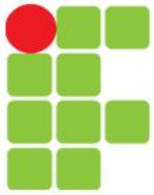




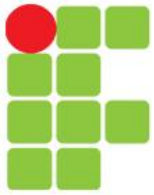
Componentes Parte 03

Prof. Edwar Saliba Júnior
Setembro de 2012



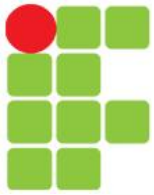
JTable

- Componente capaz de mostrar dados na tela de maneira tabular.
- Partes visuais do JTable:
 - Título: formado por um vetor de *strings*;
 - Dados: formado por uma matriz de *objects*.
- É bom saber:
 - os dados passados ao JTable ficam alojados num objeto TableModel, o qual provê métodos para facilitar o acesso a estes dados;
 - caso você não crie seu próprio “modelo de dados”, então será usado o DefaultTableModel automaticamente.



JTable

- Principais métodos:
 - Construtor: são diversos. Um dos mais simples que você pode usar é o construtor com dois parâmetros:
`JTable(dados, colunas)`, onde `dados` geralmente é uma matriz de *objects* ou *strings* e `colunas` um vetor de *strings* contendo o título das colunas;
 - `getModel()` - este método nos retorna o modelo de dados utilizado no `JTable`. Se você não criou seu próprio “modelo de dados” então o seu `JTable` está fazendo uso do `DefaultTableModel`, uma implementação de “modelo de dados” simples, através *arrays*. Ele implementa alguns métodos básicos para manipulação do modelo;
 - `getRowCount()` - retorna a quantidade de linhas existentes no `JTable`;
 - `getColumnCount()` - retorna a quantidade de colunas existentes no `JTable`;



JTable

- Principais métodos:
 - `setSelectionMode(constante)` - configura o modo de seleção de linhas para do `Jtable` de acordo com a constante escolhida. E esta poderá ser:
 - `ListSelectionMode.SINGLE_SELECTION` - permite a seleção de uma única linha;
 - `ListSelectionMode.SINGLE_INTERVAL_SELECTION` - permite a seleção de um intervalo contíguo de linhas;
 - `ListSelectionMode.MULTIPLE_INTERVAL_SELECTION` - permite a seleção de vários intervalos de linhas.

–



DefaultTableModel

- Principais métodos:
 - `getValueAt()` - obtém o valor de uma determinada célula (coluna e linha);
 - `setValueAt()` - configura o valor de uma determinada célula (coluna e linha);
 - `addRow()` - adiciona uma nova linha no modelo. Recebe um vetor;
 - `addColumn()` - adiciona uma nova coluna no modelo;
 - `removeRow()` - remove um linha do modelo;
 - `removeColumn()` - remove uma coluna do modelo;
 - etc.



Exemplo Criação de JTable

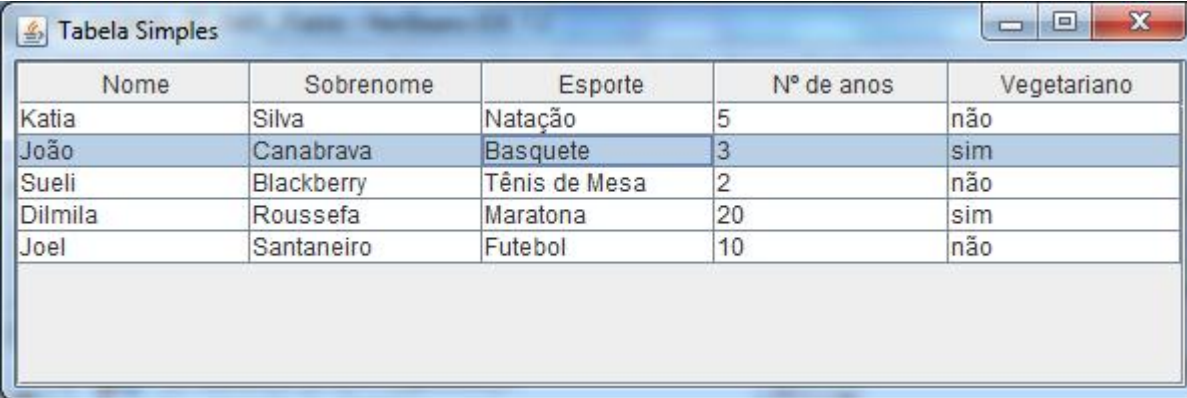
- Pode-se criar o `JTable` e deixar que ele crie o `TableModel`:

```
view plain print ?
01. String[] colunas = new String []{"Estado","Cidade"};
02. String[][] dados = new String [][] {
03.     {"SP","São Paulo"},
04.     {"RJ","Rio de Janeiro"},
05.     {"RN","Rio Grande do Norte"},
06.     {"PR","Paraná"}
07. };
08.
09. JTable jtable = new JTable(dados, colunas);
```

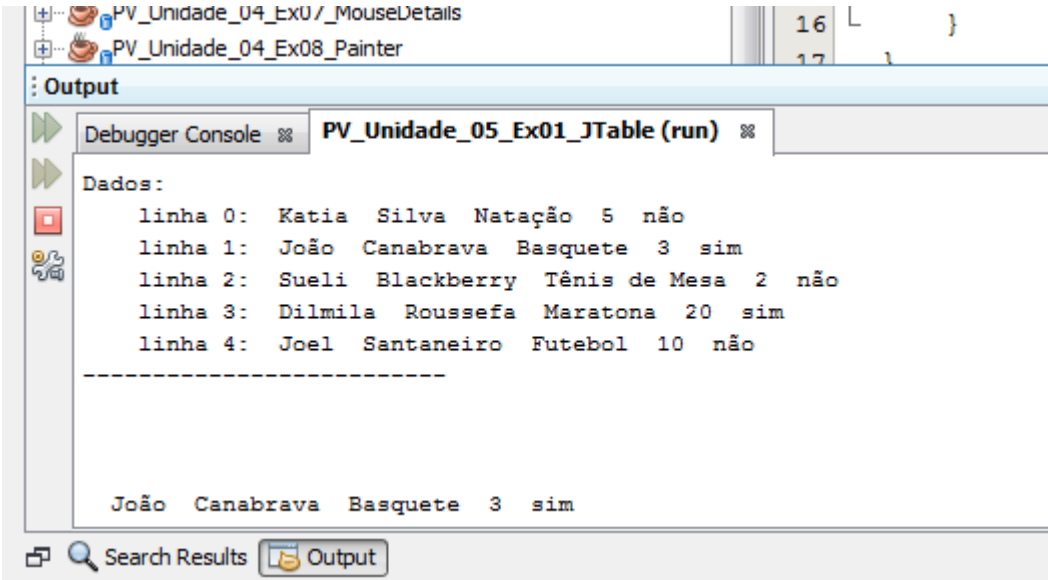
- Ou podemos criar o “modelo de dados” (`TableModel`) e passar para o `JTable`:

```
view plain print ?
01. String[][] dados = new String [][]{
02.     {"SP","São Paulo"},
03.     {"RJ","Rio de Janeiro"},
04.     {"RN","Rio Grande do Norte"},
05.     {"PR","Paraná"}
06. };
07. String[] colunas = new String []{"Estado","Cidade"};
08. // Ao invés de passar direto, colocamos os dados em um modelo
09. DefaultTableModel modelo = new DefaultTableModel(dados, colunas);
10. // e passamos o modelo para criar o jtable
11. JTable jtable = new JTable( modelo );
```

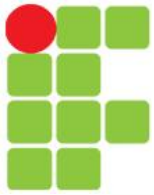
JTable - DefaultTableModel



Nome	Sobrenome	Esporte	Nº de anos	Vegetariano
Katia	Silva	Natação	5	não
João	Canabrava	Basquete	3	sim
Sueli	Blackberry	Tênis de Mesa	2	não
Dilmila	Roussefa	Maratona	20	sim
Joel	Santaneiro	Futebol	10	não



```
PV_Unidade_04_Ex07_MouseDetails
PV_Unidade_04_Ex08_Painter
Output
Debugger Console PV_Unidade_05_Ex01_JTable (run)
Dados:
linha 0: Katia Silva Natação 5 não
linha 1: João Canabrava Basquete 3 sim
linha 2: Sueli Blackberry Tênis de Mesa 2 não
linha 3: Dilmila Roussefa Maratona 20 sim
linha 4: Joel Santaneiro Futebol 10 não
-----
João Canabrava Basquete 3 sim
```



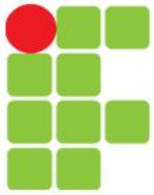
Exemplo 01 - jTable

- Parte 01 - Classe Main.
- Parte 02 - Classe SimpleTable



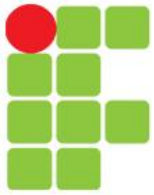
TableModel Personalizado

- Usar o `DefaultTableModel` em suas aplicações, tem suas vantagens e desvantagens:
- Vantagem:
 - Você não precisa implementar nada;
- Desvantagens:
 - Você só consegue criar tabelas com as estruturas pré-definidas no construtor da classe `JTable`;
 - o único construtor que trabalha com memória dinâmica usa a classe `Vector`, que já está ultrapassada em relação as novas estruturas da linguagem, que por sua vez, desaconselha seu uso.
- Solução:
 - Criar um `TableModel` personalizado.



TableModel Personalizado

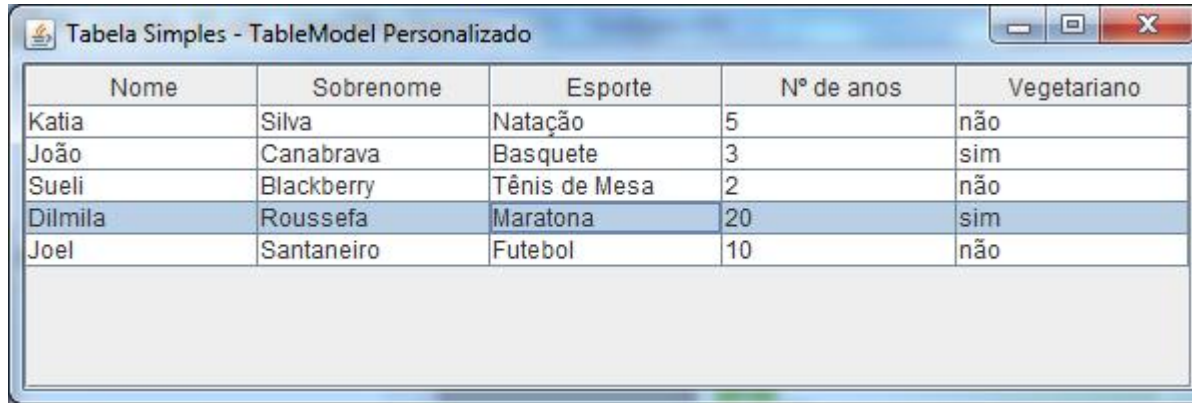
- Usar estrutura dinâmica de alocação de memória, exemplo: `ArrayList`.
- Criar métodos que ensinem o novo `TableModel` a manipular este novo tipo de estrutura.
- Criar métodos o mais genérico possível.
- Atentar-se para detalhes de edição de células que podem fazer o funcionamento do seu `TableModel Personalizado` divergir do `DefaultTableModel`.



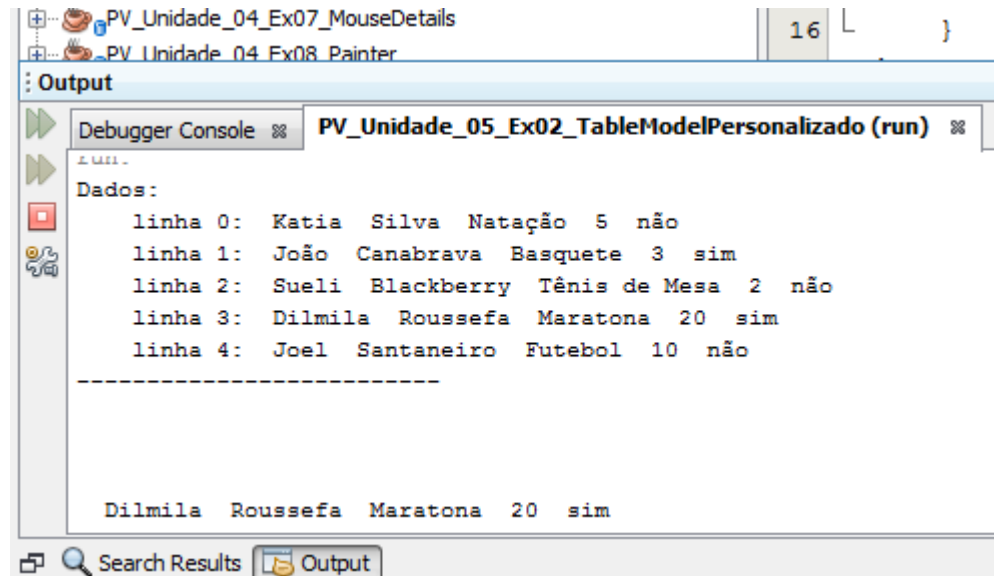
TableModel Personalizado

- Principais passos:
 - a classe deve herdar de `AbstractTableModel`;
 - deverá ter atributos para armazenar os valores de linhas e colunas;
 - o construtor deverá receber como parâmetro variáveis que inicializem estes atributos;
 - criar os métodos *get* e *set* dos atributos;
 - criar os métodos:
 - `addRow`, `removeRow` (por índice), `removeRow` (por valor e coluna).
 - sobrescrever os métodos:
 - `getRowCount`, `getColumnCount`, `getValueAt`, `setValueAt`, `isCellEditable` e `getColumnName`.

JTable - NewTableModel



Nome	Sobrenome	Esporte	Nº de anos	Vegetariano
Katia	Silva	Natação	5	não
João	Canabrava	Basquete	3	sim
Sueli	Blackberry	Tênis de Mesa	2	não
Dilmila	Roussefa	Maratona	20	sim
Joel	Santaneiro	Futebol	10	não



```
PV_Unidade_04_Ex07_MouseDetails | 16 L }
PV_Unidade_04_Ex08_Painter
Output
Debugger Console PV_Unidade_05_Ex02_TableModelPersonalizado (run)
Dados:
linha 0: Katia Silva Natação 5 não
linha 1: João Canabrava Basquete 3 sim
linha 2: Sueli Blackberry Tênis de Mesa 2 não
linha 3: Dilmila Roussefa Maratona 20 sim
linha 4: Joel Santaneiro Futebol 10 não
-----
Dilmila Roussefa Maratona 20 sim
```



Exemplo 02 - TableModelPersonalizado

- Parte 01 - Classe Main.
- Parte 02 - Classe SimpleTable
- Parte 03 - Classe NewTableModel



Bibliografia

- DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**; 6. ed., São Paulo: Pearson, 2005.
- LIMA, Bruno Rios. **Tutorial sobre jTable do Swing - Parte 01**. Disponível em: <<http://www.guj.com.br/articles/140>> Acesso em: 25 out. 2012.
- LIMA, Bruno Rios. **Tutorial sobre jTable do Swing - Parte 02**. Disponível em: <<http://www.guj.com.br/articles/147>> Acesso em: 25 out. 2012.
- ORACLE. **The Java Tutorials. How to Use Tables**. Disponível em: <<http://docs.oracle.com/javase/tutorial/uiswing/components/table.html>> Acesso em: 30 out. 2012.
- ORACLE. **A Visual Guide to Layout Managers**. Disponível em: <<http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>> Acesso em: 01 nov. 2012.