

Java First-Tier: Aplicações

Construção de Interfaces com Swing (I)

Grupo de Linguagens de Programação



Departamento de Informática
PUC-Rio

Abstract Window Toolkit (AWT)

- *Peer-based*
 - Delega ao *toolkit* nativo a criação de elementos gráficos
 - *Look & feel* da plataforma de execução
- Limitações
 - Falta de consistência entre plataformas
 - “Denominador comum” para atender a todas as plataformas
 - “Bugs” diferentes em plataformas diferentes
 - “Write once, test everywhere”

2

Swing versus AWT

- *Non-peer-based GUI toolkit*
- Conjunto maior de elementos
 - Mais funcionalidade
- Maior consistência de uso entre plataformas
 - Mais “robusto”, maior portabilidade
- Pacote **javax.swing**
 - Pacote de extensão no Java 1.1
 - Incorporado como pacote básico no Java 2

3

Swing: Características

- Componentes leves
- Não substitui integralmente o AWT
 - Estende o AWT
 - Diagramadores & *Listeners*
- *Look & Feel* configurável
 - Windows, Motif, Metal
- *Double-buffering* automático
- Arquitetura MVC

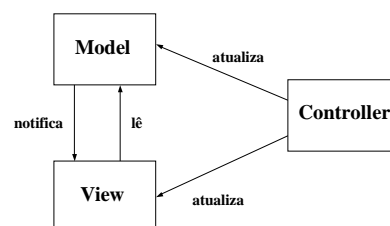
4

Arquitetura MVC

- O Swing adota uma arquitetura conhecida como *Model-View-Controller (MVC)*
 - Modelo = dados/conteúdo
 - estado de um botão, texto
 - Visão = aparência
 - cor, tamanho
 - Controle = comportamento
 - reação a eventos

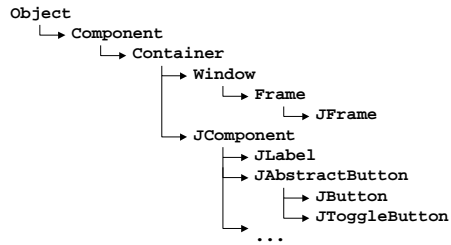
5

Interação entre os objetos MVC



6

Hierarquia de Classes



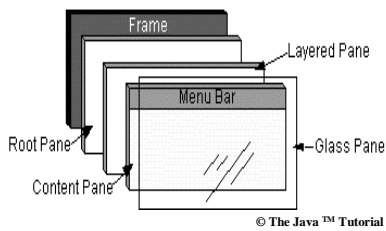
7

Top Level Containers

- Todo programa que utiliza componentes Swing tem pelo menos um TLC
 - raiz de uma “*containment hierarchy*”
- Todo TLC possui um “*content pane*” que contém os elementos visíveis
 - opcionalmente, uma “*menu bar*”
- JFrame, JDialog, JApplet
 - aplicações tipicamente possuem pelo menos uma hierarquia com um JFrame como raiz (janela principal)
 - applets Swing contém uma hierarquia com JApplet como raiz

8

Estrutura de um JFrame



9

Camadas do JFrame

- RootPane
 - gerencia as demais camadas
 - botão “default”
- LayeredPane
 - Contém a *menu bar* e o ContentPane
 - Pode conter subcamadas (*Z order*)
- ContentPane
 - contém os componentes visíveis
- GlassPane
 - invisível por *default*
 - interceptação de eventos/pintura sobre uma região

10

Exemplo de JFrame

```

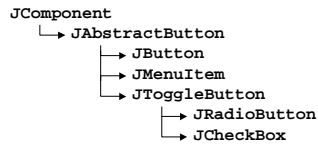
JFrame f = new JFrame("Teste");
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
Container cp = f.getContentPane();
...
cp.add(...);
...
f.pack();
f.show();
  
```

11

Botões de Dois Estados

12

Hierarquia



13

Classe JRadioButton

- Modela um botão de escolha que pode ser marcado e desmarcado
- Objetos **JRadioButton** são tipicamente organizados em grupos
 - Um único botão do grupo pode estar marcado em um dado momento



14

Métodos de JRadioButton

```

JRadioButton(String label)
JRadioButton(String label, boolean state)

boolean isSelected()
void setSelected(boolean state)
    
```

15

Classe ButtonGroup

- Cria um “escopo” de exclusão para um grupo de botões
- Basta criar um **ButtonGroup** e adicionar a ele os **JRadioButtons** que compõem o grupo

16

Métodos de ButtonGroup

```

void add(AbstractButton b)

ButtonModel getSelection()

boolean isSelected(ButtonModel m)
void setSelected(ButtonModel m,
                 boolean state)
    
```

17

Exemplo de JRadioButton

```

JFrame f = new JFrame("Teste");
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JRadioButton bm = new JRadioButton("Masculino", true);
JRadioButton bf = new JRadioButton("Feminino");
ButtonGroup bg = new ButtonGroup();
bg.add(bm);
bg.add(bf);
Container cp = f.getContentPane();
cp.setLayout(new FlowLayout());
cp.add(bm);
cp.add(bf);
f.pack();
f.show();
    
```

18

Classe JCheckBox

- Modela um botão de escolha que pode ser marcado e desmarcado



19

Métodos de JCheckBox

```
public JCheckBox(String label)
public JCheckBox(String label,
                 boolean state)

public boolean isSelected()
public void setSelected(boolean state)
```

20

Exemplo de JCheckBox

```
JFrame f = new JFrame("Teste");
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JCheckBox ci = new JCheckBox("Inglês", true);
JCheckBox ce = new JCheckBox("Espanhol", true);
JCheckBox cf = new JCheckBox("Francês");
Container cp = f.getContentPane();
cp.setLayout(new FlowLayout());
cp.add(ci);
cp.add(ce);
cp.add(cf);
f.pack();
f.show();
```

21

Elementos de Edição de Texto

22

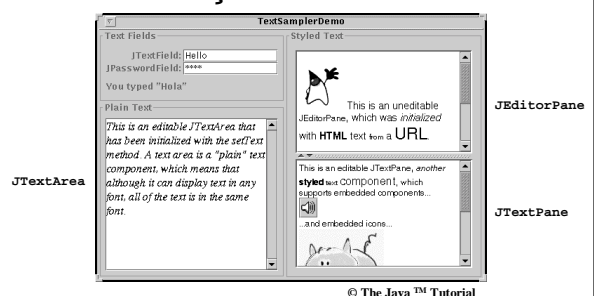
Classe JTextComponent

- Classe abstrata que modela o que há de comum entre diferentes elementos de edição de texto

```
public void setText(String t)
public String getText()
public void setEditable(boolean b)
```

23

Elementos de Edição de Texto



© The Java™ Tutorial

24

Classe **JTextField**

- Cria um campo de edição de texto de uma linha

```
JTextField()
JTextField(String text)
JTextField(int columns)
JTextField(String text, int columns)
void setColumns(int columns)
```

25

Classe **JPasswordField**

- Estende **JTextField**
- Caracteres digitados não são exibidos

```
JPasswordField()
JPasswordField(int columns)
JPasswordField(String text, int columns)
char[] getPassword()
void setEchoChar(char c)
```

26

Classe **JTextArea**

- Cria um campo de edição de texto com múltiplas linhas

```
JTextArea(int rows, int columns)
JTextArea(String text, int rows, int columns)
void append(String t)
void insert(String t, int pos)
void setLineWrap(boolean wrap)
void setWrapStyleWord(boolean word)
```

27

Classe **JTextPane**

- Componente de texto que suporta atributos representados graficamente (*styled text*)
- Permite o uso de diversas “fontes” no texto
- Permite a inclusão de imagens e de outros componentes

28

Molduras

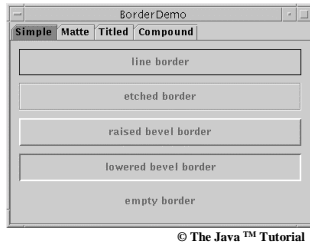
29

Bordas

- O Swing permite a criação de **bordas** (molduras) envolvendo componentes
- O método **setBorder** de **JComponent** permite “emoldurar” um componente com uma borda

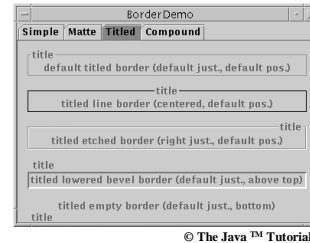
30

Exemplos de Bordas



31

Mais exemplos...



32

BorderFactory

- As bordas do Swing foram projetadas para serem compartilhadas
- Ao invés de criar explicitamente uma instância de uma das classes de bordas, deve-se usar um dos métodos de **BorderFactory** para obter uma instância compartilhada de um tipo de borda

33

Métodos de BorderFactory

```
static Border createLineBorder(Color color)
static Border createEtchedBorder()
static TitledBorder createTitledBorder(String t)
static TitledBorder createTitledBorder(Border b,
                                     String t)
```

34

Painéis, Bordas e Botões



JCheckBox

JRadioButton

35