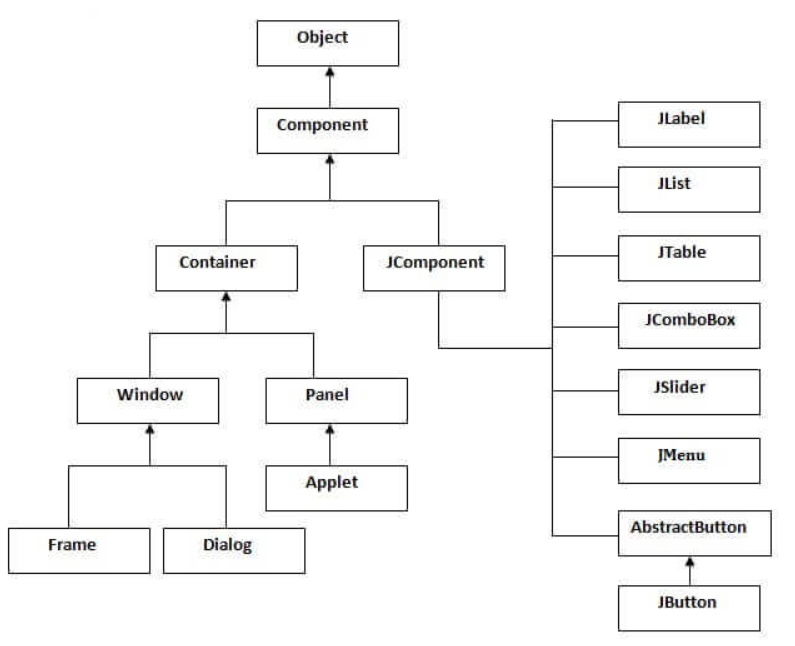
**JAVA SWING**

1. Giới thiệu chung

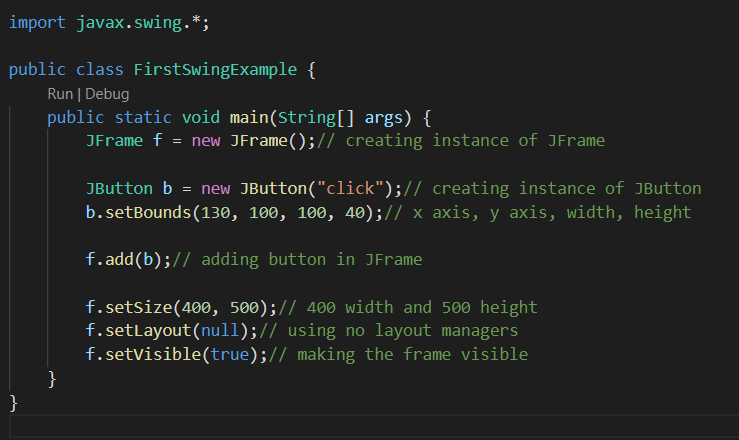
* Java Foundation Classes (JFC) là một tập hợp các giao diện người dùng đồ họa (GUI) thành phần cho các ứng dụng Java mà phần mềm tinh giản và ứng dụng điện toán đám mây phát triển. JFC chứa Abstract Window Toolkit (AWT), Java 2D và Swing.
* Java Swing là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng window-based. Nó được xây dựng trên API AWT (Abstract Windowing Toolkit) và được viết hoàn toàn bằng Java.
* Gói javax.swing cung cấp các lớp cho java swing API như JButton, JTextField, JTextArea, JRadioButton, JCheckBox, JMenu, JColorChooser, v.v.
* Hệ thống phân cấp của API java swing



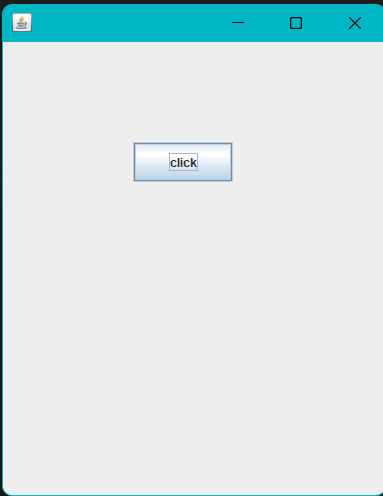
* Các phương thức của lớp Component được sử dụng rộng rãi trong java swing

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| public void add(Component c) | thêm một thành phần vào thành phần khác. |
| public void setSize(int width, int height) | thiết lập kích thước của thành phần. |
| public void setLayout(LayoutManager m) | thiết lập trình quản lý bố cục (layout) cho thành phần. |
| public void setVisible(boolean b) | thiết lập khả năng hiển thị của thành phần. Nó theo mặc định là false (ẩn) |

* Có hai cách để tạo khung (Frame):
* Bằng cách tạo đối tượng của lớp JFrame.
* Bằng cách kế thừa lớp JFrame.
* Chúng ta có thể viết code của Swing bên trong hàm main(), constructor hoặc bất kỳ phương thức nào khác.
* Ví dụ:

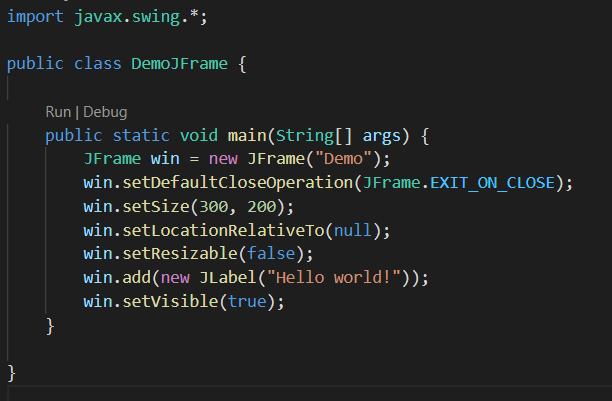


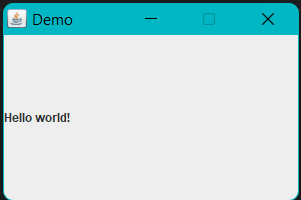
* Kết quả:



1. JFrame

* Container là thành phần chủ chốt trong các thành phần của SWING GUI. Một Container cung cấp một không gian, là nơi đặt một component. Một Container trong AWT chính là một Component và nó có thêm khả năng để thêm các component khác vào chính nó.
* Khi xem xét về Container, bạn cần chú ý các điểm sau:
* Các lớp con của Container được gọi là Container. Một số ví dụ về các lớp con của Container là JPanel, JFrame và JWindow.
* Container chỉ có thể thêm Component vào chính nó.
* Một layout mặc định có mặt trong mỗi container. Layout này có thể bị ghi đè bởi sử dụng phương thức setLayout().
* Lớp javax.swing.JFrame là một loại container kế thừa lớp java.awt.Frame. JFrame hoạt động giống như cửa sổ chính, nơi các thành phần như label, button, textfield được thêm vào để tạo GUI.
* Không giống như Frame, JFrame có tùy chọn ẩn hoặc đóng cửa sổ với sự trợ giúp của phương thức setDefaultCloseOperation (int).
* Constructor
* JFrame()
* JFrame(title: String)
* Method:
* void setSize(width: int, height: int)
* void setLocation(x: int, y: int)
* void setVisible(visible: boolean)
* void setDefaultCloseOperation(mode: int)
* void setLocationRelativeTo (c: Component)
* Ví dụ



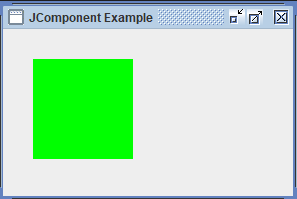
* Kết quả
* 

1. JComponent

* Lớp JComponent là lớp cơ sở của tất cả các Swing component ngoại trừ container cấp cao. Các Swing component có tên bắt đầu bằng "J" là con của lớp JComponent. Ví dụ: JButton, JScrollPane, JPanel, JTable, v.v. Nhưng, JFrame và JDialog không kế thừa lớp JComponent vì chúng là con của container cấp cao.
* Lớp JComponent extends lớp Container mà chính nó extends Component
* Ví dụ:



* Kết quả

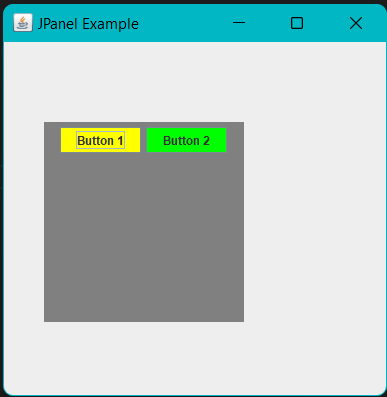


1. JPanel

* JPanel là một lớp container đơn giản nhất. Nó cung cấp không gian trong đó ứng dụng có thể đính kèm bất kỳ component nào khác. Nó kế thừa lớp JComponents.
* Nó không có thanh tiêu đề.
* Constructor:
* JPanel()
* JPanel(LayoutManager lm): Nó được sử dụng để tạo một JPanel mới với layout manager được chỉ định.
* Ví dụ



* Kết quả:

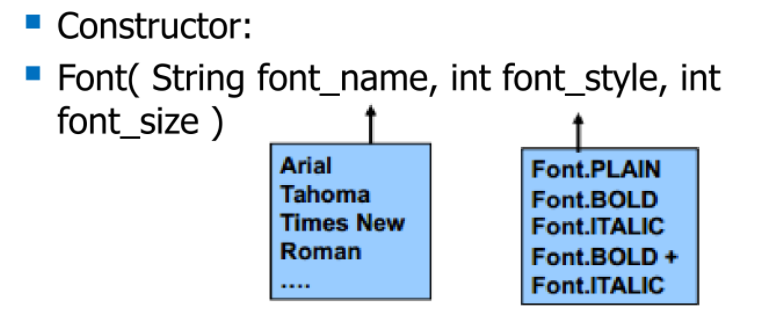


1. JLabel

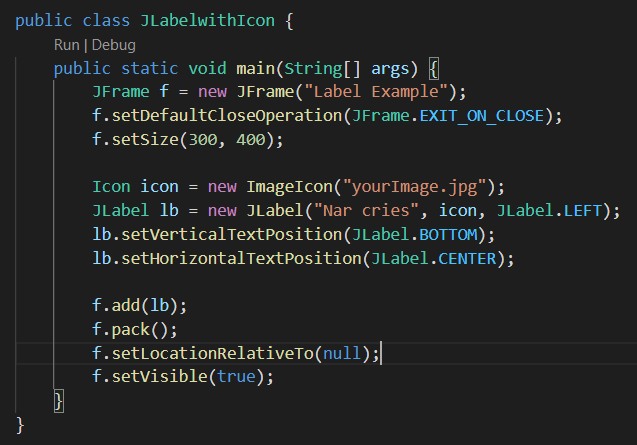
* Đối tượng của lớp JLabel là một component để đặt text trong một container. Nó được sử dụng để hiển thị một dòng văn bản chỉ đọc. Ứng dụng có thể thay đổi văn bản nhưng người dùng không thể chỉnh sửa trực tiếp. Nó kế thừa lớp JComponent.
* Constructor

|  |  |
| --- | --- |
| **Constructor** | **Mô tả** |
| JLabel() | Tạo một label instance rỗng |
| JLabel(String text) | Tạo một label instance với text được chỉ định |
| JLabel(String text, int horizontalAlignment) | Tạo một label với căn chỉnh được chỉ định, căn chỉnh có thể là left, right, center, leading hoặc trailing |
| JLabel(Icon i) | Tạo một label instance với hình ảnh được chỉ định |
| JLabel(String str, Icon img, int horizontalAlignment) | Tạo một JLabel instance với text, image, và alignment được chỉ định. |

* Method:
* setText(String label), getText()
* Font getFont()
* void setFont(Font font)
* Font cho text: Để draw kí tự ở một font, trước tiên bạn phải tạo một object của lớp Font:



* Ví dụ JLabelwithIcon



* Kết quả:



1. JTextField

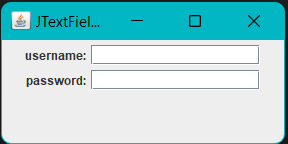
* Đối tượng của lớp JTextField là một component văn bản cho phép chỉnh sửa text dòng đơn. Nó kế thừa lớp JTextComponent
* Constructor:

|  |  |
| --- | --- |
| **Constructor** | **Mô tả** |
| JTextField() | Tạo 1 TextField mới |
| JTextField(String text) | Tạo 1 TextField mới khởi tạo với text được chỉ định |
| JTextField(String text, int columns) | Tạo một TextField mới được khởi tạo với text và cột được chỉ định |
| JTextField(int columns) | Tạo một TextField trống mới với số cột được chỉ định. |

* Method:
* String getText()
* void setText(String text)
* void setFont(Font font)
* void setEditable(boolean b): quyết định người dùng có được edit content hay không
* Ví dụ:

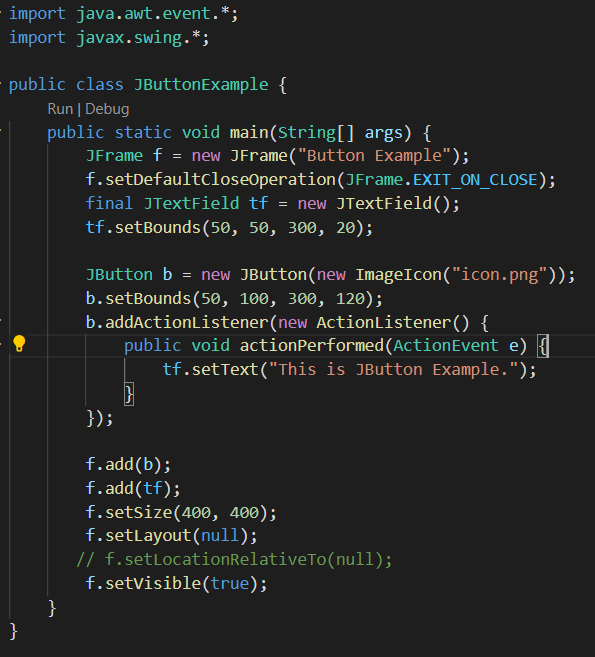


* Kết quả:

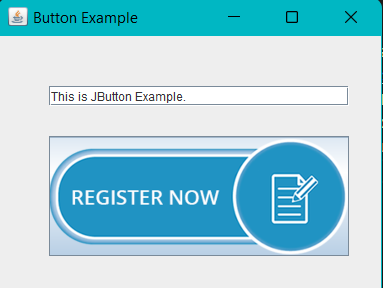


1. JButton

* Lớp JButton được sử dụng để tạo một button được gắn nhãn có triển khai độc lập với nền tảng. Ứng dụng dẫn đến một số hành động khi button được nhấn. Nó kế thừa lớp AbstractButton
* Constructor:
* JButton()
* JButton(String s)
* JButton(Icon i)
* JButton(String s, Icon i)
* Method:
* void setText(String s), String getText()
* void setIcon(Icon b), Icon getIcon()
* void setEnabled(boolean b): Nó được sử dụng để bật hoặc tắt nút.
* void setMnemonic(int a): Nó được sử dụng để đặt ghi nhớ trên nút.
* void addActionListener(ActionListener a): Nó được sử dụng để thêm action listener vào đối tượng này
* ActionListener:
* Java ActionListener được thông báo bất cứ khi nào bạn nhấp vào button hoặc mục menu. Nó được thông báo chống lại ActionEvent. Giao diện ActionListener được tìm thấy trong gói java.awt.event. Nó chỉ có một phương thức: actionPerformed ()
* Phương thức actionPerformed () được gọi tự động bất cứ khi nào bạn nhấp vào component đã đăng ký
* Ví dụ JButton:

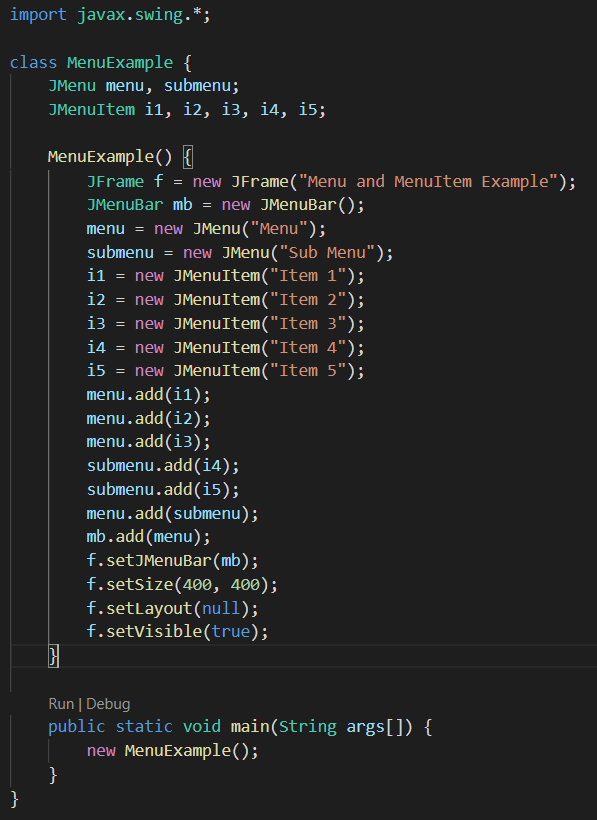


* Kết quả

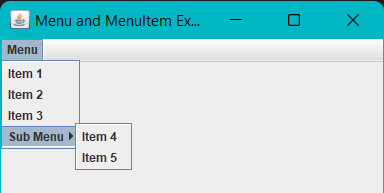


1. JMenuBar, JMenu, JMenuItem

* Lớp JMenuBar được sử dụng để hiển thị menu bar trên window hoặc frame. Nó có thể có một số menu. Nó kế thừa lớp JComponent
* Đối tượng của lớp JMenu là một component menu kéo xuống được hiển thị từ menu bar. Lớp JMenu kế thừa lớp JMenuItem.
* Đối tượng của lớp JMenuItem thêm một menu item có label đơn giản. Các item(mục) được sử dụng trong menu phải thuộc về JMenuItem hoặc bất kỳ lớp con nào của nó. Lớp JMenuItem kế thừa lớp AbstractButton
* Khai báo JMenuBar class
* Ví dụ:



* Kết quả



1. JComboBox

* JComboBox hiển thị popup menu, popup menu đó hiển thị 1 danh sách và người dùng có thể chọn một tùy chọn từ danh sách được chỉ định đó. JComboBox có thể chỉnh sửa hoặc chỉ đọc tùy thuộc vào sự lựa chọn của người lập trình. Nó kế thừa lớp JComponent.
* Constructor

|  |  |
| --- | --- |
| **Constructor** | **Mô tả** |
| JComboBox() | Tạo JComboBox với mô hình dữ liệu mặc định. |
| JComboBox(Object[] items) | Tạo một JComboBox chứa các phần tử trong mảng được chỉ định. |
| JComboBox(Vector<?> items) | Tạo một JComboBox chứa các phần tử trong Vectơ được chỉ định. |

* Method:

|  |  |
| --- | --- |
| **Methods** | **Mô tả** |
| void addItem(Object anObject) | Thêm một mục vào danh sách mục. |
| void removeItem(Object anObject) | Xóa một mục trong danh sách mục. |
| void removeAllItems() | xóa tất cả các mục khỏi danh sách |
| void setEditable(boolean b) | Xác định xem JComboBox có thể chỉnh sửa được hay không |
| void addActionListener(ActionListener a) | Thêm ActionListener. |
| void addItemListener(ItemListener i) | Thêm [ItemListener](https://www.javatpoint.com/java-itemlistener). |

* Ví dụ:



* Kết quả

