



普通高等教育“十二五”重点规划教材·计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

刘德山 金百东 张建华 主 编

Java程序设计

JAVA CHENGXU SHEJI



科学出版社

普通高等教育“十二五”重点规划教材·计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

Java 程序设计

刘德山 金百东 张建华 主 编
冯锡炜 刘江平 田英鑫 副主编

科 学 出 版 社

北 京

内 容 简 介

本书以通俗易懂的语言、翔实的案例、全新的内容组织来介绍 Java 语言。全书分为 11 章,包括 Java 语言概述、Java 语言基础知识、类与对象、继承与多态、数组和 Java 的常用类、集合类与泛型、异常处理、多线程、输入输出及文件操作、图形用户界面、网络和数据库编程等内容。本书内容以应用为核心展开,力求以知识的最小集来实现最大范围的应用。

教材的难度适中,“实用、易用、有效”是本书的写作目标。本书适合作为普通高等院校计算机及相关专业 Java 程序设计课程的教材,也可作为全国计算机等级考试或 Java 认证考试的参考教材。

本书配有辅助教材《Java 程序设计学习指导与习题解答》,进一步拓宽了本书内容和完善了实践环节。

图书在版编目(CIP)数据

Java 程序设计/刘德山,金百东,张建华主编. —北京:科学出版社,2012
(普通高等教育“十二五”重点规划教材·计算机系列)

ISBN 978-7-03-034464-9

I. ①J… II. ①刘… ②金… ③张… 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 108611 号

策划: 雋青龙

责任编辑: 雋青龙 / 责任校对: 刘玉靖

责任印制: 吕春珉 / 封面设计: 北京子时文化设计公司

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

骏主印刷厂印刷

科学出版社发行 各地新华书店经销

*

2012 年 6 月第 一 版 开本: 787×1092 1/16

2012 年 6 月第一次印刷 印张: 21 1/2

字数: 491 000

定价: 38.00 元

(如有印装质量问题,我社负责调换<骏杰>)

销售部电话 010-62142126 编辑部电话 010-62135517-2037

版权所有,侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

Java 语言的广泛应用，在于其大众性和专业性的特点。Java 语言具有面向对象、平台无关、安全机制、高可靠性和支持网络编程的特性，成为继 C 语言之后在学习计算机的不同专业的人群中使用最广泛的一种语言。其专业性表现在目前主流的软件开发平台 JavaEE 的基础也是 Java 语言。

学习 Java 语言的人群，不仅有软件开发、软件设计人员，而且包含非计算机专业人员。为了让读者能以最短的时间、最高的效率了解 Java 语言的核心和精髓，作者精选了大量的实例，编写了这套教材。目前，讲述 Java 语言的教材众多，也各有特色。本套教材的作者全部是多年主讲 Java 系列课程的教师，面向的读者主要是普通本科院校计算机及相关专业的学生，从“实用、易用、有效”的角度，来组织编写教材。本书的难度中等，内容以应用为核心展开，力求以知识的最小集来实现最大范围的应用。教材追求语言易懂、示例翔实、结构明晰的风格。

本书主要有以下特色。

1. 内容重点突出，同时加强与相关课程的整合。在保证内容科学、完整的前提下，由浅入深地安排章节次序。面向对象是教材的核心思想，在各章节的示例中，通过在不同层面强调面向对象程序设计思想的重要作用，处处引导学生用面向对象思想来思考问题、解决问题。

注重对其他课程的衔接和支持。例如，在讲解数组时，将排序与查找的算法穿插到教材中，对学习过数据结构的学生来说，是一种巩固；对没有学习过数据结构的学生来说，可为将来的学习打下基础；在讲解常用类的内容时，介绍如何使用 Java 类库中封装好的排序与查找方法，帮助读者了解 Java 语言的核心和实质。

还有，在讲解多态时，渗透设计模式的思想，在讲解 Swing 组件时，适当介绍了 MVC 模式，为学生后续课程的深入学习奠定了基础。

2. 教材配套，教学资源丰富。为了巩固教学内容和加强实践环节，编写了配套的习题与实验指导作为辅助教材。辅助教材源于主教材，又高于主教材，部分较难理解的内容和因版本更新而增加的知识，将在辅助教材中进行介绍。教材编写过程中，一直在进行教学资源的建设。本书提供教学课件、程序源码和网络课程。

3. 教材从内容上较好地处理了与全国计算机等级考试的关系和 Java 认证考试的关系。教材中的知识点基本覆盖了考试需要的核心内容，并对部分使用频率低的内容做了删减。

本书结构如下。

第 1 章和第 2 章是 Java 语言概述与 Java 语言基础知识，熟悉程序设计语言的读者可以略读。

第 3 章和第 4 章是面向对象程序设计，包括类和对象、继承与多态等内容，这部分是本书的核心之一，深入掌握这两章可为后面章节的学习打好基础。

第 5 章和第 6 章包括数组和 Java 的常用类、集合类与泛型。通过这两章的学习，读者应掌握 JDK 文档的查阅方法，了解 Java 编程的特点，学会使用各种 Java 类库，以提高编程效率。

第 7 章～第 9 章详细介绍了异常处理、多线程和输入输出及文件操作，这些都是 Java 编程的基础技术，是开发 Java 应用程序不可缺少的。异常、线程、I/O 都是对象，都存在着继承和多态，都存在着父类引用子类对象的问题，体现了面向对象编程的特点。

第 10 章和第 11 章介绍了图形用户界面、网络和数据库编程，这两章属于 Java 应用的范畴。教材详细介绍了数据库相关知识，加深了面向对象思想在数据库应用中的介绍，较同类教材的数据库部分内容介绍得更加详尽和实用，为学习其他 Java 课程打下很好的基础。

结合本书，建议教学的组织形式为理论—实例—练习—分析—练习。从应用的角度介绍理论，通过示例来说明编程的方法和过程。对理论、方法、技术的介绍力求概念明确、结构清晰、逻辑严谨。

本书适合作为计算机及相关专业 Java 程序设计课程的教材、Java 认证考试的教材或全国计算机等级考试教材。

本书由刘德山、金百东、张建华担任主编并负责全书的统稿和定稿工作，由冯锡炜、刘江平、田英鑫担任副主编，刘丹、孙美乔、王宇娇、张秋生参加了部分内容的编写工作。

由于编写时间仓促和水平有限，书中难免存在疏漏之处，敬请广大读者批评指正。

目 录

前 言

第 1 章 Java 语言概述	1
1.1 Java 语言简介	1
1.1.1 Java 语言的产生	1
1.1.2 Java 语言的版本	2
1.1.3 Java 语言的特点	3
1.1.4 Java 与 C++ 的比较	4
1.2 Java 程序示例	6
1.2.1 Java Application	6
1.2.2 Java Applet	7
1.3 Java 程序的开发和运行环境	10
1.3.1 JDK 的安装	10
1.3.2 设置环境变量	11
1.3.3 Java 开发环境	12
1.4 Java 程序的执行过程	15
习题	16
第 2 章 Java 语言基础知识	18
2.1 数据类型、常量与变量	18
2.1.1 Java 语言的数据类型	18
2.1.2 标识符、关键字、常量和变量	19
2.2 运算符和表达式	23
2.3 流程控制语句	29
2.3.1 程序的基本流程	29
2.3.2 分支结构	30
2.3.3 循环结构	33
2.3.4 跳转语句	36
2.4 程序注释与编码规范	38
2.4.1 程序注释	38
2.4.2 编码规范	38
2.5 综合示例	40
习题	42

第 3 章 类与对象	45
3.1 类概述	45
3.2 对象概述	45
3.3 类中变量	46
3.3.1 变量分类	46
3.3.2 变量初始化	48
3.4 类中方法	49
3.4.1 方法定义	49
3.4.2 方法的内存分配	50
3.4.3 构造方法	50
3.4.4 方法参数	54
3.4.5 方法重载	56
3.5 this 关键字	58
3.6 目录访问机制	59
3.6.1 classpath 存在的问题	59
3.6.2 package	59
3.6.3 import	61
3.7 访问权限	62
3.7.1 变量和方法的访问权限	62
3.7.2 类的访问权限	64
3.8 setor 与 getor	64
3.9 综合示例	65
习题	69
第 4 章 继承与多态	72
4.1 继承的概念	72
4.2 继承中的权限控制	72
4.3 子类对象构造过程	73
4.4 继承中方法重载与重写	75
4.4.1 重载与重写的含义	75
4.4.2 重写访问控制约束	75
4.5 super 关键字	76
4.6 多态	79
4.6.1 方法多态	79
4.6.2 逻辑多态思想	80
4.6.3 逻辑多态实现方法	80
4.6.4 进一步理解逻辑多态	85

4.7 instanceof 运算符	85
4.8 抽象类	87
4.9 接口	90
4.9.1 接口定义	90
4.9.2 接口实现	91
4.9.3 接口派生	93
4.10 深入理解普通类、抽象类、接口关系	93
4.10.1 三者关系	93
4.10.2 一个编程模型	94
4.10.3 多态在方法参数中的作用	95
4.11 内部类	98
4.11.1 内部类定义	98
4.11.2 内部类的作用	98
4.11.3 静态内部类	102
4.12 匿名类	105
4.12.1 匿名类定义	105
4.12.2 匿名类作用	107
4.13 final 类和 final 方法	109
习题	110
第 5 章 数组与 Java 的常用类	113
5.1 数组	113
5.1.1 一维数组	113
5.1.2 多维数组	115
5.1.3 复制数组	119
5.2 Java 基础类库	120
5.3 Object 类	122
5.4 字符串类	126
5.4.1 String 类	126
5.4.2 StringBuffer 类	129
5.4.3 StringTokenizer 类	132
5.5 Math 类	133
5.6 包装类	135
5.7 System 类	136
5.8 日期类	138
5.8.1 Date 类	139
5.8.2 Calendar 类及 GregorianCalendar 类	139

5.9 Random 类.....	140
5.10 综合示例.....	141
习题.....	144
第 6 章 集合类与泛型.....	148
6.1 集合框架与 Collection 接口.....	148
6.1.1 集合框架.....	148
6.1.2 Collection 接口.....	148
6.2 Set 接口及子类实现.....	149
6.3 List 接口及子类实现.....	151
6.3.1 List 接口.....	151
6.3.2 Vector 类与 Stack 类.....	152
6.4 Map 接口及子类实现.....	156
6.5 集合的遍历与 Iterator 接口.....	157
6.6 泛型.....	159
6.6.1 泛型类.....	159
6.6.2 泛型接口.....	160
6.6.3 泛型在集合框架中的应用.....	161
6.7 Collections 和 Arrays.....	162
6.7.1 Collections 类.....	162
6.7.2 Arrays 类.....	165
6.8 综合示例.....	168
习题.....	170
第 7 章 异常处理.....	173
7.1 异常处理概述.....	173
7.1.1 异常的概念.....	173
7.1.2 异常示例.....	174
7.1.3 异常类的层次结构.....	175
7.2 异常处理机制.....	176
7.2.1 捕获异常.....	176
7.2.2 声明异常.....	179
7.2.3 抛出异常.....	180
7.3 用户自定义异常类.....	181
习题.....	182
第 8 章 多线程.....	186
8.1 线程的概念.....	186
8.1.1 线程和进程.....	186

8.1.2	线程的运行机制	186
8.1.3	线程的优点	187
8.2	线程的创建	187
8.2.1	向 Thread 类传递 Runnable 对象创建线程	188
8.2.2	继承 Thread 类创建线程	189
8.2.3	两种方法的比较	190
8.3	Thread 类和多线程示例	190
8.3.1	Thread 类	190
8.3.2	多线程示例	191
8.4	线程的状态与控制	193
8.4.1	线程的状态	193
8.4.2	线程的控制	194
8.4.3	线程控制示例	196
8.4.4	线程的优先级	200
8.5	线程同步	202
8.5.1	多线程共享数据存在的问题	202
8.5.2	synchronized 关键字	203
8.6	线程通信	204
8.6.1	Java 线程通信机制	205
8.6.2	生产者—消费者问题	205
	习题	210
第 9 章	输入输出及文件操作	212
9.1	输入输出概述	212
9.1.1	流的层次	212
9.1.2	流的分类	214
9.2	字节流类	216
9.2.1	基本的输入输出示例	217
9.2.2	FileInputStream 和 FileOutputStream	218
9.2.3	BufferedInputStream 和 BufferedOutputStream	221
9.2.4	DataInputStream 和 DataOutputStream	223
9.3	字符流类	224
9.3.1	Reader 和 Writer	224
9.3.2	FileReader 和 FileWriter	225
9.3.3	BufferedReader 和 BufferedWriter	226
9.3.4	PrintWriter	229
9.4	File 类和 RandomAccessFile 类	231
9.4.1	File 类的方法	231

9.4.2	File 类的示例	232
9.4.3	RandomAccessFile 类	234
习题		236
第 10 章	图形用户界面	238
10.1	图形用户界面概述	238
10.1.1	AWT 组件与 Swing 组件	238
10.1.2	Swing 组件概述	238
10.2	几个常用容器与组件	239
10.2.1	JFrame 窗体	239
10.2.2	JPanel 中间容器	240
10.2.3	JButton 按钮	240
10.2.4	JLabel 标签	241
10.2.5	JTextField 编辑控件	241
10.2.6	JTextArea 文本区控件	241
10.3	布局管理器	243
10.3.1	流布局	243
10.3.2	方位布局	244
10.3.3	网格布局	245
10.3.4	卡片布局	245
10.3.5	盒式布局	246
10.3.6	null 布局	247
10.3.7	界面示例	248
10.4	事件处理	251
10.4.1	事件处理机制	251
10.4.2	事件处理内容	251
10.4.3	事件响应方法	252
10.4.4	事件响应的不同方式	254
10.4.5	事件适配器	257
10.5	其他常用组件	259
10.5.1	JCheckBox、JRadioButton 状态按钮	259
10.5.2	JComboBox 组合框	261
10.5.3	JList 列表控件	262
10.5.4	JTable 表格组件	267
10.5.5	JTree 树组件	271
10.5.6	菜单组件	274
10.6	其他常用中间容器	277
10.6.1	JScrollPane 滚动窗格	277
10.6.2	JSplitPane 拆分窗格	277

10.6.3 JTabbedPane 选项卡	278
10.7 对话框	279
10.8 绘图	283
10.8.1 绘制图形	283
10.8.2 颜色和字体控制	286
习题	287
第 11 章 网络和数据库编程	291
11.1 Socket 网络通信	291
11.1.1 概述	291
11.1.2 常用类	292
11.1.3 应用示例	292
11.2 UDP 数据报	299
11.2.1 概述	299
11.2.2 常用类	300
11.2.3 应用示例	300
11.3 URL 类	303
11.3.1 概述	303
11.3.2 常用方法	303
11.3.3 应用示例	304
11.4 数据库概述	304
11.4.1 JDBC 概述	304
11.4.2 MySQL 数据库	306
11.5 数据库操作	308
11.5.1 连接数据库	308
11.5.2 增加、删除和修改数据	310
11.5.3 查询	313
11.5.4 元数据应用	318
11.5.5 存储过程调用	322
11.6 数据库的中文乱码处理方法	323
11.7 面向对象思想在数据库操作中的应用	325
习题	328
参考文献	331

第 1 章 Java 语言概述

在大型企业级应用程序的开发中,JavaEE 是目前主流的软件开发平台之一。JavaEE 包含 Java Servlet、JSP、XML 等技术,其基础是 Java 语言。Java 语言是美国 Sun Microsystem 公司于 1995 年正式推出的、面向对象的程序设计语言。Java 语言具有简单、稳定、与平台无关、安全、解释执行、多线程等特点,除了用于开发大型程序,也可用于开发手机、数字机顶盒、汽车导航等各种嵌入式产品软件,是目前应用最为广泛的编程语言之一。

本章我们来了解 Java 语言的历史背景,学习第一个 Java 程序及 Java 的运行和编辑环境。

1.1 Java 语言简介

1.1.1 Java 语言的产生

Java 语言的前身是 Oak 语言。1991 年, Sun 公司为了寻找适合在消费电子产品上开发应用程序的编程语言,成立了由 James Gosling 和 Patrick Naughton 领导的 Green 研究小组。电子产品种类繁多,包括 PDA、机顶盒、手机等,存在不同类产品跨平台的问题。即使是同一类消费电子产品所采用的处理芯片和操作系统也不相同,同样存在跨平台的问题。起初, Green 小组考虑采用 C++ 语言来编写消费电子产品的应用程序,但是研究表明,对于消费电子产品而言, C++ 语言过于复杂和庞大,并不适用,安全性也并不令人满意。最后, Green 小组基于 C++ 开发出一种新的语言——Oak。该语言采用了许多 C 语言的语法,提高了安全性,并且是面向对象的语言。但是 Oak 语言在商业上并未获得成功。

随着互联网的蓬勃发展, Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全性高等特点,非常符合互联网的需要,于是转向互联网应用,进一步改进该语言的设计,并最终将这种语言命名为 Java。

1995 年, Sun 公司在 Sun World95 大会上正式发布了 Java 语言。IBM、Netscape、Novell、DEC、Oracle 和 Borland 等计算机公司纷纷表示支持 Java 语言。Java 语言被美国的著名杂志 *PC Magazine* 评为 1995 年十大优秀科技产品,随之出现了大量使用 Java 语言编写的软件产品,受到业界的好评。微软公司总裁比尔·盖茨说:“Java 语言是长时间以来最卓越的程序设计语言。”Sun 公司的总裁 Scott McNealy 认为, Java 为 Internet 和 WWW 开辟了一个崭新的时代。为了适应形势的需要, Sun 公司专门成立了一个子公司,即 SunSoft 公司,负责开发和推广 Java 技术。Sun 公司通过颁发许可证的方法来允许各家公司把 Java 虚拟机和 JavaApplet 类库嵌入到他们开发的操作系统中。这样,各类开发人员就能很容易地选择多种平台来使用 Java 语言编程,而且不同用户也可以脱离 Web 浏览器来运行 Java 应用程序。这些为 Java 语言的应用开拓了极为广阔的前景。

1.1.2 Java 语言的版本

Java 语言发展迅速, 1998 年, Sun 公司发表了 JDK 1.2 版本, 使用了新名称 Java 2 Platform, 即 Java 2 平台。为了区别不同方向的 Java 应用, Java2 平台分为 J2SE、J2EE、J2ME 三个版本, 并于 2005 年 6 月, 对各种版本更名, J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。Java 语言程序设计的主要内容就是 J2SE 或 JavaSE。Java 平台三个版本的关系如图 1-1 所示。

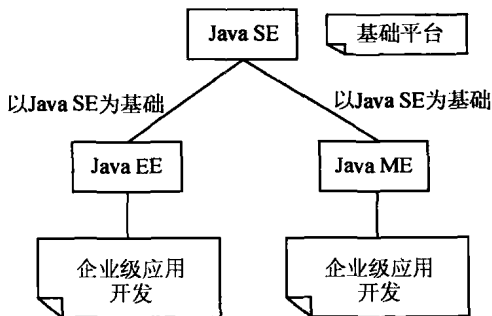


图 1-1 Java 平台三个版本的关系

1. Java SE

Java SE (Java Platform Standard Edition) 是标准版 Java 开发工具, 包含基本的 Java 开发工具。开发者可以用来编写、部署和运行 Java 应用程序 (Application) 和小程序 (Applet)。另外, 它还包括了早期 Java 开发工具包 JDK。Java SE 包含支持 Java Web 服务开发的类, 是开发大型 Java EE 应用程序的基础。

2. Java EE

Java EE (Java Platform Enterprise Edition) 是面向企业级应用的软件开发平台。它帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的, 它提供 Web 服务、组件模型、管理和通信 API, 涉及 COBRA、Java Servlets、JSP、XML 等技术, 可以用来实现企业级的面向服务体系结构 (Service-Oriented Architecture, SOA) 和 Web 2.0 应用程序。

3. Java ME

Java ME (Java Platform Micro Edition) 是为移动设备和嵌入式设备提供开发和运行的应用平台, 广泛应用于多种设备。例如, Java ME 应用于手机, 可以使手机具有灵活的电话簿编程、游戏、收发邮件、股票查询、电子地图等功能。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议, 还包括可以动态下载的联网和离线应用程序的丰富支持。基于 Java ME 规范的应用程序只需编写一次, 就可以用于许多设备, 而且可以利用每个设备的本机功能。

1.1.3 Java 语言的特点

根据 Sun 公司的“Java 白皮书”对 Java 的定义，Java 是一种简单易用、面向对象、分布式计算、解释执行、健壮性、安全性、结构无关性、可移植性、多线程、动态性的语言。

1. 简单易用

Java 语言作为面向对象的语言，通过提供最基本的方法来完成指定的任务，程序员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。另外，Java 语言自身系统非常精炼，它的基本解释器及类的支持只有 40KB 左右，加上标准类库和线程的支持也只有 200KB 左右，对硬件环境要求不高，在普通的计算机上就可以运行 Java 程序。

2. 面向对象

Java 语言是纯面向对象的编程语言，而不支持类似 C 语言那样面向过程的程序设计技术。Java 的设计集中于对象及接口，提供了简单的类机制以及动态的接口模型。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，实现了代码的复用。

3. 分布式计算

Java 是面向网络的语言，包括支持 HTTP 和 FTP 等基于 TCP/IP 协议的类库。通过它提供的类库，用户可以通过 URL 地址在网络上很方便地打开并访问其他对象。

4. 解释执行

Java 解释器（运行时系统）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，使得链接过程更加简单。

5. 健壮性

Java 程序运行时，提供自动垃圾回收机制来进行内存管理，不需要程序员管理内存。Java 提供异常处理机制，在编译时，Java 提示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的崩溃。另外，在编译时 Java 还可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

6. 安全性

Java 不支持指针和释放内存等操作，一切对内存的访问都必须通过对象的实例变量来实现，避免了非法内存操作和指针操作时容易产生的错误；类装载器加载类文件（.class 文件）到虚拟机时，需要进行安全检查；字节码校验器负责检查类文件代码中是否存在非法操作。上述方法保证了 Java 程序运行的安全性。

7. 结构无关性

Java 解释器生成与体系结构无关的字节码指令,在安装了 Java 运行时系统的计算机上,Java 字节码文件可以跨平台运行。

8. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器上。同时,Java 的类库也实现了对不同平台的接口,使这些类库可以移植。另外,Java 编译器是由 Java 语言实现的,而 Java 运行时系统由标准 C 语言实现,这使得 Java 系统本身也具有可移植性。

9. 多线程

多线程机制使应用程序能够并行执行,而且同步机制保证了对共享数据的正确操作。通过使用多线程,程序设计者可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制,这样就很容易地实现网络上的实时交互行为。

10. 动态性

Java 的动态特性是其面向对象方法的扩展,使得它适合于一个不断发展的环境。在类库中,可以自由地加入新的方法和实例变量而不影响用户程序的执行。并且 Java 通过接口来支持多重继承,使类继承具有更灵活的方式和扩展性。

1.1.4 Java 与 C++的比较

Java 使用了类似于 C++的语法,而去除了 C++中许多不合理的内容,以实现其简单、健壮、安全等特性。

1. 语言形式

C++为编译型语言,而 Java 为半编译半解释型语言。

2. 平台无关

C++编写的程序与平台相关,而 Java 语言程序与平台无关,该平台无关是通过字节码和 Java 虚拟机实现的。

3. 异常处理

与 C++类似,Java 的异常处理通过语句 `try-catch-finally` 来实现,使程序更加结构化并提高了程序的可读性。

4. 运算符重载

C++支持运算符重载,而 Java 不允许运算符重载。

5. 全局变量

Java 程序中不能定义全局变量，只能通过类中的公用、静态的变量实现全局变量。这样便保证了更好的安全性，全局变量被封装在类中。而在 C++ 中，依赖于不加封装的全局变量会造成系统的崩溃。

6. 指针

指针是 C++ 中最灵活，但也是最容易出错的数据类型。以指针进行内存操作常造成不可预知的错误。而且，通过指针对内存地址进行显式类型转换后，可以访问类的私有成员，破坏了安全性。在 Java 中，程序员不能进行任何指针操作。同时，数组在 Java 中用类来实现，很好地解决了数组越界这一 C++ 中不做检查的错误。

7. 内存管理

在 C++ 中，程序员用运算符 `new` 和 `delete` 来分配和释放内存。再次释放已释放的内存块或释放未被分配的内存块，会造成系统的崩溃，而忘记释放不再使用的内存块也会逐渐耗尽系统资源。在 Java 中，通过运算符 `new` 分配内存并得到对象的使用权，Java 自动地进行管理并进行自动垃圾收集，有效地防止了因程序员误操作而引起的错误，并更好地利用了系统资源。

8. 数据类型的一致性

在 C++ 中，不同的平台上，编译器对简单的数据类型如 `int`、`float` 等分配不同的字节数。例如，`int` 在 IBM PC 上为 16 位，在 VAXII 上为 32 位，因此导致了代码的不可移植。而在 Java 中，对某种数据类型总是分配固定的位数，这就保证了 Java 的平台无关性和可移植性。

9. 多继承的实现

在 C++ 中允许多重继承，而 Java 通过接口来实现多继承的功能，类的继承只能是单继承。

10. 头文件

在 C++ 中，使用头文件声明类的原型和全局变量及库函数等，而在大的系统中，维护这些头文件非常困难。Java 不支持头文件，类成员的类型和访问权限都封装在一个类中，运行时系统以此对访问进行控制，防止对私有成员的操作。另外，Java 中用 `import` 语句与其他类进行通信，以使用它们的方法。

11. 结构和联合

C++ 的结构和联合的成员均为公有，带来了安全性的问题。Java 不支持结构和联合，所有的内容封装在类里。