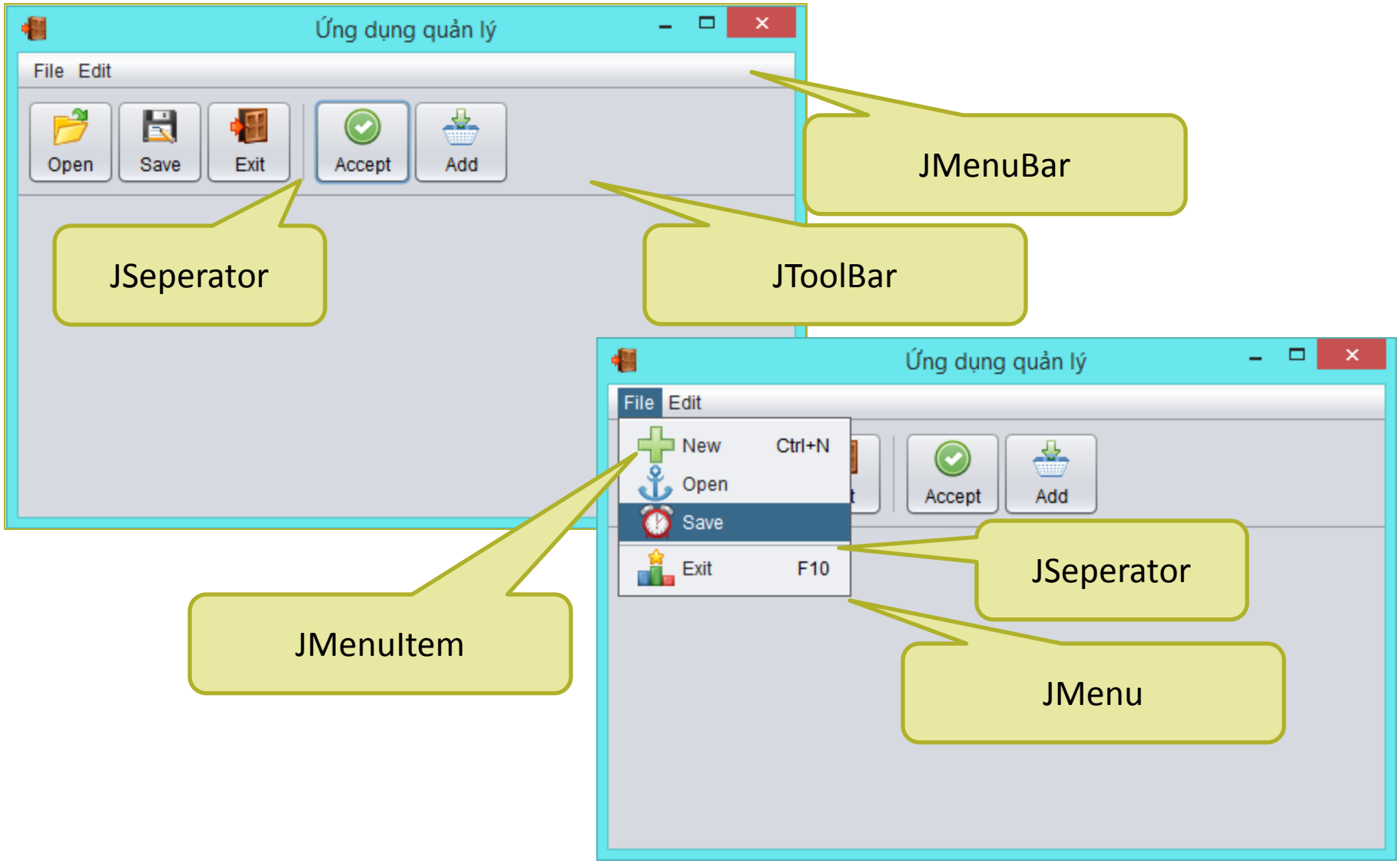
Menu và các thành phần GUI nâng cao

Nội dung bài học

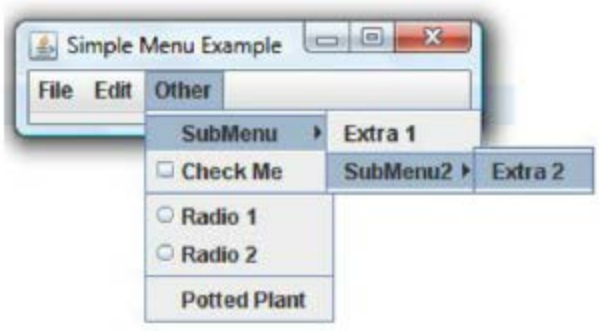
- JMenu
- JMenuBar
- JMenuItem
- JPopupMenu
- JToolBar
- JList
- JComboBox
- JTable
- JTree

MenuBar và ToolBar



JMenu

- JMenu có thể được dùng giống như một layout ,để quản lý các Component
- Chú ý:
  - Chỉ được phép có 1 lựa chọn tại 1 thời điểm
  - Icon có thể dùng để thay thế cho các menu items
  - Hầu hết các component chuẩn đều có thể là Menu Item (radio button...)
  - Có thể gán phím tắt cho các Menu Item



JMenu

- SingleSelectionModel Interface
- Có chứa 1 mảng các lựa chọn có thể và tại mỗi thời điểm chỉ được chọn duy nhất 1 lựa chọn
- SingleSelectionModel sẽ nắm giữ vị trí của chọn lựa hiện tại, nếu có sự thay đổi, ChangeEnvent sẽ được bắt
- Các thuộc tính của lớp SingleSelectionModel

Property	Data type
selected	boolean
selectedIndex	int

JMenu

- Khởi tạo
  - JMenu()
  - Khởi tạo một menu mới không có tiêu đề
  - JMenu(Action a)
  - Khởi tạo một menu mới với các thuộc tính lấy từ Action a.
  - JMenu(String s)
  - Khởi tạo một menu mới với tiêu đề là s
  - JMenu(String s, boolean b)
  - Khởi tạo một menu mới với tiêu đề là s và qui định là một menu tách rời hay không.
- Events: ChangeEvent
  - void addChangeI_istener(ChangeI_istener listener)
  - void removeChangeListener(ChangeListener listener)

JMenu

- Các phương thức

JMenuItem	add(Action a)
Component	add(Component c)
Component	add(Component c, int index)
JMenuItem	add(JMenuItem menuItem)
JMenuItem	add(String s)
void	addMenuListener(MenuListener l)
void	addSeparator()
void	applyComponentOrientation(ComponentOrientation o)
PropertyChangeListener	createActionChangeListener(JMenuItem b)
JMenuItem	createActionComponent(Action a)
JMenu.WinListener	createWinListener(JPopupMenu p)
void	doClick(int pressTime)
void	fireMenuCanceled()
	.
void	fireMenuDeselected()

JMenu

void	fireMenuSelected()
AccessibleContext	getAccessibleContext()
Component	getComponent()
	.
int	getDelay()
JMenuItem	getItem(int pos)
int	getItemCount()
Component	getMenuComponent(int n)
int	getMenuComponentCount()
Component[]	getMenuComponents()
MenuListener[]	getMenuListeners()
JPopupMenu	getPopupMenu()
Point	getPopupMenuOrigin()
MenuElement[]	getSubElements()
String	getUIClassID()

JMenu

JMenuItem	insert(Action a, int pos)
JMenuItem	insert(JMenuItem mi, int pos)
void	insert(String s, int pos)
void	insertSeparator(int index)
boolean	isMenuComponent(Component c)
boolean	isPopupMenuVisible()
boolean	isSelected()
boolean	isTearOff()
boolean	isTopLevelMenu()
void	menuSelectionChanged(boolean isIncluded)
String	paramString()
void	processKeyEvent(KeyEvent evt)
void	remove(Component c)

JMenu

void	remove(int pos)
void	remove(JMenuItem item)
void	removeAll()
void	removeMenuListener(MenuListener l)
void	setAccelerator(KeyStroke keyStroke)
void	setComponentOrientation(ComponentOrientation o)
void	setDelay(int d)
void	setMenuLocation(int x, int y)
void	setModel(ButtonModel newModel)
void	setPopupMenuVisible(boolean b)
void	setSelected(boolean b)
void	updateUI()

JMenuBar

- Dùng để tạo ra 1 Menu bar theo chiều ngang của component với 0, 1 hoặc nhiều phần tử gắn lên đó
- Bạn sẽ dùng phương thức add để thêm vào các JMenu trên JMenuBar
- JMenubar sẽ hiển thị các JMenu theo thứ tự từ trái sang phải.
- Khởi tạo
  - JMenuBar()
- Sự kiện
  - ActionListener

JMenuBar

- Các phương thức

JMenu	add(JMenu c)
void	addNotify()
AccessibleContext	getAccessibleContext()
Component	getComponent()
int	getComponentIndex(Component c)
JMenu	getHelpMenu()
Insets	getMargin()
JMenu	getMenu(int index)
int	getMenuCount()
SingleSelectionModel	getSelectionModel()
MenuElement[]	getSubElements()
MenuBarUI	getUI()
String	getUIClassID()
boolean	isBorderPainted()
boolean	isSelected()

JMenuBar

void	menuSelectionChanged(boolean isIncluded)
void	paintBorder(Graphics g)
String	paramString()
boolean	processKeyBinding(KeyStroke ks, KeyEvent e, int condition, boolean pressed)
void	processKeyEvent(KeyEvent e, MenuElement[] path, MenuSelectionManager manager)
void	processMouseEvent(MouseEvent event, MenuElement[] path, MenuSelectionManager manager)
void	removeNotify()
void	setBorderPainted(boolean b)
void	setHelpMenu(JMenu menu)
void	setMargin(Insets m)
void	setSelected(Component sel)
void	setSelectionModel(SingleSelectionModel model)
void	setUI(MenuBarUI ui)
void	updateUI()

JMenuBar

- Bạn có thể gắn 1 menu bar lên frame theo 1 trong 2 cách sau:
- Sử dụng setJMenuBar()
  - JFrame frame = new JFrame;
  - JMenuBar menuBar = new JMenuBar();
    frame.setJMenuBar(menuBar);
- Dùng layout để định vị
  - menuBar.setBorder(new BevelBorder(BevelBorder.RAISED));
    frame.getContentPane( ).add(menuBar,
    BorderLayout.SOUTH);

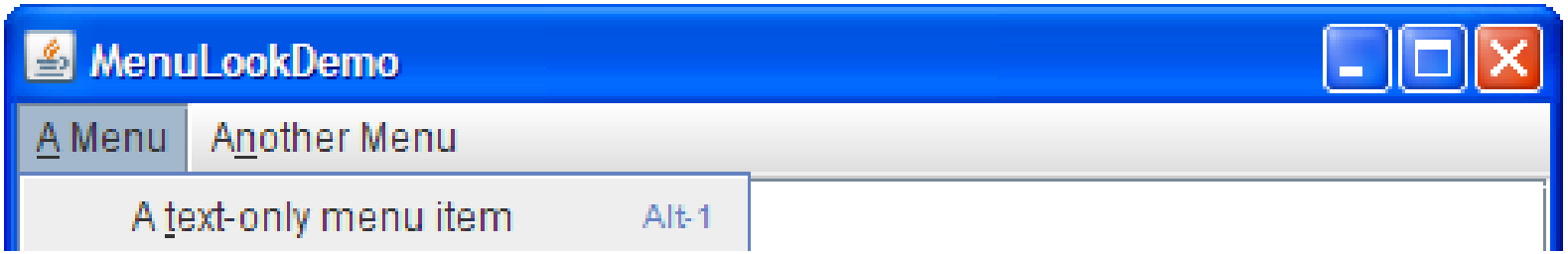
JMenuBar

- Ví dụ: Tạo 2 menu "A menu" và "Another Menu" trên JMenuBar và gán phím tắt

```
1   //Where the GUI is created:
2   JMenuBar menuBar;
3   JMenu menu, submenu;
4
5   //Create the menu bar.
6   menuBar = new JMenuBar();
7
8   //Build the first menu.
9   menu = new JMenu("A Menu");
10  menu.setMnemonic(KeyEvent.VK_A);
11  menu.getAccessibleContext().setAccessibleDescription(
12          "The only menu in this program that has menu items");
13  menuBar.add(menu);
14
15  //Build second menu in the menu bar.
16  menu = new JMenu("Another Menu");
17  menu.setMnemonic(KeyEvent.VK_N);
18  menu.getAccessibleContext().setAccessibleDescription(
19          "This menu does nothing");
20  menuBar.add(menu);
21
22  ...
23  frame.setJMenuBar(theJMenuBar);
24
```

JMenuBar

- Ví dụ: Tạo 2 menu “A menu” và “Another Menu” trên JMenuBar và gán phím tắt



JMenuItem

- Cung cấp nhiều lớp bao dùng đề chuyền 1 string hay 1 icon thành 1 JMenuItem
- JMenuItem là 1 loại nút đặc biệt(xử lý mouseListenner)
- Khởi tạo
  - JMenuItem()
    - Khởi tạo một JMenuItem không có tiêu đề và icon.
  - JMenuItem(Action a)
    - Khởi tạo một menu item với thuộc tính là a.
  - JMenuItem(Icon icon)
    - Khởi tạo một JMenuItem với icon.
  - JMenuItem(String text)
    - Khởi tạo một JMenuItem với tiêu đề là text.
  - JMenuItem(String text, Icon icon)
    - Khởi tạo một JMenuItem với tiêu đề và icon.
  - JMenuItem(String text, int mnemonic)
    - Khởi tạo một JMenuItem với tiêu đề và phím tắt

JMenuItem

- Các phương thức

void	actionPropertyChanged(Action action, String propertyName)
void	addMenuDragMouseListener(MenuDragMouseListener l)
void	addMenuKeyListener(MenuKeyListener l)
void	configurePropertiesFromAction(Action a)
void	fireMenuDragMouseDragged(MenuDragMouseEvent event)
void	fireMenuDragMouseEntered(MenuDragMouseEvent event)
void	fireMenuDragMouseExited(MenuDragMouseEvent event)
void	fireMenuDragMouseReleased(MenuDragMouseEvent event)

JMenuItem

- Các phương thức

void	fireMenuKeyPressed(MenuKeyEvent event)
void	fireMenuKeyReleased(MenuKeyEvent event)
void	fireMenuKeyTyped(MenuKeyEvent event)
KeyStroke	getAccelerator()
AccessibleContext	getAccessibleContext()
Component	getComponent()
MenuDragMouseListener[]	getMenuDragMouseListeners()

JMenuItem

MenuKeyListener[]	getMenuKeyListeners()
MenuElement[]	getSubElements()
String	getUIClassID()
void	init(String text, Icon icon)
boolean	isArmed()
void	menuSelectionChanged(boolean isIncluded)
String	paramString()
void	processKeyEvent(KeyEvent e, MenuElement[] path, MenuSelectionManager manager)
void	processMenuDragMouseEvent(MenuDragMouseEvent e)
void	processMenuKeyEvent(MenuKeyEvent e)
void	processMouseEvent(MouseEvent e, MenuElement[] path, MenuSelectionManager manager)

JMenuItem

void	removeMenuDragMouseListener(MenuDragMouseListener l)
void	removeMenuKeyListener(MenuKeyListener l)
void	setAccelerator(KeyStroke keyStroke)
void	setArmed(boolean b)
void	setEnabled(boolean b)
void	setModel(ButtonModel newModel)
void	setUI(MenuItemUI ui)
void	updateUI()

JMenuItem

Ví dụ:

```
JMenu fileMenu = new JMenu("File");

JMenuItem newMenuItem = new JMenuItem("New");
fileMenu.add(newMenuItem);

JMenuItem closeMenuItem = new JMenuItem("Close");
fileMenu.add(closeMenuItem);
fileMenu.addSeparator();
```

JPopupMenu

- Là một loại menu đặc biệt mà không cần phải gắn vào menu bar
- Nó có thể được hiển thị ở bất cứ vị trí nào trên thành phần chứa.
- Bạn có thể thêm, chèn một JMenuItem, 1 component hay 1 Action tuy ý vào popup menu này với phương thức add() và insert()
- JPopupMenu sẽ gán cho mỗi menu item 1 số thứ tự rồi gắn chúng vào popup menu theo layout mà nó có
- Bạn cũng có thể thêm 1 separator vào popup menu với phương thức addSeparator()

JPopupMenu

Khởi tạo

public JPopupMenu()

public JPopupMenu(String title)

public JMenuItem add(JMenuItem menuItem)

public Component add(Component c)

public JMenuItem add(Action a)

public JMenuItem insert(Action a, int index)

public Component insert(Component component, int index)

public void addSeparator()

JPopupMenu

- Để hiển thị popup menu dùng phương thức show()

```
public void processMouseEvent(MouseEvent e) {
        if (e.isPopupTrigger()) {
                popup.show(this, e.getX(), e.getY());
        } else {
                super.processMouseEvent(e);
        }
```

JPopupMenu

- Events: PopupMenuEvent.
  - Dùng để báo cho các listener về việc ẩn, hiện hay hủy bỏ popup menu
- Khởi tạo
  - public PopupMenuEvent(Object source)
- Sử dụng PopupMenuEvent trong 2 trường hợp sau:
  - Menu hiện <=> ẩn
  - Loại bỏ 1 lựa chọn
- public void addPopupMenuListener(PopupMenuListener I)
- public void removePopupMenuListener(PopupMenuListener I)

JPopupMenu

- public void addPopupMenuListener(PopupMenuListener I)
- public void removePopupMenuListener(PopupMenuListener I)
- PopupMenuListener Interface
  - Là nơi tiếp nhận PopupMenuEvent
  - public abstract void
- popupMenuCanceled(PopupMenuEvent e)
  - public abstract void popupMenuWillBecomeInvisible(PopupMenuEvent e)
  - public abstract void popupMenuWillBecomeVisible(PopupMenuEvent e)

JPopupMenu

- Ví dụ:

```
1   //...where instance variables are declared:
2   JPopupMenu popup;
3       //...where the GUI is constructed:
4       //Create the popup menu.
5       popup = new JPopupMenu();
6       menuItem = new JMenuItem("A popup menu item");
7       menuItem.addActionListener(this);
8       popup.add(menuItem);
9       menuItem = new JMenuItem("Another popup menu item");
10      menuItem.addActionListener(this);
11      popup.add(menuItem);
12      //Add listener to components that can bring up popup menus.
13      MouseListener popupListener = new PopupListener();
14      output.addMouseListener(popupListener);
15      menuBar.addMouseListener(popupListener);
16  ...
17  class PopupListener extends MouseAdapter {
18      public void mousePressed(MouseEvent e) {
19          maybeShowPopup(e);
20      }
21      public void mouseReleased(MouseEvent e) {
22          maybeShowPopup(e);
23      }
24      private void maybeShowPopup(MouseEvent e) {
25          if (e.isPopupTrigger()) {
26              popup.show(e.getComponent(),
27                         e.getX(), e.getY());
28          }
29      }
30  }
```

JToolBar

- Là lớp chứa cho nhiều thành phần khác
- Khi 1 component được gắn vào tool bar nó sẽ được định vị từ trái sang phải theo chỉ mục của nó
- Khởi tạo
  - JToolBar()
    - Khởi tạo một tool bar; Chiều mặc định nằm ngang.
  - JToolBar(int orientation)
    - Khởi tạo một tool bar với chiều ngang/dọc.
  - JToolBar(String name)
    - Khởi tạo một tool bar với tên gọi name.
  - JToolBar(String name, int orientation)
- Sự kiện
  - Dùng Sự kiện PropertyChangeEvent khi có 1 sự thay đổi giá trị của các thuộc tính

JToolBar

- Các phương thức

JButton	add(Action a)
protected void	addImpl(Component comp, Object constraints, int index)
void	addSeparator()
void	addSeparator(Dimension size)
protected PropertyChangeListener	createActionChangeListener(JButton b)
protected JButton	createActionComponent(Action a)
AccessibleContext	getAccessibleContext()
Component	getComponentAtIndex(int i)
int	getComponentIndex(Component c)
Insets	getMargin()
int	getOrientation()
ToolBarUI	getUI()
String	getUIClassID()
boolean	isBorderPainted()
boolean	isFloatable()

JToolBar

- Các phương thức

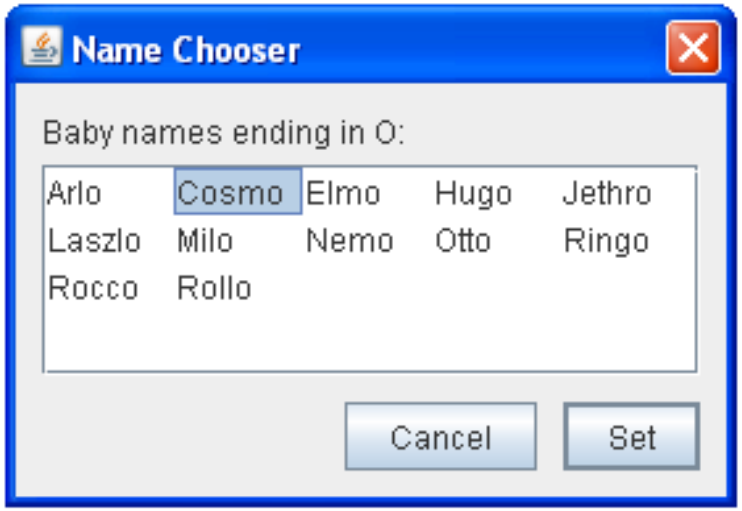
boolean	isRollover()
protected void	paintBorder(Graphics g)
protected String	paramString()
void	setBorderPainted(boolean b)
void	setFloatable(boolean b)
void	setLayout(LayoutManager mgr)
void	setMargin(Insets m)
void	setOrientation(int o)
void	setRollover(boolean rollover)
void	setUI(ToolBarUI ui)
void	updateUI()

Danh sách

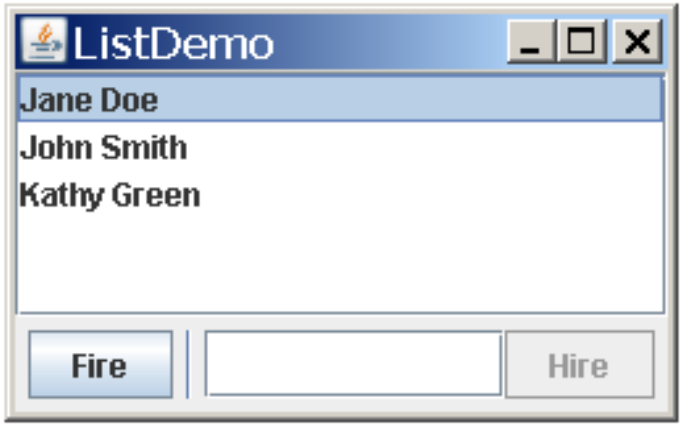
- Thường dùng để liệt kê
- Java co 2 loại danh sách cơ bản:
  - JList: dùng cho danh sách có các khoản mục cố định Người sử dụng có thể chọn một hoặc nhiều mục
  - JComboBox: dùng cho danh sách mà người dùng có thể lựa chọn mục nhập

JList

- Một JList trình bày cho người dùng một nhóm các item, thể hiện trên một hoặc nhiều cột để lựa chọn.
- JList thường có nhiều item, vì vậy chúng được đặt trong một ScrollPane



ListDialog



ListDemo

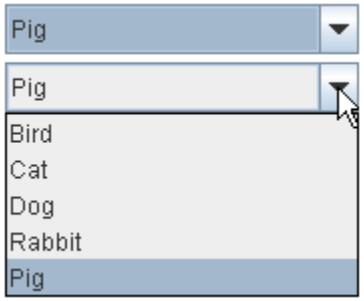
JList

- Khởi tạo
  - JList()
    - Khởi tạo một JList rỗng,chỉ đọc.
  - JList(E[] listData)
    - Khởi tạo một JList hiển thị các phần tử trong mảng E.
- Ví dụ:

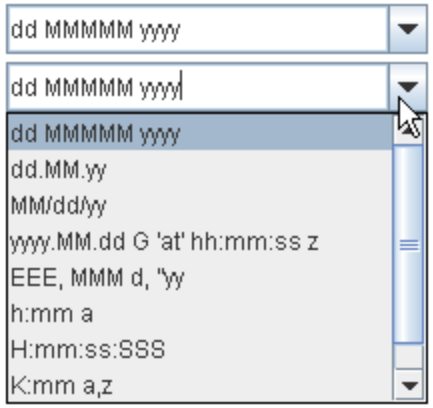
```
String[] selections = { "green", "red", "orange", "dark blue" };
    JList list = new JList(selections);
    list.setSelectedIndex(1);
    System.out.println(list.getSelectedValue());
    frame.add(new JScrollPane(list));
```

JComboBox

- Là thành phần cho phép người dùng chọn một item từ một danh sách item, có 2 loại JComboBox
- JComboBox mặc định là dạng không chỉnh sửa được, người dùng chỉ được phép chọn từ danh sách



- Loại thứ 2 là JComboBox chỉnh sửa được, người dùng gõ vào vùng text hoặc chọn từ danh sách



JComboBox

- Khởi tạo
  - JComboBox()
    - Tạo một ComboBox với dữ liệu mặc định
  - JComboBox(E[] items)
    - Tạo một ComboBox chứa các item trong mảng E
- Ví dụ:

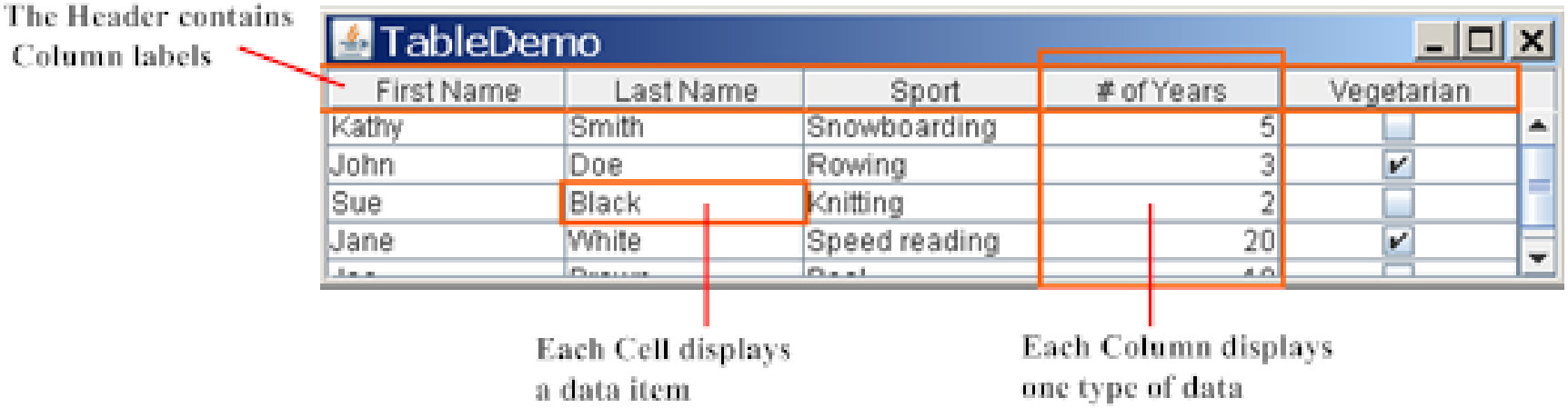
```
JComboBox combo = new JComboBox();
combo.addItem("A");
combo.addItem("H");
combo.addItem("P");
combo.setEditable(true);
System.out.println("#items=" + combo.getItemCount());

combo.addActionListener(this);

getContentPane().add(combo);
```

JTable

- Là thành phần để hiển thị các bảng dữ liệu, (tùy chọn) cho người dùng chỉnh sửa dữ liệu



JTable

- Khởi tạo
  - JTable()
    - Khởi tạo một bảng kích thước 1x1
  - JTable(int numRows, int numColumns)
    - Khởi tạo một bảng với kích thước numRows x numColumns
  - JTable(Object[][] rowData, Object[] columnNames)
    - Khởi tạo một mảng 2 chiều (rowData) với tên các cột (columnNames)

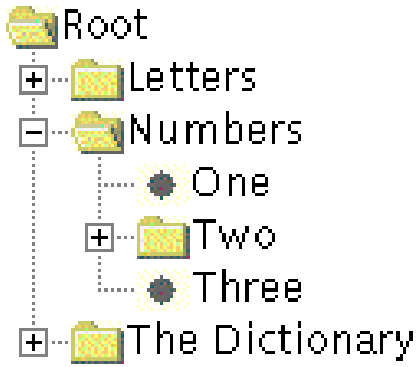
JTable

- Ví dụ

```
1  Object[][] data = {
2      {"Kathy", "Smith",
3       "Snowboarding", new Integer(5), new Boolean(false)},
4      {"John", "Doe",
5       "Rowing", new Integer(3), new Boolean(true)},
6      {"Sue", "Black",
7       "Knitting", new Integer(2), new Boolean(false)},
8      {"Jane", "White",
9       "Speed reading", new Integer(20), new Boolean(true)},
10     {"Joe", "Brown",
11      "Pool", new Integer(10), new Boolean(false)}
12 };
13 Object columnNames = {"First Name",
14                        "Last Name",
15                        "Sport",
16                        "# of Years",
17                        "Vegetarian"};
18 JTable table = new JTable(data, columnNames);
19 JScrollPane scrollPane = new JScrollPane(table);
20 frame.add(scrollPane, BorderLayout.CENTER);
```

JTree

- Với lớp JTree, bạn có thể hiển thị dữ liệu phân cấp.
- Một đối tượng JTree không thực sự chứa dữ liệu của bạn; nó chỉ đơn giản là cung cấp một cái nhìn của các dữ liệu.



- Như hình vẽ, JTree hiển thị dữ liệu theo chiều dọc. Mỗi hàng được hiển thị bởi các cây có chứa chính xác một mục dữ liệu, được gọi là một nút. Mỗi cây có một nút gốc nút gốc duy nhất.

JTree

- Một nút có thể hoặc là có con hay không. Các nút có nút con gọi là các nút nhánh. Các nút mà không có nút con gọi là các nút lá.
- Một nút nhánh có thể có nhiều nút con. Để xem các nút con của một nhánh, ta bấm vào dấu "mở rộng". Một chương trình có thể phát hiện những thay đổi trong trạng thái mở rộng các nút nhánh
- Một nút cụ thể trong một cây được xác định, hoặc bởi một TreePath, một đối tượng mà đóng gói một nút, hoặc bởi hàng đặc trưng của nó, trong đó mỗi hàng trong khu vực hiển thị sẽ hiển thị một nút

JTree

- Khởi tạo
  - JTree(Object[] value)
    - Khởi tạo một JTree với các thành phần của mảng Object là nút con, nút gốc chưa được xác định
  - JTree(TreeNode root)
    - Khởi tạo một JTree với nút gốc là root

JTree

- Ví dụ: Tạo một nút gốc, khởi tạo JTree từ nút gốc vừa tạo và đặt Tree vào một Scroll Pane

```
//Where instance variables are declared:
private JTree tree;
...
public TreeDemo() {
    ...
    DefaultMutableTreeNode top =
        new DefaultMutableTreeNode("The Java Series");
    createNodes(top);
    tree = new JTree(top);
    ...
    JScrollPane treeView = new JScrollPane(tree);
    ...
}
```

JTree

```
//Add node con của root
DefaultMutableTreeNode category =
    new DefaultMutableTreeNode("Books for Java Programmers");
top.add(category);


DefaultMutableTreeNode book = null;

//Add node con cho node nhánh catagory
book = new DefaultMutableTreeNode(new BookInfo
    ("The Java Tutorial Continued: The Rest of the JDK",
    "tutorialcont.html"));
category.add(book);
book = new DefaultMutableTreeNode(new BookInfo
    ("The Swing Tutorial: A Guide to Constructing GUIs",
    "swingtutorial.html"));
category.add(book);
```

JTree

- Một số Phương thức
  - node.add()
  - node.remove()
  - node.removeAllChildren()
  - node.getChildCount()
  - node.getChildAt()
  - node.getParent()
  - node.setParent()
  - node.removeFromParent()
  - node.getUserObject()
  - node.setUserObject()
  - model.getRoot()
  - model.setRoot()
- Sự kiện
  - TreeExpansion, TreeSelection, TreeWillExpand