

1CD-ROM



Java

游戏编程导学

陆光义 宋现锋 编著



清华大学出版社

Java 游戏编程导学

陆光义 宋现锋 编著

清华大学出版社
北 京

内 容 提 要

本书通过编写趣味游戏程序来引导读者学习 Java 编程的方法和技巧,形式新颖活泼,别具一格。

全书从 Java 语言基础知识和编制简单的程序入手,将 Java 编程的知识点有机地分散在“幸运 52”,“速算 24”,“俄罗斯方块”等多个趣味游戏的程序设计示例中,使得 Java 语言中类、对象、属性、方法、接口、继承等抽象概念变得具体形象,通俗易懂;并引导读者掌握 Java 中数组、字符串、事件处理、异常处理、图形和多媒体应用、Swing 组件和网络等知识的运用和技巧。

本书以示例教学方式组织内容,集趣味性、直观性、可操作性于一体,适合于 Java 初学者及对游戏程序感兴趣的电脑爱好者阅读参考。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Java 游戏编程导学/陆光义,宋现锋编著.

—北京:清华大学出版社,2004

ISBN 7-302-09776-3

I. J… II. ①陆… ②宋… III. JAVA 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 108635 号

出 版 者:清华大学出版社

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

社 总 机:010-62770175

组稿编辑:科海

文稿编辑:洪英

封面设计:林陶

版式设计:科海

印 刷 者:北京市耀华印刷有限公司

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:23.5 字数:572 千字

版 次:2004 年 11 月第 1 版 2004 年 11 月第 1 次印刷

书 号:ISBN 7-302-09776-3/TP·6750

印 数:1~5000

定 价:35.00 元(1CD)

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

本书如存在文字不清、漏印以及缺页、倒角、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:82896445

前言

对于一个软件开发人员来说，当前发展最快的主流计算机语言就是由Sun公司开发的Java语言。Java语言，因为它的简单易用、平台无关性和健壮性，在短短10年时间之内，很快就发展成为当前最流行的编程语言。同时，为了适应企业应用和嵌入式系统开发，Sun公司推出了相应的J2EE和J2ME开发工具包和响应规范。目前，它们已经成为各自相关领域内的主流产品。在当今的网络时代，学习Java，掌握Java编程，对程序员来说不再是可有可无，Java已经成为开发人员必备的工具了。

本书讲解Java语言的基本知识和应用，这同时也是J2EE和J2ME应用开发的基础。

本书的内容

本书首先介绍了Java基础知识，之后每章将介绍一个或多个精心制作的趣味游戏，它们各自侧重于应用Java语言的某些特性，循序渐进，详细讲述了Java语言，并给出大量示例和非常有价值的编程方法。

第1章介绍了Java特性和一些基本语法。在这一章，读者可以了解到Java的历史、现状及其发展前景，同时也将学会如何编写和运行一个简单的“Hello World” Java程序。

第2章先讲解Java面向对象编程的一些基本概念，然后通过“幸运52”模拟游戏的编写，让读者对Java的面向对象编程有更深入的理解。

第3章讲解了持有对象和异常处理的知识，同时编写了“球迷必答”和“速算24”游戏。在“球迷必答”游戏中，用户被询问5道问题，前一道问题必须答对，方可进入下一道问题。在“速算24”游戏中，用户随意抽出4张扑克牌，用加、减、乘、除的方法将它们连接起来，使得结果等于24。这一章通过这两个游戏来加深读者对数据结构、异常处理等知识的理解。

第4章通过“精彩闹钟”和“模拟钢琴”游戏，重点讲解Java在图形和多媒体方面的一些简单应用，AWT的组件编程，也涉及到事件处理及简单的动画处理等内容。

第5章的“拼图”游戏和文曲星里的拼图游戏类似。这一章接着上一章，继续讲述Java的图形和多媒体应用，以及Applet的知识和多线程技术等内容。

第6章通过编写“记事本”和“弹球”游戏，来介绍Java语言编程中的Swing包和I/O知识。Swing包基本已经取代AWT包，成为目前最流行的GUI编程技术；I/O知识是程序本地化必不可少的条件。

第7章编写了一个“俄罗斯方块”游戏，我们要编写的俄罗斯方块是一个应用程序（Java Application），不再是嵌到网页里的那种小应用程序（Applet）。游戏设计过程中涉及到游戏框架、游戏界面的编写，菜单处理、变量的定义、算法设计以及预览功能的实现。通过这一章，读者可以领略到Java的整体编程风格。同时，我们还为游戏添加了其他功能，如

设计About对话框，实现游戏分数的存档，为游戏添加状态栏等。在此我们可以学到如何定义类和类成员及其方法，如何使用JBuilder来创建一个对话框，并将对话框和主应用程序联系起来。

在第8章中，我们将把这个游戏改编为网络俄罗斯方块游戏，并使用了最新的Java 2标准Swing组件，使得界面更具有专业效果。本章主要讲解游戏的网络实现，使游戏能够实现多用户网络对战，其中涉及到网络模块的编写，ServerSocket的使用，如何实现服务器/客户间即时通信，NetRead接口的定义，聊天界面的设计，网络协议的设计，网络连接的实现及如何将游戏打包并发布。通过这些内容让读者领略Java强大的网络功能，学习Java的网络编程技巧。

本书在选题、策划及编写时，努力做到以下几点：

- 趣味性
- 直观性
- 可操作性
- 循序渐进

我们将电脑游戏和程序设计这两个精彩的世界嫁接在一起，希望读者能在充满趣味的学习过程中轻松地入门，尝试编程的乐趣，以便尽早掌握这一现代编程工具。

本书所附光盘的使用说明

随本书带有一张光盘。光盘中含有本书中涉及到的全部示例的源代码及各种资源文件，以方便读者在学习过程中查阅、参考。

编 者

2004年10月

目 录

第1章 Java基础	1
1.1 Java简介	1
1.1.1 Java的历史	1
1.1.2 Java的特性	2
1.1.3 Java的应用	4
1.1.4 J2SE 1.5的新特性	4
1.2 Java语言基本概念	6
1.2.1 基本数据类型	6
1.2.2 数组	11
1.2.3 运算符和表达式	13
1.2.4 基本控制语句	17
1.3 编写和运行Java程序	25
1.3.1 Java开发工具简介	25
1.3.2 Hello World	26
1.3.3 编译和运行	26
1.4 本章知识回顾	27
第2章 面向对象编程起步	32
2.1 类和对象	32
2.1.1 类	32
2.1.2 对象	34
2.1.3 一个小问题——static	35
2.2 类的继承和多态	36
2.2.1 Java的继承	36
2.2.2 abstract类和接口	38
2.2.3 多态	40
2.3 包	41
2.3.1 包的定义	41
2.3.2 包的使用	41
2.3.3 对包内类的访问权限	42
2.4 “幸运52”游戏	43
2.4.1 游戏效果说明	43
2.4.2 编写HTML页面	44
2.4.3 编写Java代码	45
2.4.4 程序在JBuilder中的编写	55
2.5 进一步实践	60

2.6 本章知识点回顾	60
第3章 持有对象与异常处理	64
3.1 持有你的对象	64
3.1.1 Array (数组)	64
3.1.2 Collection (集合)	64
3.1.3 Mapping (映射)	68
3.2 “球迷必答”	70
3.2.1 游戏规则	70
3.2.2 游戏注意点	70
3.2.3 建立工程	71
3.2.4 编写问题封装文件QuestionBean	72
3.2.5 编写游戏界面	73
3.2.6 处理事件	76
3.2.7 使用UI编辑器来编写消息对话框	77
3.3 异常处理机制	81
3.3.1 基本异常	81
3.3.2 捕获异常	82
3.3.3 重新抛出异常	85
3.3.4 标准Java异常	85
3.3.5 创建自己的异常	86
3.4 “速算24”游戏	87
3.4.1 游戏效果说明	87
3.4.2 编写游戏规则	88
3.4.3 创建工程和Applet	88
3.4.4 设计游戏界面	90
3.4.5 增加对表达式处理的方法	97
3.4.6 添加对Applet中按钮的事件处理	99
3.4.7 进一步实践	101
3.5 本章知识点回顾	101
第4章 Java编程深入——图像与多媒体	105
4.1 AWT简介	105
4.2 布局管理器	106
4.2.1 FlowLayout	106
4.2.2 BorderLayout	107
4.2.3 GridLayout	107
4.2.4 CardLayout	107
4.3 图像简介	108
4.3.1 文件格式	108
4.3.2 图像的创建、加载和显示	108
4.3.3 ImageObserver	109

4.3.4	MediaTracker	111
4.3.5	Graphics类	112
4.4	事件处理	114
4.4.1	事件处理机制	114
4.4.2	事件类	115
4.4.3	事件监听器接口	115
4.5	“精彩闹钟”	117
4.5.1	程序效果说明	117
4.5.2	实现简单的界面	117
4.5.3	画出表盘和表针	123
4.5.4	让闹钟动起来	125
4.5.5	给闹钟加上声音	126
4.6	“模拟钢琴”游戏	128
4.6.1	游戏效果说明	128
4.6.2	实现简单的界面	129
4.6.3	添加事件处理	133
4.6.4	继续完善这个游戏	137
4.6.5	加上音响效果	142
4.6.6	鼠标拖动时实现琴键的自动按下和释放	144
4.6.7	动画效果的改进	148
4.6.8	Java的局限	149
4.7	本章知识点回顾	149
第5章	拼图游戏——Applet和线程	156
5.1	Applet基础	156
5.1.1	Applet简介	156
5.1.2	Applet体系结构	156
5.1.3	Applet框架	157
5.1.4	其他一些有用的方法	158
5.1.5	AppletContext接口的主要方法	159
5.2	线程技术	159
5.2.1	继承线程	160
5.2.2	Thread和Runnable	161
5.2.3	线程的优先级	162
5.2.4	线程同步	164
5.2.5	多线程技术	164
5.3	“拼图”游戏	165
5.3.1	游戏的简单设计	166
5.3.2	实现简单的界面	166
5.3.3	事件处理	172
5.3.4	让游戏能够判断游戏当前状态，并能重新开始	180
5.3.5	让游戏的每次初始化状态都不一样	182

5.3.6 消除闪烁问题	183
5.3.7 让游戏记录玩家所用的时间, 并计算出分数	183
5.3.8 利用多线程技术来实现计时器, 记录玩家玩的时间	184
5.3.9 用F1键重新开始游戏	186
5.3.10 利用HTML的param标记来改变不同的图片	187
5.3.11 增加预览的功能	193
5.3.12 加入音响效果	196
5.3.13 CGI程序——进一步实践	197
5.4 本章知识点回顾	200
第6章 Swing和I/O简介	203
6.1 Swing组件简介	203
6.1.1 JApplet	203
6.1.2 按钮类	203
6.1.3 JComboBox	210
6.1.4 滚动窗格	210
6.1.5 树	210
6.1.6 表格 (JTable)	211
6.2 I/O系统	213
6.2.1 输入和输出	213
6.2.2 FilterInputStream和FilterOutputStream	215
6.2.3 File	216
6.2.4 Java1.1的I/O流	218
6.2.5 几个比较重要的类	218
6.3 “记事本”程序示例	222
6.3.1 建立工程和框架文件	223
6.3.2 完成界面的设计	224
6.3.3 添加“文件”主菜单事件响应	229
6.3.4 添加“编辑”和“关于”主菜单的事件响应	232
6.3.5 添加按钮的事件	233
6.4 “弹球”游戏	234
6.4.1 游戏的简单设计	234
6.4.2 实现简单的界面	234
6.4.3 让小球运动起来	238
6.4.4 事件处理——让游戏能够玩起来	242
6.4.5 让游戏能够判断当前状态, 并能重新开始	243
6.4.6 让游戏记录玩家的生命, 并计算出分数	244
6.4.7 加入音响效果	245
6.5 本章知识点回顾	246
第7章 俄罗斯方块游戏——综合应用示例	254
7.1 游戏效果说明	254

7.2 游戏的简单设计	255
7.3 编写游戏框架	255
7.3.1 编写游戏框架	255
7.3.2 为游戏编写菜单项	258
7.3.3 为菜单增加事件处理	261
7.4 开始编写游戏界面	264
7.4.1 在Panel中加入几个必要的常量和变量	264
7.4.2 在Panel上画出游戏界面	265
7.4.3 将Panel加到框架里去	267
7.5 开始编写游戏内容	267
7.5.1 定义一个方块类	267
7.5.2 定义描述游戏状态的变量	268
7.5.3 初始化游戏状态	269
7.5.4 编写产生新的方块组的算法	269
7.5.5 编写方块组移动的算法	271
7.5.6 编写将一行消去的算法	273
7.5.7 在游戏里实现一个线程	273
7.5.8 控制游戏开始	274
7.5.9 让游戏暂停	275
7.5.10 让游戏结束	276
7.5.11 判断游戏是否结束	276
7.5.12 让玩家控制游戏——添加事件处理	276
7.5.13 重新编写画图的功能代码	278
7.5.14 类的同步方法	279
7.5.15 计算游戏的得分和当前游戏级别	280
7.6 添加游戏的其他功能	280
7.6.1 设计About对话框	280
7.6.2 设计设定游戏等级的对话框	285
7.6.3 为游戏添加状态栏	289
7.7 封装得分情况	290
7.7.1 定义Score类和类成员	290
7.7.2 定义方法writeToFile()	291
7.7.3 定义方法readFromFile()	293
7.7.4 定义方法sortScore()	294
7.7.5 定义方法isScoreTop(int nScore)	294
7.7.6 定义方法insertNameScore(String sName,int nScore)	295
7.7.7 定义方法getScore()和getName()	295
7.8 编写对话框，让玩家输入名字	296
7.8.1 输入玩家名字的对话框	296
7.8.2 查看排名榜的对话框	298
7.9 本章知识点回顾	301

第8章 网络俄罗斯方块游戏——Swing组件与网络功能	304
8.1 游戏效果说明	304
8.2 游戏的简单设计	306
8.3 编写简单的网络模块	307
8.3.1 网络的基本概念	307
8.3.2 ServerSocket简介	309
8.3.3 Socket简介	310
8.3.4 InetAddress类方法简介	310
8.3.5 简单的服务器端实现	311
8.3.6 简单的客户端实现	318
8.3.7 用两个端口实现即时通信	319
8.3.8 定义接口NetRead	325
8.3.9 如何使用这个模块——server包和client包	326
8.4 改编游戏框架	326
8.4.1 让界面画出两个游戏区域	327
8.4.2 增加菜单及其事件处理功能	329
8.4.3 连接对方的对话框设计	331
8.4.4 设计聊天界面	333
8.4.5 编写显示双方分数的对话框	335
8.4.6 编写警告对方不能运行某个命令的提示框	337
8.5 把网络模块加入到游戏之中	340
8.5.1 网络协议的设计	340
8.5.2 实现网络连接	341
8.5.3 实现聊天功能	346
8.5.4 能够显示对方玩的状态	347
8.5.5 当一方消去一行时，对方随机增加一行	348
8.5.6 游戏开始、停止、暂停	350
8.5.7 一方结束时，使对方结束，并弹出游戏得分状况的对话框	351
8.5.8 在游戏关闭的时候，将所有网络资源释放	351
8.6 将游戏打包，发布我们的游戏	352
8.6.1 打包的好处	352
8.6.2 如何打包	352
8.7 Java代码风格和编写文档	356
8.7.1 Java的代码风格	356
8.7.2 变量命名规则	357
8.7.3 编写文档	358
8.8 进一步实践	361
8.8.1 游戏还存在的问题	361
8.8.2 使游戏界面变得更漂亮	362
8.9 本章知识点回顾	362

第1章 Java 基础

Java是一门发展很快的计算机语言，现在许多大型项目开始逐渐考虑使用Java来编写。Java以其独特的可跨平台运行和开发速度快、维护方便等特性受到许多开发者的青睐。这一章我们将要介绍一些Java的基础知识，从中读者可以了解到Java的历史、现状和前景，同时也会了解到如何编写和运行一个简单的Java程序。

1.1 Java简介

Java是由Sun公司开发的新一代编程语言，使用它可在各种不同机器、不同操作平台的网络环境中开发软件和使用软件。不论你使用的是哪一种网络浏览器，哪一种计算机，哪一种操作系统，只要机器上已经安装了Java运行环境（Java Runtime Environment）或者系统已经注明了“支持Java”，你就可以看到嵌有Java小应用程序（Applet）的生动活泼的网页，你也可以运行别人在任何机器、任何操作系统上编写和编译的Java应用程序。Java正在逐步成为Internet上主要的开发语言，也正在成为许多大型网络系统的首选开发语言。它彻底改变了应用软件的开发模式，带来了自PC机面世以来的又一次技术革命，为迅速发展的信息世界增添了新的活力。

1.1.1 Java的历史

Sun的Java语言开发小组成立于1991年，其目的是开拓消费类电子产品市场，例如，交互式电视、烤面包机等。Sun内部人员把这个项目称为Green，那时World Wide Web还没有发展呢！该小组的领导人James Gosling是一位非常杰出的程序员。他出生于1957年，于1984年加盟Sun Microsystem公司，之前在一家IBM研究机构工作。他是Sun NeWs窗口系统的总设计师，也是第一个用C实现的EMACS的文本编辑器COS MACS的开发者的。

由于消费类电子产品追求的是可靠性、标准化、使用简单、费用低等特性，对采用什么处理器不感兴趣。所以，迫切需要一种语言能够使所开发的系统与应用平台（例如：无论是什么型号的交互式电视）无关，使得在一种平台上开发的控制系统能够很方便地移植到各种类型的平台上。

为了实现系统与平台无关的目标，Gosling决定改写C编译器。但是，Gosling马上就发现仅仅靠改写是无法满足需要的，于是在1991年6月决定自己开发一门新的语言。他为这门语言起名为Oak，这就是Java语言的前身。因为Oak是Sun公司另一种语言的注册商标，所以Java是后来才改的名。

Gosling在开始写Java时，不仅仅扩充了语言机制本身，更侧重于语言所运行的软硬件

环境。Gosling在设计中采用了虚机器码（Virtual Machine Code）方式，即Java语言编译后产生的是虚机器码，虚机器码由Java虚拟机解释执行，Java虚拟机中有一个解释器。每一个操作系统都可以创建一个Java虚拟机以解释执行Java的虚机器码。这样一来，Java就成了与平台无关语言。

到了1994年，WWW已如火如荼地发展起来。Gosling意识到WWW需要一个中性的浏览器，它不依赖于任何硬件平台和软件平台，它应是一种实时性较高、安全可靠、有交互功能的浏览器。于是Gosling决定用Java开发一个新的Web浏览器，这就是Hot Java浏览器。它虽然没有获得大的发展，但是却展示了Java可能带来的广阔市场前景。Hot Java在1995年5月23日发布，在业界引起了巨大的轰动，Java的地位也随之得到肯定。又经过一年的试用和改进，Java 1.0版终于在1996年初正式发布。

Hot Java浏览器的出现，也导致了几个主流浏览器开始支持Java小应用程序（Applet）。现在，IE、NS都能够支持Java Applet，使得页面更加丰富多彩了。

1997年11月，国际标准化组织正式批准了Sun等公司提出的Java标准，Java标准化促进了它的进一步发展，也标志着Java语言的成熟。

目前Java已经发展了多个版本，从1.02到1.1、1.2 到目前的1.5 版本。Java 1.2以后的版本统称为Java 2。每次新版本发布，都会给Java注入新的生命力。

1.1.2 Java的特性

Java具有两大特性：可跨平台运行，面向对象。这两个特性是Java得以迅速发展的重要原因。

跨平台运行机制

Java程序具有可跨平台运行的特性，这个特性取决于它的应用程序的运行机制。通常计算机语言的运行机制如图1.1所示。

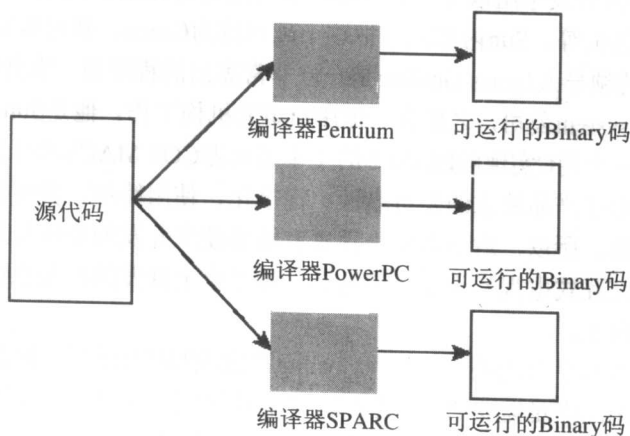


图 1.1 通常计算机语言的运行机制

而Java的运行机制则如图1.2所示。

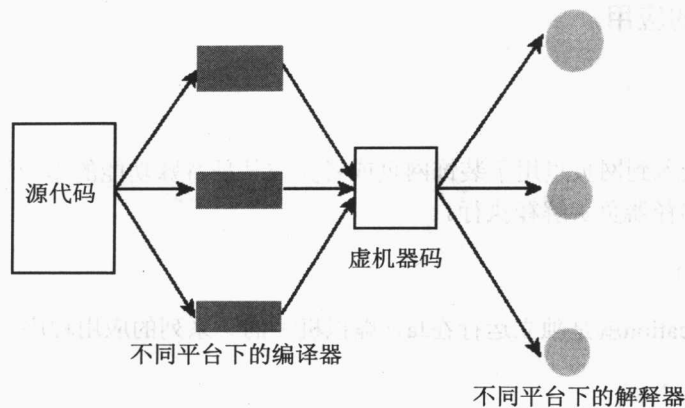


图 1.2 Java 运行机制

由图1.1、图1.2可以得知，普通的计算机编程语言大都是将源代码编译成机器码，对不同的平台其机器码是不同的。由于不同平台的特性和应用程序接口（API）的不同，其源代码往往也是不同的。这就造成了普通编程语言在移植上的困难。而Java引进的虚机器码的概念解决了这个问题。因为所有的Java解释器都是按照同一个规则来解释执行Java虚机器码的，所以所有的Java源代码最终生成的虚机器码对不同平台都是一样的。同时所有的虚机器码都是运行在相同的Java虚拟机上，所以，Java对于不同的平台却可以有相同的源代码。这就是Java可跨平台运行的特性。

对于运行于Internet上的网络程序，一个最大的要求是可以跨平台运行。这是因为网络上会存在许多不同的机器、不同的操作系统。而Java的跨平台特性使之成为最理想的网络编程语言。

面向对象

面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象。对象之间通过信息相互作用。另外，现实世界中任何实体都可归属于某类事物，任何对象都是某一类事物的实例。如果说传统的过程式编程语言是以过程为中心、以算法为驱动的话，面向对象的编程语言则是以对象为中心、以信息为驱动。用公式表示，过程式编程语言为：程序=算法+数据；而面向对象编程语言为：程序=对象+信息。

所有面向对象编程语言都支持3个概念：封装、多态性和继承，Java也不例外。Java提供了类机制和接口。通过类和接口，我们就可以实现对方法和数据的封装，同时通过类的继承以及继承时的多态性，我们实现了代码复用。Java程序是由一个一个的类组成，这决定了Java是一种纯粹的面向对象的编程语言。Java的这个特性决定了使用Java语言开发产品的高速度和产品的可维护性。

其他特性

Java还有一些特性，比如分布性、健壮性、安全性、支持多线程等等，这些特性都反映了Java语言设计的精妙。同时，Java也是一门简单易学的语言，相信读者很快就会认识到这一点。

1.1.3 Java的应用

Applet

Applet是嵌入到网页里用于装饰网页或者完成某种特殊功能的小应用程序。Applet由浏览器中的Java解释器负责解释执行。

Application

Java Application就是独立运行在Java虚拟机上的一系列的应用程序。

servlet

servlet是运行在服务器端的小程序，它负责处理客户端传来的请求（request），然后传给客户端一个响应（response）。

J2SE、J2EE 和 J2ME

J2SE（Java 2 Standard Edition）就是我们印象中的普通的Java语言，这是一个标准的API和工具包，是J2EE和J2ME的基础。

J2EE（Java 2 Enterprise Edition），顾名思义，它向企业级用户提供服务，使得企业可以把自身的信息通过Internet的形式发布，也就是电子商务技术。同时可以用于解决电子政务等一系列问题。

电子商务和信息技术的快速发展及对它的需求给应用程序开发人员带来了新的压力。为了降低成本并加快企业应用程序的设计和开发，J2EE平台提供了一个基于组件的方法，来设计、开发、装配及部署企业应用程序。J2EE平台提供了多层分布式应用模型、组件重用、一致化的安全模型以及灵活的事务控制。

J2ME（Java 2 Micro Edition）类似于J2EE的一个平台。其实Java最早的目的就是满足一些嵌入式装置的开发需要。在Java 1.0.2以前的几个版本，这项功能一度被忽略，但是在1.0.2版本的时候进行了重新开发。当发展到Java 2的时候，Sun推出了一个专门的软件包J2MEWDK，支持进行嵌入式开发。目前大家关注最多的是手机游戏的开发，但是J2ME能做的远远不止这些。

如果读者对J2EE或者是J2ME有兴趣的话，请到<http://java.sun.com/>上寻找更多的信息。

1.1.4 J2SE 1.5的新特性

J2SE 1.5（开发代号Tiger）是Java平台和语言上的一个重要修改，目前主要包括了15个JSR的请求和其他一些类似的更新。这次的Release主要关注于几个关键的主题：品质、监视和管理、性能和可扩展性、轻松的开发以及桌面客户端。

轻松的开发（Ease of Development）

你可能已经听到过关于减轻开发难度而做的Java语言上的修改。这次J2SE根据JSR的要求实现以下几个JSR：JSR 201 包含了4个修改，JSR 175的核心是支持元数据（metadata），

而JSR 14则规范了泛型。元数据功能提供了声明式的开发，并且取代了一些工具的代码生成和维护功能。泛型提升了无需手动转型（manual casting）的代码复用，通常manual casting都会带来类型安全性上的一些问题。

另外的4个修改分别是：

- （1）用for循环来遍历容器（Collection 类型），而不需要显式的声明容器的迭代器（Iterator）。
- （2）枚举类型提供了超越类似final static int的、增强的类型安全性。
- （3）在泛型中使用基本数据类型（primitive types）的时候提供自动装箱（autoboxing）功能（原来的泛型中是不能使用基本类型的）。
- （4）引入了静态常量（static constants）类，改进显式的共享一个静态数据。

为了实现轻松的开发，除了语言上的修改之外，还有一些额外的东西，比如一些怀念printf函数的人会发现它又回来了；一个新的并发工具（在JSR 166中提及）将使得多线程编程变得更加简单轻松。

可扩展性和性能（Scalability and Performance）

J2SE 1.5版本承诺改进软件的可扩展性和性能，尤其是在启动时间和内存印记（memory footprint）上，将使得用户能以最快的速度发布一个应用程序。从JSR 163上将实现内建的性能工程（performance ergonomics）和一个功能强劲的API profile工具。

监视和管理（Monitoring and Manageability）

监视和管理是Tiger中的一个主要特性。那些在J2EE平台上使用JMX的开发看到这样的特性能在J2SE中实现将会非常开心。通过对JVM的监视，将允许对已发布的应用程序健康性的完全检查，包括对底层内存泄漏检测，错误处理甚至是API堆栈跟踪（stack trace）的监视。

核心的XML支持（Core XML Support）

J2SE 1.5的介绍被修改为XML的核心平台，表示Java的核心API将包括XML1.1、SAX 2.0和DOM Level3。Web service方面的API、JAX-RPC和JAXB将继续出现在Web Service的包中，在以后的新版本中这些API将被添加到核心API中去。

桌面客户端（Desktop Client）

最后一个重点的新特性是桌面客户端。这将带来几个内建的新的Look & Feels支持，并且增强了对皮肤（skins）的支持。除了启动速度和内存印记的增强，桌面开发者又多了几个值得期待的新特性。

对Unicode 3.1的支持（Unicode 3.1 Support）

32位的代理字符（surrogate character）支持将会很谨慎地添加到新的版本中，所以1.5版本将仍然使用16位的char类型。

新的 IO 支持 (New IO Support)

新版本将提供对异步IO的支持, 并且支持在平台中适当的地方开拓更深入的使用这些 API。

1.2 Java语言基本概念

前一节已经简要介绍了Java语言, 现在就开始深入到Java语言内部, 来介绍Java语言的基本数据类型、控制语句等重要组成部分。学完这一节, 我们就可以编写一个非常简单的小程序。

1.2.1 基本数据类型

数据类型指明了变量和表达式的状态和行为。本节我们将讲述Java的简单数据类型。

常量与变量

(1) 常量

Java有几种类型的常量, 如整型常量345, 浮点型常量33.2等, 所有常量都用final做修饰符。

下面是几个常量的定义:

```
final int a=345;
final float b=33.2;
final double c=33.2d;
final String d='dddddd';
```

(2) 变量

变量是Java程序中的基本存储单元, 它的定义包括变量名、变量类型和作用域。

变量名应该是一个合法的标识符, 它允许包括字母、数字、下划线、\$字符, 不能以数字开头, 而且不能为保留字。

变量的声明格式为:

```
type identifier [=value][, identifier[=value]...];
```

多个变量之间用“,”分隔, 结束用“;”。

下面是几个合法的变量声明:

```
int a1,b2,c3;
int a5=6,b9=8;
float tr_$=33.;
double sxf$;
String _s44='dddd';
```