



**Java™**

# **LẬP TRÌNH JAVA 2**

## **BÀI 2: SWING ADVANCED**

### **PHẦN 1**

🎯 Kết thúc bài học này bạn có khả năng

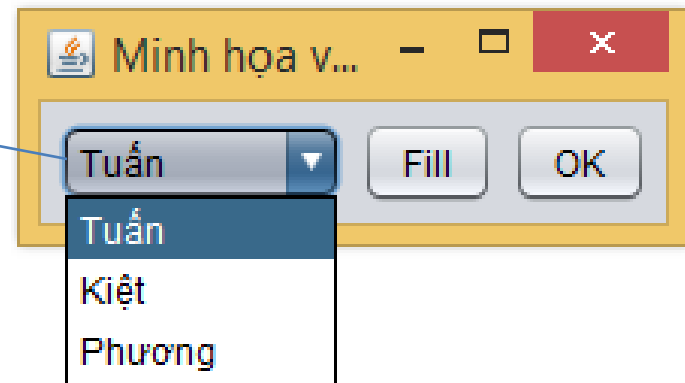
- ❖ Jcombobox, Jlist
- ❖ Jtable
- ❖ Jtree
- ❖ Timer - ProgressBar
- ❖ JSlider



❑ JComboBox được sử dụng để tạo danh sách xổ xuống và chỉ cho phép chọn 1. Các thuộc tính thường sử dụng.

- ❖ Model: chứa danh sách dữ liệu
- ❖ SelectedIndex: vị trí mục được chọn
- ❖ SelectedItem: dữ liệu mục chọn
- ❖ Name: Tên, nên bắt đầu bởi cbo
- ❖ Ví dụ:

➤ cboNames



- ❑ Đổ dữ liệu vào JComboBox bạn sử dụng đoạn mã sau đây

```
String[] data = {  
    "Việt Nam", "United States", "Singapore", "Malaysia"  
};  
cboCountry.setModel(new DefaultComboBoxModel(data));
```

- ❑ Làm việc với mục chọn

```
cboCountry.setSelectedIndex(1)  
int index = cboCountry.getSelectedIndex()  
String country = (String)cboCountry.getSelectedItem()
```

- ❑ Các phương thức sau đây giúp bạn làm việc với các mục trong JComboBox
  - ❖ `cboCountry.getItemCount()`
  - ❖ `cboCountry.getItemAt(index)`
  - ❖ `cboCountry.addItem(item)`
  - ❖ `cboCountry.removeItem(item)`
  - ❖ `cboCountry.removeItemAt(index)`
  - ❖ `cboCountry.removeAllItems()`

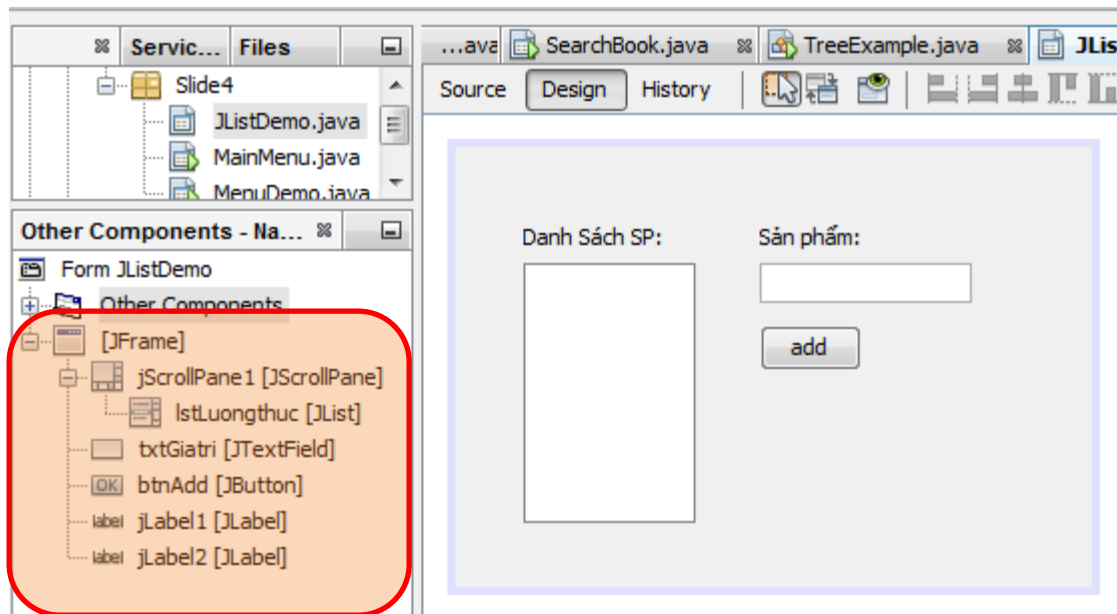


# DEMO

Chạy và giải thích



- ❑ Đối tượng dùng để hiển thị danh sách, người sử dụng có thể chọn một hoặc nhiều mục
- ❑ 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.



```
public class JListDemo extends javax.swing.JFrame {
    DefaultListModel model;
    public JListDemo() {
        initComponents();
        model = new DefaultListModel();
        String items[]={"Gạo","Cá","Đường","Muối"};
        for(String item : items){
            model.addElement(item);
        }
        lstLuongthuc.setModel(model);
    }
    //.....
}
```

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    model.addElement(txtGiatri.getText());
}
private void lstLuongthucMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    txtGiatri.setText(lstLuongthuc.getSelectedValue());
}
```



JList - Fpoly

Danh Sách SP: Sản phẩm:

Gạo

Cá

Đường

Muối

Gạo

add

JList - Fpoly

Danh Sách SP: Sản phẩm:

Gạo

Cá

Đường

Muối

Tiêu

add

JList - Fpoly

Danh Sách SP: Sản phẩm:

Gạo

Cá

Đường

Muối

Tiêu

Tiêu

add



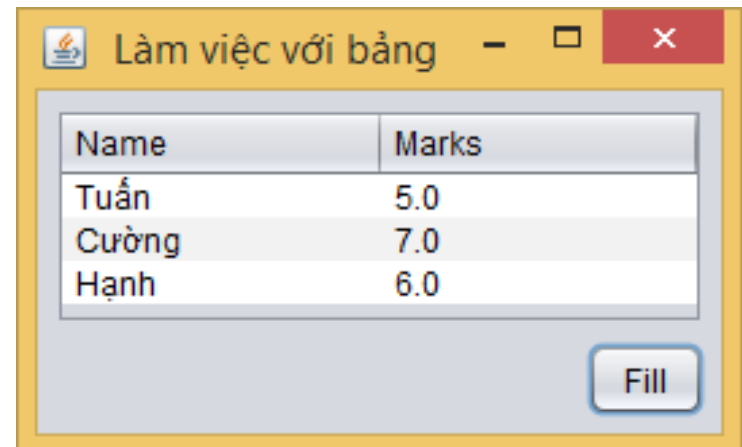
# DEMO

Chạy và giải thích



❑ JTable được sử dụng để biểu diễn dữ liệu dạng bảng. Các thuộc tính thường dùng

- ❖ Model: dữ liệu bảng
- ❖ SelectionMode: chế độ chọn các hàng
- ❖ RowHeight: chiều cao mỗi hàng
- ❖ Name: Tên, nên bắt đầu bởi tbl
- ❖ Ví dụ:
  - tblStudents



❑ Bạn có thể đổ dữ liệu vào bảng bằng đoạn mã sau:

```
String [] columns = {  
    "NO", "FULLNAME", "EMAIL", "COUNTRY", "GENDER", "STATUS"  
}  
  
Object[][] rows = {  
    {1, "Nguyễn Chí Phèo", "PheoNC@gmail.com", true, false},  
    {2, "Phạm Thị Nở", "NoPT@gmail.com", false, true}  
}  
  
tblStaffList.setModel(new DefaultTableModel(rows, columns))
```

## ❑ Lập trình lấy vị trí các hàng được chọn

- ❖ `int index = tblStaffList.getSelectedRow();`
  - Vị trí hàng được chọn đầu tiên
- ❖ `int count = tblStaffList.getSelectedRowCount();`
  - Số hàng được chọn
- ❖ `int[] indices = tblStaffList.getSelectedRows();`
  - Vị trí các hàng được chọn

## ❑ Lấy Model

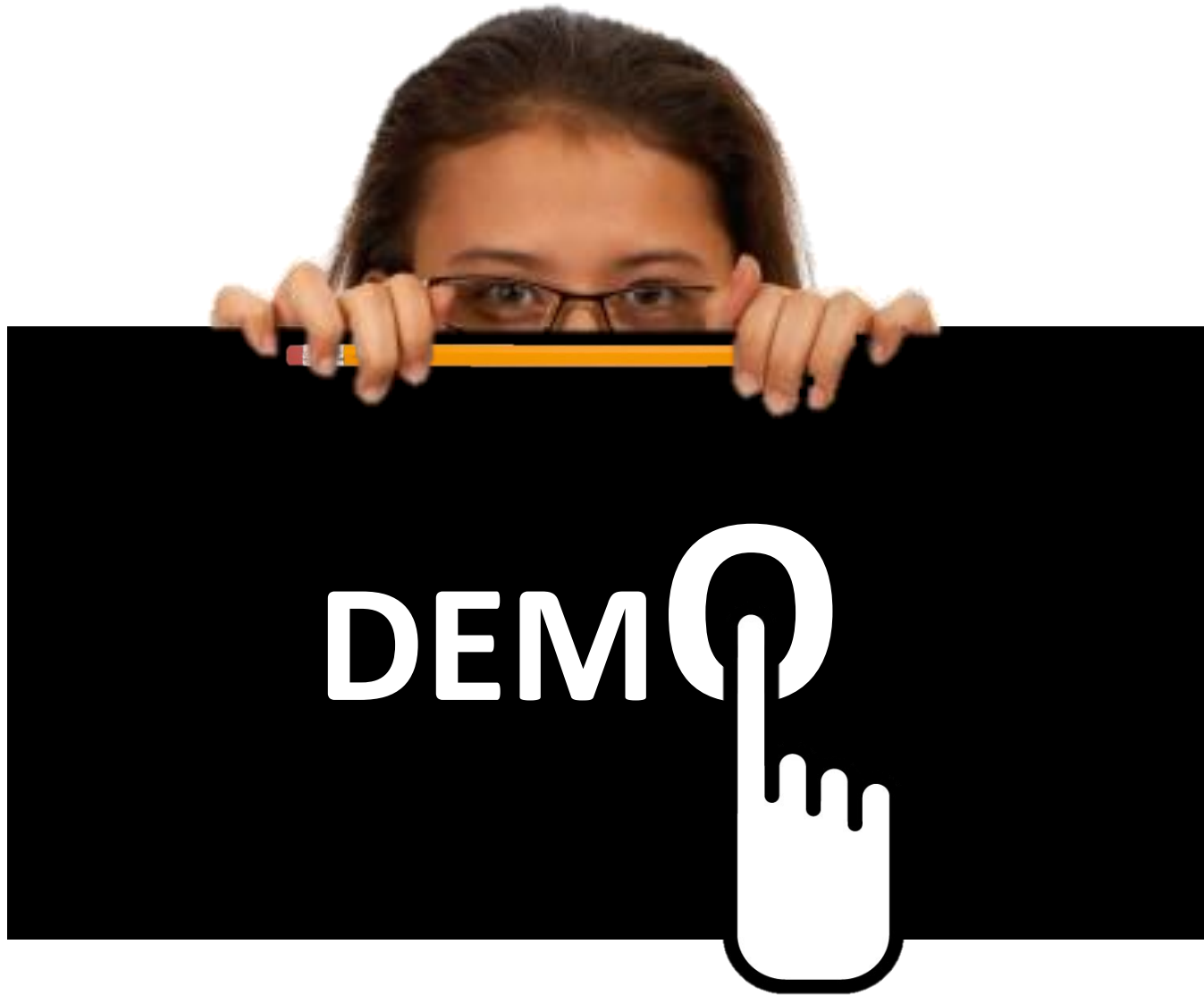
- ❖ `DefaultTableModel model = (DefaultTableModel) tblStaffList.getModel();`

## ❑ Làm việc với các hàng

- ❖ `model.addRow(rowData);`
- ❖ `model.insertRow(rowIndex, rowData);`
- ❖ `model.removeRow(rowIndex);`
- ❖ `model.setRowCount(rowCount);`

## ❑ Làm việc với dữ liệu trong ô

- ❖ `model.getValueAt(rowIndex, columnIndex);`
- ❖ `model.setValueAt(value, rowIndex, columnIndex);`





**FPT POLYTECHNIC**



**Java™**

# **LẬP TRÌNH JAVA 2**

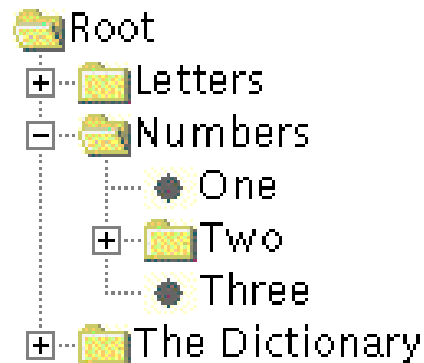
**BÀI 2: SWING ADVANCED**

**PHẦN 2**

[www.poly.edu.vn](http://www.poly.edu.vn)



- ❑ 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.

- ❑ 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

## □ 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

- ❑ 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);
    ...
}
```

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

```
DefaultMutableTreeNode book = null;
```

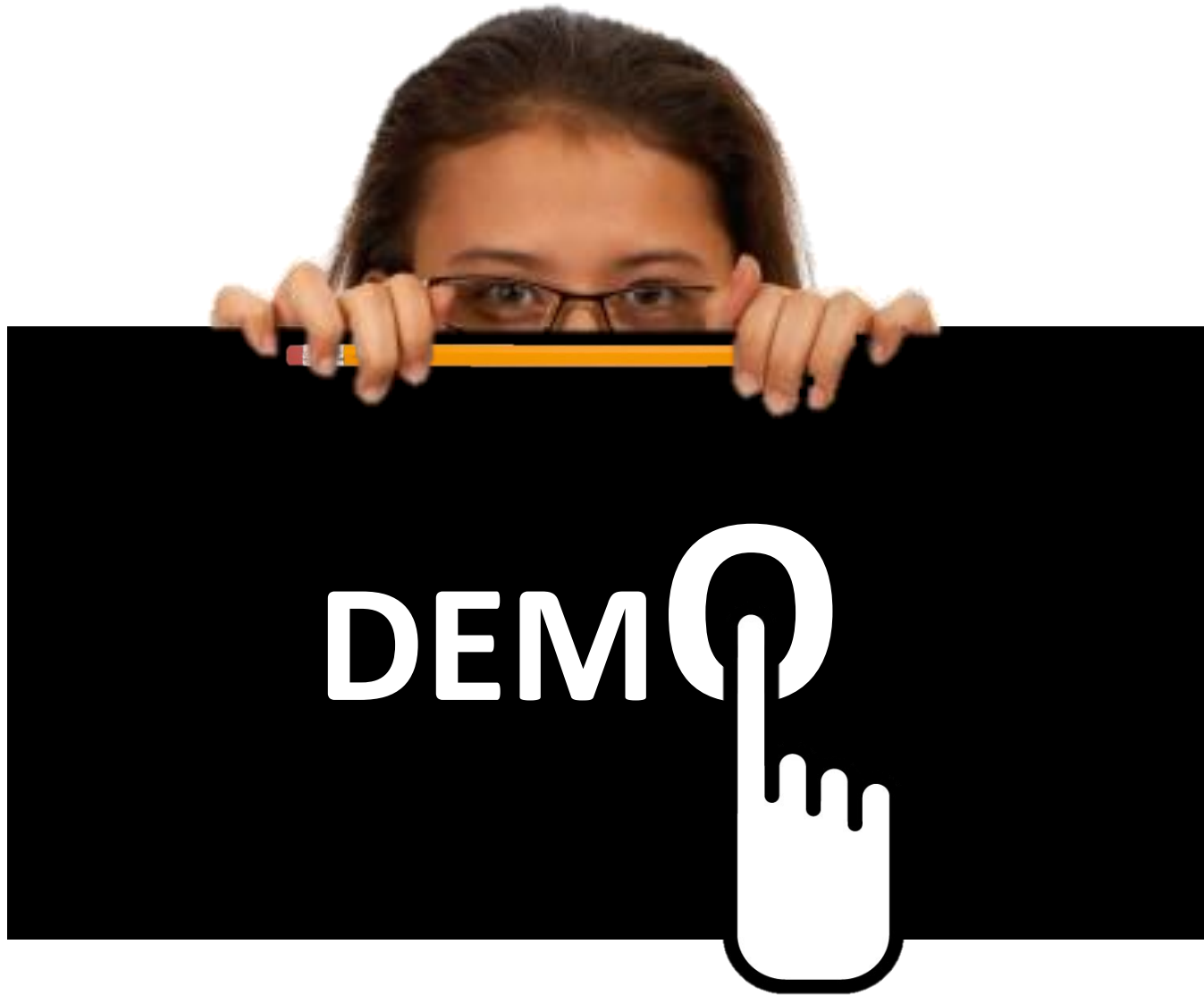
```
//Add node con cho node nhánh category
book = new DefaultMutableTreeNode(new BookInfo
    ("The Java Tutorial Continued: The Rest of the JDK",
    "tutorialcont.html"));
category.add(book);
book = new DefaultMutableTreeNode(new BookInfo
    ("The Swing Tutorial: A Guide to Constructing GUIs",
  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

## ❑ 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

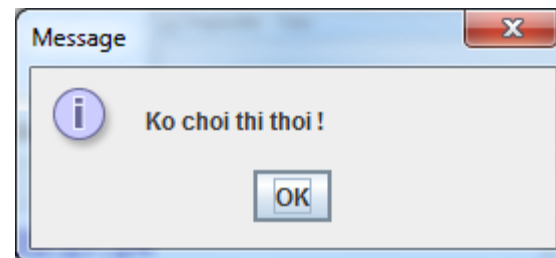
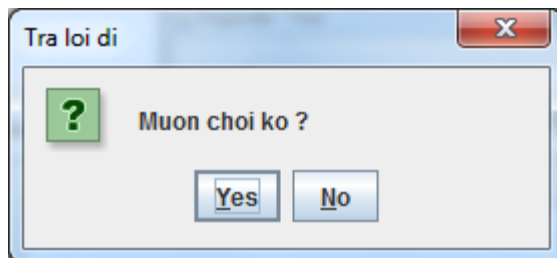
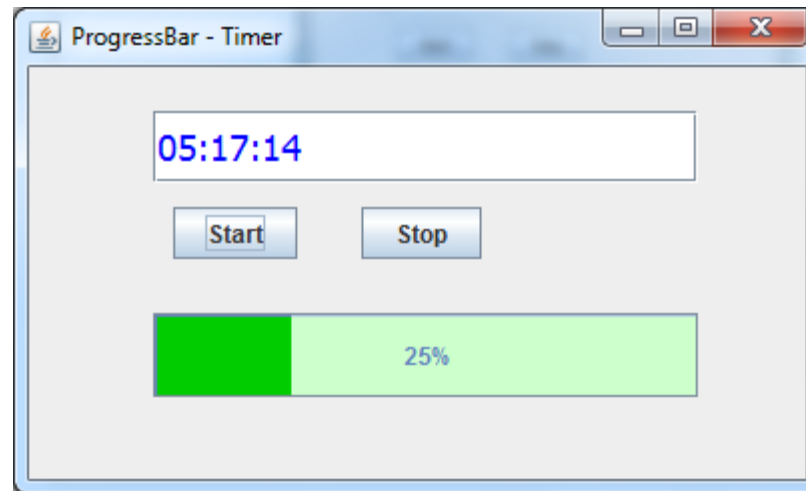
- ❖ TreeExpansion, TreeSelection, TreeWillExpand



- ❑ Timer dùng để kích hoạt một hoặc nhiều sự kiện ActionEvents sau một khoảng thời gian nhất định.
- ❑ Hàm tạo(constructor):  
Timer(int delay, ActionListener listener)  
Hàm tạo này sẽ tạo một sự kiện sau mỗi khoảng thời gian tính theo đơn vị mili giây



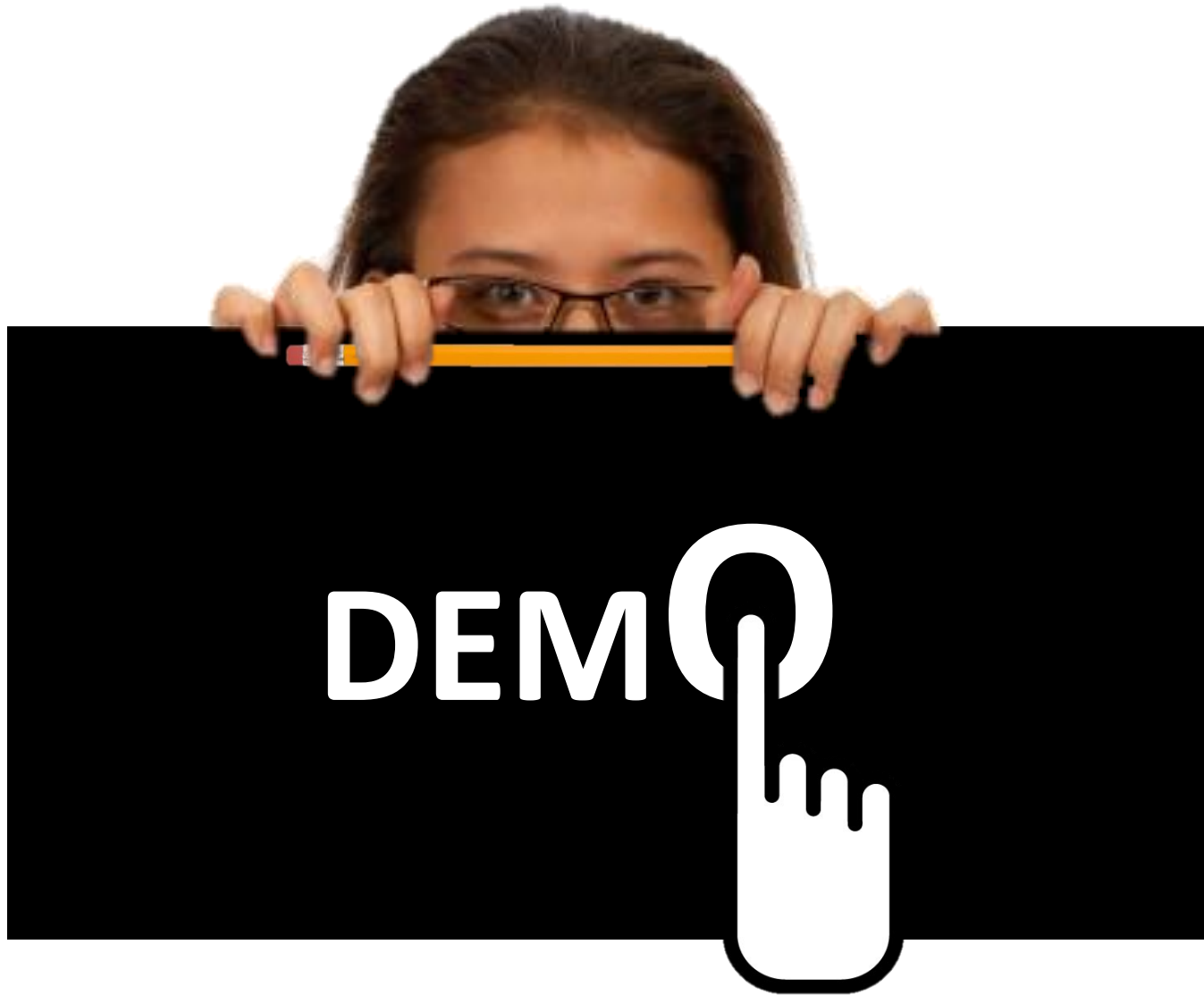
## □ Demo



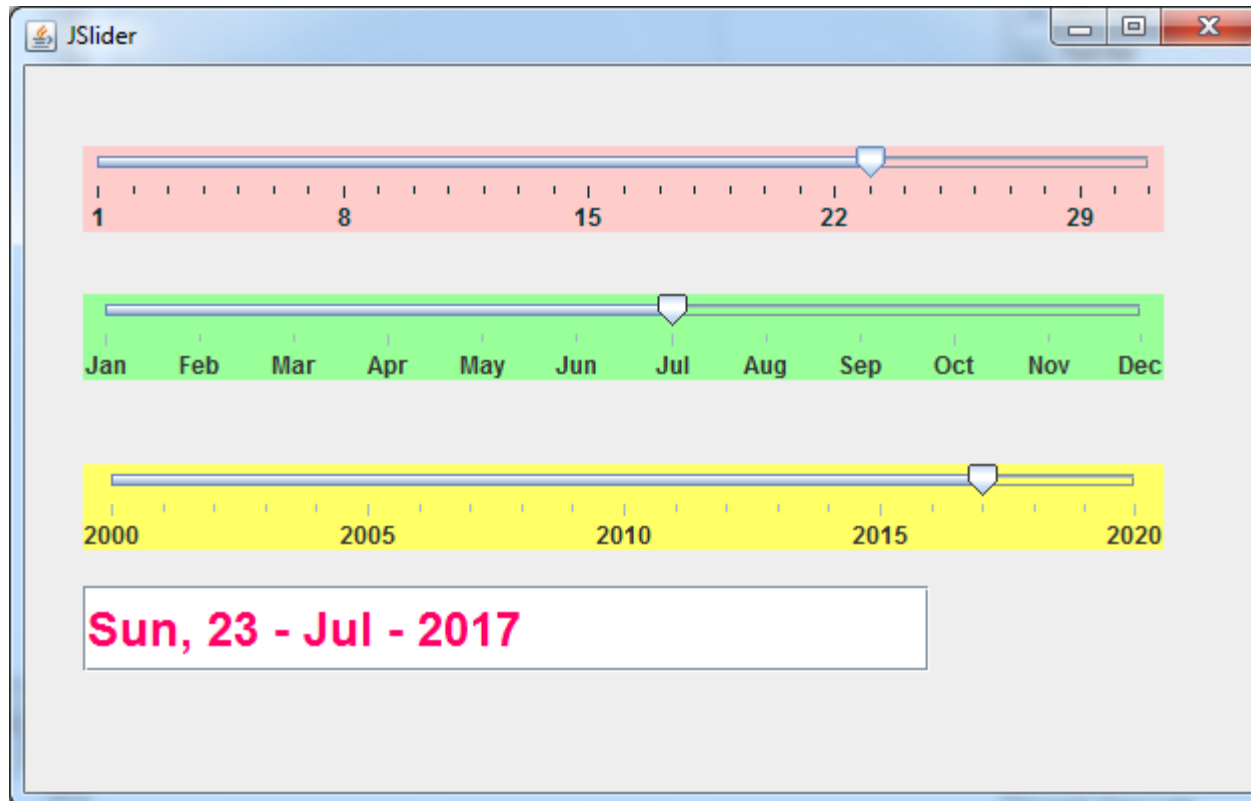
```
public class JProgressBarTimer extends javax.swing.JFrame {
    javax.swing.Timer t;
    public JProgressBarTimer() {
        initComponents();
        ActionListener a = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss");
                txtClock.setText(sdf.format(new Date()));
                int value = prgTienDo.getValue()+1;
                prgTienDo.setValue(value<=100?value:0);
            }
        };
        t = new javax.swing.Timer(1000, a);
    }
    //.....
}
```

```
private void btnStartActionPerformed(java.awt.event.ActionEvent evt) {
    int i = JOptionPane.showConfirmDialog(null, "Muon choi ko ?", "Tra loi di",
        JOptionPane.YES_NO_OPTION);
    if(i==0)
        t.start();
    else
        JOptionPane.showMessageDialog(null, "Ko choi thi thoi !");
}
```

```
private void btnStopActionPerformed(java.awt.event.ActionEvent evt) {
    t.stop();
}
```

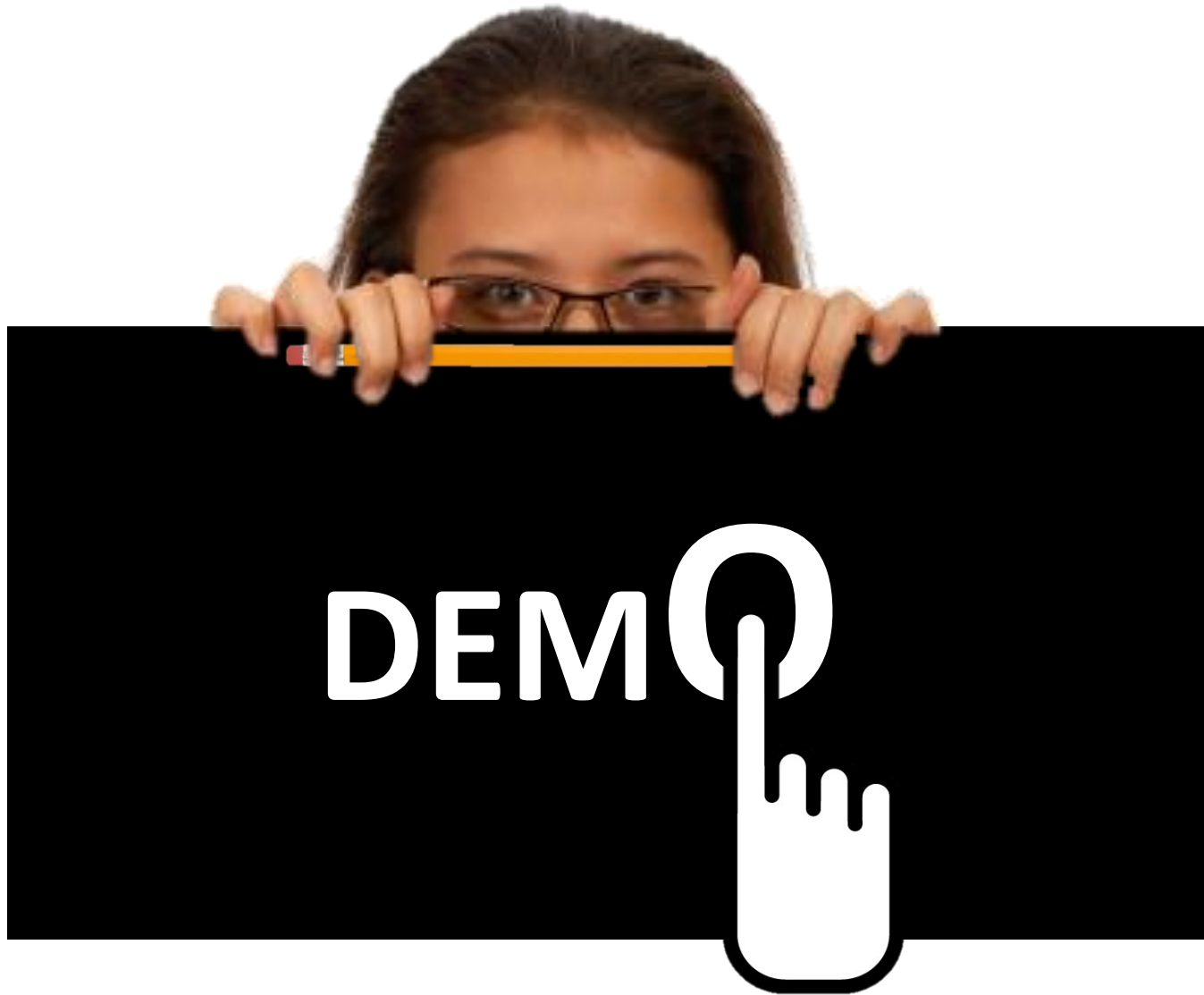


## □ Demo JSlider



```
public class JSliderDemo extends javax.swing.JFrame {
    public JSliderDemo() {
        initComponents();
        Hashtable data = new Hashtable();
        String [] sMonths= {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
            "Sep", "Oct", "Nov", "Dec"};
        for(int i =0; i<12; i++){
            data.put(i, new JLabel(sMonths[i]));
        }
        sldThang.setLabelTable(data);
        Calendar can = Calendar.getInstance();
        sldNgay.setValue(can.get(Calendar.DAY_OF_MONTH));
        sldThang.setValue(can.get(Calendar.MONTH));
        sldNam.setValue(can.get(Calendar.YEAR));
        displayDate();
    }
    void displayDate(){
        SimpleDateFormat sdf = new SimpleDateFormat("EEE, dd - MMM - yyyy");
        int dd = sldNgay.getValue();
        int mm = sldThang.getValue();
        int yy = sldNam.getValue()-1900;
        Date d = new Date(yy, mm, dd);
        txtNgay.setText(sdf.format(d));
    }
    //.....

    private void sldNgayStateChanged(javax.swing.event.ChangeEvent evt) {
        displayDate();
    }
    private void sldThangStateChanged(javax.swing.event.ChangeEvent evt) {
        displayDate();
    }
    private void sldNamStateChanged(javax.swing.event.ChangeEvent evt) {
        displayDate();
    }
}
```



- ❖ JComboBox
- ❖ JList
- ❖ JTable
- ❖ JTree
- ❖ JTimer
- ❖ JSlider





**Cảm ơn**