

21世纪高等学校规划教材 | 计算机科学与技术



Java 程序设计 实践教程

颜志军 栾媛媛 编著



清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术



Java 程序设计 实践教程

颜志军 栾媛媛 编著

清华大学出版社
北京

内 容 简 介

本书介绍了使用 Java SE 平台进行 Java 开发的技术, 主要内容包括 Java 开发环境的搭建、Java 开发的必备基础知识、Java 开发的高级技术及使用 Java 进行网络程序和数据库程序开发。Java 开发的必备基础知识部分主要介绍了 Java 的基本语法、数组、字符串、类、枚举、异常、输入输出操作等内容; Java 开发高级技术则包含有多线程、容器类、反射和注解等相关内容; 最后一部分则介绍了图形界面编程、使用 TCP 和 UDP 协议进行网络编程、对数据库进行读写更新删除操作的技术。

本书循序渐进地介绍了 Java 的主要开发技术, 非常适合高校计算机相关专业作为学习 Java 编程技术的教材, 也可以作为培训机构的培训教材, 同时, 对于正在使用 Java 作为编程语言的程序员本书也有一定的参考价值。

图书在版编目(CIP)数据

Java 程序设计实践教程/颜志军, 栾媛媛编著. —北京: 清华大学出版社, 2011.1
(21 世纪高等学校规划教材·计算机科学与技术)
ISBN 978-7-302-23755-6

I. ①J… II. ①颜… ②栾… 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 (2010) 第 168120 号

责任编辑: 梁 颖

责任校对: 梁 毅

责任印制: 何 芊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京嘉实印刷有限公司

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 23.5 字 数: 569 千字

版 次: 2011 年 1 月第 1 版 印 次: 2011 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 36.00 元

产品编号: 038640-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色

精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与计算机应用。

(7) 21 世纪高等学校规划教材·电子商务。

清华大学出版社经过二十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail:weijj@tup.tsinghua.edu.cn

前言

Java 作为面向对象编程语言的杰出代表正被应用在越来越多的领域,从掌上设备到大型机,从嵌入式应用到企业级应用,Java 的身影随处可见。Java 作为开源的编程语言和开发工具,使得开发者无须担心版权和许可费用的问题,这也使得越来越多的项目倾向于使用 Java 作为开发语言。更为重要的是,Java 作为跨平台的语言,使得 Java 应用可以方便地不同操作系统平台之间进行移植,这使得那些将来可能更换运行平台的应用会更多地考虑将 Java 作为其开发语言。

事实上,Java 不只是一门编程语言,它还是一个规范,很多具体的技术实现 Java 本身并没有给出一个硬性的规定而只是给出了一个技术规范;同时,Java 也是一个开发工具,它提供了开发 Java 程序所必要的手段。当前,有多个国际大公司及技术社区在热情支持 Java 的发展,这使得 Java 具有极强的生命力。

本书从 Java 的特点及 Java 开发环境的搭建开始讲解 Java 开发技术,从 Java 的最基本数据类型、表达式、流程控制到面向对象技术最重要类的定义和使用、异常机制的原理和应用,由浅入深逐一涉及,而多线程、容器类等 Java 高级编程技术本书也有详尽介绍。最后还简单介绍了利用 Java 开发数据库应用程序、网络应用程序的技术。在整个过程中以图解为重要手段,以理论分析为导向,带领读者慢慢步入 Java 开发的殿堂,掌握 Java 开发技巧。

本书的内容结构

本书侧重于介绍 Java 开发技术中的 Java SE,Java SE 是学习 Java ME 和 Java EE 的必经之路,为学习 Java ME 和 Java EE 提供了必要的技术储备。从内容上说,本书覆盖了 Java SE 的主要内容,从学习 Java 入门知识到部分 Java 较底层的原理、从程序运行表象到本质,本书循序渐进均有涉及。

本书主要分为三大部分。

第一部分包含第 1~9 章的内容,主要介绍了 Java 开发的基础知识和基本技术,其中包括 Java 开发环境的搭建、Java 的基本语法、流程控制、数组及类、枚举、异常、输入与输出等内容。通过该部分的内容,读者可以从对 Java 技术一无所知逐渐上升到掌握 Java 开发的基本技术的水准。

第二部分包含第 10~12 章的内容,它讲述了 Java 中较高级的开发技术,主要内容包括多线程技术、容器类、反射与注解等内容。在通常的开发中,它们直接被应用的不多,但是掌握这部分技术后,对于开发一些有难度的项目以及理解 Java EE 技术及一些框架会极有帮助。所以通过该部分内容,读者可以了解 Java 技术的一些特性及一些底层运作机制,提升自己的技术层级。

第三部分包括第 13~15 章的内容,主要讲述了利用 Java 技术编写图形界面桌面应用

程序、使用 TCP、UDP 协议进行网络编程、操作数据库等内容。通过该部分内容，读者可以拓展视野，了解在实际应用中 Java 的应用情况，同时，也学习到了几乎在所有应用中都需要使用到的数据库操作技术，为进一步学习 Java 更高级技术做好技术准备。

本书另提供内容丰富的电子资源，可从 <http://www.tup.com.cn> 网址下载使用。

本书的读者对象

本书以步步图解的方式对 Java 开发环境搭建、Java 程序执行过程等进行逐步图解，非常适合进入 Java 开发的初学者，本书对程序执行步骤进行步步分解并进行批注图解，浅显易懂，即使复杂的程序也变得非常简单。如果您是以下类型的读者，此书会适合您：

1. 对 Java 所知甚少或者一无所知，但希望学习 Java 开发技术的读者。
2. 对 Java 早期版本较为熟悉，希望学习了解 Java 新技术的读者。
3. 对 Java 比较熟悉，希望了解 Java 更多知识点及技术的读者。

同时，本书也适合希望使用 Java 从事开发、培训等相关工作的朋友。

致谢

本书能够编写完成，首先要感谢那些在学习过程中不断提出问题的学生们，是他们的困惑、提问促成了编写本书的念头；其次，要感谢我的家人，没有他们的支持与鼓励，本书可能不会最后完成；最后，要感谢清华大学出版社的编辑梁颖先生，梁先生的建议令本书增色不少，同时也正是在梁先生的帮助下，本书方得以付梓。

读者在使用本书的过程之中，如果有任何的疑问，欢迎来信探讨，作者的电子邮箱是 ansreader@126.com。

编 者

2010 年 7 月

目 录

第 1 章 面向对象和 Java 语言	1
1.1 面向对象初探	1
1.1.1 程序设计语言及其发展历程	1
1.1.2 面向对象编程的主要优点	2
1.2 Java 的特点	3
1.2.1 简单性	3
1.2.2 面向对象	4
1.2.3 健壮性	4
1.2.4 安全性	4
1.2.5 解释执行	5
1.2.6 高性能	5
1.2.7 可移植性	5
1.3 Java 应用平台简介	5
1.3.1 Java SE (J2SE)	5
1.3.2 Java EE (J2EE)	6
1.3.3 Java ME (J2ME)	6
1.4 学习 Java 的建议	6
1.5 小结	9
1.6 习题	9
第 2 章 搭建 Java 学习环境	10
2.1 下载并安装 JDK	10
2.2 设置环境变量	15
2.2.1 设定 JAVA_HOME	16
2.2.2 设定 CLASSPATH	17
2.2.3 修改 Path	17
2.3 实现经典的 Hello World	18
2.3.1 源代码的编写及保存	18
2.3.2 编译源代码	19
2.3.3 执行程序	20
2.4 编辑及编译环境的选择	20
2.5 小结	21

2.6 习题	21
第3章 Java 的基本语法	22
3.1 Java 的基本数据类型	22
3.1.1 基本数据类型及其取值范围	22
3.1.2 常量及变量	24
3.1.3 数据类型转换	26
3.1.4 变量的作用域	28
3.1.5 局部变量的初始化	28
3.1.6 装箱与拆箱	29
3.2 表达式与位运算	32
3.2.1 算术表达式	32
3.2.2 赋值表达式	33
3.2.3 关系表达式	35
3.2.4 逻辑表达式	36
3.2.5 位运算	38
3.2.6 运算符的优先级	41
3.3 程序流程控制	42
3.3.1 顺序结构	42
3.3.2 分支结构	42
3.3.3 循环结构	52
3.4 小结	61
3.5 习题	61
第4章 数组	62
4.1 一维数组	62
4.1.1 数组的概念及定义形式	62
4.1.2 数组应用示例	67
4.1.3 Java 数组的实质	68
4.2 二维数组及多维数组	70
4.2.1 二维数组的定义及使用	70
4.2.2 深入理解二维数组	71
4.2.3 多维数组	73
4.3 数组操作类	74
4.3.1 System 类	74
4.3.2 Arrays 类	77
4.4 使用 foreach 遍历数组	81
4.5 小结	82
4.6 习题	82

第 5 章 字符串	83
5.1 String 类	83
5.1.1 String 对象的赋值	83
5.1.2 字符串操作方法	84
5.1.3 字符串池	89
5.2 StringBuffer 类与 StringBuilder 类	91
5.3 正则表达式	93
5.4 小结	96
5.5 习题	97
第 6 章 类的定义及使用	98
6.1 面向对象概述	98
6.2 Java 类定义格式	99
6.3 实现封装	102
6.3.1 类成员访问权限控制	102
6.3.2 封装数据成员	103
6.3.3 封装方法成员	106
6.3.4 方法的重载	108
6.3.5 特殊的方法成员——构造器	113
6.4 类的继承	118
6.4.1 扩展类	118
6.4.2 子类实例化过程	120
6.4.3 重写父类方法	122
6.4.4 Object 类	125
6.5 多态、抽象类及接口	128
6.5.1 多态及其实现	128
6.5.2 基于抽象类实现多态	131
6.5.3 基于接口实现多态	132
6.5.4 实现接口和类之间的适配器	135
6.6 包	137
6.6.1 Java 的类管理机制	137
6.6.2 导入其他包中的类	139
6.6.3 类成员访问权限控制补充及类的访问权限	141
6.7 内嵌类	141
6.7.1 成员内嵌类	142
6.7.2 局部内嵌类	144
6.7.3 匿名内嵌类	145
6.8 小结	146

6.9 习题	146
第7章 枚举	147
7.1 枚举的定义	147
7.1.1 使用常量增强程序的可读性及可维护性	147
7.1.2 使用枚举类型替代常量	149
7.2 枚举的实质	150
7.2.1 枚举中可以使用的方法	151
7.2.2 为枚举添加构造器	152
7.2.3 为枚举添加方法	153
7.3 小结	155
7.4 习题	155
第8章 异常处理及自定义异常	156
8.1 在代码中处理异常	156
8.1.1 异常处理机制	156
8.1.2 异常处理示例	157
8.2 在方法中抛出异常	164
8.2.1 通过 throws 抛出要求调用者处理的异常	164
8.2.2 使用 throw 抛出异常	166
8.3 自定义异常	168
8.3.1 异常的继承关系	168
8.3.2 实现自定义的异常类	170
8.3.3 重写方法时的异常处理	173
8.4 断言	175
8.5 小结	177
8.6 习题	177
第9章 输入与输出操作	178
9.1 接受键盘输入	178
9.1.1 System.in 及流的概念	178
9.1.2 Scanner	181
9.2 输出及输出格式控制	183
9.2.1 System.out	183
9.2.2 输出格式控制	184
9.3 文件	186
9.3.1 File 类	186
9.3.2 RandomAccessFile 类	189
9.4 字节流	191

9.4.1	InputStream 和 OutputStream	191
9.4.2	FileInputStream 和 FileOutputStream	193
9.4.3	BufferedInputStream 和 BufferedOutputStream	194
9.4.4	DataInputStream 和 DataOutputStream	196
9.4.5	ObjectInputStream 和 ObjectOutputStream	198
9.4.6	ByteArrayInputStream 和 ByteArrayOutputStream	200
9.4.7	PushbackInputStream	203
9.5	字符流	206
9.5.1	Reader 和 Writer	206
9.5.2	InputStreamReader 和 OutputStreamWriter	207
9.5.3	FileReader 和 FileWriter	208
9.5.4	BufferedReader 和 BufferedWriter	209
9.5.5	PrintWriter	210
9.6	小结	211
9.7	习题	211
第 10 章	多线程	213
10.1	多线程简介	213
10.1.1	多线程原理	213
10.1.2	线程的生命周期	214
10.2	通过继承类 Thread 产生线程	215
10.3	通过实现接口 Runnable 产生线程	225
10.4	线程同步	227
10.4.1	多线程带来的问题	227
10.4.2	同步线程	230
10.4.3	线程协同	234
10.5	Daemon 线程	242
10.6	线程组	245
10.7	小结	247
10.8	习题	247
第 11 章	容器类与泛型编程	248
11.1	泛型编程简介及泛型编程思想	248
11.2	List	256
11.2.1	List 接口	256
11.2.2	ArrayList	257
11.2.3	LinkedList	259
11.3	Set	262
11.3.1	Set 接口	262

11.3.2	HashSet	262
11.3.3	TreeSet	263
11.4	Map	265
11.4.1	Map 接口	265
11.4.2	HashMap	266
11.4.3	TreeMap	268
11.5	小结	270
11.6	习题	270
第 12 章	反射与注解	271
12.1	加载类	271
12.1.1	通过 Class 对象取得类的基本信息	272
12.1.2	类的加载时机	275
12.1.3	使用 Class.forName 加载类	277
12.2	使用反射生成与操作对象	279
12.2.1	生成对象	279
12.2.2	调用方法	281
12.2.3	访问数据成员	284
12.3	注解	287
12.3.1	说明重写父类——@Override	287
12.3.2	说明不建议使用——@Deprecated	289
12.3.3	抑制编译警告——@SuppressWarnings	289
12.4	小结	291
第 13 章	图形界面编程	292
13.1	AWT 与 Swing 简介	292
13.1.1	AWT 简介	292
13.1.2	Swing 简介	293
13.2	Swing 容器及组件	296
13.2.1	Swing 容器——JFrame 及 JPanel	296
13.2.2	Swing 组件——JLabel、JTextField、JButton 及 JTextArea	298
13.2.3	其他容器与组件简介	301
13.3	Java 事件模型	308
13.3.1	事件处理机制简介	308
13.3.2	事件监听器	309
13.3.3	事件适配器	311
13.4	布局管理器	312
13.4.1	布局管理器概述	313
13.4.2	BorderLayout	313

13.4.3	FlowLayout	314
13.4.4	GridLayout	315
13.4.5	其他布局管理器	316
13.5	为窗口添加菜单	316
13.5.1	Java 菜单简述	316
13.5.2	为英汉电子辞典添加菜单	318
13.6	小结	319
第 14 章	网络编程	320
14.1	TCP/IP 协议与套接字	320
14.2	基于 TCP 协议的网络程序	321
14.2.1	TCP 协议简介	322
14.2.2	互联网协议地址类 InetAddress	323
14.2.3	服务器套接字类 ServerSocket	323
14.2.4	套接字类 Socket	325
14.2.5	基于 TCP 协议的服务程序	326
14.2.6	基于 TCP 协议的客户端程序	329
14.3	基于 UDP 协议的网络程序	331
14.3.1	UDP 协议简介	331
14.3.2	数据报套接字类 DatagramSocket	332
14.3.3	数据报包类 DatagramPacket	333
14.3.4	基于 UDP 协议的发送程序	335
14.3.5	基于 UDP 协议的接收程序	336
14.4	小结	337
第 15 章	数据库编程	338
15.1	JDBC 简介	338
15.2	连接数据库	341
15.2.1	使用 JDBC-ODBC 桥连接 Access 数据库	342
15.2.2	连接 SQL Server 数据库	347
15.2.3	连接 Oracle 数据库	348
15.2.4	连接 MySQL 数据库	349
15.3	查询数据	351
15.4	更新、删除与插入数据	352
15.4.1	使用 ResultSet 更新、删除与插入数据	353
15.4.2	使用 Statement 更新、删除与插入数据	356
15.5	小结	358

第1章

面向对象和 Java 语言

面向对象编程是当前主流的编程方式。它之所以能够流行，与面向对象具有的诸多优点密切相关，而 Java 则堪称面向对象编程语言的典范。通过本章的学习，读者可以对 Java 有一个初步的印象，而在后继章节中读者将逐渐深入学习 Java 各方面的知识。本章主要涉及的内容有：

- 面向对象程序设计的优点。
- Java 的特点。
- Java 应用平台简介。
- 学好 Java 的一些建议。

1.1 面向对象初探

当前软件开发的主流是采用面向对象的编程思想进行设计与编程，由于面向对象的开发思想拥有众多的优点，以至于它受到了众多开发者的欢迎。本节将简单介绍程序设计语言的发展历程及面向对象编程的优点。

1.1.1 程序设计语言及其发展历程

通常，人们将计算机语言分为机器语言、汇编语言和高级语言三类。

机器语言是计算机可以识别的语言，它是仅由 1 和 0 组成的二进制机器码。事实上，无论何种语言，最终都需要以机器码的形式被计算机执行。在计算机出现初期，各类程序都是一系列长长的二进制机器码的集合。

毫无疑问，直接以机器码进行编程对编程人员的细心及耐心的极限都是一个挑战，因此以助记符为主要标志的汇编语言的出现也就成为自然而然的事情。助记符来源于自然语言，它极大地简化了编程人员的工作，不过它依然十分接近机器的底层，在汇编程序中，包含大量的对寄存器及内存的操作，所以虽然汇编语言极大地简化了编程，但它依然与特定的机器有密切关系，不同的 CPU 使用不同的指令集，因此汇编语言因硬件的不同也会存在或多或少的差异。对于编程人员而言，在使用汇编语言时意味着他需要清楚地知道这个程序最终会在何种机器上运行，并且程序员需要了解这种机器的硬件情况。

随后出现的高级语言对于编程人员可以称得上是一次大解放，编程人员无须再了解硬

件的细节，而且高级语言使用接近自然语言（英语）的语法，学习起来比汇编语言也容易得多。高级语言大致可以分为两类：解释型语言和编译型语言，解释型语言在运行时要求目标计算机上有一个相对应的解释器，解释器一边读入程序源代码一边将它翻译成机器可以识别的机器码，然后再交给机器执行；编译型语言则在源代码完成之后需要对源代码进行编译和连接，源代码经过编译之后可以得到所谓的目标代码，而目标代码再经过连接就可以得到一个可执行文件，可执行文件已经是由机器码组成的指令序列，因此它可以直接在机器上运行。解释型语言中的代表是 BASIC 语言，它简单易学，曾经是大部分高校计算机专业的教学语言，也是众多程序员的入门语言；编译型语言的代表则是至今仍被广泛使用的 C 语言，它功能强大、语法简捷灵活。而在 C 语言流行之后被大多数人认为是 C 语言的替代语言（也有部分程序员认为 C 与 C++ 是两个完全无关的语言）的 C++ 逐渐流行起来，而正是由于 C++ 的流行也将面向对象的编程思想带到了更多程序员面前，尽管 C++ 并不是第一个面向对象的语言，但它却是最早流行的面向对象语言，而 Java 的出现则将面向对象编程推向了一个高潮。对于编程人员而言，面向对象的编程思想的出现可以称得上是一次革命。

1.1.2 面向对象编程的主要优点

面向对象编程思想主张从客观世界固有的事物出发来构造系统，它使用人类在现实生活中的思维方法来构建软件系统，强调最终建立的系统能够映射需要解决的问题域。也就是说，在使用面向对象思想构建的系统中，对象及对象之间的关系可以如实反映问题域中的固有事物及其关系。

面向对象的程序设计方法之所以能够越来越受到开发人员的欢迎，成为主流的软件开发方法，与面向对象具有的优点是密不可分的。面向对象的优点简述如下。

1. 符合人类的思维习惯

传统的程序设计方法以算法为核心，它将数据和相关操作过程作为相互独立的部分。也就是说，这种设计方法将一个实体中的数据与操作分离，使得程序具有一定程度上的通用性，但却要求软件设计人员拥有极强的分析能力及设计能力，而且一旦需求发生变更，很可能只是一个细小的变更，就有可能使得原有的设计不再具有合理性，不得不重新设计。

面向对象的程序设计方法则以对象为核心，它将现实系统中存在的不同实体进行抽象处理，在软件系统中形成与现实系统中对应的对象，并且通过对象之间的消息传递实现这些对象之间的关系。而面向对象思想中的“对象”需要通过所谓的“类”来实现，面向对象中的类与我们平时所说的类的含义几乎是一样的，它不仅是一系列实体的抽象，而且也能够反映父类与子类这样的继承关系。归根到底，面向对象的程序设计方法以极其接近人类思维的方式设计构造系统。

2. 系统稳定

在现实世界中，很少有正在运行的系统会因为局部变化而被全盘推翻，进而出现不得不全新设计构建新系统的情况。基于面向对象程序设计思想构建的系统与现实系统是类似的，所以使用面向对象的程序设计方法构建的系统也很少需要全新设计，通常只是需要针对现实中的系统改变做一些修正而已。因此，使用面向对象思想构建的系统有较高的稳定性，几乎不会出现在结构化设计中因为新的需求而被迫重新设计整个系统的现象。

另一方面，面向对象程序中的基本逻辑单元是类，它构成了程序中的模块，这使得模块之间有较高的独立性，即类之间耦合度较低，而类的内部则具有较高的内聚性，这使得

在个别类设计不当或需要进行变更时对整个系统的影响较低。

3. 有较高的可重用性

现在软件产品的规模越来越大, 开发商之间的竞争也越来越激烈, 所以软件公司现在呈现专业化的趋势, 即一个软件公司或者一个部门是专注于某一个行业的。专业化的运作方式使得不同客户的需求之间存在较大的相似性, 这为重复使用代码提供了可能性, 而如果在项目中能够从以前的项目中复用代码, 将能够在保证质量的前提下缩短开发周期, 降低软件的开发成本。

如果不是采用面向对象的方式, 复用以前的代码通常只能采用复制与粘贴的方式进行, 而如果被复制的代码用到了原来工程中的其他变量或声明, 往往被复制的代码需要大量的时间进行调试才可以融入到当前系统之中。而采用面向对象的方式, 则可以通过类的继承或实现类的实例两种方式对代码进行复用, 因为这两种方式都不会涉及将代码加入当前系统中的问题, 所以不用调试就可以在当前系统中利用此前已经存在代码, 使得复用代码的可能性与效率大为提高。

4. 易于开发大型软件产品

面向对象的编程天生具有很好的模块化特征, 每一个类就是一个基本模块。在开发过程中, 控制及保证一个类的质量是比较容易的, 而由这些类再组成一个软件系统时可能主要的问题是类之间的协作问题, 一般很少出现由于逻辑上过于复杂造成无法构建系统的现象。而对于传统的结构化编程则可能由于这些庞杂关系而使得编程人员在构建系统时思维混乱, 最终无法完成系统。事实上, 如果一个项目采用面向对象的方法进行设计, 并采用了行之有效的管理手段, 它在大部分情况下都会是一个成功的项目, 对于大型项目尤为如此。

需要特别说明一点, 结构化编程与面向对象并不是对立的, 在面向对象的编程中, 在实现类中的一个方法时, 依然会采用结构化编程方式实现其功能。

5. 具有较高的可维护性

软件系统的维护在软件生命周期中占有较重要的地位。实际上, 软件的维护不只发生在维护阶段, 通常在软件进行升级的同时也需要进行一些维护性的操作。而基于面向对象的软件系统因为其系统结构易于理解、系统稳定性好, 负责维护的人员可以很快地弄清整个系统运作流程及机制, 又因类具有良好的封装性, 在进行代码修改时它影响的范围较小, 一般不会产生那种牵一发而动全身的影响, 所以在实施维护时可以很快地完成修改、调试及测试等工作, 因此面向对象的系统具有较强的适应性。而通过对软件的维护, 在很大程度上可以延长软件的生命周期, 它意味着成本的降低, 所以较高的可维护性是面向对象编程生命力的表现之一。

1.2 Java 的特点

作为一门编程语言, Java 具有许多重要的特点, 而这些特点之中有一些就是 Java 相对于其他开发语言的优势。关于这一点, 在众多的 Java 社区中可能已经有充分的讨论, 本节仅作简单介绍。

1.2.1 简单性

大部分“年长”的 Java 程序员是由 C/C++ 程序员转化而来, C/C++ 的强大功能令大多