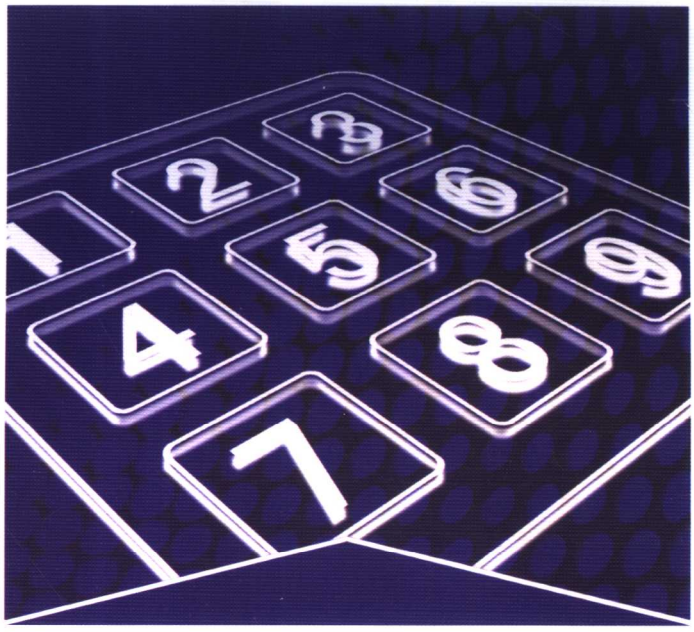
电脑编程实例导航丛书



Java 编程

典型实例解析

《电脑编程技巧与维护》杂志社 编著

中国水利水电出版社
www.waterpub.com.cn

电脑编程实例导航丛书

Java 编程典型实例解析

《电脑编程技巧与维护》杂志社　编著

中国水利水电出版社

内 容 提 要

本书根据 Java 的不同应用对象，精选了 56 个 Java 编程应用实例。全书分 4 章：第 1 章 Java 编程基础与应用实例，为初学 Java 编程和应用的读者提供了 29 个入门的实例；第 2 章数据库编程实例，为编程人员提供了 10 个使用 Java 进行数据库编程方法和技巧的实例；第 3 章图形图像与游戏编程实例，介绍了 7 个 Java 实现图形图像处理与游戏程序开发的编程技巧的实例；第 4 章网络与通信编程实例，介绍了 10 个 Java 在网络与通信应用方面的实例。这些典型实例所涵盖的编程方法和编程技巧都是 Java 编程高手们经验的积累与总结，具有代表性和典型性，值得读者参考和借鉴。

本书是 Java 编程人员经验和智慧的结晶，通过一个个典型实例解析总结了使用 Java 进行项目开发的技术难点和关键技术，详尽而具体地解说了重点和运用的编程技巧。该书对有 Java 应用基础的编程人员和项目开发人员和对初学 Java 编程的新手都是一本很好的参考资料。

本书附赠实例源代码，读者可以从中国水利水电出版社网站免费下载，网址为：http：//www.waterpub.com.cn/softdown/。

图书在版编目（CIP）数据

Java 编程典型实例解析/《电脑编程技巧与维护》杂志社编著. —北京：中国水利水电出版社，2006

（电脑编程实例导航丛书）

ISBN 7-5084-4103-6

Ⅰ.J… Ⅱ.电… Ⅲ.JAVA 语言—程序设计

Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2006）第 116328 号

书　　名	Java 编程典型实例解析
作　　者	《电脑编程技巧与维护》杂志社　编著
出版　发行	中国水利水电出版社（北京市三里河路 6 号　100044）
	网址：www.waterpub.com.cn
	E－mail：mchannel@263.net（万水）
	sales@waterpub.com.cn
	电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787mm×1092mm　16 开本　18.25 印张　623 千字
版　　次	2006 年 10 月第 1 版　2006 年 10 月第 1 次印刷
印　　数	0001—4000 册
定　　价	32.00 元

丛 书 序

《电脑编程技巧与维护》杂志是为从事电脑编程、系统应用开发的人员创办的专业性和实用性都很强的技术刊物。自1994年创刊以来，始终以“实用第一，智慧密集”为宗旨，坚持“质量第一”、“读者第一”的原则，为广大电脑编程爱好者、软件开发人员和专业计算机系统维护人员提供第一手技术资料、编程技巧和维护经验；紧密跟踪计算机软硬件技术发展和应用趋势，不断求变创新，针对软件开发过程中的许多关键技术问题着重提供各类解决方案，在业内获得一致好评，是广大编程和维护人员的首选刊物。在栏目内容上，选题覆盖面广，涉及技术领域宽、信息量大，帮助程序员开阔视野；在技术水平上，始终把握计算机技术发展的大方向，提供先进、详尽、准确的技术指导，并在长期工作中与国际性大公司建立了良好的合作关系，为读者提供全球最新、最全的实用信息；在实用性上，稿源来自于专业开发和维护人员的实践经验，是普通书籍难以获得的编程心得、体会与技巧。

2006年是《电脑编程技巧与维护》创刊十二周年，为了最大限度地开发和利用本刊宝贵而丰富的资源，更好地服务和真诚回报多年来一直关爱和支持本刊的广大读者，《电脑编程技巧与维护》杂志社和中国水利水电出版社共同策划出版了本套“电脑编程实例导航丛书”。

本套丛书包括《Visual C/C++系统开发典型实例解析》、《Visual C/C++图形图像与游戏编程典型实例解析》、《Visual Basic编程典型实例解析》、《Delphi编程典型实例解析》、《C#编程典型实例解析》、《Java编程典型实例解析》、《计算机系统安全与维护编程典型实例解析》、《计算机网络与通信编程典型实例解析》、《编程疑难问题解析126例》，一套9册共680多个典型实例。每册书的编程实例均依不同的编程应用分成若干章，条目清晰可查，使用极为方便。

这套丛书选编了《电脑编程技巧与维护》杂志近两年发表的和一部分尚未发表而又极为实用、精彩的典型编程实例。该套书的特点是：其各册内容来自编程高手的智慧和经验总结，其中不少文章的作者是业界资深程序员和技术专家，内容有深度、思路有新意、讲解深入浅出，编程技巧新颖实用，构思巧妙；丛书中的实例都是作者从实际项目提炼出的开发范例，实例讲解部分先给出设计目标，然后介绍实现目标的基本思想和方法，最后详细给出其核心程序的源代码，对程序的关键部分进行讲解并给出程序的运行效果；丛书中每一个实例

的程序源代码都经过上机调试通过，对编程中的疑难问题进行了深入解答，给程序开发人员移植源代码和学用编程带来了方便，加快了编程应用的步伐。全套书既讲究内容的深入性、专业性、权威性和实用性，同时兼顾轻松、通俗易懂、时效性强的特点。

本套丛书是《电脑编程技巧与维护》资源的二次深入开发，浓缩了当前主流编程语言Visual C/C++、Visual Basic、Delphi、Java、C#等程序设计的精华，其目的是力求为读者建造一个真正的知识整合、编程思想、编程技术、技巧交流的平台，让读者从中学习到编程高手的诀窍，丰富编程技巧，拓宽编程思路，迅速提升程序开发能力。对电脑编程人员来说，程序开发能力的提高，除了对语言和算法的不断钻研学习、不断实践、不断总结提高，练好基本功，打好基础外，还要集思广益，善于学习，善于借鉴参考别人的经验，深入透彻地理解其中的精髓，然后融入到自己的设计方案中去，这无疑是一条有效的学习途径，对于自身编程能力的增强和编程水平的迅速提高十分重要，这也正是我们编写本套丛书想要达到的目的。

本套丛书可作为高等院校学生进行课程项目开发、毕业项目设计的参考书，也可作为软件从业人员及编程爱好者的珍藏宝典，还可作为高等培训学校的实例教程。

《电脑编程技巧与维护》杂志社

中国水利水电出版社

2006年5月

前　言

在“电脑编程实例导航丛书”中，《Java 编程典型实例解析》精选了《电脑编程技巧与维护》杂志近两年共 24 期已发表的精彩编程实例 56 个。根据 Java 的不同应用对象，将精选的这 56 个实例分为 4 章。第 1 章 Java 编程基础与应用实例，为初学 Java 编程和应用的读者提供了 29 个入门的实例；第 2 章数据库编程实例，为编程人员提供了 10 个使用 Java 进行数据库编程方法和技巧的实例；第 3 章图形图像与游戏编程实例，介绍了 7 个使用 Java 实现图形图像处理与游戏程序开发的编程实例，第 4 章网络与通信编程实例，介绍了 10 个 Java 在网络与通信应用方面的实例。

全书每一章都本着实用第一的原则，深入地介绍了使用 Java 进行应用程序开发的编程方法与编程技巧。书中的每一个实例都给出了开发过程、技术难点及其解决的方法和技巧。这些典型案例所涵盖的 Java 编程技巧都是经验的总结，具有一定的代表性和典型性，很值得读者参考和借鉴。

本书的主要特色如下：第一，每一章都是通过一个个的实例来介绍 Java 应用编程方法和技巧，避免了枯燥、空洞的理论，并且每一个实例都具有很强的实用性和代表性。在实例的讲解上一般都是先给出设计目标，然后介绍实现该目标的基本思想和方法，最后详细给出其核心程序的源代码，并对程序的关键部分进行讲解，给出程序的运行效果。第二，所选的每一个实例都是从事 Java 应用编程人员的经验总结，具有很强的实用性，其中很多编程技巧可供借鉴。第三，每一个实例的程序源代码都经过上机调试通过，给程序开发人员移植源代码带来了方便。本书提供实例中的源程序代码，读者可以从中国水利水电出版社网站免费下载，网址为：http://www.waterpub.com.cn/softdown/。

本书是《电脑编程技巧与维护》宝贵资源的二次开发，浓缩了 Java 程序设计的精华。该书对于有 Java 应用基础的编程人员和应用开发人员以及初学 Java 编程的新手都有一定的参考价值。该书结构清晰、层次分明，实例典型而实用，但不足甚至疏漏之处在所难免，恳请广大读者批评指正。

《电脑编程技巧与维护》杂志社

2006 年 5 月

目　录

第 2 章　数据库编程实例

第 3 章　图形图像与游戏编程实例

第 4 章　网络与通信编程实例

第 1 章　基础与应用编程实例

实例 1 Java 基础知识应用编程

一、Java 数据类型及应用

1. Java 数据类型

Java 是一门强类型语言，每个变量都必须声明类型。其数据类型可以分为基本类型和扩展类型两大类。基本数据类型包括：boolean、char、byte、int、short、long、float、double 八种；扩展数据类型主要是指字符串类型（String）。

八种基本数据类型的定义、使用方法与常用的其他程序设计语言，如 C/C++，Pascal 等比较类似，一般的 Java 程序设计教程都会对其进行细致的讲解，本文不再重复。请读者注意的是，Java 程序运行在 Java 虚拟机之上，与具体的计算机和操作系统无关，所以 Java 数据类型的位数是固定的，不会根据硬件平台以及操作系统的不同而不同。Java 基本数据类型的取值范围如表 1-1 所示。

表 1-1

类型	描 述	取值范围
boolean	布尔型	true 或 false
byte	8 位带符号整数	-128 ~ 127 之间的任意整数
char	字符型	2 字节，以 Unicode 编码表示
short	16 位带符号整数	-32768 ~ 32767 之间的任意整数
int	32 位带符号整数	-231 ~ 231 - 1 之间的任意整数
long	64 位带符号整数	-263 ~ 263 - 1 之间的任意整数
float	32 位单精度浮点数	根据 IEEE754 - 1985 标准
double	64 位双精度浮点数	根据 IEEE754 - 1985 标准

实际开发 Java 程序时经常需用到 String 类型的，本文将重点讲述如何正确地使用 String 类型，主要包括字符串初始化、比较、连接以及改变字符串中字母的大小写等操作。

2. Java 字符串类型（String）

Java 中字符串类型不是一种简单的类型，它也不是简单的字符数组（在C/C++ 中就是如此）。Java 中没有内置字符串类型，而是在标准 Java 库中包含一个名为 String 的预定义类。将字符串作为预定义的对象处理允许 Java 提供十分丰富的功能特性以方便处理字符串。

当创建一个 String 对象后，这个 String 对象是不能被改变的，初学者可能对这种情况感到有些意外。字符串不能改变是指一旦一个 String 对象被创建，将无法改变那些组成字符串的字符。每次需要改变字符串时都要创建一个新的 String 对象来保存新的内容，而原始的字符串不变。下面说明 String 类的一些用法。

（1）字符串初始化。字符串的初始化主要有两种方法：

```
String s = "Hello World!";
String s = new String("Hello World");
```

另外还支持由字符数组生成字符串的初始化方法，如下所示：

```
char c[] = {'a', 'b', 'c'};
String s = new String(c); // s = 'abc';
```

（2）字符串长度。字符串的长度是指其包含的字符的个数。调用 length()方法可以得到这个值。

下面的程序段输出 3，因为在字符串 s 中有 3 个字符。

```
char c[] = {'a', 'b', 'c'};
String s = new String(c);
System. out. println(s. length());
```

（3）字符串连接。“+”运算符用来连接两或多个字符串，产生一个新的 String 对象。下面的程序段将 3 个字符串连接起来：

```
String age = "22";
String s = "She is" + age + " years old. ";
System. out. println(s);
```

输出为字符串“She is 22 years old. ”。

字符串连接的一个实际应用是当创建一个很长的字符串时，可以将它拆开，使用 + 号将它们连接起来，避免源代码中长字符串的换行。下面是在数据库程序设计中的一个例子：

```
String longSql = "SELECT CustID, CustName, CustAddress" +
    "FROM Customer" +
    "WHERE Age < 30" +
    "AND Sex = 'male'";
System. out. println(longSql);
```

字符串也可以和其他类型的数据进行连接，生成新的字符串。例如：

```
int age = 22;
String s = "She is" + age + " years old. ";
System. out. println(s);
```

在这里，age 是一个整型（int）而不是字符串（String）型值，但是程序的输出与前面的例子相同。这是因为编译器将 age 的整型（int）值自动转换为它的字符串（String）形式。然后这个字符串就和前面一样被连接。

应当小心的是，在将其他类型的操作与字符串连接表达式混合时，有可能得到意想不到的结果。

请看下面的例子：

```
String s = "four: " + 2 + 2;
System. out. println(s);
```

程序显示：

```
four: 22
```

而不是

```
four: 4
```

运算符的优先级造成了首先是 four 和 2 的连接，这个连接的结果再和下一个 2 进行连接。若要先实现整数相加，可以使用括号。

```
String s = “four: ” + (2 + 2);
```

现在 s 就包含了字符串“four: 4”。

（4）字符截取。在一个 String 对象中构成字符串的字符不能像字符数组一样被索引，但 String 类提供了许多从 String 对象中截取字符的方法。这些方法利用下标对字符串进行操作，并且和数组一样，字符串下标从 0 开始。下面举两个例子进行说明。

为了从一个字符串（String）中截取一个字符，可以通过 charAt()方法直接引用单个字符。其一般形式如下：

```
char charAt(int where)
```

这里，where 是想要得到的字符的下标。where 的值必须是非负的，它指定了在字符串中的位置。charAt()方法返回指定位置的字符。例如：

```
char ch;
ch = "abc". charAt(1);
```

执行完这段代码后，ch 的值就是 'b'。

如果想将字符串（String）对象中的字符转换为一个字符数组，最简单的方法就是调用 toCharArray() 方法。对应整个字符串，它返回一个字符数组。其一般形式为：

```
char[] toCharArray()
```

例如：

```
String str = "abc";
char c[] = new char[3];
c = str. toCharArray();
for(int i =0; i <3; i + +)
 System. out. print(c[i] +", ");
```

程序代码输出 a, b, c。

（5）字符串比较。对 String 的比较有两种情况，一个是使用 = =，另一个是使用 equals()方法。= =是对 String 对象的地址进行比较，而 String 中的 equals()方法实现对 String 对象的内容的比较。所以 String s1 = new String ("hello") ; String s2 = new String ("hello")，对 s1 和 s2 进行上述比较时，前者应该返回 false，因为使用 new 生成的是两个不同的对象；后者应该返回 true，因为它们的内容是一样的，都是 "hello" 。那么如果还有一个 String s3 = "hello "；它和 s1 的比较应该是什么样子的呢？答案是 s1 = =s3 为 false，equals 的比较为 true。事实上 String 类是维持着一个 String 池的，这个池初始化为空的，当 String x = "hello "的时候，hello 就会被放入这个池中，当再次 String y = "hello "的时候，它首先去检查池中是否存在一个和 hello 内容一样的对象，如果存在的话就会把这个引用返回给 y，如果不存在的话，就会新创建一个 "hello" 并放入到池中，这样就实现了复用。看下面的例子：

```
public class StringTest {
   public static void main(String[] args) {
      String s1 = "hello";
      String s2 = new String("hello");
      String s3 = new String("hello");
      System. out. println("s1 = s2 is" + (s1 = =s2));
      System. out. println("s2 = s3 is" + (s2 = =s3));
      System. out. println("s1. equals(s2) is" + s1. equals(s2));
      System. out. println("s2. equals(s3)is" + s2. equals(s3));
   }
}
```

输出结果为：

```
s1 = s2 is false
s2 = s3 is false
s1. equals(s2) is true
s2. equals(s3) is true
```

（6）修改字符串。前面已经介绍了字符串（String）对象是不可改变的，每当想修改一个字符串（String）时，就需要使用下面字符串方法中的一种，这些方法都将构造一个完成修改的字符串的拷贝。

replace()方法用另一个字符代替调用字符串中一个字符的所有具体值，它具有如下的一般形式：

```
String replace(char original, char replacement)
```

这里 original 指定它所代替的字符返回得到的字符串。例如：

```
String s = "Hello". replace('l', 'w');
```

将字符串"Hewwo"赋给 s。

trim()方法返回一个调用字符串的拷贝，该字符串是将位于调用字符串前面和后面的空白字符删除

后的剩余部分。它的一般形式如下：

String trim()

这里是一个例子：

String s = " Hello World ". trim();

将字符串"Hello World"赋给 s。

trim()方法在处理用户输入的命令时，是十分有用的。例如，下面的程序提示用户输入一个客户的姓名后显示该客户的年龄。程序中使用 trim()方法删除在用户输入期间，不经意间输入的前缀或后缀空白符。

```
public class testTrim {
    public static void main(String args[]) {
        BufferedReader br = new
        BufferedReader(new InputStreamReader(System. in));
        String str;
        System. out. println("Enter 'stop' to quit. ");
        System. out. println("Enter Name. ");
        do {
            str = br. readLine();
            str = str. trim();
            if(str. equals("Tom"))
                System. out. println("Tom is 10 years old. ");
        }while(! str. equals("stop");
    }
}
```

二、Java 基础应用编程实例——Java 资源管理器制作

现在类似资源管理器制作的方法很多，可用 VB，Delphi 等语言编写，但用 Java 编写的资源管理器很少见，究其原因，是 Java 语言的特殊性（比其他语言中的对话框更复杂）是入门学习提高的一个好实例，如图 1－1 上、下所示的分别是用 VB 与 Java 编写的资源管理器源程序。

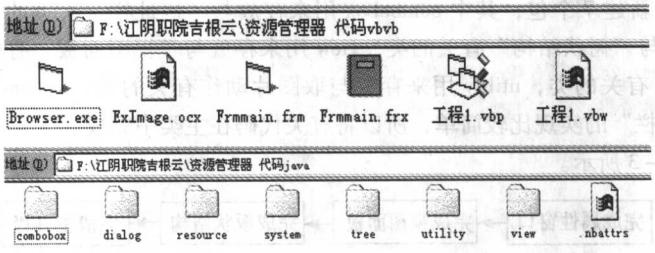


图 1－1

1. 资源管理器的简介

“资源管理器”是对计算机中数据进行管理的工具，可对数据进行浏览、排序等操作，每个图形界面的操作系统都有“资源管理器”这样的工具，其中最广为人知的资源管理器是视窗操作系统中的。本“资源管理器”就是模仿视窗操作系统中的“资源管理器”所做，只是一个雏形，许多功能还有待完善，如查找功能等还无法实现。

2. Java 简介及软件运行环境

Java 是 SUN 公司的一种编程工具语言，它的一个显著优点是运行时环境提供了平台独立性：可以在 Windows、Soloris、Linux 或其他操作系统上使用完全一样的代码，这点对于在各种不同平台上运行

从互联网上下载的程序来说非常有必要。另一个优点是 Java 具有和C + + 类似的语法，这一点使得 C 或 C + + 程序员可以很容易地学习 Java。Java 也是完全面向对象的，这点要强于 C + + 。除此以外，Java 还有许多显而易见的优点，比如，它的简单、分布式、健壮性、安全中立体系、可移值性等。

要运行本程序，需要下载 SUN 公司的 Java 虚拟机 JDK1. 4. 1（j2sdk - 1_4_1_02. exe, 为 36MB），很占内存，效果好的还有 Windows 98 下的 Java 虚拟机 86 版 msjavx86. rar，小巧精干，为 5MB。本书附赠的源代码把文件分成了四个压缩包，请下载后解压到同一个文件夹中即可安装！当然如要调试还需要下载 forte[tm]4. 0 编译程序集，为 32MB。

3. 制作过程

做一个软件，首先要对它的结构十分清楚。一个“资源管理器”由“菜单栏”、“工具栏”、“地址栏”、“树状结构”和“视图面板”组成，如图 1 - 2 所示。

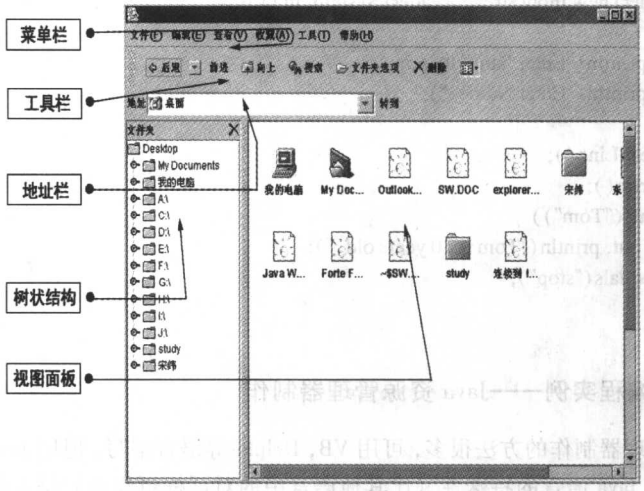


图 1 - 2

根据以上分析，新建 7 个包，其中 combobox 用来存放与“地址栏”有关的类，resource 用来存放图片，tree 用来存放与“树状结构”有关的类，view 用来存放与“视图面板”有关的类，dialog 用来存放与“属性对话框”有关的类，utility 用来存放与取图片动作有关的类，system 用来存放主类，由于“菜单栏”和“工具栏”的实现比较简单，所以将有关代码在主类中完成。

设计流程如图 1 - 3 所示。

图 1 - 3

如图 1 - 4 所示的是开发视图面板的步骤。

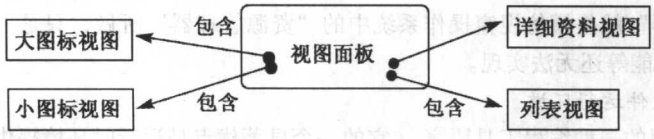


图 1 - 4

视图面板是资源管理器的重中之重，它包含 4 个视图：大图标视图（BigIconView）、详细资料视图（DetailView）、小图标视图（smallIconView）和列表视图（ListView）。

DetailView 继承 JTable，把文件名、时间、大小作为列，文件的各种属性作为行。把每个图标和文件名看作是 BUTTON，然后改写 BUTTON 的 UI，利用绘制器把图标和文件名绘制到视图面板上。SmallIconView, BigIconView, listView 都大同小异，renderer 都一样，只是绘制的时候注意排列方式不同。renderer 是这样的意思，传给它一个东西，它给你一个你所需要的组件，为了以后更换视图，还应添加相应的代码。

但同时问题又出来了，要取很多图片，那是非常烦琐的。为了取图片方便，就写一个这样一个接口，给每个取图片的动作都起名，这样每当要用到该图片时，只要调用该图片动作的名字就可以了。该接口起名为 Constants，放在 utility 包中，以下是实现的部分代码，都是静态方法。

```
public interface Constants {
    public static Icon SMOLLFILE = Utility. getIcon("Unknown. gif");
    public static Icon BIGFILE = Utility. getIcon("Unknown_P. gif");
    public static Icon BIGCOMPUTER = Utility. getIcon("computer2. jpg");
    //public static Icon BIGMYDOCUMENT = Utility. getIcon("MyDocument. gif");
    public static Icon deletIcon = Utility. getIcon("Delet. gif");
    public static Icon previousIcon = Utility. getIcon("Previous. gif");
    public static Icon upleverIcon = Utility. getIcon("Uplever. gif");
    public static Icon searchIcon = Utility. getIcon("SearchWeb. gif");
    public static Icon viewIcon = Utility. getIcon("view. gif");
    public static Icon detailIcon = Utility. getIcon("Details. gif");
    public static Icon largeIcon = Utility. getIcon("Large. gif");
    public static Icon listIcon = Utility. getIcon("List. gif");
    public static Icon viewsIcon = Utility. getIcon("Views. gif");
}
```

同时再写一个类用来实现取得图片的方法和取图片时的判断（比如判断是用“文件夹”图片还是“文件”图片）。该类命名为 Utility，存放在 utility 包中；以下是部分代码：

```
public class Utility {
    private static final String IMAGE_PATH = "explorer/resource/";
    /**
    * 获得图标的静态方法
    * @param name 图标的名字,不需要告诉绝对路径名
    */
    public static ImageIcon getIcon(String name) {
        ImageIcon icon = null;
        if(name. indexOf(IMAGE_PATH) != -1) {
            if(url != null)
            {
                icon = new ImageIcon(ClassLoader. getSystemResource(name));
            }
        }
        else {
            System. out. println("path" + (IMAGE_PATH + name));
            System. out. println("url" + ClassLoader. getSystemResource(IMAGE_PATH + name));
            System. out. println("** URL: " + ClassLoader. getSystemResource(IMAGE_PATH + name));
            java. net. URL url = ClassLoader. getSystemResource(IMAGE_PATH + name);
            if (url! =null)
            {
                icon = new ImageIcon(ClassLoader. getSystemResource(IMAGE_PATH + name));
```

```
            }
        }
        return icon;
    }
    public static boolean isComputerNode(File file) {
        return file.toString().equalsIgnoreCase("我的电脑") ||
        file.getName().equalsIgnoreCase("MyComputer");
    }
    public static boolean isDesktop(File file) {
        return file.getName().equalsIgnoreCase("桌面") ||
        file.getName().equalsIgnoreCase("desktop");
    }
    public static boolean isMyDocument(File file) {
        return file.getName().equalsIgnoreCase("我的文档") ||
        file.getName().equalsIgnoreCase("My Documents");
    }
```

接下来构造“树状结构”、“工具栏”、“文件夹选项”、“视图”、“菜单栏”和“地址栏”。

“树状结构”继承 JTree，改写 Jtree 的 render，把文件夹作为树枝，文件作为树叶。“工具栏”有不少监听器，有“后退”，“前进”，“文件夹选项”，“视图”。后退按钮和前进按钮是这样处理的，建立一个 arraylist，每打开一个文件，就把该文件名加到该 list 中，需要的时候再取出来，当然要加一些判断。“文件夹选项”比较简单，就是 setVisible 等一些类，“视图”就是在各个视图间切换，“菜单栏”和“地址栏”比较简单，在这里就不再列出了。

属性对话框可以显示文件的多种属性。继承 Jdialog，各种属性直接有方法可以取到。

最后，在主类中，把这些东西按绝对布局加进去，同时要注意添加监听器，让“地址栏”、“视图面板”和“树状结构”连动起来，到此，本次设计基本完成。

（邱　鹏　吉根云　蒋　蕾）

实例 2　Java 中的正则表达式及其应用

正则表达式（Regular Expression），相信所有玩过 UNIX/Linux 的朋友都了解或使用过它。不同版本的 UNIX/Linux 的一些命令使用了正则表达式，包括 ed、sed、awk、grep、文本编辑器 vi 及 emacs 等。此外，很多编译器也用正则表达式来验证源程序的语法是否正确；很多的编程语言如 Perl、PHP、Python、JavaScript 和 VBScript，也都支持用正则表达式处理文本（字符串）；一些文本编辑器如 EditPlus 也用正则表达式实现高级“搜索 - 替换”等功能。

由此可见，正则表达式已经超出了某种语言或某个系统的局限，成为操作系统或应用开发中一个不可缺少的功能。

但令人遗憾的是，SUN 在其发布的 Java 编程语言中，直到 1.4 版才正式支持正则表达式。在此之前，如果要想使用正则表达式，则只能使用第三方开发的支持正则表达式的包，如隶属于 Apache.org 下的开放源代码的 Jakarta - ORO 库。那么如何在 1.4 版中使用由 SUN 官方提供的正则表达式的相关包呢？

本文先简要地介绍正则表达式的基础知识，然后以 1.4 版中正则表达式的相关 API 为基础，并给出一定的实例，讲述如何使用正则表达式。

一、实例

正则表达式在 Java 中的使用方法有两种。一种是直接使用 String 类中的 matches、replaceAll 和 split 等方法；另一种是引进 java. util. regex 包，下面将分别举例说明。

1. String 类中的相关方法

（1）字符串校验。这里，用一个简单的 Java 应用程序来说明如何利用正则表达式进行数据校验。RegexStringTest. java 展示了如何对中文、英文及数字字符进行匹配的用法。RegexStringTest. java 提供一个图形界面便于用户输入数据，然后再利用正则表达式对用户输入的数据进行校验。

RegexStringTest. java 源程序清单：

```
1: import java. awt. *;
2: import java. awt. event. *;
3: import javax. swing. *;
4:
5: public class RegexStringTest extends JFrame {
6:  private JTextField phoneTextField, zipTextField, addressTextField,
7:  firstTextField, lastTextField, chineseTextField;
8: private Font songTi = new Font("宋体", Font. PLAIN, 12);
9:
10:  public RegexStringTest() {
11:  super("基于字符串的正则表达式");
12:  //创建图形界面
13:  JLabel phoneLabel = new JLabel("电话");
14:  phoneLabel. setFont(songTi);
15:  JLabel zipLabel = new JLabel("邮政编码");
16:  zipLabel. setFont(songTi);
17:  JLabel addressLabel = new JLabel("通信地址");
18:  addressLabel. setFont(songTi);
19:  JLabel firstLabel = new JLabel("First Name: (英文, 第一个字母必须大写)");
20:  firstLabel. setFont(songTi);
21:  JLabel lastLabel = new JLabel("Last Name: (英文, 第一个字母必须大写)");
22:  lastLabel. setFont(songTi);
23:  JLabel chineseLabel = new JLabel("中文:");
24:  chineseLabel. setFont(songTi);
25:  JButton okButton = new JButton("验证");
26:  okButton. setFont(songTi);
27:  okButton. addActionListener(new ActionListener() {
28:  //内部类开始
29:  public void actionPerformed(ActionEvent event) {
30:  validateDate();
31:  }
32:  }//内部类结束
33:
34:  ); //调用 addActionListener 结束
35:
36:  phoneTextField = new JTextField(15);
37:  zipTextField = new JTextField(6);
38:  addressTextField = new JTextField(35);
39:  firstTextField = new JTextField(20);
40:  lastTextField = new JTextField(20);
41:  chineseTextField = new JTextField(30);
```