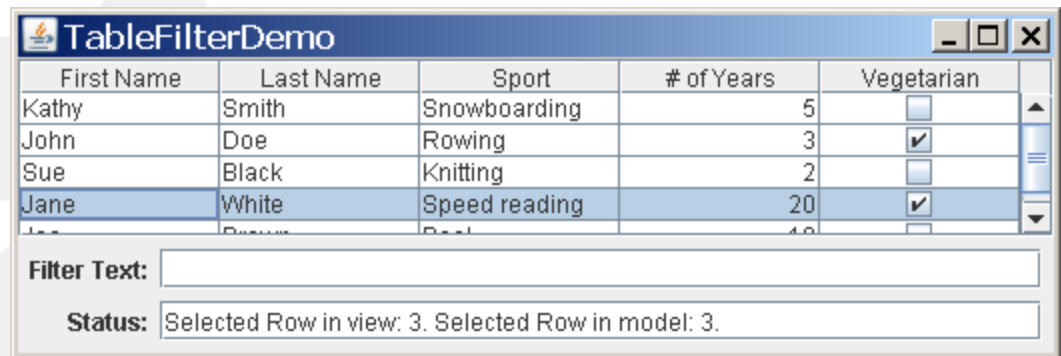
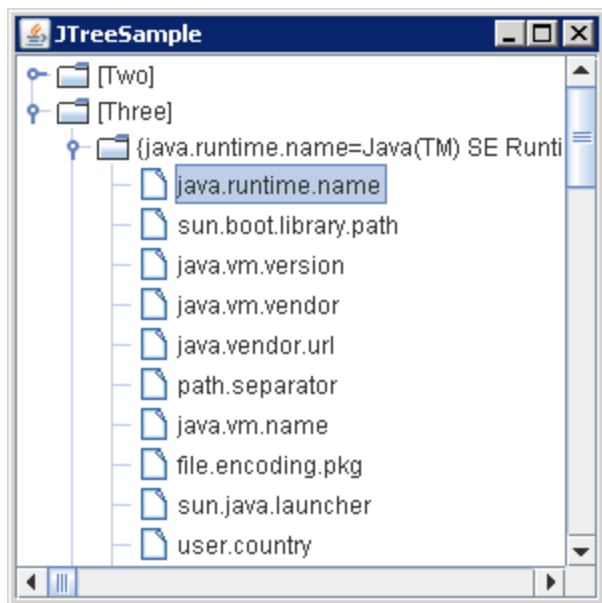


Bài 5

Điều khiển nâng cao

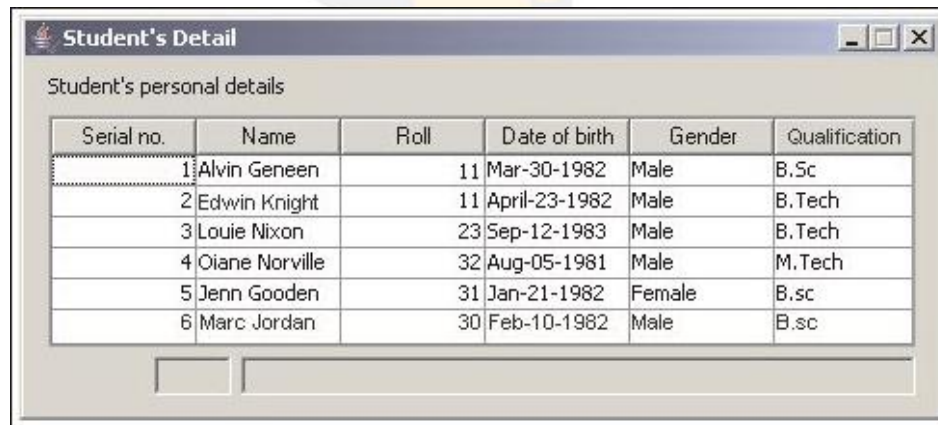
Mục tiêu

- Giới thiệu Jtable
- Giới thiệu Jtree



JTable

- Jtable là thành phần hiển thị dữ liệu theo định dạng 2 chiều (hàng và cột).
- Dữ liệu thể hiện ở dưới dạng bảng.
- Jtable không hỗ trợ cuộn, nó cần hỗ trợ của JScrollPane để có chức năng này.
- Jtable có 2 thành phần độc lập:
 - 1) **Column Header**: là dòng đơn có các cột tiêu đề
 - 2) **Data**: có 2 hướng dòng-cột cho dữ liệu.



The screenshot shows a Java Swing window titled "Student's Detail" with a standard Mac OS X-style title bar (red, yellow, green buttons and a close button). Inside the window, the text "Student's personal details" is displayed above a JTable. The table has six columns: "Serial no.", "Name", "Roll", "Date of birth", "Gender", and "Qualification". It contains six rows of student data. The first row is highlighted with a blue background. Below the table, there are two empty rectangular input fields.

Serial no.	Name	Roll	Date of birth	Gender	Qualification
1	Alvin Geneen	11	Mar-30-1982	Male	B.Sc
2	Edwin Knight	11	April-23-1982	Male	B.Tech
3	Louie Nixon	23	Sep-12-1983	Male	B.Tech
4	Oiane Norville	32	Aug-05-1981	Male	M.Tech
5	Jenn Gooden	31	Jan-21-1982	Female	B.sc
6	Marc Jordan	30	Feb-10-1982	Male	B.sc



JTable

Jtable có các constructor:

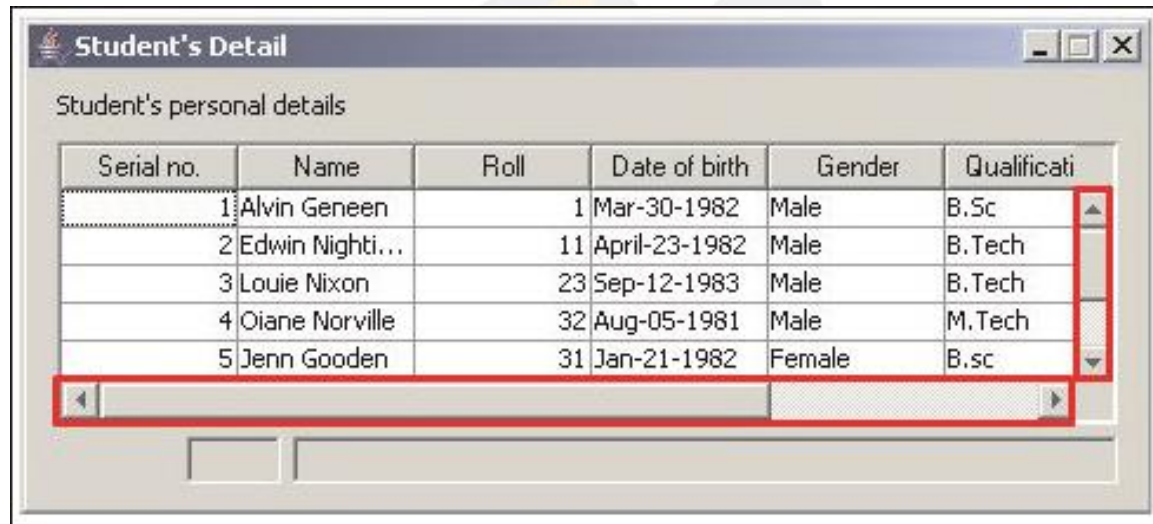
- `public JTable()`
- `public JTable(int numRows, int numColumns)`
- `public JTable(Object[][] data, Object[] columns)`
- `public JTable(Vector data, Vector columns)`
- `public JTable(TableModel model)`



JTable

Để thêm và hiển thị Jtable, thực hiện các bước sau:

- Khởi tạo đối tượng Jtable.
- Tạo **JScrollPane** và đặt **Jtable** đó như tham số.
- Thêm JScrollPane vào container.
- Jscrollpane cuộn theo chiều dọc mặc định.

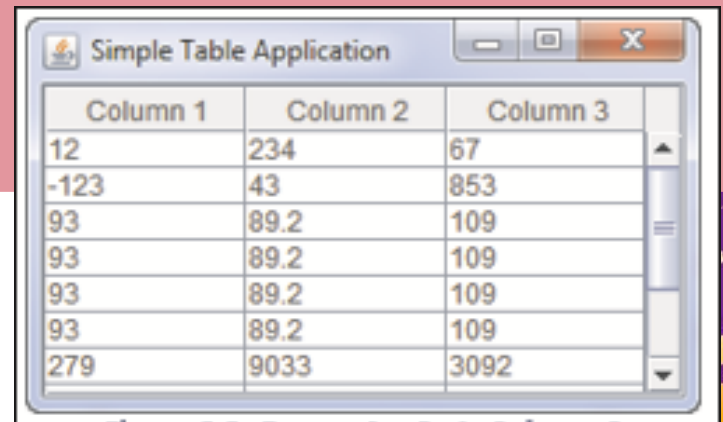


JTable

Demo:

Code Snippet

```
JFrame frmReports;  
JTable tblReports;  
Vector vecData, vecColumns;  
JScrollPane scpScroller;  
// Creates a table with the specified data and columns  
tblReports = new JTable(data, columns);  
// Provides the table to the scrollpane to facilitate scrolling  
scpScroller = new JScrollPane(tblReports);  
// Adds the scrollpane to the frame  
frmReports.getContentPane().add(scpScroller);
```



Column 1	Column 2	Column 3
12	234	67
-123	43	853
93	89.2	109
93	89.2	109
93	89.2	109
93	89.2	109
279	9033	3092

JTable

Những hạn chế chính của JTable:

1

- Mặc định, tất cả các cột có độ rộng đều nhau.

2

- Dấu 3 chấm được hiển thị nếu dữ liệu vượt quá chiều rộng cột.

3

- Cuộn ngang cột sẽ không có.

4

- Dữ liệu của Jtable tạo từ mảng hoặc Vector có hạn chế là khi có thay đổi sẽ không được cập nhật lên component.



JTable

- Để tùy chỉnh data model cho Jtable, có thể sử dụng đối tượng tạo từ class thực thi interface `javax.swing.table.TableModel`.
- Nói chung, cũng không nhất thiết phải tạo class thực thi từ interface này vì...
- Có 2 triển khai mặc định của `TableModel` là:
 - ✓ **AbstractTableModel**: Lớp abstract `AbstractTableModel` cung cấp thực thi phương thức trừu tượng của interface `TableModel`.
 - ✓ **DefaultTableModel**: Là thực hiện cụ thể của interface `TableModel`, lập trình viên có thể lưu trữ dữ liệu và cột bằng vector. Tất cả phương thức interface `TableModel` được thực thi.



AbstractTableModel Interface

- Thông thường, sẽ tạo class thực thi **AbstractTableModel** để cung cấp sự thực hiện cụ thể.
- Constructor của lớp data model nhận mảng 2 chiều là **dữ liệu** và mảng đơn chiều là **cột**.
- Để cập nhật sự thay đổi dữ liệu, phương thức **fireTableDataChanged()** được gọi.
- Các phương thức sau cần được thực thi:
 1. **public int getRowCount()** : trả về số dòng. Trong trường hợp dữ liệu mảng 2 chiều, giá trị length được trả về. Nếu dữ liệu là vector, phương thức `size()` được sử dụng trả về số phần tử.
 2. **public int getColumnCount()** : Trả về số cột. Trong trường hợp cột là mảng đơn, giá trị length được trả về.
 3. **public Object getValueAt(int row, int column)** : Trả về giá trị phần tử chứa tại vị trí dòng-cột đó.



AbstractTableModel Interface

Demo:

Code Snippet

```
import javax.swing.*;
import javax.swing.table.*;
public class ReportTable extends AbstractTableModel {
    // Arrays to hold data and columns
    String[][] data;
    String[] columns;
    ...
    // Construct to initialize the model with data and columns
    public ReportTable (String[][] data, String[] columns) {
        // Initialize the data and columns
        this.data = data;
        this.columns = columns;
    }
    // A user defined method to set new data
    // This method should be called subsequently on data change
```

AbstractTableModel Interface

```
public void setData(Object[][] data) {  
    this.data= data;  
    fireTableDataChanged();  
}  
public int getRowCount() {  
    return data.length; }  
  
public int getColumnCount() {  
    return column.length;  
}  
public Object getValueAt(int row, int column) {  
    return data(row,column);  
}
```



AbstractTableModel Interface

Khởi tạo dữ liệu cho bảng từ data model:

Code Snippet

```
ReportTable reportTable;  
String[][] data;  
String[] columns;  
// Initialize data and columns  
    String[][] data = { {...}, {...} };  
    String[] columns = { "...", "..."};  
    ...  
// Instantiate the data model reportTable with the data and columns from the arrays  
reportTable = new ReportTable(data, columns);
```



AbstractTableModel Interface

Tạo bảng từ data model:

Code Snippet

```
JTable tblReports;  
ReportTable reportTable;  
String[][] data;  
String[] columns;  
// Instantiates the table tblreports with the data model object  
tblReports = new JTable(reportTable);  
...  
// Whenever the data Changes replace the new data into the two dimensional array  
// Assigns the new data to the model by invoking setData()
```



AbstractTableModel Interface

Xử lý sự kiện click chuột:

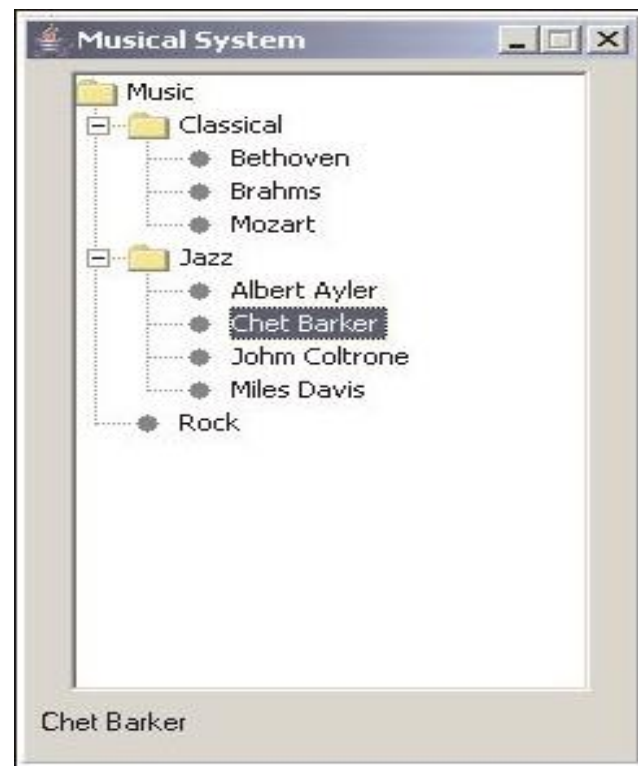
Code Snippet

```
// Register the mouse listener to the reports table  
tblReports.addMouseListener(new MouseAdapter() {  
    // Provide implementation to the mouse clicked event  
    public void mouseClicked(MouseEvent me) {  
        // Add the code  
        ...  
        ...  
    }  
});
```



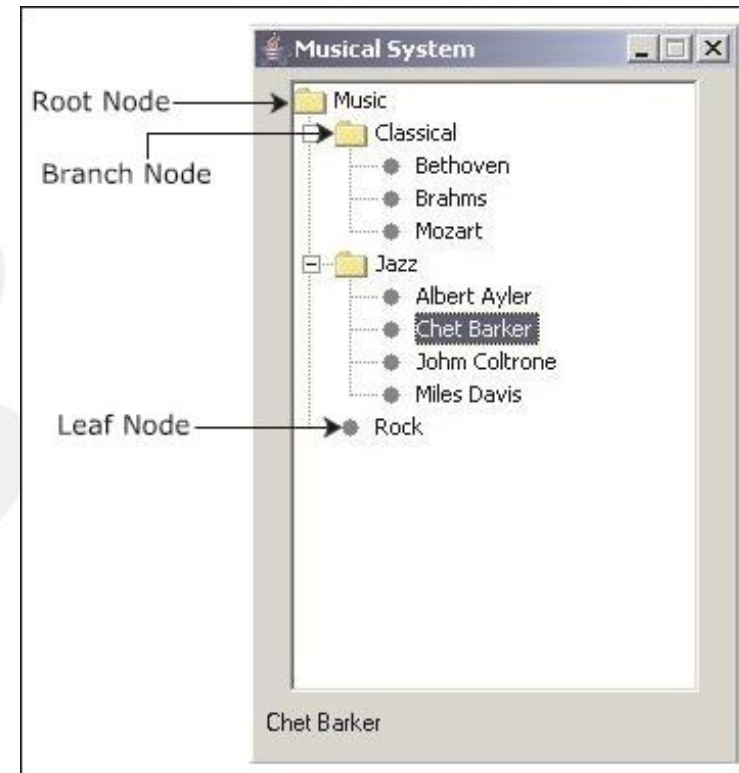
JTree

- Là thành phần hiển thị dữ liệu có **dạng thứ bậc**.
- Thường dùng biểu diễn cấu trúc phức hợp mà không thể biểu diễn dưới **dạng hàng và cột**, chỉ có thể biểu diễn **dạng hình cây (tree)**.
- Một Jtree có thể dùng biểu diễn:
 - Tổ chức **folder/tập tin** trong máy tính.
 - **Hệ thống tổ chức** (doanh nghiệp, chính phủ...)
 - **Phả hệ**



JTree

- Một Jtree hiển thị dữ liệu theo chiều dọc, mỗi dòng chứa 1 item gọi là **node**.
- Mỗi **Jtree** có một root node ở đỉnh và các node các mở xuống bên dưới.
- Mặc định các nhánh node được **đóng lại**.
- **Click** vào root node, nhánh để mở.
- Các node con có thể có các **nhánh khác** nữa.
- Node không có nhánh gọi là **leaf node** và nó ko mở rộng thêm.



JTree

Để tạo Jtree:

- Tạo thực thể DefaultMutableTreeNode.
- Tạo đối tượng Jtree.

Code Snippet

```
DefaultMutableTreeNode dmtnRoot;  
JTree treTree;  
// Creates an instance of a JTree with the root node as an  
// argument  
treTree = new JTree(dmtnRoot);
```



JTree

Step by Step:

1

- Tạo thực thể `DefaultMutableTreeNode` để hình thành các nhánh node và leaf nodes.

2

- Thêm các nhánh nodes và leaf nodes tới root node.

3

- Tạo một `JScrollPane` và gửi cấu trúc cây như là một tham số.

4

- Thêm `JScrollPane` tới container.

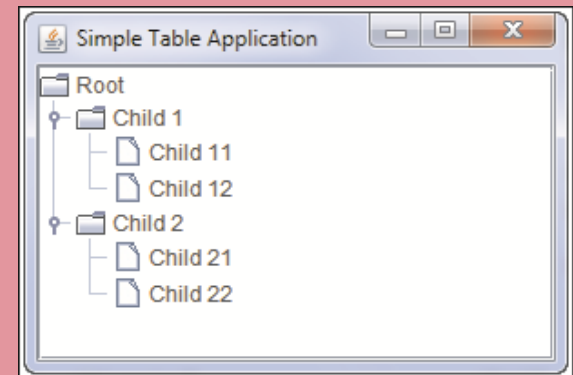


JTree

Demo:

Code Snippet

```
JFrame frmTree;  
DefaultMutableTreeNode dmtnBranch1, dmtnBranch2;  
DefaultMutableTreeNode dmtnLeaf1, dmtnLeaf2;  
DefaultMutableTreeNode dmtnRoot; JTree treTree;  
...  
// Adds the branch-nodes to the root node  
dmtnRoot.add(dmtnBranch1);  
dmtnRoot.add(dmtnBranch2);  
// Adds the leaf-nodes to the branch node  
dmtnBranch1.add(dmtnLeaf1);  
dmtnBranch2.add(dmtnLeaf2);  
// Creates a JScrollPane with the tree instance as an argument  
JScrollPane scpScroller = new JScrollPane(treTree);  
// Adds the JScrollPane to the frame  
frmTree.getContentPane().add(scpScroller);
```



JTree

Xử lý sự kiện:

Code Snippet

```
JTree treTree;  
// Registers the tree selection listener to the tree  
treTree.addTreeSelectionListener(new TreeSelectionListener() {  
    // Handles value change event  
    public void valueChanged(TreeSelectionEvent e) {  
        // Add the code to handle the event  
    }  
});  
frmTree.getContentPane().add(scroller);
```



JTree

Xử lý sự kiện:

Code Snippet

```
JTree treTree;  
...  
treTree.addTreeExpansionListener(new TreeExpansionListener() {  
    // Handles the node expansion event  
    public void treeExpanded(TreeExpansionEvent e)  
    {  
        // Action code for the event tree expanded  
        ...  
    }  
    public void treeCollapsed(TreeExpansionEvent evt)  
    {  
        JTree tree = (JTree) evt.getSource();  
        TreePath path = evt.getPath();  
        System.out.println("treeCollapsed");  
    }  
});
```



Tóm tắt bài học

- ✓ **Jtable** là thành phần hiển thị dữ liệu dạng bảng.
- ✓ Jtable thường sử dụng với **data model** để cập nhật và hiển thị dữ liệu dễ dàng.
- ✓ Có 2 loại data model mặc định là:
AbstractTableModel và **DefaultTableModel**.
- ✓ **Jtree** là thành phần hiển thị dữ liệu có cấu trúc phức **hợp dạng hình cây**.
- ✓ Jtree được dùng để hiển thị dữ liệu mà cấu trúc **hàng và cột** không thể mô tả được.
- ✓ Lớp **DefaultMutableTreeNode** được sử dụng để tạo các nhánh (**branch**), **node** và **leaf node** cho Jtree.



HẾT
XIN CẢM ƠN!

