

Módulo III

Padrões GOF-VI: MVC

Professores

Eduardo Bezerra – edubezerra@gmail.com

Ismael H F Santos – ismael@tecgraf.puc-rio.br

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

1

Ementa

- Padrões – Parte VI
 - Model View Controller
 - MVC em Swing

Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

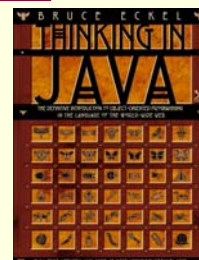
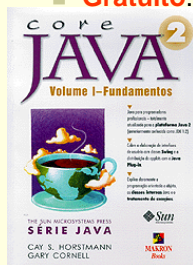
2

Bibliografia

- *Craig Larman, Utilizando UML e Padrões, Ed Bookman*
- *Eric Gamma, et ali, Padrões de Projeto, Ed Bookman*
- *Martin Fowler, Analysis Patterns - Reusable Object Models, Addison-Wesley, 1997*
- *Martin Fowler, Refatoração - Aperfeiçoando o projeto de código existente, Ed Bookman*

Livros

- **Core Java 2**, Cay S. Horstmann, Gary Cornell
 - Volume 1 (Fundamentos)
 - Volume 2 (Características Avançadas)
- **Java: Como Programar**, Deitel & Deitel
- **Thinking in Patterns with JAVA**, Bruce Eckel
 - **Gratuito.** <http://www.mindview.net/Books/TIJ/>



POO-Java

*Padrões
Parte VI*



Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

5

POO-Java

*Model
View
Controller*



Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

6

POO-Java

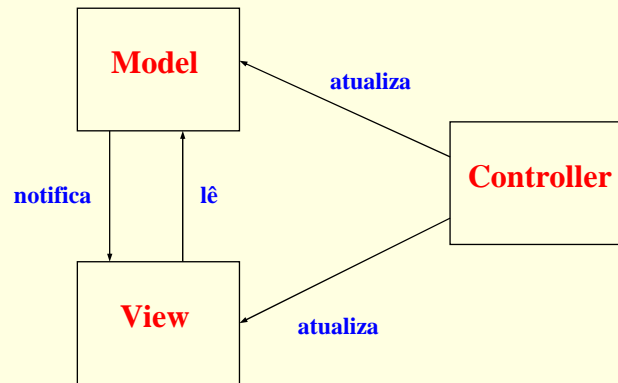
MVC
em
Swing



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

Interação entre os objetos MVC

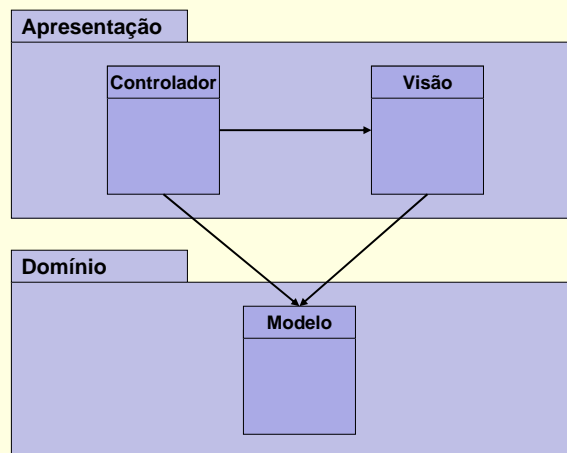


Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

9

Arquitetura MVC no Swing

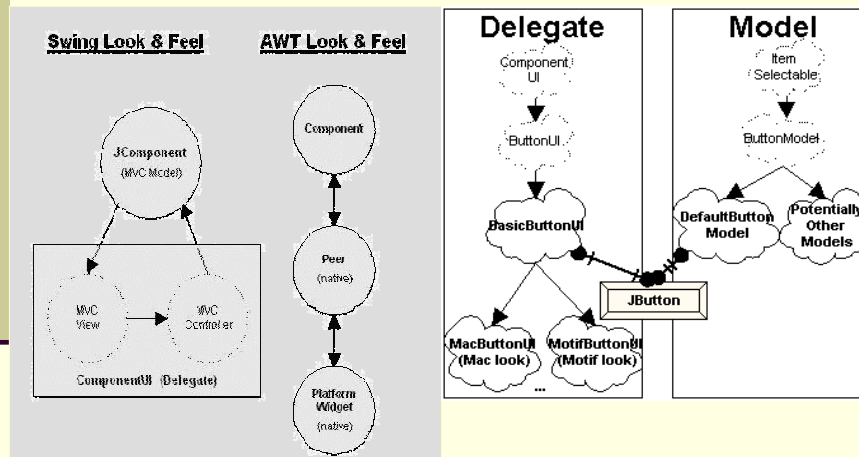


Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

10

Modelo-Delegado

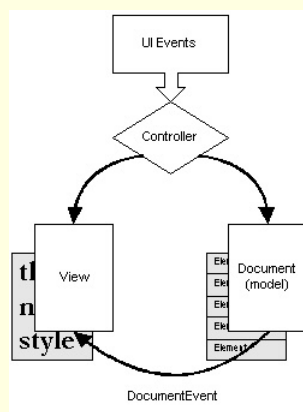


Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

11

Documentos Swing



Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

12

Explorando a Arquitetura MVC

- Como os dados (o modelo) não fazem parte integrante do elemento de interface que os exibe, podemos gerenciá-los em separado
- Por exemplo, é possível exibir um mesmo conjunto de dados em mais de um elemento de interface, simultaneamente
- Também é possível fazer com que o elemento de interface use os dados originais, sem copiá-los

Exemplo de Uso

- Suponha que você tem uma lista de nomes muito grande e deseja exibi-la em uma **JList**
- Usando a forma que vimos, esses nomes seriam copiados para dentro da lista
- Para evitar essa replicação, podemos utilizar um modelo próprio, que permitirá à **JList** acessar diretamente a lista de nomes

Interface **ListModel**

- Define o modelo usado pela classe **JList**
- Abrange dois aspectos:
 1. o acesso aos dados
 2. o controle da modificação dos dados

- Métodos de **ListModel**

```
int getSize()  
Object getElementAt(int index)  
void addListDataListener(ListDataListener l)  
void removeListDataListener(ListDataListener l)
```

De Volta ao Exemplo

- Imagine que os nomes estão armazenados em um *array* de **String**
- Assumindo que a lista de nomes não é modificada, podemos ignorar o *listener*
- Basta, então, definir uma classe que implemente **ListModel** e utilize o *array* como fonte dos dados

Criando um Modelo

```
class ListaDeNomes implements ListModel {
    private String[] nomes;
    ListaDeNomes(String[] nomes) {
        this.nomes = nomes;
    }
    public int getSize() {
        return nomes.length;
    }
    public Object getElementAt(int index) {
        return nomes[index];
    }
    public void addListDataListener(ListDataListener l) {}
    public void removeListDataListener(ListDataListener l) {}
}
```

Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

17

Usando o Modelo

```
JFrame f = new JFrame("Teste");
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
String[] nomes = {"a", "b", "c", "d", "e", "f"};
JList l = new JList(new ListaDeNomes(nomes));
Container cp = f.getContentPane();
cp.add(new JScrollPane(l));
f.pack();
f.setVisible(true);
```

Exercícios – Questão 24 (again) - Exemplo com DefaultListModel !

Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

18

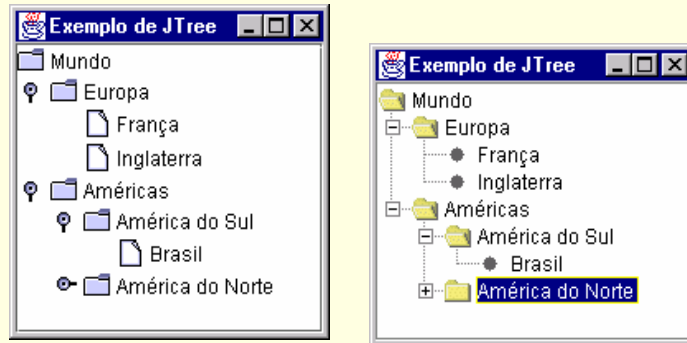
Classe JTree

- Componente que exhibe uma estrutura de dados hierárquica (árvore)
- Segue o padrão MVC: os dados a serem exibidos são obtidos de um modelo (**TreeModel**)
 - o modelo a ser utilizado é fornecido no construtor do objeto **JTree**

Terminologia

- Uma árvore é composta de nós
 - um **nó** ou é uma **folha** ou possui nós filhos
 - todo nó, com exceção da raiz, tem exatamente um nó pai
 - toda árvore tem exatamente um nó raiz
- Tipicamente, o usuário pode **expandir** ou **colapsar** nós, tornando seus filhos, respectivamente, visíveis ou invisíveis

Exemplos



Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

21

Interface **TreeModel**

- Define um modelo de dados adequado para um **JTree**
- Pertence ao pacote **javax.swing.tree**
- O Swing oferece uma implementação dessa interface: a classe **DefaultTreeModel**
 - modelo de árvore que utiliza **TreeNode**s

Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

22

Métodos de **DefaultTreeModel**

```
DefaultTreeModel(TreeNode root)

Object getRoot()
int getChildCount(Object parent)
Object getChild(Object parent, int index)

void insertNodeInto(MutableTreeNode child,
                    MutableTreeNode parent,
                    int index)
void removeNodeFromParent(MutableTreeNode node)

void addTreeModelListener(TreeModelListener l)
```

Interface **MutableTreeNode**

- É uma subinterface de **TreeNode**
- Modela um nó que pode ser modificado
 - adição/remoção de filhos
 - modificação do conteúdo armazenado no nó ("user object")
- O Swing oferece uma implementação dessa interface: a classe **DefaultMutableTreeNode**

Métodos de DefaultMutableTreeNode

```
DefaultMutableTreeNode(Object userObject)
DefaultMutableTreeNode(Object userObject,
                        boolean allowsChildren)
void add(DefaultMutableTreeNode child)
void remove(DefaultMutableTreeNode child)

Object getUserObject()
void setUserObject(Object userObject)
String toString()

boolean isLeaf()
Enumeration children()
```

Criando um JTree

```
DefaultMutableTreeNode mundo =
    new DefaultMutableTreeNode ("Mundo");
DefaultMutableTreeNode europa =
    new DefaultMutableTreeNode ("Europa");
DefaultMutableTreeNode americas =
    new DefaultMutableTreeNode ("Américas");
mundo.add(europa);
mundo.add(americas);
...
JTree arvore = new JTree(new DefaultTreeModel(mundo));
```

Modos de Seleção

- O modo de seleção de um **JTree** é configurado (e gerenciado) por um “modelo de seleção” (**TreeSelectionModel**)
- Modos disponíveis:
 - SINGLE_TREE_SELECTION
 - CONTIGUOS_TREE_SELECTION
 - DISCONTIGUOUS_TREE_SELECTION

Configurando o modo de seleção

- Configurando modo de seleção

```
JTree arvore = new JTree(raiz);
int modo = TreeSelectionModel.SINGLE_TREE_SELECTION;
TreeSelectionModel tsm = arvore.getSelectionModel();
tsm.setSelectionMode(modo);
```

- Obtendo a seleção corrente

```
TreePath path = getSelectionPath()
if (path != null) {
    DefaultMutableTreeNode selNode =
        (DefaultMutableTreeNode)path.getLastPathComponent();

    String selValue = (String)selNode.getUserObject();
    ...
}
```

Eventos de Seleção

- Eventos de seleção são gerados sempre que a seleção de uma árvore é alterada.
- Esses eventos podem ser tratados através da adição de um **TreeSelectionListener**.
- A interface **TreeSelectionListener** pertence ao pacote `javax.swing.event` e define apenas um método: `valueChanged`

Exercícios – Questão 27

Exercícios – Questão 28

Conclusões

- Componentes essenciais de uma aplicação Swing:
 - Contêineres são janelas ou painéis que contêm componentes.
 - Layouts especificam como arranjar componentes em um contêiner.
 - Componentes: são os controles da interface gráfica com o usuário.
 - Ouvintes (listeners) são conectados a componentes e contém o código que é executado quando o componente é usado.
 - É desse modo que uma ação do usuário sobre um componente é conectada a um método Java.

Conclusões

- Alguns IDEs têm eles próprios facilidades de construção da interface gráfica (editores de formulários)
 - e.g. NetBeans - www.netbeans.org
- Também há ferramentas específicas para a criação de GUIs em Java. Exemplos são:
 - XUI - <http://xui.sourceforge.net/>
 - UICompiler - <http://uic.sourceforge.net/>