



# Desenvolvimento de interfaces gráficas em Java

Prof. Nelson Bellincanta Filho

# GUI (Graphical User Interface)

- ▶ Uma GUI (Graphical User Interface) é uma interface padronizada simples e intuitiva para os usuários;
- ▶ A interface gráfica de um sistema é um dos elementos mais impactantes para a aceitação de um sistema pelo cliente e validação do mesmo pelos usuários;;
- ▶ Java oferece capacidades únicas na criação de aplicações gráficas que, sem modificação ou recompilação, são executadas em diferentes ambientes.

# Componentes

- ▶ Um dos elementos-chave do sucesso das GUIs é a utilização de componentes padronizados (components) para representação e operação dessas interfaces;
- ▶ Dessa forma, em todo o processo de criação de uma interface é necessário seguir as orientações de boas práticas na utilização de componentes, inclusive quanto ao posicionamento e agrupamento (DEITEL, 2016);
- ▶ A construção de interfaces gráficas é feita a partir da inserção de componentes em uma tela, como por exemplo, botões, caixas de seleção e a própria representação da tela, dentre outros.

# Componentes

- ▶ O Java oferece duas bibliotecas de componentes para GUI na forma de classes pertencentes aos pacotes `java.awt` e `javax.swing`;
- ▶ Que representam os frameworks de componentes AWT e Swing, em que predomina o modelo componente/compartimento:
  - ▶ Todo componente é suportado pelo compartimento (container) no qual se localiza e cuja interface dá acesso aos serviços gerais providos pelo container.

AWT



# AWT

- ▶ A AWT (Abstract Window Toolkit) constitui a primeira biblioteca de componentes do Java;
- ▶ Contém um conjunto de componentes básicos, um modelo de eventos robusto, gerenciadores de layout e classes auxiliares;
- ▶ Sua proposta central era fornecer o maior conjunto comum entre as plataformas suportadas pelo Java.

# AWT

- ▶ Possui um mecanismo complexo e pesado de troca de mensagens, exigido entre o sistema operacional e a JVM;
- ▶ O número de componentes disponíveis é relativamente pequeno e a construção de novos widgets é complexa ou dependente da plataforma.

Swing



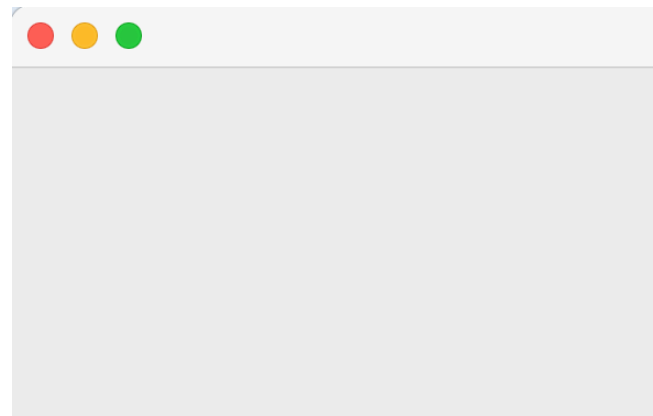


# Swing

- ▶ Para superar as limitações da AWT e suprir outras necessidades relacionadas às interfaces GUI, foi desenvolvida a Java Foundation Classes (JFC) no Java 1.2, um framework destinado à construção de aplicações baseadas em componentes e gráficas que incluiu:
  - ▶ Swing;
  - ▶ Pluggable Look and Feel;
  - ▶ Java 2D;
  - ▶ Acessibilidade;
  - ▶ Drag and Drop.

# Swing

- ▶ O Swing é uma biblioteca que vem incorporada no Java Development Kit, dispondo de diversos elementos para a produção dessas telas (HORSTMANN, 2016);
- ▶ O primeiro elemento a ser utilizado em uma interface gráfica é a representação de uma área que apresenta uma barra de título e um espaço reservado para se adicionar componentes;



JFrame.

# Integrated Development Environment (IDE) Eclipse

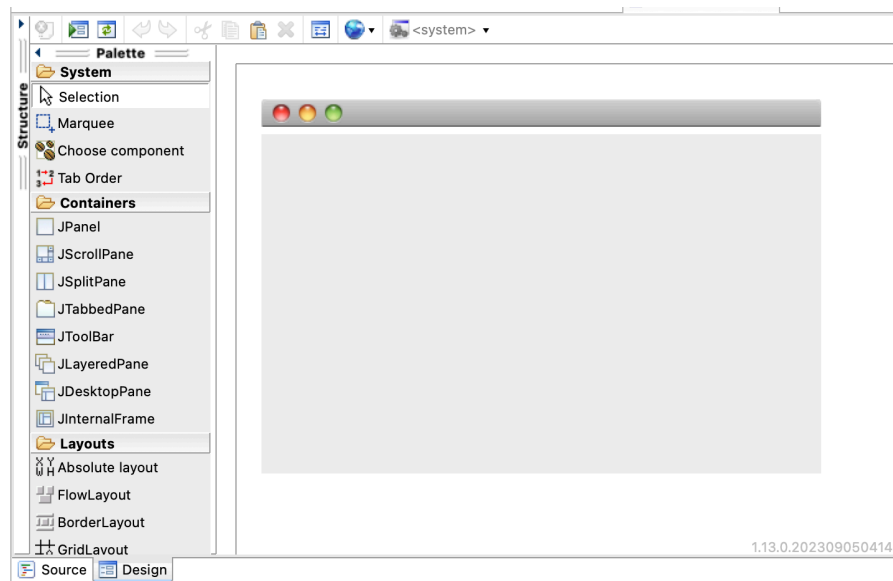


# Integrated Development Environment (IDE) Eclipse

- ▶ Todos esses elementos de tela são passíveis de alteração;
- ▶ As modificações podem ser feitas por código em Java sem a necessidade de utilizar algum editor gráfico;
- ▶ No entanto, existem editores para a criação de telas utilizando o Swing;
- ▶ No caso do Integrated Development Environment (IDE) Eclipse, existem plug-ins e extensões que propiciam esse editor (WINDER, 2009).

# Integrated Development Environment (IDE) Eclipse

- Um dos editores gráficos acoplados ao Eclipse é o Windows Builder, que fornece diversas ferramentas e formas de criar a janela (WINDER, 2009).



Editor para interfaces gráficas acoplado ao Eclipse.

Criando um interface gráfica



# Criando um interface gráfica

- ▶ Para se criar uma interface gráfica são necessários cinco passos (DEITEL, 2016):
  - ▶ 1. Criar uma relação de especialização com a classe que representa sua tela;
  - ▶ 2. Declarar como atributos os elementos que serão adicionados à tela;
  - ▶ 3. Definir a forma de alocação dos elementos gráficos na tela;
  - ▶ 4. No construtor, instanciar, configurar e posicionar os itens na tela;
  - ▶ 5. Tratar os eventos dos componentes para tratar as ações do usuário com a interface gráfica.

# Criando um interface gráfica

- ▶ Os componentes que farão parte da interface gráfica podem ser configurados e posicionados na tela através dos atributos e métodos;
- ▶ Por exemplo, para inserir uma legenda e um campo para o usuário digitar seu nome, utilizam-se os componentes JLabel e JTextField;
- ▶ Segundo Deitel (2016) os objetos do tipo JLabel devem ser usados para representar textos estáticos em interfaces;
- ▶ Já para receber entradas de texto sem formatação, Horstmann (2016) indica o uso de objetos do tipo JTextField.



# Criando um interface gráfica

Componente	Descrição
JButton	Objeto usado para criar botões.
JCheckBox	Objeto usado para oferecer uma opção para o usuário. Normalmente é representado por uma caixa de seleção, que quando está com “check” representa “sim” e quando está com “não” representa “não”.
JComboBox	Objeto usado para oferecer mais de uma opção para o usuário em forma de lista drop-down. Para cada lista somente um item pode ser selecionado.

Componentes básicos da biblioteca Swing.

# Criando um interface gráfica

Componente	Descrição
JList	Objeto usado para oferecer mais de uma opção para o usuário, mas, diferentemente do JComboBox, esse componente permite a seleção de mais que uma opção.
JPanel	Objeto usado para organizar diversos componentes.

Componentes básicos da biblioteca Swing.

# Criando um interface gráfica

- ▶ Para campos de entrada que precisam de formatação específica, como, por exemplo, um CPF, o objeto `JFormattedTextField` pode ser usado para especificar máscaras;
- ▶ Segundo Deitel (2016), para a máscara do objeto `JFormattedTextField` é possível utilizar os seguintes codificadores:
  - ▶ # para representar a entrada de números;
  - ▶ U para letras em caixa alta;
  - ▶ L para letras em caixa baixa;
  - ▶ A para qualquer número ou letra;
  - ▶ ? para qualquer caractere;
  - ▶ \* para qualquer elemento;
  - ▶ H para entrada em hexadecimal.

# Criando um interface gráfica

- ▶ O Java possui uma vasta documentação e ferramentas para ajudar no desenvolvimento de interfaces gráficas, tais como:

- ▶ Documentação geral do Java

<https://docs.oracle.com/en/java/javase/21/>

- ▶ Tutorias para Java Swing

<https://docs.oracle.com/javase/tutorial/uiswing/>

<https://web.mit.edu/6.005/www/sp14/psets/ps4/java-6-tutorial/components.html>

# Referências

FURGERI, Sérgio. **Java 8: Ensino Didático**. 1ª ed., São Paulo: Érica, 2015.

DEITEL, Harvey M.; DEITEL, Paul J. **Java: como programar**. 10ª ed. Editora Pearson Education do Brasil, 2017.

JUNIOR, Peter Jandl. **Java Guia do Programador**. 4a Edição: Atualizado para Java 16. Novatec Editora, 2021.