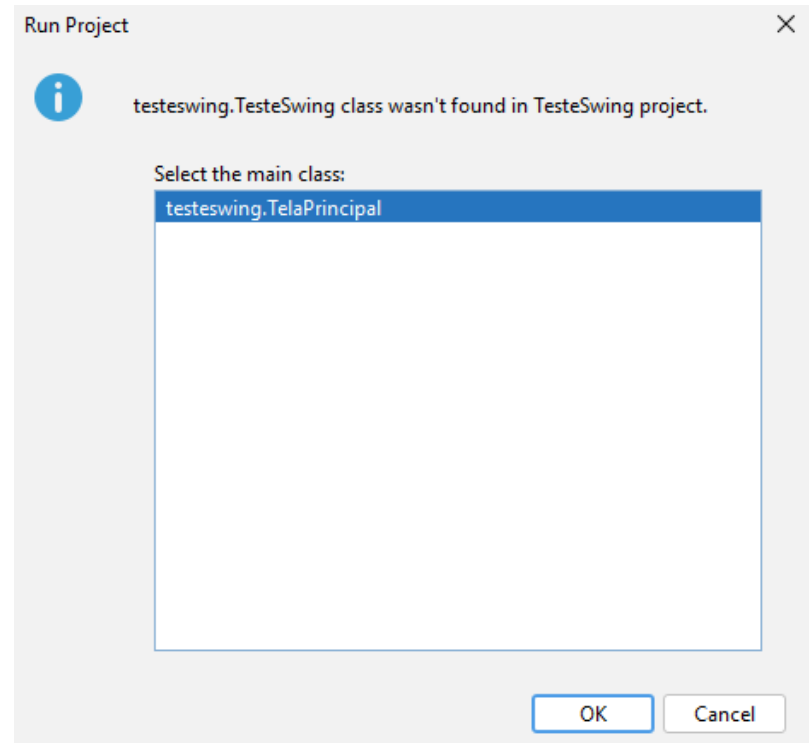


```
public static void main(String[] args) {  
    TelaPrincipal tp = new TelaPrincipal();  
    tp.setVisible(true);  
}
```

```
public static void main(String[] args) {  
    new TelaPrincipal().setVisible(true);  
}
```

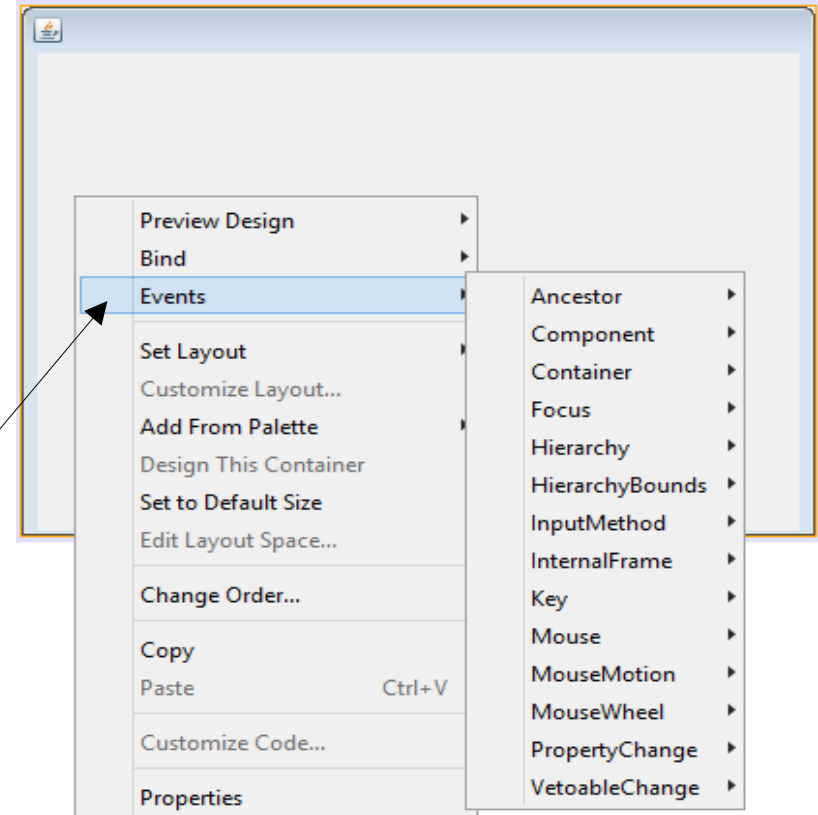
Run Project

- Opcionalmente, podemos excluir a classe principal;
- O NetBeans, nesse caso, perguntará qual será a nova classe principal a ser executada quando clicarmos no botão Run project:



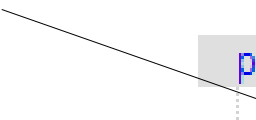
Eventos

- Podemos configurar todo tipo de evento em qualquer tipo de componente da biblioteca Swing;
- Exemplos: clique no botão, foco na caixa de texto, mouse passando sobre o botão, fechamento, minimização etc;
- Clique com o botão direito sobre o componente e selecione a opção “Events”.



Codificando evento no botão Fechar

- Ajuste o nome da variável do botão Fechar para **btnFechar**;
- Para codificar o evento do botão, dê um duplo clique no botão e será direcionado ao código fonte:



```
private void btnFecharActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0); // fecha o programa  
}
```

- **Atenção:** As linhas em cinza só podem ser alteradas pelo NetBeans;
- Execute o projeto e clique em Fechar para testar a funcionalidade.

Convenções Java

- **Case Sensitive:** Java diferencia letras maiúsculas de minúsculas:
 - Idade, idade e IDADE são 3 variáveis diferentes.
- **Modificadores de acesso:** public, private ou protected;
- **Tipos de retorno:** void, tipo primitivo ou classe;
- **Pascal Case:** classes cujo nome é formado de múltiplas palavras, devem ter o primeiro caractere de cada palavra em maiúsculo:
 - Usuario, Pessoa, Produto, Fabricante, TelaPrincipal, ActionPerformed, ActionEvent, JFrame, String
- **Camel Case:** métodos/variáveis/objetos :
 - Variáveis/objetos:
 - nome, usuario, cpfUsuario, nomeUsuario, dataNasc, evt, args
 - Métodos:
 - setNome, btnFechar, btnFecharActionPerformed, initComponents, setVisible
- **Snake Case:** constantes (utilizam a palavra reservada **final**):
 - ALTURA, ALTURA_TELA, LARGURA_TELA, PI, WIDTH, HEIGHT, YES_OPTION, CANCEL_OPTION

Path (caminho)

- No Windows:

C:\Windows\System32\xcopy.exe

- No Linux:

/var/log/syslog

- Em Java:

java.awt.event.ActionEvent

pacotes

classe

diretórios

arquivo


```
private void btnFecharActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0); // fecha o programa  
}
```

- **private:** método privado
- **void:** não retorna valor
- **btnFecharActionPerformed:** método criado automaticamente pelo NetBeans
 - para gerenciar o evento “Action Performed” (ação executada)
 - disparado quando o usuário clica ou pressiona enter quando o botão está selecionado;
- **java.awt.event.ActionEvent:** classe Java para gerenciar eventos
 - java.awt.event é o pacote
 - ActionEvent é a classe
- **evt:** objeto (variável) criado pelo NetBeans para gerenciar o evento atual

Classe e Objeto

- Uma classe é a planta da casa:
 - Classe é um arquivo .java com as características definidas pelo programador: campos (variáveis) e métodos.
- O objeto é a casa construída:
 - Podemos criar quantos objetos forem necessários, a partir da mesma classe.
- Como criar um objeto (**instanciar** um objeto):
 <nome da classe> <nome do objeto> = new <método construtor>
 - Exemplo: Pessoa p = **new** Pessoa()
- Ao criar um objeto, podemos executar seus métodos:
 - p.setNome("João"); p.getNome(); pessoaController.salvar(p);

Classe Abstrata

- Não permite criar objetos (abstract);
- Somente permite executarmos seus métodos estáticos (static);
- Utiliza-se o nome da classe ao invés do nome do objeto:

- Exemplos:

- Msg.bomDia();**

- Printer.print(f);**

- System.exit(0);**

System.exit(0)

- Comando Java para fechar o programa:
 - System é uma classe Java
 - exit é um método da referida classe
 - argumento **zero**: significa que programa está sendo fechado por opção do usuário e não por um problema interno
 - argumento **> 0**: significa que o fechamento do programa ocorreu por uma falha ou exceção interna do seu programa
 - O programador deve escolher a opção mais adequada para cada caso.

Classe Math

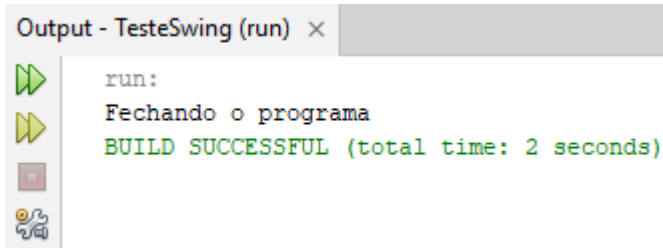
- Classe que disponibiliza métodos para operações matemáticas;
- Faz parte da biblioteca padrão da linguagem Java:
 - Não é necessário importar.

Exemplos de métodos da classe Math

- `Math.PI`
Constante que retorna o valor 3,14;
- `Math.ceil(valor)`
Arredondar para cima;
- `Math.floor(valor)`
Arredondar para baixo;
- `Math.max(valor1, valor2)`
Retorna o maior valor;
- `Math.min(valor1, valor2)`
Retorna o menor valor;
- `Math.sqrt(valor)`
Retorna a raiz quadrada;
- `Math.pow(base, potencia)`
Retorna a potência;
- `Math.random()`
Retorna um valor aleatório entre 0 e 0.99;

Depurando o código

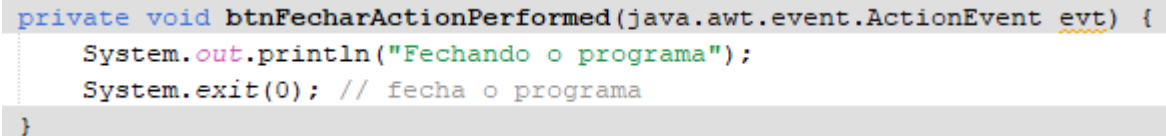
- Uma forma de testar a execução de um trecho de código, é a utilização de saída de texto no prompt de comandos (terminal);
- Vamos incluir uma mensagem de saída ao clicar no botão Fechar;
- Execute novamente e veja o resultado.



Output - TesteSwing (run) ×

```
run:  
Fechando o programa  
BUILD SUCCESSFUL (total time: 2 seconds)
```

An arrow points from the third bullet point to this screenshot.



```
private void btnFecharActionPerformed(java.awt.event.ActionEvent evt) {  
    System.out.println("Fechando o programa");  
    System.exit(0); // fecha o programa  
}
```

An arrow points from the second bullet point to this code snippet.

System.out.println()

- System é a classe Java;
- out é um campo da classe;
- println é um método do campo “out” (classe PrintStream):
 - Esse método recebe um dado como argumento: texto, variável etc e **apresenta-o no console (prompt)**, quebrando a linha ao final;
 - Opcionalmente podemos utilizar o print, sem quebrar linha ao final:
`System.out.print(“Hello World!”);`

Botão Salvar

- Altere os nomes dos campos de texto para txtNome e txtCpf;
- Adicione o código do botão btnSalvar:

```
private void btnSalvarActionPerformed(java.awt.event.ActionEvent evt) {  
    String nome = txtNome.getText();  
    String cpf = txtCpf.getText();  
  
    JOptionPane.showMessageDialog(rootPane, nome+"\n"+cpf);  
}
```

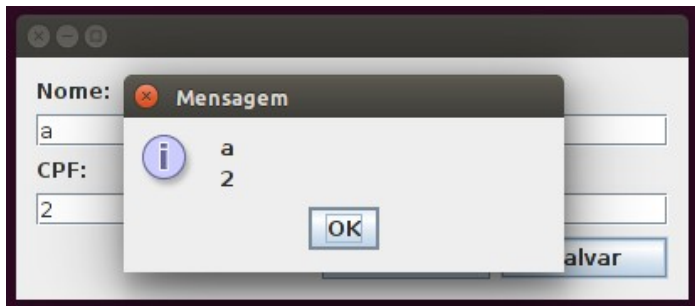
Podemos substituir aqui por null



Execute o projeto

- Se o seu JFrame não aparecer na tela, falta configurar o método *main* para exibi-lo:

```
public static void main(String[] args) {  
    // instancia a janela e a exibe  
    new PessoaCadastro().setVisible(true);  
}
```



Variáveis do tipo String

- Java utiliza o caractere de soma para permitir concatenação:
 - **Concatenar:** juntar duas Strings
 - ou uma String e outro tipo de dado primitivo;

```
String nome = "João";
```

```
nome = nome+"!"; // concatenando duas Strings
```

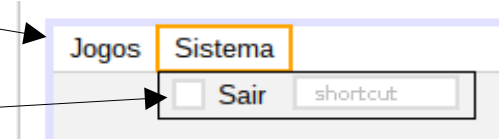
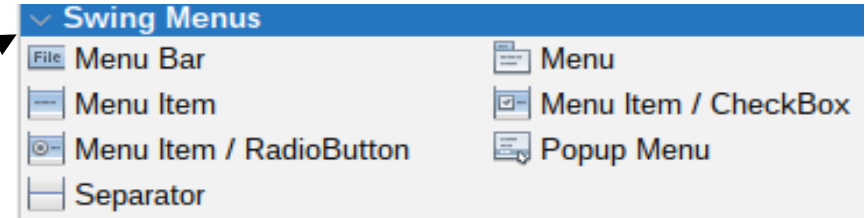
```
nome += "!"; // idem anterior
```

```
String a = "[" +1+ "]"; // concatenando Strings com inteiro
```

```
System.out.println(a); // resulta em [1]
```

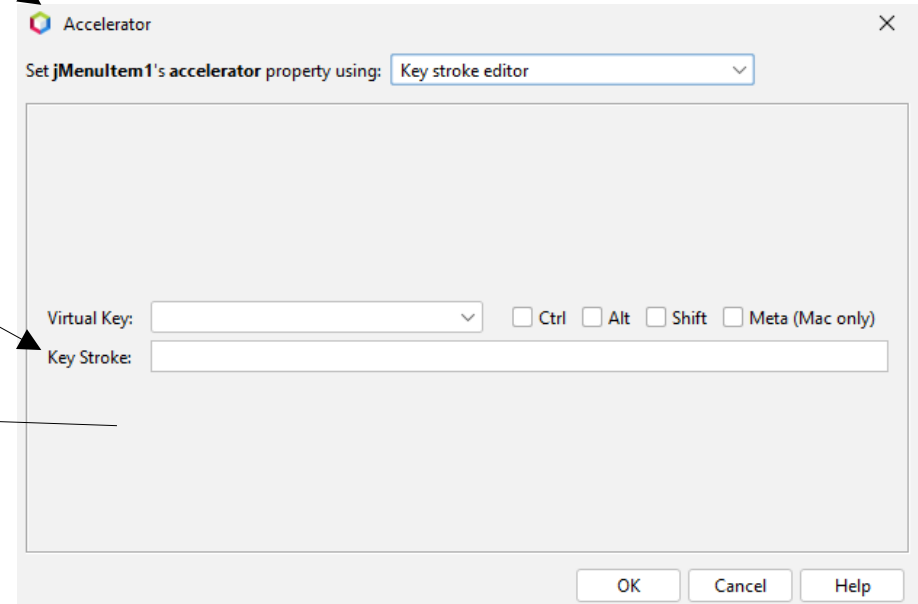
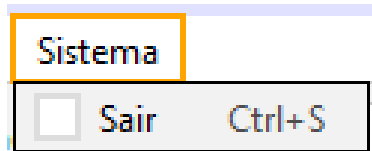
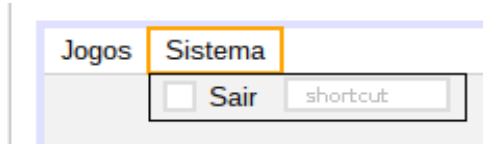
Menus

- Na Palette do NetBeans, temos a categoria de componentes ‘Swing Menus’.
- Arraste um ‘Menu Bar’ para adicionar uma barra de menus;
- Arraste um ‘Menu Item’ para adicionar cada item de menu:
 - F2 para renomear o menu.



Atalho de Menu

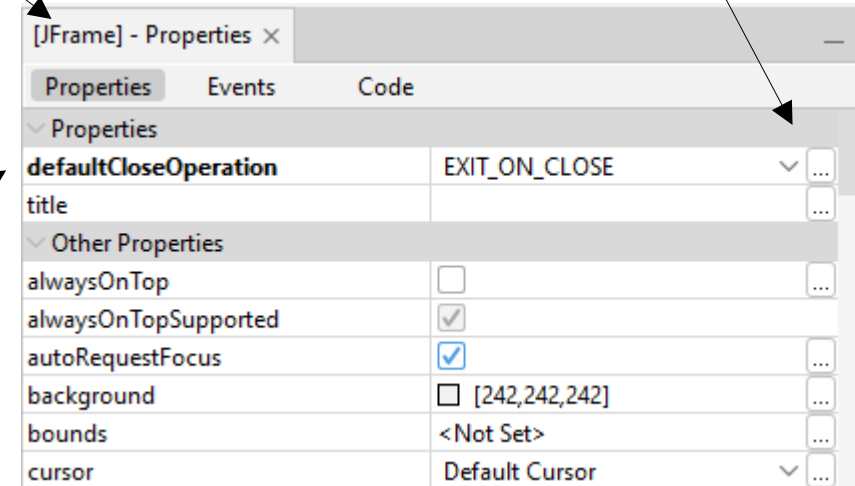
- Duplo clique em “shortcut” para abrir o utilitário ‘Accelerator’;
- Clique em ‘Key Stroke’ e digite a combinação de teclas Ctrl+S para definir.



Properties

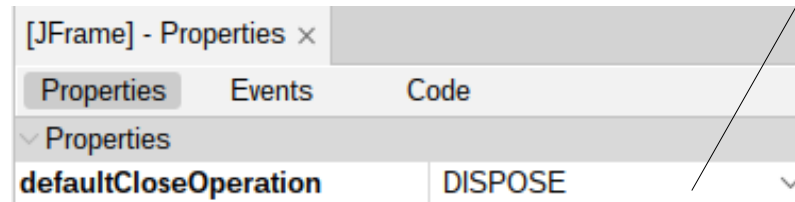
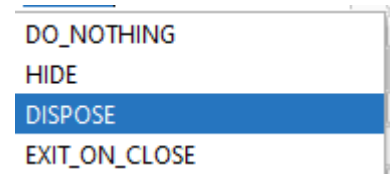
- Janela de propriedades do elemento atual (elemento selecionado);
- Atalho: Ctrl+Shift+7
- O nome da classe do elemento selecionado é exibida entre colchetes na aba da janela;
- As propriedades são listadas, permitindo alterar suas configurações padrão;
- As propriedades que forem alteradas ficam em negrito;

O botão reticências [...] na frente de cada propriedade abre um utilitário de configuração (opcional).



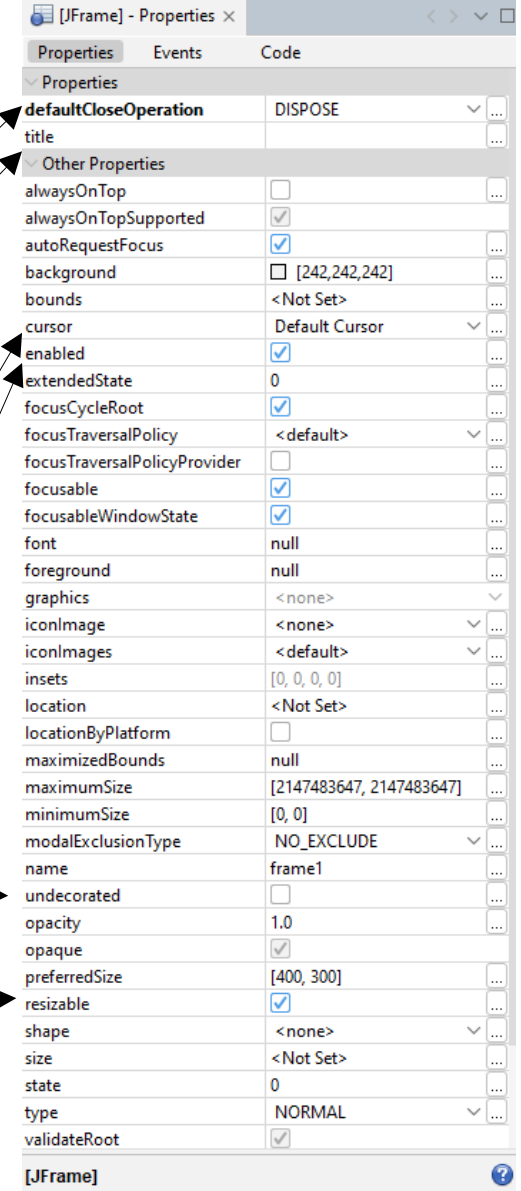
defaultCloseOperation

- Permite definir a operação executada o usuário fecha a janela (JFrame):
 - **DO_NOTHING** (fazer nada): desativa o botão fechar do canto superior direito da janela;
 - **HIDE** (ocultar): oculta a janela, mas ela continua em execução em segundo plano (consumindo processamento e memória RAM);
 - **DISPOSE** (abrir mão / dispor): a janela é fechada e o programa principal (caso exista) continua em execução;
 - **EXIT_ON_CLOSE** (sair ao fechar): encerra o programa todo.
- Quando temos sub-janelas, esse comportamento deve ser alterado para DISPOSE:

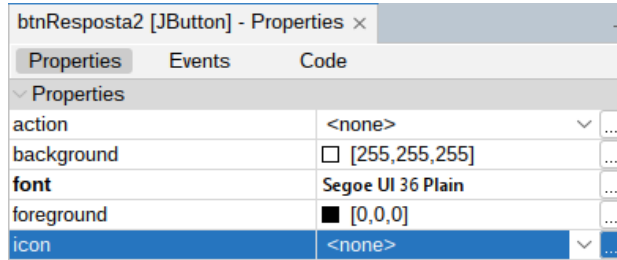


Propriedades JFrame

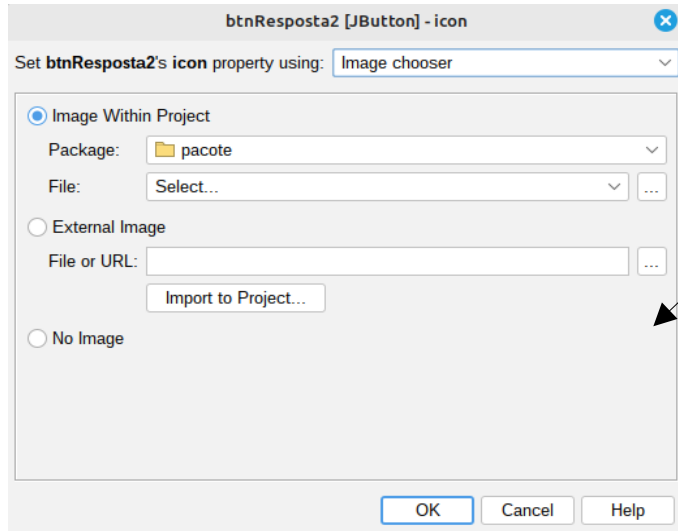
- Operação do botão de fechar (x) no canto superior direito da janela;
- Título da janela;
- Muda o cursor quando o mouse passa sobre o elemento (neste caso a janela);
- Permite desabilitar (ex: botão);
- Retira os botões de minimizar, restaurar e fechar;
- Impede o usuário de maximizar ou alterar o tamanho da janela;



Icon



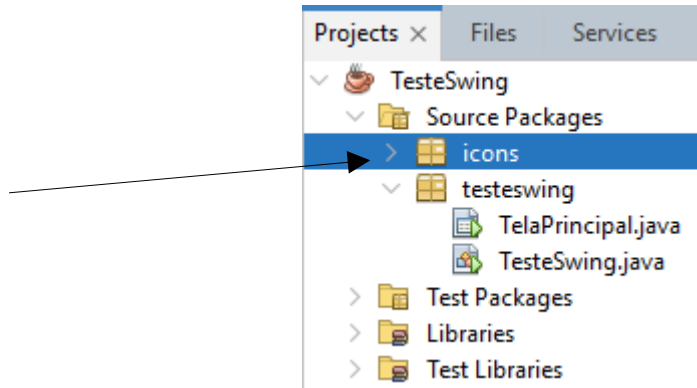
- Selecione o botão ou rótulo e clique no botão de [...] na propriedade 'icon';



- O utilitário permitirá selecionar a imagem a ser incluída na interface.

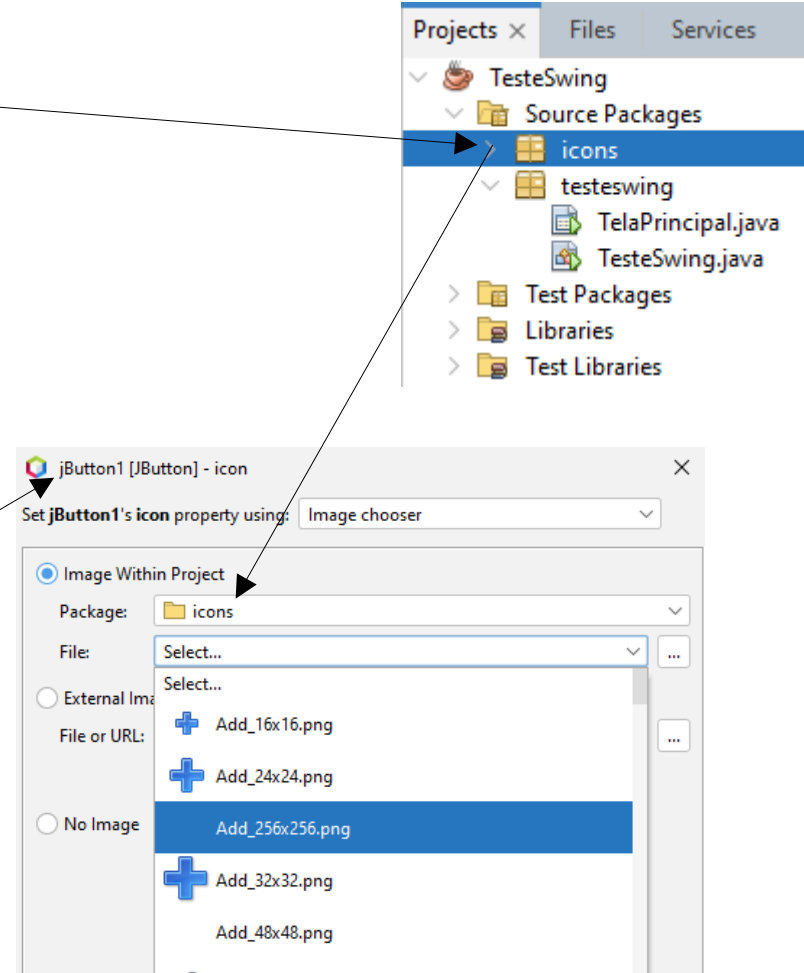
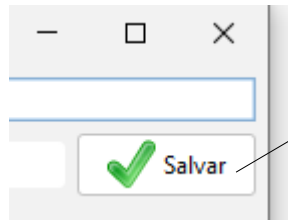
Pacote de ícones

- Descompacte o arquivo **icons.zip**, arraste a pasta e solte sobre o pasta 'Source Packages' do projeto:



Pacotes de ícones

- Se você criar um pacote e colocar arquivos de imagens ao invés de arquivos java, terá um pacote de ícones;
- Assim, você poderá selecionar o ícone através do utilitário do slide anterior;



O que aprendemos?

- Classes de componentes gráficos;
- Desenvolvimento de UI no NetBeans;
- Conceitos introdutórios sobre orientação a objetos.
- Eventos em componentes;

Para praticar...

- Crie um projeto no NetBeans, faça os exemplos e teste os recursos estudados nesta aula;
- **Desafio:** além do evento de clique no botão, seria possível mostrar uma mensagem ao clicar em um rótulo?
- Anote suas dúvidas e traga para a próxima aula;