

ALPOO


Teoria - Java

TableModel

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

1



Disciplina: ALPOO


Tema: Apresentação em Formato Grid - TableModel

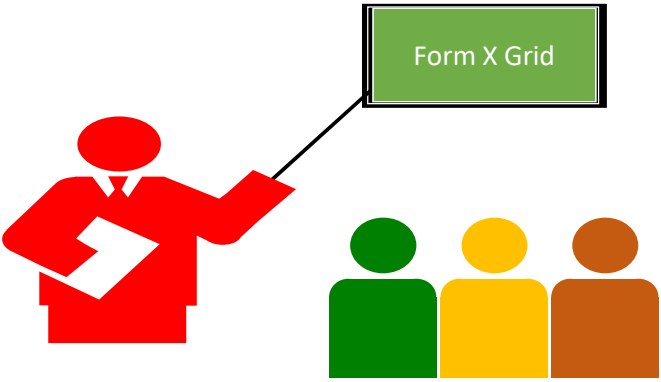
Professor: Ricardo Drudi

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

2






Form X Grid

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

3



Form X Grid

- O padrão SAA/CUA define o padrão de entrada de dados em Forms.
- A apresentação de dados em Grids deveria ser utilizada somente para exibição de informação.
- A utilização de Grids para entrada de dados deve ser utilizada criteriosamente, pois é uma interface sujeita a erros.
- O Swing permite entrada de dados em Grids, mas seu uso é desaconselhado, exceto em situações excepcionais.
- No Swing, o componente Grid é o JTable

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

4

Form

Agenda

Código:

Nome:

Celular:

Fone Residencial:

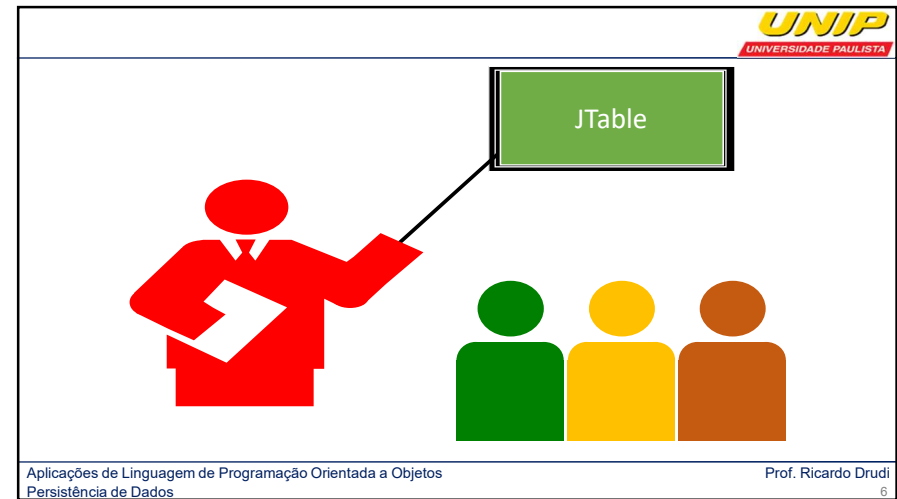
Email:

Nome	Celular	Residencial
Bill Gates	(11) 111-111-111	(11) 1111-1111
Steve Jobs	(22) 222-222-222	(22) 2222-2222
James Gosling	(33) 333-333-333	(33) 3333-3333
Richard Stallman	(44) 444-444-444	(44) 4444-4444
Linus Torvalds	(55) 555-555-555	(55) 5555-5555

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

5



JTable

- Utilizada para visualizar dados em *grid* no Swing, classes adicionais definidas em `javax.swing.table`.
- Um JTable é implementado segundo o padrão MVC.
- O Model é modelo de dados (`String[]`) que fornece os dados para exibição.
- A View implementado pela interface que desenha os dados no Grid.
- O Controller é implementado pelos Listeners que respondem às ações dos usuários.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

7

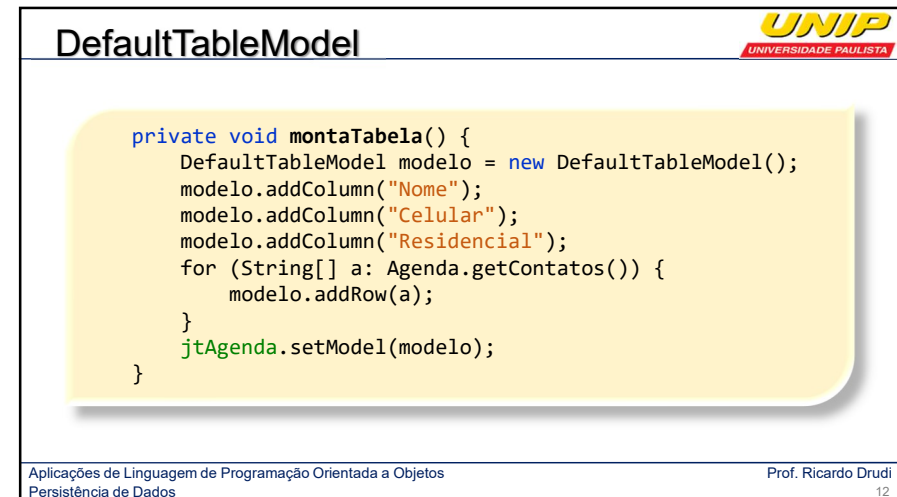
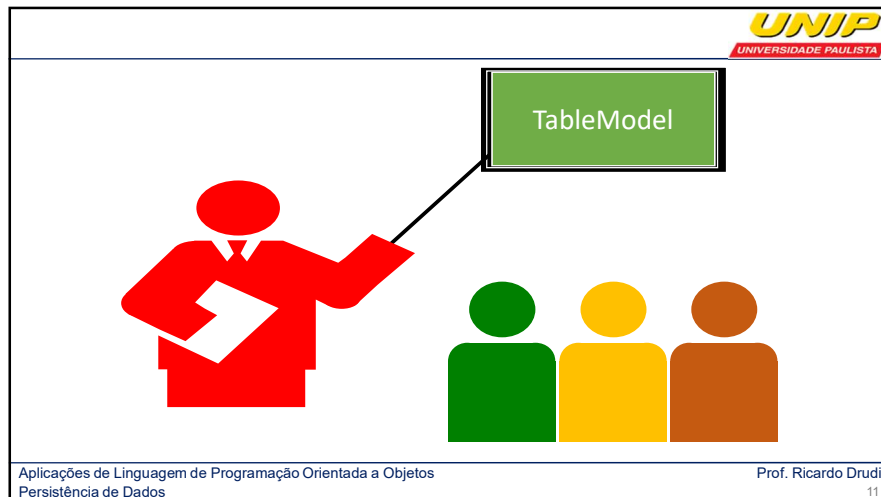
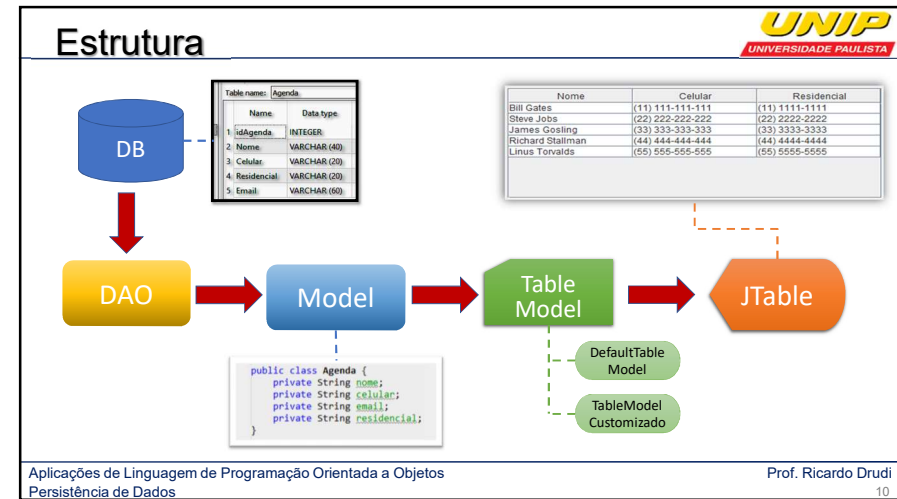
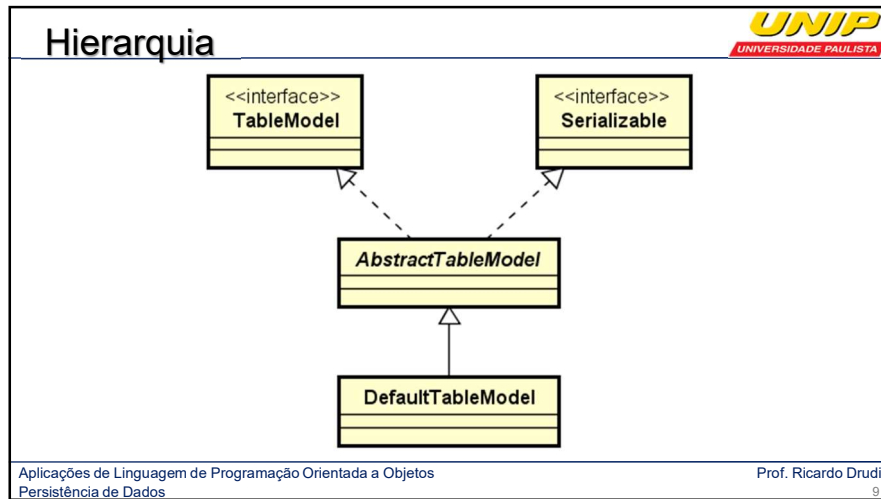
JTable

- Model
 - implementado pela interface `TableModel` (`AbstractTableModel` e `DefaultTableModel`)
 - responsável por fornecer os dados para a tabela, através do método `Object getValueAt(row, col)`
- View
 - implementado pela interface `CellRenderer`. Pode ser fornecido tanto um renderer para a tabela inteira como para uma coluna específica.
 - Para desenhar uma célula a view requisita o objeto que irá apresentar os dados pelo método `Jcomponent.getCellRendererComponent(row, col)`. O `DefaultCellRenderer` usa `JLabel` para apresentar os dados, que é a forma de apresentação mais comum para um valor.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi

8



DefaultTableModel



- Métodos da *interface* `TableModel` (`DefaultTableModel`)
 - `getValueAt(row, col)`: obtém o valor de uma determinada linha e coluna na `JTable`
 - `setValueAt(newValue, row, col)`: seta o valor em uma determinada linha e coluna na `JTable`.
 - `addColumn(Object name, Object[] row)`: adiciona uma nova coluna no modelo. Opcionalmente, os valores da coluna podem ser fornecidos.
 - `addRow(Object[] row)`: adiciona uma nova linha na `JTable`, recebe um `array` simples.
 - `removeRow(int row)`: remove linha `row` da tabela

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
13

Exemplo



```
String[] colunas = new String [] {"Estado", "Cidade"};
String[][] dados = new String [][] {
    {"SP", "São Paulo"},
    {"RJ", "Rio de Janeiro"},
    {"RN", "Rio Grande do Norte"},
    {"PR", "Paraná"}
};
// Colocamos os dados em um modelo
DefaultTableModel modelo = new DefaultTableModel(dados, colunas);
// e passamos o modelo para o JTable
JTable tabela = new JTable(modelo);
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
14

Exemplo



```
// Adiciona linha no modelo

public void adicionaLinha(JTable tabela) {
    DefaultTableModel m = (DefaultTableModel)tabela.getModel();
    m.addRow(new String [] {"SC", "Santa Catarina"} );
}
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
15

Exemplo



```
// Remove linha do modelo

public void removeLinha(JTable tabela, int row) {
    DefaultTableModel m = (DefaultTableModel)tabela.getModel();
    m.removeRow(row);
}

public void removeLinha(JTable tabela) {
    int linhaSelecionada = tabela.getSelectedRow();
    if( linhaSelecionada < 0 ) return;
    else removeLinha(tabela, linhaSelecionada);
}
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
16

Exemplo



```
// Substitui um valor por outro no modelo

public void troca(String antigo, String novo, int coluna){
    DefaultTableModel m = (DefaultTableModel)tabela.getModel();
    for (int linha=0; linha<m.getRowCount(); linha++){
        String atual = (String)m.getValueAt(linha,coluna);
        if (atual.equals(antigo)) {
            m.setValueAt(novo,linha,coluna);
        }
    }
}
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
17

Observações



- Para o padrão *TableModel*, não importa como você armazena seus dados. Para a *JTable* é indiferente se os dados vieram de uma base de dados, se foram digitados pelo usuário ou se foram criados diretamente no código.
- O importante é fornecer os valores para o modelo através dos métodos, e informar as alterações ocorridas no modelo seguindo a estrutura da arquitetura MVC.
- Inclusões, alterações e exclusões devem ser feitas no modelo, e serão automaticamente refletidas na *view*.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
18

Formatação



```
private void formataTabela(JTable tabela) {
    DefaultTableCellRenderer centro = new DefaultTableCellRenderer();
    centro.setHorizontalAlignment(SwingConstants.CENTER);
    DefaultTableCellRenderer direita = new DefaultTableCellRenderer();
    direita.setHorizontalAlignment(SwingConstants.RIGHT);
    tabela.getTableHeader().setDefaultRenderer(centro);
    tabela.getColumnModel().getColumn(0).setCellRenderer(centro);
    tabela.getColumnModel().getColumn(1).setCellRenderer(direita);
    tabela.getTableHeader().getColumnModel().getColumn(0).setMinWidth(50);
    tabela.getTableHeader().getColumnModel().getColumn(1).setMaxWidth(150);
}
```

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
19


Observações



- As formatações são configurações de exibição, e, portanto, devem ser feitas diretamente na *view (JTable)*.
- O ideal é criar uma classe utilitária com as formatações mais utilizadas, para facilitar a programação e incrementar o reuso de código.
- Como a *JTable* utiliza *JLabels* para a exibição dos dados de cada célula da tabela, todas as formatações possíveis para um *JLabel* estão disponíveis para cada célula individualmente.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
20



Muito obrigado a todos.

Copyright © 2018 Prof. Ricardo Drudi

Todos direitos reservados. A reprodução ou divulgação total ou parcial deste documento é expressamente proibida sem o consentimento formal, por escrito, do professor Ricardo Drudi.

Aplicações de Linguagem de Programação Orientada a Objetos
Persistência de Dados

Prof. Ricardo Drudi
21