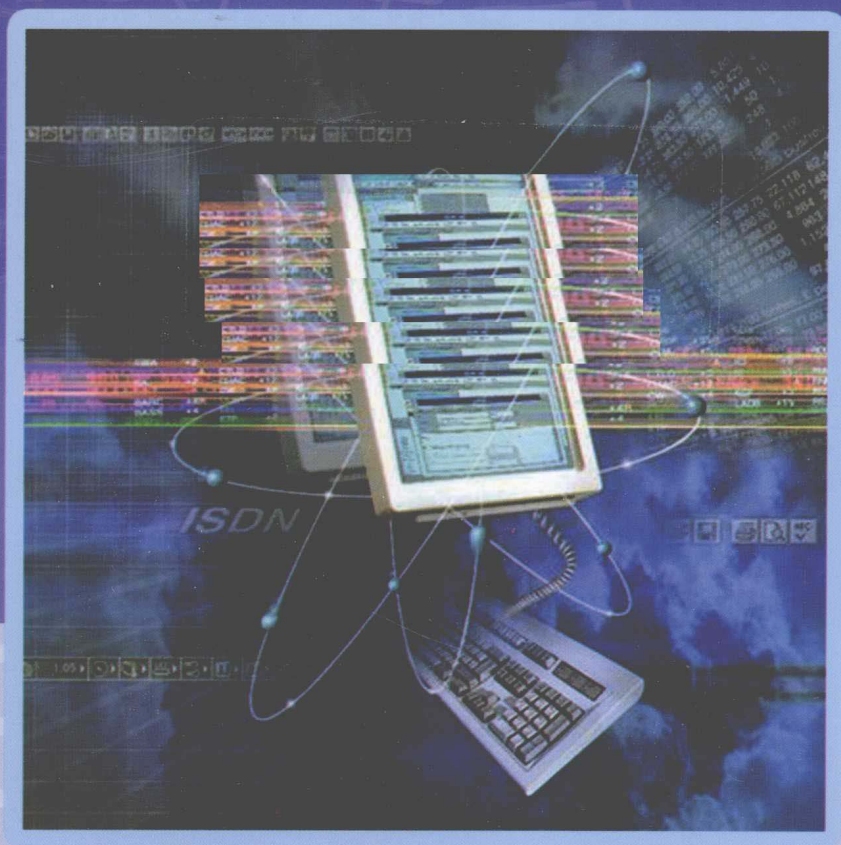


JAVA YUYAN CHENGXU SHEJI SHIYONG JIAOCHENG

Java语言程序设计 实用教程

主 编 傅晓阳
副主编 郭晓燕 李 昱



 吉林大学出版社

吉林大学珠海学院立项教材

Java语言程序设计实用教程

主 编：傅晓阳

副主编：郭晓燕 李 昱

吉林大学出版社

前 言

自20世纪90年代以来,随着Internet的迅猛发展,使得人们对网络的应用和需求大大提高,网络本身也为这些需求提供了丰富多彩的应用平台。在这个技术飞速发展的时代,Sun公司于1995年推出了Java语言,这是一种完全面向对象的程序设计语言,具有简单易用、跨平台运行以及网络安全性强等特点,它在整个IT行业中掀起了研究开发应用的浪潮。随着十多年的发展和技术积淀,Java语言广泛应用于企业信息化、网络通信、智能终端等各行各业,从而奠定了Java成为计算机主流编程语言的地位,对整个计算机软件行业的发展产生了深远的影响。因此,在大学本科理工科专业开设Java程序设计课程作为程序设计学习以及面向对象的思想训练是十分必要的。

尽管当前市场上关于Java语言程序设计的书籍比较丰富,但随着Java语言的发展,关于Java语言编程的内容也需要不断地更新换代,基础知识的介绍性书籍已经无法满足广大读者和开发人员的需要。针对Java语言读者群体的需求层次多、应用面广等特点,亟需系统的、应用指导性强的Java书籍以适应不同读者的需求。笔者多年从事计算机本科学生的Java程序设计的教学工作,并为高年级同学讲授应用性很强的J2EE高级编程课程,参加并组织了基于3G通信网络的移动教育系统的开发工作,具有一定的实际应用开发经验。因此,笔者在教学讲稿的基础上,结合了实际应用的经验,组织编写了这本教材。

全书分为14章,主要讲授Java面向对象编程原理、文件与数据处理,Java的异常处理,小应用程序编程,基于swing的图形用户界面设计,Java网络编程和多线程编程。这些课程是为希望系统了解Java基础知识的读者开设的。而集合类和数据库编程这两章,可作为读者进一步深入学习Java高级编程的基础章节,授课教师可以根据课时安排自由选取。

第1章 介绍Java语言的发展历史与编程特点,建立开发环境以及说明开发流程与运行原理。

第2章 概要介绍Java语言的数据类型、语句的控制逻辑和编程结构。

第3章 介绍Java面向对象编程的基本概念和术语,详细说明了类、对象、成员变量和成员方法的定义和用法,并进一步说明了面向对象程序设计的封装性、继承性和多态性是如何实现的,通过例题详细介绍了重载、包、接口、内部类的基本用法。

第4章 介绍了Java语言中如何使用数组和字符串进行程序设计。着重介绍了数组的基本用法和常量字符串String以及可变字符串StringBuffer。

第5章 介绍Java语言异常处理,着重讲解异常处理类型和异常处理结构。

第6章 介绍Java语言的输入输出的相关知识,结合特定类库讲解文件处理和数据处理编程的方法。

第7章 介绍jdk 1.5的高级特性,说明泛型、枚举在迭代中的用法。

第8章 着重介绍Java.util类包中关于集合类中集(set)、列表(List)和映射(Map)

的基本用法。

第9章 具体介绍一些编程工具,如`java.text`, `java.math`, `java.util`等类包中的相关工具类的基本用法。

第10章 介绍Java图形用户界面(GUI)程序设计的基本原理与实现,着重介绍基于Swing组件的GUI编程的组件布局、事件处理方面的知识及用法。

第11章 介绍Java小应用程序的编程方法,包括小应用程序在多媒体编程以及网络应用方面的知识。

第12章 介绍Java语言的多线程程序设计的两种基本方法,并举例说明了线程同步以及线程死锁的基本知识与编程方法。

第13章 介绍了Java语言的网络编程的基本原理与方法,举例说明了URL类、Socket套接字以及UDP网络编程的具体用法。

第14章 介绍了Java语言数据库编程的基本原理与方法,简单介绍了数据库语言和数据库工具,举例说明了JDBC数据库编程操作的基本用法。

本书的后面还提供了附录A作为大学本科教学的实验指导,包含了与教学相关的实验内容。

本书在编写过程中,主要突出以下特点:(1)知识系统:本书主要使用对象是大学本科的学生以及Java语言爱好者。对于初学者,组织结构系统合理、内容深入浅出将有助于读者较好地理解课程内容,提高阅读兴趣,为进一步深入学习打下良好基础。(2)案例实用:本书所有章节都配有详尽的实例,并配有详细的注解,有利于读者理解知识概念,并提高编程能力。书后附录配有实验指导,有助于学生上机操作。(3)技术更新:Java语言技术发展日新月异,本书在编写过程中,注意基础知识和实际应用的点面结合,并及时更新了Java语言最新的编程技术。结合Java语言的实际开发经验,注重读者的理论知识与实际编程操作能力相结合,力求增强读者系统学习Java语言的兴趣和扩展实际编程的能力。

本书可作为高等学校计算机应用专业以及其他相关专业本科生的教材,也可以作为Java语言编程人员的参考资料和培训教材。

本书前言、第1章—第4章、第6章—10章、第13章以及相关实验部分由傅晓阳编写,第11章、第12章以及相关实验部分由郭晓燕编写,第5章、第14章以及相关实验部分由李昱编写,由孙孔峰参与了本书例题以及实验部分的校对工作。全书由傅晓阳统一审稿校对,得到了吉林大学珠海学院计算机系姜云飞教授、陈守孔教授的大力支持并提出许多宝贵意见,同时本书在编写过程中还得到许多老师的大力支持和帮助,对于他们的辛勤付出,在此表示衷心的感谢。

在本书的编写过程中,作者参阅借鉴了大量的参考文献,在此向诸多学者表示衷心的感谢。由于作者水平有限,虽然对本书进行了反复审校,但错误和缺点在所难免,希望读者给予批评指正以及提出宝贵建议。

编者

2010年3月于珠海

目 录

前 言	1
第1章 Java简介	1
1.1 Java语言的发展历史及特点	1
1.1.1 发展历史	1
1.1.2 编程语言的特点	2
1.1.3 应用领域与发展前景	3
1.2 建立Java开发环境	4
1.2.1 安装JDK	4
1.2.2 Java集成开发工具eclipse下载和安装	9
1.3 Java程序工作原理与开发流程	14
小 结	15
习 题	16
第二章 Java语言基础	17
2.1 标识符和关键字	17
2.2 数据类型	19
2.2.1 基本数据类型	20
2.2.2 常量与变量	21
2.3 运算逻辑	22
2.3.1 算术运算符	23
2.3.2 关系运算符	23
2.3.3 逻辑运算符	24
2.3.4 位运算符	25
2.4 控制流程	27
2.4.1 分支语句if/switch	27
2.4.2 循环语句while/do-while/for	31
2.4.3 跳转语句break/continue	35
小 结	36
习 题	37
第三章 Java面向对象编程基础	38
3.1 类与对象	38
3.1.1 类的定义	40

3.1.2 对象的创建与使用	41
3.2 变量与方法	44
3.2.1 实例变量与静态变量	44
3.2.2 方法的声明与调用	46
3.2.3 方法的参数传递	48
3.2.4 变量与方法的使用范围	50
3.3 包与封装	53
3.3.1 包	53
3.3.2 封装	57
3.4 继承与多态	61
3.4.1 继承性	61
3.4.2 多态性	71
3.5 抽象与接口	74
3.5.1 抽象类	74
3.5.2 接口	77
3.6 内部类	84
3.6.1 实名内部类	84
3.6.2 匿名内部类	88
小 结	90
习 题	90
第四章 数组和字符串	91
4.1 数组的基本操作	91
4.1.1 一维数组	91
4.1.2 多维数组	95
4.2 字符串操作	97
4.2.1 常量字符串String	98
4.2.2 变量字符串StringBuffer	107
小 结	111
习 题	112
第五章 异常处理	113
5.1 异常的类型	113
5.2 异常处理结构	115
5.2.1 捕获异常	116
5.2.2 抛出异常	121
5.3 自定义异常类型	123
小 结	125
习 题	126
第六章 文件与数据处理	127

6.1 字节流与字符流	127
6.1.1 字节流类和字符流类概述	127
6.1.2 标准输入输出流	129
6.1.3 数组流	130
6.1.4 字符串流	131
6.1.5 数据流	131
6.1.6 缓冲流	135
6.2 控制台输入输出操作	138
6.3 文件数据操作	140
6.3.1 FileInputStream和FileOutputStream	141
6.3.2 FileReader和FileWriter	143
6.3.3 文件目录操作	145
6.3.4 文件的随机存取操作	148
小 结	151
习 题	151
第七章 Java高级特性：泛型与枚举	152
7.1 泛型	152
7.1.1 定义泛型类与泛型方法	153
7.1.2 类型参数的基本用法	154
7.2 枚举	157
7.3 装箱与拆箱	160
小 结	161
习 题	161
第八章 集合类*	162
8.1 集合接口与实现	162
8.2 迭代器 (Iterator<T>)	163
8.3 集(set)	164
8.3.1 哈希集合类 (HashSet<E>)	164
8.3.2 树集合类 (TreeSet<E>)	166
8.4 列表 (List)	168
8.4.1 矢量类 (Vector<E>)	168
8.4.2 链表类 (LinkedList<E>)	170
8.4.3 堆栈类 (Stack<E>)	173
8.5 映射(map)	175
8.5.1 哈希映射 (HashMap<K,V>)	175
8.5.2 树映射 (TreeMap<K,V>)	177
8.6 算法集合Collections类	181
小 结	183

习 题	183
第九章 Java实用工具类	184
9.1 Object类和Class类	184
9.2 Random类生成随机数	188
9.3 BigDecimal类与BigInteger类	190
9.4 Date类、GregorianCalendar类和DateFormat类	195
小 结	198
习 题	199
第十章 Swing图形用户界面 (GUI) 编程	200
10.1 组件与容器	200
10.1.1 Swing组件概述	200
10.1.2 顶层容器和中间容器	201
10.1.3 JLabel标签组件和JTextField文本框组件	203
10.1.4 JButton、JRadioButton单选框和JCheckBox复选框组件	204
10.1.5 JTextArea、JScrollPane、JList和JComboBox组件	205
10.2 布局管理器	208
10.2.1 流式布局FlowLayout	209
10.2.2 方位布局BorderLayout	210
10.2.3 网格布局 GridLayout	211
10.2.4 卡片布局 CardLayout	213
10.2.5 箱式布局 BoxLayout	215
10.3 事件处理	217
10.3.1 Java事件处理模型	217
10.3.2 点击事件ActionEvent	219
10.3.3 鼠标事件MouseEvent和鼠标移动事件MouseMoveEvent	223
10.3.4 键盘事件KeyEvent和焦点事件FocusEvent	227
10.3.5 窗口事件WindowEvent	230
10.4 Swing高级组件编程	233
10.4.1 JMenu菜单组件	233
10.4.2 JTable表格组件	236
10.4.3 JTree树组件	241
10.4.4 多文档界面	244
小 结	247
习 题	248
第十一章 Applet应用	249
11.1 HTML与Applet	249
11.2 Applet的生命周期	251
11.3 开发小应用程序	253

11.3.1利用Applet布局图片	254
11.3.2弹出新窗口	256
11.3.3利用Applet播放声音	260
11.4 在Web浏览器中使用Applet插件	261
11.4.1采用Applet网页标记	261
11.4.2归档文件压缩Jar	262
11.4.3 为Applet传递参数	263
11.5 应用程序与Applet的区别	265
小 结	266
习 题	266
第十二章 多线程编程.....	267
12.1 实现线程的方法	268
12.1.1继承Thread类	268
12.1.2实现Runnable接口	270
12.2 线程的生命周期	271
12.3 线程的调度	273
12.3.1线程的优先级	273
12.3.2线程的常用控制方法	275
12.4 线程同步	277
12.4.1同步关键字synchronized	277
12.4.2 对象锁	283
12.4.3 wait()、notify()及notifyAll()方法	285
12.5 线程死锁	287
小 结	289
习 题	290
第十三章 网络编程.....	291
13.1 网络编程基础	291
13.2 URL编程	291
13.2.1 URL类	292
13.2.2 URLConnection类	294
13.2.3 InetAddress类	295
13.3 基于TCP的网络编程	296
13.3.1 ServerSocket类	297
13.3.2 Socket类	299
13.4 基于UDP的网络编程	301
13.4.1 DatagramPacket类	301
13.4.2 DatagramSocket类	302
小 结	305

习 题	305
第十四章 数据库编程*	306
14.1 数据库与SQL语言介绍	306
14.1.1 数据库基本概念	306
14.1.2 SQL语言组成	307
14.1.3 SQL数据定义	308
14.1.4 SQL数据查询	309
14.1.5 SQL数据更新	310
14.2 JDBC概述	311
14.2.1 JDBC结构模型	311
14.2.2 JDBC API	311
14.2.3 JDBC驱动程序	313
14.3 JDBC数据库操作	315
14.3.1 JDBC工作原理	315
14.3.2 设置环境	315
14.3.3 加载可用驱动程序	318
14.3.4 创建与数据源的连接	318
14.3.5 执行SQL语句	320
14.3.6 处理查询结果	321
14.3.7 关闭查询语句及与数据库的连接	322
14.4 综合应用	323
14.4.1 使用JDBC创建基本表及视图	323
14.4.2 使用JDBC执行数据库查询	325
14.4.3 使用JDBC更新数据库	330
14.5 数据库操作性能优化设计	332
14.5.1 优化Statement对象及ResultSet对象	332
14.5.2 合理使用JDBC的数据库事务操作	334
14.5.3 采用数据连接池提高效率	334
小 结	335
习 题	335
附录A 实验指导	336
实验一 Java开发环境设置	336
实验二 Java基本控制结构	336
实验三 类和对象	337
实验四 封装、继承和包	341
实验五 多态和接口	343
实验六 数组、字符串和异常处理	345
实验七 文件输入与输出	347

实验八 图形用户界面设计——布局管理与事件处理	349
实验九 基于swing组件的GUI应用设计	351
实验十 Applet应用	356
实验十一 多线程应用	358
实验十二 网络编程	361
实验十三* 数据库编程	364

第1章 Java简介

自从1936年,阿兰·图灵提出一种抽象的计算模型——图灵机以来,人们就梦想可以让机器辅助人们完成各种繁杂的计算工作。十年后,第一台数字式电子计算机在美国宾夕法尼亚诞生了,尽管这个30吨重的庞然大物每秒只能计算5000次加法运算,但这是人类文明里程碑式的发展,开启了从工业革命转变到信息化革命的重大社会变革。计算机产业的发展也已经渗入到人类生活的各个领域,成为人们日常生活中的重要工具。计算机产品由计算机硬件与软件组成,软件是计算机的灵魂,是人类与机器沟通的重要手段。从二进制机器语言、汇编语言等低级编程语言到C语言、Pascal语言等高级语言,计算机软件的发展历史还不到一百年。随着信息化社会的迅猛发展,以及Internet网络的广泛应用,Sun公司在1995年5月正式推出了一种完全面向对象的程序设计语言,并以其跨平台、简单易用、网络安全性强的特点,在短短的十多年间,成为继C语言以来最有影响力的一门程序设计语言,并广泛应用于通信、Internet网络、电子商务、企业系统集成等领域,成为计算机主流编程语言。

本章着重介绍Java语言的发展历史、语言特点、工作原理以及开发环境。

1.1 Java语言的发展历史及特点

1.1.1 发展历史

早在1990年,Sun公司为了进入消费类电子产品市场,由James Gosling负责成立了Green team项目开发小组。该项目主要目的是在小型电子设备中,通过编程实现设备的智能化。在开发期间,为了解决C语言内存管理方面的种种缺陷以及跨平台移植的困难,Gosling毅然决定在C++的基础上设计一种新的面向对象、跨平台的语言,并取名为Oak。然而,智能电子消费市场的发展远低于Sun公司的预想;尽管Oak语言有众多C++无法比拟的特点,采用该语言所完成的交互式电视的项目却遭到了失败。但Gosling所领导的开发小组没有放弃对Oak语言的修改完善,在1994年,随着Internet网络的广泛应用,Oak语言的稳定性、网络安全性以及良好的跨平台特点得到了业界的广泛赞誉。“柳暗花明又一村”,1995年,Sun公司重整旗鼓,将Oak注册为Gosling开发小组使用在工作中经常提到的“爪哇”(Java)咖啡的名称对Oak语言进行重新修订和注册该产品,并成为今天计算机的主流开发语言——Java语言。

Java语言在1995年发布了JDK 1.0版以来,陆续在1997年后发布了jdk 1.1版等几个JDK1.x版本,到1998年12月,发布了JDK1.2版,正式进入Java 2时代,这个时期是Java发展最为迅速的阶段。Java 2逐渐改变了JDK1.x版本发布以来的一些缺点,特别是将Java开发包分为J2SE、J2EE、J2ME三个版本。

· J2SE (Java 2 Platform, Standard Edition) 称为Java标准平台, 主要提供可以开发桌面应用和低端服务器应用的基础平台。

· J2EE (Java 2 Platform, Enterprise Edition) 称为Java企业平台, 支持构建企业级服务应用。

· J2ME (Java 2 Platform, Micro Edition) 称为Java微型平台, 支持嵌入式电子消费产品应用, 如移动电话、掌上电脑等手提式设备。

从Java 1.2版本开始, Sun公司以平均2年一个版本的速度开发新的Java版本。2000年, sun公司发布了Java1.3版本, 对Java 1.2版本进行了查漏补缺以及API的扩展工作。2002年, Sun公司发布了有史以来最为成熟的Java版本: J2SE 1.4。J2SE1.4对平台的性能进行了较多的考虑和改进, 大大提高了Java语言的运行效率。另外, 由于Sun公司结合IBM、Compaq等多个计算机知名公司共同制定J2SE1.4规范, 从而形成了一个高质量的、代表了Java共同体多样性的规范。尽管J2SE1.4版的执行效率得到了显著提高, 但Java面临另外一个问题就是其复杂性, 比如不支持微软.Vet中使用的一些高级语言特性, 如泛型、增强的For语句等。2004年, Sun公司发布了业界期盼已久的JDK 1.5, 并将其命名为J2SE 5.0。与J2SE1.4的主要区别是, J2SE5.0主要增加了Java的易用性。J2SE5.0增加了泛型、增强的For语句、Annotation注释语句、自动拆箱、装箱等功能, 减少了代码的重复性, 从而提高了Java语言的编程效率。

学习Java语言可以根据编程的应用领域, 学习J2SE、J2EE和J2ME。但首先应该学习J2SE, 因为J2SE是J2EE和J2ME的基础, 目前Sun公司已经推出J2SE6.0版本, 即JDK 1.6版, 可以登陆到Sun公司网站 (<http://Java.sun.com/javase/downloads/index.jsp>) 免费下载各种版本的JDK。下载软件是自解压的压缩文件, 运行该文件, 按照屏幕提示操作, 即可完成安装。具体安装过程参见1.2节建立Java开发环境。

1.1.2 编程语言的特点

早期的完全面向对象语言的代表是SmallTalk语言, 为面向对象语言奠定了基本框架。Java语言之所以成为面向对象语言的主流代表, 除了具有封装性、继承性、多态性等面向对象语言的主要特点外, 还与其发展历史有着紧密的联系。由于Java语言是在C/C++语言的基础上发展而来的, 并广泛地应用于Internet网络编程, 因此Java语言还具备跨平台性、编程可靠性高、网络安全性强、内置多线程、动态扩展性好等诸多主流语言的必备特点。

1. 面向对象编程特点

Java语言是由C++语言发展而来的, 是一种完全面向对象的编程语言。Java语言在语法上继承了C++语言的简洁明了的特点, 同时, Java语言为程序员提供了大量的具有强大功能的类库, 提高了程序员的编程效率; 在面向对象编程方面, 继承了SmallTalk语言的封装性、继承性和多态性等面向对象语言的主要特性, 允许程序员封装自定义的功能类, 增强了代码的灵活性和可重用性, 更加适合解决社会生活中复杂的实际问题。

2. 跨平台的特点

Java语言是随着Internet网络一起发展起来的。Java语言的最初设计是希望应用于网

络智能家电的集成控制，但由于家电设备芯片种类繁多，对于不同家电，相同程序的移植工作极为繁重。这就是Java语言诞生的原动力，并成为首个跨平台语言的代表。Java语言跨平台的工作原理是首先通过Java编译器将Java源程序编译成为一种成为字节码（bytecode）的中间代码，字节代码不是二进制机器指令代码，不能直接在计算机上运行，必须在一个特定的运行环境中运行，这个运行环境称为Java虚拟机（JVM）。当Java字节码在JVM上运行时，JVM运行环境将Java字节码解释成安装了JVM的计算机的CPU机器指令，于是就可以运行在这台机器上了。因此，只要不同的计算机平台安装了相应的Java虚拟机环境，相同的Java字节码就可以在不同的系统平台上运行了，从而实现了Java语言的平台无关性，即跨平台性。但Java语言由于其跨平台特点，使得Java语言成为一种解释性语言，这在一定程度上降低了Java程序的执行效率。

3. 编程可靠性高

Java语言在集成了C++语言的简洁特性的同时，避免了C++语言中的对指针程序代码的维护以及多重继承带来的困扰。通过内存对象垃圾的自动回收，减轻了程序员对内存空间垃圾的程序处理，统一了各种数据类型在不同操作平台上所占内存的大小，简化了编程中的类型转换工作，从而提高了编程的可靠性和编程效率。

4. 网络安全性强

Java作为网络编程语言，提供了足够的网络安全设计，比如运行小应用程序时，小应用程序必须运行在“沙箱”中，由于小应用程序使用字节码而不是机器码进行运行，字节码在运行前，需要经过JVM环境的代码校验、代码段格式检查、检测指针操作、对象操作等一系列代码验证工作，从而确保小应用程序不含计算机病毒。另外，“沙箱”将会禁止小应用程序对本地计算机磁盘进行写操作等，提高了网络应用的安全性。

5. 内置多线程

Java语言提供了内置多线程操作。通过多线程操作，编程人员可以使用不同的线程完成特定的功能，从而使应用程序具有并行处理能力。在多线程操作中，提供了同步操作的相关方法，保证了对共享数据的正确操作。多线程功能对于在多媒体应用中实现动画展现比较方便。

6. 动态扩展性好

Java语言可以打包为jar包文件，并可以动态地从本地或网上加载或替换功能类包，甚至可以在应用程序运行时，通过特定的应用框架，实现程序应用的功能插件“热”插拔技术。Java的类库可以根据需要进行扩展，另外，Java语言支持多重接口，使其具有更灵活的扩展方式。

1.1.3 应用领域与发展前景

由于Java语言的完全面向对象编程特点以及跨平台运行的方式，使得工业界不少人预言：“Java语言的出现，将会引起一场软件革命。”随着Internet网的无所不在和Java语言的发展，Java应用领域逐渐拓展到各行各业。Java语言对软件行业的开发技术有如下影响：

1) 软件的需求分析：可将用户的需求进行动态的、可视化描述，以提供设计者更加

直观的要求。而用户的需求是各色各样的，不受地区、行业、部门、爱好的影响，都可以用Java语言描述清楚。

2) 软件的开发方法：由于Java语言的面向目标的特性，所以完全可以用o-o的技术与方法来开发，这是符合最新的软件开发规范要求的。

3) 多媒体应用：动画效果远比GUI技术更加逼真，尤其是利用WWW提供的巨大动画资源空间，可以共享全世界的动态画面的资源。

4) 软件GUI应用：用Java语言开发的软件具有可视化、可听化、可操作化的效果。

5) 提高开发效率：Java语言适合于大型复杂应用软件的开发，开发效益、开发价值都有比较明显的提升。

Java语言在未来的IT行业中具有广泛的应用前景：

1) 所有面向对象的应用开发，包括面向对象的事件描述、处理、综合等。

2) 计算过程的可视化、可操作化的软件的开发，适合各种GUI应用程序的开发。

3) 动态画面的设计，包括图形图像的调用。

4) 交互操作的设计（选择交互、定向交互、控制流程等）。

5) Internet的系统管理功能模块的设计，包括Web页面的动态设计、管理和交互操作设计等。

6) Intranet。（企业内部网）上的大型软件企业应用集成（EAI）的开发。

7) 与各类数据库连接查询的SQL语句实现，应用于各种企业信息系统的构建。

8) 由于具有J2ME平台和J2EE平台，Java可以完成智能控制应用、移动计算、Web服务应用等。

1.2 建立Java开发环境

目前可供程序员使用的Java语言开发工具比较多，开发人员可以使用任何常用的编辑器（比如UltraEditor、EditPlus等）进行Java源程序的编写，然后使用Sun公司提供的JDK开发工具包中的命令工具对源代码进行编译、运行。但通常情况，集成开发环境（IDE）工具包装了Sun公司提供的JDK开发包，并集成了开发Java语言所需要的各种工具，同时为用户提供了开发项目所需的集成资源，方便了程序员对程序的开发。比较典型的IDE工具有JCreator，eclipse，JBuilder，NetBeans，IntelliJ IDEA等。不同的IDE有着不同的特性和优点，根据开发的不同需要可能会选择不同的IDE。对J2SE平台应用的开发，通常可以选择开源IDE eclipse。在使用eclipse开发工具前，必须先安装JDK，并设置相应的环境变量。

1.2.1 安装JDK

Sun公司提供的Java开发工具包（Java Development Kit）包含了编译和运行程序所必需的工具和相关类库，最新的JDK可以从Sun公司的官方网站上免费下载（<http://Java.sun.com>）。

1. 下载JDK6

1) 打开http://java.sun.com网站, 由于Sun公司已经被Oracle公司收购, 因此, 该下载网站作为Oracle公司的一个链接, 点击菜单Downloads->Java SE子菜单, 则进入Java开发工具包JDK6的下载页面, 如图1-1所示。



图1-1 Sun 公司JDK6下载页面1

2) 点击JDK 6 Update 20部分的红色下载按钮 (Download JDK) 进入下载页面, 如图1-2所示。

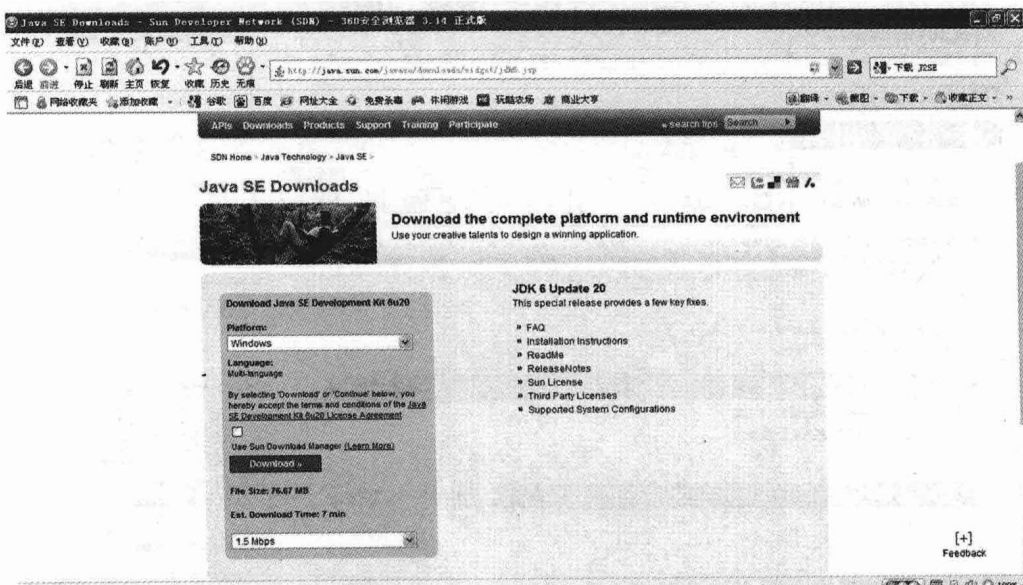


图1-2 Sun 公司JDK6下载页面2

3) 点击红色Download按钮, 弹出下载登录页面, 如图1-3所示, 点击最下面的Skip this Step链接, 可以跳过账户登录, 直接进入下载页面。如图1-4所示。

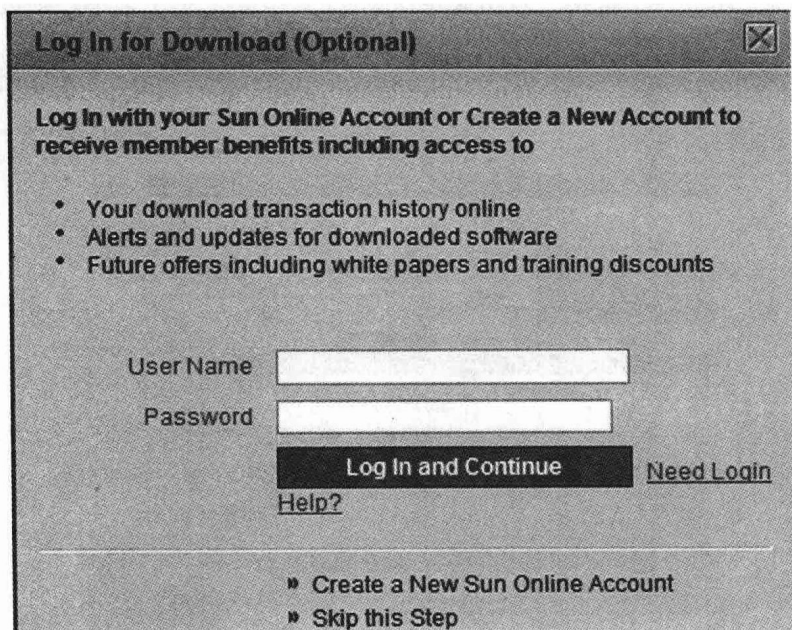


图1-3 Sun 公司JDK6下载页面3

4) 可以用鼠标点击红色箭头边上的jdk-6u20-windows-i586.exe下载JDK6安装程序。

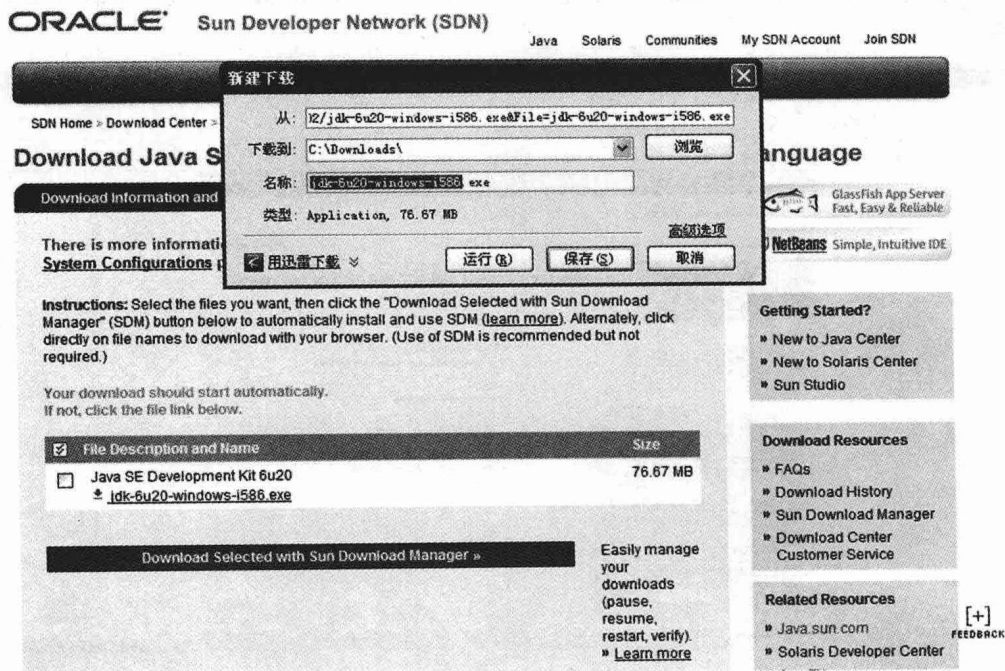


图1-4 Sun 公司JDK6下载页面4