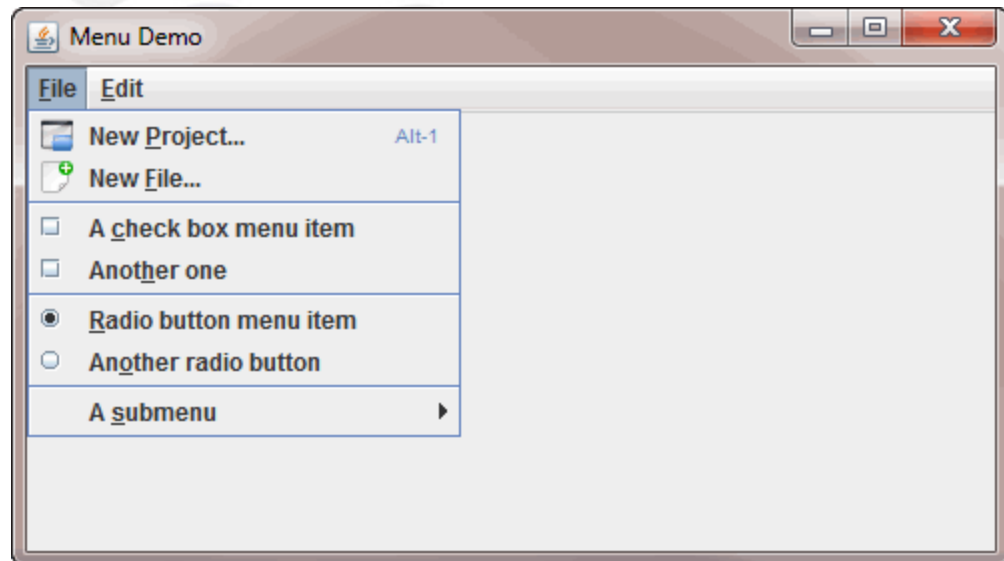


Bài 3

Menu

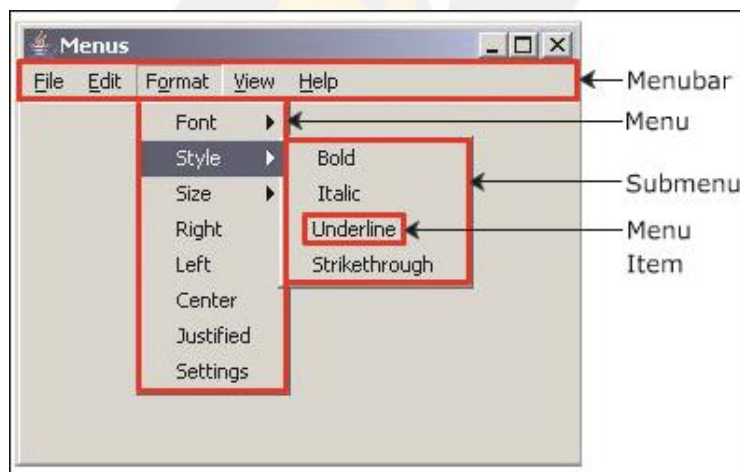
Mục tiêu

- Giới thiệu
- JMenuBar
- JMenu & JMenuItem
- Menu Checkbox & Radio
- JPopupMenu
- JFileChooser
- JToolBar



Giới thiệu

- Hệ thống **menu** cung cấp thanh menu ở dưới thanh tiêu đề cửa sổ.
- Thanh này chứa các **item**, khi click vào chúng xổ ra danh sách thả xuống chứa các item bên trong.
- Mỗi item liên kết với **action** cụ thể để thực hiện nhiệm vụ xác định.



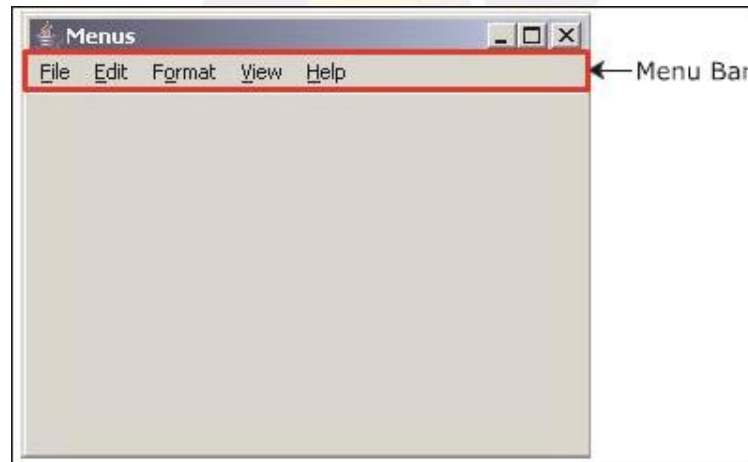
Giới thiệu

- Item trên menu có thể chứa các item con gọi là **sub-menu**.
- Các sub-menu có thể lại chứa các item khác.
- Chúng kết hợp tạo thành hệ thống **menu đa cấp** (nếu cần).
- Ngoài ra Java Swing cũng cung cấp một dạng menu gọi là **popup**, menu này mặc định ẩn, chỉ được kích hoạt khi có hành động cụ thể (**right click**).
- **Popup-menu** xuất hiện ở ngay vị trí con trỏ chuột.
- Loại menu này cho phép lựa chọn một nhóm hành vi đối với **thành phần cụ thể** trên giao diện GUI.



JMenuBar

- JMenuBar là lớp dùng tạo lên thanh menu.
- Thanh menu xuất hiện ở bên dưới thanh tiêu đề cửa sổ.
- Menu bar chứa các item, ví dụ như "File", "Edit", "View"...
- Menu bar được tạo bằng cách sử dụng constructor của class JMenuBar.



JMenuBar

Demo: từ **top container** gọi tới hàm **setMenuBar()**.

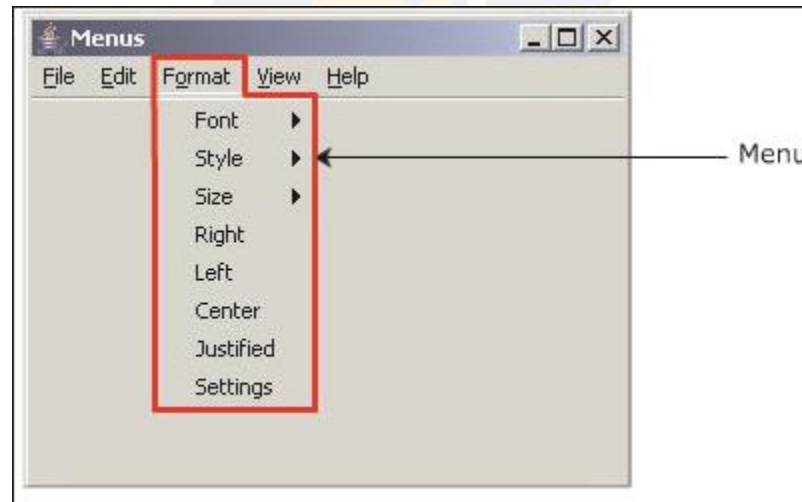
Code Snippet

```
JMenuBar mbrMenuBar;  
JFrame frmMenu;  
frmMenu = new JFrame("Menu bar");  
// Creates a menu bar  
mbrMenuBar = new JMenuBar();  
...  
// Adds the JMenuBar to the top level container JFrame  
frmMenu.setMenuBar( mbrMenuBar);
```



JMenu và JMenuItem

- **Jmenu** là lớp đại diện trình đơn trên menu bar.
- Khi click vào sẽ xổ ra danh sách thả xuống chứa một hoặc nhiều item con.
- Jmenu được thêm vào **JMenuBar** bằng cách gọi phương thức `add()`.
- Phương thức **`add()`** nhận Jmenu như tham số.



JMenu và JMenuItem

Demo Jmenu:

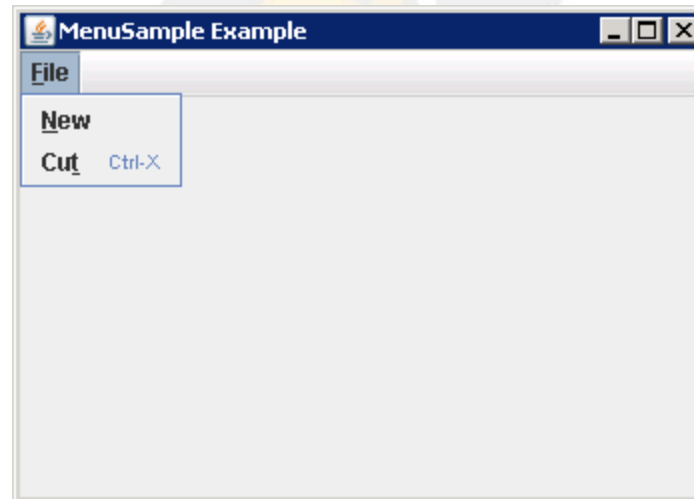
Code Snippet

```
JMenuBar mbrMenuBar;  
JMenu mnuFile, mnuEdit;  
// Creates a menu bar  
mbrMenuBar = new JMenuBar();  
...  
// Creates a menu with the specified label  
mnuFile = new JMenu("File");  
mnuEdit = new JMenu("Edit");  
// Adds the menu to the menu bar  
mbrMenuBar.add(mnuFile);  
mbrMenuBar.add(mnuEdit);
```



JMenu và JMenuItem

- **JMenuItem** là lớp tạo các item cho menu, đối tượng này thêm trực tiếp vào **JMenu**.
- Có thể cài đặt hành động thực hiện nhiệm vụ cụ thể.
- JMenuItem có nhiều **constructor** khác nhau.
- JMenuItem có thể enable/un-enable và cài đặt phím tắt.



JMenu và JMenuItem

Demo JMenuItem

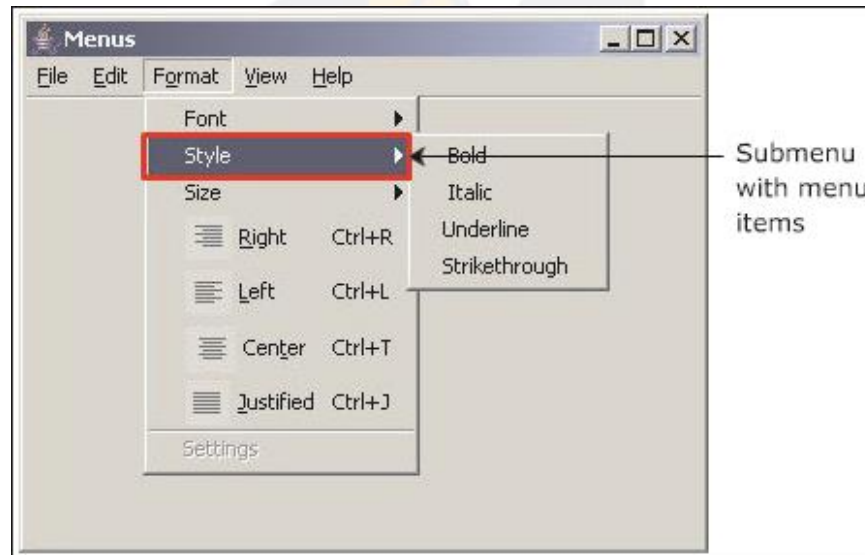
Code Snippet

```
// An int specifying the numeric code for a keyboard key
int key = KeyEvent.VK_O;
// An integer specifying the modifier
int modifier = KeyEvent.ALT_MASK;
KeyStroke keystroke;
// Returns instance of a KeyStroke, given a numeric key code and
// a set of modifiers.
keystroke = KeyStroke.getKeyStroke(key, modifier);
// Sets the key combination which invokes the menu item's action
// listeners
mniOpen.setAccelerator(keystroke);
```



JMenu và JMenuItem

- **Sub-Menu** là item menu được thêm vào Jmenu bằng phương thức add().
- Phương thức add() nhận JMenuItem như tham số để tạo sub-menu.
- Sub-menu có thể tạo đơn giản bằng thêm Jmenu vào JMenuItem.



JMenu và JMenuItem

Sub-Menu demo:

Code Snippet

```
...
JMenu mnuTable, mnuDelete;
JMenuItem mniTable, mniColumns, mniRows;
...
// Adds the menu item to the "Delete" menu.
mnuDelete.add(mniTable);
mnuDelete.add(mniColumns);
mnuDelete.add(mniRows);
...
// Creates sub-menu(cascaded) "Delete" in the "Table" menu.
mnuTable.add(mnuDelete);
```



JMenu và JMenuItem

Cài đặt sự kiện:

Code Snippet

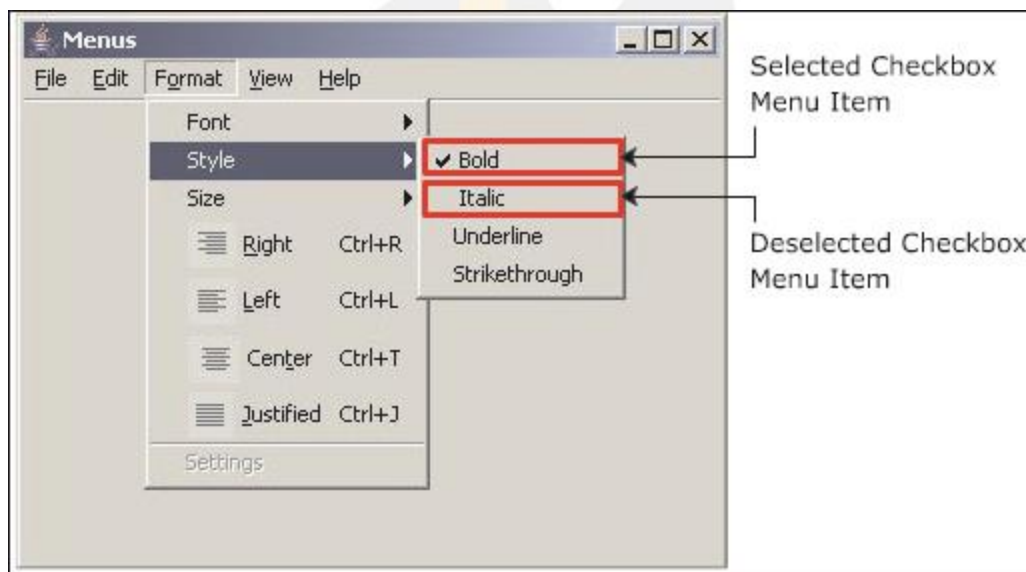
```
// Event handling of the "Exit "  
menuItem mniExit.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        // Closes the application  
        System.exit(0);  
    }  
});
```



Menu Checkbox & Radio

JCheckBoxMenuItem

- Tương tự như một JMenuItem nhưng có tùy chọn select.
- Dấu tích sẽ được đặt tùy vào item có được tích hay không.
- Menu loại này có thể chứa text, icon.



Menu Checkbox & Radio

Demo code:

Code Snippet

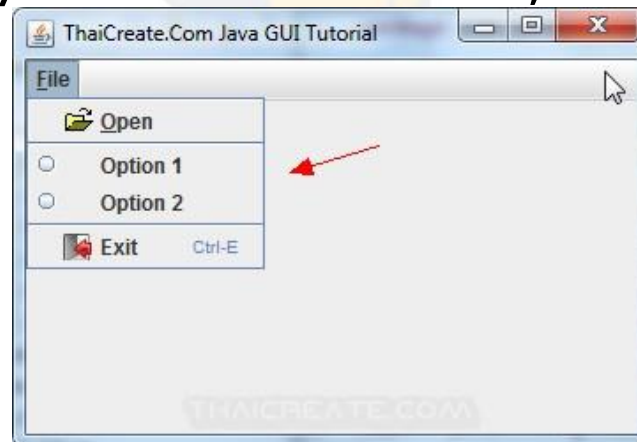
```
JMenuItem mniToolbars;  
JCheckBoxMenuItem mncStandard, mncFormatting, mncAutoText;  
// Creates the checkbox menuitem with the label "Standard" and  
// initially selected  
mncStandard = new JCheckBoxMenuItem("Standard", true);  
// Creates the checkbox menuitem with the label "Formatting" and  
// initially selected  
mncFormatting = new JCheckBoxMenuItem("Formatting", true);  
// Creates the checkbox menuitem with the label "Auto Text" and  
// initially selected  
mncAutoText = new JCheckBoxMenuItem("Auto Text");  
// Adds the checkbox menuitem to the "Toolbars"  
menuItem mniToolbars.add(mncStandard);  
mniToolbars.add(mncFormatting);  
mniToolbars.add(mncAutoText);
```



Menu Checkbox & Radio

JRadioButtonMenuItem

- Tương tự như một JRadioButton trong cả cách hiển thị và chức năng.
- Chỉ có khác biệt là nó có khả năng thêm vào hệ thống menu.
- Để tạo thành nhóm, chúng được đặt vào một ButtonGroup.
- Menu loại này có thể chứa text, icon.



Menu Checkbox & Radio

Demo code:

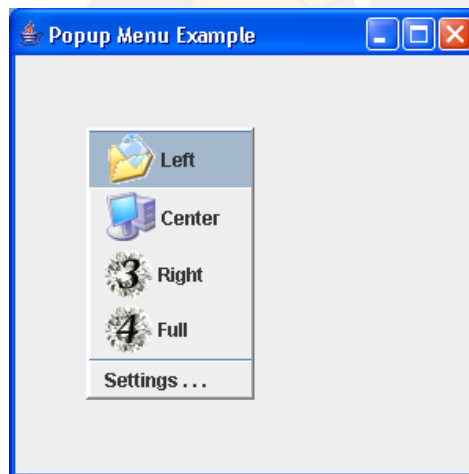
Code Snippet

```
JMenu mnuView;  
JRadioButtonMenuItem mnrNormal, mnrWeb, mnrPrint;  
ButtonGroup bgView;  
// Creates the radio button menu item with the label "Normal  
// Layout" and selected state.  
    mnrNormal = new JRadioButtonMenuItem("Normal Layout",true);  
// Creates the radio button menu item with the specified label  
    mnrWeb = new JRadioButtonMenuItem("Web Layout");  
    mnrPrint = new JRadioButtonMenuItem("Print Layout");  
// Creating ButtonGroup instance  
    bgView = new ButtonGroup();  
// Adds the radio button menu item to the buttongroup  
    bgView.add(mnrNormal);  
    bgView.add(mnrWeb);  
    bgView.add(mnrPrint);  
// Adds the radio button menu item to the "View"  
    mnuView.add(mnrNormal);  
    mnuView.add(mnrWeb);  
    mnuView.add(mnrPrint);
```

JPopupMenu

JPopupMenu

- Là dạng menu hiển thị khi có tác động chuột phải vào container.
- Menu xuất hiện chính xác tại vị trí con trỏ chuột.
- Trong nhiều trường hợp menu dạng này tiện lợi hơn menu bar vì ít phải di chuyển chuột.
- Menu này có thể chứa nhiều menu item bên trong.



JFileChooser

JFileChooser:

- Là một hộp dialog chuẩn sử dụng để điều hướng tới hệ thống file, giúp lựa chọn folder hoặc file.
- Các đặc điểm:

1

- Cho phép tạo mới folder, file.

2

- Thực hiện thao tác mở, lưu với các nhãn và nút khác nhau.

3

- Cung cấp các xem trước và biểu tượng thích hợp.

4

- Hiển thị các file với tiêu chí bắt buộc bằng bộ lọc.



JFileChooser

JFileChooser:

Code Snippet

```
JFileChooser fcrChooser;  
// Creates a file chooser  
fcrChooser = new JFileChooser();  
// Opens the standard "Open File"  
dialog box. int returnValue = fcrChooser.showOpenDialog(parent);  
// Checks whether you have accepted the file  
if (returnValue == JFileChooser.APPROVE_OPTION) {  
    // Retrieve the file name  
    File file = fcrChooser.getSelectedFile();  
} else {  
    System.out.println("File not selected.");  
}
```



JFileChooser

JFileChooser: lấy về đường dẫn thư mục/file được chọn

Code Snippet

```
JFileChooser fcrChooser;  
File currentDir;  
// Creates a file chooser  
fcrChooser = new JFileChooser();  
// Retrieves the selected  
directory. currentDir = fcrChooser.getCurrentDirectory();
```



JFileChooser

JFileChooser: bộ lọc filter

Code Snippet

```
public class ImageFilter extends FileFilter {  
    // Method to decide the criteria  
    public boolean accept(File file) {  
        // If it is a directory it may contain image files so ignore it  
        if (file.isDirectory())  
            return true;  
        // Retrieves the file name.  
        String name = file.getName();  
        // Only accept images with extension of .gif, .jpg and .png  
        if (name.endsWith(".gif") || name.endsWith(".jpg") || name.endsWith(".png"))  
            return true;  
        else  
            return false;  
    }  
    public String getDescription() {  
        return "Image files";    }  
}
```



JFileChooser

JFileChooser: cài đặt bộ lọc filter vào JFileChooser.

Code Snippet

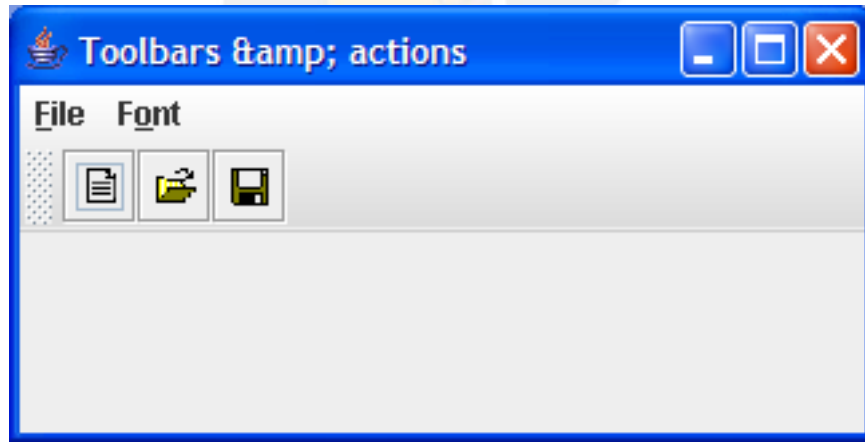
```
// Create a file chooser
fcrChooser = new JFileChooser();
// Creates an object of class
ImageFilter ImageFilter filter = new ImageFilter();
// Sets the filter
fcrChooser.setFileFilter(filter);
```



JToolBar

JToolBar:

- Tương tự thanh menu, có chứa button và icon.
- Nút bấm có sự kiện và được liên kết với hành động cụ thể.
- Có thể di chuyển (dockable), kéo tới vị trí khác nếu muốn.



JToolBar

Các bước tạo **JToolBar**:

1

- Khai báo đối tượng từ lớp `JButton`.

2

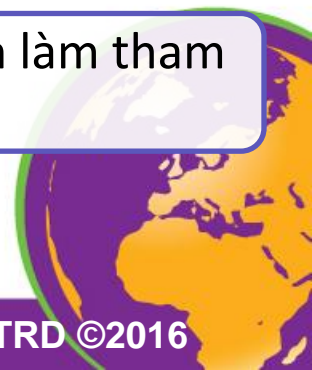
- Khởi tạo đối tượng với `ImageIcon`.

3

- Sử dụng phương thức `setToolTipText()` của lớp `Jbutton` để cài đặt tool tip.

4

- Sử dụng phương thức `add()` của `JToolBar` với button trên làm tham số.



JToolBar

Code demo:

Code Snippet

```
JToolBar tbrTools;  
JButton btnSave;  
...  
...  
// Creates a toolbar  
tbrTools = new JToolBar();  
// Adds the toolbar to the frame in the north direction  
getContentPane().add(tbrTools,BorderLayout.NORTH);  
...  
// Creates a button with the icon  
btnSave = new JButton(new ImageIcon("Save.gif"));  
// Sets the tool tip of the button "Save"  
btnSave.setToolTipText("Save");  
// Adds the button "Save" to the toolbar  
tbrTools.add(btnSave);  
...
```

JToolBar

Dòng phân cách:

Code Snippet

```
public class Demo {  
    public static void main(String[] args){  
        JFrame frm;  
        JToolBar tbrTool;  
        JButton btnOpen, btnSave;  
        // Creates a frame  
        frm=new JFrame();  
        frm.setSize(400,400);  
        frm.setVisible(true);  
        // Creates a toolbar  
        tbrTool = new JToolBar();  
        // Adds the toolbar to the frame in the north direction  
        frm.add(tbrTool, BorderLayout.NORTH);  
    }  
}
```



JToolBar

```
// Create the Open button
btnOpen = new JButton(new ImageIcon("Open.gif"));
// Add the open button
tbrTool.add(btnOpen);
// Add a separator before the next button on the toolbar
tbrTool.addSeparator();
// Create the Save button
btnSave = new JButton(new ImageIcon("Save.gif"));
// Add the open button
tbrTool.add(btnSave);
}
```



Tóm tắt bài học

- ✓ Hệ thống **menu** cung cấp những tùy chọn để thực hiện hành động, tác vụ trên ứng dụng GUI.
- ✓ **JFileChooser** là menu dạng cửa sổ dialog, cho phép truy cập tới hệ thống file/folder.
- ✓ **JToolBar** là menu có chứa icon và button.
- ✓ **JPopupMenu** là menu có dạng bong bóng, xuất hiện khi thao tác chuột phải.



HẾT
XIN CẢM ƠN!

