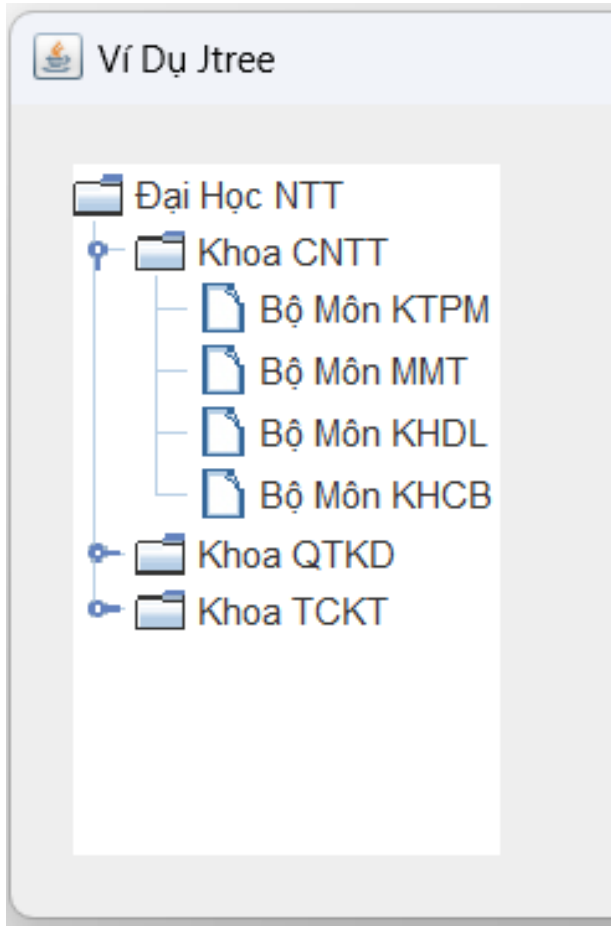


Bài 05:

JTree, JMenuBar, JToolBar

Giảng Viên: ThS. Giang Hào Côn

1) Giới Thiệu



Một số tính năng quan trọng của JTree bao gồm:

- 1) Hiển thị dữ liệu theo cấu trúc cây.
- 2) Tùy chỉnh giao diện.
- 3) Xử lý sự kiện.
- 4) Tìm kiếm và lọc dữ liệu.

2) Tạo 1 đối tượng JTree

```
JTree tree = new JTree();
```

3) Tạo 1 đối tượng JTree

```
1 DefaultMutableTreeNode root = new DefaultMutableTreeNode("Root");  
2 DefaultMutableTreeNode child1 = new DefaultMutableTreeNode("Child 1");  
3 DefaultMutableTreeNode child2 = new DefaultMutableTreeNode("Child 2");  
4 root.add(child1);  
5 root.add(child2);  
6 tree.setModel(new DefaultTreeModel(root));
```

4) Đặt JTree vào giao diện

```
1 JFrame frame = new JFrame();  
2 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
3 frame.add(new JScrollPane(tree));  
4 frame.pack();  
5 frame.setVisible(true);
```

5) Tùy chỉnh biểu tượng cho node

```
1 DefaultTreeCellRenderer renderer = new DefaultTreeCellRenderer();  
2 renderer.setLeafIcon(myLeafIcon);  
3 renderer.setClosedIcon(myClosedIcon);  
4 renderer.setOpenIcon(myOpenIcon);  
5 tree.setCellRenderer(renderer);
```

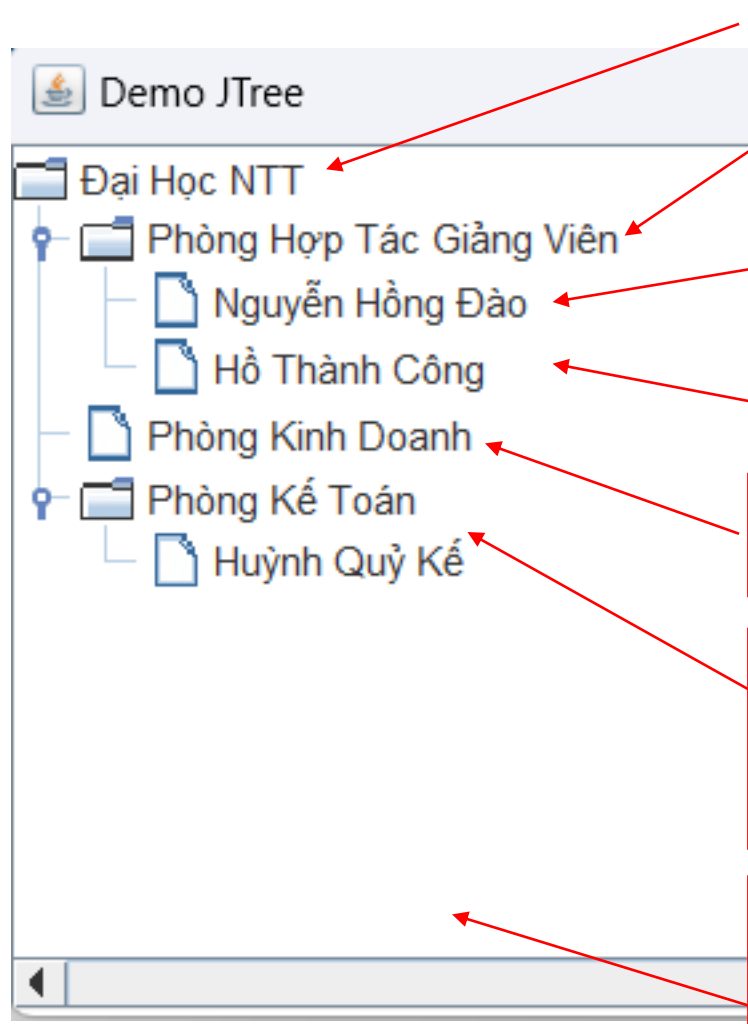
Biểu tượng node lá

Biểu tượng node chưa mở rộng

Biểu tượng node đã mở rộng

Cài đặt các khai báo biểu tượng cho tree.

Jtree – Ví dụ



```

root=new DefaultMutableTreeNode("Đại Học NTT");
tree=new JTree(root);

DefaultMutableTreeNode nodePhtgv=new DefaultMutableTreeNode("Phòng Hợp Tác Giảng Viên");
root.add(nodePhtgv);

DefaultMutableTreeNode nodeNv1=new DefaultMutableTreeNode("Nguyễn Hồng Đào");
nodePhtgv.add(nodeNv1);

DefaultMutableTreeNode nodeNv2=new DefaultMutableTreeNode("Hồ Thành Công");
nodePhtgv.add(nodeNv2);

DefaultMutableTreeNode nodePhKd=new DefaultMutableTreeNode("Phòng Kinh Doanh");
root.add(nodePhKd);

DefaultMutableTreeNode nodekt=new DefaultMutableTreeNode("Phòng Kế Toán");
root.add(nodekt);

DefaultMutableTreeNode nodeNv3=new DefaultMutableTreeNode("Huỳnh Quỳ Kế");
nodekt.add(nodeNv3);

tree.expandRow(0);

JScrollPane sc=new JScrollPane(tree, JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
                                JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
pnTree.add(sc, BorderLayout.CENTER);

```

JMenu

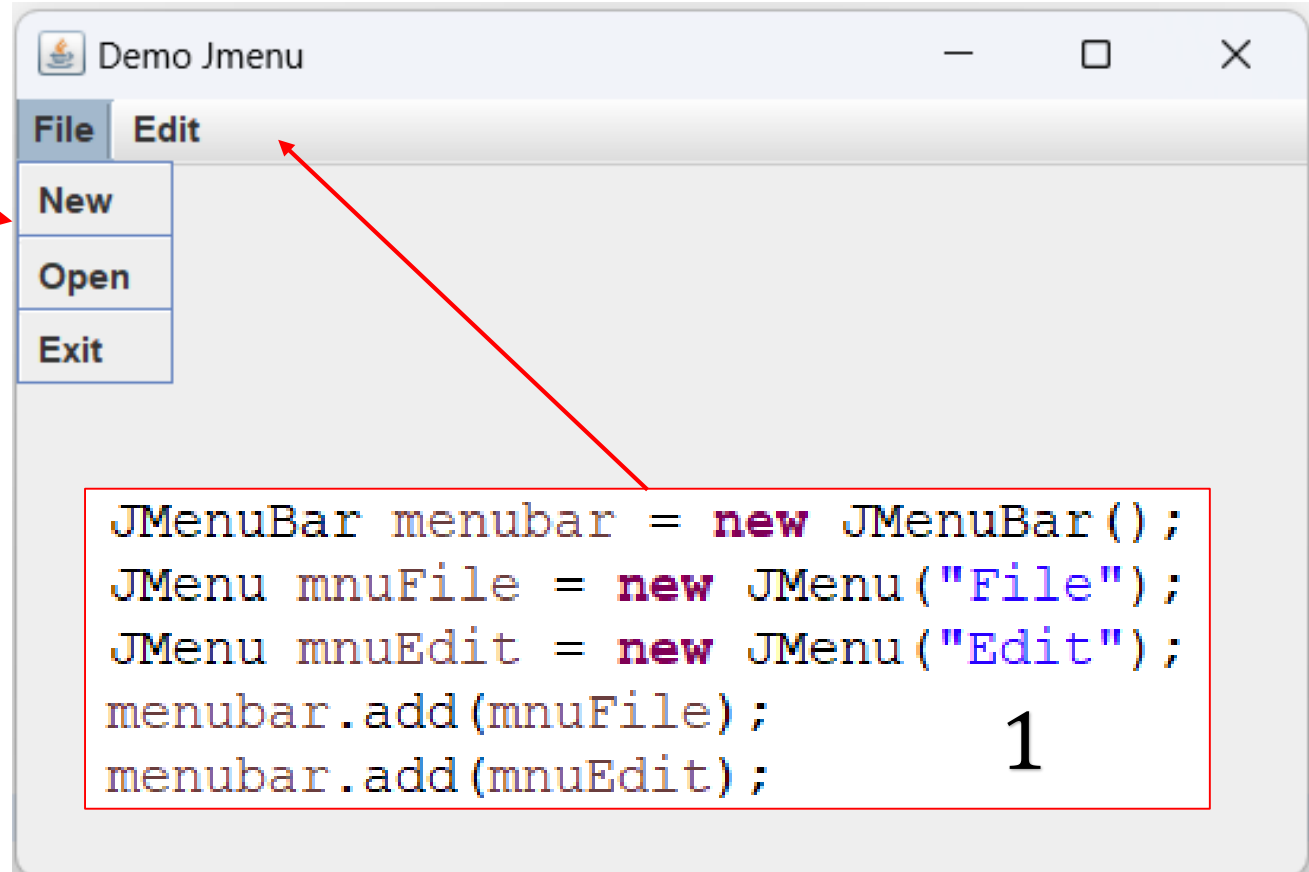
```
JMenuItem mnuFileNew=new JMenuItem("New");  
JMenuItem mnuFileOpen=new JMenuItem("Open");  
JMenuItem mnuFileExit=new JMenuItem("Exit");  
mnuFile.add(mnuFileNew);  
mnuFile.addSeparator();  
mnuFile.add(mnuFileOpen);  
mnuFile.addSeparator();  
mnuFile.add(mnuFileExit);
```

2

Gắn MenuBar vào cửa sổ

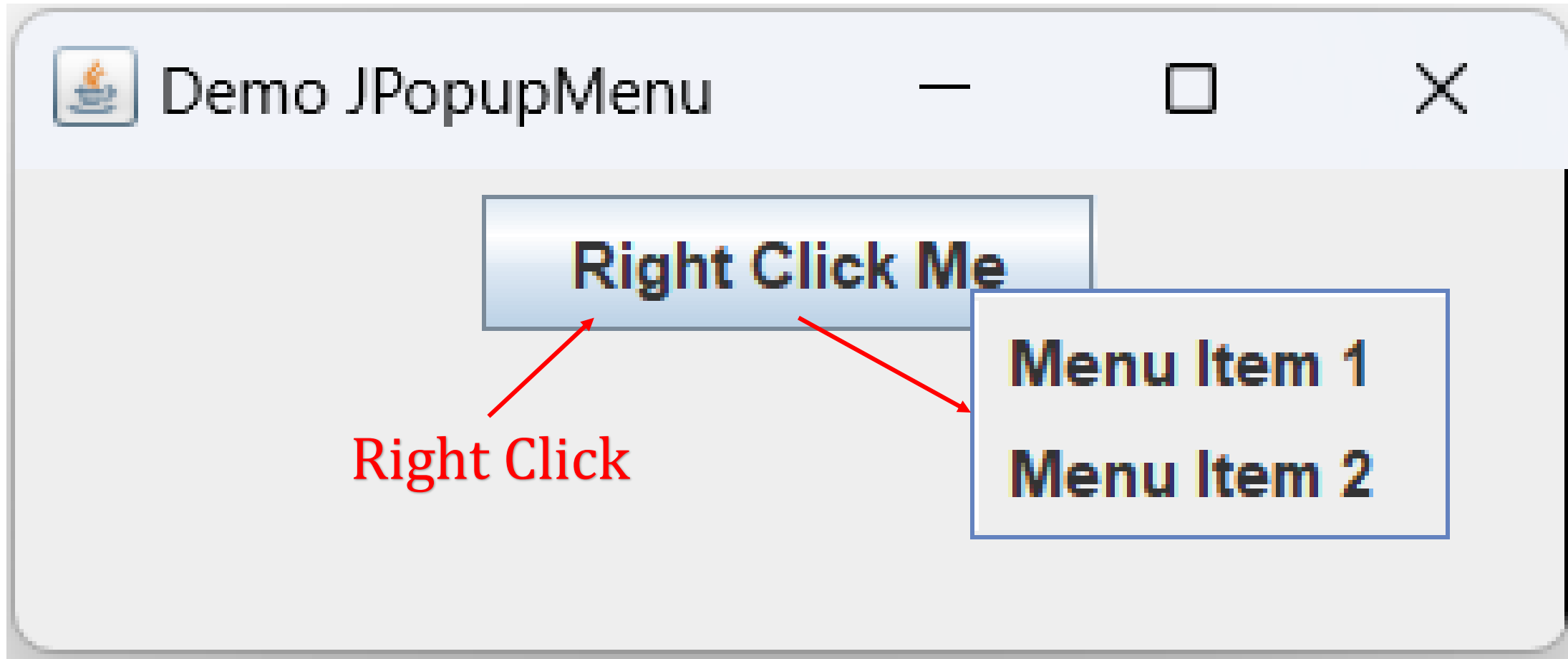
```
frmVidu04 = new JFrame();  
frmVidu04.setJMenuBar(menuBar);
```

3



1

JPopupMenu



JPopupMenu

Các bước tạo Popup Menu

Bước 01: Tạo đối tượng JPopupMenu:

```
JPopupMenu popupMenu = new JPopupMenu();
```

Bước 02: Thêm các mục menu vào JPopupMenu:

```
JMenuItem menuItem1 = new JMenuItem("Menu Item 1");  
JMenuItem menuItem2 = new JMenuItem("Menu Item 2");  
  
popupMenu.add(menuItem1);  
popupMenu.add(menuItem2);
```

JPopupMenu

Các bước tạo Popup Menu

Bước 03: Liên kết JPopupMenu với thành phần giao diện người dùng (ví dụ: JButton):

```
JButton button = new JButton("Right Click Me");

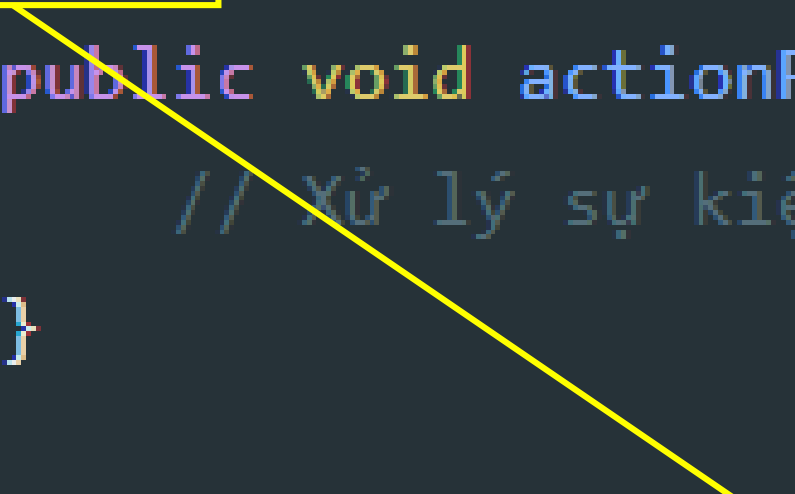
button.addMouseListener(new MouseAdapter() {
    public void mouseReleased(MouseEvent e) {
        if (e.isPopupTrigger()) {
            popupMenu.show(e.getComponent(), e.getX(), e.getY());
        }
    }
});
```

JPopupMenu

Các bước tạo Popup Menu

Bước 04: Xử lý sự kiện khi người dùng chọn một mục menu:

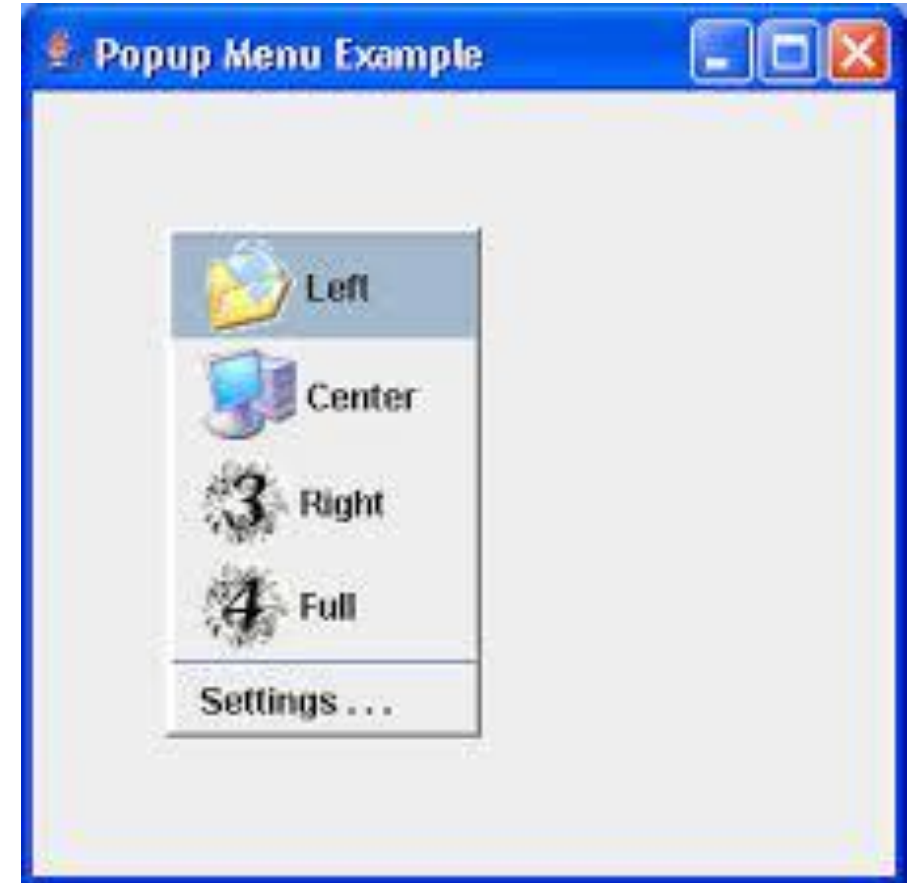
```
menuItem1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        // Xử lý sự kiện cho mục menu 1  
    }  
});
```



Tên biến mục chọn đã khai báo ở bước 02

Tính năng và lợi ích của JPopupMenu

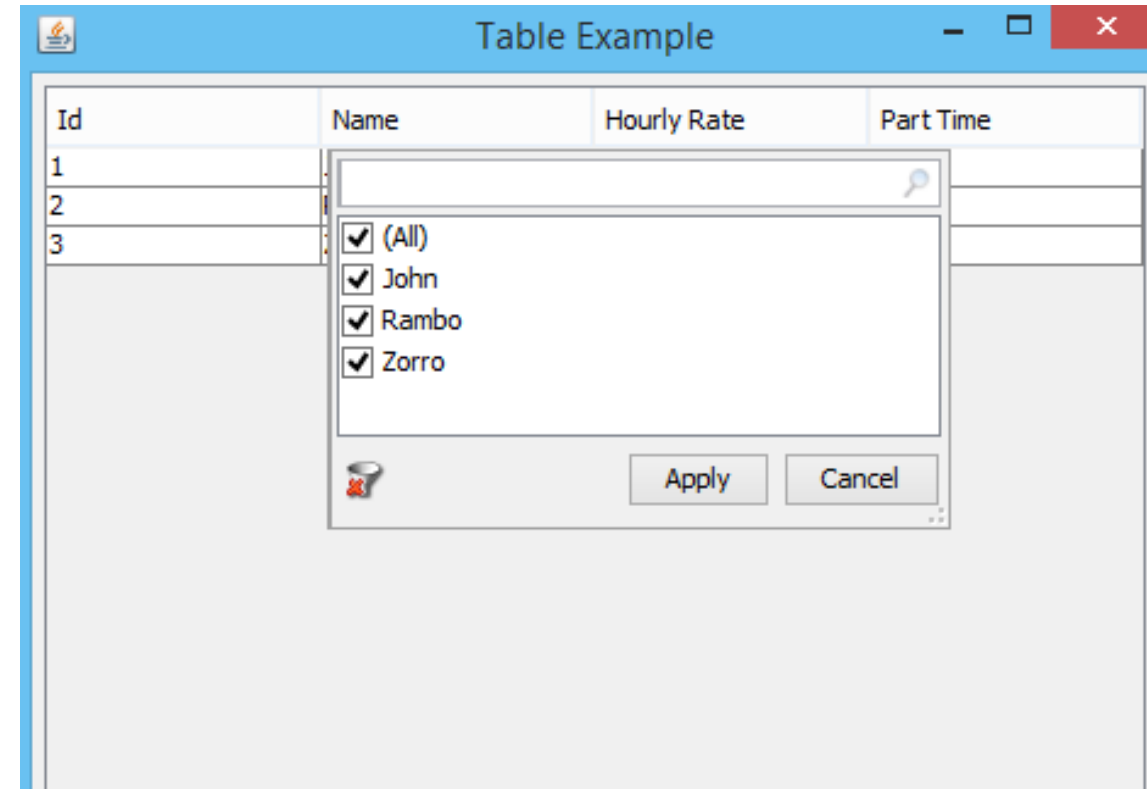
- 1) Hiển thị menu context.
- 2) Tích hợp dễ dàng.
- 3) Đa dạng hóa menu.
- 4) Xử lý sự kiện menu.
- 5) Đa nền tảng.
- 6) Tính linh hoạt và tùy chỉnh



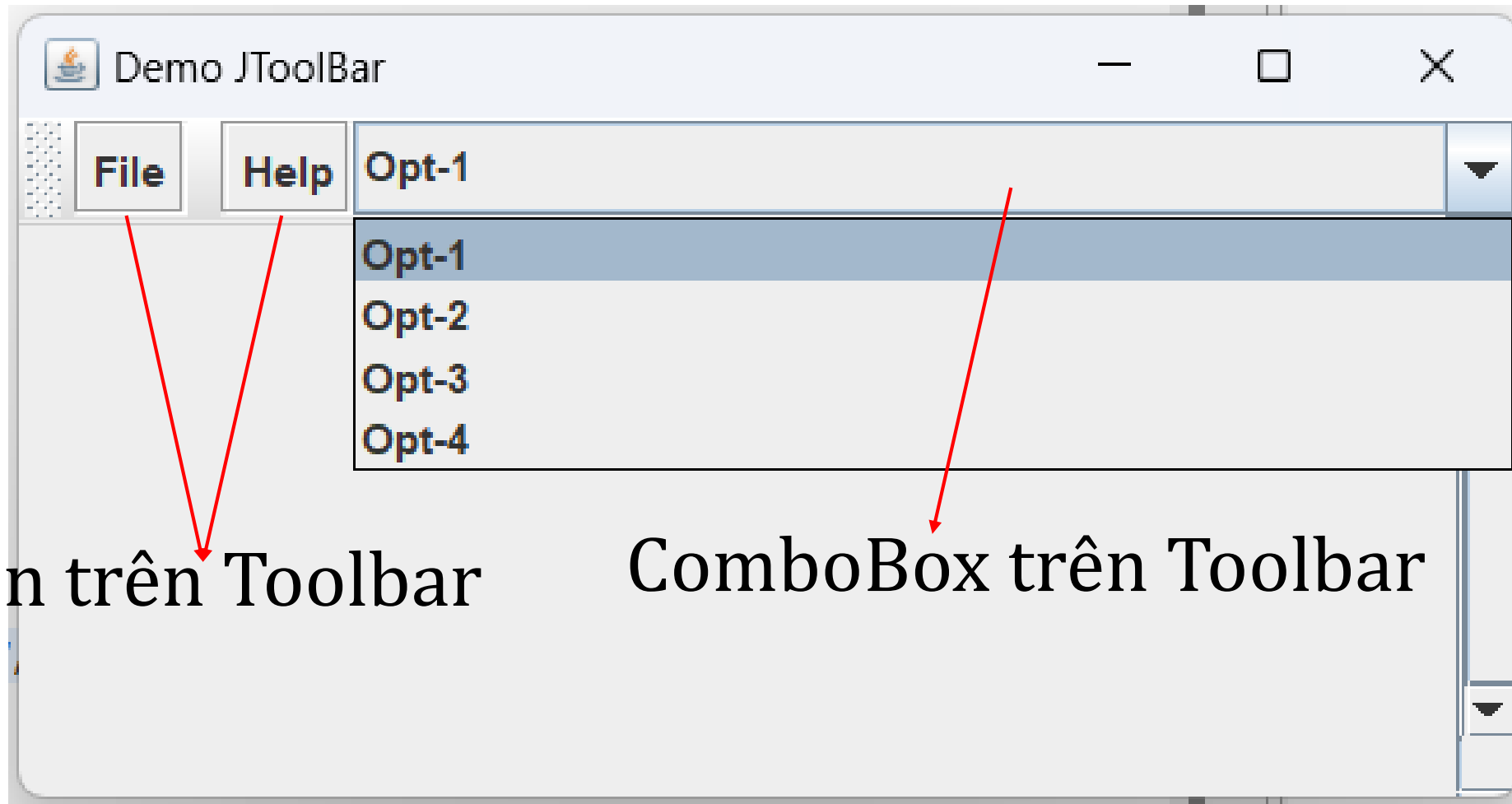
JPopupMenu

Một số lưu ý khi sử dụng JPopupMenu

- 1) Xác định rõ vị trí hiển thị.
- 2) Đăng ký bộ xử lý sự kiện.
- 3) Kiểm tra trạng thái.
- 4) Định dạng menu.
- 5) Xử lý sự kiện bên ngoài menu.
- 6) Xử lý menu trống.



JToolBar



Button trên Toolbar

ComboBox trên Toolbar

Các bước tạo JToolBar

1. Tạo một đối tượng JToolBar:

```
JToolBar toolBar = new JToolBar();
```

2. Thêm các thành phần vào JToolBar, ví dụ: nút (JButton), hộp kiểm (JCheckBox), hộp chọn (JComboBox), v.v.:

```
1 JButton button1 = new JButton("Button 1");  
2 JButton button2 = new JButton("Button 2");  
3 JCheckBox checkBox = new JCheckBox("Check Box");  
4 JComboBox<String> comboBox = new JComboBox<>(new String[]{"Option 1", "Option 2", "Option 3"});  
   toolBar.add(button1);  
5 toolBar.add(button2);  
6 toolBar.add(checkBox);  
7 toolBar.add(comboBox);
```

Các bước tạo JToolBar

3. Đặt vị trí và các thuộc tính cho JToolBar:

```
1 toolBar.setFloatable(false); // Không cho phép di chuyển thành công cụ
2 toolBar.setOrientation(JToolBar.HORIZONTAL); // Định dạng theo chiều ngang
```

4. Thêm JToolBar vào JFrame hoặc JPanel:

```
1 JFrame frame = new JFrame();
2 JToolBar toolBar = new JToolBar(); // Thêm các thành phần vào JToolBar
3 frame.add(toolBar, BorderLayout.PAGE_START); // Đặt JToolBar ở vị trí trên cùng của JFrame
```

```
1 JPanel panel = new JPanel();
2 JToolBar toolBar = new JToolBar(); // Thêm các thành phần vào JToolBar
3 panel.add(toolBar); // Đặt JToolBar vào JPanel
```


Các bước tạo JToolBar

5. Xử lý sự kiện với JToolBar - Xử lý sự kiện khi nhấn vào một JButton trong JToolBar:

```
button.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // Xử lý sự kiện khi nhấn vào button trong JToolBar  
    }  
});
```

Các bước tạo JToolBar

5. Xử lý sự kiện với JToolBar - Xử lý sự kiện khi chọn một JCheckBox trong JToolBar:

```
checkBox.addItemListener(new ItemListener() {  
    @Override  
    public void itemStateChanged(ItemEvent e) {  
        // Xử lý sự kiện khi trạng thái của checkbox trong JToolBar thay đổi  
        if (checkBox.isSelected()) {  
            // Checkbox được chọn  
        } else {  
            // Checkbox không được chọn  
        }  
    }  
});
```

Các bước tạo JToolBar

5. Xử lý sự kiện với JToolBar - Xử lý sự kiện khi chọn một JRadioButton trong JToolBar:

```
radioButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // Xử lý sự kiện khi chọn radio button trong JToolBar  
    }  
});
```

Các bước tạo JToolBar

5. Xử lý sự kiện với JToolBar - Xử lý sự kiện khi chọn một JMenuItem trong JToolBar:

```
menuItem.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // Xử lý sự kiện khi chọn menu item trong JToolBar  
    }  
});
```

Ứng dụng của JToolBar trong giao diện người dùng

Bài Tập

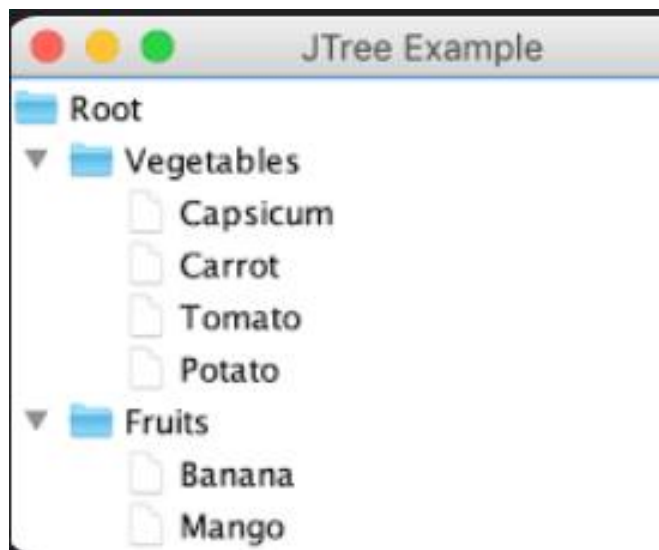
Câu 01: Hãy trình bày ứng dụng của Jmenu và Jtoolbar trong giao diện người dùng.

Câu 02: Hãy Viết code để tạo các cây sau đây:

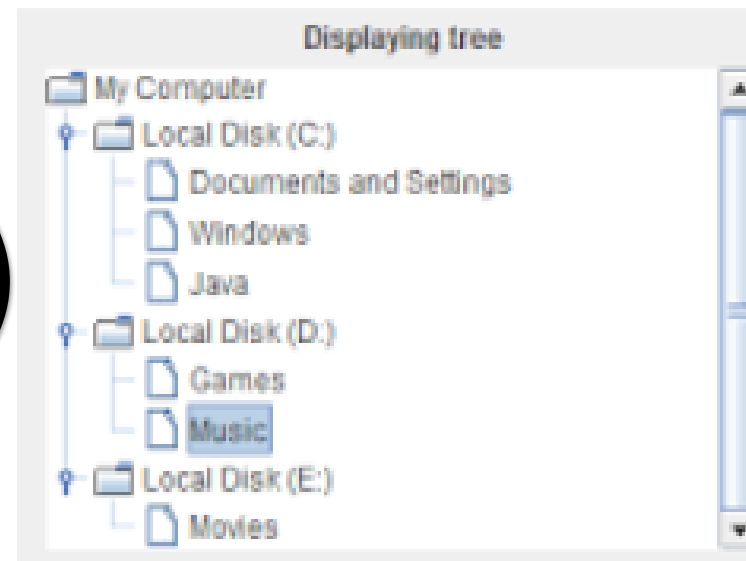
a)



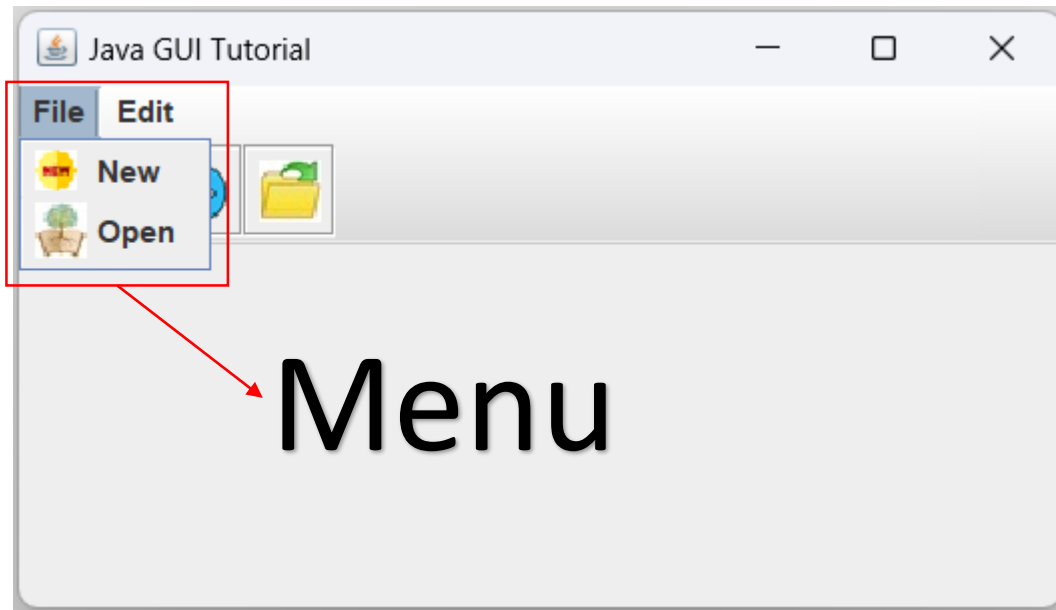
b)



c)



Câu 03: Hãy Viết code
tạo menu cho Frame



Câu 04: Hãy Viết code
tạo toolbar cho Frame

