

◇ 高等院校计算机专业应用型规划教材 ◇

程序设计系列

Java语言 程序设计

于红 徐敦波 冯艳红 等编著

- 注重知识的系统性
- 注重代码的规范性
- 注重例题的实用性和层次性
- 注重教材的易读性
- 注重教学内容的层次性和启发性



系列教材主编 陈明



机械工业出版社
CHINA MACHINE PRESS



高等院校计算机专业应用型规划教材

Java 语言程序设计

于红 徐敦波 冯艳红 等编著



机械工业出版社

本书主要介绍 Java 语言与面向对象思想、Java 语言开发环境、Java 语言基础、Java 的面向对象程序设计、数组和字符串、Java 语言的实用类与类的设计、Java 的异常处理、I/O 流、多线程、图形用户界面、网络编程和数据库编程等内容。本书注重培养学生分析问题、解决问题的能力，注重启发式教学方法的运用，让学生切实掌握运用 Java 解决实际问题的方法。

本书可作为高等学校计算机及相关专业的教材，也可作为软件开发人员或研究人员的参考用书。

图书在版编目(CIP)数据

Java 语言程序设计/于红等编著. —北京:机械工业出版社, 2012. 4
高等院校计算机专业应用型规划教材
ISBN 978-7-111-36546-4

I. ①J… II. ①于… 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(2012)第 046535 号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)
责任编辑:郝建伟 罗子超
责任印制:杨 曦
唐山丰电印务有限公司印刷

2012年5月第1版·第1次印刷
184mm×260mm·25.25印张·626千字
0001-3000册
标准书号:ISBN 978-7-111-36546-4
定价:49.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

高等院校计算机专业应用型规划教材

编委会成员名单

主 任 陈 明

委 员 (以姓氏拼音为序)

曹永存	李猛坤	李海生	李 环
刘振华	刘贵龙	鹿 旸	郝 莹
孙践知	王锁柱	王智广	王振武
谢 俊	徐孝凯	姚 琳	袁 薇
张晓明	张建林		

丛 书 序

随着计算机科学与技术的飞速发展，现代计算机系统的功能越来越强大，应用也越来越广泛，计算机科学对人类社会的发展做出了卓越的贡献。社会的需求与应用是推动计算机学科发展的源动力，一直受到社会的大力关注。

计算机学科的发展呈现出学科内涵宽泛化、分支相对独立化、社会需求多样化、专业规模巨大化和计算机教育大众化等特点，一方面使得计算机企业成为朝阳企业，软件公司、网络公司等 IT 企业的数量和规模越来越大；另一方面对计算机人才的需求规格也发生了巨大的变化。在大学中单一的计算机精英型教育培养的人才已不能满足实际需要，社会需要大量的具有职业特征的计算机应用型人才。

计算机应用型教育的培养目标可以利用知识、能力和素质三个基本要素来描述。知识是基础、载体和表现形式，从根本上影响着能力和素质。学习知识的目的是为了获得能力和不断地提升能力。能力和素质的培养必须通过具体的知识传授来实现，能力和素质也必须通过知识来表现。能力是核心，是人才特征的最突出的表现。计算机学科人才应具备计算思维能力、算法设计与分析能力、程序设计能力和系统能力（系统的认知、设计、开发和应用）。计算机应用型人才的能力主要包括应用能力和通用能力。应用能力主要是指用所学知识解决专业实际问题的能力。通用能力表现为跨职业能力，并不是具体的专业能力和职业技能，而是对不同职业的适应能力。计算机应用型本科人才所应具备的三种通用能力是学习能力、工作能力、创新能力。基本素质是指具有良好的公民道德和职业道德，具有合格的政治思想素养，遵守计算机相关法规和法律，具有人文、科学素养和良好的职业素质等。计算机应用型人才素质主要是指工作的基本素质，且要求在从业中必须具备责任意识，能够对自己职责范围内的工作认真负责地完成。

计算机应用型教育课程类型分为通用课程、基础课程、专业核心课程、专业选修课程、应用课程、实验课程、实践课程。课程是载体、是实现培养目标的重要手段。教育理念的实现必须借助于课程来完成。教育类型和教育层次的划分实质上是课程内容和课程性质的划分。因此，计算机应用型教育培养目标的实现关键在于课程体系的构建，即课程内容和课程性质的确定。本系列规划教材的特点是重点突出、理论够用、注重应用、内容先进实用。

计算机应用型人才培养的研究方兴未艾，本系列教材有不足之处，敬请各位专家、老师和广大同学赐教，为推动计算机应用型人才的培养做出贡献。

陈明

前 言

Java 是可用于撰写跨平台应用程序的面向对象的程序设计语言，由于其良好的跨平台特性和强大的网络编程能力，使其成为目前最流行的程序设计语言之一。

目前几乎所有的高等学校计算机及相关专业均开设了 Java 语言程序设计课程，Java 语言程序设计的相关教材也有很多，每本教材都有鲜明的特点，适合不同的人群使用。本书是针对目前大多数学生语言知识点掌握较好，但语言运用能力较差的特点，基于调研和分析而获得出现这一问题的主要原因，并结合 Java 语言学习的实际需要而编写的，目的是希望学生不仅学会单个知识点，更要学会知识的融会贯通和综合运用。本书具有以下特点：

1. 注重知识的系统性

按照从总体到局部，再从局部到整体的总—分—总结构来建立内容体系，使得读者一开始对 Java 程序的结构有一个整体认识，再详细学习每一个知识点，最后通过完整实例把各个知识点串起来，让学生对整个课程内容有一个系统的认识。

2. 注重代码的规范性

注重例题代码的规范性和效率，本书的目的不仅是让学生学会编程，而且是让学生学会编写好的、实用的软件，与公司需求接轨。

3. 注重例题的实用性

本书从项目的角度学习 Java，在第一部分先抛出具体的需求，在讲解各个知识点时尽量用实际需求中的内容作为实例和练习题，尽量采用完整的实例来讲解每一个知识点，以期达到学完整本书之后，就完成了一个项目的目的。

4. 注重教材的易读性

加大例题的比例，让学生从实际需求中掌握编程技巧，达到从感性认识到理性认识的过程。

5. 注重教学内容的层次性

对比较复杂的内容采用由简单到复杂、由浅入深逐步完善的结构，体现出逐步优化的过程，很多章节开始时给出的例题都是学生容易理解、但是解决实际问题时又存在缺陷的代码，然后逐步优化为解决实际问题时比较有效的代码，让学生通过对这两种代码进行比较，自己得出结论，从而加深对知识的理解。

6. 注重教学内容的启发性

在阐述各部分知识的过程中抛出一些思考题，这些思考题启发学生思考一些与所讲知识相关的问题，一方面可以培养学生在看书的过程中思考的习惯，另一方面还可以帮助学生把相关的知识衔接起来。

7. 注重例题的层次性

每章的课后习题分基础题、编程题和拓展题三个层次，分别考量学生的基础知识、基本技能和综合运用知识的能力。

本书第 1、2、13 章及第 6.5 节由徐敦波编写，第 3、5 章由于红编写，第 4、6 章的其他内容由冯艳红编写，第 7、8、9 章由史鹏辉编写，第 10、11、12 章由孙庚编写。全书最后由于红、徐敦波和冯艳红进行统校定稿，冯艳红和于红测试了书中全部例题的程序。崔春雷和隋瑞雪对本书的编写提供了大量的帮助，在此一并表示感谢。

由于作者水平有限，书中难免有不足之处，恳请读者批评指正。

编 者

目 录

丛书序

前言

第 1 章 Java 语言与面向对象思想概述	1
1.1 Java 语言的产生与发展	1
1.1.1 Java 语言的产生	1
1.1.2 Java 语言的发展	2
1.2 Java 语言的特点	3
1.3 面向对象与面向过程的比较	4
1.3.1 面向过程的程序设计	5
1.3.2 面向对象的程序设计	5
1.4 面向对象程序设计中的主要概念和特征	6
1.4.1 主要概念	7
1.4.2 主要特征	8
1.5 Java 与 C++ 的差异	9
1.6 本章小结	11
1.7 习题	11
第 2 章 Java 语言开发环境	12
2.1 JDK 工具包与基本 Java 程序	12
2.1.1 JDK 的下载与安装	12
2.1.2 JDK 目录的构成	14
2.1.3 设置环境变量	15
2.1.4 Applications 程序的结构	16
2.1.5 JDK 的使用	17
2.1.6 典型的 Java 编程规范	19
2.2 IDE 环境概述	21
2.2.1 IDE 简介	22
2.2.2 Eclipse	22
2.2.3 JBuilder	27
2.2.4 相关资源	27
2.3 Project 管理	28
2.3.1 Project 的含义	28
2.3.2 可行的 Project 组织模式	28
2.3.3 主要 IDE 工具的 Project 包结构	29

2.4	本章小结	30
2.5	习题	30
第3章	Java 语言基础	31
3.1	标识符和关键字	31
3.2	常量、变量与基本数据类型	33
3.2.1	常量、变量和数据类型体系	33
3.2.2	整数类型	34
3.2.3	浮点型	35
3.2.4	逻辑型	37
3.2.5	字符型	37
3.3	运算符与表达式	38
3.3.1	算术运算符	38
3.3.2	关系运算符	40
3.3.3	逻辑运算符	41
3.3.4	位运算符	42
3.3.5	移位运算符 (选学)	43
3.3.6	赋值运算符	43
3.3.7	特殊运算符	44
3.3.8	运算符的优先级	44
3.4	基本数据类型转换	46
3.4.1	类型转换的形式	46
3.4.2	类型转换机理	48
3.4.3	类型转换的规则	49
3.5	流程控制	50
3.5.1	布尔表达式	51
3.5.2	分支结构	51
3.5.3	循环结构	55
3.6	本章小结	57
3.7	习题	58
第4章	Java 的面向对象程序设计	61
4.1	类与对象	61
4.1.1	类的定义	61
4.1.2	方法	63
4.1.3	构造方法与对象创建	66
4.1.4	访问权限	77
4.2	继承	84
4.2.1	继承的定义	85
4.2.2	子类对象构造过程	86
4.2.3	访问控制	92

4.2.4	变量隐藏和方法覆盖	95
4.2.5	多态	99
4.3	特殊修饰控制符	104
4.3.1	static	104
4.3.2	final	108
4.4	对象的生命周期	112
4.5	Object 类	114
4.6	抽象类和接口	116
4.6.1	抽象类	116
4.6.2	接口	119
4.7	内部类	122
4.7.1	内部类的作用	123
4.7.2	内部类的定义和使用	123
4.7.3	匿名内部类	127
4.8	本章小结	128
4.9	习题	128
第 5 章	数组和字符串	131
5.1	数组	131
5.1.1	一维数组	131
5.1.2	多维数组	136
5.2	字符串	137
5.2.1	字符串类	137
5.2.2	字符串类与字符数组的关系	138
5.2.3	字符串类的常用方法	139
5.2.4	字符串运算与转换	141
5.3	StringBuffer	142
5.4	本章小结	145
5.5	习题	145
第 6 章	实用类与类的设计	148
6.1	串分解类	148
6.2	正则表达式	150
6.2.1	正则表达式的语法	150
6.2.2	正则表达式的使用	152
6.3	集合框架	154
6.3.1	集合框架的概念	154
6.3.2	列表	156
6.3.3	散列表	160
6.4	常用类	161
6.4.1	Date 类	161

6.4.2	Calendar 类	164
6.4.3	Math 类	167
6.5	类的设计	169
6.5.1	面向对象的知识要点	170
6.5.2	面向对象入门者的混沌与困惑	170
6.5.3	类的分类	174
6.5.4	类的设计原则	175
6.5.5	类图	177
6.5.6	单类设计实例	179
6.5.7	多类设计实例	185
6.6	本章小结	188
6.7	习题	188
第7章	Java 的异常处理	190
7.1	异常的概念	190
7.2	异常处理机制	192
7.2.1	异常的捕获	192
7.2.2	异常的抛出	194
7.2.3	异常的声明	195
7.2.4	重抛异常	197
7.3	自定义异常类	197
7.4	本章小结	202
7.5	习题	202
第8章	I/O 流	205
8.1	流的概述	205
8.2	File 类	205
8.3	字节流	210
8.3.1	InputStream	210
8.3.2	OutputStream	211
8.4	文件字节流	212
8.4.1	FileInputStream	212
8.4.2	FileOutputStream	212
8.5	字符流	215
8.5.1	Reader 类	215
8.5.2	Writer 类	219
8.6	RandomAccessFile	221
8.7	其他数据流	225
8.7.1	DataOutputStream	225
8.7.2	DataInputStream	226
8.8	本章小结	228

8.9 习题	228
第9章 多线程	230
9.1 多线程机制	230
9.1.1 基本概念	230
9.1.2 线程的状态	230
9.2 线程类和线程接口	231
9.2.1 Thread 类	231
9.2.2 Runnable 接口	234
9.2.3 线程组	236
9.3 线程的同步与通信	238
9.3.1 线程同步	238
9.3.2 线程通信	242
9.4 本章小结	246
9.5 习题	246
第10章 图形用户界面	247
10.1 AWT 和 Swing 组件	247
10.1.1 AWT 组件	247
10.1.2 Swing 组件	247
10.2 框架	248
10.2.1 窗口	248
10.2.2 布局管理器	250
10.2.3 中间容器	257
10.3 Java 事件处理机制	261
10.3.1 Java 事件的基本概念	261
10.3.2 Java 事件的处理机制	263
10.3.3 适配器类	267
10.4 其他常用组件	270
10.4.1 标签	270
10.4.2 按钮	271
10.4.3 选择框	274
10.4.4 文本组件	278
10.4.5 列表框	281
10.4.6 对话框	287
10.4.7 菜单	293
10.5 本章小结	299
10.6 习题	299
第11章 网络编程	301
11.1 网络编程基础	301
11.2 URL 编程	302

11.2.1	URL 类	302
11.2.2	URLConnection 类	304
11.2.3	InetAddress 类	305
11.3	Socket 编程	308
11.3.1	Socket 编程模式	308
11.3.2	Socket 和 ServerSocket 类	308
11.4	UDP 编程 (选学)	313
11.4.1	DatagramPacket 类	313
11.4.2	DatagramSocket 类	314
11.5	本章小结	315
11.6	习题	316
第 12 章	数据库编程	317
12.1	JDBC 技术	317
12.1.1	JDBC 的概念	317
12.1.2	JDBC 编程接口	317
12.2	基本 JDBC 编程	318
12.2.1	连接	318
12.2.2	查询	322
12.2.3	更新	326
12.2.4	事务	336
12.3	本章小结	338
12.4	习题	338
第 13 章	编程实例	339
13.1	基本需求描述与分析设计方案	339
13.2	应用基本面向对象的实现技术	345
13.3	应用集合框架的实现技术	357
13.4	引入 I/O 流的实现技术	359
13.5	基于网络工作环境的实现技术	367
13.6	应用 GUI 进行独立的表示层实现	380
13.7	采用数据库的实现技术	382
13.8	扩展需求后的整体设计方案	387
13.9	本章小结	390
13.10	习题	390
参考文献		391

第 1 章 Java 语言与面向对象思想概述

Java 语言不仅具备了完全面向对象、动态、安全、多线程等特点，还具有跨平台和支持分布式开发等特性，是目前软件项目开发中的主流语言。随着 Java 技术的发展，它对语言、开发、应用、部署各层次环境的支持越来越完善。

本章的目标是在介绍 Java 语言的发展、特点等常识的基础上，讲述面向对象的思想和方法，使读者初步建立面向对象的认知模式，以期尽快进入角色并掌握后续章节的全部技术，达到完全理解面向对象的境界。为了使学过 C++ 的读者能更快地掌握 Java 技术，本章的最后介绍了 Java 和 C++ 的区别，没有学过 C++ 的读者可以跳过这部分内容。

1.1 Java 语言的产生与发展

1.1.1 Java 语言的产生

20 世纪 90 年代初期，Sun 公司为了开拓计算机在消费类电子产品中的应用，试图研究一种小巧的、能适用于未来智能设备的编程语言。为了满足未来智能设备的编程需求，该语言需要避免类似 C++ 的过于复杂、安全性差等问题，同时要兼顾消费类电子产品中五花八门的软硬件环境和有限的系统资源。

该语言是在 C++ 的基础上开发的，保留了 C++ 的大部分语法，消除了直接的资源引用、指针运算、运算符重载、多重继承等危险和复杂的特性。同时，由于是面向嵌入式应用领域，开发小组的技术人员选择了一种假设方案，定义了一种结构简单、符合嵌入式应用的硬件平台体系结构规范，其中包括能执行“字节码”的机器指令系统，这也就是后来 Java 技术的核心——虚拟机。

该语言产生于 Internet 迅速发展的早期阶段，因而开发小组的技术人员也认识到了 Internet 所带来的机遇，1994 年，他们用该语言开发了一个实时性较高、可靠、安全、有交互功能的新型 Web 浏览器，命名为 WebRunner，它不依赖于任何硬件平台和软件平台。其主要特点是利用 Applet（小应用程序）解决了用户—浏览器—服务器之间的动态交互问题。同时，Applet 结构小巧，方便自动即时更新。

该语言起初命名为 Oak，起源于开发小组一位作者办公室窗外的一棵橡树（Oak）。1995 年，在注册的时候遇到了冲突，于是技术人员就从手中的热咖啡联想到了印度尼西亚一个盛产咖啡的岛屿，中文名叫爪哇，Java 语言便得名于此。同年，Sun 公司将 WebRunner 浏览器更名为 HotJava，同 Java 语言一起正式在业界对外发布，引起了巨大的轰动，Java 的地位随之得到肯定，此后的发展非常迅速。

Java 语言的产生并不是一个完全事先预想的目标和过程，但是却完美地利用了一切技术发展的最佳机会。Java 语言从一开始就直接支持网络环境编程，因而适合于分布式软件开

发，同时又具有跨平台特性，因而受到 IT 领域各大专业公司的支持。

以下是对 Java 编程语言的主要设计目标进行的整理，虽然不排除某些条目并非设计的初衷，但是对于 Java 技术的学习者和开发者来说，还是有重要意义的。

1) 提供一种易于编程的语法，消除过去的语言在诸如指针运算和存储器管理方面影响健壮性的缺陷。

2) 充分利用面向对象的概念，使 Java 语言编写的程序真正成为面向对象的程序。

3) 改善编译器的编译方法，使程序代码尽可能清晰合理、简明流畅。

4) 建立能够获得如下两个关键点的一种解释环境。

① 提高开发速度：缩减或消除编译—链接—装载—测试周期中的一部分。

② 代码可移植性：使操作系统能为运行环境做系统级调用。

5) 建立类似多线程的运行机制，为运行不止一个活动线程的程序提供支持。

6) 提供下载代码模块机制，使得程序在运行时也能动态改变。

7) 提供一种代码安全性检查方法，为需要保证安全性的欲装载代码模块进行检查。

实际上，Java 虚拟机、自动垃圾收集、代码安全性这 3 种 Java 核心技术为上述目标的实现提供了基本保证。

1.1.2 Java 语言的发展

在 Java 语言发布之后，作为 Java 语言的最基本支持，Sun 公司在 1996 年发布了 Java 开发工具包 JDK 1.0 (Java Develop Kit, JDK)，其中包括了进行 Java 开发所需要的各种实用程序（编译、执行、文档生成器等）、基本类库（相当于 C 语言的函数库和 C++ 的类库）、程序实例等。当时 IT 领域的专业公司非常看好 Java 的前景，纷纷购买 Java 技术许可证，开发与 Java 技术匹配的产品。

1998 年，Sun 公司发布了更新的 JDK 1.2，由于在技术思想方面有很多改进，所以，此后的 Java 技术一般称为 Java 2。此时的 Java 不仅支持早先的消费类电子产品开发，其高级版本也支持大型企业级应用服务器开发。

随后，针对不同的领域特征，Sun 公司将 Java 2 技术分为 3 种不同的平台。

(1) J2SE——标准 Java 平台

J2SE (Java 2 Standard Edtion) 是 Java 语言的标准版，指的就是 JDK (1.2 及其以后版本)，包含 Java 基础类库和语法，适用于开发具有丰富图形用户界面 (GUI)、复杂逻辑和高性能的桌面应用程序。

(2) J2EE——企业级 Java 平台

J2EE (Java 2 Enterprise Edtion) 建立在 J2SE 之上，用于开发和实施企业级应用程序。它是一个标准的多层体系结构，可以将企业级应用程序划分为客户层、表示层、业务层和数据层，主要用于开发和部署分布式、基于组件、安全可靠、可伸缩和易于管理的企业级应用程序。

(3) J2ME——嵌入式 Java 技术平台

J2ME (Java 2 Micro Edtion) 也是建立在 J2SE 之上，主要用于开发具有有限的连接、内存和用户界面能力的设备应用程序，即针对早期 Java 支持的消费类电子产品开发，泛称为嵌入式开发。

Java 平台的最新称谓又去掉了意义模糊的 2，如 J2SE 称为 JavaSE。

Java 语言的学习者会逐渐发现 Java 技术体系的庞大。本书的内容仅相当于 Java 技术的入门，这是因为 Java 语言的发展目标并不是仅仅建立一种编程语言，同时还要构建一种开发环境、一种应用环境、一种部署环境。因此，Java 开发人员需要掌握更多的技术。

编程语言可以建立的应用程序类型与应用程序的运行环境有关，在多数情况下，不同的运行环境需要不同类型的应用程序，程序语言的学习者需要了解该语言可以建立哪些类型的应用程序。Java 语言的初学者首先需要了解 Applications 和 Applet 两种程序。

1) Applications 是一种独立的程序。它是一种典型的通用程序，可运行于任何具备 Java 运行环境的机器设备中。

2) Applet 是一种存储于 WWW 服务器上的用 Java 语言编写的程序。它通常嵌入在网页中，由浏览器下载到客户系统中，并通过浏览器运行。Applet 通常较小（俗称 Java 小程序），以减少下载时间。

Java 语言的发展使得 Java 一词不仅代表一种编程语言，也代表一种技术体系。在 Java 语言的发展过程中，基本语言的变化是极小的，主要是开发工具、中间件、框架技术、专用组件（类库）等外延技术的不断涌现和改进。

学习 Java 语言需要逐步了解 Java 技术体系，从局部细节开始学习，从整体中进行认识并选择方向，从应用中进行巩固提高。

JavaSE 作为 Java 语言的标准部分，是大多数 Java 教程的主要内容。学习 Java 语言首先要学会利用 Java 语言的标准知识去解决一般性问题，然后再学习 Java 技术体系中的其他内容。

1.2 Java 语言的特点

Java 语言适用于网络环境，是一种被广泛使用的网络编程语言，它具有如下特点。

1. 平台无关性

Java 虚拟机（JVM）是指在各种体系结构的真实机器中用软件模拟的一种想象机器，它们执行统一的指令，指令系统与底层机器的体系结构无关。

Java 编译器将 Java 应用程序的源代码文件（.java）翻译成 Java 字节码文件（.class），它是由 Java 虚拟机指令构成的。依赖于虚拟机技术，Java 语言具有与机器体系结构无关的特性，即 Java 程序一旦编写好之后，不需要进行修改就可以移植到任何一台体系结构不同的机器上。

2. 简单

Java 语言的语法规则和 C++ 类似，但取消了指针和多重继承等复杂结构，统一使用引用来指示对象（C++ 中有两种形式，实际上是两种产生对象的途径，而 Java 中只有一种），降低了编程的难度。Java 提供的自动垃圾收集功能也降低了程序设计的复杂性。

3. 完全面向对象

为了提高效率，Java 语言定义了几个基本数据类型以非类的方式实现，其他数据类型都以类的形式进行封装，程序系统的构成单位也是类，因此，可以认为是完全面向对象。

4. 面向网络编程

Java 语言产生之初就面向网络，在 JDK 中包括了支持 TCP/IP、HTTP 和 FTP 等协议 的类库，这些类库为 Java 语言提供了强大的网络编程支持。

5. 提供多线程机制

多线程机制能够使应用程序并发执行多项任务，其同步机制保证了各线程对共享数据 的正确操作。Java 在语言层面直接提供多线程支持。

6. 良好的代码安全性

Java 运行环境加载类的时候，使用字节码校验器执行代码校验，保证代码符合 JVM 规 范并且不破坏系统的完整性，例如检查伪造指针，违反对象访问权限，试图改变对象类型的 非法代码。这些校验操作可以保证 Java 语言具有良好的代码安全性。

7. 自动垃圾收集

大部分程序在运行时需要使用动态内存，许多编程语言都允许在程序运行时动态分配内 存块，但是大部分编程语言的内存分配和回收工作需要程序员在编程时进行，这样增加了编 程的复杂性，而且如果程序员不能有效地解决内存块释放工作，就会使程序产生内存漏洞。

Java 编程语言提供了一种系统级线程以跟踪存储区分配，用来完成自动垃圾收集，从而 可以把程序员从复杂的内存分配中解放出来。

8. 反射机制

反射是指类或对象能够自描述或自控制。Java 的反射机制允许程序在运行时访问类的内 部信息，适合于程序运行时加载分析一个编译时期完全未知的类。

1.3 面向对象与面向过程的比较

面向对象和面向过程是目前两种主流的编程思想，这两种编程思想在解决问题的着眼 点、对现实世界的描述方法、采用的程序开发方法和对应的编程语言方面均有差异。

面向过程是一种以事件为中心的编程思想，它着眼于事件发生的前后顺序，分析出解决 问题所需要的步骤，每个模块实现一个步骤，每个步骤对应一个事件，即事件之间的逻辑关 系。面向过程用流程图来描述现实世界中事件之间的逻辑关系。程序开发的主导方法还是控 制流或者数据流，即采用结构化开发方法。以 C 语言为代表的语言是典型的面向过程编程 语言。程序的表达形式是以过程为基本元素，程序的基本要素是模块——表达处理逻辑的 过程。

面向对象是一种以事物为中心的编程思想，它着眼于事物（对象）能产生的动作，事 物的属性等，强调的是对象之间的组织关系。面向对象用类图来描述现实世界对象之间的 关系。程序开发的主导方法是面向对象的开发方法。以 Java 为代表的语言是典型的面向对 象编程语言。语言的表达形式是以对象为基本元素，程序的基本要素是类——封装了属性 和方法的群体集合。

例如，简单探讨学生成绩分析这一问题。面向过程的编程思想着眼于设计成绩分析函 数的算法步骤，成绩只是与算法无明确关联约束的独立数据；而面向对象的程序设计着眼 于建立完整的、易于使用和修改的学生类，成绩是学生类的属性，分析算法是学生类的方法， 成绩和算法是明确关联的，二者构成一个整体，并且可以对外隐藏。