

高等学校公共课计算机规划教材

Java实用培训教程

(初级版)

刘天华 刘 杰 刘俊明 马佳琳 编著

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



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

高等学校公共课计算机规划教材

Java 实用培训教程

(初级版)

刘天华 刘 杰

刘俊明 马佳琳



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书作为 Java 认证培训系列教材的基础篇,系统、全面地阐述了 Java 语言和面向对象的程序设计方法。本书内容比较全面,例题丰富,讲述简明清晰。书中的概念和语法都以丰富的例题为背景来讲解,易读易懂,实用性较强。

本书面向参加 Sun 认证 Java 程序员考试的读者,是获取 SCJP 认证的首选辅导教材。也适用于 Java 语言初、中级水平甚至是零基础的读者。对于 Java 程序员也具有较好的参考价值,也可作为大、专院校计算机专业的教材和参考资料。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 实用培训教程:初级版 / 刘天华等编著. —北京:电子工业出版社, 2008.1

高等学校公共课计算机规划教材

ISBN 978-7-121-05706-9

I. J… II. 刘… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 202629 号

策划编辑: 许菊芳

责任编辑: 周宏敏

印 刷: 北京市通州大中印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 18 字数: 403 千字

印 次: 2008 年 1 月第 1 次印刷

定 价: 26.80 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话: (010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010)88258888。

前言

Java 是 Sun 公司推出的新一代面向对象程序设计语言,自 1995 年诞生以来,已经从一种编程语言逐渐发展为一个平台、一个社群、一个产业。Java 语言的支持网络应用编程、可跨平台使用、安全性好、支持线程功能等特点,使它成为非常适合网络应用程序开发的一种程序设计语言。

Java 语言的使用是免费的,全世界许多计算机专家、机构、公司、大学等都在为 Java 的发展在自己的领域出谋划策,而这一切都源自于 Sun 公司天才的构想和富有激情的创造。如今,众多的学子和计算机高手逐渐被 Java 的魅力所感染,成为 Java 技术最狂热的追星族。

Java 以其独有的开放性、跨平台性和面向网络的交互性席卷全球,以其安全性、易用性和开发周期短的特点,迅速从最初的编程语言发展成为全球第二大软件开发平台。这些优点已引起国内外计算机界的极大关注,Java 公用规范(PAS)已被国际标准化组织(ISO)认定,Java 技术已列为当今世界信息技术三大要点之一。特别是针对最先进的科技,推出了 Java 及 Solaris 的认证方案。根据这些认证,在企业方面可以借此作为招聘人才的评判标准,或是作为衡量员工技术水准的依据;在技术方面,通过这些认证也可以证明个人的技术能力。因此,早日掌握 Java 技术对每个有志于在 IT 行业发展的人来说是尤为重要的。我们深信 Java 技术的应用和普及必将成为势不可挡的潮流迅速在世界各地发展。Sun Java 认证是业界唯一经 Sun 授权的 Java 认证,考试内容涉及所有 Java 的相关知识、编程概念及 applet 开发技巧。Sun 认证 Java 程序员考试旨在考查考生通过 Java 开发工具进行复杂编程的能力。中国地区的 Java 认证考试有两个版本,即英文版和中文版。但是,中文版的 Java 认证不能够全球通用,只适合在中国地区使用,所以,绝大多数考生希望能够取得英文版的 Java 认证。市场上也有一些相关的英文版辅导教材,但由于受英语水平所限,大多数考生在阅读英文版教材时存在着诸多障碍,希望先通过中文版教材全面掌握 Java 技术内容,然后再阅读英文原版教材以学习更丰富的技术内容。这样,可以加强对英文版的 Java 知识点的理解能力,有利于通过英文版的 Java 认证考试。

本书的编者长期从事 Java 的开发和培训工作,编者将多年的教学经验和授课精华展示在本书中与读者共享,其目的是为广大考生提供一本中文版的、针对性强的 Java 认证考试辅导教材。

本书的内容从简单的语法分析到面向对象编程，涵盖了Sun Java 程序员的考试要求。同时也有许多编者的个人见解，在一些讲解重要概念的章节后都附有大量针对性很强的练习题。

本书针对准备参加 Sun 认证 Java 程序员考试的读者，是获取 SCJP 认证的首选辅导教材，适用于 Java 语言初、中级水平甚至是零基础的读者，对于 Java 程序员也具有较好的参考价值。

本书由刘天华、刘杰、刘俊明、马佳琳主持编著，赵志刚、陈泉、李强、杨睿参加了部分章节的编写和程序调试工作。

由于编者教学任务繁重且水平有限，加之时间紧迫，书中难免存在错误和不妥之处，诚挚欢迎广大读者批评指正。编者联系邮箱：liujunming@sict.ac.cn。另外，本书配有电子课件及程序源代码，读者可到华信教育资源网(<http://www.hxedu.com.cn>)查阅下载。

编者

2007年12月

目 录

第1章 Java 语言简介	1
1.1 Java 语言的发展	1
1.2 Java 语言的特点	1
1.2.1 简单性	2
1.2.2 面向对象	2
1.2.3 分布性	4
1.2.4 纠错性	4
1.2.5 安全性	4
1.2.6 体系结构中立	5
1.2.7 可移植性	6
1.2.8 解释执行	7
1.2.9 高性能	7
1.2.10 多线程	7
1.2.11 动态性	8
1.3 Java 和 C++ 的比较	8
1.4 Java 运行时环境	10
1.4.1 Java 运行时环境简介	10
1.4.2 JDK 简介	10
1.4.3 Java 运行时环境的安装和配置	11
1.5 Java 语言的用途	13
1.6 Java 程序简介	14
1.6.1 一个简单的 Java 应用程序	14
1.6.2 将应用程序编写为 applet 小程序	16
小结	18
第2章 面向对象	19
2.1 面向对象的概念	19
2.2 传统开发方法存在的问题	19
2.3 面向对象的基本概念	20

2.3.1	面向对象的特征	21
2.3.2	面向对象的要素	21
2.3.3	面向对象的开发方法	22
2.3.4	面向对象的模型	23
2.3.5	面向对象的分析	26
2.4	面向对象的设计	28
2.4.1	面向对象设计的准则	28
2.4.2	面向对象设计的启发规则	29
2.4.3	系统设计	30
2.5	面向对象的实现	31
2.5.1	设计语言	31
2.5.2	类的实现	31
2.5.3	应用系统的实现	32
2.5.4	面向对象的测试	32
	小结	32
第3章	Java 程序设计基础	33
3.1	Java 程序的基本结构	33
3.2	标识符命名	34
3.3	数据类型与常量	35
3.3.1	数据类型	35
3.3.2	常量	36
3.4	变量	39
3.4.1	数值型变量	40
3.4.2	字符型变量	41
3.4.3	布尔型变量	41
3.5	表达式与运算符	41
3.5.1	什么是表达式	41
3.5.2	赋值运算符	42
3.5.3	算术运算符	42
3.5.4	关系运算符和逻辑运算符	44
3.5.5	位运算符	46
3.5.6	条件运算符	48
3.5.7	浮点运算符	48
3.5.8	数组运算符	49

3.5.9	字符串运算符	50
3.6	类型转换	50
3.7	属性定义	51
3.7.1	成员属性	51
3.7.2	局部属性	52
	小结	54
	练习	54
	练习答案	55
第4章	程序流程控制	57
4.1	内容概述	57
4.2	程序的3种基本结构	57
4.2.1	顺序结构程序	57
4.2.2	选择结构程序	57
4.2.3	循环结构程序	58
4.3	条件语句	58
4.3.1	if 语句	59
4.3.2	条件语句的使用	60
4.4	循环语句	61
4.4.1	while 循环语句	61
4.4.2	构造循环结构的 do-while 语句	62
4.4.3	for 循环语句	63
4.4.4	3种循环语句的比较	64
4.5	switch 开关语句	65
4.6	转移语句	66
4.6.1	break 语句	66
4.6.2	continue 语句	67
4.6.3	return 语句	69
	小结	69
	练习	69
	练习答案	74
第5章	数组	76
5.1	数组	76
5.1.1	创建和使用一维数组	76
5.1.2	多维数组	79

02	小结	83
02	练习	83
12	练习答案	84
12	第6章 类设计	85
52	6.1 类与对象	85
12	6.2 类中的高级应用	88
12	6.2.1 static 关键字	88
22	6.2.2 this 关键字	89
52	6.2.3 get、set 方法和 JavaBean 技术	91
72	6.3 封装类	92
72	6.4 继承	94
72	6.4.1 基类和派生类	95
72	6.4.2 super 关键字	97
82	6.4.3 继承中的构造器	98
82	6.4.4 包的概念与环境变量的设置	100
92	6.4.5 范围限定修饰符	102
00	6.5 多态性	103
10	6.5.1 方法重载与重写	103
10	6.5.2 抽象类和抽象方法	106
50	6.5.3 final 类和 final 方法	108
60	6.5.4 对象造型	108
10	6.6 Object 类	109
20	6.6.1 toString() 方法	109
00	6.6.2 equals() 方法	110
00	6.7 接口	111
70	小结	112
00	练习	113
00	练习答案	121
45	第7章 异常处理	123
05	7.1 异常的基本概念	123
05	7.1.1 什么是异常	123
05	7.1.2 Java 标准异常	123
05	7.1.3 Java 异常处理机制	124
05	7.2 捕获异常	124

7.3	创建自定义异常	128
7.4	异常的尴尬	133
7.5	使用异常的注意事项	137
7.5.1	异常匹配	137
小结		139
练习		139
练习答案		142
第8章	输入与输出	143
8.1	Java 输入/输出类和接口	143
8.2	File (文件类)	144
8.2.1	目录	146
8.2.2	使用 FilenameFilter	147
8.2.3	listFiles()方法	148
8.2.4	创建目录	149
8.3	流类	149
8.4	字节流类	149
8.4.1	InputStream (输入流)	150
8.4.2	OutputStream (输出流)	150
8.4.3	FileInputStream (文件输入流)	150
8.4.4	FileOutputStream (文件输出流)	153
8.4.5	ByteArrayInputStream (字节数组输入流)	154
8.4.6	ByteArrayOutputStream (字节数组输出流)	155
8.4.7	过滤字节流	157
8.4.8	缓冲字节流	157
8.4.9	SequenceInputStream (顺序输入流)	161
8.4.10	PrintStream (打印流)	162
8.4.11	RandomAccessFile (随机访问文件类)	162
8.5	字符流	163
8.5.1	Reader	163
8.5.2	Writer	164
8.5.3	FileReader	164
8.5.4	FileWriter	165
8.5.5	CharArrayReader	166
8.5.6	CharArrayWriter	167

8.5.7	BufferedReader	168
8.5.8	BufferedWriter	169
8.5.9	PushbackReader	170
8.5.10	PrintWriter	171
8.6	使用流式输入/输出	171
8.6.1	用 StreamTokenizer (流标记) 来改善 wc()	173
8.7	序列化	175
8.7.1	Serializable 接口	176
8.7.2	Externalizable 接口	176
8.7.3	ObjectOutput 接口	176
8.7.4	ObjectOutputStream 类	177
8.7.5	ObjectInput	177
8.7.6	ObjectInputStream	178
8.7.7	序列化示例	179
	小结	181
	练习	181
	练习答案	183
第9章	图形界面	184
9.1	用 AWT 生成图形化用户界面	184
9.1.1	java.awt 包	184
9.1.2	组件和容器	185
9.1.3	常用容器	186
9.1.4	LayoutManager 布局管理器	187
9.2	AWT 事件处理模型	193
9.2.1	事件类	195
9.2.2	事件监听器	196
9.2.3	AWT 事件及其相应的监听器接口	197
9.2.4	事件适配器	200
9.3	AWT 组件库	203
	小结	208
	练习	209
	练习答案	210
第10章	Swing	211
10.1	Swing 简介	211

10.1.1	概述	211
10.1.2	Swing 的类层次结构	211
10.1.3	Swing 组件的多样化	213
10.1.4	MVC 体系结构	213
10.1.5	可存取性支持	213
10.1.6	支持键盘操作	213
10.1.7	设置边框	214
10.1.8	使用图标 (Icon)	214
10.1.9	Swing 程序结构简介	215
10.2	Swing 组件和容器	216
10.2.1	组件的分类	216
10.2.2	使用 Swing 的基本规则	217
10.2.3	各种容器面板和组件	217
10.2.4	布局管理器	227
	小结	229
第 11 章	applet	230
11.1	创建一个 applet	230
11.1.1	applet 开发步骤	230
11.1.2	Applet 类	232
11.2	构建 applet 应用	235
11.2.1	使用 applet 响应事件	235
11.2.2	使用 applet 画图	237
11.2.3	使用 applet 构建多媒体应用	240
	小结	249
第 12 章	多线程	250
12.1	使用 Java 创建多线程	251
12.1.1	Thread 类	251
12.1.2	Runnable 接口	252
12.1.3	synchronized 关键字	253
12.2	线程调度	254
12.3	Java 编程语言中的高级多线程支持	255
12.3.1	线程组	255
12.3.2	线程间的发信	255
12.3.3	将 CPU 让给另一个线程	256

11.5	12.3.4 守护线程	256
11.5	12.4 不提倡使用的方法	257
8.1.5	12.4.1 死锁	257
8.1.5	12.4.2 活锁	257
8.1.5	12.4.3 资源耗尽	257
8.1.5	12.5 调试大量的线程	257
4.1.5	12.6 限制线程优先级和调度	258
4.1.5	小结	258
2.1.5	练习	259
0.1.5	练习答案	262
2.1.5	第 13 章 Java Socket 编程	263
7.1.5	13.1 Socket 简介	263
7.1.5	13.2 Socket 传输模式	263
7.2.5	13.3 创建 Socket 应用	264
0.2.5	13.3.1 创建客户端	264
0.3.0	13.3.2 创建服务器	268
0.3.0	13.3.3 Java 数据报类	269
0.3.0	13.3.4 简单的 Web 服务器	272
0.3.5	小结	274
0.3.5	参考文献	275

第1章 Java 语言简介

Java 是一种适合于分布式计算的面向对象的新型程序设计语言, Java 将面向对象、平台无关性、稳固性与安全性、多线程等诸多特性集于一身, 为用户提供了一个良好的程序设计环境。Java 是 C++ 的衍生语言, 它从 C++ 中继承了大量的语言成分, 但摒弃了 C++ 中冗余和容易引起问题的功能。本章介绍 Java 语言的发展历程、Java 语言的特性以及 Java 语言的运行环境等内容。

1.1 Java 语言的发展

Java 语言的前身称为 Oak 语言, 它是由美国 Sun MicroSystem 公司于 1991 年开发的一个称为“Green”的软件项目, 该项目的本意是开发一种用于消费类电子产品的与平台无关的软件技术。从那以后, Oak 语言一直被认为是用来开发消费类电子产品和交互式电视控制器的工具。1994 年, Sun 分公司的两名开发人员在 Oak 的基础上创建了 HotJava 的第一个版本, 当时称为 Webrunner, 即在 Web 上使用的图形浏览器, 经过一段时间后才称为 Java。1995 年 5 月, Sun 公司对外正式发布了 Java 语言, 随后立即得到了各 WWW 厂商的大力支持, 纷纷在浏览器上加入 Applet 小程序 (即用 Java 语言编写的小应用程序), 并通过 Internet 在世界各地进行传播。

实际上, Java 语言是和 Internet 同步发展起来的一种新型网络语言, 是近 20 年来计算机软件环境中最有意义的进步之一。Java 语言在网络中的地位同超文本标记语言 HTML (HyperText Makeup Language) 一样重要。

Java 语言是一种强有力的网络编程语言, 它最大限度地利用了网络资源。applet 小程序可以跨平台、跨操作系统, 跨网络运行, applet 代码小, 易于在网络上快速下载和发送, 且具有不需要修改应用程序就可以增加 Web 页的新功能。Java 还为编程人员提供了许多公用的系统接口。随着 Internet 在全世界范围的广泛流行以及在各个领域的渗透, Java 语言已被各行各业的人士所接受。

1.2 Java 语言的特点

Java 语言是一种跨平台的、适合于分布式计算环境的网络编程语言, 它简单, 面向

对象，不依赖于机器的结构，不受 CPU 和环境的限制，具有可移植性、鲁棒性、安全性，并且提供了并发机制，具有很高的性能。其次，Java 语言提供了丰富的类库，使程序设计人员可以方便地建立自己的系统。

1.2.1 简单性

Java 语言是一种面向对象的语言，语法规则与 C 语言类似，它通过提供最基本的方法来完成指定的任务，用户只需理解一些基本概念，就可以用它编写出适合于各种情况的应用程序。Java 语言略去了指针、运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集功能大大简化了程序设计者的内存管理工作。另外，Java 语言对环境的要求非常低，其基本解释器及类的支持只有几千字节，加上标准类库和线程的支持也只有 200 多千字节。Java 语言的简单性主要归因于如下几个因素：

- Java 的风格类似于 C++，因而对 C++ 程序员而言是非常熟悉的；从某种意义上讲，Java 语言本身是 C 语言及 C++ 语言的一个变种，因此，C++ 程序员可以很快掌握 Java 编程技术。
- Java 摒弃了 C++ 中容易引发程序错误的地方，如指针和内存管理。
- Java 提供了自动内存垃圾收集机制，从而减轻了编程人员进行内存管理的负担，有助于减少软件错误。
- Java 是完全面向对象的，它是最容易学习的面向对象的编程语言之一，同时它还提供了大量可重用的类库。

Java 的简单性是以增加运行时系统的复杂性为代价的。以内存管理为例，自动内存垃圾处理减轻了面向对象编程的负担，但 Java 运行时系统却必须内嵌一个内存管理模块。然而无论如何，对编程人员而言，Java 的简单性只会是一个优点，它可以使我们的学习曲线更趋合理化，加快我们的开发进度，减小程序出错的可能性。

1.2.2 面向对象

面向对象是 Java 最重要的特性。Java 语言的设计是完全面向对象的，它不支持类似 C 语言那样的面向过程的程序设计技术。Java 支持静态和动态风格的代码继承及重用。单从面向对象的特性来看，Java 类似于 SmallTalk，但它的其他特性（尤其是适用于分布式计算机环境的特性）却远远超过 SmallTalk。

“面向对象”目前是一个非常流行的术语，其影响领域从操作系统、编程语言及其开发环境、数据库管理系统直至软件工程。可以说，无论合适与否，也不管是否真正具有面向对象的特性，人们已经习惯了给自己的产品贴上“面向对象”的标签。但事实上，这些产品有许多并不是真正面向对象的，或者说不是完全面向对象的。

何谓“面向对象”？面向对象其实是现实世界模型的一个自然延伸。现实世界中的任何实体都可以看成对象。对象之间通过消息相互作用。传统过程式编程语言支持以下公式：

程序 = 算法 + 数据

而面向对象的编程语言也有一个公式，如下所示：

程序 = 对象 + 消息

所有面向对象的编程语言均支持 3 个概念：封装、多态性和继承。现实世界中的对象均具有属性和行为，将其映射到计算机程序中，属性表示数据，行为代表程序代码。所谓封装，就是用一个自主式框架把代码和数据联编在一起，形成一个对象。也就是说，对象是支持封装的手段，是封装的基本单位。对象内的数据和代码可以是公有的，也可以是私有的。私有代码和数据只能被本类内部访问，公有代码和数据则可被其他对象访问。一般情况下，对象的公用部分是对象之间交互的机制。

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及多态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法实现代码的复用。Java 摒弃了 C++ 中的非面向对象特性，如结构和函数调用，Java 也不再支持全局变量。

下面是一个完整的 Java 程序：

```
Public class HelloJava {  
    Public Static void main ( string argv[] ) {  
        System.out.println("Hello JAVA!");  
    }  
}
```

该程序由一个类定义组成。在 Java 中，如果不创建新类就无法创建程序，Java 程序在运行时必须先创建一个类的实例，然后才能提交运行。

Java 同样支持继承特性，Java 的类可以从其他类中继承行为，但 Java 只支持类的单重继承，即每个类只能从一个类中继承。其他一些面向对象的语言（如 C++）支持多重继承，从而有可能引起混乱并使语言变得复杂。

在 Java 中像数值和布尔这种简单的数据类型并不是对象，这是为了使 Java 语言更简单、更高效，但 Java 提供了所有简单类型包装对象，因此这些简单的类型也能被实现为对象。

Java 支持界面，界面允许程序员定义方法但又不立即实现。一个类可以实现多个界面，利用可以得到多重继承的许多优点而又没有多重继承的问题。Java 的界面十分类似于 IDL（Interface Definition Language）界面，构造 IDL 到 Java 的编译器是十分方便的，这就意味着可以在 CORBA 对象系统中使用 Java 来建立分布对象系统。这种兼容性是十分重要的，因为 IDL 界面和 CORBA 对象系统已被广泛使用。

1.2.3 分布性

分布性包括数据分布和操作分布。数据分布是指数据可以分散存放于网络上的不同主机中，操作分布则指把计算分散并由不同主机进行处理。Java 支持 WWW 客户/服务器计算模式，因此，它可以支持所有这两种分布性。

对于数据分布，Java 提供了一个 URL 对象，利用此对象可以打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。对于操作分布，Java 的客户/服务器模式可以把运算从服务器分散到客户一端，从而提高整个系统的执行效率，避免瓶颈制约，增加动态可扩充性。对于编程人员来说，Java 的网络类库是对分布编程的最好支持。Java 网络类库是支持 TCP/IP 协议的子例程序库。目前支持的协议有 HTTP 和 FTP 等。同时，通过编写协议句柄，编程人员还可以扩充 Java 支持的协议集合。

1.2.4 纠错性

Java 语言在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。它提供自动垃圾收集来进行内存管理，防止程序在管理内存时容易产生的错误。通过集成的面向对象的异常处理机制，在编译时，Java 语言提示出可能出现但未被处理的异常，帮助程序员正确地进行选择以防止系统崩溃。另外，Java 语言在编译时还可以捕获类型声明中的许多常见错误，防止动态运行时由于不匹配而出现问题。

1.2.5 安全性

在分布式环境中，安全性是一个十分重要的问题。Java 为了适合于分布式应用的开发，在语言设计过程中考虑了安全性问题，在 Java 语言和运行环境中引入了多级安全措施，其采用的主要安全机制如下。

1. 内存分配及布局由 Java 运行时系统决定

Java 编译器为安全性而提供的主要防御手段是它的内存分配和引用模型。首先，内存布局并不是像 C 和 C++ 那样由 Java 编译决定，而是延迟到运行时由 Java 运行时系统决定，内存布局依赖于 Java 运行时系统所在软/硬件平台的特性；其次，Java 并没有 C 和 C++ 那样传统意义上的内存单元指针，Java 编译器是通过符号指针来引用内存的，符号指针由 Java 运行时系统在运行时具体解释为实际的内存地址，Java 程序员不能强制引用内存指针，因此 Java 的内存分配和引用模型对于程序员是透明的，它完全由底层的运行时系统控制，程序员不能通过修改代码而直接指向一个物理内存布局。通过放弃 C 和 C++ 的内