

THAM KHẢO THÊM VỀ JAVA AWT

AWT trong Java

Java AWT (Abstract Window Toolkit) là một API để phát triển các ứng dụng dựa trên GUI hoặc cửa sổ trong java.

Các thành phần Java AWT là các thành phần phụ thuộc vào nền tảng được hiển thị theo giao diện của hệ điều hành. AWT có nghĩa là các thành phần của nó sử dụng các tài nguyên của hệ điều hành.

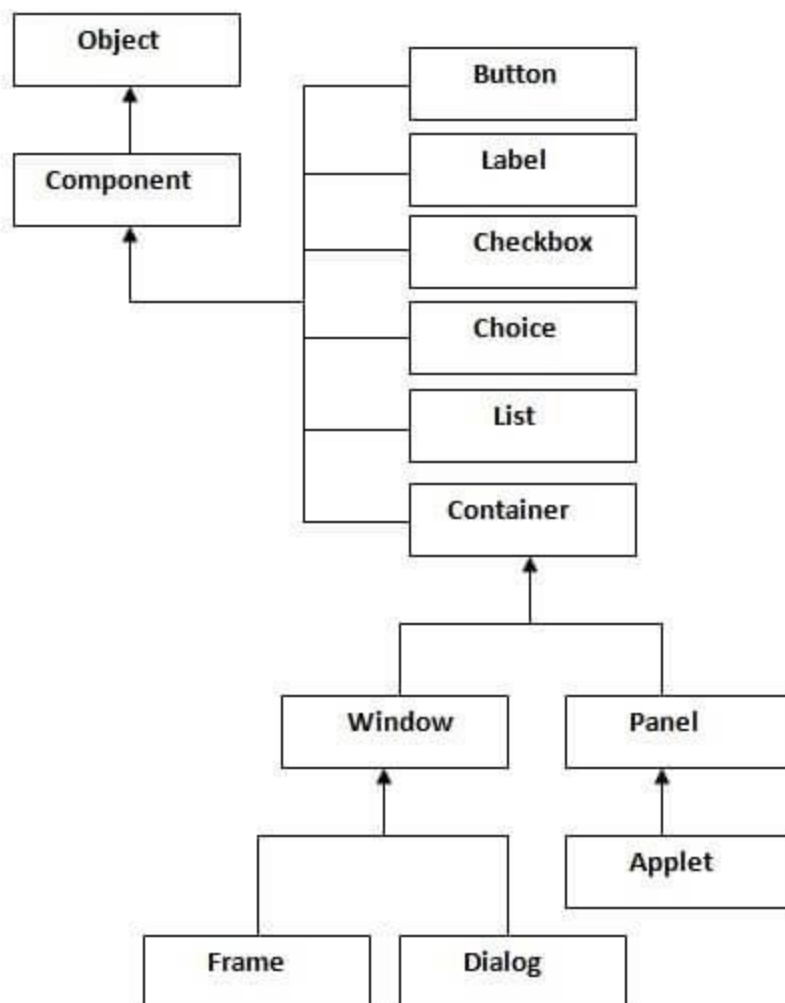
Gói java.awt cung cấp các lớp cho api AWT như TextField, Label, TextArea, RadioButton, CheckBox, Choice, List, v.v.

Nội dung chính

- [Phân cấp Java AWT](#)
 - [Container](#)
 - [Window](#)
 - [Panel](#)
 - [Frame](#)
- [Các phương thức hữu ích của lớp Component](#)
- [Ví dụ về AWT trong java](#)
 - [Ví dụ về AWT trong java - kế thừa lớp Frame](#)
 - [Ví dụ về AWT trong java - tạo thể hiện của lớp Frame](#)

Phân cấp Java AWT

Hệ thống phân cấp của các lớp Java AWT được đưa ra dưới đây.



Container

Vùng chứa (Container) là thành phần trong AWT mà chứa các thành phần khác như button, textfield, label, ... Các lớp được kế thừa lớp Container được biết đến như các container như Frame, Dialog, Panel.

Window

The window is the container that have no borders and menu bars. You must use frame, dialog or another window for creating a window.

Panel

Panel là vùng chứa mà không chứa thanh tiêu đề và thanh menu. Nó có thể có các thành phần như button, textfield, ...

Frame

Khung (Frame) là vùng chứa thanh tiêu đề và có thể có các thanh trình đơn. Nó có thể có các thành phần khác như button, textfield, ...

Các phương thức hữu ích của lớp Component

Phương thức	Mô tả
<code>public void add(Component c)</code>	chèn một thành phần vào thành phần this.
<code>public void setSize(int width,int height)</code>	Thiết lập kích thước (chiều rộng và chiều cao) của thành phần.
<code>public void setLayout(LayoutManager m)</code>	Thiết lập trình quản lý bố cục cho thành phần.
<code>public void setVisible(boolean status)</code>	Thiết lập ẩn / hiện thành phần, mặc định false (ẩn).

Ví dụ về AWT trong java

Để tạo ví dụ AWT đơn giản, bạn cần sử dụng Frame, có 2 cách để tạo Frame:

- Bởi kế thừa lớp Frame.
- Bởi việc tạo đối tượng của lớp Frame.

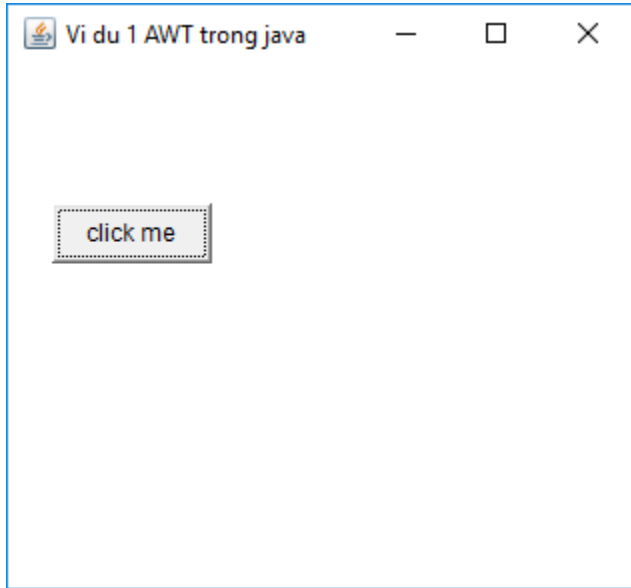
Ví dụ về AWT trong java - kế thừa lớp Frame

Trong ví dụ này, chúng ta sẽ hiển thị thành phần Button trên khung.

```
1    package vn.plpsoft;
2
3    import java.awt.Button;
4    import java.awt.Frame;
5
6    public class AWTExample1 extends Frame {
7        AWTExample1() {
8            setTitle("Vi du 1 AWT trong java");
9            Button b = new Button("click me");
10           b.setBounds(30, 100, 80, 30); // cai dat vi tri button
11           add(b); // them doi tuong button vao frame
12           setSize(300, 300); // kích thước frame với width = 300 và
```

```
13     height = 300
14         setLayout(null); // không trình quản lý layout
15         setVisible(true); // hiển thị frame
16     }
17
18     public static void main(String args[]) {
19         new AWTEExample1();
20     }
```

Kết quả:



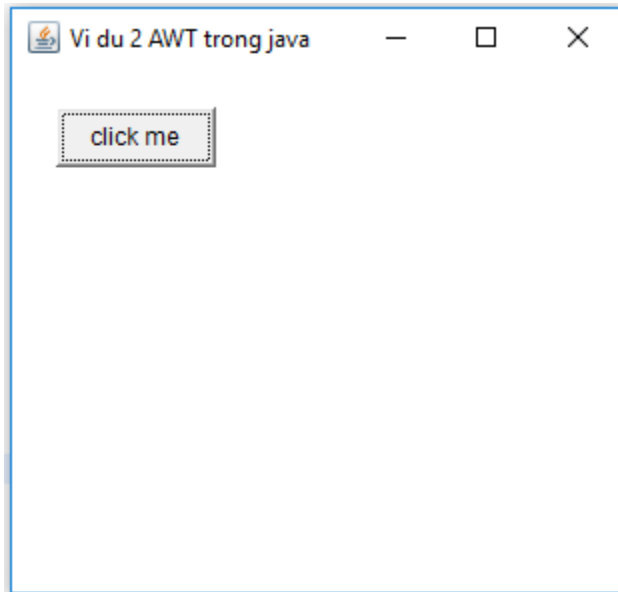
Ví dụ về AWT trong java - tạo thể hiện của lớp Frame

Ví dụ tạo thể hiện của lớp Frame:

```
1     package vn.plpsoft;
2
3     import java.awt.Button;
4     import java.awt.Frame;
5
6     public class AWTEExample2 {
7         AWTEExample2() {
8             Frame f = new Frame();
9             f.setTitle("Ví dụ 2 AWT trong java");
10            Button b = new Button("click me");
11            b.setBounds(30, 50, 80, 30);
12            f.add(b);
13            f.setSize(300, 300);
14            f.setLayout(null);
15            f.setVisible(true);
16        }
17    }
```

```
18     public static void main(String args[]) {  
19         new AWTEExample2();  
20     }  
21 }
```

Kết quả:



Xử lý sự kiện Java AWT

Trạng thái của một đối tượng bị thay đổi được gọi là một sự kiện (event). Ví dụ, bấm vào nút, kéo chuột, vv Gói java.awt.event cung cấp nhiều lớp sự kiện và interface Listener để xử lý sự kiện Java AWT.

Nội dung chính

- [Các lớp Java Event và các interface Listener](#)
- [Các bước để thực hiện xử lý sự kiện trong Java AWT](#)
 - [Phương thức đăng ký](#)
- [Ví dụ xử lý sự kiện trong Java AWT](#)
 - [1. Xử lý sự kiện Java AWT - bên trong lớp hiện tại](#)
 - [Xử lý sự kiện Java AWT - bên trong lớp hiện khác](#)
 - [Xử lý sự kiện Java AWT - bên trong lớp kế thừa](#)

Các lớp Java Event và các interface Listener

Các lớp Event	Các interface Listener
ActionEvent	ActionListener
MouseEvent	MouseListener and MouseMotionListener
MouseWheelEvent	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener
AdjustmentEvent	AdjustmentListener
WindowEvent	WindowListener
ComponentEvent	ComponentListener
ContainerEvent	ContainerListener
FocusEvent	FocusListener

Các bước để thực hiện xử lý sự kiện trong Java AWT

Các bước sau được yêu cầu để thực hiện xử lý sự kiện:

1. Đăng ký thành phần với trình Listener.

Phương thức đăng ký

Để đăng ký thành phần với trình Listener, có nhiều lớp cung cấp các phương thức đăng ký. Ví dụ:

- **Button**
 - `public void addActionListener(ActionListener a){}`
- **MenuItem**

- `public void addActionListener(ActionListener a){}`
- **TextField**
 - `public void addActionListener(ActionListener a){}`
 - `public void addTextListener(TextListener a){}`
- **TextArea**
 - `public void addTextListener(TextListener a){}`
- **Checkbox**
 - `public void addItemListener(ItemListener a){}`
- **Choice**
 - `public void addItemListener(ItemListener a){}`
- **List**
 - `public void addActionListener(ActionListener a){}`
 - `public void addItemListener(ItemListener a){}`

Ví dụ xử lý sự kiện trong Java AWT

Chúng tôi có thể đặt mã xử lý sự kiện vào một trong các vị trí sau:

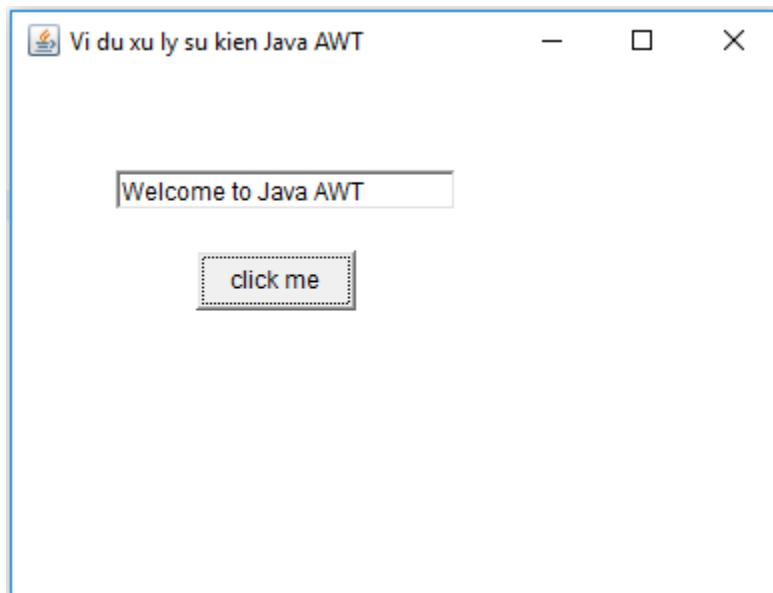
1. Bên trong lớp hiện tại.
2. Bên trong lớp khác.
3. Bên trong lớp nặc danh.

1. Xử lý sự kiện Java AWT - bên trong lớp hiện tại

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Button;
4 import java.awt.Frame;
5 import java.awt.TextField;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8
9 public class AWTEventHandling1 extends Frame implements ActionListener {
10     private TextField textField;
11
12     public AWTEventHandling1() {
13
14         // tạo các thành phần
15         textField = new TextField();
16         textField.setBounds(60, 80, 170, 20);
17         Button button = new Button("click me");
18         button.setBounds(100, 120, 80, 30);
```

```
19
20     // đăng ký trình listener
21     button.addActionListener(this);
22
23     // thêm thành phần, kích thước, layout, khả năng hiển thị
24     setTitle("Vi du xu ly su kien Java AWT");
25     add(button);
26     add(textField);
27     setSize(400, 300);
28     setLayout(null);
29     setVisible(true);
30 }
31
32 @Override
33 public void actionPerformed(ActionEvent e) {
34     textField.setText("Welcome to Java AWT");
35 }
36
37 public static void main(String args[]) {
38     new AWTEventHandling1();
39 }
40 }
```

Kết quả:



Xử lý sự kiện Java AWT - bên trong lớp hiện khác

Tạo lớp view: AWTVView.java

```
1     package vn.plpsoft.awt;
2
3     import java.awt.Button;
4     import java.awt.Frame;
```



```
5    import java.awt.TextField;
6
7    public class AWTView extends Frame {
8        TextField textField;
9
10       AWTView() {
11           // tạo các thành phần
12           textField = new TextField();
13           textField.setBounds(60, 80, 170, 20);
14           Button button = new Button("click me");
15           button.setBounds(100, 120, 80, 30);
16           // đăng ký listener
17           Controller obj = new Controller(this);
18           button.addActionListener(obj);
19           // thêm thành phần, kích thước, layout, khả năng hiển thị
20           setTitle("Vi du xu ly su kien Java AWT");
21           add(button);
22           add(textField);
23           setSize(400, 300);
24           setLayout(null);
25           setVisible(true);
26       }
27   }
```

Tạo lớp controller: Controller.java

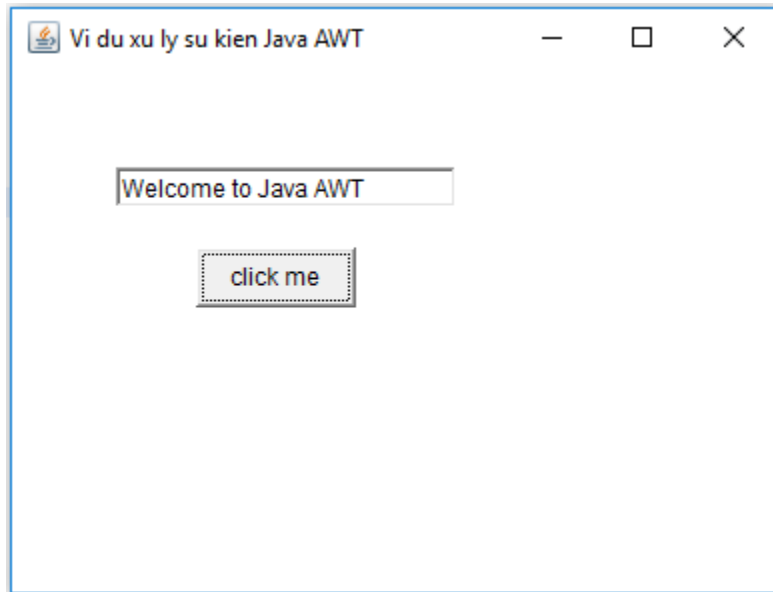
```
1    package vn.plpsoft.awt;
2
3    import java.awt.event.ActionEvent;
4    import java.awt.event.ActionListener;
5
6    public class Controller implements ActionListener {
7        AWTView obj;
8
9        Controller(AWTView obj) {
10            this.obj = obj;
11        }
12
13        @Override
14        public void actionPerformed(ActionEvent e) {
15            obj.textField.setText("Welcome to Java AWT");
16        }
17    }
```

Tạo lớp AWTEventHandling2.java

```
1    package vn.plpsoft.awt;
2
3    public class AWTEventHandling2 {
```

```
4     public static void main(String[] args) {  
5         AWTView view = new AWTView();  
6     }  
7 }
```

Kết quả:

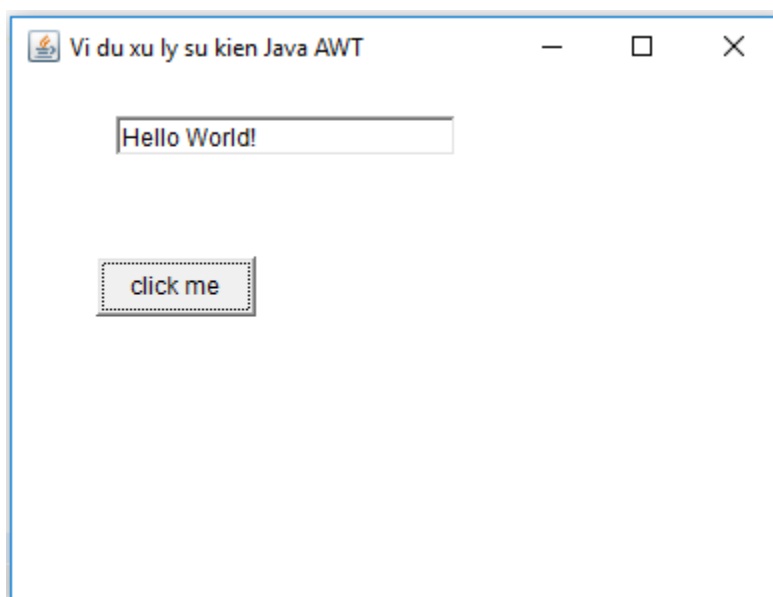


Xử lý sự kiện Java AWT - bên trong lớp nặc danh

```
1     package vn.plpsoft.awt;  
2  
3     import java.awt.Button;  
4     import java.awt.Frame;  
5     import java.awt.TextField;  
6     import java.awt.event.ActionEvent;  
7     import java.awt.event.ActionListener;  
8  
9     public class AWTEventHandling3 extends Frame {  
10         TextField textField;  
11  
12         AWTEventHandling3() {  
13             textField = new TextField();  
14             textField.setBounds(60, 50, 170, 20);  
15             Button button = new Button("click me");  
16             button.setBounds(50, 120, 80, 30);  
17  
18             button.addActionListener(new ActionListener() {  
19                 @Override  
20                 public void actionPerformed(ActionEvent e) {  
21                     textField.setText("Hello World!");  
22                 }  
23             })  
24         }  
25     }
```

```
24     });  
25     setTitle("Vi du xu ly su kien Java AWT");  
26     add(button);  
27     add(textField);  
28     setSize(400, 300);  
29     setLayout(null);  
30     setVisible(true);  
31 }  
32  
33 public static void main(String args[]) {  
34     new AWTEventHandling3();  
35 }  
36 }
```

Kết quả:



Button trong Java AWT

Lớp Button trong Java AWT được sử dụng để tạo button được gắn nhãn có triển khai độc lập nền tảng.

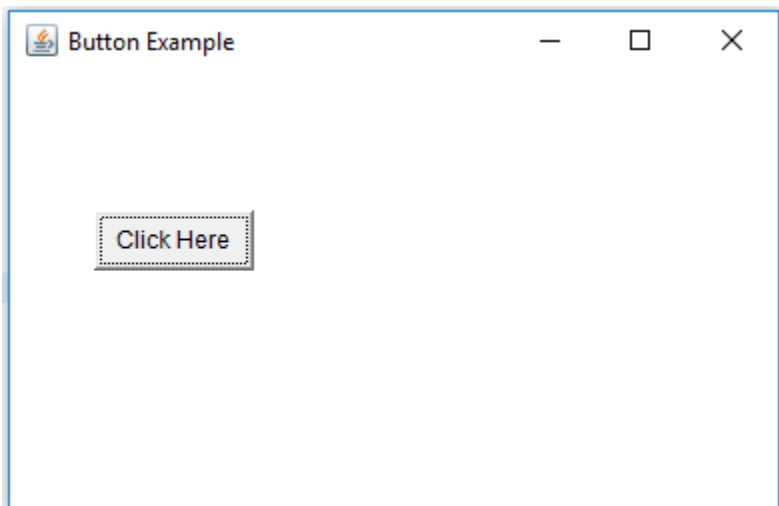
Khai báo lớp AWT Button

```
1 public class Button extends Component implements Accessible
```

Ví dụ Button trong Java AWT

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Button;
4    import java.awt.Frame;
5
6    public class ButtonExample1 {
7        public static void main(String[] args) {
8            Frame f = new Frame("Button Example");
9            Button b = new Button("Click Here");
10           b.setBounds(50, 100, 80, 30);
11           f.add(b);
12           f.setSize(400, 200);
13           f.setLayout(null);
14           f.setVisible(true);
15       }
16   }
```

Kết quả:

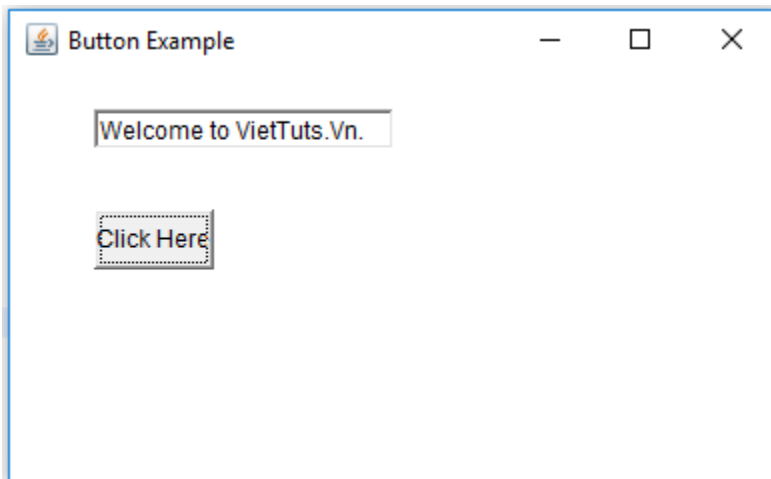


Ví dụ Button trong Java AWT với ActionListener

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Button;
4    import java.awt.Frame;
5    import java.awt.TextField;
6    import java.awt.event.ActionEvent;
```

```
7    import java.awt.event.ActionListener;
8
9    public class ButtonExample2 {
10        public static void main(String[] args) {
11            Frame f = new Frame("Button Example");
12            final TextField tf = new TextField();
13            tf.setBounds(50, 50, 150, 20);
14            Button b = new Button("Click Here");
15            b.setBounds(50, 100, 60, 30);
16            b.addActionListener(new ActionListener() {
17                public void actionPerformed(ActionEvent e) {
18                    tf.setText("Welcome to Plpsoft.Vn.");
19                }
20            });
21            f.add(b);
22            f.add(tf);
23            f.setSize(400, 200);
24            f.setLayout(null);
25            f.setVisible(true);
26        }
27    }
```

Kết quả:



Label trong Java AWT

Đối tượng của lớp **Label** là một thành phần để đặt văn bản trong một vùng chứa. Nó được sử dụng để hiển thị một dòng văn bản chỉ đọc. Văn bản có thể được thay đổi bởi một ứng dụng nhưng người dùng không thể chỉnh sửa trực tiếp.

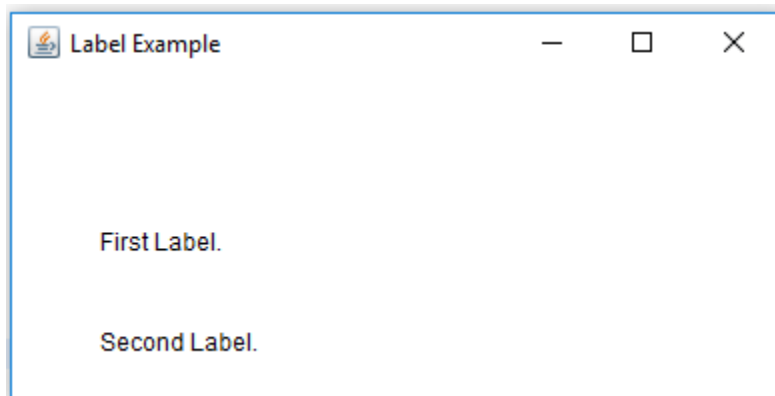
Khai báo lớp AWT Label

```
1 public class Label extends Component implements Accessible
```

Ví dụ Label trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.Label;
5
6 public class LabelExample1 {
7     public static void main(String args[]) {
8         Frame f = new Frame("Label Example");
9         Label l1, l2;
10        l1 = new Label("First Label.");
11        l1.setBounds(50, 100, 100, 30);
12        l2 = new Label("Second Label.");
13        l2.setBounds(50, 150, 100, 30);
14        f.add(l1);
15        f.add(l2);
16        f.setSize(400, 200);
17        f.setLayout(null);
18        f.setVisible(true);
19    }
20 }
```

Kết quả:

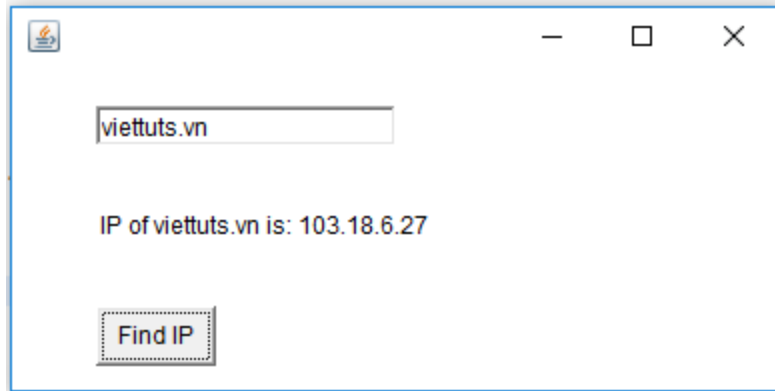


Ví dụ Label trong Java AWT với ActionListener

```
1 package vn.plpsoft.awt;
```

```
2
3 import java.awt.Button;
4 import java.awt.Frame;
5 import java.awt.Label;
6 import java.awt.TextField;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 public class LabelExample2 extends Frame implements ActionListener {
11     private TextField textField;
12     private Label label;
13     private Button button;
14
15     public LabelExample2() {
16         textField = new TextField();
17         textField.setBounds(50, 50, 150, 20);
18         label = new Label();
19         label.setBounds(50, 100, 250, 20);
20         button = new Button("Find IP");
21         button.setBounds(50, 150, 60, 30);
22         button.addActionListener(this);
23         add(button);
24         add(textField);
25         add(label);
26         setSize(400, 200);
27         setLayout(null);
28         setVisible(true);
29     }
30
31     @Override
32     public void actionPerformed(ActionEvent e) {
33         try {
34             String host = textField.getText();
35             String ip = java.net.InetAddress.getByName(host).getHostAddress();
36             label.setText("IP of " + host + " is: " + ip);
37         } catch (Exception ex) {
38             System.out.println(ex);
39         }
40     }
41
42     public static void main(String[] args) {
43         new LabelExample2();
44     }
45 }
```

Kết quả:



TextField trong Java AWT

ĐỐI TƯỢNG của một lớp **TextField** là một thành phần văn bản cho phép chỉnh sửa văn bản một dòng. Nó **KẾ THỪA** lớp **TextComponent**.

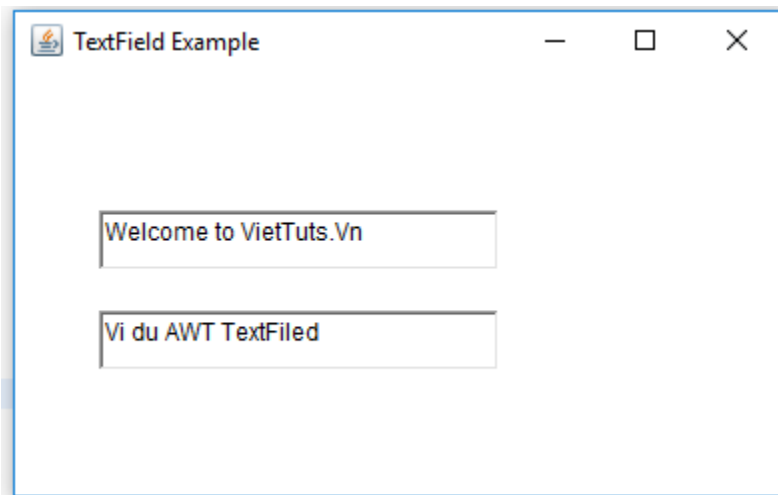
Khai báo lớp AWT TextField

```
1 public class TextField extends TextComponent
```

Ví dụ TextField trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.TextField;
5
6 public class TextFieldExample1 {
7     public static void main(String args[]) {
8         Frame f = new Frame("TextField Example");
9         TextField t1, t2;
10        t1 = new TextField("Welcome to Plpsoft.Vn");
11        t1.setBounds(50, 100, 200, 30);
12        t2 = new TextField("Vi du AWT TextFiled");
13        t2.setBounds(50, 150, 200, 30);
14        f.add(t1);
15        f.add(t2);
16        f.setSize(400, 250);
17        f.setLayout(null);
18        f.setVisible(true);
19    }
20 }
```


Kết quả:

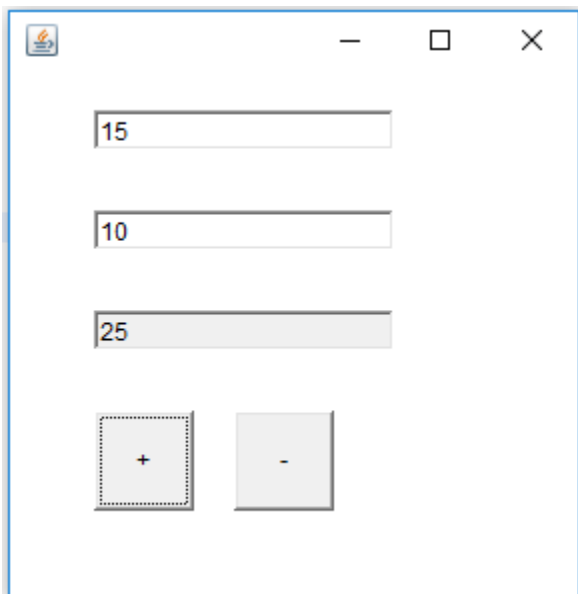


Ví dụ TextField trong Java AWT với ActionListener

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Button;
4    import java.awt.Frame;
5    import java.awt.TextField;
6    import java.awt.event.ActionEvent;
7    import java.awt.event.ActionListener;
8
9    public class TextFieldExample2 extends Frame implements ActionListener {
10        TextField textField1, textField2, textField3;
11        Button button1, button2;
12
13        TextFieldExample2() {
14            textField1 = new TextField();
15            textField1.setBounds(50, 50, 150, 20);
16            textField2 = new TextField();
17            textField2.setBounds(50, 100, 150, 20);
18            textField3 = new TextField();
19            textField3.setBounds(50, 150, 150, 20);
20            textField3.setEditable(false);
21            button1 = new Button("+");
22            button1.setBounds(50, 200, 50, 50);
23            button2 = new Button("-");
24            button2.setBounds(120, 200, 50, 50);
25            button1.addActionListener(this);
26            button2.addActionListener(this);
27            add(textField1);
28            add(textField2);
29            add(textField3);
```

```
30     add(button1);
31     add(button2);
32     setSize(300, 300);
33     setLayout(null);
34     setVisible(true);
35 }
36
37 public void actionPerformed(ActionEvent e) {
38     String s1 = textField1.getText();
39     String s2 = textField2.getText();
40     int a = Integer.parseInt(s1);
41     int b = Integer.parseInt(s2);
42     int c = 0;
43     if (e.getSource() == button1) {
44         c = a + b;
45     } else if (e.getSource() == button2) {
46         c = a - b;
47     }
48     String result = String.valueOf(c);
49     textField3.setText(result);
50 }
51
52 public static void main(String[] args) {
53     new TextFieldExample2();
54 }
55 }
```

Kết quả:



TextArea trong Java AWT

Đối tượng của một lớp **TextArea** là một vùng nhiều dòng để hiển thị văn bản. Nó cho phép chỉnh sửa văn bản nhiều dòng. Nó kế thừa lớp **TextComponent**.

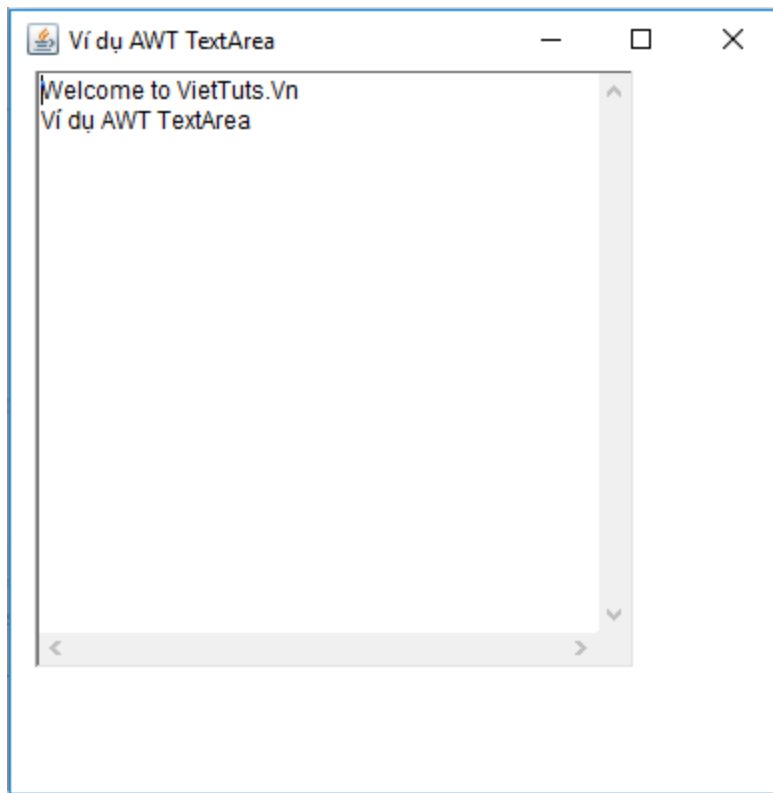
Khai báo lớp AWT TextArea

```
1 public class TextArea extends TextComponent
```

Ví dụ TextArea trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.TextArea;
5
6 public class TextAreaExample1 {
7
8     public TextAreaExample1() {
9         Frame f = new Frame();
10        TextArea area = new TextArea("Welcome to Plpsoft.Vn\n"
11            + "Ví dụ AWT TextArea");
12        area.setBounds(20, 30, 300, 300);
13        f.setTitle("Ví dụ AWT TextArea");
14        f.add(area);
15        f.setSize(400, 400);
16        f.setLayout(null);
17        f.setVisible(true);
18    }
19
20    public static void main(String args[]) {
21        new TextAreaExample1();
22    }
23 }
```

Kết quả:

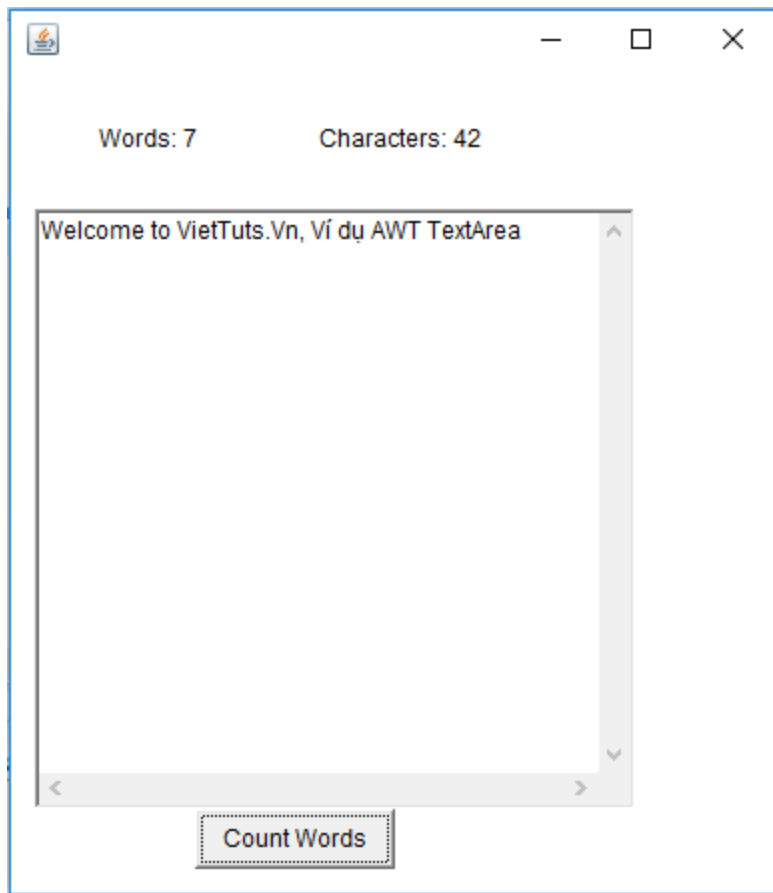


Ví dụ TextArea trong Java AWT với ActionListener

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Button;
4    import java.awt.Frame;
5    import java.awt.Label;
6    import java.awt.TextArea;
7    import java.awt.event.ActionEvent;
8    import java.awt.event.ActionListener;
9
10   public class TextAreaExample2 extends Frame implements ActionListener {
11       private Label label1, label2;
12       private TextArea textArea;
13       private Button button;
14
15       public TextAreaExample2() {
16           label1 = new Label();
17           label1.setBounds(50, 50, 100, 30);
18           label2 = new Label();
19           label2.setBounds(160, 50, 100, 30);
20           textArea = new TextArea();
21           textArea.setBounds(20, 100, 300, 300);
22           button = new Button("Count Words");
```

```
23     button.setBounds(100, 400, 100, 30);
24     button.addActionListener(this);
25     add(label1);
26     add(label2);
27     add(textArea);
28     add(button);
29     setSize(400, 450);
30     setLayout(null);
31     setVisible(true);
32 }
33
34 @Override
35 public void actionPerformed(ActionEvent e) {
36     String text = textArea.getText();
37     String words[] = text.split("\\s");
38     label1.setText("Words: " + words.length);
39     label2.setText("Characters: " + text.length());
40 }
41
42 public static void main(String[] args) {
43     new TextAreaExample2();
44 }
45 }
```

Kết quả:



CheckBox trong Java AWT

Lớp Checkbox được sử dụng để tạo một hộp kiểm. Nó được sử dụng để biến một tùy chọn on (true) hoặc off (false). Click vào một hộp kiểm thay đổi trạng thái của nó từ "on" thành "off" hoặc từ "off" thành "on".

Khai báo lớp AWT Checkbox

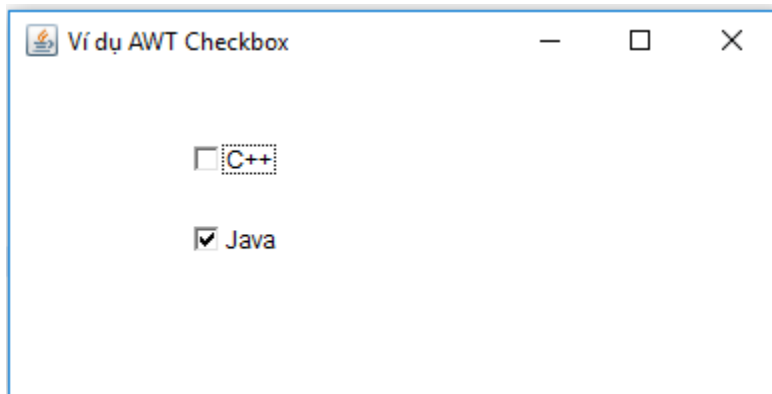
```
1 public class Checkbox extends Component implements ItemSelectable, Accessible
```

Ví dụ Checkbox trong Java AWT

```
1 package vn.plpsoft.awt;  
2  
3 import java.awt.Checkbox;  
4 import java.awt.Frame;
```

```
5
6 public class CheckBoxExample1 {
7     public CheckBoxExample1() {
8         Frame f= new Frame("Ví dụ AWT Checkbox");
9         Checkbox checkbox1 = new Checkbox("C++");
10        checkbox1.setBounds(100, 50, 50, 50);
11        Checkbox checkbox2 = new Checkbox("Java", true);
12        checkbox2.setBounds(100, 90, 50, 50);
13        f.add(checkbox1);
14        f.add(checkbox2);
15        f.setSize(400, 200);
16        f.setLayout(null);
17        f.setVisible(true);
18    }
19
20    public static void main(String args[]) {
21        new CheckBoxExample1();
22    }
23 }
```

Kết quả:

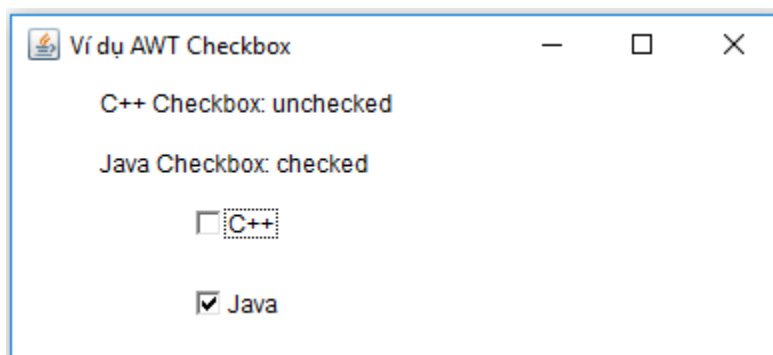


Ví dụ Checkbox trong Java AWT với ItemListener

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Checkbox;
4 import java.awt.Frame;
5 import java.awt.Label;
6 import java.awt.event.ItemEvent;
7 import java.awt.event.ItemListener;
8
9 public class CheckBoxExample2 {
10     public CheckBoxExample2() {
11         Frame f = new Frame("Ví dụ AWT Checkbox");
12         final Label label1 = new Label();
```

```
13     final Label label2 = new Label();
14     label1.setBounds(50, 20, 200, 50);
15     label2.setBounds(50, 50, 200, 50);
16     Checkbox checkbox1 = new Checkbox("C++");
17     checkbox1.setBounds(100, 80, 50, 50);
18     Checkbox checkbox2 = new Checkbox("Java");
19     checkbox2.setBounds(100, 120, 50, 50);
20     f.add(checkbox1);
21     f.add(checkbox2);
22     f.add(label1);
23     f.add(label2);
24     checkbox1.addItemListener(new ItemListener() {
25         public void itemStateChanged(ItemEvent e) {
26             label1.setText("C++ Checkbox: " + (e.getStateChange() == 1
27                 ? "checked" : "unchecked"));
28         }
29     });
30     checkbox2.addItemListener(new ItemListener() {
31         public void itemStateChanged(ItemEvent e) {
32             label2.setText("Java Checkbox: " + (e.getStateChange() == 1
33                 ? "checked" : "unchecked"));
34         }
35     });
36     f.setSize(400, 200);
37     f.setLayout(null);
38     f.setVisible(true);
39 }
40
41 public static void main(String args[]) {
42     new CheckBoxExample2();
43 }
44 }
```

Kết quả:



CheckBoxGroup trong Java AWT

Đối tượng của lớp `CheckboxGroup` được sử dụng để nhóm một tập hợp các [Checkbox](#) lại với nhau. Tại một thời điểm chỉ có một nút hộp kiểm được phép ở trạng thái "on" và các nút hộp kiểm còn lại ở trạng thái "off". Nó kế thừa [lớp Object](#).

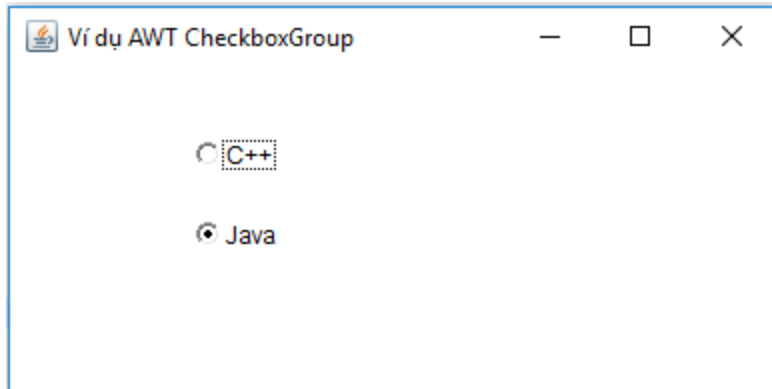
Khai báo lớp AWT `CheckboxGroup`

```
1 public class CheckboxGroup extends Object implements Serializable
```

Ví dụ `CheckboxGroup` trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Checkbox;
4 import java.awt.CheckboxGroup;
5 import java.awt.Frame;
6
7 public class CheckBoxExample1 {
8     public CheckBoxExample1() {
9         Frame f = new Frame("Ví dụ AWT CheckboxGroup");
10        CheckboxGroup cbg = new CheckboxGroup();
11        Checkbox checkbox1 = new Checkbox("C++", cbg, false);
12        checkbox1.setBounds(100, 50, 50, 50);
13        Checkbox checkbox2 = new Checkbox("Java", cbg, true);
14        checkbox2.setBounds(100, 90, 50, 50);
15        f.add(checkbox1);
16        f.add(checkbox2);
17        f.setSize(400, 200);
18        f.setLayout(null);
19        f.setVisible(true);
20    }
21
22    public static void main(String args[]) {
23        new CheckBoxExample1();
24    }
25 }
```

Kết quả:

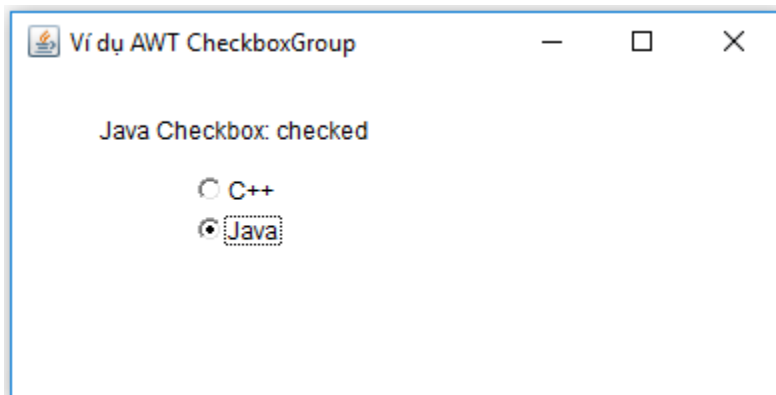


Ví dụ CheckboxGroup trong Java AWT với ItemListener

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Checkbox;
4    import java.awt.CheckboxGroup;
5    import java.awt.Frame;
6    import java.awt.Label;
7    import java.awt.event.ItemEvent;
8    import java.awt.event.ItemListener;
9
10   public class CheckBoxExample2 {
11       public CheckBoxExample2() {
12           Frame f = new Frame("Ví dụ AWT CheckboxGroup");
13           final Label label1 = new Label();
14           label1.setBounds(50, 50, 200, 20);
15           CheckboxGroup cbg = new CheckboxGroup();
16           Checkbox checkbox1 = new Checkbox("C++", cbg, false);
17           checkbox1.setBounds(100, 80, 50, 20);
18           Checkbox checkbox2 = new Checkbox("Java", cbg, true);
19           checkbox2.setBounds(100, 100, 50, 20);
20           f.add(checkbox1);
21           f.add(checkbox2);
22           f.add(label1);
23           checkbox1.addItemListener(new ItemListener() {
24               public void itemStateChanged(ItemEvent e) {
25                   label1.setText("C++ Checkbox: " + (e.getStateChange() == 1
26                       ? "checked" : "unchecked"));
27               }
28           });
29           checkbox2.addItemListener(new ItemListener() {
30               public void itemStateChanged(ItemEvent e) {
31                   label1.setText("Java Checkbox: " + (e.getStateChange() == 1
32                       ? "checked" : "unchecked"));
33               }
34           });
35       }
36   }
```

```
34     });  
35     f.setSize(400, 200);  
36     f.setLayout(null);  
37     f.setVisible(true);  
38 }  
39  
40 public static void main(String args[]) {  
41     new CheckBoxExample2();  
42 }  
43 }
```

Kết quả:



Choice trong Java AWT

Đối tượng của lớp Choice được sử dụng để hiển thị menu popup của các lựa chọn. Lựa chọn do người dùng lựa chọn được hiển thị ở đầu trình đơn. Nó kế thừa lớp Component.

Khai báo lớp AWT Choice

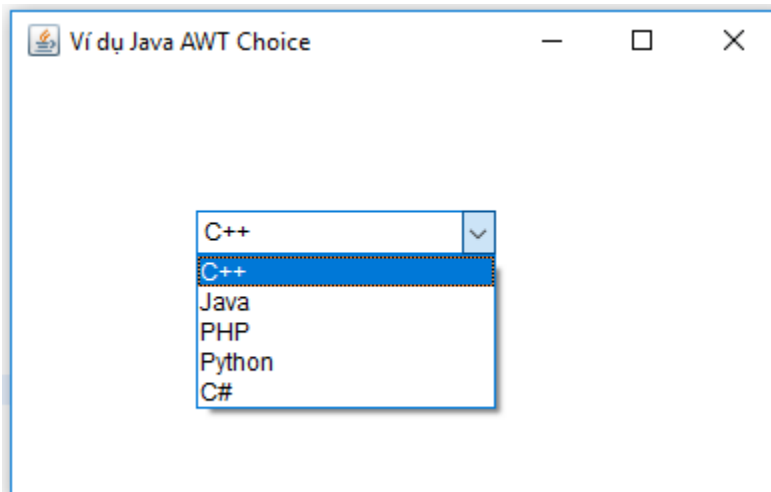
```
1 public class Choice extends Component implements ItemSelectable, Accessible
```

Ví dụ Choice trong Java AWT

```
1 package vn.plpsoft.awt;  
2  
3 import java.awt.Choice;  
4 import java.awt.Frame;  
5  
6 public class ChoiceExample1 {
```

```
7      public ChoiceExample1() {
8          Frame frame = new Frame("Ví dụ Java AWT Choice");
9          Choice choice = new Choice();
10         choice.setBounds(100, 100, 150, 150);
11         choice.add("C++");
12         choice.add("Java");
13         choice.add("PHP");
14         choice.add("Python");
15         choice.add("C#");
16         frame.add(choice);
17         frame.setSize(400, 250);
18         frame.setLayout(null);
19         frame.setVisible(true);
20     }
21
22     public static void main(String args[]) {
23         new ChoiceExample1();
24     }
25 }
```

Kết quả:

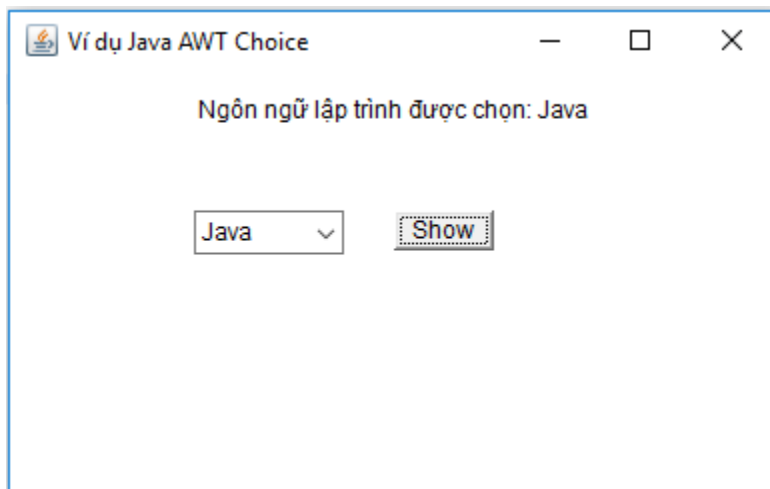


Ví dụ Choice trong Java AWT với ActionListener

```
1      package vn.plpsoft.awt;
2
3      import java.awt.Button;
4      import java.awt.Choice;
5      import java.awt.Frame;
6      import java.awt.Label;
7      import java.awt.event.ActionEvent;
8      import java.awt.event.ActionListener;
9
```

```
10 public class ChoiceExample2 {
11     public ChoiceExample2() {
12         Frame frame = new Frame("Ví dụ Java AWT Choice");
13         final Label label = new Label();
14         label.setAlignment(Label.CENTER);
15         label.setSize(400, 100);
16         Button button = new Button("Show");
17         button.setBounds(200, 100, 50, 20);
18         final Choice choice = new Choice();
19         choice.setBounds(100, 100, 75, 75);
20         choice.add("C++");
21         choice.add("Java");
22         choice.add("PHP");
23         choice.add("Python");
24         choice.add("C#");
25         frame.add(choice);
26         frame.add(label);
27         frame.add(button);
28         frame.setSize(400, 250);
29         frame.setLayout(null);
30         frame.setVisible(true);
31         button.addActionListener(new ActionListener() {
32             public void actionPerformed(ActionEvent e) {
33                 String data = "Ngôn ngữ lập trình được chọn: " +
34                     choice.getItem(choice.getSelectedIndex());
35                 label.setText(data);
36             }
37         });
38     }
39
40     public static void main(String args[]) {
41         new ChoiceExample2();
42     }
43 }
```

Kết quả:



List trong Java AWT

Đối tượng của lớp List thể hiện một danh sách các mục văn bản. Với sự giúp đỡ của đối tượng List, người dùng có thể chọn một trong hai mục hoặc nhiều mục. Nó kế thừa lớp Component.

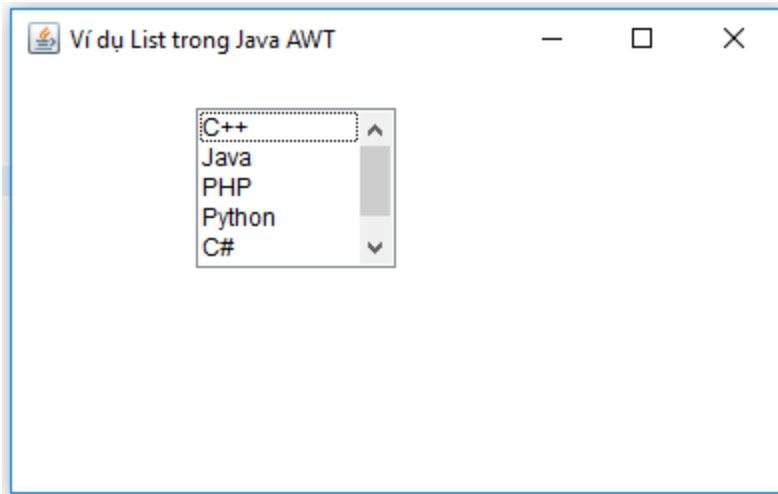
Khai báo lớp AWT List

```
1 public class List extends Component implements ItemSelectable, Accessible
```

Ví dụ List trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.List;
5
6 public class ListExample1 {
7     public ListExample1() {
8         Frame frame = new Frame();
9         frame.setTitle("Ví dụ List trong Java AWT");
10        List list = new List(5);
11        list.setBounds(100, 50, 100, 80);
12        list.add("C++");
13        list.add("Java");
14        list.add("PHP");
15        list.add("Python");
16        list.add("C#");
17        list.add("Adroid");
18        frame.add(list);
19        frame.setSize(400, 250);
20        frame.setLayout(null);
21        frame.setVisible(true);
22    }
23
24    public static void main(String args[]) {
25        new ListExample1();
26    }
27 }
```

Kết quả:



Ví dụ List trong Java AWT với ActionListener

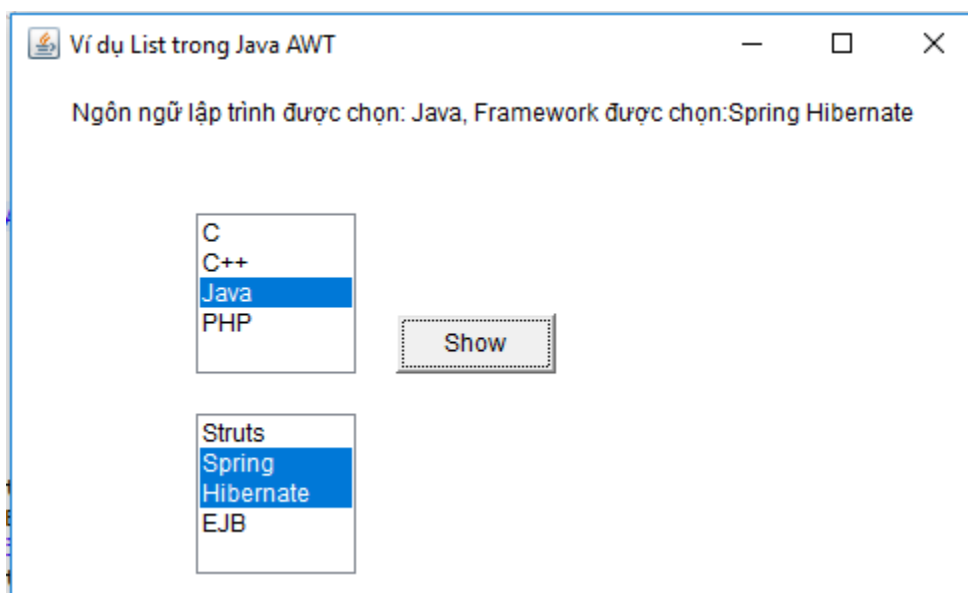
```
1    package vn.plpsoft.awt;
2
3    import java.awt.Button;
4    import java.awt.Frame;
5    import java.awt.Label;
6    import java.awt.List;
7    import java.awt.event.ActionEvent;
8    import java.awt.event.ActionListener;
9
10   public class ListExample2 {
11       ListExample2() {
12           Frame frame = new Frame();
13           final Label label = new Label();
14           label.setAlignment(Label.CENTER);
15           label.setSize(500, 100);
16           Button button = new Button("Show");
17           button.setBounds(200, 150, 80, 30);
18           // tham so thu hai cua ham khai tao List = false, không cho phép chọn nhiều
19           final List list1 = new List(4, false);
20           list1.setBounds(100, 100, 80, 80);
21           list1.add("C");
22           list1.add("C++");
23           list1.add("Java");
24           list1.add("PHP");
25           // tham so thu hai cua ham khai tao List = true, cho phép chọn nhiều
26           final List list2 = new List(4, true);
27           list2.setBounds(100, 200, 80, 80);
28           list2.add("Struts");
29           list2.add("Spring");
30           list2.add("Hibernate");
```

```

31     list2.add("EJB");
32     // add các thành phần vào frame
33     frame.setTitle("Ví dụ List trong Java AWT");
34     frame.add(list1);
35     frame.add(list2);
36     frame.add(label);
37     frame.add(button);
38     frame.setSize(500, 300);
39     frame.setLayout(null);
40     frame.setVisible(true);
41     // đăng ký Listener
42     button.addActionListener(new ActionListener() {
43         public void actionPerformed(ActionEvent e) {
44             String data = "Ngôn ngữ lập trình được chọn: " +
45                 list1.getItem(list1.getSelectedIndex());
46             data += ", Framework được chọn:";
47             for (String frame : list2.getSelectedItems()) {
48                 data += frame + " ";
49             }
50             label.setText(data);
51         }
52     });
53 }
54
55 public static void main(String args[]) {
56     new ListExample2();
57 }
58 }

```

Kết quả:



Canvas trong Java AWT

Canvas đại diện cho một vùng hình chữ nhật trống, nơi ứng dụng có thể vẽ hoặc bày các sự kiện đầu vào từ người dùng. Nó kế thừa lớp Component.

Khai báo lớp AWT Canvas

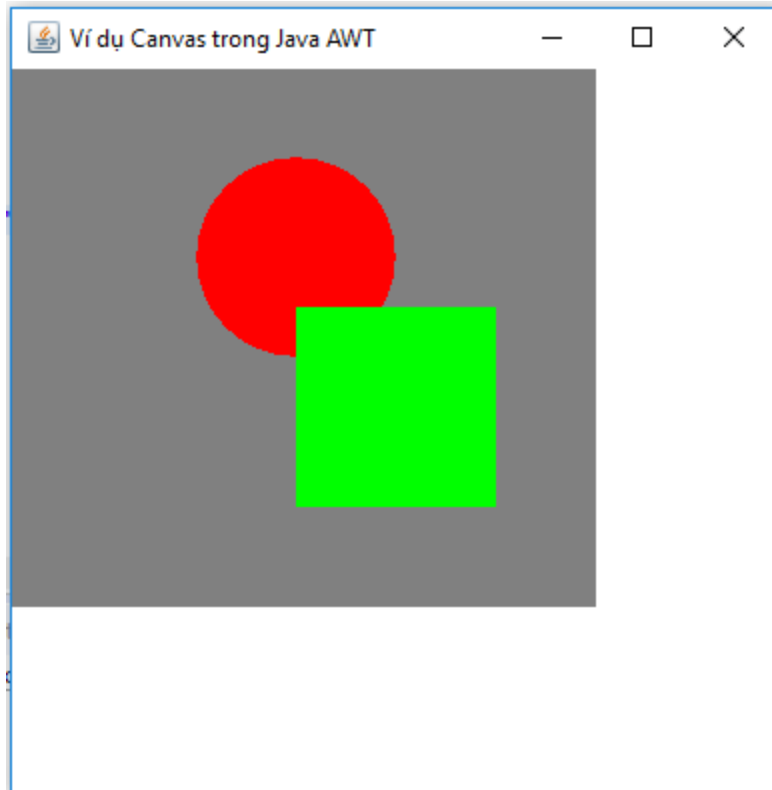
```
1 public class Canvas extends Component implements Accessible
```

Ví dụ Canvas trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Canvas;
4 import java.awt.Color;
5 import java.awt.Frame;
6 import java.awt.Graphics;
7
8 public class CanvasExample {
9     public CanvasExample() {
10         Frame f = new Frame("Ví dụ Canvas trong Java AWT");
11         f.add(new MyCanvas());
12         f.setLayout(null);
13         f.setSize(400, 400);
14         f.setVisible(true);
15     }
16
17     public static void main(String args[]) {
18         new CanvasExample();
19     }
20 }
21
22 class MyCanvas extends Canvas {
23     public MyCanvas() {
24         setBackground(Color.GRAY);
25         setSize(300, 300);
26     }
27
28     @Override
29     public void paint(Graphics g) {
30         // vẽ hình tròn
31         g.setColor(Color.red);
32         g.fillOval(100, 75, 100, 100);
33         // vẽ hình vuông
34         g.setColor(Color.green);
```

```
35         g.fillRect(150, 150, 100, 100);  
36     }  
37 }
```

Kết quả:



Scrollbar trong Java AWT

Đối tượng của lớp Scrollbar được sử dụng để thêm thanh cuộn ngang và dọc. Scrollbar là một thành phần GUI cho phép chúng ta thấy số hàng và cột vô hình.

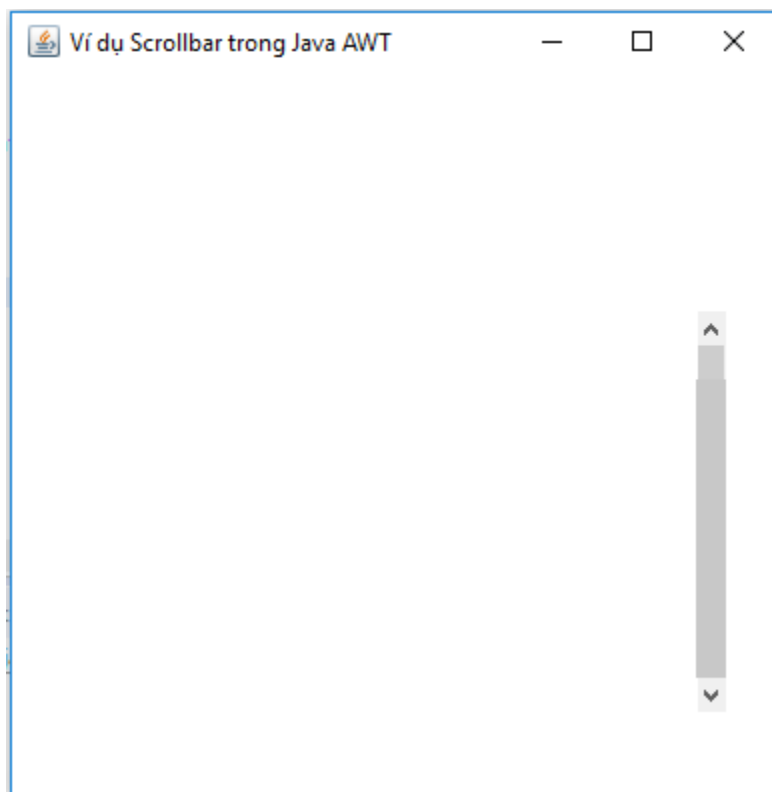
Khai báo lớp AWT Scrollbar

```
1     package vn.plpsoft.awt;  
2  
3     import java.awt.Frame;  
4     import java.awt.Scrollbar;  
5  
6     public class ScrollbarExample1 {  
7         public ScrollbarExample1() {  
8             Frame f = new Frame("Ví dụ Scrollbar trong Java AWT");  
9             Scrollbar s = new Scrollbar();
```

```
10      s.setBounds(350, 150, 15, 200);
11      f.add(s);
12      f.setSize(400, 400);
13      f.setLayout(null);
14      f.setVisible(true);
15  }
16
17  public static void main(String args[]) {
18      new ScrollbarExample1();
19  }
20  }
```

Ví dụ Scrollbar trong Java AWT

Kết quả:

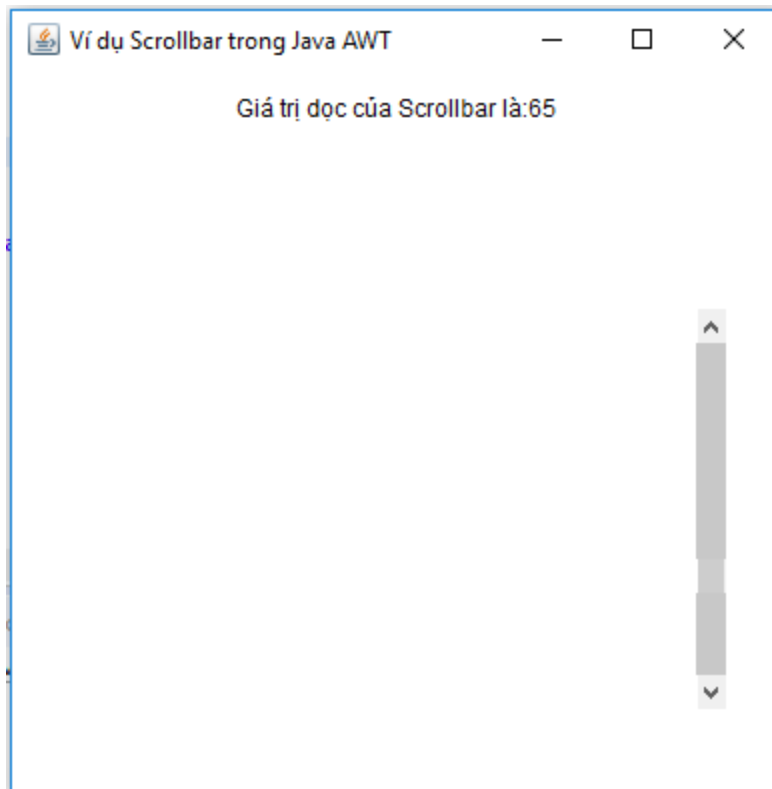


Ví dụ Scrollbar trong Java AWT với AdjustmentListener

```
1  package vn.plpsoft.awt;
2
3  import java.awt.Frame;
4  import java.awt.Label;
```

```
5    import java.awt.Scrollbar;
6    import java.awt.event AdjustmentEvent;
7    import java.awt.event AdjustmentListener;
8
9    public class ScrollbarExample2 {
10        public ScrollbarExample2() {
11            Frame f = new Frame("Ví dụ Scrollbar trong Java AWT");
12            final Label label = new Label();
13            label.setAlignment(Label.CENTER);
14            label.setSize(400, 100);
15            final Scrollbar s = new Scrollbar();
16            s.setBounds(350, 150, 15, 200);
17            f.add(s);
18            f.add(label);
19            f.setSize(400, 400);
20            f.setLayout(null);
21            f.setVisible(true);
22            s.addAdjustmentListener(new AdjustmentListener() {
23                @Override
24                public void adjustmentValueChanged(AdjustmentEvent e) {
25                    label.setText("Giá trị dọc của Scrollbar là:" + s.getValue());
26                }
27            });
28        }
29
30        public static void main(String args[]) {
31            new ScrollbarExample2();
32        }
33    }
```

Kết quả:



MenuItem và Menu trong Java AWT

Đối tượng của lớp MenuItem thêm một mục menu có nhãn vào menu. Các mục được sử dụng trong menu phải thuộc về MenuItem hoặc bất kỳ lớp con nào của nó.

Đối tượng của lớp Menu là một thành phần menu thả xuống được hiển thị trên thanh trình đơn. Nó kế thừa lớp MenuItem.

Khai báo lớp AWT MenuItem

```
1 public class MenuItem extends MenuComponent implements Accessible
```

Khai báo lớp AWT Menu

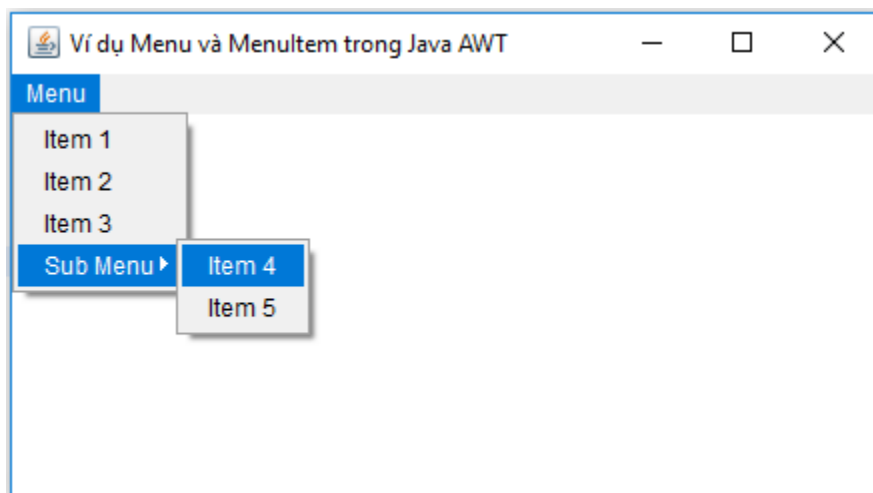
```
1 public class Menu extends MenuItem implements MenuContainer, Accessible
```

Ví dụ MenuItem và Menu trong Java AWT

```
1 package vn.plpsoft.awt;
```

```
2
3 import java.awt.Frame;
4 import java.awt.Menu;
5 import java.awt.MenuBar;
6 import java.awt.MenuItem;
7
8 public class MenuItemExample {
9     public MenuItemExample() {
10         Frame frame = new Frame("Ví dụ Menu và MenuItem trong Java AWT");
11         MenuBar menuBar = new MenuBar();
12         Menu menu = new Menu("Menu");
13         Menu submenu = new Menu("Sub Menu");
14         MenuItem menuItem1 = new MenuItem("Item 1");
15         MenuItem menuItem2 = new MenuItem("Item 2");
16         MenuItem menuItem3 = new MenuItem("Item 3");
17         MenuItem menuItem4 = new MenuItem("Item 4");
18         MenuItem menuItem5 = new MenuItem("Item 5");
19         menu.add(menuItem1);
20         menu.add(menuItem2);
21         menu.add(menuItem3);
22         submenu.add(menuItem4);
23         submenu.add(menuItem5);
24         menu.add(submenu);
25         menuBar.add(menu);
26         frame.setMenuBar(menuBar);
27         frame.setSize(450, 250);
28         frame.setLayout(null);
29         frame.setVisible(true);
30     }
31
32     public static void main(String args[]) {
33         new MenuItemExample();
34     }
35 }
```

Kết quả:



PopupMenu trong Java AWT

PopupMenu có thể được tự động xuất hiện ở vị trí cụ thể trong một thành phần. Nó kế thừa [lớp Menu](#).

Khai báo lớp PopupMenu

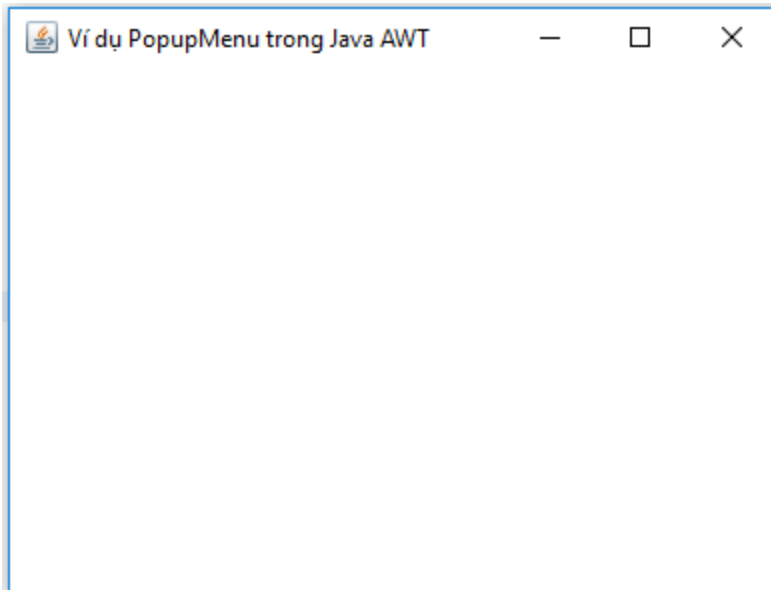
```
1 public class PopupMenu extends Menu implements MenuContainer, Accessible
```

Ví dụ PopupMenu trong Java AWT

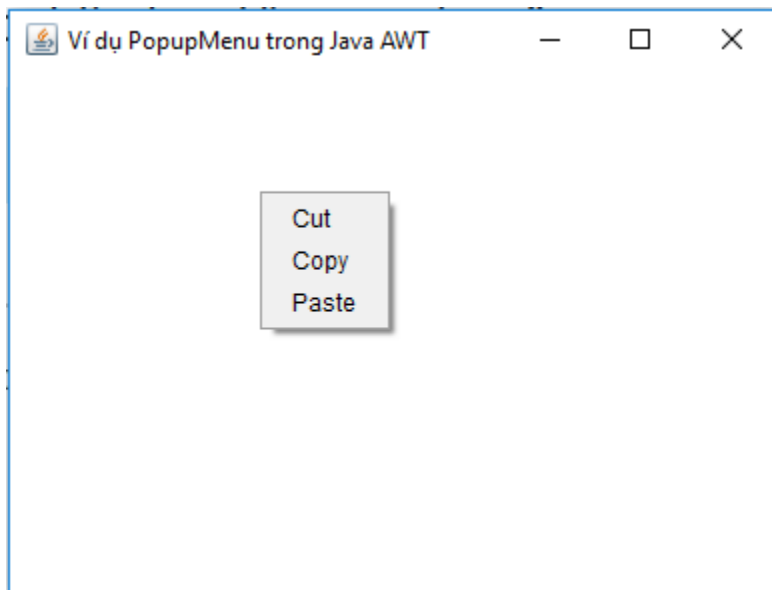
```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.MenuItem;
5 import java.awt.PopupMenu;
6 import java.awt.event.MouseAdapter;
7 import java.awt.event.MouseEvent;
8
9 public class PopupMenuExample {
10
11     public PopupMenuExample() {
12         final Frame frame = new Frame("Ví dụ PopupMenu trong Java AWT");
13         final PopupMenu popupmenu = new PopupMenu("Edit");
14         MenuItem cut = new MenuItem("Cut");
15         cut.setActionCommand("Cut");
16         MenuItem copy = new MenuItem("Copy");
17         copy.setActionCommand("Copy");
18         MenuItem paste = new MenuItem("Paste");
19         paste.setActionCommand("Paste");
20         popupmenu.add(cut);
21         popupmenu.add(copy);
22         popupmenu.add(paste);
23         frame.addMouseListener(new MouseAdapter() {
24             public void mouseClicked(MouseEvent e) {
25                 popupmenu.show(frame, e.getX(), e.getY());
26             }
27         });
28         frame.add(popupmenu);
29         frame.setSize(400, 300);
30         frame.setLayout(null);
31         frame.setVisible(true);
32     }
}
```

```
33  
34     public static void main(String args[]) {  
35         new PopupMenuExample();  
36     }  
37 }
```

Kết quả:



Click chuột phải vào cửa sổ của chương trình:



Panel trong Java AWT

Panel là một lớp container đơn giản nhất. Nó cung cấp không gian trong đó một ứng dụng có thể đính kèm bất kỳ thành phần nào khác. Nó kế thừa lớp Container.

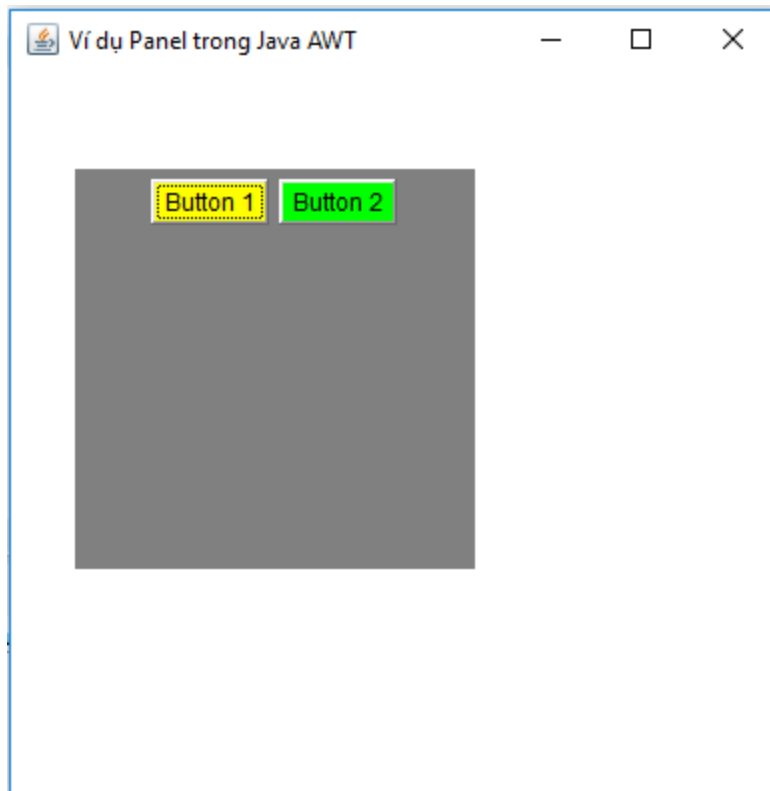
Khai báo lớp AWT Panel

```
1 public class Panel extends Container implements Accessible
```

Ví dụ Panel trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Button;
4 import java.awt.Color;
5 import java.awt.Frame;
6 import java.awt.Panel;
7
8 public class PanelExample {
9     PanelExample() {
10         Frame frame = new Frame("Ví dụ Panel trong Java AWT");
11         Panel panel = new Panel();
12         panel.setBounds(40, 80, 200, 200);
13         panel.setBackground(Color.gray);
14         Button button1 = new Button("Button 1");
15         button1.setBounds(50, 100, 80, 30);
16         button1.setBackground(Color.yellow);
17         Button button2 = new Button("Button 2");
18         button2.setBounds(100, 100, 80, 30);
19         button2.setBackground(Color.green);
20         panel.add(button1);
21         panel.add(button2);
22         frame.add(panel);
23         frame.setSize(400, 400);
24         frame.setLayout(null);
25         frame.setVisible(true);
26     }
27
28     public static void main(String args[]) {
29         new PanelExample();
30     }
31 }
```

Kết quả:



Dialog trong Java AWT

Dialog (hộp thoại) đại diện cho một cửa sổ cấp cao nhất với một đường viền và một tiêu đề được sử dụng để tạo hộp thoại lấy thông tin từ người dùng. Nó kế thừa lớp Window.

Khai báo lớp AWT Dialog

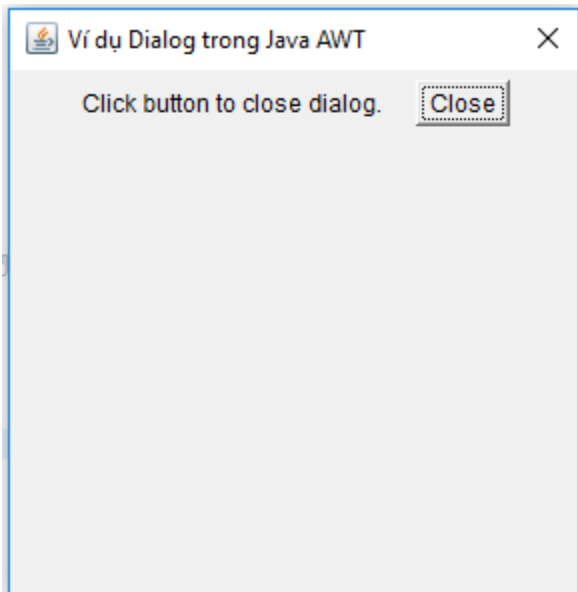
```
1 public class Dialog extends Window
```

Ví dụ Dialog trong Java AWT

```
1 package vn.plpsoft.awt;  
2  
3 import java.awt.Button;  
4 import java.awt.Dialog;  
5 import java.awt.FlowLayout;  
6 import java.awt.Frame;  
7 import java.awt.Label;  
8 import java.awt.event.ActionEvent;  
9 import java.awt.event.ActionListener;
```

```
10
11 public class DialogExample {
12     private static Dialog dialog;
13
14     public DialogExample() {
15         Frame frame = new Frame();
16         dialog = new Dialog(frame, "Ví dụ Dialog trong Java AWT", true);
17         dialog.setLayout(new FlowLayout());
18         Button button = new Button("Close");
19         button.addActionListener(new ActionListener() {
20             public void actionPerformed(ActionEvent e) {
21                 DialogExample.dialog.setVisible(false);
22             }
23         });
24         dialog.add(new Label("Click button to close dialog."));
25         dialog.add(button);
26         dialog.setSize(300, 300);
27         dialog.setVisible(true);
28     }
29
30     public static void main(String args[]) {
31         new DialogExample();
32     }
33 }
```

Kết quả:



Toolkit trong Java AWT

Toolkit là lớp cha trừu tượng của tất cả cài đặt trong Abstract Window Toolkit. Các lớp con của Toolkit được sử dụng để liên kết các thành phần khác nhau. Nó kế thừa lớp Object.

Khai báo lớp AWT Toolkit

```
1 public abstract class Toolkit extends Object
```

Ví dụ Toolkit trong Java AWT - lấy độ phân giải màn hình

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Dimension;
4 import java.awt.Toolkit;
5
6 public class ToolkitExample1 {
7     public static void main(String[] args) {
8         Toolkit t = Toolkit.getDefaultToolkit();
9         System.out.println("Độ phân giải màn hình = "
10             + t.getScreenResolution());
11         Dimension d = t.getScreenSize();
12         System.out.println("Width = " + d.width);
13         System.out.println("Height = " + d.height);
14     }
15 }
```

Kết quả:

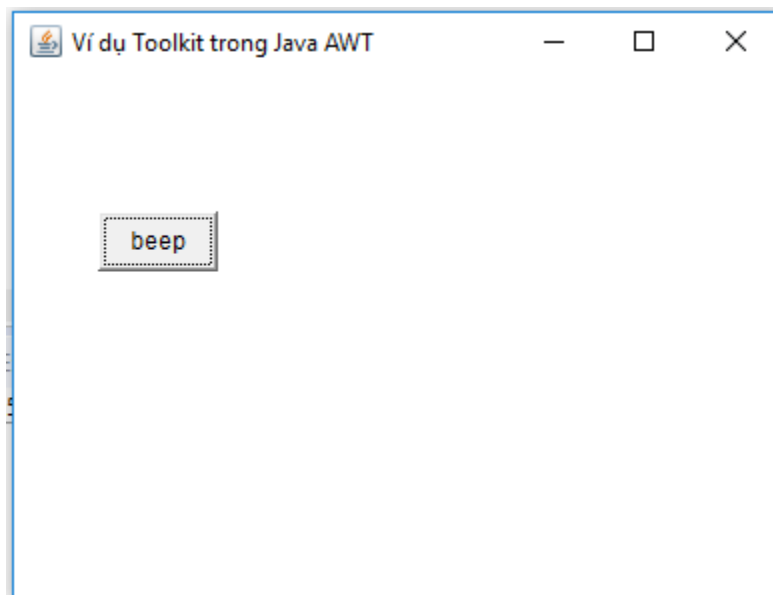
```
Độ phân giải màn hình = 96
Width = 1366
Height = 768
```

Ví dụ Toolkit trong Java AWT - gọi tiếng beep

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Button;
4 import java.awt.Frame;
5 import java.awt.Toolkit;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
```

```
8
9  public class ToolkitExample2 {
10     public static void main(String[] args) {
11         Frame frame = new Frame("Ví dụ Toolkit trong Java AWT");
12         Button button = new Button("beep");
13         button.setBounds(50, 100, 60, 30);
14         frame.add(button);
15         frame.setSize(400, 300);
16         frame.setLayout(null);
17         frame.setVisible(true);
18         button.addActionListener(new ActionListener() {
19             public void actionPerformed(ActionEvent e) {
20                 Toolkit.getDefaultToolkit().beep();
21             }
22         });
23     }
24 }
```

Kết quả:

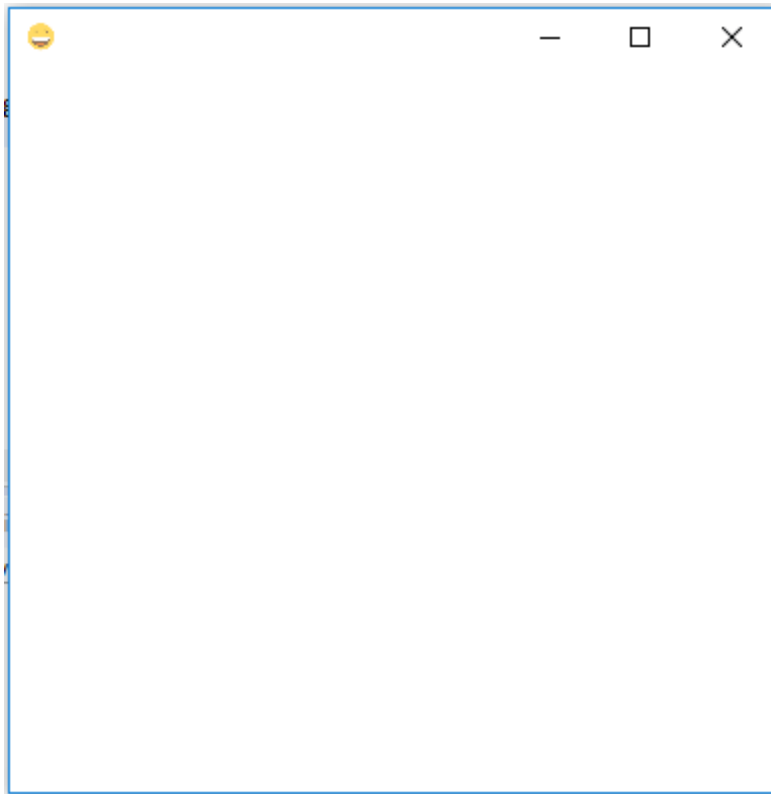


Ví dụ Toolkit trong Java AWT - thay đổi biểu tượng icon trên thanh tiêu đề

```
1  package vn.plpsoft.awt;
2
3  import java.awt.Frame;
4  import java.awt.Image;
```

```
5    import java.awt.Toolkit;
6
7    public class ToolkitExample3 {
8        public ToolkitExample3() {
9            Frame frame = new Frame();
10           Image icon = Toolkit.getDefaultToolkit().getImage("D:\\icon.png");
11           frame.setIconImage(icon);
12           frame.setLayout(null);
13           frame.setSize(400, 400);
14           frame.setVisible(true);
15       }
16
17       public static void main(String args[]) {
18           new ToolkitExample3();
19       }
20   }
```

Kết quả:



ActionListener trong Java AWT

ActionListener được gọi bất cứ khi nào bạn click button hoặc menu item. ActionListener thuộc về [package](#) java.awt.event, nó chỉ có một phương thức actionPerformed().

Phương thức actionPerformed()

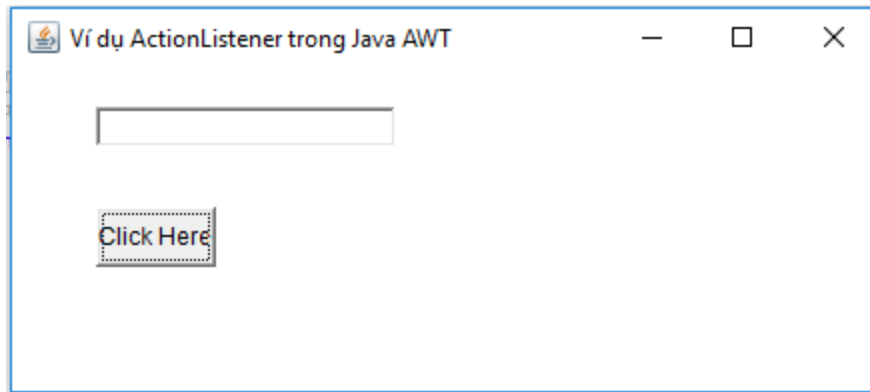
Phương thức actionPerformed() được gọi tự động bất cứ khi nào bạn nhấp vào thành phần đã đăng ký.

```
1 public abstract void actionPerformed(ActionEvent e);
```

Ví dụ về Java ActionListener: click button

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Button;
4 import java.awt.Frame;
5 import java.awt.TextField;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8
9 public class ActionListenerExample {
10     public static void main(String[] args) {
11         Frame frame = new Frame("Ví dụ ActionListener trong Java AWT");
12         final TextField textField = new TextField();
13         textField.setBounds(50, 50, 150, 20);
14         Button button = new Button("Click Here");
15         button.setBounds(50, 100, 60, 30);
16
17         button.addActionListener(new ActionListener() {
18             public void actionPerformed(ActionEvent e) {
19                 textField.setText("Welcome to Plpsoft.Vn");
20             }
21         });
22         frame.add(button);
23         frame.add(textField);
24         frame.setSize(450, 200);
25         frame.setLayout(null);
26         frame.setVisible(true);
27     }
28 }
```

Kết quả:



MouseListener trong Java AWT

Java MouseListener được gọi bất cứ khi nào bạn thay đổi trạng thái của chuột. Interface MouseListener được thuộc về package `java.awt.event`. Nó có năm phương thức.

Các phương thức của interface MouseListener

Interface MouseListener có 5 phương thức được đưa ra như dưới đây.

```
1 public abstract void mouseClicked(MouseEvent e);
2 public abstract void mouseEntered(MouseEvent e);
3 public abstract void mouseExited(MouseEvent e);
4 public abstract void mousePressed(MouseEvent e);
5 public abstract void mouseReleased(MouseEvent e);
```

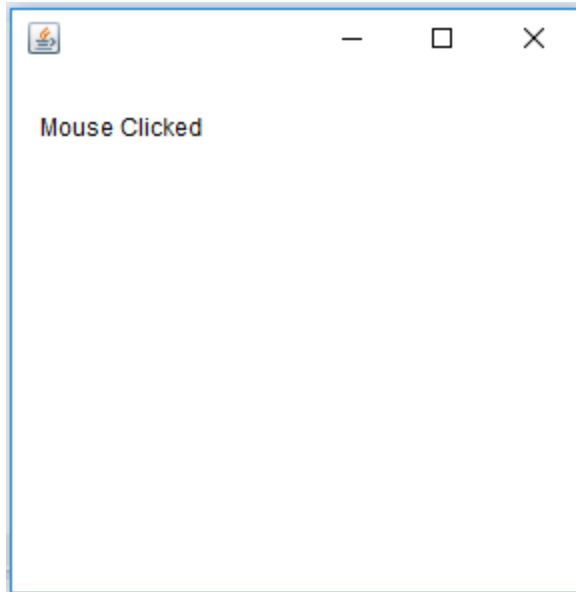
Ví dụ về Java MouseListener trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.Label;
5 import java.awt.event.MouseEvent;
6 import java.awt.event.MouseListener;
7
8 public class MouseListenerExample1 extends Frame implements MouseListener {
9     private Label label;
10
11     public MouseListenerExample1() {
12         addMouseListener(this);
```



```
13
14     label = new Label();
15     label.setBounds(20, 50, 100, 20);
16     add(label);
17     setSize(300, 300);
18     setLayout(null);
19     setVisible(true);
20 }
21
22 public void mouseClicked(MouseEvent e) {
23     label.setText("Mouse Clicked");
24 }
25
26 public void mouseEntered(MouseEvent e) {
27     label.setText("Mouse Entered");
28 }
29
30 public void mouseExited(MouseEvent e) {
31     label.setText("Mouse Exited");
32 }
33
34 public void mousePressed(MouseEvent e) {
35     label.setText("Mouse Pressed");
36 }
37
38 public void mouseReleased(MouseEvent e) {
39     label.setText("Mouse Released");
40 }
41
42 public static void main(String[] args) {
43     new MouseListenerExample1();
44 }
45 }
```

Kết quả:

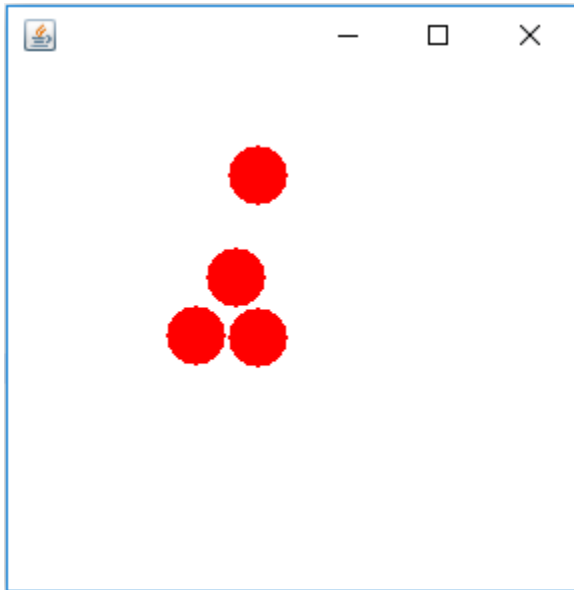


Ví dụ 2 về Java MouseListener trong Java AWT

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Color;
4    import java.awt.Frame;
5    import java.awt.Graphics;
6    import java.awt.event.MouseEvent;
7    import java.awt.event.MouseListener;
8
9    public class MouseListenerExample2 extends Frame implements MouseListener {
10        public MouseListenerExample2() {
11            addMouseListener(this);
12
13            setSize(300, 300);
14            setLayout(null);
15            setVisible(true);
16        }
17
18        public void mouseClicked(MouseEvent e) {
19            Graphics g = getGraphics();
20            g.setColor(Color.RED);
21            g.fillOval(e.getX(), e.getY(), 30, 30);
22        }
23
24        public void mouseEntered(MouseEvent e) {
25        }
```

```
26
27     public void mouseExited(MouseEvent e) {
28     }
29
30     public void mousePressed(MouseEvent e) {
31     }
32
33     public void mouseReleased(MouseEvent e) {
34     }
35
36     public static void main(String[] args) {
37         new MouseListenerExample2();
38     }
39 }
```

Kết quả:



MouseMotionListener trong Java AWT

Java MouseMotionListener gọi bất cứ khi nào bạn di chuyển hoặc kéo chuột. Interface MouseMotionListener thuộc về gói java.awt.event. Nó có hai phương thức.

Các phương thức của interface MouseMotionListener

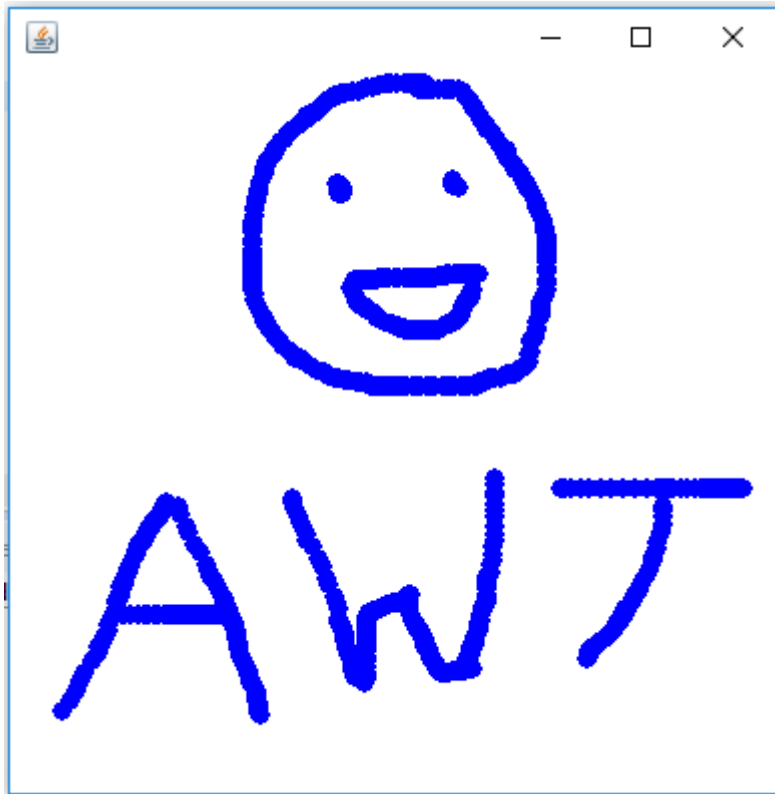
Interface MouseMotionListener có 2 phương thức được đưa ra như dưới đây.

```
1     public abstract void mouseDragged(MouseEvent e);
2     public abstract void mouseMoved(MouseEvent e);
```

Ví dụ về Java MouseMotionListener trong Java AWT

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Color;
4    import java.awt.Frame;
5    import java.awt.Graphics;
6    import java.awt.event.MouseEvent;
7    import java.awt.event.MouseMotionListener;
8
9    public class MouseMotionListenerExample1 extends Frame
10        implements MouseMotionListener {
11        public MouseMotionListenerExample1() {
12            addMouseMotionListener(this);
13
14            setSize(400, 400);
15            setLayout(null);
16            setVisible(true);
17        }
18
19        public void mouseDragged(MouseEvent e) {
20            Graphics g = getGraphics();
21            g.setColor(Color.BLUE);
22            g.fillOval(e.getX(), e.getY(), 10, 10);
23        }
24
25        public void mouseMoved(MouseEvent e) {
26        }
27
28        public static void main(String[] args) {
29            new MouseMotionListenerExample1();
30        }
31    }
```

Kết quả:



Ví dụ 2 về Java MouseMotionListener trong Java AWT

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Color;
4    import java.awt.Frame;
5    import java.awt.Graphics;
6    import java.awt.Label;
7    import java.awt.event.MouseEvent;
8    import java.awt.event.MouseMotionListener;
9
10   public class MouseMotionListenerExample2 extends Frame
11       implements MouseMotionListener {
12       Label label;
13       Color color = Color.BLUE;
14
15       public MouseMotionListenerExample2() {
16           label = new Label();
17           label.setBounds(20, 40, 100, 20);
18           add(label);
19
20           addMouseMotionListener(this);
21
```

```
22     setSize(400, 400);
23     setLayout(null);
24     setVisible(true);
25 }
26
27 public void mouseDragged(MouseEvent e) {
28     label.setText("X=" + e.getX() + ", Y=" + e.getY());
29     Graphics g = getGraphics();
30     g.setColor(Color.RED);
31     g.fillOval(e.getX(), e.getY(), 10, 10);
32 }
33
34 public void mouseMoved(MouseEvent e) {
35     label.setText("X=" + e.getX() + ", Y=" + e.getY());
36 }
37
38 public static void main(String[] args) {
39     new MouseMotionListenerExample2();
40 }
41 }
```

Kết quả:



ItemListener trong Java AWT

Java `ItemListener` được gọi bất cứ khi nào bạn click vào hộp kiểm (checkbox). Interface `ItemListener` thuộc về package `java.awt.event`. Nó chỉ có một phương thức: `itemStateChanged()`.

Phương thức `itemStateChanged()`

Phương thức `itemStateChanged()` được gọi tự động bất cứ khi nào bạn nhấn hoặc bỏ chọn vào thành phần checkbox đã đăng ký.

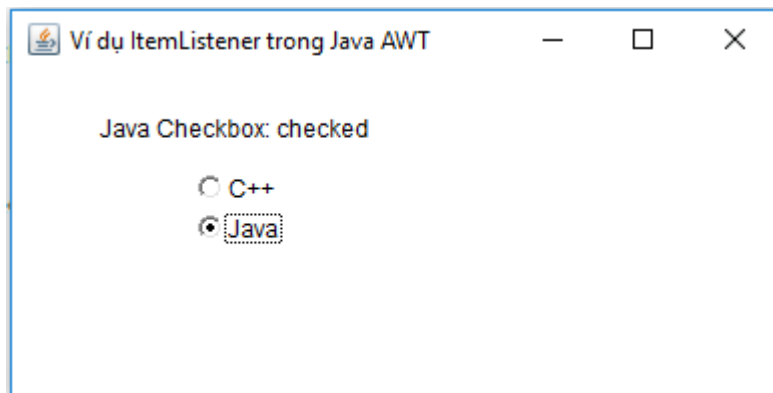
```
1 public abstract void itemStateChanged(ItemEvent e);
```

Ví dụ `ItemListener` trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Checkbox;
4 import java.awt.CheckboxGroup;
5 import java.awt.Frame;
6 import java.awt.Label;
7 import java.awt.event.ItemEvent;
8 import java.awt.event.ItemListener;
9
10 public class ItemListenerExample {
11     public ItemListenerExample() {
12         Frame frame = new Frame("Ví dụ ItemListener trong Java AWT");
13         final Label label1 = new Label();
14         label1.setBounds(50, 50, 200, 20);
15         CheckboxGroup cbg = new CheckboxGroup();
16         Checkbox checkbox1 = new Checkbox("C++", cbg, false);
17         checkbox1.setBounds(100, 80, 50, 20);
18         Checkbox checkbox2 = new Checkbox("Java", cbg, true);
19         checkbox2.setBounds(100, 100, 50, 20);
20         frame.add(checkbox1);
21         frame.add(checkbox2);
22         frame.add(label1);
23         checkbox1.addItemListener(new ItemListener() {
24             public void itemStateChanged(ItemEvent e) {
25                 label1.setText("C++ Checkbox: " + (e.getStateChange() == 1
26                     ? "checked" : "unchecked"));
27             }
28         });
29         checkbox2.addItemListener(new ItemListener() {
30             public void itemStateChanged(ItemEvent e) {
31                 label1.setText("Java Checkbox: " + (e.getStateChange() == 1
32                     ? "checked" : "unchecked"));
33             }
34         });
35     }
36 }
```

```
34     });
35     frame.setSize(400, 200);
36     frame.setLayout(null);
37     frame.setVisible(true);
38 }
39
40 public static void main(String args[]) {
41     new ItemListenerExample();
42 }
43 }
```

Kết quả:



KeyListener trong Java AWT

Java KeyListener được gọi bất cứ khi nào bạn thay đổi trạng thái của bàn phím (key). Interface KeyListener thuộc về gói java.awt.event. Nó có ba phương thức.

Các phương thức của interface KeyListener

Interface KeyListener có 3 phương thức được đưa ra như dưới đây.

```
1 public abstract void keyPressed(KeyEvent e);
2 public abstract void keyReleased(KeyEvent e);
3 public abstract void keyTyped(KeyEvent e);
```

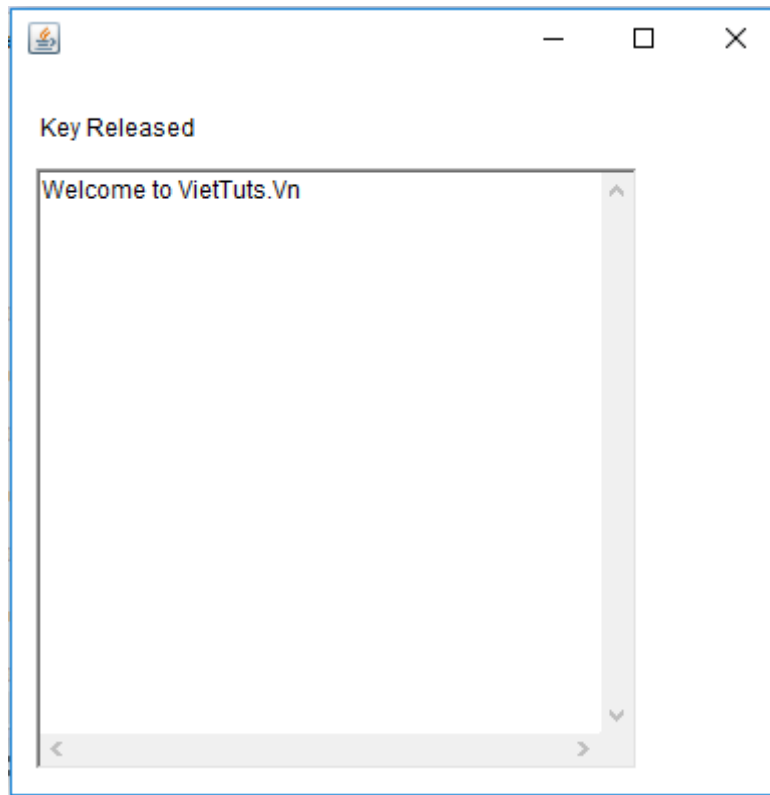
Ví dụ về Java KeyListener trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.Label;
5 import java.awt.TextArea;
6 import java.awt.event.KeyEvent;
7 import java.awt.event.KeyListener;
8
9 public class KeyListenerExample1 extends Frame implements KeyListener {
```



```
10     private Label label;
11     private TextArea textArea;
12
13     public KeyListenerExample1() {
14         label = new Label();
15         label.setBounds(20, 50, 100, 20);
16         textArea = new TextArea();
17         textArea.setBounds(20, 80, 300, 300);
18         textArea.addKeyListener(this);
19
20         add(label);
21         add(textArea);
22         setSize(400, 400);
23         setLayout(null);
24         setVisible(true);
25     }
26
27     public void keyPressed(KeyEvent e) {
28         label.setText("Key Pressed");
29     }
30
31     public void keyReleased(KeyEvent e) {
32         label.setText("Key Released");
33     }
34
35     public void keyTyped(KeyEvent e) {
36         label.setText("Key Typed");
37     }
38
39     public static void main(String[] args) {
40         new KeyListenerExample1();
41     }
42 }
```

Kết quả:

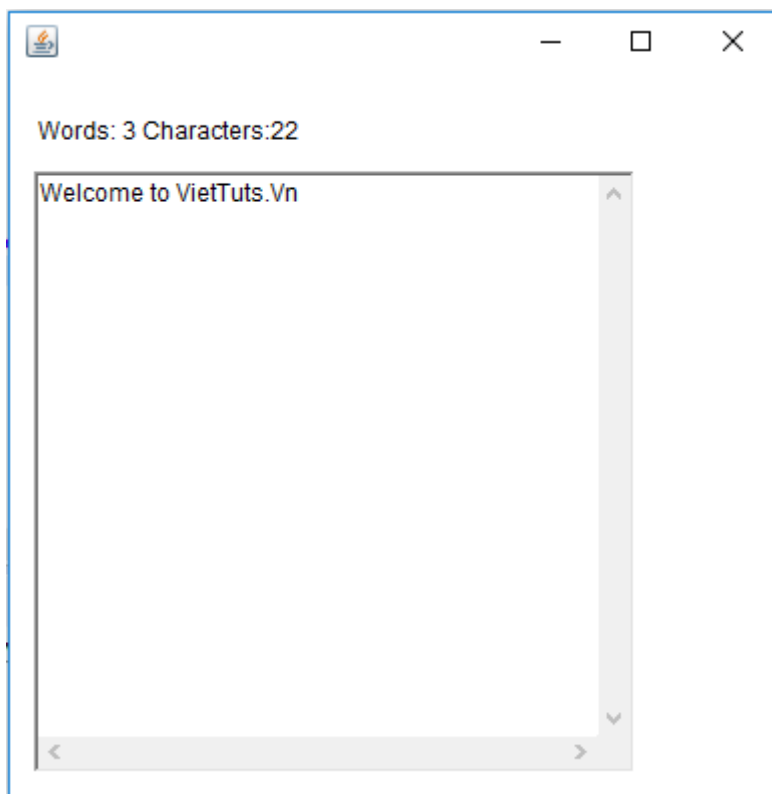


Ví dụ 2 về Java KeyListener trong Java AWT - Đếm số từ và ký tự

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Frame;
4    import java.awt.Label;
5    import java.awt.TextArea;
6    import java.awt.event.KeyEvent;
7    import java.awt.event.KeyListener;
8
9    public class KeyListenerExample2 extends Frame implements KeyListener {
10        private Label label;
11        private TextArea textArea;
12
13        public KeyListenerExample2() {
14            label = new Label();
15            label.setBounds(20, 50, 200, 20);
16            textArea = new TextArea();
17            textArea.setBounds(20, 80, 300, 300);
18            textArea.addKeyListener(this);
19        }
```

```
20     add(label);
21     add(textArea);
22     setSize(400, 400);
23     setLayout(null);
24     setVisible(true);
25 }
26
27 public void keyPressed(KeyEvent e) {
28 }
29
30 public void keyReleased(KeyEvent e) {
31     String text = textArea.getText();
32     String words[] = text.split("\\s");
33     label.setText("Words: " + words.length + " Characters:" + text.length());
34 }
35
36 public void keyTyped(KeyEvent e) {
37 }
38
39 public static void main(String[] args) {
40     new KeyListenerExample2();
41 }
42 }
```

Kết quả:



WindowListener trong Java AWT

Java WindowListener được gọi bất cứ khi nào bạn thay đổi trạng thái của window. Interface WindowListener thuộc về package java.awt.event. Nó có 7 phương thức.

Các phương thức của interface WindowListener

Interface WindowListener có 7 phương thức được đưa ra như dưới đây.

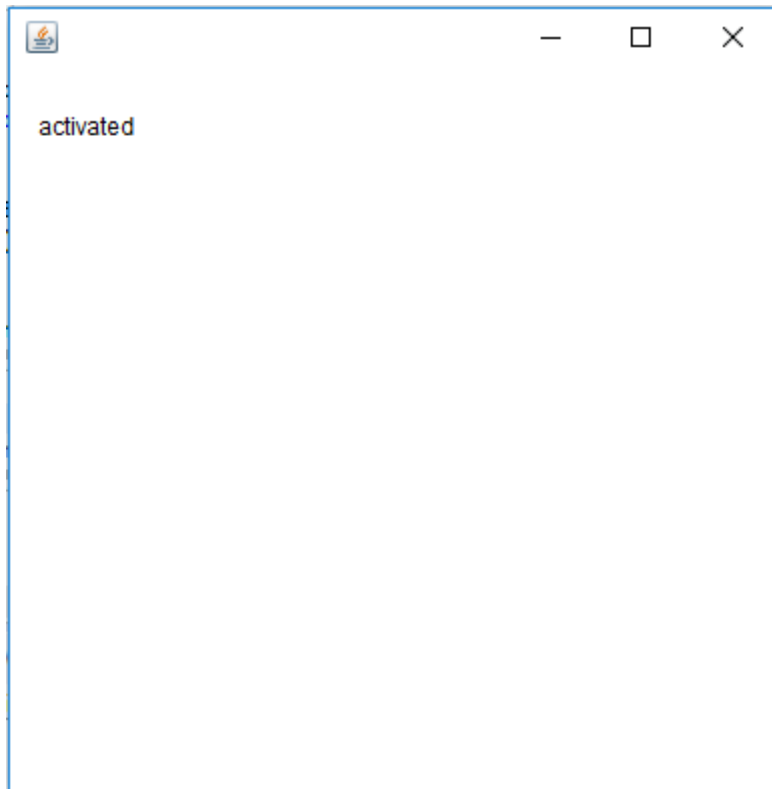
```
1 public abstract void windowActivated(WindowEvent e);
2 public abstract void windowClosed(WindowEvent e);
3 public abstract void windowClosing(WindowEvent e);
4 public abstract void windowDeactivated(WindowEvent e);
5 public abstract void windowDeiconified(WindowEvent e);
6 public abstract void windowIconified(WindowEvent e);
7 public abstract void windowOpened(WindowEvent e);
```

Ví dụ về Java WindowListener trong Java AWT

```
1 package vn.plpsoft.awt;
2
3 import java.awt.Frame;
4 import java.awt.Label;
5 import java.awt.event.WindowEvent;
6 import java.awt.event.WindowListener;
7
8 public class WindowListenerExample1 extends Frame
9     implements WindowListener {
10     private Label label;
11     public WindowListenerExample1() {
12         label = new Label();
13         label.setBounds(20, 50, 200, 20);
14         addWindowListener(this);
15         add(label);
16         setSize(400, 400);
17         setLayout(null);
18         setVisible(true);
19     }
20
21     public void windowActivated(WindowEvent arg0) {
22         label.setText("activated");
23     }
24
25     public void windowClosed(WindowEvent arg0) {
26         System.out.println("closed");
27     }
28
29     public void windowClosing(WindowEvent arg0) {
30         System.out.println("closing");
31         dispose();
32     }
```

```
33
34     public void windowDeactivated(WindowEvent arg0) {
35         label.setText("deactivated");
36     }
37
38     public void windowDeiconified(WindowEvent arg0) {
39         System.out.println("deiconified");
40     }
41
42     public void windowIconified(WindowEvent arg0) {
43         System.out.println("iconified");
44     }
45
46     public void windowOpened(WindowEvent arg0) {
47         System.out.println("opened");
48     }
49
50     public static void main(String[] args) {
51         new WindowListenerExample1();
52     }
53 }
```

Kết quả:



Close Window trong Java AWT

Chúng ta có thể **đóng cửa sổ AWT hoặc Frame** bằng cách gọi phương thức **dispose()** hoặc **System.exit()** bên trong phương thức **windowClosing()**. Phương thức **windowClosing()** được định nghĩa trong interface **WindowListener** và lớp **WindowAdapter**.

Lớp **WindowAdapter** cung cấp bản cài đặt chi tiết cho interface **WindowListener**. Nó cung cấp việc thực hiện mặc định tất cả 7 phương thức của interface **WindowListener**. Để ghi đè phương thức **windowClosing()**, bạn có thể sử dụng lớp **WindowAdapter** hoặc interface **WindowListener**.

Nếu bạn implements interface **WindowListener**, bạn sẽ bị buộc phải ghi đè tất cả 7 phương thức của interface **WindowListener**. Vì vậy, tốt hơn là sử dụng lớp **WindowAdapter**.

Các cách khác nhau để ghi đè phương thức **windowClosing()**

- Bằng cách sử dụng lớp nặc danh.
- Bằng cách kế thừa lớp **WindowAdapter**.
- Bằng cách triển khai interface **WindowListener**.

Ví dụ 1 về close window trong Java AWT - sử dụng lớp nặc danh

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Frame;
4    import java.awt.event.WindowAdapter;
5    import java.awt.event.WindowEvent;
6
7    public class CloseWindowExample1 extends Frame {
8        public CloseWindowExample1() {
9            addWindowListener(new WindowAdapter() {
10                public void windowClosing(WindowEvent e) {
11                    dispose();
12                }
13            });
14            setSize(400, 400);
15            setLayout(null);
16            setVisible(true);
17        }
18
19        /**
```

```
20     * main
21     *
22     * @param args
23     */
24     public static void main(String[] args) {
25         new CloseWindowExample1();
26     }
27 }
```

Ví dụ 2 về close window trong Java AWT - kế thừa lớp WindowAdapter

```
1     package vn.plpsoft.awt;
2
3     import java.awt.Frame;
4     import java.awt.event.WindowAdapter;
5     import java.awt.event.WindowEvent;
6
7     public class CloseWindowExample2 extends WindowAdapter {
8         private Frame frame;
9
10        public CloseWindowExample2() {
11            frame = new Frame();
12            frame.addWindowListener(this);
13
14            frame.setSize(400, 400);
15            frame.setLayout(null);
16            frame.setVisible(true);
17        }
18
19        public void windowClosing(WindowEvent e) {
20            frame.dispose();
21        }
22
23        /**
24         * main
25         *
26         * @param args
27         */
28        public static void main(String[] args) {
29            new CloseWindowExample2();
30        }
31    }
```

Ví dụ 3 về close window trong Java AWT - triển khai interface WindowListener

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Frame;
4    import java.awt.event.WindowEvent;
5    import java.awt.event.WindowListener;
6
7    public class CloseWindowExample3 extends Frame implements WindowListener {
8        public CloseWindowExample3() {
9            addWindowListener(this);
10
11            setSize(400, 400);
12            setLayout(null);
13            setVisible(true);
14        }
15
16        public void windowActivated(WindowEvent e) {
17        }
18
19        public void windowClosed(WindowEvent e) {
20        }
21
22        public void windowClosing(WindowEvent e) {
23            dispose();
24        }
25
26        public void windowDeactivated(WindowEvent e) {
27        }
28
29        public void windowDeiconified(WindowEvent e) {
30        }
31
32        public void windowIconified(WindowEvent e) {
33        }
34
35        public void windowOpened(WindowEvent arg0) {
36        }
37
38        /**
39         * main
40         *
41         * @param args
42         */
43        public static void main(String[] args) {
44            new CloseWindowExample3();
45        }
46    }
```


Các lớp Adapter trong Java AWT

Các lớp Adapter trong Java AWT cung cấp bản cài đặt mặc định cho các interface Listener. Nếu bạn kế thừa lớp Adapter, bạn sẽ không bị buộc phải cung cấp cài đặt cho các phương thức của các interface Listener. Vì vậy, nó giúp tiết kiệm code.

Các lớp Adapter được tìm thấy trong các gói `java.awt.event`, `java.awt.dnd` và `javax.swing.event`. Các lớp Adapter với các interface Listener tương ứng của chúng được đưa ra dưới đây.

Các lớp `java.awt.event` Adapter

Lớp Adapter	Listener <u>interface</u>
WindowAdapter	<u>WindowListener</u>
KeyAdapter	<u>KeyListener</u>
MouseAdapter	<u>MouseListener</u>
MouseMotionAdapter	<u>MouseMotionListener</u>
FocusAdapter	FocusListener
ComponentAdapter	ComponentListener
ContainerAdapter	ContainerListener
HierarchyBoundsAdapter	HierarchyBoundsListener

Các lớp `java.awt.dnd` Adapter

Lớp Adapter	Listener interface
DragSourceAdapter	DragSourceListener
DragTargetAdapter	DragTargetListener

Các lớp javax.swing.event Adapter

Lớp Adapter	Listener interface
MouseInputAdapter	MouseListener
InternalFrameAdapter	InternalFrameListener

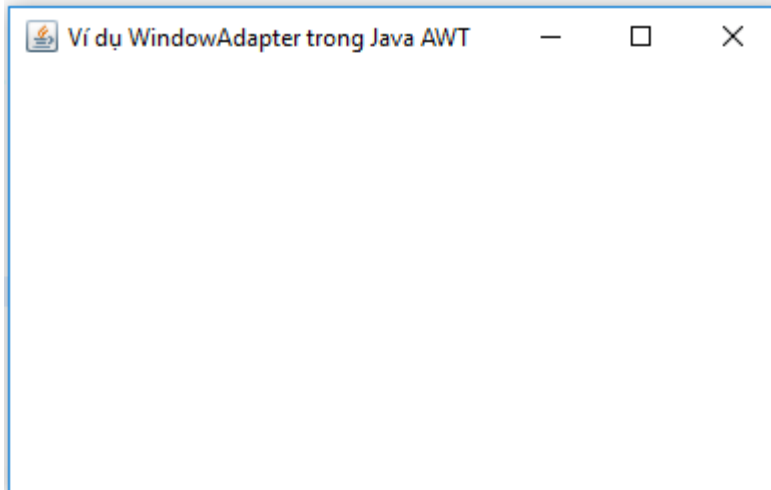
Ví dụ WindowAdapter trong Java AWT

```

1    package vn.plpsoft.awt;
2
3    import java.awt.Frame;
4    import java.awt.event.WindowAdapter;
5    import java.awt.event.WindowEvent;
6
7    public class AdapterExample1 {
8        Frame frame;
9
10       AdapterExample1() {
11           frame = new Frame("Ví dụ WindowAdapter trong Java AWT");
12           frame.addWindowListener(new WindowAdapter() {
13               public void windowClosing(WindowEvent e) {
14                   frame.dispose();
15               }
16           });
17
18           frame.setSize(400, 250);
19           frame.setLayout(null);
20           frame.setVisible(true);
21       }
22
23       public static void main(String[] args) {
24           new AdapterExample1();
25       }
26   }

```

Kết quả:

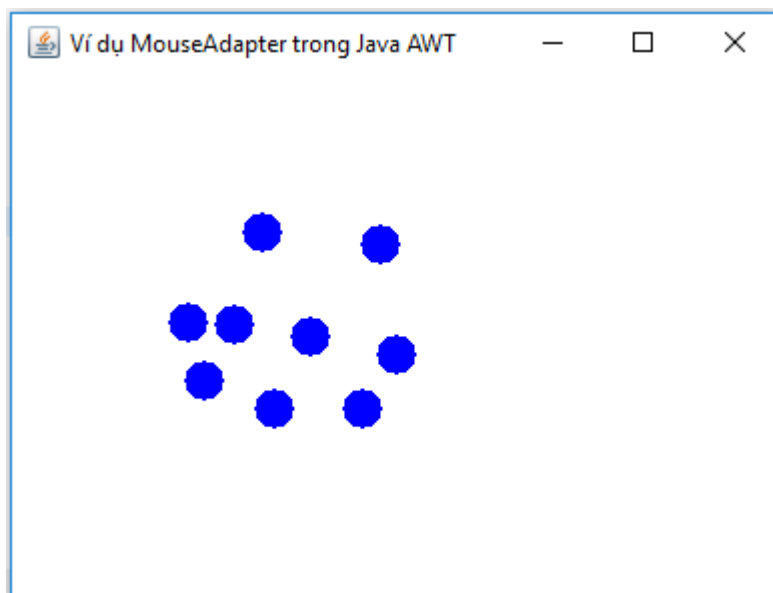


Ví dụ MouseAdapter trong Java AWT

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Color;
4    import java.awt.Frame;
5    import java.awt.Graphics;
6    import java.awt.event.MouseAdapter;
7    import java.awt.event.MouseEvent;
8    import java.awt.event.WindowAdapter;
9    import java.awt.event.WindowEvent;
10
11   public class MouseAdapterExample extends MouseAdapter {
12       private Frame frame;
13
14       public MouseAdapterExample() {
15           frame = new Frame("Ví dụ MouseAdapter trong Java AWT");
16           frame.addMouseListener(this);
17
18           frame.setSize(400, 300);
19           frame.setLayout(null);
20           frame.setVisible(true);
21           // close window
22           frame.addWindowListener(new WindowAdapter() {
23               public void windowClosing(WindowEvent e) {
24                   System.exit(1);
25               }
26           });
27       }
28
29       public void mouseClicked(MouseEvent e) {
30           Graphics g = frame.getGraphics();
```

```
31     g.setColor(Color.BLUE);
32     g.fillOval(e.getX(), e.getY(), 20, 20);
33 }
34
35 /**
36  * main
37  *
38  * @param args
39  */
40 public static void main(String[] args) {
41     new MouseAdapterExample();
42 }
43 }
```

Kết quả:

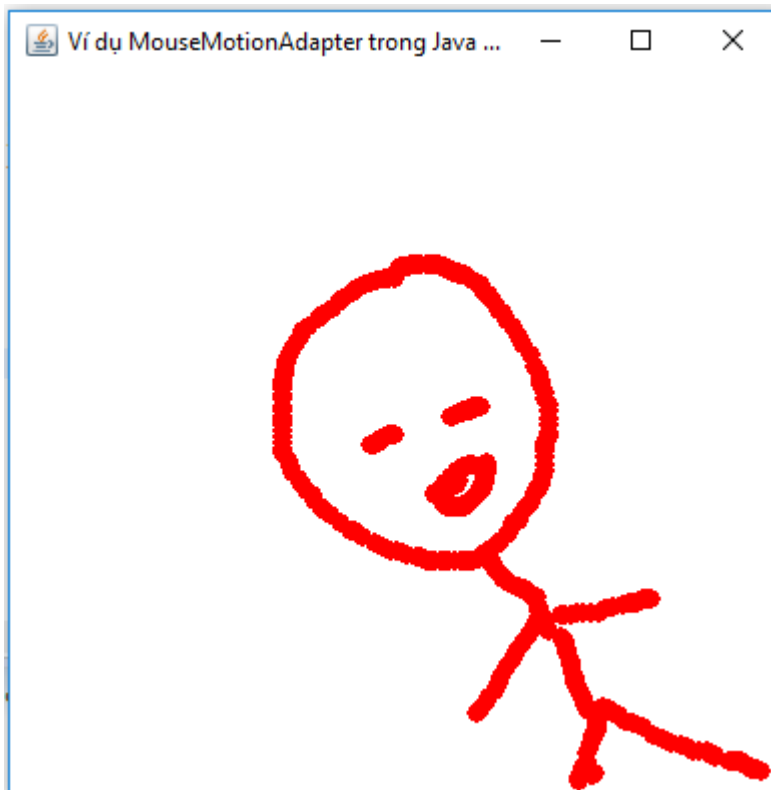


Ví dụ MouseMotionAdapter trong Java AWT

```
1     package vn.plpsoft.awt;
2
3     import java.awt.Color;
4     import java.awt.Frame;
5     import java.awt.Graphics;
6     import java.awt.event.MouseEvent;
7     import java.awt.event.MouseMotionAdapter;
8     import java.awt.event.WindowAdapter;
9     import java.awt.event.WindowEvent;
10
11     public class MouseMotionAdapterExample extends MouseMotionAdapter {
12         private Frame frame;
13
14         public MouseMotionAdapterExample() {
```

```
15     frame = new Frame("Ví dụ MouseMotionAdapter trong Java AWT");
16     frame.addMouseListener(this);
17
18     frame.setSize(400, 400);
19     frame.setLayout(null);
20     frame.setVisible(true);
21
22     // close window
23     frame.addWindowListener(new WindowAdapter() {
24         public void windowClosing(WindowEvent e) {
25             System.exit(1);
26         }
27     });
28 }
29
30 public void mouseDragged(MouseEvent e) {
31     Graphics g = frame.getGraphics();
32     g.setColor(Color.RED);
33     g.fillOval(e.getX(), e.getY(), 10, 10);
34 }
35
36 public static void main(String[] args) {
37     new MouseMotionAdapterExample();
38 }
39 }
```

Kết quả:



Ví dụ KeyAdapter trong Java AWT

```
1    package vn.plpsoft.awt;
2
3    import java.awt.Frame;
4    import java.awt.Label;
5    import java.awt.TextArea;
6    import java.awt.event.KeyAdapter;
7    import java.awt.event.KeyEvent;
8    import java.awt.event.WindowAdapter;
9    import java.awt.event.WindowEvent;
10
11   public class KeyAdapterExample extends KeyAdapter {
12       Label label;
13       TextArea textArea;
14       Frame frame;
15
16       KeyAdapterExample() {
17           frame = new Frame("Ví dụ KeyAdapter trong Java AWT");
18           label = new Label();
19           label.setBounds(20, 50, 200, 20);
20           textArea = new TextArea();
21           textArea.setBounds(20, 80, 300, 300);
22           textArea.addKeyListener(this);
23
24           frame.add(label);
25           frame.add(textArea);
26           frame.setSize(400, 400);
27           frame.setLayout(null);
28           frame.setVisible(true);
29
30           // close window
31           frame.addWindowListener(new WindowAdapter() {
32               public void windowClosing(WindowEvent e) {
33                   System.exit(1);
34               }
35           });
36       }
37
38       public void keyReleased(KeyEvent e) {
39           String text = textArea.getText();
40           String words[] = text.split("\\s");
41           label.setText("Words: " + words.length + " Characters:" + text.length());
42       }
43
44       public static void main(String[] args) {
45           new KeyAdapterExample();
46       }
47   }
```

Kết quả:

