

挑战 Java 程序员系列丛书

免费售后培训

免费电子课件及案例源代码

大量论坛技术支持

# Java SE

## 应用程序设计

*BitService*

北京比特塞威斯图书服务事业部 组编

王养廷 编著



机械工业出版社  
CHINA MACHINE PRESS

挑战 Java 程序员系列丛书

# Java SE 应用程序设计

北京比特塞威斯图书服务事业部 组编

王养廷 编著



机械工业出版社

本书从应用的角度介绍了 Java SE 的主要技术、开发工具和实战项目。与传统图书写法的区别在于：本书强调实际应用，从软件开发的角来组织内容，将知识点融入到每一个具体的实例中，注重提高读者的程序设计能力。

本书强调实例教学，着重介绍如何应用 Java SE 开发软件项目。可以作为学习 Java SE 的教材或参考书。

## 图书在版编目 (CIP) 数据

Java SE 应用程序设计/北京比特塞威斯图书服务事业部组编. —北京：机械工业出版社，2007.7

(挑战 Java 程序员系列丛书)

ISBN 978-7-111-21787-9

I. J… II. 北… III. Java 语言—程序设计 IV. TP312  
中国版本图书馆 CIP 数据核字 (2007) 第 096713 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑：李利健

责任印制：杨 曦

三河市国英印务有限公司印刷

2007 年 9 月第 1 版 · 第 1 次印刷

184mm × 260mm · 22.5 印张 · 554 千字

0001—5000 册

标准书号：ISBN 978-7-111-21787-9

定价：34.00 元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换  
销售服务热线电话：(010) 68326294

购书热线电话：(010) 88379639 88379641 88379643

编辑热线电话：(010) 88379739

封面无防伪标均为盗版

## 丛书序

在信息技术高速发展的今天,计算机技术已经广泛地应用于各行各业。软件行业作为计算机技术领域的核心产业得到了蓬勃的发展,而软件开发技术作为计算机及其相关行业的关键技术,越来越受到重视。当前,软件开发基本上有两种主流软件开发工具,即 Java 和 .NET。这两种开发工具功能强大、应用领域广泛,是其他开发工具所不能比拟的。可以说,熟练掌握这两种开发工具中的一种,再拥有一定程度的开发经验,就可以成为一名合格的软件开发人员。

目前,相对于大多数职位的人才都面临激烈竞争的情况,合格的软件开发人员却一直是供不应求的。这说明市场上对软件人才需求量很大。但由于软件开发需要丰富的实际工作经验,而对仅仅接受过课本教育,却很少开发实际项目的毕业生来说,是很难立刻胜任软件开发的工作岗位的,从而也就无法达到用人企业的要求,这就是所谓的“人才错位”。而能够有效弥补毕业生与合格的软件开发人员之间差距的途径,目前主要有两种:一是通过上专业培训班或者自己看书学习,并配合相关上机操作自学成才;二是通过用人单位提供实践的机会来获取能力。显然,对刚毕业的学生来说,第一种途径实现起来更容易。

虽然市场上有很多针对 Java 程序设计的参考书和教材,但是,已有的同类书都无法回避的一个事实就是:仅仅依靠纸介质图书和有限的源代码程序来学习 Java 程序设计,不但学习起来非常吃力,而且所得到的结果距离真正的 Java 程序员的需求还有一段距离。我们的结论就是:想真正掌握 Java 这样一种复杂的程序开发工具,除了要学习一套易学易用的图书,还需要看懂大量的实例源代码,以及专家随时的答疑解惑和面对面的学习指导,再加上充足的上机实践。只有这样,才能够真正达到或者接近一个合格的 Java 程序员的水平。

我们开发的这套“挑战 Java 程序员”系列丛书不但图书本身遵循了由浅入深、循序渐进的学习规律,能够使读者轻松快速地掌握 Java 开发的系统知识,而且还配备了完整的售后服务系统,包括:3 个学时的免费培训服务、免费从网上下载 PPT 电子课件、免费下载实例源代码、论坛、E-mail 技术支持服务等。通过这些全方位的售后服务,将 Java 程序员的必备技能依次讲授给读者,为读者铺就一条通向 Java 程序员的成功之路。

本套丛书十分适合于立志成为一个程序员的大中专院校相关专业的学生,或具有一定基础的软件开发爱好者。本套丛书共 6 个品种,包括《Java 面向对象程序设计》、《Java SE 应用程序设计》、《Java XML 应用程序设计》、《Java 数据库应用程序设计》、《Java Web 应用程序设计》和《Java EJB 应用程序设计》,如图 1 所示。

### 免费售后服务说明

我们为本套丛书的读者提供了丰富的免费售后服务,包括以下 4 方面。

1. 每本书免费赠送 3 个学时的培训(培训地点在北京)

“挑战 Java 程序员”系列丛书中的每本图书自带了 3 个学时的售后培训,购买该书的读者都可以参加。但需要提前预约,否则培训机构不能提供对应的培训内容。

每本图书的售后服务培训班通常根据图书销售情况,周期性推出新班,并通过网站

(<http://www.bits-service.com>) 发布。读者需要提前到网站查询相应班级, 然后通过网络或电话报名预约, 预约电话为 010-64446364 和 010-64446332。预约完的读者在参加培训前需要到网站上或通过电话确认对应班级的变更情况, 培训机构会取消一些人数不多的班, 并将取消信息显示在网站上, 我们将不再一一通知读者, 请读者朋友认真核实。图 2 显示了报名预约的基本流程。

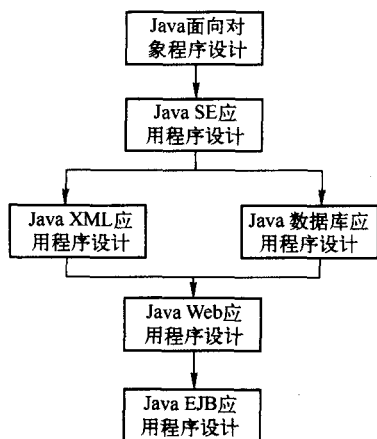


图 1

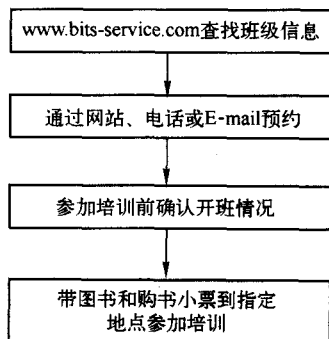


图 2

## 2. 免费提供精美的 PPT 电子课件

读者可以登录 [www.bits-service.com](http://www.bits-service.com) 下载各本图书的电子课件。

## 3. 免费提供实例源代码

要下载代码的读者, 请首先登录 <http://www.bits-service.com>, 在主页上选择“比特塞威斯图书服务中心 (BITS Book)”, 进入图书服务首页, 在首页菜单条上选择“读者服务”进入“读者服务”频道, 随后可以利用网站的搜索工具或使用标题列表找到对应图书, 在相应的图书网页上单击“源代码下载”超链接, 下载该图书的源代码压缩包。本系列丛书的源代码清单通常按照章节组织, 读者参考图书, 可以使用对应的源代码。

## 4. 论坛和 E-mail 技术支持

我们为本系列图书提供了专业服务论坛, 网址是 <http://www.bits-service.com> 主页→图书服务论坛→Java 系列图书, 读者在阅读过程中若发现问题, 可以到论坛里发帖子, 培训机构的专业技术老师会定期进行回复。论坛为每本书提供一个讨论区, 建议读者将问题发送到对应图书的讨论区中, 以便作者及时收集和处理。读者也可以发送 Email 给 [booksupport@bits-service.com](mailto:booksupport@bits-service.com) 寻求专家技术支持。

此外, 我们还会提供一部分特级教师的课堂视频教学文件, 读者可以通过网站下载。

## 增值服务

除了上述提供的免费服务外, 如果读者希望进一步学习, 我们还提供了一部分增值服务, 包括 Java 项目实习服务、Java 程序员就业推荐服务等。这部分服务需要收取一定费用, 读者如果有兴趣, 可以按照以下联系方式与比特塞威斯公司联系。

增值服务咨询邮箱: [allsupport@bits-service.com](mailto:allsupport@bits-service.com); 增值服务咨询电话: 010-64446368。

## 前 言

本书不是一本 Java 语言的入门教材,在学习本书之前,要求读者已经了解 Java 语言的基本知识,知道 Java 语言的基本语法和语句,初步理解面向对象的含义,能够编写简单的 Java 程序。

本书从应用的角度出发,采用案例讲解的方式,让读者了解 Java 语言标准版 Java SE 的主要应用领域;学习 Java SE 的开发技术;知道常用 Java 开发工具的使用;通过实例项目学习开发 Java 软件的过程。通过本书的学习,让读者能够学会如何应用 Java 语言来开发一个桌面应用软件。本书内容按照由浅入深、由具体到抽象的方法进行组织,通过一个个具体的实例来学习这些内容,提高读者的学习兴趣,鼓励读者自己动手模仿、编写和调试程序,通过实践来掌握程序设计的基本方法和规律,建立编程的自信心,提高自己的程序设计能力。全书共分 15 章,具体内容如下:

第 1~10 章为技术篇,介绍 Java SE 的主要技术。第 1 章介绍 Java SE 的基本概念和主要内容;第 2 章介绍 Java 桌面应用程序的设计方法;第 3 章介绍 Java SE 的事件处理模型;第 4 章介绍 Java SE 常用的布局管理器;第 5 章介绍 Swing 桌面应用程序的设计;第 6 章介绍 Java SE 的图形和图像处理;第 7 章介绍 Java 小应用程序 Applet 的机制和程序设计方法;第 8 章介绍 Java 的流模型和对象序列化;第 9、10 章介绍 Java 语言的一些新特性:Annotation、反射和范型。这部分内容从不同的方面来介绍 Java 语言的应用,每个问题都给出了 Java 程序代码,通过代码来讲解相关知识。

第 11~13 章为工具篇,介绍常用的 Java 开发工具。第 11 章介绍使用 JBuilder 开发桌面应用程序;第 12 章介绍使用 Eclipse 开发桌面应用程序;第 13 章介绍工具 Ant 和 JUnit。这部分内容介绍了 JBuilder 和 Eclipse 的安装、开发环境,应用程序的开发和调试过程;介绍了如何使用 JUnit 进行单元测试,使用 Ant 实现程序的自动构建。读者可以了解常用 Java 工具的主要特点和使用方法,为以后的项目开发打下基础。

第 14、15 章为项目实战篇,介绍项目的开发过程。第 14 章是 Java 程序编辑器;第 15 章介绍抢钱游戏的开发过程。这两个项目都从需求分析开始,获取用户的功能需求,进行界面设计;完成项目的构架设计和详细设计;给出实现程序的源代码;完成软件的测试、部署和项目总结。通过这两个项目的介绍,可以让读者了解开发一个 Java 软件的过程和方法,学习和积累项目开发经验。

为了让读者更好地了解 Java 新技术,本书介绍了 Java JDK 1.6,所有的实例都在 JDK 1.6 和相关的工具下调试并通过。本书可以作为高等院校学生学习 Java SE 的教材,也可以作为从事计算机工作的技术人员学习 Java SE 的参考书。

本书由王养廷老师编写,在编写过程中,比特塞威斯职业教育部和企业培训部的有关老师为本书大纲的制定和编写提供了很多宝贵意见,某些老师还参与了部分章节的整理和案例制作,在此表示感谢。

书中遗漏或错误之处,敬请读者批评指正。读者若对本书有什么疑问,可以发 E-Mail 到:jsjfw@mail.machineinfo.gov.cn,我们会尽快给予解决。

本书提供了一个专业的服务论坛，读者在阅读过程中若发现问题，也可以到论坛里发帖子，比特塞威斯的技术老师会定期进行回复。论坛的网址是：

<http://www.bits-service.com> 主页->图书服务论坛->Java 系列图书->Java SE 应用程序设计

北京比特塞威斯图书服务事业部

## 目 录

## 丛书序

## 前言

|                                  |    |                               |    |
|----------------------------------|----|-------------------------------|----|
| 第 1 章 Java SE 技术概述 .....         | 1  | 2.4.5 自定义事件测试代码 .....         | 37 |
| 1.1 Java SE 应用程序概述 .....         | 1  | 2.5 小结 .....                  | 38 |
| 1.1.1 Java SE 与 Java EE 技术 ..... | 1  | 第 3 章 AWT 与事件处理模式 .....       | 39 |
| 1.1.2 桌面应用程序简介 .....             | 2  | 3.1 Java 内部类 .....            | 39 |
| 1.1.3 Java 小应用程序简介 .....         | 3  | 3.2 内部类事件处理模式 .....           | 42 |
| 1.1.4 AWT 与 Swing .....          | 4  | 3.3 外部类事件处理模式 .....           | 45 |
| 1.2 Java SE 功能概述 .....           | 5  | 3.4 自身事件处理模式 .....            | 46 |
| 1.2.1 Java 事件处理模型 .....          | 5  | 3.5 小结 .....                  | 48 |
| 1.2.2 Java 图像处理 .....            | 5  | 第 4 章 AWT 容器的布局管理器 .....      | 49 |
| 1.2.3 Java 对象序列化 .....           | 6  | 4.1 FlowLayout 布局管理器 .....    | 49 |
| 1.2.4 Annotation 与反射 .....       | 7  | 4.2 GridLayout 布局管理器 .....    | 51 |
| 1.2.5 Java 泛型 .....              | 7  | 4.3 BorderLayout 布局管理器 .....  | 53 |
| 1.3 小结 .....                     | 7  | 4.4 CardLayout 布局管理器 .....    | 54 |
| 第 2 章 AWT 组件、容器与事件处理 .....       | 8  | 4.5 GridBagLayout 布局管理器 ..... | 56 |
| 2.1 面向对象窗口系统 .....               | 8  | 4.6 组件的绝对布局管理器 .....          | 59 |
| 2.1.1 AWT 窗口对象 .....             | 8  | 4.7 自定义布局管理器 .....            | 60 |
| 2.1.2 组件与容器 .....                | 11 | 4.8 小结 .....                  | 64 |
| 2.2 AWT 事件处理机制 .....             | 13 | 第 5 章 Swing 桌面应用程序设计 .....    | 65 |
| 2.2.1 AWT 事件处理模型 .....           | 14 | 5.1 Swing 概述 .....            | 65 |
| 2.2.2 AWT 中的事件类 .....            | 16 | 5.1.1 Swing 简介 .....          | 65 |
| 2.2.3 事件处理接口 .....               | 17 | 5.1.2 Swing 的布局管理 .....       | 67 |
| 2.2.4 事件适配器 .....                | 20 | 5.1.3 Swing 的事件处理 .....       | 68 |
| 2.3 常见事件类型处理 .....               | 22 | 5.2 Swing 的组件对象模型 .....       | 70 |
| 2.3.1 键盘事件处理 .....               | 22 | 5.2.1 Swing 的窗口类 .....        | 70 |
| 2.3.2 窗口事件处理 .....               | 24 | 5.2.2 Swing 的组件类 .....        | 71 |
| 2.3.3 鼠标事件处理 .....               | 27 | 5.3 常用的 Swing 窗口类 .....       | 72 |
| 2.3.4 菜单事件处理 .....               | 30 | 5.3.1 JPanel 面板 .....         | 72 |
| 2.4 自定义事件及其处理 .....              | 34 | 5.3.2 JFrame 面板 .....         | 73 |
| 2.4.1 自定义事件类 .....               | 34 | 5.4 常用的 Swing 组件 .....        | 75 |
| 2.4.2 自定义事件处理接口 .....            | 35 | 5.4.1 标签 .....                | 75 |
| 2.4.3 自定义事件处理类 .....             | 35 | 5.4.2 文本输入和编辑组件 .....         | 76 |
| 2.4.4 自定义事件产生者 .....             | 35 | 5.4.3 选择组件 .....              | 83 |

|        |                           |     |       |                         |     |
|--------|---------------------------|-----|-------|-------------------------|-----|
| 5.4.4  | 列表框和下拉列表框                 | 88  | 7.3.2 | 图形与图像                   | 159 |
| 5.4.5  | 图标和边框                     | 94  | 7.3.3 | URL 与资源的获取              | 160 |
| 5.4.6  | 表格                        | 96  | 7.3.4 | 声音与动画                   | 161 |
| 5.4.7  | 树                         | 108 | 7.3.5 | 小应用程序的封装                | 164 |
| 5.4.8  | 菜单                        | 112 | 7.4   | Applet 对数据的操作和<br>安全性管理 | 165 |
| 5.4.9  | 对话框                       | 118 | 7.4.1 | 小应用程序的数据操作<br>原则        | 165 |
| 5.4.10 | 自定义 Swing 组件              | 120 | 7.4.2 | 小应用程序的安全性<br>管理         | 165 |
| 5.4.11 | 滚动条处理                     | 122 | 7.5   | 小结                      | 166 |
| 5.5    | 小结                        | 124 | 第 8 章 | 流的高级特性与对象<br>序列化        | 167 |
| 第 6 章  | Java SE 图形图像处理            | 125 | 8.1   | 流的完整模型                  | 167 |
| 6.1    | 图形图像处理概述                  | 125 | 8.1.1 | 节点流和处理流                 | 167 |
| 6.2    | 位图图像处理                    | 126 | 8.1.2 | 流的对象模型                  | 170 |
| 6.2.1  | 图像文件编码格式介绍                | 126 | 8.1.3 | 标准输入和标准输出               | 172 |
| 6.2.2  | 图像文件装入                    | 126 | 8.1.4 | 字节流文件                   | 174 |
| 6.2.3  | 图像类                       | 128 | 8.1.5 | 字符流文件                   | 180 |
| 6.2.4  | 图像的处理                     | 129 | 8.1.6 | 内存流                     | 183 |
| 6.3    | 矢量图形处理                    | 133 | 8.1.7 | 缓冲区流                    | 185 |
| 6.3.1  | 矢量图形绘图简介                  | 133 | 8.1.8 | 管道                      | 187 |
| 6.3.2  | 常用的绘图方法                   | 134 | 8.1.9 | 目录和文件操作                 | 190 |
| 6.3.3  | 矢量图形文件的保存                 | 136 | 8.2   | Java 的对象序列化             | 193 |
| 6.3.4  | 矢量图形文件的处理                 | 138 | 8.2.1 | 对象序列化的基本概念              | 193 |
| 6.4    | 非图形类图形处理方法                | 139 | 8.2.2 | Java 对象序列化              | 193 |
| 6.4.1  | 文本与字体                     | 139 | 8.2.3 | 对象序列化的基本原则              | 195 |
| 6.4.2  | 颜色与线形                     | 140 | 8.3   | Java 对 ZIP 流的处理         | 196 |
| 6.4.3  | 形状与填充                     | 143 | 8.3.1 | ZIP 文件的表示格式             | 196 |
| 6.5    | 小结                        | 146 | 8.3.2 | ZIP 文件的读写               | 196 |
| 第 7 章  | Java Applet 小应用程序         | 147 | 8.3.3 | ZIP 文件的完整性              | 200 |
| 7.1    | Applet 概述                 | 147 | 8.4   | Java 对 JAR 流的处理         | 202 |
| 7.1.1  | Applet 的基本概念              | 147 | 8.4.1 | JAR 文件的表示<br>格式         | 202 |
| 7.1.2  | Applet 的生命周期              | 149 | 8.4.2 | JAR 文件的读写               | 203 |
| 7.1.3  | Applet 的设计原则              | 150 | 8.4.3 | JAR 文件的完整性              | 205 |
| 7.1.4  | Applet 应用程序和桌面<br>应用程序的区别 | 151 | 8.5   | 小结                      | 205 |
| 7.2    | Applet 基本机制               | 152 | 第 9 章 | Annotation 与反射机制        | 206 |
| 7.2.1  | Applet 事件处理               | 152 | 9.1   | Java 反射机制概述             | 206 |
| 7.2.2  | Applet 和网页的数据交互           | 154 |       |                         |     |
| 7.2.3  | Applet 的网络交互              | 155 |       |                         |     |
| 7.3    | Applet 程序设计               | 157 |       |                         |     |
| 7.3.1  | 窗口的弹出与处理                  | 157 |       |                         |     |

|        |                                 |     |        |                                 |     |
|--------|---------------------------------|-----|--------|---------------------------------|-----|
| 9.2    | Java 反射模型 .....                 | 210 | 10.4.2 | 泛型和面向对象之间的<br>性能比较 .....        | 264 |
| 9.3    | 使用反射机制进行类<br>方法调用 .....         | 213 | 10.5   | 泛型的应用 .....                     | 265 |
| 9.3.1  | 调用构造方法实例化<br>类对象 .....          | 214 | 10.5.1 | 使用泛型描述类的成员<br>变量 .....          | 266 |
| 9.3.2  | 通过反射机制查询方法<br>接口信息 .....        | 218 | 10.5.2 | 使用系统中标准的<br>泛型类 .....           | 267 |
| 9.3.3  | 调用类的实例方法 .....                  | 220 | 10.6   | 小结 .....                        | 268 |
| 9.3.4  | 调用类中的静态方法 .....                 | 221 | 第 11 章 | 使用 JBuilder 开发桌面<br>应用程序 .....  | 269 |
| 9.4    | 使用反射机制访问成员<br>变量 .....          | 223 | 11.1   | JBuilder 概述 .....               | 269 |
| 9.4.1  | 使用反射机制访问实例<br>成员变量 .....        | 223 | 11.1.1 | JBuilder 简介 .....               | 269 |
| 9.4.2  | 使用反射机制访问静态<br>成员变量 .....        | 226 | 11.1.2 | Jbuilder 的安装 .....              | 269 |
| 9.4.3  | 使用反射机制构造和访问<br>数组成员变量 .....     | 227 | 11.1.3 | JBuilder 应用浏览器 .....            | 273 |
| 9.5    | 基于配置文件的对象构造 .....               | 229 | 11.1.4 | 开发环境的配置 .....                   | 276 |
| 9.6    | Java Annotation .....           | 233 | 11.2   | JBuilder 开发桌面应用<br>程序 .....     | 278 |
| 9.6.1  | 标准的 Annotation .....            | 234 | 11.2.1 | JBuilder 开发程序的<br>方法 .....      | 278 |
| 9.6.2  | 自定义 Annotation 标注 .....         | 237 | 11.2.2 | JBuilder 中的调试与<br>跟踪 .....      | 283 |
| 9.7    | 使用反射机制分析<br>Annotation .....    | 241 | 11.3   | 小结 .....                        | 285 |
| 9.8    | Annotation 标注和反射的<br>综合应用 ..... | 243 | 第 12 章 | 使用 Eclipse 开发桌面<br>应用程序 .....   | 286 |
| 9.8.1  | 类启动器的设计 .....                   | 244 | 12.1   | Eclipse 概述 .....                | 286 |
| 9.8.2  | 通过标注定义方法中的<br>次序 .....          | 247 | 12.1.1 | Eclipse 简介 .....                | 286 |
| 9.9    | 小结 .....                        | 249 | 12.1.2 | Eclipse 的安装 .....               | 287 |
| 第 10 章 | Java 与泛型 .....                  | 250 | 12.1.3 | Eclipse 开发环境 .....              | 288 |
| 10.1   | 泛型概述 .....                      | 250 | 12.2   | Eclipse 开发桌面应用<br>程序 .....      | 289 |
| 10.2   | 泛型类的定义和使用 .....                 | 252 | 12.2.1 | Eclipse 开发程序的<br>方法 .....       | 289 |
| 10.2.1 | 标识符的声明时限制 .....                 | 253 | 12.2.2 | 界面设计器的安装 .....                  | 292 |
| 10.2.2 | 使用通配符进行泛型<br>引用时限制 .....        | 254 | 12.2.3 | 设计桌面应用程序 .....                  | 294 |
| 10.2.3 | 限定通配符的范围 .....                  | 256 | 12.2.4 | 程序的调试与跟踪 .....                  | 299 |
| 10.3   | 泛型方法的定义和使用 .....                | 257 | 12.3   | 小结 .....                        | 301 |
| 10.4   | 泛型和面向对象的比较 .....                | 258 | 第 13 章 | 使用 Ant+JUnit 开发桌面<br>应用程序 ..... | 302 |
| 10.4.1 | 在代码复用上的差异 .....                 | 259 | 13.1   | Ant 概述 .....                    | 302 |

|                                   |                     |     |                            |                |     |
|-----------------------------------|---------------------|-----|----------------------------|----------------|-----|
| 13.1.1                            | Ant 简介 .....        | 302 | 14.2.1                     | 项目的构架 .....    | 323 |
| 13.1.2                            | Ant 的安装运行 .....     | 303 | 14.2.2                     | 项目的详细设计 .....  | 323 |
| 13.1.3                            | 编写 build.xml .....  | 303 | 14.2.3                     | 项目的实现 .....    | 324 |
| 13.1.4                            | 构建文件的结构 .....       | 305 | 14.2.4                     | 项目的测试和部署 ..... | 329 |
| 13.1.5                            | Ant 的 Task .....    | 307 | 14.2.5                     | 项目总结 .....     | 330 |
| 13.2                              | JUnit 概述 .....      | 311 | 14.3                       | 小结 .....       | 330 |
| 13.2.1                            | JUnit 的基本测试模型 ..... | 311 | <b>第 15 章 编制抢钱游戏 .....</b> | <b>331</b>     |     |
| 13.2.2                            | 编写 JUnit 测试代码 ..... | 312 | 15.1                       | 抢钱游戏简介 .....   | 331 |
| 13.2.3                            | 测试类的编写原则 .....      | 315 | 15.1.1                     | 项目背景 .....     | 331 |
| 13.3                              | 增量式桌面应用程序           |     | 15.1.2                     | 游戏的图形描述 .....  | 332 |
|                                   | 开发 .....            | 316 | 15.1.3                     | 项目的界面设计 .....  | 332 |
| 13.3.1                            | 增量式开发概述 .....       | 316 | 15.2                       | 游戏的开发 .....    | 333 |
| 13.3.2                            | 增量式开发方法 .....       | 316 | 15.2.1                     | 项目的构架 .....    | 333 |
| 13.4                              | 小结 .....            | 319 | 15.2.2                     | 项目的详细设计 .....  | 333 |
| <b>第 14 章 编制 Java 程序编辑器 .....</b> | <b>320</b>          |     | 15.2.3                     | 项目的实现 .....    | 335 |
| 14.1                              | 项目分析 .....          | 320 | 15.2.4                     | 项目的测试和部署 ..... | 341 |
| 14.1.1                            | 项目背景 .....          | 320 | 15.2.5                     | 项目总结 .....     | 342 |
| 14.1.2                            | Java 程序开发环境 .....   | 321 | 15.3                       | 小结 .....       | 342 |
| 14.1.3                            | 项目的 UI 设计 .....     | 321 | <b>附录 软件调试环境 .....</b>     | <b>343</b>     |     |
| 14.1.4                            | 项目的功能说明 .....       | 322 | <b>参考文献 .....</b>          | <b>345</b>     |     |
| 14.2                              | 项目的实现 .....         | 323 |                            |                |     |

# 第 1 章 Java SE 技术概述

J2SE (Java 2 Standard Edition) 称为 Java 2 标准版。1999 年, SUN 公司将 Java 2 平台分为三块: J2SE、J2EE 和 J2ME。J2SE 主要定位于桌面应用程序开发。2004 年底, SUN 公司推出了 J2SE 1.5, 并统一更名为 Java SE 5。目前的最高版本为 Java SE6。本章将概要介绍 Java SE 技术以及 Java SE 的应用, 重点讨论如下内容:

- Java SE 简介
- 桌面应用程序简介
- 事件处理模型简介
- 图像处理简介
- 流和对象序列化
- Annotation、反射和泛型

## ▶▶ 1.1 Java SE 应用程序概述

### ▶▶▶ 1.1.1 Java SE 与 Java EE 技术

随着 Java 技术的不断发展, 它根据市场进一步细分为三个版本: 针对企业应用的 Java EE (Java Enterprise Edition)、针对普通 PC 应用的 Java SE (Java Standard Edition) 和针对嵌入式设备及消费类电器的 Java ME (Java Micro Edition)。

Java EE 应用系统要求开发者将工作分成两类: 商业逻辑和表示逻辑, 其余由系统资源自动处理。这样, 开发人员就能将更多的时间用在商业逻辑和表示逻辑的开发上。Java EE 平台是运行 Java EE 应用的环境, 由 Java EE 部署规范、IETF 标准集、JavaBean 组件模型和 CORBA 标准组成。开发人员利用 JavaBean 组件模型定义 Java 类实例, 通过已经定义的事件访问 Java 类。Java EE 通过定义一组标准结构实现它的优势, 例如:

- Java EE 应用模型是一种用于开发多层次、瘦客户用户程序的设计模型;
- Java EE 平台是一个标准平台, 用来整合 Java EE 应用程序;
- Java EE 测试套件是一套兼容测试组件, 用来检测产品是否同 Java EE 平台兼容。

Java EE 平台是 Java 技术企业级应用的最佳平台, 它可以让程序员迅速、快捷地开发和部署企业级应用程序。

Java 标准版是一个 Java 开发和部署平台, 提供编写桌面、工作站应用程序所有必需的

功能。Java SE 的主要特点如下：

- 具有良好的计算性能和缩放能力，Java SE 是一个快速开发和配置跨平台的应用与服务综合性平台。
- 能提供良好的图形界面功能，图形应用程序的界面主要是基于 AWT 和 Swing 的。
- Java SE 是一个具有综合性能的平台，可以方便地把一个应用配置到多种平台上。
- Java SE 的各个版本保持与以前的版本全面兼容，因此，保证了在不同平台、系统和环境间的互操作性。
- Java SE 支持 XML 技术和一整套安全机制，可以为各种标准的应用程序提供可交互操作的服务。

Java 之所以能得到快速的发展，主要是因为互联网的迅速发展。Java 可以和浏览器结合在一起，因而在桌面系统中也得到了广泛应用。

### 1.1.2 桌面应用程序简介

目前已经存在大量的非常成功的 Java 桌面程序。

Java 提供了两种界面设计包：AWT (Abstract Window Toolkit) 包和 Swing 包。AWT 是一种最基本的界面设计类包，包含很多基本的窗口界面组件，例如 Button、TextBox 等；Swing 是一种轻量级组件包，其组件丰富，能够独立于本地窗口，界面友好，在不同的系统上界面一致。它是 AWT 包的部分替代品。下面给出一个使用 AWT 组件的例子，本例中只有一个按钮，程序代码如下：

程序编号：0101

文件名称：chapter01\buttonDemo.java

```
/*001*/ import java.awt.*;
/*002*/ import java.awt.event.*;
/*003*/ public class buttonDemo
/*004*/ {
/*005*/     public static void main( String args[])
/*006*/     {
/*007*/         Frame f = new Frame("桌面应用程序示例");
/*008*/         Button button1 = new Button("打开");
/*009*/         f.add(button1);
/*010*/         f.setVisible( true);
/*011*/         f.addWindowListener(new WindowAdapter(){
/*012*/             public void windowClosing(WindowEvent e){
/*013*/                 System.exit(0);
/*014*/             }
/*015*/         });
/*016*/     }
/*017*/ }
```

编译运行这个程序后，在窗口中显示一个按钮，结果如图 1-1 所示。

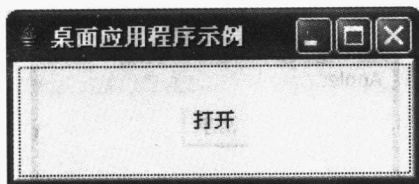


图 1-1 程序 0101 运行结果

程序 0101 是一个最简单的桌面应用程序，单击标题栏上的“关闭”按钮，结束应用程序的运行。在实际的桌面应用程序设计中，需要考虑窗口设计、组件选择、组件布局、组件事件处理程序设计等内容。我们将在第 2 章中详细介绍基于 AWT 的桌面应用程序设计。

### 1.1.3 Java 小应用程序简介

小应用程序 Applet 是 Java 的另一个重要的应用。小应用程序可以内嵌在 HTML 网页中，使网页具有动画、声音和图像等多媒体展示效果。同时，它还可以与用户进行动态交互，接受用户输入，根据不同的输入做出不同的反应。随着 Web 技术的发展，它已经被广泛地应用到网站开发中。下面给出一个例子来演示小应用程序如何工作，该例中也只有一个按钮，程序代码如下：

程序编号：0102

文件名称：chapter01\appletDemo.java

```
/*001*/ import java.applet.*;
/*002*/ import java.awt.*;
/*004*/ public class appletDemo extends Applet{
/*005*/     public void init(){
/*006*/         Button play = new Button("演示");
/*007*/         add(play);
/*008*/     }
/*009*/ }
```

编译这个程序后得到 class 文件，但现在还不能运行这个程序。下面还要设计一个 HTML 文件，把这个 class 文件嵌入进去。HTML 文件的代码如下：

程序编号：0103

文件名称：chapter01\appletDemo.html

```
/*001*/ <html>
/*002*/ <body>
/*003*/     <applet code=appletDemo.class width=120 height=120>
/*004*/ </applet>
/*005*/ </body>
/*006*/ </html>
```

使用小应用程序查看器 appletViewer 运行这个程序，结果如图 1-2 所示。

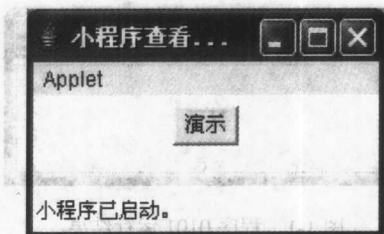


图 1-2 程序 0103 运行结果

这个程序可以被浏览器打开，从服务器端下载到客户端来执行。与桌面应用程序一样，小应用程序同样支持图形用户界面，可以使用各种图形组件支持事件处理和组件的布局管理。有关小应用程序的设计将在第 7 章详细介绍。

### 1.1.4 AWT 与 Swing

在应用程序设计中，用户界面是非常重要的。现代程序设计语言几乎都支持图形用户界面，Java SE 也不例外。在 Java 的早期版本中提供了抽象窗口工具包 AWT，使用它可以编写基于图形用户界面的 Java 程序。由于 AWT 产生的图形接口不够理想，同时还有很多限制，Java SE 扩展了 JFC，增加了一个被称为“Swing”的图形用户接口包。它是 AWT 的扩展，提供了更强大、更灵活的组件集合，主要有：按钮、标签、复选框、表格、弹出式菜单、剪贴板等。在程序 0101 中演示了使用 AWT 组件设计应用程序。下面使用 Swing 来完成这个程序，程序代码如下：

程序编号：0104

文件名称：chapter01\buttonDemo.java

```
/*001*/ import javax.swing.*;
/*002*/ import java.awt.event.*;
/*003*/ public class buttonDemo
/*004*/ {
/*005*/     public static void main( String args[])
/*006*/     {
/*007*/         JFrame f = new JFrame("桌面应用程序示例");
/*008*/         JButton button1 = new JButton("打开");
/*009*/         f.add(button1);
/*010*/         f.setVisible( true);
/*011*/         f.addWindowListener(new WindowAdapter(){
/*012*/             public void windowClosing(WindowEvent e){
/*013*/                 System.exit(0);
/*014*/             }
/*015*/         });
/*016*/     }
/*017*/ }
```

编译运行这个程序后，在窗口中显示一个按钮，结果如图 1-3 所示。

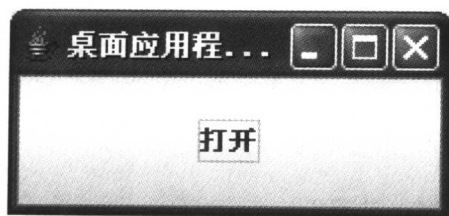


图 1-3 程序 0104 的运行结果

从程序 0104 的代码可以看出，该程序和程序 0101 基本相同，不同之处在于所用的组件全部换成 Swing 组件。Swing 程序设计将在第 5 章介绍。

## 1.2 Java SE 功能概述

本书主要从以下几个方面介绍 Java SE: AWT 桌面应用程序设计、AWT 事件处理模型、布局管理器的使用、Swing 程序设计、图形和图像处理、小应用程序、对象序列化、Annotation 和反射机制以及 Java 泛型。

### 1.2.1 Java 事件处理模型

Java 桌面应用程序提供了事件处理机制。桌面应用程序的界面是由组件构成的，每个组件是一个独立的单元。这里把用户或系统发送给每个组件的命令或信息称为事件，所有组件都使用相同的模式来处理接受和处理事件。例如，程序 0101 中的程序段：

```
/*011*/      f.addWindowListener(new WindowAdapter(){
/*012*/          public void windowClosing(WindowEvent e){
/*013*/              System.exit(0);
/*014*/          }
/*015*/      });
```

给窗口组件 Frame 对象 f 添加一个事件监听器，这个事件监听器负责接收窗口关闭事件，当收到这个事件后，关闭窗口 Frame。有关事件处理的详细内容在第 2、3 章中介绍。

### 1.2.2 Java 图像处理

Java SE 提供了图形和图像处理功能，包括图像的装入、缩放、旋转等，还包括对图像位图进行逐点处理。下面的例子是演示如何装载一个位图。程序代码如下：

```
程序编号：0105
文件名称：chapter01\loadPicDemo.java

/*001*/ import javax.swing.*;
/*002*/ import java.awt.*;
/*003*/ import java.awt.event.*;
/*004*/ public class loadPicDemo extends JFrame
```

```

/*005*/ {
/*006*/     public static void main( String args[])
/*007*/     {
/*008*/         new loadPicDemo();
/*009*/     }
/*010*/     public void loadPicDemo( )
/*011*/     {
/*012*/         setSize( 400,400);
/*013*/         setVisible( true);
/*014*/         addWindowListener(new WindowAdapter(){
/*015*/             public void windowClosing(WindowEvent e){
/*016*/                 System.exit(0);
/*017*/             }
/*018*/         });
/*019*/     }
/*020*/     public void paint(Graphics g){
/*021*/         Image img = Toolkit.getDefaultToolkit().
                getImage ("lily.jpg");
/*022*/         g.drawImage(img,100,100,this);
/*023*/     }
/*024*/ }

```

编译运行这个程序后，在窗口中显示一幅图像，结果如图 1-4 所示。



图 1-4 程序 0105 运行结果

上面这个例子装载了一幅图像，接下来就可以对它进行处理。有关图形和图像处理的内容，将在第 6 章介绍。

### 1.2.3 Java 对象序列化

本书第 8 章介绍 Java SE 的输入和输出，Java 的输入和输出使用了流概念，把输入视作从外部设备向计算机的流入，称为输入流；把输出视作从计算机向设备的流出，称为输出流。在 Java SE 中提供了流模型来完成流的操作，流模型包括大量与流相关的类。这些类根据处理方式可以分成字节流类和字符流类；根据与设备的对应关系分成节点流和处理流。在 Java SE 中有两类特殊的流，即 ZIP 流和 JAR 流，这两个流非常类似，都用来完成数据的打包和压缩功能。