

JAVATM PROGRAMMING



Chapter 17: Applets, Images, and Sound



Objectives

- Learn about applets
- Write an HTML document to host an applet
- Use the `init()` method
- Work with `JApplet` components
- Understand the `JApplet` life cycle
- Understand multimedia and use images
- Add sound to `JApplets`



Introducing Applets – Part 1

- **Applets**
 - Java programs called from within another application
 - Frequently are run from a Web page
 - Can contain any number of components
 - Can respond to user-initiated events
 - Behaviors come from a Java class named `JApplet`



Introducing Applets – Part 2

- Similar to a Java application
 - Use a .java file extension
 - Compile into bytecode using the `javac` command
 - Saved with the .class extension
 - Can contain:
 - Methods you define
 - Variables and constants
 - Decisions, loops, and arrays
 - GUI elements



Introducing Applets – Part 3

- Different from a Java application
 - Descend from the `JApplet` class
 - Run from another application
 - Do not use the `java` command to execute an applet
 - Do not contain a `main()` method
 - Do not set a default close operation
 - Cannot delete, read, or create files on the user's system
 - Cannot run any other program on the user's system



Understanding the JApplet Class

- Import JApplet
 - `import javax.swing.JApplet;`
- **JApplet**
 - A `Swing` class from which you can inherit



Running an Applet

- You run an applet from a document, usually in HTML
- **HTML, or Hypertext Markup Language**
 - A simple language used to create Web pages for the Internet
 - Contains many commands
- Two ways to run an applet:
 - In a Web browser
 - Using the Applet Viewer
 - `appletviewer` command

Writing an HTML Document to Host an Applet – Part 1

- When you create an applet:
 - Write the applet in Java
 - Save with a .java file extension
 - Compile the applet into bytecode using the `javac` command
 - Write an HTML document that includes a statement to call your compiled Java class
 - Load the HTML document into a Web browser or run the Applet Viewer program

Writing an HTML Document to Host an Applet – Part 2

- **Web browser**
 - A program that allows the display of HTML documents
 - Often contains Java applets
- Untrusted code carries the possibility of doing harm
 - Applet code is not trusted
- Sandbox
 - A safe area in which applet code can run

Writing an HTML Document to Host an Applet – Part 3

- Run an applet from within an HTML document:

```
<HTML>
```

```
  <object code = "AClass.class" width =  
    300 height = 200> </object>
```

```
</HTML>
```

- Three object tag attributes:
 - code
 - The name of the compiled applet
 - width
 - height

Writing an HTML Document to Host an Applet – Part 4

- Pixels
 - Set with `width` and `height`
- `<applet>` and `</applet>` tag set
 - Can be used instead of the `<object>` tag
 - It's better to use the `<object>` tag

Using the `init()` Method – Part 1

- JApplet class methods
 - Invoked by the Web browser when the browser runs an applet
 - `public void init()`
 - `public void start()`
 - `public void paint()`
 - `public void stop()`
 - `public void destroy()`

Using the `init()` Method – Part 2

```
import javax.swing.*;
import java.awt.*;
public class JHello extends JApplet
{
    JLabel greeting = new JLabel("Hello. Who are you?");
    public void init()
    {
        add(greeting);
    }
}
```

Figure 17-3 The JHello JApplet



Working with JApplet Components

- All the techniques that are used with JFrames can also be used with JApplets:
 - Change the font and color of labels
 - Use layout managers
 - Add multiple GUI components
 - Change the background color
 - Add listeners for user events
 - Add images and sounds



Understanding the JApplet Life Cycle

- Applets are popular
 - Easy to use in Web pages
- JApplet class
 - Contains methods automatically called by the browser:
 - `init()`
 - `start()`
 - `stop()`
 - `destroy()`



The `init()` Method

- `init()` method
 - Executes when a Web page containing a `JApplet` is loaded or when the `appletviewer` command is run



The `start()` Method

- **`start()` method**
 - Executes after the `init()` method
 - Executes again every time the applet becomes active after it has been inactive



The `stop()` Method

- **`stop()` method**
 - Invoked when a user leaves a Web page



The `destroy()` Method – Part 1

- **`destroy()` method**
 - Called when the user closes the browser or the Applet Viewer
 - Releases any resources `JApplet` might have allocated
- Every `JApplet` has the same life cycle outline

The `destroy()` Method – Part 2

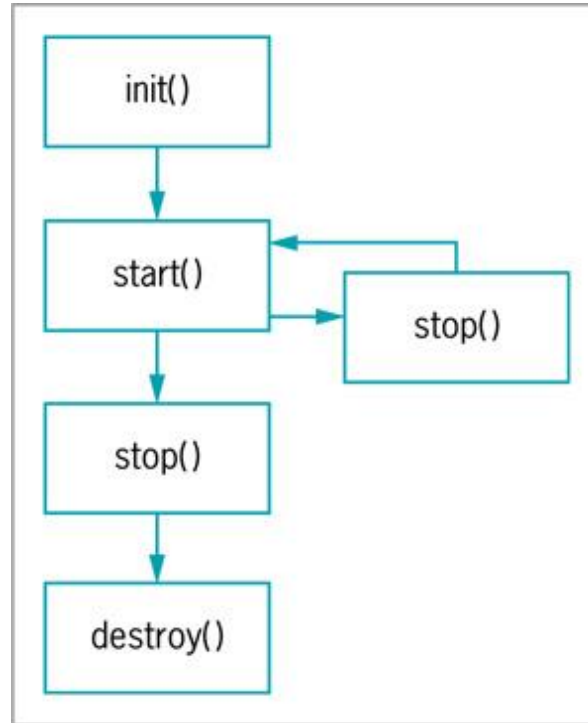


Figure 17-10 The JApplet life cycle

Understanding Multimedia and Using Images

- **Multimedia**
 - The use of sound, images, graphics, and video in computer programs
- Java provides extensive multimedia tools:
 - Java 2D or Java 3D Application Programming Interface (API)
 - Java Media Framework (JMF) API
 - Java Sound
 - Java Advanced Imaging API
 - Java Speech API

Adding Images to Japplets – Part 1

- **Image**
 - A likeness of a person or thing
- Image formats supported by Java:
 - Graphics Interchange Format (GIF)
 - Joint Photographic Experts Group (JPEG)
 - Portable Network Graphics (PNG)
- Image **class**
 - Provides many of Java's image capabilities

Adding Images to JApplets – Part 2

- **Declare an Image**
 - `companyLogo = getImage(getCodeBase(), "logo.gif");`
- **Use the applet `paint()` method to display an Image**
- **`drawImage()` method**
 - Is a `Graphics` method
- **`ImageObserver`**
 - Can be any object that implements the `ImageObserver` interface
 - All `Components` inherit this implementation

Figure 17-15 The JCompanyImage JApplet

```
import java.awt.*;
import java.applet.*;
import javax.swing.*;
public class JCompanyImage extends JApplet
{
    Image companyLogo;
    final int WIDTH = 287;
    final int HEIGHT = 129;
    final int FACTOR = 2;
    public void init()
    {
        companyLogo = getImage(getCodeBase(), "CompanyLogo.png");
    }
    public void paint(Graphics g)
    {
        super.paint(g);
        // Draw image at its natural size
        g.drawImage(companyLogo, 0, 0, this);
        // Draw the image scaled - twice as large
        g.drawImage(companyLogo, 0, HEIGHT, WIDTH * FACTOR,
            HEIGHT * FACTOR, this);
    }
}
```


Adding Images to Japplets – Part 3

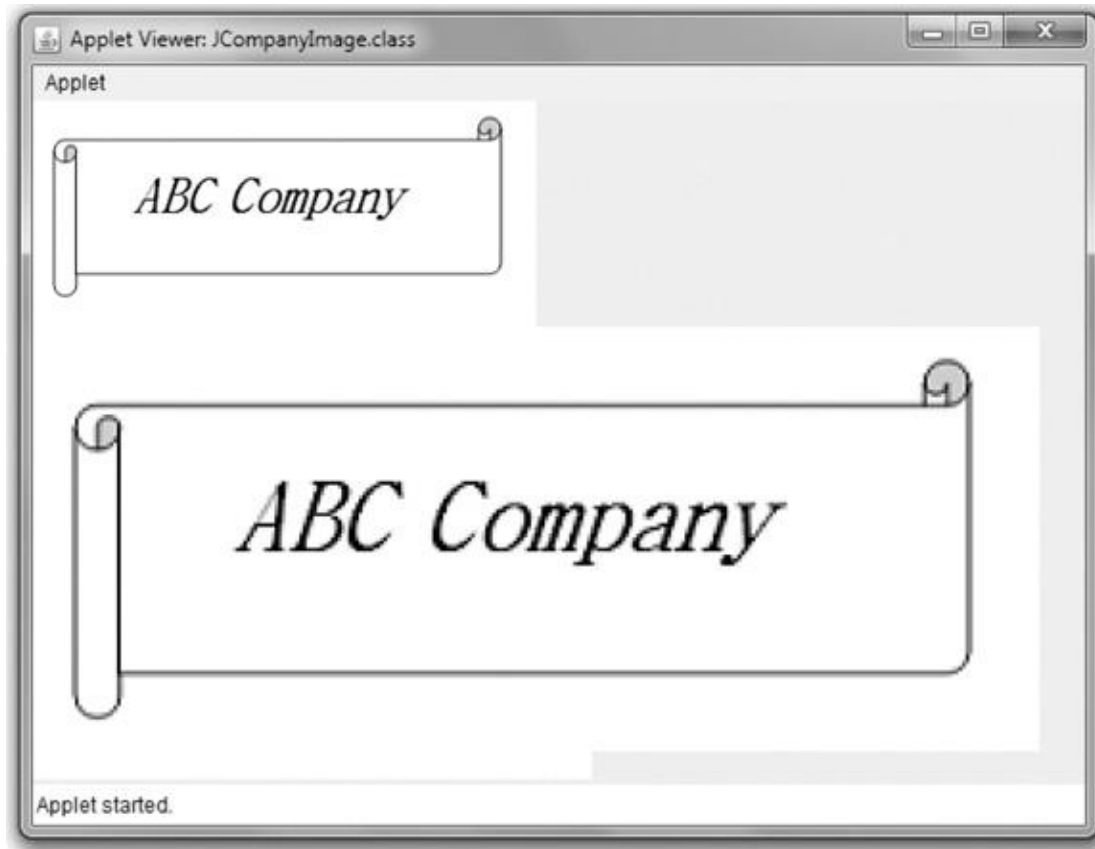


Figure 17-17 Output of the JCompanyImage JApplet



Using ImageIcon

- ImageIcon class
 - Creates images in applications and applets
 - Is simpler than working with Image
- You can place an ImageIcon on a Component

```
ImageIcon arrowPicture = new ImageIcon("arrow.gif");
JButton arrowButton = new JButton(arrowPicture);
```
- paintIcon() method
 - Displays ImageIcon images

Adding Sound to JApplets

- Java supports sound using methods from the `Applet` class
- `play()` method of the `Applet` class
 - The simplest way to retrieve and play a sound
 - Two forms
 - Codebase attribute
 - Indicates the filename of the applet's main class file
 - `getCodeBase()` method:

```
AudioClip aClip =  
new AudioClip(getCodeBase(), "tune.au");
```



You Do It

- Creating an HTML Document to Host an Applet
- Creating and Running a JApplet
- Running a JApplet in Your Web Browser
- Creating a More Complicated JApplet
- Making the JApplet's Button Respond to Events
- Understanding the Applet Life Cycle
- Creating an HTML Document to Host the JApplet
- Displaying Images
- Playing Sounds



Don't Do It

- Don't forget a matching closing tag for every opening tag in an HTML document
- Don't forget to use the .class extension with the name of a JApplet
- Don't add a `main()` method to a JApplet
- Don't try to execute an applet using the `java` command



Summary – Part 1

- Applets
 - Programs called from within another application
 - Run within a Web page or within another program called Applet Viewer
- Applet life cycle
 - `init()`
 - `start()`
 - `stop()`
 - `destroy()`



Summary – Part 2

- **Images**
 - Image
 - ImageIcon
- **Sounds**
 - Applet