

# Cơ bản về Applet trong Java

Một Applet là một chương trình Java chạy trên một trình duyệt Web. Một applet có thể là một ứng dụng Java đầy đủ tính năng bởi vì nó có toàn bộ Java API tại cách bố trí của nó.

Để hiểu sâu hơn các khái niệm được trình bày trong chương này, mời bạn tham khảo loạt bài: [Ví dụ về Applet trong Java](#).

Có một số điểm khác nhau quan trọng giữa một applet và một ứng dụng Standalone Java, đó là:

- Một applet là một lớp Java mà kế thừa lớp `java.applet.Applet`.
- Một phương thức `main()` không được gọi trên một applet, và một lớp applet sẽ không định nghĩa `main()`.
- Các applet được thiết kế để được nhúng bên trong một trang HTML.
- Khi một người dùng quan sát một trang HTML mà chứa một applet, thì code cho applet này được tải tới thiết bị của người sử dụng.
- Một JVM được yêu cầu để quan sát một applet. JVM có thể là hoặc một plugin của trình duyệt Web hoặc một môi trường runtime riêng rẽ.
- JVM trên thiết bị người dùng tạo một instance của lớp applet và gọi các phương thức khác nhau trong suốt vòng đời của applet.
- Các applet có các qui tắc bảo mật nghiêm ngặt được thi hành bởi trình duyệt Web. Sự bảo mật của một applet thường được xem như là Sandbox Security, so sánh applet như là một đứa trẻ đang chờ trong một sandbox với các qui tắc đã định mà nó phải tuân theo.
- Các lớp khác, mà applet cần, có thể được tải trong một file đơn là Java Archive (JAR).

## Vòng đời của một Applet trong Java

4 phương thức trong lớp Applet cung cấp cho bạn Framework, trên đó bạn xây dựng bất kỳ applet quan trọng nào:

- **init:** Phương thức này được dự định cho khởi tạo bất cứ thứ gì cần thiết cho applet. Nó được gọi sau khi các thẻ `param` bên trong thẻ `applet` đã được xử lý.

- **start:** Phương thức này tự động được gọi sau khi trình duyệt gọi phương thức `init`. Nó cũng được gọi bất cứ khi nào người dùng trả về trang mà chứa applet sau khi đã rời khỏi trang để tới trang khác.
- **stop:** Phương thức này tự động được gọi khi người dùng rời khỏi trang mà trên đó có applet đó. Vì thế, nó có thể được gọi một cách lặp đi lặp lại với cùng applet.
- **destroy:** Phương thức này chỉ được gọi khi trình duyệt ngừng theo cách thông thường. Bởi vì các applet là sống trên một trang HTML, nên khi người dùng rời khỏi trang mà chứa applet thì applet sẽ hủy.
- **paint:** Được gọi ngay lập tức sau khi phương thức `start()`, và cũng được gọi bất kỳ khi nào applet cần repaint chính nó trong trình duyệt. Phương thức `paint()` thực sự được kế thừa từ `java.awt`.

## Ví dụ “Hello, World” Applet trong Java

Sau đây là một applet đơn giản tên là `HelloWorldApplet.java`:

```
import java.applet.*;
import java.awt.*;

public class HelloWorldApplet extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString ("Hello World", 25, 50);
    }
}
```

Các lệnh `import` mang các lớp vào trong phạm vi của lớp applet của chúng ta:

- `java.applet.Applet`.
- `java.awt.Graphics`.

Không có các lệnh `import` này, bộ biên dịch trong Java sẽ không nhận ra các lớp `Applet` và `Graphics`, mà lớp applet tham chiếu tới.

## Lớp Applet trong Java

Mỗi applet là một kế thừa của lớp *java.applet.Applet*. Lớp Applet cơ bản cung cấp các phương thức mà một lớp Applet được nội suy có thể gọi để thu được thông tin và dịch vụ từ ngữ cảnh trình duyệt.

Nó bao gồm các phương thức mà thực hiện các công việc sau:

- Nhận các tham số applet.
- Nhận vị trí mạng của HTML file mà chứa applet đó.
- Nhận vị trí của thư mục lớp applet
- In một thông báo trạng thái (status) trong trình duyệt.
- Rút ra (Fetch) một hình ảnh
- Rút ra (Fetch) một audio
- Play một audio
- Phục hồi kích cỡ (resize) applet

Ngoài ra, lớp applet cung cấp một interface, bởi nó trình duyệt thu nhận thông tin về applet và điều khiển sự thực thi của applet. Viewer có thể:

- Yêu cầu thông tin về author, version và copyright của applet.
- Yêu cầu một mô tả của các tham số mà applet thừa nhận
- Khởi tạo applet
- Hủy applet
- Bắt đầu sự thực thi của applet
- Dừng sự thực thi của applet

Lớp Applet cung cấp các sự triển khai mặc định của mỗi phương thức này. Những sự triển khai này có thể được override nếu cần thiết.

## Gọi một applet trong Java

Một applet có thể được gọi bằng việc nhúng các chỉ thị trong một HTML file và quan sát file thông qua một applet viewer hoặc Java-enabled browser.

Thẻ <applet> là cơ sở để nhúng một applet trong một HTML file. Dưới đây là ví dụ gọi một applet là "Hello, World":

```
<html>
<title>The Hello, World Applet</title>
<hr>
<applet code="HelloWorldApplet.class" width="320" height="120">
If your browser was Java-enabled, a "Hello, World"
message would appear here.
</applet>
<hr>
</html>
```

**Ghi chú:** Bạn có thể tham khảo [Thẻ Applet trong HTML](#) để hiểu sâu hơn về cách gọi Applet từ HTML.

Thuộc tính code của thẻ <applet> là cần thiết. Nó xác định lớp Applet để chạy. Hai thuộc tính width và height cũng cần thiết để xác định kích thước ban đầu của ô trong đó applet chạy. Chỉ thị applet phải được đóng với một thẻ </applet>.

Nếu một applet nhận các tham số, các giá trị có thể được truyền cho các tham số bởi việc thêm các thẻ <param> giữa thẻ <applet> và </applet>. Trình duyệt bỏ qua text và các thẻ khác giữa các thẻ applet.

Các trình duyệt non-java-enabled không xử lý <applet> và </applet>. Vì thế, bất kỳ cái gì xuất hiện giữa các thẻ này, không liên quan tới applet, là nhìn thấy trong các trình duyệt đó.

Viewer và Browser tìm Java code được biên dịch tại vị trí của tài liệu. Nếu không thì, để xác định, sử dụng thuộc tính codebase của thẻ <applet> như sau:

```
<applet codebase="http://amrood.com/applets"
code="HelloWorldApplet.class" width="320" height="120">
```

Nếu một applet cư trú trong một gói thay vì mặc định, gói đang giữ này phải được xác định trong thuộc tính code bởi sử dụng ký tự dấu chấm (.) để phân biệt các thành phần của package/class. Ví dụ:

```
<applet code="mypackage.subpackage.TestApplet.class"
        width="320" height="120">
```

## Nhận các tham số của applet trong Java

Ví dụ sau minh họa cách tạo một Applet Respond để thiết lập các tham số được xác định trong tài liệu. Applet này hiển thị một checkerboard pattern (dạng bàn cờ) màu đen và một màu thứ hai.

Màu thứ hai và kích cỡ của mỗi hình vuông có thể được xác định như là các tham số tới applet bên trong tài liệu.

CheckerApplet nhận các tham số của nó trong phương thức *init()*. Nó cũng có thể nhận các tham số này trong phương thức *paint()*. Tuy nhiên, việc nhận các giá trị và lưu giữ các thiết lập một lần tại trạng thái start của applet, thay vì tại mỗi khi refresh, là tiện lợi và hiệu quả hơn.

Applet Viewer hoặc Browser gọi phương thức *init()* của mỗi applet nó chạy. Viewer này gọi *init()* một lần, ngay sau khi tải applet. (*Applet.init()* được triển khai để không làm cái gì cả.) Override sự triển khai mặc định để chèn custom code khởi tạo.

Phương thức *Applet.getParameter()* gọi ra một tham số mà đã cung cấp tên (giá trị của một tham số luôn luôn là một chuỗi). Nếu giá trị là số hoặc dữ liệu khác chuỗi, thì String phải được parse.

Sau đây là một khung của *CheckerApplet.java*:

```
import java.applet.*;
import java.awt.*;

public class CheckerApplet extends Applet
{
    int squareSize = 50; // initialized to default size

    public void init () {}

    private void parseSquareSize (String param) {}

    private Color parseColor (String param) {}

    public void paint (Graphics g) {}
}
```

Đây là phương thức `init()` của `CheckerApplet` và phương thức riêng `parseSquareSize()`:

```
public void init ()
{
    String squareSizeParam = getParameter ("squareSize");
    parseSquareSize (squareSizeParam);
    String colorParam = getParameter ("color");
    Color fg = parseColor (colorParam);
    setBackground (Color.black);
    setForeground (fg);
}
private void parseSquareSize (String param)
{
    if (param == null) return;
    try {
        squareSize = Integer.parseInt (param);
    }
    catch (Exception e) {
        // Let default value remain
    }
}
```

Applet gọi phương thức `parseSquareSize()` để parse tham số `squareSize`. Phương thức này lại gọi phương thức thư viện `Integer.parseInt()`, mà parse một string và trả về một integer. Phương thức `Integer.parseInt()` ném một exception bất cứ khi nào tham số của nó là hết hiệu lực (không hợp lệ).

Vì thế, phương thức `parseSquareSize()` bắt các exception, hơn là cho phép applet thất bại trên bad input.

Applet gọi phương thức `parseColor()` để parse tham số `color` vào trong một giá trị `Color`. Phương thức này thực hiện một loạt so sánh về chuỗi để so khớp giá trị tham số với tên của một màu đã định trước. Bạn cần triển khai những phương thức này để làm applet này làm việc.

## Xác định các tham số của applet trong Java

Sau đây là ví dụ của một HTML file với một CheckerApplet được nhúng trong nó. HTML file này xác định cả các tham số tới applet đó bởi thẻ <param>.

```
<html>
<title>Checkerboard Applet</title>
<hr>
<applet code="CheckerApplet.class" width="480" height="320">
<param name="color" value="blue">
<param name="squaresize" value="30">
</applet>
<hr>
</html>
```

**Ghi chú:** Tên các tham số là không phân biệt kiểu chữ.

## Biến đổi ứng dụng thành Applet trong Java

Nó là dễ dàng để biến đổi một ứng dụng Graphic Java (đó là, một ứng dụng mà sử dụng AWT và bạn có thể bắt đầu với Java program launcher) thành một applet mà bạn có thể nhúng vào trong trang Web.

Here are the specific steps for converting an application to an applet.

- Tạo một trang HTML với thẻ thích hợp để tải applet code.
- Cung cấp một lớp phụ của lớp JApplet, và làm lớp này là public. Nếu không thì, applet này không thể được tải.
- Loại bỏ phương thức main trong ứng dụng. Không xây dựng một frame window cho ứng dụng. Ứng dụng của bạn sẽ được hiển thị bên trong trình duyệt.
- Di chuyển bất kỳ code khởi tạo nào từ Frame window constructor tới phương thức init của applet. Bạn không cần xây dựng rõ ràng đối tượng applet. Trình duyệt khởi tạo nó cho bạn và gọi phương thức init.
- Gỡ bỏ triệu hồi tới setSize; với các applet, việc tạo kích thước được thực hiện bởi các tham số width và height trong HTML file.

- Gỡ bỏ triệu hồi tới `setDefaultCloseOperation`. Một applet không thể bị đóng; nó kết thúc khi trình duyệt thoát ra.
- Nếu ứng dụng gọi `setTitle`, thì loại bỏ triệu hồi tới phương thức này. Applet không thể có các Title bar. (Tất nhiên bạn có thể tạo tiêu đề cho chính trang web đó, bởi sử dụng thẻ HTML `title`.)
- Đừng gọi phương thức `setVisible(true)`. Applet được hiển thị một cách tự động.

## Xử lý sự kiện trong Java

Applet kế thừa một nhóm các phương thức event-handling từ lớp `Container`. Lớp `Container` định nghĩa một số phương thức, như `processKeyEvent` và `processMouseEvent`, để xử lý các kiểu sự kiện cụ thể, và sau đó một phương thức catch-all được gọi `processEvent`.

Để phản ứng lại một sự kiện, một applet phải override phương thức event cụ thể.

```
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.applet.Applet;
import java.awt.Graphics;

public class ExampleEventHandling extends Applet
    implements MouseListener {

    StringBuffer strBuffer;

    public void init() {
        addMouseListener(this);
        strBuffer = new StringBuffer();
        addItem("initializing the apple ");
    }

    public void start() {
        addItem("starting the applet ");
    }
}
```



```
public void stop() {
    addItem("stopping the applet ");
}

public void destroy() {
    addItem("unloading the applet");
}

void addItem(String word) {
    System.out.println(word);
    strBuffer.append(word);
    repaint();
}

public void paint(Graphics g) {
    //Draw a Rectangle around the applet's display area.
    g.drawRect(0, 0,
               getWidth() - 1,
               getHeight() - 1);

    //display the string inside the rectangle.
    g.drawString(strBuffer.toString(), 10, 20);
}

public void mouseEntered(MouseEvent event) {
}

public void mouseExited(MouseEvent event) {
}

public void mousePressed(MouseEvent event) {
}

public void mouseReleased(MouseEvent event) {
```

```
    }

    public void mouseClicked(MouseEvent event) {
        addItem("mouse clicked! ");
    }
}
```

Bây giờ, chúng ta gọi applet này như sau:

```
<html>
<title>Event Handling</title>
<hr>
<applet code="ExampleEventHandling.class"
width="300" height="300">
</applet>
<hr>
</html>
```

Ban đầu, applet sẽ hiển thị "initializing the applet. Starting the applet." Sau đó, khi bạn click vào bên trong hình chữ nhật, thì "mouse clicked" cũng sẽ được hiển thị.

## Hiển thị hình ảnh trong Java

Một applet có thể hiển thị hình ảnh trong định dạng GIF, JPEG, BMP, etc. Để hiển thị một hình ảnh bên trong một applet, bạn sử dụng phương thức *drawImage()* trong lớp *java.awt.Graphics*.

Ví dụ sau minh họa tất cả các bước để hiển thị hình ảnh trong Java:

```
import java.applet.*;
import java.awt.*;
import java.net.*;

public class ImageDemo extends Applet
{
    private Image image;
    private AppletContext context;

    public void init()
    {
```

```
context = this.getAppletContext();
String imageURL = this.getParameter("image");
if(imageURL == null)
{
    imageURL = "java.jpg";
}
try
{
    URL url = new URL(this.getDocumentBase(), imageURL);
    image = context.getImage(url);
} catch(MalformedURLException e)
{
    e.printStackTrace();
    // Display in browser status bar
    context.showStatus("Could not load image!");
}
}
public void paint(Graphics g)
{
    context.showStatus("Displaying image");
    g.drawImage(image, 0, 0, 200, 84, null);
    g.drawString("www.javalicense.com", 35, 100);
}
}
```

Bây giờ, chúng ta gọi applet này như sau:

```
<html>
<title>The ImageDemo applet</title>
<hr>
<applet code="ImageDemo.class" width="300" height="200">
<param name="image" value="java.jpg">
</applet>
<hr>
```

&lt;/html&gt;

## Play một Audio trong Java

Một applet có thể play một audio file được biểu diễn bởi AudioClip interface trong gói java.applet. AudioClip interface có 3 phương thức, gồm:

- **public void play():** Play một audio clip một lần từ khi bắt đầu.
- **public void loop():** Làm audio clip được replay một cách liên tục.
- **public void stop():** Dừng play một audio clip.

Để nhận một đối tượng AudioClip, bạn phải triệu hồi phương thức getAudioClip() của lớp Applet. Phương thức getAudioClip() trả về ngay lập tức, có hay không URL là chuyển tới một audio file thực sự. Audio file này không được tải tới khi một sự cố gắng được tạo để play một audio clip đó.

Following is the example showing all the steps to play an audio:

```
import java.applet.*;
import java.awt.*;
import java.net.*;

public class AudioDemo extends Applet
{
    private AudioClip clip;
    private AppletContext context;

    public void init()
    {
        context = this.getAppletContext();
        String audioURL = this.getParameter("audio");
        if(audioURL == null)
        {
            audioURL = "default.au";
        }
        try
        {
            URL url = new URL(this.getDocumentBase(), audioURL);
```

```
        clip = context.getAudioClip(url);
    }catch(MalformedURLException e)
    {
        e.printStackTrace();
        context.showStatus("Could not load audio file!");
    }
}
public void start()
{
    if(clip != null)
    {
        clip.loop();
    }
}
public void stop()
{
    if(clip != null)
    {
        clip.stop();
    }
}
}
```

Bây giờ, chúng ta gọi applet này như sau:

```
<html>
<title>The ImageDemo applet</title>
<hr>
<applet code="ImageDemo.class" width="0" height="0">
<param name="audio" value="test.wav">
</applet>
<hr>
</html>
```

Bạn có thể sử dụng test.wav tại PC của bạn để test ví dụ trên.

Để hiểu sâu hơn các khái niệm được trình bày trong chương này, mời bạn tham khảo loạt bài: **Ví dụ về Applet trong Java**.