

商业开发代码库系列

# Java

## 案例开发集锦

袁 然 郑自国 邹丰义 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

商业开发代码库系列

# Java案例开发集锦

袁 然 郑自国 邹丰义 编著

電子工業出版社.

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

本书精选了几十个来自实战中的Java案例，每个案例配有详细的实现步骤和完整的源代码解释。通过精确分析Java在页面特效、文件处理、游戏、动画处理、数据库、网络、安全、Web服务器部署等方面的案例开发过程，在描述Java应用技术知识的同时，展现Java的强大编程功能。

本书结构合理、内容丰富，可以作为Java编程爱好者提高编程水平的参考书，也可作为大中专院校计算机专业学生学习的辅助教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

### 图书在版编目（CIP）数据

Java案例开发集锦/袁然，郑自国，邹丰义编著. —北京：电子工业出版社，2005.9

（商业开发代码库系列）

ISBN 7-121-01599-4

I. J… II. ①袁… ②郑… ③邹… III. Java语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字（2005）第085077号

责任编辑：徐云鹏

特约编辑：卢国俊

印 刷：北京天竺颖华印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编：100036

北京市海淀区翠微东里甲2号 邮编：100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：31.75 字数：810千字

印 次：2005年9月第1次印刷

定 价：48.00元（含光碟1张）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换，若书店售缺，请与本社发行部联系。联系电话：010-68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

# 前 言

---

距离Sun公司第一次发布Java已经整整10年了，10年相对于计算机飞速发展的进程来说不算短，它足以淘汰掉许多技术，也足以考验真正的强者。时至今日，Java已成为Internet中最受欢迎、最具影响的编程语言之一，并且聚集了大批的拥趸，笔者也是其中之一。在与Java多年朝夕相处的日子里，我们积累了丰富的经验和教训，现呈献出来，希望能对大家有所裨益。

信息的价值在于使用和共享，Internet和Web是信息使用和共享的最快捷、最便宜的方式，Java连同Internet、WWW正在改变应用软件的开发和使用方式，一切都要围绕着网络，与平台无关。Java是面向对象的程序设计语言，通过解释方式来执行，具有跨平台性、简单、面向对象、分布式、可靠、安全、结构中立性、可移植性、高性能、多线程、动态性等许多优点。Java不仅是一种开发语言，还是一个平台，Java平台包括两部分内容：Java虚拟机和Java API。

本书从实际编程出发，共收集了几十个案例，每个案例详细讲解了实现的步骤和制作的要点，并提供了源代码。本书所有案例均调试运行通过，运行环境和配置在书中或者配书光碟中都有说明。我们通过案例学习Java的应用，以便能尽快进入实战状态。

全书共分10章，第1章 Java与Applet，收集了10个应用Applet技术的案例，从案例实现步骤的讲解中反映Applet技术的重点。

第2章 Java与特效，通过12个页面特效和文字特效的例子，说明Java在处理特效方面的独到之处，利用Java能轻松制作又酷又炫的网页特效。

第3章 XML与其他，XML是数据描述的标准，Java和XML的组合具有广泛而深远的意义，本章收录了3个关于XML处理的程序和其他6个小应用。

第4章 Java与游戏，Java与游戏的结合已成为新一代网络游戏的主角。本章有6个好玩的游戏供读者欣赏，同时讲解了在编写游戏过程中用到的方法和技巧。

第5章 Java与文件操作，提供了10个文件处理的案例。计算机上的各种资源，无论是外围设备，还是硬盘的数据资料，都是以“文件”方式进行访问处理的，文件操作是编程中的重要内容之一，Java在这方面提供了丰富的类和方法。

第6章 Java与安全, Java本身具有很强的安全性, 本章共介绍了7个加密和数字证书方面的案例。

第7章 Java与数据库, 通过6个案例讲解数据库操作在Java中的实现方法。

第8章 Java与Servlet, Servlet运行在服务器端, 本章共有8个案例, 说明如何部署Servlet。

第9章 Java与网络, 本章提供8个案例来说明网络设计语言——Java在网络应用上的得心应手。

第10章介绍了4个综合案例。

本书案例中涉及到Java的类、对象、接口、继承、线程、异常处理、I/O流、Applet、Servlet、NET、XML、Security、DB、File、JavaBean、Socket、JVM等内容, 同时也体现了MVC的编程思想。

本书的分类并不严格, 比如第1章, Java与Applet是按照应用技术进行组织的, 后面章节中的特效应用和有些游戏是按应用类型分类的, 但也用到了Applet技术。笔者想全力展现Java的博大精深与魅力, 但因为水平有限, 有时辞不达意, 敬请读者谅解。

本书的编写过程得到了电子工业出版社和北京美迪亚电子信息有限公司和山东大众信息产业有限公司的大力协助及许多网友的支持, 在此表示衷心的感谢。参加本书编校的还有王明杰、平凡、韩冬佑、王艳、郭莹、刘有志、杨萍、陈圣琳、向小平等。由于时间仓促, 且作者水平有限, 书中的错误在所难免, 真诚希望广大读者提出宝贵意见。

谨向阅读本书的读者朋友们表示诚挚的敬意和谢意。

# 目 录

---

---

|                                 |     |
|---------------------------------|-----|
| 第1章 Java与Applet .....           | 1   |
| 案例1 图形按钮 .....                  | 1   |
| 案例2 模拟工具条 .....                 | 8   |
| 案例3 Applet与Applet在页内的通信 .....   | 13  |
| 案例4 电子相册 .....                  | 16  |
| 案例5 百叶窗效果 .....                 | 19  |
| 案例6 波浪彩虹文字 .....                | 27  |
| 案例7 3D立体渐层文字 .....              | 35  |
| 案例8 飞行文字 .....                  | 38  |
| 案例9 聚光灯效果 .....                 | 41  |
| 案例10 伸展文 .....                  | 45  |
| 本章小结 .....                      | 49  |
| 第2章 Java与特效 .....               | 50  |
| 案例1 火焰招牌 .....                  | 50  |
| 案例2 闪电中的城市 .....                | 54  |
| 案例3 激光绘图 .....                  | 60  |
| 案例4 水面倒影 .....                  | 66  |
| 案例5 图片放大镜 .....                 | 71  |
| 案例6 浮动的气泡 .....                 | 76  |
| 案例7 烟花汇演 .....                  | 81  |
| 案例8 星空模拟 .....                  | 88  |
| 案例9 阴影跑马灯 .....                 | 94  |
| 案例10 下雪的图片 .....                | 98  |
| 案例11 动态分割线 .....                | 102 |
| 案例12 飞流直下 .....                 | 106 |
| 本章小结 .....                      | 109 |
| 第3章 XML与其他 .....                | 110 |
| 案例1 将HTML文件转成XML文件 .....        | 110 |
| 案例2 把XML文件转换成可浏览的HTML格式文件 ..... | 113 |
| 案例3 用JDOM解析XML文件 .....          | 116 |

|            |                                   |            |
|------------|-----------------------------------|------------|
| 案例4        | Java编制的时钟 .....                   | 120        |
| 案例5        | 简单日历 .....                        | 124        |
| 案例6        | 系统内存状态监视程序 .....                  | 130        |
| 案例7        | 简单的计算器 .....                      | 134        |
| 案例8        | 多线程断点续传 .....                     | 144        |
| 案例9        | 笛卡儿曲线 .....                       | 153        |
| 本章小结       | .....                             | 155        |
| <b>第4章</b> | <b>Java与游戏 .....</b>              | <b>156</b> |
| 案例1        | 幸运52游戏 .....                      | 156        |
| 案例2        | “速算24”扑克游戏 .....                  | 163        |
| 案例3        | 拼图游戏 .....                        | 172        |
| 案例4        | 贪吃蛇游戏 .....                       | 189        |
| 案例5        | 打球游戏 .....                        | 196        |
| 案例6        | 棒打猪头 .....                        | 207        |
| 本章小结       | .....                             | 219        |
| <b>第5章</b> | <b>Java与文件操作 .....</b>            | <b>220</b> |
| 案例1        | 目录列表的显示 .....                     | 220        |
| 案例2        | 检查与创建目录 .....                     | 223        |
| 案例3        | 文件复制 .....                        | 227        |
| 案例4        | 文件查看器 .....                       | 231        |
| 案例5        | 字符串的查找和替换 .....                   | 235        |
| 案例6        | 利用RandomAccessFile类来实现文件的追加 ..... | 242        |
| 案例7        | 用Zip进行多文件保存 .....                 | 244        |
| 案例8        | 用JUNIT建立测试类 .....                 | 247        |
| 案例9        | 用Java保存位图文件 .....                 | 252        |
| 案例10       | 获取文件属性 .....                      | 261        |
| 本章小结       | .....                             | 265        |
| <b>第6章</b> | <b>Java与安全 .....</b>              | <b>266</b> |
| 案例1        | 用户登录验证的完整程序 .....                 | 266        |
| 案例2        | MD5的JavaBean实现 .....              | 272        |
| 案例3        | 用公钥计算消息摘要的验证码 .....               | 278        |
| 案例4-1      | Java中数字证书的生成及维护方法 .....           | 283        |
| 案例4-2      | 数字证书的签发(签名) .....                 | 288        |
| 案例4-3      | 利用数字证书给Applet签名 .....             | 296        |
| 案例5        | 利用DES加密解密 .....                   | 303        |
| 本章小结       | .....                             | 311        |

|                            |     |
|----------------------------|-----|
| <b>第7章 Java与数据库</b>        | 312 |
| 案例1 在Applet中应用JDBC访问数据库    | 312 |
| 案例2 通过JDBC-ODBC桥连接数据库      | 323 |
| 案例3 通过Tomcat数据源访问数据库       | 327 |
| 案例4 JDBC连接池的实现             | 329 |
| 案例5 用JavaBean实现MySQL的分页显示  | 340 |
| 案例6 利用JDBC - ODBC查看查询结果    | 348 |
| 本章小结                       | 353 |
| <b>第8章 Java与Servlet</b>    | 354 |
| 案例1 利用Servlet打开非HTML格式的文档  | 354 |
| 案例2 Servlet和JSP的通信         | 359 |
| 案例3 Servlet和Servlet的通信     | 367 |
| 案例4 Servlet动态生成图像          | 370 |
| 案例5 用Servlet连接数据库          | 373 |
| 案例6 用Servlet实现页面注册和登录      | 377 |
| 案例7 运用Servlet实现BBS功能       | 388 |
| 案例8 侦听Web服务器信息             | 394 |
| 本章小结                       | 399 |
| <b>第9章 Java与网络</b>         | 400 |
| 案例1 显示你的IP                 | 400 |
| 案例2 用Socket进行客户与服务器通信      | 404 |
| 案例3 利用UDP Socket技术实现IP多点传送 | 409 |
| 案例4 利用Java API发送E-mail     | 416 |
| 案例5 从Mail Server删除一条消息     | 427 |
| 案例6 在Java程序中实现FTP的功能       | 431 |
| 案例7 一个最简单的聊天程序             | 437 |
| 案例8 代理服务器的实现               | 446 |
| 本章小结                       | 452 |
| <b>第10章 Java综合实例</b>       | 453 |
| 案例1 用Java实现zip压缩解压缩        | 453 |
| 案例2 简易图书管理系统               | 460 |
| 案例3 UFO攻击游戏                | 467 |
| 案例4 制作一个MP3播放器             | 488 |
| 本章小结                       | 500 |

# 第1章

## Java与Applet



### 本章内容

- 案例1 图形按钮
- 案例2 模拟工具条
- 案例3 Applet与Applet在页内的通信
- 案例4 电子相册
- 案例5 百叶窗效果
- 案例6 波浪彩虹文字
- 案例7 3D立体渐层文字
- 案例8 飞行文字
- 案例9 聚光灯效果
- 案例10 伸展文字
- 本章小结



### 案例1 图形按钮

#### 案例运行效果与操作

本案例是一个用Java Applet实现页面响应鼠标事件的例子。实现了鼠标移入、移出、点击事件的响应功能。程序运行后，界面如图1-1所示。

把鼠标移到图片上面, 效果如图1-2所示, 并伴随播放声音。如果把鼠标移出, 将返回图1-1的状态。

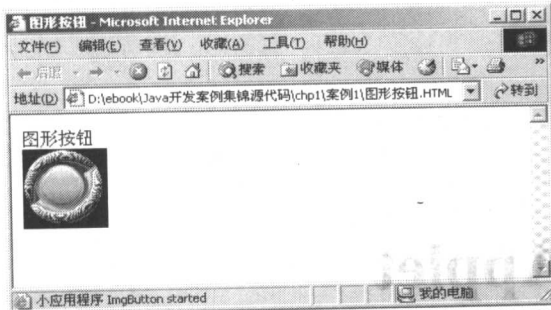


图1-1 运行界面

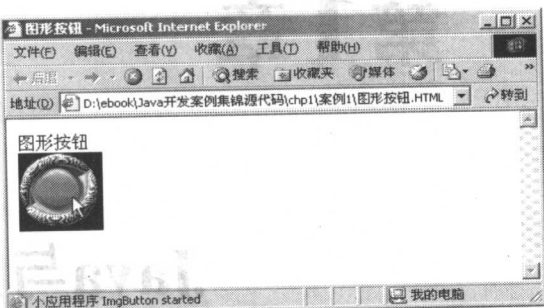


图1-2 鼠标移到图片上面的效果

单击图片, 程序会再次更改图片的显示效果, 如图1-3所示。

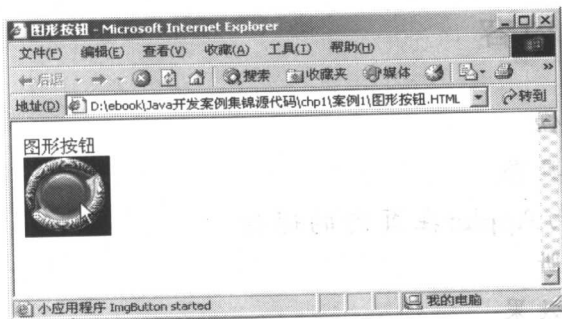


图1-3 单击图片产生的效果

### 制作要点

1. 获取HTML中的param的参数值
2. Applet的声音播放AudioClip类的使用
3. 鼠标事件的监听
4. MediaTracker类的使用

### 步骤详解

本例采用了Java小应用程序Java Applet, Java Applet首先需要嵌入到网页中, 执行时需下载到用户本地进行执行, 用户端需要安装Java虚拟机。

1. 获取HTML中的param的参数值。

我们先来看一下HTML的源代码:

```
<applet code=ImgButton.class width=75 height=75>
<param name=soundA value="midiA.mid">
<param name=soundB value="midiB.mid">
```

```

<param name=Images1      value="button1.gif">
<param name=Images2      value="button2.gif">
<param name=Images3      value="button3.gif">
<param name=URL value=http://www.dzwww.com>
</applet>

```

我们需要在Java中取得这些参数，就要用java.applet.Applet类的方法：

```
public String getParameter(String name)
```

例如Applet标记在HTML网页中是这样的：

```

<applet code="Clock" width=50 height=50>
<param name=Color value="blue">
</applet>

```

我们在程序中调用方法getParameter("Color")，这个方法将返回值“blue”。

```

String colorStr = "";
colorStr = getParameter("Color");
if(colorStr==null){
    colorStr="blue"    ;
}

```

## 2. Applet的声音播放AudioClip类的使用。

图像格式各种各样，如BMP、GIF和JPEG等。声音文件也一样，WAV和AU是最常用的两种声音文件。在JDK1.2之前Java仅支持AU文件，从JDK1.2开始，Java提供了对WAV、MIDI等声音文件的支持。

使用Applet播放声音时，需首先定义AudioClip对象，getAudioClip方法能把声音赋予AudioClip对象，如果仅想播放一遍声音，应调用AudioClip类的play方法；如果想循环播放，应选用AudioClip类的loop方法。

```

sound1 = getAudioClip(getDocumentBase(), param);
sound1.play();    //播放一次
sound1.loop();    //循环播放

```

## 3. 鼠标点击事件的监听。

类java.awt.event.MouseListener共有以下5个方法来实现鼠标点击事件：

### 方法摘要

void

mouseClicked(MouseEvent e)当用户按下并松开鼠标按钮时发生。  
 用户在选择或双击图标的时候通常会点击鼠标按钮，如果在松开鼠标之前移动鼠标，点击不会导致鼠标相应事件出现。  
 因为点击鼠标是按下鼠标和松开鼠标的结合，在事件分配给mouseClicked()方法之前，mousePressed()和mouseReleased()方法已同时被调用。

|      |                                                    |
|------|----------------------------------------------------|
| void | mouseEntered(MouseEvent e)当鼠标离开当前组件并进入所监听的组件时激活事件。 |
| void | mouseExited(MouseEvent e)当鼠标离开所监听的组件时发生。           |
| void | mousePressed(MouseEvent e)当用户按下鼠标按钮时发生。            |
| void | mouseReleased(MouseEvent e)当用户松开鼠标按钮时发生。           |

①处理鼠标移入事件:

```
offG.drawImage(img1, 0, 0, width, height, this);
```

②处理鼠标按钮松开事件:

```
offG.drawImage(img2, 0, 0, width, height, this);
repaint();
soundB.play();
getAppletContext().showDocument(url);
```

③单击鼠标左键事件:

```
offG.drawImage(img3, 0, 0, width, height, this);
repaint();
soundA.play();
System.out.println("soundB play");
```

#### 4. MediaTracker类的使用。

Java专门提供了用于跟踪包括图像和声音等多媒体对象的ImageObserver类和MediaTracker类。

在本书程序中主要用到的是跟踪多幅图像状态的MediaTracker类。

```
imageTracker = new MediaTracker(this);
try {
    //开始装载所有图片，多媒体对象imageTracker会跟踪所有指定图片的状态。
    //这个方法会等待，直到所有指定的图片都转载完成，可能会抛出异常
    InterruptedException
    imageTracker.waitForID(0);
} catch (InterruptedException e) {
}
```

#### MediaTracker的主要方法及用途

| 返回类型 | 方法                               | 用途                                |
|------|----------------------------------|-----------------------------------|
|      | MediaTracker<br>(Component comp) | 构造方法，为指定的组件comp创建一个MediaTracker对象 |

| 返回类型    | 方法                                                                 | 用途                                                                                  |
|---------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| void    | <b>addImage</b><br>(Image image, int id)                           | 将图片加入到 <b>MediaTracker</b> 的监视队列中去, image为要被监视的图像对象, id为监视图像在监视队列中的标识号              |
| void    | <b>AddImage(Image</b><br>image,int id,int w,int h)                 | 将图片加入到 <b>MediaTracker</b> 的监视队列中去, w为所监视的图像对象的宽度, h为所监视的图像对象的高度                    |
| boolean | <b>CheckAll()</b>                                                  | 检查所有被监视的图像对象是否已经完成了装载过程。如果所有图像对象已经完成了装载, 则返回true, 否则为false                          |
| boolean | <b>CheckAll(Boolean load)</b>                                      | 检查所有被监视的图像对象是否已经完成了装载过程。如果load为true, 则开始装载那些还没有被装载的图像                               |
| boolean | <b>CheckId(int id)</b>                                             | 检查在监视队列中所有标识号为id的图像对象是否已经完成了装载过程, 如果已经完成了装载过程, 返回true                               |
| boolean | <b>CheckId</b><br>(int id,Boolean load)                            | 检查在监视队列中所有标识号为id的图像对象是否已经完成了装载过程, 如果load为true, 且在监视队列中的标识号为id的图像没有被装载的话, 则开始装载此图像对象 |
| boolean | <b>IsErrorId(int id)</b>                                           | 检查在监视队列中标识号为id的图像对象是否装载有错误, 如有错误, 则返回true                                           |
| void    | <b>RemoveImage</b><br>(Image image)                                | 从监视队列中移去指定的图像对象image                                                                |
| void    | <b>RemoveImage</b><br>(Image image,int id)                         | 从监视队列中移去指定id的图像对象image                                                              |
| void    | <b>RemoveImage(Image</b><br>image,int id,int width,<br>int height) | 从监视队列中移去指定id的图像对象image, 且这个图像对象被指定了大小                                               |
| void    | <b>WaitForAll()</b>                                                | 开始装载监视队列中所有没有被装载的图像对象, 如果装载不成功, 则会抛出一个异常                                            |
| boolean | <b>WaitForAll(long ms)</b>                                         | 开始装载监视队列中所有没有被装载的图像对象。但是, 如果装载时间超时, 则会取消装载, 并返回false                                |

| 返回类型    | 方法                        | 用途                                              |
|---------|---------------------------|-------------------------------------------------|
| void    | WaitForId(int id)         | 开始装载监视队列中标识号为id的图像对象，如果装载不成功，则会抛出一个异常           |
| boolean | WaitForId(int id,long ms) | 开始装载监视队列中标识号为id的图像对象。但是如果装载时间超时，则会取消装载，并返回false |

### 程序源代码与解释

```

/**
 *本案例是一个用java applet实现页面响应鼠标事件的例子。
 *实现了鼠标移入、移出、点击事件的响应的功能。
 */
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.net.*;

//页面响应鼠标事件
public class ImgButton extends Applet implements MouseListener {
    private int width, height;           //定义图像按钮的宽度和高度
    private Image offI, img1, img2, img3; //定义Image图像
    private Graphics offG;                //定义Graphics图像
    private MediaTracker imageTracker;     //定义图像跟踪器
    private boolean isMouseEnter = false, isPress = false; //定义鼠标状态变量
    private Color ButtonColor = Color.yellow, lightC, darkC; //定义按钮颜色
    private URL url;                      //定义超链接变量
    private AudioClip soundA, soundB;     //定义声音文件变量，以便读取声音文件
    private String param;                 //定义参数变量，以便从HTML文件获取参数
    public void init() {                  //Applet的初始化方法
        param = new String();
        //从HTML文件获取声音文件参数
        param = getParameter("soundA");
        if (param == null)
            param = "midiA.mid";
        soundA = getAudioClip(getDocumentBase(), param);
        param = getParameter("soundB");
        if (param == null)
            param = "midiB.mid";
        soundB = getAudioClip(getDocumentBase(), param);
        //从HTML文件获取图形按钮尺寸

```

```

width = getSize().width;
height = getSize().height;
//从HTML文件获取超链接参数
param = getParameter("URL");
try {
    url = new URL(param);
} catch (MalformedURLException e) {
}
//创建图像并加载到图像跟踪器
offI = createImage(width, height);
offG = offI.getGraphics();
imageTracker = new MediaTracker(this);
param = getParameter("Images1");
img1 = getImage(getCodeBase(), param);
imageTracker.addImage(img1, 0);
param = getParameter("Images2");
img2 = getImage(getCodeBase(), param);
imageTracker.addImage(img2, 0);
param = getParameter("Images3");
img3 = getImage(getCodeBase(), param);
imageTracker.addImage(img3, 0);
try {
    imageTracker.waitForID(0);
} catch (InterruptedException e) {
}
//安装鼠标事件监听器
addMouseListener(this);
}

public void start() { //Applet的启动方法
    offG.drawImage(img1, 0, 0, width, height, this);
    repaint();
}

public void mouseClicked(MouseEvent e) { //点击鼠标事件
}

public void mousePressed(MouseEvent e) { //点住鼠标不放产生的事件
    offG.drawImage(img3, 0, 0, width, height, this);
    repaint(); //重画图像
    soundA.play(); //播放声音文件
}

public void mouseReleased(MouseEvent e) { //松开鼠标产生的事件
    offG.drawImage(img2, 0, 0, width, height, this);
    repaint(); //重画图像
    soundB.play(); //播放声音文件
}

```

```

        getAppletContext().showDocument(url); //链接到指定网页
    }
    public void mouseEntered(MouseEvent e) { //鼠标移入事件
        offG.drawImage(img2, 0, 0, width, height, this);
        repaint();
    }
    public void mouseExited(MouseEvent e) { //鼠标移出事件
        offG.drawImage(img1, 0, 0, width, height, this);
        repaint();
    }
    public void paint(Graphics g) { //画出图像
        g.drawImage(offI, 0, 0, width, height, this);
    }
}

```



## 案例2 模拟工具条

### 案例运行效果与操作

本案例是使用Java Applet响应鼠标事件，点击图标后转入相关的页面。程序运行后，界面如图1-4所示。

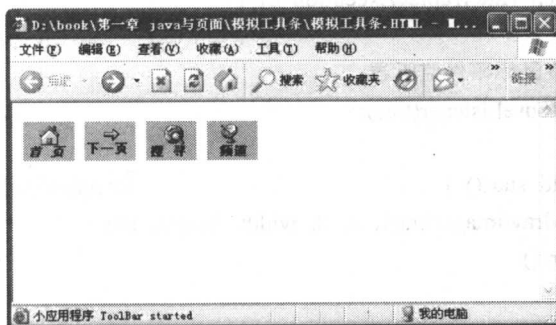


图1-4 运行界面

### 制作要点

1. URL显示
2. 鼠标点击事件的显示

### 步骤详解

在HTML源文件中定义好参数，由Applet获得参数，点击图片后，根据获得的参数，转入相关页面。

## 1. URL显示。

步骤如下：

- ①使用URL类按接收字符串生成URL对象。
- ②通过Applet类中的getAppletContext()方法取得小应用程序的AppletContext。
- ③通过AppletContext接口的showDocument方法显示URL内容。

该方法的具体格式和参数说明如下：

```
void showDocument(URL url , String target)
```

其中，target参数用于指定文件内容显示的位置，其具体值及作用如下：

|           |            |
|-----------|------------|
| “-self”   | 当前帧        |
| “-parent” | 父帧         |
| “-top”    | 最顶部的帧      |
| “-blank”  | 开辟新浏览器窗口显示 |
| “帧名”      | 在指定的帧中显示   |

## 2. 鼠标点击事件。

参考本章案例1：图形按钮。

## 3. 嵌入Applet的HTML页面源码。

```
<HTML>
<HEAD>
<TITLE>
模拟工具条
</TITLE>
</HEAD>
<BODY>
模拟工具条<BR>
<applet code="SimToolBar.class" width=52 height=45 >
  <PARAM name="target" value="http://www.dzwww.com">
  <PARAM name="enterImage" value="home2.gif">
  <PARAM name="exitImage" value="home.gif">
  <PARAM name="where" value="xinwen">
</applet>
<applet code="SimToolBar.class" width=52 height=45 >
  <PARAM name="target" value="http://bbs.dzwww.com">
  <PARAM name="enterImage" value="next.gif">
  <PARAM name="exitImage" value="next2.gif">
</applet>
<applet code="SimToolBar.class" width=52 height=45 >
  <PARAM name="target" value="http://www.baidu.com">
  <PARAM name="enterImage" value="search2.gif">
```