

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

面向对象程序设计

BitService

北京比特塞威斯图书服务事业部 组编

栗菊民 编著

 机械工业出版社
CHINA MACHINE PRESS



挑战 Java 程序员系列丛书

Java 面向对象程序设计

北京比特塞威斯图书服务事业部 组编

栗菊民 编著



机械工业出版社

本书详细讲解了 Java 面向对象程序设计的知识。主要内容包括 Java 面向对象基本概念、Java 基本语法、类的构造和设计、抽象类与接口、类的高级特性、Java 中常用的类等。面向对象是本书的重点，在对面向对象知识进行了深入讨论后，作为面向对象思想的应用又介绍了 Java IO 程序设计、多线程、Socket 程序设计、数组和常用的数据结构，最后讲解了使用 Java 技术开发软件系统的方法。

本书内容安排由浅入深，通过丰富的教学案例全面讲解了面向对象的应用技术，非常适合初学者。书中的程序不需要特殊运行条件和编程环境，只需安装 JDK 1.6 和记事本软件。

本书适用于 Java 面向对象程序设计的初学者，也可作为各类 Java 培训班的教材。

图书在版编目 (CIP) 数据

Java 面向对象程序设计 / 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 978-7-111-21846-3

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 102887 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: 孙 业

责任印制: 杨 曦

三河市宏达印刷有限公司印刷

2007 年 9 月第 1 版·第 1 次印刷

184mm×260mm·19.25 印张·474 千字

0001—5000 册

标准书号: ISBN 978-7-111-21846-3

定价: 29.00 元

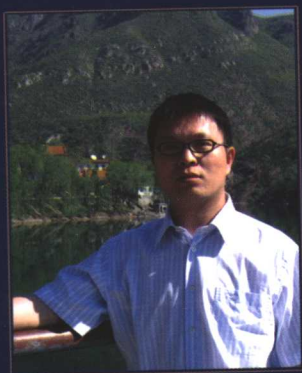
凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

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



栗松涛

毕业于清华大学自动化系。先后担任过 Intel 公司项目经理、新东方软件培训部主任等职。2004 年加盟北京数据服务公司从事软件架构研究。目前担任北京比特塞威斯图书服务事业部经理，《挑战 Java 程序员系列丛书》的主编；兼任北京数据服务公司软件反向工程副主任，负责多个大型软件反向工程项目。

著有多本软件架构类专业图书

《Java 服务器架构设计与 Tomcat 源代码详解》

《MVC 架构设计与 Struts 源代码详解》

《面向接口架构设计与 Spring 源代码详解》

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。

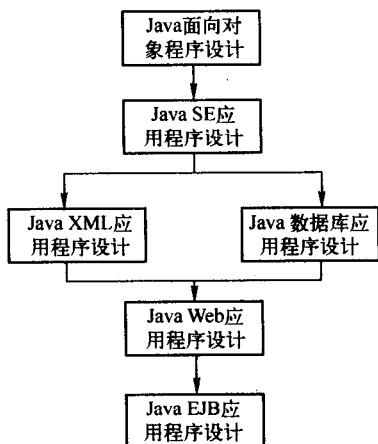


图 1

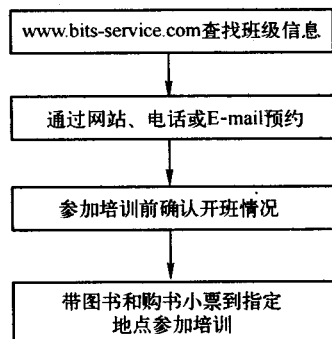


图 2

2. 免费提供精美的 PPT 电子课件

读者可以登录 www.bits-service.com 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 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；增值服务咨询电话：010-64446368。

前 言

Java 是由 Sun 公司推出的一种面向对象程序开发语言，目前在软件行业有着近一半的市场占有率，是主流程程序开发技术之一。从 1996 年 Sun 发布 Java 的第一个版本以来，Java 以其跨平台特性和其成熟的面向对象特性在全世界迅速普及。目前 Java 的最新版本是 2005 年发布的 Java SE 6（传统上的 Java JDK 1.6），本书就是基于该平台进行创作的。Java 语言具有良好的平台兼容性，本书几乎所有例子都可以在 Java JDK 1.4 及其以后平台上成功运行。

Java 是一种十分实用的程序设计语言，无论读者是否决心投身软件行业，熟练掌握一门程序开发语言，无疑对自身发展都是十分有益的。但 Java 技术领域广泛，新工具、新产品层出不穷，如何在千变万化的表象中梳理出 Java 技术的本质，则是本书的着力点，也是本书的特点。所有 Java 技术有两个基本特点：Java 语法特征和 Java 面向对象，离开这些讨论 Java 的其他技术没有任何意义。本书可以作为初学 Java 技术读者的第一本读物，也可以作为已初步掌握 Java 技术的读者加深自己对 Java 理解的工具。本书一共分成以下几个部分：

第一部分：Java 技术特点概述，包括第 1 章。该章对 Java 相关性的概念进行了概括讨论，有利于初学者，特别是没有接触过 Java 的读者，使他们对 Java 技术有一个整体概念。已经初步了解 Java 技术的读者可以略过该章，直接进入后面章节学习。

第二部分：Java 基本语法，包括第 3~6 章。该部分是在第 2 章对 Java 面向对象概念进行初步讨论后，才开始讨论的。原因就在于 Java 是一种纯面向对象程序设计语言，没有任何面向对象概念，就无法讨论任何 Java 技术，包括最简单的程序。

第三部分：Java 面向对象，包括第 2 章和第 7~11 章。此部分重点讲解 Java 面向对象技术。面向对象通常是刚开始接触 Java 技术的读者朋友，碰到的第一个难题，很多已经使用一段时间 Java 的读者，对面向对象的本质也是不十分清楚的。本部分的特点是使用十分通俗的语言，向读者讲解 Java 面向对象原理，该部分是本书的重点。

第四部分：Java 面向对象应用，包括第 12~19 章。重点讨论了 Java 中的数组、数据结构、常用类型、文件读写、线程、异常处理、socket 读写等。该部分形式上在讨论具体的程序设计技术，本质上却是使读者在有了初步面向对象概念后，有一个应用所学概念的场所。通过这部分的学习，加深对面向对象知识的理解。

第五部分：使用 Java 面向对象进行项目开发，包括第 20 章。该章通过一个多线程聊天室的例子，向读者展现使用本书所讨论的 Java 知识进行 Java 项目开发的基本过程。使读者对 Java 技术的应用有一个整体上的认识。

本书由栗菊民老师负责整理编写，在编写过程中比特塞威斯职业教育部和企业培训部的所有老师都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，这里不再一一列出，并由衷的表示感谢。作者在编写该书过程中和很多北京



资深的 Java 职业讲师进行过讨论，这里对他们表示感谢。

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，我们会尽快给予解答。

本书提供了一个专业的服务论坛，读者在阅读过程中发现问题或疑难，可以到论坛里发帖子，比特塞威斯的专业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 面向对象程序设计

北京比特塞威斯图书服务事业部

总序	
前言	
第1章 Java 技术概述	1
1.1 Java 语言的发展历史	1
1.2 Java 平台的分类	2
1.2.1 Java SE 平台	2
1.2.2 Java EE 平台	3
1.2.3 Java ME 平台	3
1.3 Java 的技术特点	4
1.3.1 Java 的语言特性	4
1.3.2 Java 语言的运行机制	5
1.3.3 Java 的虚拟机与跨平台特性	6
1.3.4 Java 自动垃圾收集技术	6
1.4 Java 开发环境	7
1.4.1 Java 运行环境安装	8
1.4.2 Java 运行环境配置	9
1.5 第一个 Java 应用程序	11
1.6 小结	14
第2章 Java 面向对象基础	15
2.1 面向对象的基本思想	15
2.1.1 面向对象思想中的类和对象	15
2.1.2 面向对象的程序设计思想	17
2.1.3 Java 中的类和对象	18
2.1.4 引导类	21
2.2 面向对象的软件开发方法	21
2.3 面向对象设计举例	22
2.4 小结	25
第3章 数据类型和变量	26
3.1 Java 基本数据类型	26
3.1.1 整数类型	26
3.1.2 浮点类型	28
3.1.3 boolean 类型	29
3.1.4 char 类型	30
3.2 Java 中的引用类型	32
3.3 Java 中的局部变量	33
3.3.1 局部变量的初始化	34
3.3.2 局部变量的作用域	35
3.4 Java 中的成员变量	37
3.4.1 Java 中的静态成员变量	37
3.4.2 Java 中的实例变量	39
3.4.3 Java 成员变量的初始化	39
3.5 小结	41
第4章 Java 中的运算符	42
4.1 基本运算	42
4.1.1 算术运算	42
4.1.2 递增递减运算	43
4.1.3 扩展赋值运算	45
4.2 逻辑运算	45
4.3 位运算	47
4.4 字符串串接运算	48
4.5 浮点类型变量的运算	49
4.6 小结	50
第5章 Java 中的程序语句	51
5.1 分隔符与语句块 (block)	51
5.1.1 普通语句块	51
5.1.2 静态语句块	52
5.2 Java 中的标识符和关键字	54
5.3 循环语句	56
5.3.1 for 循环语句	56
5.3.2 while 循环语句	57
5.3.3 do-while 循环语句	58
5.4 条件语句	59
5.4.1 if-else 分支控制语句	59
5.4.2 if-elseif 分支控制语句	60
5.4.3 if 控制语句	61
5.4.4 问号表达式	62
5.5 switch 语句	62
5.6 break 和 continue 语句	64
5.6.1 循环语句中的 break 语句	64
5.6.2 循环语句中的 continue 语句	66

5.7	注释语句与 Javadoc	68	8.1.3	Java 文件书写原则	106
5.8	小结	72	8.2	Java 中的命名规则	107
第 6 章	Java 中类的设计方法	73	8.2.1	包与类的命名规则	107
6.1	对象状态刻画——类的信息		8.2.2	变量与常量的命名规则	108
	封装	73	8.2.3	方法的命名规则	109
6.1.1	Java 中的访问修饰符	73	8.3	Java 语句书写规范	110
6.1.2	类信息的封装和隐藏	75	8.3.1	普通语句书写规范	110
6.1.3	Java 类的属性	78	8.3.2	控制语句书写规范	111
6.2	对象状态初始化——类的构造		8.3.3	方法语句书写规范	113
	方法	79	8.4	小结	114
6.2.1	构造方法存在的原因	79	第 9 章	Java 中类的继承	115
6.2.2	构造方法的特征	81	9.1	继承的基本概念	115
6.3	对象行为的刻画——方法的		9.1.1	类与类之间的从属关系	115
	重载	82	9.1.2	代码的重用机制	117
6.3.1	普通方法的重载	82	9.1.3	继承的原则	119
6.3.2	构造方法的重载	84	9.2	父类对继承的限制	119
6.4	对象的生命周期——构造与		9.3	方法的重写	123
	销毁	86	9.3.1	方法重写的概念	123
6.4.1	Java 对象的构造过程	86	9.3.2	方法重写实现细节	126
6.4.2	Java 对象的垃圾回收	87	9.4	面向对象中的多态	128
6.5	类特征的刻画——静态成员	88	9.5	构造方法的继承规则	130
6.5.1	类状态的刻画——静态变量	88	9.5.1	默认构造方法调用	130
6.5.2	类状态的初始化——静态语句	90	9.5.2	显式构造方法调用	132
6.5.3	类功能的刻画——静态方法	91	9.5.3	继承中构造方法调用原则	133
6.6	小结	93	9.6	小结	135
第 7 章	Java 中类的组织 (包)	94	第 10 章	Java 中的接口	136
7.1	Java 类的文件意义	94	10.1	接口的概念	136
7.2	包的第一个例子	96	10.2	接口的继承	141
7.3	包类型编译语句	98	10.3	接口的使用原则	143
7.4	package 的打包与引用	98	10.4	小结	146
7.4.1	jar 命令打包	98	第 11 章	Java 中的高级类	148
7.4.2	打包 Java package	99	11.1	抽象类 (abstract)	148
7.4.3	调用打包后的 package	100	11.1.1	抽象类的基本概念	148
7.5	包的作用	103	11.1.2	抽象类的使用原则	151
7.6	小结	103	11.1.3	接口和抽象类的比较	152
第 8 章	Java 编码规范	104	11.2	final 类型的类、方法和	
8.1	Java 文件结构规范	104		变量	153
8.1.1	Java 文件的结构	104	11.2.1	类的继承性与 final 类	153
8.1.2	Java 程序模板	105	11.2.2	final 类型的方法	154

11.2.3 final 类型的变量	155	13.4 小结	195
11.3 Java 内部类	157	第 14 章 Java 数据结构	196
11.3.1 内部类的定义和使用	157	14.1 Java 中的集合类	196
11.3.2 静态内部类	158	14.1.1 集合 (Set) 的基本概念	196
11.3.3 局部内部类	159	14.1.2 集合的枚举	198
11.3.4 内部类的继承	160	14.1.3 foreach 循环语句	200
11.3.5 内部接口和接口中的 内部类	161	14.2 Java 链表 (List)	202
11.3.6 内部类的使用原则	164	14.2.1 链表的基本概念	202
11.4 Java 匿名类	164	14.2.2 链表类 Vector	204
11.5 小结	166	14.3 Java 栈	205
第 12 章 Java 中的常用类	167	14.4 Java Map	207
12.1 Java 体系中的 Object 类	167	14.4.1 Hashtable 的概念和使用	207
12.1.1 Object 类的概念	167	14.4.2 HashMap 的概念和使用	209
12.1.2 对象的比较	168	14.4.3 HashMap 和 Hashtable 的 比较	210
12.1.3 对象的 toString 方法	170	14.5 小结	211
12.2 String 类	172	第 15 章 Java 异常处理	212
12.2.1 字符串的基本操作	172	15.1 异常的基本概念	212
12.2.2 字符串的拆分与合并	173	15.1.1 Java 异常处理机制	213
12.2.3 字符串的对象特点	175	15.1.2 异常和错误的区别	214
12.3 StringBuffer 类	176	15.1.3 异常的分类	215
12.4 Math 类	177	15.2 异常的处理	217
12.5 Random 类	178	15.2.1 try-catch 异常处理	218
12.6 日期处理类	179	15.2.2 throws 异常处理	221
12.7 Java 体系中的封装类	180	15.2.3 异常的状态	222
12.8 小结	182	15.2.4 方法重写中的异常处理	223
第 13 章 Java 中的数组	183	15.3 自定义异常	225
13.1 数组的基本概念	183	15.4 异常的定义和处理原则	228
13.1.1 基本类型数组	183	15.5 小结	229
13.1.2 对象型数组	185	第 16 章 Java 输入与输出	230
13.1.3 一维数组的数组 (多维 数组)	187	16.1 数据输入输出基本模型	230
13.2 Java 数组的操作	189	16.2 数据输入输出基本机制	231
13.2.1 数组初始化	189	16.3 文本文件的输入和输出	233
13.2.2 数组的枚举	191	16.3.1 文本文件读取	233
13.2.3 数组的复制	192	16.3.2 文本文件输出	236
13.2.4 识别数组类型	193	16.4 命令行输入和输出	237
13.2.5 Arrays 类	193	16.5 二进制文件输入和输出	238
13.3 Java 数组使用原则	195	16.6 小结	240
		第 17 章 Java 线程	241

17.1 线程的概念	241	18.5 断言使用原则	272
17.1.1 线程的基本模型	242	18.6 小结	274
17.1.2 Thread 类创建线程	243	第 19 章 Java Socket 程序设计	275
17.2 线程状态管理	245	19.1 Socket 基本模型	275
17.2.1 线程的终止	246	19.2 Socket 连接的建立	277
17.2.2 通过优先级属性更改线程执行 状态	247	19.3 通过 Socket 进行数据通信	280
17.2.3 通过 sleep 和 interrupt 控制线程 状态	249	19.4 小结	283
17.3 线程的同步	251	第 20 章 多线程聊天室系统实战	284
17.3.1 线程 Join 同步	252	20.1 系统需求概要和项目背景	284
17.3.2 线程 Synchronized 同步	253	20.1.1 项目范围	284
17.4 wait-notify 方法协调线程	257	20.1.2 项目开发计划	285
17.5 面包作坊的多线程实现	259	20.2 WinChat1.0 系统的设计与 实现	285
17.6 小结	263	20.2.1 系统基本结构	285
第 18 章 Java 断言	264	20.2.2 MyChat 通信协议	286
18.1 Assertion 的基本概念和 用法	264	20.2.3 MyChat 的系统结构与 实现	287
18.2 Assertion 命令使用	266	20.2.4 MyChat 系统结构优化	293
18.3 继承和 Assertion 断言	268	20.3 小结	297
18.4 断言和异常的区别	270	附录 软件调试环境	298

第

1

章

Java 技术概述

Java 是软件开发领域的主流开发语言，也是一套完整的 IT 解决方案。它包括两个主要方面：首先是 Java 面向对象编程语言本身，其次是和 Java 语言相关的服务器平台技术。前者侧重于将 Java 理解为一种高级的面向对象编程语言，和 C# 或 C++ 等语言类似；后者则侧重于 Java 虚拟机和相关的服务器技术。任何 Java 程序在所有 Java 虚拟机上都能正常运行，和具体的操作系统平台没有太大关系。本章重点包括以下内容。

- Java 语言的发展历史
- Java 平台分类
- Java 技术特点
- Java 运行环境的安装和配置

1.1 Java 语言的发展历史

Java 语言是 1995 年诞生的，它的前身是 Oak 语言。Java 语言从其第一个版本开始就是和 Internet 密切绑定的一种语言。关于它的名字的由来，则有一个十分有趣的传说。据说有一天 Java 语言开发成员组的会员正在一个咖啡馆中边喝咖啡边讨论如何给新开发的编程语言起一个名字，激烈的讨论没有产生任何的结论，最后一个人端着一杯热的 Java（爪哇）咖啡无奈的提议，干脆就叫 Java 语言吧。于是这个提议得到了大家一致的同意，Java 语言应运而生，一杯热咖啡也成了 Java 技术的形象代表，图 1-1 展示了 Java 的形象图片。



图 1-1 一杯令人提神的咖啡（Java 的图标）

1996 年，Sun 公司发布了 JDK 1.0。这是一个相对成形的 Java 版本，一经推出就展现了巨大的技术潜力，为各大操作系统提供商所看中。世界上知名的计算机公司，如 IBM, Apple, DEC, Adobe, Silicon Graphics, HP, Oracle, Toshiba 和 Microsoft 等，都在自己的产品中提供了对 Java 技术的支持。

1997 年，Sun 公司发布了 JDK 1.1，这是一个使用十分广阔的软件包，据统计，该软件一共被下载了 200 万次。该版本为 Java 应用普及起到了至关重要的作用。

1998 年，Sun 公司发布了 JDK 1.2，该软件包又被称为 Java 2，大名鼎鼎的 Java 2 技术

就是由此开始的。在随后的一年中，Sun 公司又根据使用背景的不同将 Java 分成 3 个版本：J2SE、J2EE 和 J2ME。其中 J2SE 就是新版本的 JDK，它为其他 Java 程序的开发和运行提供了最基本的技术支持；而 J2EE 则重点定位于企业级别的分布式软件系统；J2ME 定位于嵌入式系统的开发和运行，例如手机和微波炉设备中。

在随后的若干年发展中，Java 程序开发技术得到了飞速的普及，目前有几亿部支持 Java 程序的手机；5.5 亿台桌面电脑上运行 Java 程序，75% 的开发人员将 Java 作为首选的程序设计语言。2004 年 Sun 推出了 Java 的一个里程碑式的版本 J2SE 1.5，并从此将其更名为 J2SE 5.0，以此来显示该版本的重要意义。

2006 年，JavaOne 大会召开，对 Java 技术进行了统一的重新命名，并推出 Java SE 6。在这种统一的命名方案中，J2SE 被更名为 Java SE；J2EE 被更名为 Java EE；J2ME 被更名为 Java ME；本书就是针对 Java SE 6 开发的，反映了最新的 Java 面向对象特性。

▷▷ 1.2 Java 平台的分类

Java 技术根据其使用背景特征被分成 3 个平台：Java SE (Java Platform, Standard Edition)、Java EE (Java Platform, Enterprise Edition) 和 Java ME (Java Platform, Micro Edition)。这 3 个平台将程序开发人员、应用服务平台提供商和设备提供商分成了多个领域，每个领域都有自己的特点，但这三个平台不是相互独立的，不同的平台之间有着相互联系。

▷▷▷ 1.2.1 Java SE 平台

Java SE 平台在以前版本中称为 J2SE，它是整个 Java 平台中最基础的一个平台，是 Java EE 平台的基础。Java SE 平台提供了编写桌面、服务器应用程序所有必要的功能，很多服务器和应用软件都是用 Java SE 开发的，例如 Eclipse、JBuilder、Tomcat 等著名软件。Java SE 软件是快速开发、部署关键任务的首选解决方案，在软件行业得到了广泛应用。目前最新的版本是 Java SE 1.6 (又称为 Java SE6，本书以后不再区分)，本书讨论的所有特点都是基于该平台的。Java SE 的主要特点包括如下方面：

- 1) 提供良好的图形界面功能：图形应用程序界面主要是基于 AWT 和基于 Swing 的应用程序。图形功能保证了应用程序界面在不同操作系统平台上显示的一致性。
- 2) Java SE 是一个综合性平台，能使一个完整的、可缩放的高性能应用系统配置在任何种类的平台，实现 Java 应用程序的跨平台特性。
- 3) 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能力，本书尽管是在 Java SE 1.6 平台上编写的，但很多内容是 Java SE 的基本内容，在 Java SE 的其他版本上也同样能够运行。
- 4) Java SE 支持 XML 技术和一整套安全服务，因此为基于各类开放标准的可互操作应用与服务的创建奠定了基础，特别是为 Web Service 的支持与实现提供了固有的便利性。
- 5) Java SE 是一个完整的开发环境。开发商可基于这一环境创建功能丰富的交互式应用。这些应用可以配置在任何平台上。
- 6) Java 之所以能够发展得很快，得益于互联网的快速发展。Java 可以和浏览器结合在一起，因而很自然地在桌面系统得到广泛应用，这也是 Java SE 程序一个十分成功的应用领

域。此外,Java 应用程序、Java 图形界面应用程序等都在软件行业中占据着十分重要的位置。

Java SE 平台是本书讨论的重点,本书主要在该平台上讨论 Java 的面向对象程序设计思想。至于 Java SE 平台的开发技能将在本系列丛书的《Java SE 应用程序设计》一书中进行讨论,感兴趣的读者可以参考相应图书。

1.2.2 Java EE 平台

Java EE 以前称为 J2EE,是为 Java 企业级应用提供的一个十分成功的平台。作为企业级程序有什么特点呢?首先是软件的高负载性,同时在线人数很多;其次是分布性和事务性。这些都对 Java EE 平台提出了很高的要求,也是 Java EE 平台十分成功的地方。下面分别从几个方面讨论 Java EE 平台的特点。

首先是 Java EE 平台的分布式计算技术。该技术使企业能通过机械性的增加计算机的数量而提高系统的运行性能,而不是通过升级或购买更高档次的计算机硬件设备来实现。在这种思想中,程序逻辑根据其实现的不同被封装到不同组件中,J2EE 应用程序根据不同组件的业务特点将其部署到不同的计算机上,而不同计算机上的组件在整体上又相互协作,从而实现 Java EE 程序的分布性。Java EE 程序中支持的组件种类很多,下面根据其特点分成不同的类进行讨论:

1) Applet 组件,Applet 是一种嵌入到浏览器中的 Java 组件,通常又将该组件称为 Applet 应用程序,这种程序使浏览器中的网页具备了丰富多彩的界面交互功能,是目前十分流行的 Java 组件封装形式。

2) JavaBean 组件,是 Java 技术中最基本的一种组件。JavaBeans 组件通常封装了实例变量,为访问这些变量提供了十分友好的 get 方法和 set 方法,特别是在网页中访问。这种用途的 JavaBeans 组件在设计 and 执行时很简单,但必须遵守 JavaBeans 组件体系结构的相应要求。

3) EJB 组件,这是一种功能十分强大的组件,该组件将一个复杂的功能分成不同的任务,分派给了服务器开发商、服务器管理员和组件开发人员。这种分工留给组件开发人员的工作量十分少,因而可以通过很少的代码实现强大的功能,这种组件目前在软件行业中的使用十分普遍。

4) Web 组件,这种组件主要包括 Servlet 组件和 JSP 页面组件。它们都是编写十分简单的组件,可以实现功能强大的交互式网站,本套丛书中的《Java Web 应用程序设计》,主要讨论该方面的内容,感兴趣的读者可以参考该书。

整体上,Java EE 程序是目前 Java 企业应用程序中最主要的存在形式,几乎所有使用 Java 的软件企业都要用到这种技术。

1.2.3 Java ME 平台

Java ME 平台主要用来开发在微型设备中运行的应用程序,例如手机,微波炉,PDA 等。目前,手机在中国的使用率很高,并且大多带有手机游戏,这些程序大多是由 Java ME 开发的。

如何使用 Java ME 平台为手机开发程序呢?为手机开发的程序通常是在 PC 机上完成的,然后通过 PC 机上的一个软件——某款手机的模拟器(用来在计算机上模拟一个手机的

软件)来测试所开发的软件,如果软件没有问题,则通过红外线或数据线软件传输到手机等设备上,通过和计算机相似的安装过程,就可以实现将 PC 计算机上开发的手机应用程序安装到手机设备上。

Sun 公司将 Java ME 平台定义为“一种以广泛的消费性产品为目标的高度优化的 Java 运行环境,包括寻呼机、移动电话、可视电话、数字机顶盒和汽车导航系统”。图 1-2 展示了各种设备。



图 1-2 各式各样的微型设备

不同数字平台之间的差异性要远远高于不同计算机平台间的差异性,而 Java ME 平台的出现为所有的硬件平台提供了统一的程序控制接口,读者可以通过 Java 写出一段程序,该程序在所有的硬件平台上都能运行,从而为程序员提供了一种“万能的手”。

▶▶ 1.3 Java 的技术特点

Java 是一套完整的 IT 行业解决方案,包含很多技术。从整体上划分,可以分成 Java 语言和 Java 系统平台,后者包括 Java 虚拟机和各种 Java 应用服务器,例如 Tomcat, JBoss 等。本节首先讨论 Java 的语言特性和执行机制,然后讨论 Java 虚拟机的原理和 Java 的内存管理机制,至于 Java 服务器平台的相关特性则在本套丛书的《Java Web 应用程序设计》中进行讨论。

▶▶▶ 1.3.1 Java 的语言特性

Java 语言是一种纯面向对象的语言,和 C++语言有很大的区别。比较而言,C++既支持面向对象的语言特性也支持和 C 兼容的过程化语言特性,而 Java 在语法特征方面仅支持面向对象特性,没有对过程化程序设计进行任何支持。Java 和目前的 C#以及 VB.NET 等语言相比,有很大的类似性和相通性,掌握其中任何一种语言后都能很容易地掌握其他语言。但从实现形式上讲,Java 则显得更加简洁而完善,也更容易被用户接受。

从 Java 语言的创作根源上讲, Java 继承于 C 和 C++ 语言, 但删除了 C 和 C++ 中影响程序健壮性的部分, 如指针、内存申请和释放、无条件转移语句等, 使 Java 语言显得更简洁而完整。这种做法使 Java 成为一种更容易学习的语言, 无论新手还是老手, 都能写出健壮而稳定的代码。

1.3.2 Java 语言的运行机制

在分析程序设计语言的运行机制时, 通常会讨论两种类型的程序执行模式: 编译型的程序语言和解释性的程序语言。语言的编译过程类似于日常生活中的翻译过程——翻译人员将一篇文章从中文翻译成英文, 在过程上有批次处理和离线完成的特点, C 语言是这类语言中的典型代表。解释性语言则类似于国际会议上的“中英同传”——在发言者讲出中文的同时, 对方的耳机中就有了对应的英文发言, 此时的翻译过程就称为“同传”, 在过程上有在线和及时的特点, BASIC 语言是此类中的典型代表。

Java 语言是一种编译解释性语言, 同时具备编译特性和解释特性。作为编译性语言, Java 程序要被统一编译成字节码文件——文件的后缀是“.class”, 此种文件又被称为 Java 类文件。Java 类文件不能够在计算机上直接运行, 它需要被 Java 虚拟机翻译成本地的机器码后才能执行, 而 Java 虚拟机的翻译过程则是解释性的。Java 字节码文件首先被加载到计算机内存中, 然后读取一条指令, 翻译一条指令, 执行一条指令, 该过程被称为 Java 语言的解释执行过程, 是由 Java 虚拟机完成的, 图 1-3 演示了这种过程。

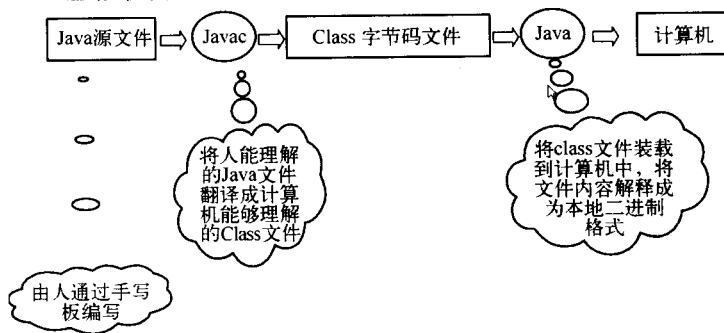


图 1-3 Java 程序的执行过程

如图 1-3 所示, Java 开发工具 JDK 提供了两个很重要的命令来完成上面的翻译和解释过程。两个命令分别是 `java.exe` 和 `javac.exe`, 前者负责加载 Java 类文件, 并逐句的对字节码进行解释并执行, 对应了 Java 语言的解释阶段; 后者则负责将 Java 文件翻译成为类文件, 对应了 Java 语言的编译过程。在次序上, Java 语言要先经过编译的过程, 才能进行解释执行过程, 如图 1-3 从左向右所示的次序。为了详细刻画 Java 程序在计算机中的执行过程, 给出了图 1-4 来清晰地解释 Java 程序的执行过程。

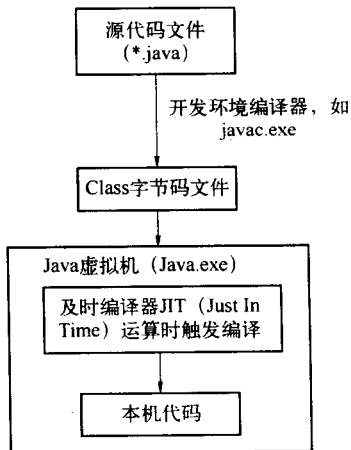


图 1-4 Java 程序执行机制示意图