

◎ 计算机

“卓越工程师计划”应用型教材



Computer



Java 程序设计 实践教程

Java Programming Tutorial Practice

◎ 张永常 胡局新 主编
◎ 康晓凤 杨 磊 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

计算机“卓越工程师计划”应用型教材

Java 程序设计实践教程

张永常 胡局新 主 编

康晓凤 杨 磊 副主编

電子工業出版社

Publishing House of Electronics Industry

内 容 简 介

本教程从 Java 的入门知识开始进行介绍,力图让读者通过 13 章的学习和实践,由浅入深、由点到面、逐步提高,掌握 Java 程序设计技术。这 13 章介绍的内容分别是 Java 入门、Java 编程基础、数组和字符串、面向对象程序设计基础、面向对象程序设计进阶、常用类库和基本类型包装器类、泛型与集合、异常处理、图形用户界面、多线程编程、数据库操作、I/O 流与文件处理、综合项目实践。本教程共计有 134 个案例,111 个照猫画虎实战训练项目,80 个思考题,49 个牛刀初试项目,6 个创新挑战的综合性实践项目。

本书可作为高等院校学生学习 Java 基础的教材,非计算机专业读者学习 Java 语言、Java 技术培训的教材和教学参考书,也可作为没有任何编程基础知识的读者或 Java 爱好者的学习参考书。

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 程序设计实践教程 / 张永常, 胡局新主编. —北京: 电子工业出版社, 2013.8

计算机“卓越工程师计划”应用型教材

ISBN 978-7-121-20478-4

I. ①J… II. ①张… ②胡… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 106166 号

责任编辑: 刘海艳

印 刷: 北京京师印务有限公司

装 订: 北京京师印务有限公司

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: 787×1 092 1/16 印张: 21.75 字数: 628.5 千字

印 次: 2013 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 39.80 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。

丛 书 序 言

党的十八大提出要“努力办好人民满意的教育”，要“推动高等教育内涵式发展”，“全面实施素质教育，深化教育领域综合改革，着力提高教育质量，培养学生社会责任感、创新精神、实践能力。”这对高等教育提出了新的要求，明确了人才培养的目标和标准。

十八大明确指出“坚持走中国特色新型工业化、信息化、城镇化、农业现代化道路，推动信息化和工业化深度融合、工业化和城镇化良性互动、城镇化和农业现代化相互协调，促进工业化、信息化、城镇化、农业现代化同步发展。”“推动信息化和工业化深度融合”对高等工程教育改革提出了迫切要求。

遵照《国家中长期教育改革和发展规划纲要（2010—2020 年）》和《国家中长期人才发展规划纲要（2010—2020 年）》，为贯彻落实教育部“卓越工程师教育培养计划”，促进我国由工程教育大国迈向工程教育强国，培养造就一批创新能力强、适应经济社会发展需要的高质量计算机工程技术人才，电子工业出版社决定组织相关实施和计划实施卓越计划以及江浙两省实施软件服务外包人才培养试点的地方高校的相关教师，在以往实践校企合作人才培养的基础上编写一套适合地方高校的计算机“卓越工程师计划”人才培养系列教材。

我们将秉承“行业指导、校企合作、分类实施、形式多样”的“卓越工程师教育培养计划”四原则，坚持“学科规范、本科平台、行业应用”，以“具备较为扎实的专业基础知识、拥有良好的职业道德素质、具有创新的计算机应用能力”为目标，探索“校企一体化”产学研结合人才培养模式改革，强化“岗位目标、职业培养”，努力实现计算机工程型技术人才（应用型）培养目标：

（1）尝试以“知识保障、能力渐进、素质为本，重视技术应用能力培养为主线”，坚持以“素质教育，能力培养”为导向，体现本科平台、能力定位、应用背景构建课程体系。

（2）尝试“以学生工程意识、创新精神和工程实践能力培养”为核心，坚持以“培养学生的工程化开发能力和职业素质”为原则，校企合作构建实践教学体系。

本系列教材基于“以德为先、能力为重、全面发展”的人才培养观念，在内容选择、知识点覆盖、课程体系安排、实践环节构建、企业强化训练上按照能力培养和满足职业需求为本进行了有益的、初步的探索。

然而，由于社会对计算机人才的需求广泛而多样，各领域的人才规格和标准既有共性又有特殊性，同时各相关高校在计算机相关专业设置以及人才培养的探索上各有特点，我们编写的本套系列教材目前只能部分满足计算机相关专业人才培养的需要。我们力争建立一个体系，以模块构建的增量方式实现教材编写的滚动、增加和淘汰，逐步建设可供地方高校计算机不同专业、针对不同领域培养计算机工程技术人才选择的教材库：①所有专业的公共基础课相对统一，不同专业的专业基础课按模块划分、各自专业的专业课按领域整合、拓展课紧跟技术和行业发展；②公共基础课、专业基础课以经典知识为主，专业课、拓展课与国际主流技术接轨；③实践环节或实践课程必须接纳企业文化、优选企业实际工程项目，体现校企合作、重视企业导师的参与。

“卓越工程师教育培养计划”的实施具有三个特点：一是行业企业深度参与培养过程；二是学校按通用标准和行业标准培养工程人才；三是强化培养学生的工程能力和创新能力。

本系列教材的编写得到了中软国际、苏州花桥国际商务区（及所属企业）、常州创意产业

基地（及所属企业）等热心和关注计算机类人才培养的国家重点企业、园区的大力支持。我们曾以“目标明确、责任共担、实现共赢”为原则探索了多种人才培养合作途经：从师资培养到校企共建实训基地，到建立校内软件学院，再到学生进企业强化、顶岗实训……取得了一定的经验。在“卓越工程师教育培养计划”的实施中，企业和学校签订了全面合作协议，共同确定人才培养标准、制订人才培养方案、参与人才培养过程，提供企业学习课程和项目案例，确保学生在企业的学习时间。

同样，本系列教材的编写总结了参编高校和支撑企业在校企合作人才培养过程中共同取得的经验和教训，并涵盖了我们已经做的、想要做的实施卓越计划的理念和努力。这仅是初步的尝试，会存在许多不足和缺陷，但希望由此能起到抛砖引玉的作用。在卓越计划的实施探索中，我们衷心地希望能有更多的地方高校计算机院系、更多的行业企业加入团队，面对企业必须参与的国际化产业竞争，为培养优秀的、具有应用创新精神的计算机工程技术（包括软件）人才，企业和学校能深度合作、各尽职责；每一位教育工作者都能贡献自己的聪明才智，尽一份绵薄之力。

对给予本套丛书编审大力支持的江苏计算机学会、中国矿业大学计算机学院以及参与编写教材的高校、单位表示由衷的感谢！

计算机“卓越工程师计划”应用型教材编委会

前 言

在 2012 年的编程语言排行榜上, Java 语言仍然名列前茅, 对 Java 软件工程师的需求量依然很大。可以看到, 从 2002 年以来, Java 一直是稳居第一的, 说明了 Java 的王者地位。为什么说 Java 是王者呢? 它又有何神奇之处呢?

原因很简单, Java 作为网络时代的语言, 主要靠的是跨平台性和安全性, Java 是现在大型软件项目中的主角, 市场用人需求量大。进入信息时代后, 信息科技给人类的生产、生活、娱乐等方式带来了深刻的变革, 信息产业已成为推动国家经济发展的主导产业之一; 在《中华人民共和国国民经济和社会发展第十二个五年规划纲要》中, 软件产业被提到了一个空前的战略高度, 软件产业的飞速发展导致了这个行业的人才需求与日俱增。

由于 Java 的跨平台特性比较突出, 企业对 Java、Java ME、Java EE 开发有着巨大的需求, 随着高等教育大众化, 就业竞争的形势日趋激烈, 计算机专业及想在 IT 行业就业的学生学习 Java 语言的热情也随之不断高涨。有些学校原来把 Java 程序设计这门课程放在专业选修课中让学生选修, 但几乎所有的学生都选修, 成为事实上的必修课, 所以学校也就适应市场需求, 把这门课程改为必修课程。

大家都已注意到, 教育部考试中心已经把“Java 语言程序设计”列入了全国计算机等级考试项目。全国计算机等级考试(National Computer Rank Examination, NCRE), 是经原国家教育委员会(现教育部)批准, 由教育部考试中心主办, 面向社会, 用于考查应试人员计算机应用知识与技能的全国性计算机水平考试体系。把“Java 语言程序设计”列入全国计算机等级考试, 从另一方面说明了 Java 课程在学生就业方面的重要作用。

由于高等教育大众化和市场经济对人才需求的不断深化, 企业往往对求职者——高校的应届毕业生提出了直接上岗的能力要求, 但是刚刚从高校毕业的学生往往缺乏工作经历, 而企业招聘人才时对能够直接上岗的 Java 程序员的需求却是巨大的, 这就催生了一些培训企业, 这些企业专门做高校毕业生毕业后到企业上岗前的这一段空间的培训工作, 这些培训的企业开发了一些成功的项目案例, 让刚刚毕业或尚未毕业的学生真刀真枪地进行项目训练, 在这些企业培训过后, 受训者的项目能力得到了较大幅度地提高, 用人单位就愿意接收他们。这些培训企业往往对毕业生提出应该有 Java 基础知识和基本实践能力的要求。普通高校培养出来的大学生还有些青涩, 经过培训企业的催熟, 使得学生就业变得顺利起来, 当然, 这些培训不是免费的午餐。

本实践教程的编写目标是, 针对卓越工程师教育培养计划的实施, 强化该课程培养学生实践能力的特色, 强调学生在软件开发实践活动中的直接上岗能力和创新精神的培养; 力求突出“坚持理论根本, 突出项目实践”的理念, 让读者跟随教程中案例分析, 掌握基础知识和基本能力; 在此基础上, 利用照猫画虎、牛刀初试、创新挑战三个阶段的实践活动, 使读者受到软件工程师基本实践能力的锻炼。本教程共计有 134 个案例、111 个照猫画虎实战训练项目、80 个思考题、49 个牛刀初试项目、6 个创新挑战的综合性实践项目。在编写过程中, 作者团队力图努力打造内容全面、示例丰富、深入浅出、通俗易懂、培养兴趣、注重实践的特色。

本教程中用到的全部程序代码已在 JDK1.6.30 环境中调试运行通过。虽然我们提供了本教程的所有案例的源代码电子稿, 但是我们仍然建议初学 Java 语言的读者逐个输入各程序的源代码。这样做的最大优点是可以培养严谨的学风, 熟悉 Java 的程序结构, 便于记忆和熟练地运用 Java 中的类、方法等, 而且输入和调试的过程也是学习的过程。

使用本实践教程的教师或读者若需要本教程的源程序、PPT 时, 可与电子工业出版社刘海艳

编辑 (lhy@phei.com.cn) 联系索取或登录到华信教育网下载。

本教程由张永常、胡局新任主编,张永常完成全部书稿的统稿工作。其中第 1~5 章由江苏师范大学张永常编写,第 6、8、9 章由徐州工程学院康晓凤编写,第 7、10、11、12 章由徐州工程学院胡局新编写,第 13 章由张永常、康晓凤共同编写,徐州工程学院张旭隆、曹言敬、徐海棠,中国矿业大学徐海学院杨磊完成了编写过程中程序的编辑、调试和校对工作。

本教程的编写工作是在江苏师范大学、徐州工程学院、徐海学院的领导和同事们的大力支持下、鼓励与帮助下,在电子工业出版社的鼎力扶助和指导下完成的,我们诚挚地向他们表示衷心的感谢!

再次感谢全体作者的家庭成员给予的支持!

由于作者教学任务繁重且水平有限,加之时间紧迫,对于书中存在的错误和不妥之处,诚挚欢迎批评指正,作者联系邮箱:yczhang@jsnu.edu.cn。

张永常

目 录

第 1 章 Java 入门	1	2.2.4 字符类型 (char)	39
1.1 Java 有何特色	1	2.2.5 数据类型的转换	40
1.2 学习 Java 有何用途	3	2.3 常量和变量	42
1.3 如何学习 Java	4	2.3.1 常量	42
1.4 怎样搭建 Java 运行环境	6	2.3.2 变量	43
1.4.1 下载 JDK	6	2.3.3 变量的作用域	44
1.4.2 安装 JDK	7	2.4 接收用户的输入和程序输出	45
1.4.3 设置环境变量	9	2.4.1 输入单个字符	46
1.5 进行 Java 开发需要哪些工具	10	2.4.2 输入多个字符	47
1.5.1 UltraEdit	11	2.4.3 使用 BufferedReader 类 输入数值型数据	48
1.5.2 Eclipse	12	2.4.4 使用图形方式输入数据	48
1.6 Java 程序是什么样子	13	2.4.5 使用 Scanner 类 输入数据	50
1.6.1 Java Application 程序	13	2.5 运算符与表达式	51
1.6.2 Java Applet 程序	15	2.5.1 赋值运算符和语句	52
1.6.3 在 Eclipse 中编写和运行 程序	18	2.5.2 算术运算符和算术表达式	53
1.6.4 编码规范概述	21	2.5.3 关系运算符和关系表达式	54
1.7 编写程序的关键是什么——算法	29	2.5.4 逻辑运算符和逻辑表达式	55
1.7.1 算法及其特性	29	2.5.5 位运算符	56
1.7.2 传统流程图	30	2.5.6 条件赋值运算符	58
1.7.3 用传统流程图描述 算法举例	31	2.5.7 括号与方括号运算符	58
1.7.4 N-S 流程图及其应用举例	32	2.5.8 广义赋值运算符	58
1.8 如何获得 Java 帮助	33	2.5.9 对象运算符	59
1.8.1 下载帮助文档	33	2.5.10 运算符的优先级	59
1.8.2 使用帮助文档	34	2.6 程序的流程控制	60
1.9 思考与实践	34	2.6.1 顺序结构	60
1.9.1 实训目的	34	2.6.2 分支结构	61
1.9.2 实训内容	35	2.6.3 循环结构	67
第 2 章 Java 编程基础	36	2.6.4 转移语句	71
2.1 标识符与关键字	36	2.6.5 综合实践	73
2.1.1 关键字	37	2.7 思考与实践	75
2.1.2 标识符	37	2.7.1 实训目的	75
2.2 基本数据类型	38	2.7.2 实训内容	76
2.2.1 整数类型 (int、short、 long、byte)	38	第 3 章 数组和字符串	77
2.2.2 浮点类型 (float、 double)	39	3.1 数组的概念	77
2.2.3 布尔类型 (boolean)	39	3.2 一维数组	77
		3.2.1 一维数组的声明与创建	77
		3.2.2 一维数组的初始化	78

3.2.3 一维数组的使用	78	5.3.1 super	143
3.3 二维数组	82	5.3.2 this	143
3.3.1 二维数组的声明与创建	82	5.4 类的多态性	147
3.3.2 二维数组的初始化	83	5.5 抽象类	151
3.3.3 二维数组的使用	84	5.5.1 抽象类的概念	151
3.4 多维数组	87	5.5.2 创建抽象类	151
3.5 Java 新特性对数组的支持	88	5.5.3 抽象方法	151
3.6 数组操作	89	5.6 接口	161
3.7 字符串类	94	5.6.1 接口的定义	161
3.7.1 String 类	94	5.6.2 接口的实现	162
3.7.2 StringBuffer 类	99	5.6.3 接口的继承	171
3.7.3 字符串数组	105	5.7 内部类	174
3.8 思考与实践	106	5.7.1 实名内部类	174
3.8.1 实训目的	106	5.7.2 匿名内部类	175
3.8.2 实训内容	106	5.8 包	177
第 4 章 面向对象程序设计基础	108	5.8.1 包的创建	178
4.1 类与对象	108	5.8.2 包的导入	178
4.1.1 类的声明和创建	109	5.8.3 编译和运行包中的程序	179
4.1.2 类的实例化结果 ——对象	110	5.9 枚举类型	184
4.1.3 对象的访问	111	5.9.1 枚举类型定义	184
4.1.4 分析与设计类时的 注意事项	112	5.9.2 枚举类型使用	184
4.2 方法	113	5.9.3 案例分析	185
4.2.1 方法的创建	113	5.10 思考与实践	187
4.2.2 方法的调用	113	5.10.1 实训目的	187
4.2.3 常用访问控制修饰符	115	5.10.2 实训内容	187
4.2.4 参数传递	118	第 6 章 常用类库和基本类型	
4.2.5 方法的重载	119	包装器类	188
4.2.6 方法的递归调用	121	6.1 系统相关类——Runtime 类	188
4.2.7 数学类中的常用方法	123	6.2 常用类	191
4.3 构造方法	125	6.2.1 Object 类	191
4.4 静态方法和静态变量	130	6.2.2 Math 类	192
4.5 思考与实践	133	6.2.3 BigInteger 类	192
4.5.1 实训目的	133	6.2.4 BigDecimal 类	194
4.5.2 实训内容	133	6.2.5 Comparable 接口	196
第 5 章 面向对象程序设计进阶	135	6.3 日期处理类	197
5.1 类的封装性	135	6.3.1 Date 类	197
5.2 类的继承性	138	6.3.2 Calendar 类	197
5.2.1 继承的概念	138	6.4 格式化类	199
5.2.2 继承的语法格式与实现	138	6.4.1 NumberFormat 数字 格式化类	199
5.3 隐藏、覆盖与 super、this 关键字	143	6.4.2 DateFormat 日期 格式化类	201
		6.5 正则表达式	201

6.6 基本类型包装器类	203	9.1.3 GUI 程序开发的一般步骤	244
6.6.1 Byte、Integer、Short、Long 类	204	9.2 顶层容器——窗口	245
6.6.2 Character 类	205	9.2.1 窗口	245
6.6.3 Float、Double 类	207	9.2.2 窗口案例	246
6.7 思考与实践	208	9.2.3 对话框	247
6.7.1 实训目的	208	9.3 常用组件	250
6.7.2 实训内容	208	9.3.1 标签 (JLabel)	250
第 7 章 泛型与集合	210	9.3.2 文本框 (JTextField)	250
7.1 泛型	210	9.3.3 密码框 (JPasswordField)	250
7.1.1 泛型类的声明和使用	210	9.3.4 按钮 (JButton)	251
7.1.2 泛型接口声明与使用	213	9.3.5 组合框 (JComboBox)	251
7.1.3 泛型方法	214	9.3.6 实践案例——登录窗体	252
7.2 集合框架概述	216	9.4 菜单和工具栏	253
7.2.1 集合的概念	216	9.4.1 菜单	253
7.2.2 Collection 接口常用方法	216	9.4.2 菜单案例	254
7.2.3 Set 接口与实现类	217	9.4.3 工具栏	256
7.2.4 List 接口和实现类	222	9.4.4 工具栏案例	256
7.2.5 Map 接口和实现类	226	9.5 布局管理器	257
7.2.6 集合工具类——Collections	230	9.5.1 流式布局 (FlowLayout)	257
7.3 思考与实践	231	9.5.2 边界布局 (BorderLayout)	258
7.3.1 实训目的	231	9.5.3 网格布局 (GridLayout)	259
7.3.2 实训内容	231	9.5.4 卡片布局 (CardLayout)	260
第 8 章 异常处理	232	9.5.5 空布局	262
8.1 异常的概念及分类	232	9.6 事件处理	262
8.1.1 异常的概念	232	9.6.1 事件处理模型	262
8.1.2 异常分类	232	9.6.2 事件处理的过程	263
8.2 异常处理	234	9.6.3 常见事件处理方法	264
8.2.1 捕获异常语句 (try-catch-finally)	234	9.7 思考与实践	269
8.2.2 抛出异常 (throw 和 throws) 语句	237	9.7.1 实训目的	270
8.3 自定义异常	239	9.7.2 实训内容	270
8.4 思考与实践	241	第 10 章 多线程编程	271
8.4.1 实训目的	241	10.1 线程概述	271
8.4.2 实训内容	241	10.1.1 多任务处理	271
第 9 章 图形用户界面	243	10.1.2 程序、进程和线程	271
9.1 Java 图形界面基础	243	10.1.3 多线程的优点	272
9.1.1 AWT 与 Swing 组件	243	10.2 线程的创建和启动	272
9.1.2 图形 API 与 Swing 组件结构	244	10.2.1 继承 Thread 类	272

10.2.2 实现 Runnable 接口	275	OutputStream)	313
10.2.3 建立线程的两种方法 的比较	277	12.2.2 文件输入/输出流 (FileInputStream/ FileOutputStream)	314
10.3 线程的生命周期	277	12.2.3 数据输入/输出流 (DataInputStream 和 DataOutputStream)	316
10.4 线程优先级及调度	278	12.2.4 标准输入/输出流 (System.in、 System.out 和 System.err)	318
10.4.1 线程优先级	278	12.3 字符流的使用	319
10.4.2 线程调度	280	12.3.1 字符输入/输出流 (Reader/Writer)	319
10.5 线程同步与互斥	284	12.3.2 字符文件输入/输出流 (FileReader/ FileWriter)	319
10.5.1 多线程访问带来 的问题	284	12.3.3 字符缓冲流 (BufferedReader/ BufferedWriter)	321
10.5.2 线程同步	285	12.3.4 转换流的使用	322
10.6 并发协作——生产者 消费者模型	287	12.4 对象流与序列化	322
10.7 思考与实践	290	12.5 Scanner 类	324
10.7.1 实训目的	290	12.6 File 类	325
10.7.2 实训内容	291	12.6.1 访问文件和目录	325
第 11 章 数据库操作	292	12.6.2 文件过滤器	327
11.1 JDBC 基础	292	12.7 思考与实践	329
11.1.1 什么是 JDBC	292	12.7.1 实训目的	329
11.1.2 JDBC 重要的类和接口	293	12.7.2 实训内容	329
11.1.3 JDBC 连接数据库的 四种方法	293	第 13 章 综合项目实践——创新挑战	330
11.1.4 JDBC 操作数据库的 一般步骤	294	13.1 综合项目实践的意义	330
11.2 JDBC 编程前期准备——学生 信息管理系统案例分析	294	13.1.1 设置本章的目的	330
11.3 数据库的连接	295	13.1.2 综合项目实践的意义——主动 迎接创新挑战	330
11.4 数据的基本操作	298	13.2 综合项目实践课题	330
11.4.1 数据查询操作	298	13.2.1 综合项目 1——科学 计算器	330
11.4.2 数据更新操作	304	13.2.2 综合项目 2——万年历	331
11.5 事务处理	306	13.2.3 综合项目 3——学生成绩管理 系统的设计与实现	332
11.5.1 事务的概念	306	13.2.4 综合项目 4——同学通讯录 管理系统的设计与实现	333
11.5.2 JDBC 的事务支持	306	13.2.5 综合项目 5——图书管理系统 的设计与实现	333
11.6 思考与实践	309	13.2.6 综合项目 6——点名考核系统 的设计与实现	334
11.6.1 实训目的	309		
11.6.2 实训内容	309		
第 12 章 I/O 流与文件处理	311		
12.1 Java I/O 流简介	311		
12.1.1 流的概念	311		
12.1.2 流的分类	311		
12.1.3 流的层次结构	312		
12.2 字节流的使用	313		
12.2.1 字节输入/输出流 (InputStream/ OutputStream)	313		

第1章

Java 入门

本章学习要点与训练目标

- ◆ 了解 Java 的特色，激发学习兴趣；
- ◆ 初步掌握学习 Java 的方法和进行实践的步骤；
- ◆ 学会搭建 Java 运行环境；
- ◆ 初步掌握绘制流程图的方法；
- ◆ 初步掌握使用 Java 帮助文档解决自己的疑问；
- ◆ 能够使用常用编辑软件，编写、编译和运行自己的 Java 程序。

有很多人听别人说：计算机专业学生掌握了 Java 开发技术就不愁没有好工作。也有人很早就说过：学习 Java 语言已经成为一种趋势、潮流。在 Java 技术推向市场后，微软公司总裁比尔·盖茨（Bill Gates）观察了一段时间之后，不无感慨地说：“Java 是长时间以来最卓越的程序设计语言”，“Java 是用来推翻我们以前所创建的东西的一种语言”。麻省理工学院计算机科学系早在 1997 年就用 Java 取代了 C++ 作为学生必须掌握的主要软件开发语言。

本章的 8 节围绕 8 个问题引领读者进入 Java 世界。

1.1 Java 有何特色

在《The Java Language: A White Paper》中是这样描述 Java 的：“Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language.” 其意思是：“Java 是简单的、面向对象的、分布式的、解释的、健壮的、安全的、结构中立的、可移植的、高效的、多线程的和动态的语言。”

1. 简单性

Java 语言的简单性其实是指这门语言既易学又好用。但是，Java 自身小巧玲珑，有丰富的类库可以给开发者带来便利，并且对硬件的要求很低，这也体现了其简单性。

Java 语言的简单性还表现在它与传统的程序语言（如 C、C++）相比较。Java 语言简单易学，使用它编程时间短、功能强，人们接受起来也更快、更简便。不需要任何编程经验做基础就可以学习 Java。当然，如果一个已经学过其他程序设计语言的人再来学 Java，他将更快地掌握 Java 技术。

Java 通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。在 Java 中略去了运算符重载、多重继承等模糊的概念，并且

通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。

提示：上面叙述中涉及的“运算符重载”（这是 C++ 语言中的内容）、“多重继承”等概念，如果读者现在不懂，可以暂时搁置不管。“多重继承”在后续课程会介绍。本节中以及以后的学习中还会有类似情况，可以用同样的方式做临时性处理。

对于没有编程基础的读者来说，应该正确理解 Java 的简单性，只要树立信心，就会很快学会 Java 编程。当然，作者建议读者要把“不经学习就可掌握”（例如××三日通、××21 日通等）的美丽说法理解成容易学会就行了。

2. 面向对象

如果读者学习过 C++ 语言，就会觉得 Java 很眼熟，因为 Java 中许多基本语句的语法和 C++ 一样，像常用的分支、循环语句等与 C++ 几乎一样，但是，不要误解成 Java 没有什么新意，也不要认为 Java 是 C++ 的增强版，其实，Java 和 C++ 是两种完全不同的语言。

许多读者学习或使用过 C++，个别人甚至把 C++ 作为面向对象程序设计的标准，若用 Java 与 C++ 相比，Java 的面向对象技术更加彻底，这是因为 Java 要求所有内容都必须封装在类中，即以类作为程序的基本单位。大家可以看到，Java 程序最直观的是不允许类的外面有变量、方法等内容，所以 Java 是一种纯面向对象的程序设计语言。

3. 分布式

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

分布式包括数据分布和操作分布。Java 支持 WWW 客户机/服务器计算模式，因此，它可以支持这两种分布性。

4. 健壮性（也称鲁棒性）

Java 在编译和运行程序时，都要对可能出现的问题进行检查，以避免产生错误。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时出现容易产生的错误。通过集成的面向对象的异常（例外）处理机制，在编译时，Java 提示出可能出现但未被处理的异常（例外）。另外，Java 在编译时还可捕获类型声明中的许多常见错误，防止动态运行时出现不匹配问题。

Java 与 C/C++ 的最大区别是，采取了一个安全的指针模型，能减少重写内容和崩溃数据的可能性。

自动垃圾回收机制是 Java 的又一个特色，这种机制防止了内容丢失等动态内存分配导致的问题。

Java 提供了较完善的异常（例外）处理机制，程序员可以把一组可能产生运行异常的代码放在异常处理结构中，这样大大简化了异常处理过程，也使程序更加健壮。

需要说明的是，Java 并不是否定 C/C++ 的，它只是舍去了 C/C++ 中难以理解的、复杂的、不安全的内容。

5. 安全性

在网络、分布环境下的程序必须要防止非法的入侵。Java 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。Java 通过自己的安全机制预防了病毒下载程序对本地系统的破坏和威胁。

6. 体系结构中立

用 Java 解释器生成的与体系结构无关的字节码指令，只要安装了 Java 运行时系统，Java 程序就可以在任意的处理器上运行。Java 虚拟机（Java Virtual Machine, JVM）能够识别这些字节码指令，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台上运行。虚拟机的存在，大大增强了程序的跨平台运行能力，这也正是采用虚拟机的重要目的之一，它实现了 Java 的口号：“一次书写，到处运行（Write once, run anywhere）”。

7. 可移植性

与平台无关的特性使 Java 程序不必重新编译就可以方便地被移植到网络上的不同机器。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。

例如，Java 编译后并不生成可执行文件（.exe 文件），而是生成一种中间字节码文件（.class 文件）。任何操作系统，只要装有 Java 虚拟机，就可以解释并执行这个中间字节码文件。这正是 Java 实现可移植的机制。

在 Java 中的原始数据类型存储方法是固定的，避开了移植时可能产生的问题。

8. 解释型

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时的信息，使得连接过程更加简单。Java 语言的程序可以在提供 Java 语言解释器和实时运行系统的任意环境上运行。

9. 高性能

Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU（Central Processing Unit）的机器码，从而得到较高的性能。用 Java 编写的程序在网络上运行时，其速度要比 C/C++ 编写的程序快得多。

10. 多线程

多线程技术允许同一个程序中有两个以上的执行线路，即同时做两件事情，这样可以满足一些复杂软件的需要。

在 Java 中内置了对多线程的支持，在多线程模型中，多个线程共存于同一块内存中，且共享资源。这样，使用多线程机制提高了程序性能，每个线程分配有限的时间片来处理任务，由于 CPU 在各个线程之间的切换速度非常快，用户感觉不到，从而认为是在并行运行。

11. 动态性

Java 自身的设计使它比 C/C++ 更具有动态性，从而更加适合于一个不断发展的环境。Java 程序的基本组成单元是类（程序员编制的类或类库中的类），而类又是运行时动态装载的，这就使得 Java 可以在分布式环境中动态地维护程序及类库。

1.2 学习 Java 有何用途

对于高等教育大众化背景下的许多在校大学生来说，学习 Java 的最大用途是找到自己比较满意的工作岗位。

由于 Java 的跨平台、面向对象、安全性和健壮性等特点非常突出，所以企业明确提出开发时必须使用 Java 技术，Java 技术已经在各个领域获得广泛的应用，从而 Java 开发人员的需求量大幅度增加，软件企业在招聘员工时更是明确要求应聘者须掌握和熟练应用 Java 开发技术。

学习 Java，适应需求，发展爱好。

1.3 如何学习 Java

“磨刀不误砍柴工”。不要认为本节的内容不是 Java 程序设计语言本身的介绍，接受作者发自肺腑的指导（不仅仅是学习 Java），可以使读者少走弯路，达到事半功倍的效果。

有人说：学生能学好程序设计，不是仅凭教师教的，而是在教师的引导下通过大量的项目“练”出来的。这样说是很有道理的。

发掘自己的编程兴趣，掌握好的学习方法，养成好的学习习惯，培养自己吃苦耐劳、勤于钻研、科学严谨和团队协作的精神是非常重要的。为此作者向读者提出如下建议。

1. 快乐学习

“知之者不如好之者，好之者不如乐之者。”要学好 Java 技术，需要有非常浓厚的兴趣和坚持不懈的努力。

对于 Java 技术的学习，可能有的读者已经具有一定的或者非常浓厚的兴趣，也可能有的读者还谈不上有多少兴趣，但是，兴趣是可以培养的，而且，有的兴趣更是根据国家的需要培养的。例如，世界著名的杰出华人科学家、教育家、社会活动家，我国近代力学奠基人之一钱伟长先生（1912.10.09—2010.07.30），他 1931 年考大学时中文和历史两门学科都是 100 分，其余四门课（数学、物理、化学和英文）却总共考了 25 分，其中物理只考了 5 分，由于英文从没有学过，得 0 分。他以中文和历史两个 100 分的成绩进入了清华大学历史系，同年 9 月 18 日发生了“九·一八”事变，钱伟长决定要转学物理系以振兴中国的军力，当时的系主任吴有训先生一开始拒绝其转学要求，后被其诚意所打动而同意。钱伟长也通过勤奋学习证明了自己的实力。国际上以钱氏命名的力学、应用数学科研成果就有“钱伟长方程”、“钱伟长方法”、“钱伟长一般方程”、“圆柱壳的钱伟长方程”等。

每个人天生都具有很强的学习欲望，如果这些天然的动机能一直坚持下去，那么他迟早会成为优秀的人才。兴趣是学习的永久动力，作为学生，我们更应该意识到这一点，从学习 Java 技术这一刻开始，改变自己，把学习当成最有趣的事。当你感觉自己由对程序设计一窍不通到完成了一个程序的编写，当你看到你的程序解决了实际问题，当你看到自己能为信息产业做出贡献的时候，不仅成就感油然而生，兴趣也越发浓厚，继续学习去获得更大成就的动力将会倍增，这是莫大的快乐！

2. 乐于动手

“熟读《游泳学》，不如下大河”，“百闻不如一见，百见不如一干”。学习任何程序设计语言，都不能仅仅处于理论学习阶段，生活中很容易发现有的人能把程序设计语言的知识点背得滚瓜烂熟，却不会写出一个完整的程序的例子。学的目的是为了用，即重要的是要利用这门语言为自己的思想服务。理解语言的语法结构是重要的，但是要达到心领神会、融会贯通就必须多动手实践，阅读了教程中的例题的源代码，马上就想动手验证它是否可以正常运行，这才是良好学习的开端。计算机科学是一门十分重视实践的学科，成功的软件开发人员无不经过大量的编程实践锻炼，只有理论和实践相结合才能真正掌握知识和技能。

3. 不怕失败

“失败是成功之母”。在学习的过程中，不同的人会遇到不同的困难，有的人遇到的困难甚至很多。遇到困难时，有的人挺过去了，有的人放弃了。挺过去的人成功了，放弃的人失败了。很少有一个程序员写出的代码一次就成功，只有在不断的调试、修正中才能编写出真正的好代码。调试、解决问题的过程就是自己学习提高的过程。

伟人之所以伟大，是因为他与别人共处逆境时，别人失去了信心，他却下决心实现自己的目标。很多人羡慕名人成功的光环，笔者强烈建议读者要更关注的是名人的成名过程是一个受苦受难的历程。学习程序设计时，更应该注重学习的过程。

企业不仅看求职者做成功了多少项目，还要看求职者有多少项目的经历，求职者从经历中收获了什么，以及求职者对项目过程的看法。这样就更需要读者去享受过程。

4. 勤于动脑

“读书不想，隔靴挠痒。”“聪明人听到一次，思考十次；看到一次，实践十次。”在动手实践的过程中，要勤于动脑，无论成功还是失败，都要认真动脑思考其原因，总结经验，培养自己科学的、严谨的逻辑思维能力。例如，验证了教程上的例题程序可以正常运行后，要思考这个程序为什么是这样编写的？这个程序还有什么地方不周到的？我能否改动使其更加严密？实际上，教程上的例题程序受限于教学进程和篇幅，往往不是十分严密的，只要读者做个有心人，勤于动脑思考，就会有很大收获。

除了勤于动脑思考之外，还要善于用面向对象的思维方式来思考问题的解决办法。要时刻考虑到 Java 的面向对象特色，Java 是纯面向对象的，更重要的是建立面向对象思考问题的思维习惯，要掌握 Java 技术，就不能仅限于对语言本身的学习，如果想把 Java 学习提升到一个更高的层次，建议从一开始就用面向对象的思维方式去思考问题的解决方案。

5. 遵守规范

“没有规矩，不成方圆。”编写程序也有相应的编程规范。最基本的编码规范有命名规范（程序文件名、类名、对象名、成员变量名、成员方法名等，用有意义的名字并且符合规范）、程序文档排版规范（锯齿缩入式的排版）、注释规范（在必要的位置加入注释）等。

为什么要有编码规范？有以下几个主要原因：

- ① 一个软件的生命周期中，80%的费用花在维护上。
- ② 几乎没有任何一个软件，在其整个生命周期中，还由最初的开发人员来维护。
- ③ 编码规范可以改善软件的可读性，可以让程序员尽快而彻底地理解新代码。
- ④ 如果你将源码作为产品发布，就需要确认它是否被很好地打包并且清晰无误，一如你已构建的其他任何产品。

软件开发不仅仅考虑前期开发，而且更要考虑后期维护。

编程规范就是为了便于自己和他人阅读理解源程序，而制定的一个规范。当然，编程规范只是一个规范，也可以不遵守，但是，不遵守编程规范者一定是害己又害人的：首先，不遵守规范会被人认为是不专业的，也会在自己的调试过程中浪费较多的时间，在一些公司不遵守编程规范还将被扣奖金以示惩罚；再者，由于自己写的编码不符合规范，给其他合作者增加了阅读难度，浪费了他人的时间和精力。

作为程序设计的初学者，一定要遵守编码规范，养成良好习惯，以免“眼下省事，今后麻烦。”

各个公司往往都有自己的编程规范，但是，这些规范都是在基本编程规范的基础上加上本公司的特殊要求而成的。

6. 学会求助

“师傅领进门，修行靠个人。”教师在课堂教学过程中，已经把基本知识进行了介绍、把与知识点相关的案例进行了演示。在后续的项目训练过程中，一般人可能还会遇到种种不解或困难，在这种情况下，应该充分发挥自己的聪明才智，若自己认真思考仍然不得其解、反复实

践仍然没有成功的背景下，要及时向老师请教，不要等问题积累得太多了，见了老师无法或无从问起。

“三人行必有吾师。”多与同学讨论、参加 Java 技术论坛讨论，也是很好的求助方式。把自己对某个问题的认识、思考给自己的同学讲讲，既能梳理自己的思路，又能锻炼自己的表达能力，还能在讨论的过程中碰撞火花、激发灵感。当然，有时候自己感觉遇到了天大的困难，但是，让别人一看这个问题是如此的简单：例如，你在某个语句行后面忘记了一个分号。这个例子告诉我们，编程是一个非常辛苦、要求非常仔细的工作，来不得半点粗心大意。

“宁吃鲜桃一口，不吃烂杏一筐”多查帮助文档——Java API (Java Application Programming Interface)，多看 JDK 中带的演示文档（都是编程专家写的），都是很有效的向高手学习的方法。

7. 敢于创新

“能正确地提出问题就是迈出了创新的第一步。”这是诺贝尔物理学奖获得者李政道先生的名言。一个没有创新能力的民族，难以屹立于世界先进民族之林。

在创新的过程中，项目需要参与者综合运用各方面的知识，需要参与者充分发挥自己的聪明才智，这样，参与者的实践能力将会得到更大的提高。

在创新的过程中，同样重要的是“树立信心，相信自己”，没有天生的信心，只有不断培养的信心。相信自己包括两方面，一是相信自己的能力，二是相信自己的答案。

相信自己的能力就是要相信自己具有解决问题的能力。一个程序员的好坏并不是直接决定是否能编写出好的代码，更重要的是能否自己去解决调试过程中遇到的任何问题。

相信自己的解决方案、编程中用到的算法就是在充分论证成功的前提下，相信自己的程序运行的结果。

1.4 怎样搭建 Java 运行环境

在进行 Java 的开发之前，必须拥有 Java 的运行环境。有了 Java 运行环境，就可以利用文本编辑工具编写 Java 源程序，再使用 Java 编译程序对源程序进行编译，之后就可以运行了。

除了用一般的文本编辑器编写 Java 源程序外，还可以使编写源程序与运行程序在集成开发环境 (Integrated Development Environment, IDE) 中进行。

1.4.1 下载 JDK

Java SDK (Java Software Development Kit) 是由 Sun 公司所推出的 Java 开发工具，由于人们对早期的版本简称为 JDK，到现在人们往往还将 Java SDK (也有称为 J2SDK) 简称为 JDK。Java SDK 有以下三个版本：

① Java SE 标准版 (Java Platform, Standard Edition)，简称 Java SE。这是 Java 系统的标准与核心平台，主要面向个人用户，可以免费下载。

② Java EE 企业版 (Java Platform, Enterprise Edition)，简称 Java EE。这是标准版的企业级扩展版，面向企业和网络用户，需要注册。Java EE 是 Java 语言在企业级解决方案中的应用，大部分做 Java 的公司，基本上都是依靠 Java EE 盈利的。

③ Java ME 微型版 (Java Platform, Micro Edition)，简称 Java ME。这个版本是面向嵌入式系统应用的，例如小型家电、移动设备等消费类电子设备的编程，现在大部分的手机都支持 Java ME 的某个子集。

Java SE 是 Java EE 和 Java ME 的基础。作为初学者，我们这里仅以 Java SE 为例进行介绍。