

北京大学信息技术系列教材 蔡翠平 主编

BEIJING DAXUE XINXI JISHU XILIE JIAOCAI

全国高校出版社优秀畅销书

Java 程序设计

(第2版) 

■ 唐大仕 编著



清华大学出版社
<http://www.tup.com.cn>



北京交通大学出版社
<http://www.bjtup.com.cn>



北京大学信息技术系列教材

Java 程序设计

(第2版)

唐大仕 编著

清华大学出版社

北京交通大学出版社

• 北京 •

内 容 简 介

本书详细介绍了 Java 程序设计的基本环境、概念、方法和应用。内容分为三个部分：第一部分介绍了 Java 语言基础，包括数据、控制结构、数组、类、包、对象、接口等；第二部分介绍了 Java 深入知识，包括传值调用、虚方法调用、异常处理、Lambda 表达式、流式处理；第三部分是 Java 的应用，包括工具类与算法、线程、文件、AWT 及 Swing 图形用户界面，以及 Java 在网络、多媒体、数据库等方面的应用。本书对 JDK7 及 JDK8 中的新内容也进行了介绍。

本书内容详尽，循序渐进，在介绍编程技术的同时，还着重讲解了有关面向对象程序设计的基本概念和方法。书中提供了丰富的典型实例，具有可操作性，便于读者的学习与推广应用。各章附有习题，便于读者思考和复习。

本书内容和组织方式立足于高校教学教材的要求，同时可作为计算机技术的培训教材。本书也适用于作者在 coursera 上开设的慕课（MOOC）“Java 程序设计”。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13501256678 13801310933

图书在版编目（CIP）数据

Java 程序设计 / 唐大仕编著. —2 版. —北京：北京交通大学出版社：清华大学出版社，2015.2
（北京大学信息技术系列教材）

ISBN 978-7-5121-2189-8

I. ①J… II. ①唐… 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（2015）第 017525 号

责任编辑：谭文芳

出版发行：清华大学出版社	邮编：100084	电话：010-62776969	http://www.tup.com.cn
北京交通大学出版社	邮编：100044	电话：010-51686414	http://www.bjtup.com.cn

印刷者：北京交大印刷厂

经 销：全国新华书店

开 本：185×260 印张：24.25 字数：680 千字

版 次：2015 年 3 月第 2 版 2015 年 3 月第 1 次印刷

书 号：ISBN 978-7-5121-2189-8/TP·802

印 数：1~3 000 册 定价：45.00 元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。

投诉电话：010-51686043, 51686008；传真：010-62225406；E-mail: press@bjtu.edu.cn。

第 2 版前言

在程序设计的教学中，选择一种适合的语言是十分重要的。比较多种程序设计语言，笔者认为 Java 具有如下突出的优点。

其一，Java 是面向对象的语言，与现代面向对象的设计与分析的软件工程相一致，也是当前的主程序设计语言之一。

其二，简单易学。其中的数据类型、数据运算、程序控制结构等基本概念对于任何语言都是一致的；而其语法相对于 C++ 等语言而言更简单，更容易掌握。

其三，Java 语言本身就支持一些高级特性，如自动垃圾回收、异常处理、多线程、并行计算等，这些特性使 Java 成为极优秀的语言之一。

此外，Java 具有广泛的用途。Java 具有跨平台的特点，在各种平台上都有应用，它还可以有效地进行数据库、多媒体及网络的程序设计。不仅如此，Java 还可以编写桌面应用、手机应用及网络应用。

综上所述，Java 是特别适合于程序设计学习的基础语言。

对于学习者而言，选择一本好的教材至关重要。笔者基于多年程序设计语言的教学经验，结合个人的软件开发实践，力图使本书突出以下特色。

1. 对 Java 语言的基础知识，包括数据类型、流程控制、类的封装与继承、多态、虚方法调用、传值调用等进行系统讲解，让学习者知其然，并知其所以然。

2. 对 Java 中的类库中的基本类，包括 Math、字符串、集合进行详细讲解，以利于学习者打下牢固的基础。

3. 对 Java 中的基本应用，包括 I/O、文本界面、图形界面等，精选大量典型而实用的例子，力图使学习者触类旁通，举一反三。

4. 对一些高级应用，如数据库编程、网络编程、多媒体编程、多线程、并行计算等内容，介绍了其概念、原理，以利于学习者能了解 Java 的实际应用及最新发展。

5. 在讲解语言的同时，介绍它所采用的面向对象技术的基础理论、主要原则和思维方法，以及在编程中写好代码的一些技巧与经验。

6. 在讲解、举例时充分考虑到各个层次的需要，力求语言简洁，内容循序渐进。同时，考虑到部分读者的需要，本书提供了较多的习题。

本书在内容安排上，大致可以分为三部分：第一部分介绍了 Java 语言基础，包括数据、控制结构、数组、类、包、对象、接口等；第二部分介绍了 Java 深入知识，包括传值调用、虚方法调用、异常处理、工具类与算法；第三部分是 Java 的应用，包括线程、并行编程、流式文件、AWT 及 Swing 图形用户界面，以及 Java 在网络、多媒体、数据库等方面的应用。

本书历经了第 1 版、修订版，这是第 2 版，书中融合了多年的教学经验，也同时针对广大读者的反馈进行了改进。另外，本书对 Java5 至 Java8 的新特性进行了介绍，如 Java5 中的泛型、增强的 for 语句、装包等，Java7 中增加的如 switch 与字符串，自动尝试关闭资源、新

目 录

第 1 章 Java 语言与面向对象的程序设计	1
1.1 Java 语言简介	1
1.1.1 Java 语言的发展	1
1.1.2 Java 三大平台	2
1.1.3 Java 语言的特点	3
1.1.4 Java 和 C、C++ 比较	5
1.2 面向对象程序设计	6
1.2.1 面向对象概述	6
1.2.2 对象、类与实体	7
1.2.3 对象的状态与行为	7
1.2.4 对象的关系	8
1.2.5 面向对象的软件开发过程	9
习题	10
第 2 章 简单的 Java 程序	11
2.1 Application 与 Applet 程序	11
2.1.1 Application 程序	11
2.1.2 Applet 程序	12
2.1.3 Java 程序的基本构成	13
2.2 程序的编辑、编译与运行	15
2.2.1 Java 工具包 JDK	15
2.2.2 Application 的编辑、编译与运行	15
2.2.3 Applet 的编辑、编译与运行	18
2.2.4 使用 jar 打包程序	20
2.3 Java 程序中的基本输入与输出	21
2.3.1 字符界面的输入与输出	21
2.3.2 Applet 图形界面输入与输出	23
2.3.3 Application 图形界面输入与输出	24
2.3.4 同时作为 Application 与 Applet 的程序	26
2.4 Java 集成开发环境	27
2.4.1 几种辅助工具的使用	27
2.4.2 几种集成工具的使用	29
2.4.3 建构工具及代码混淆	33
习题	34

第3章 数据运算、流程控制和数组	35
3.1 数据类型、变量与常量	35
3.1.1 数据类型	35
3.1.2 标识符	36
3.1.3 常量	36
3.1.4 变量	37
3.1.5 程序的注释	38
3.2 运算符与表达式	39
3.2.1 算术运算符	40
3.2.2 关系运算符	42
3.2.3 逻辑运算符	42
3.2.4 位运算符	43
3.2.5 赋值与强制类型转换	47
3.2.6 条件运算符	48
3.2.7 表达式及运算的优先级、结合性	48
3.3 流程控制语句	49
3.3.1 结构化程序设计的三种基本流程	49
3.3.2 简单语句	50
3.3.3 分支语句	50
3.3.4 循环语句	56
3.3.5 跳转语句	61
3.4 数组	63
3.4.1 一维数组	63
3.4.2 多维数组	67
3.4.3 数组与增强的 for 语句	69
3.4.4 数组的复制	69
习题	69
第4章 类、包和接口	71
4.1 类、字段、方法	71
4.1.1 定义类中的字段和方法	71
4.1.2 构造方法与对象的创建	73
4.1.3 使用对象	74
4.1.4 方法的重载	74
4.1.5 this 的使用	76
4.2 类的继承	77
4.2.1 派生子类	77
4.2.2 字段的继承、隐藏与添加	78
4.2.3 方法的继承、覆盖与添加	78
4.2.4 super 的使用	79

4.2.5	父类对象与子类对象的转换	80
4.3	包	83
4.3.1	package 语句	83
4.3.2	import 语句	84
4.3.3	编译和运行包中的类	85
4.3.4	CLASSPATH 变量	85
4.4	访问控制符	86
4.4.1	成员的访问控制符	86
4.4.2	类的访问控制符	90
4.4.3	setter 与 getter	91
4.4.4	构造方法的隐藏	91
4.5	非访问控制符	91
4.5.1	static	92
4.5.2	final	94
4.5.3	abstract	96
4.5.4	其他修饰符	98
4.5.5	一个应用模式——单子	98
4.6	接口	99
4.6.1	接口的概念	99
4.6.2	定义接口	100
4.6.3	实现接口	101
4.6.4	对接口的引用	102
4.6.5	Java8 对接口的扩展	103
4.7	枚举	103
4.7.1	枚举的基本用法	103
4.7.2	枚举的深入用法	104
	习题	105
第 5 章	深入理解 Java 语言	106
5.1	变量及其传递	106
5.1.1	基本类型变量与引用型变量	106
5.1.2	字段变量与局部变量	107
5.1.3	变量的传递	108
5.1.4	变量的返回	109
5.1.5	不定长参数变量	109
5.2	多态与虚方法调用	110
5.2.1	上溯造型	110
5.2.2	虚方法调用	111
5.2.3	动态类型确定	113
5.3	对象构造与初始化	114

5.3.1	调用本类或父类的构造方法	114
5.3.2	构造方法的执行过程	117
5.3.3	构造方法内部调用的方法的多态性	118
5.3.4	实例初始化与静态初始化	119
5.4	对象清除与垃圾回收	120
5.4.1	对象的自动清除	120
5.4.2	System.gc()方法	121
5.4.3	finalize()方法	121
5.5	内部类与匿名类	122
5.5.1	内部类	123
5.5.2	方法中的局部类及匿名类	126
5.5.3	匿名类	127
5.6	Lambda 表达式与函数式接口	128
5.6.1	Lambda 表达式的书写与使用	128
5.6.2	函数式接口	129
5.6.3	高阶函数	132
5.7	注解与反射	133
5.7.1	注解的定义与使用	133
5.7.2	反射	135
	习题	140
第 6 章	异常处理	142
6.1	异常处理	142
6.1.1	异常的概念	142
6.1.2	捕获和处理异常	144
6.1.3	应用举例	147
6.2	创建用户自定义异常类	149
6.2.1	自定义异常类	149
6.2.2	重抛异常及异常链接	151
6.3	异常与资源管理	153
6.3.1	使用 finally	153
6.3.2	使用 try...with...resource	153
6.4	断言及程序的测试	154
6.4.1	使用 assert	155
6.4.2	程序的测试及 JUnit	155
	习题	156
第 7 章	工具类及常用算法	157
7.1	Java 语言基础类	157
7.1.1	Java API	157
7.1.2	Object 类	159

7.1.3	基本数据类型的包装类	162
7.1.4	Math 类	163
7.1.5	System 类	164
7.2	字符串和日期	165
7.2.1	String 类	165
7.2.2	StringBuffer 类	169
7.2.3	StringTokenizer 类	170
7.2.4	日期相关类	171
7.3	集合类	173
7.3.1	集合与 Collection API	173
7.3.2	Set 接口及 HashSet、TreeSet 类	174
7.3.3	List 接口及 ArrayList、Vector 类	175
7.3.4	栈与队列	177
7.3.5	键-值集合及 Map 接口	179
7.4	泛型及集合遍历	181
7.4.1	泛型	181
7.4.2	装包与拆包	184
7.4.3	Iterator 及 Enumeration	184
7.4.4	集合与增强的 for 语句	186
7.5	排序与查找	188
7.5.1	使用 Arrays 类	188
7.5.2	使用 Collections 类	190
7.5.3	编写排序程序	193
7.6	遍试、迭代、递归及回溯	196
7.6.1	遍试	197
7.6.2	迭代	199
7.6.3	递归	201
7.6.4	回溯	206
	习题	207
第 8 章	线程与并行编程	210
8.1	线程的创建与运行	210
8.1.1	Java 中的线程	210
8.1.2	创建线程对象的两种方法	211
8.1.3	多线程	213
8.1.4	使用 Timer 类	215
8.1.5	应用举例	216
8.2	线程的控制与同步	220
8.2.1	线程的状态与生命周期	220
8.2.2	对线程的基本控制	221

8.2.3	synchronized 关键字	226
8.2.4	线程间的同步控制	229
8.3	并行 API	234
8.3.1	并行集合	234
8.3.2	原子变量	236
8.3.3	读写锁	237
8.3.4	Executor 与 Future	238
8.4	流式操作	240
8.4.1	使用流的基本方法	240
8.4.2	流及操作的种类	243
	习题	245
第 9 章	流、文件及基于文本的应用	246
9.1	流式输入与输出	246
9.1.1	字节流与字符流	246
9.1.2	节点流和处理流	248
9.1.3	标准输入和标准输出	250
9.1.4	文本文件及二进制文件流应用示例	252
9.1.5	对象序列化	255
9.2	文件及目录	257
9.2.1	文件与目录管理	257
9.2.2	使用 NIO2 文件系统 API	259
9.2.3	文件输入与输出流	261
9.2.4	RandomAccessFile 类	263
9.3	基于文本的应用	265
9.3.1	Java Application 命令行参数	265
9.3.2	环境参数	266
9.3.3	处理 Deprecated 的 API	267
9.4	正则表达式	268
9.4.1	正则表达式的基本元素	268
9.4.2	Pattern 及 Matcher	269
9.5	XML 处理	271
9.5.1	XML 基本概念	271
9.5.2	XML 编程	274
	习题	279
第 10 章	图形用户界面	280
10.1	界面组件	280
10.1.1	图形用户界面概述	280
10.1.2	界面组件分类	281
10.1.3	Component 的方法	283

10.2	布局管理	284
10.2.1	FlowLayout	284
10.2.2	BorderLayout	285
10.2.3	CardLayout	286
10.2.4	GridLayout	287
10.2.5	GridBagLayout	288
10.2.6	通过嵌套来设定复杂的布局	288
10.3	事件处理	289
10.3.1	事件及事件监听器	289
10.3.2	事件监听器的注册	291
10.3.3	事件适配器	294
10.3.4	内部类及匿名类在事件处理中的应用	295
10.4	常用组件的使用	297
10.4.1	标签、按钮与动作事件	298
10.4.2	文本框、文本区域与文本事件	299
10.4.3	单、复选按钮，列表与选择事件	300
10.4.4	调整事件与滚动条	304
10.4.5	鼠标、键盘事件与画布	306
10.4.6	Frame 与窗口事件	307
10.4.7	Panel 与容器事件	308
10.4.8	组件事件、焦点事件与对话框	310
10.5	绘图、图像和动画	311
10.5.1	绘制图形	311
10.5.2	显示文字	314
10.5.3	控制颜色	315
10.5.4	显示图像及实现动画	316
10.6	Applet	318
10.6.1	Applet 的基本工作原理	318
10.6.2	Applet 类	319
10.6.3	HTML 文件参数传递	320
10.6.4	Applet 的局限	321
10.7	Swing 组件的使用	326
10.7.1	Swing 的特点	326
10.7.2	几种 Swing 组件介绍	327
10.7.3	Swing 组件与线程	329
10.8	基于 GUI 的应用程序	330
10.8.1	使用可视化设计工具	330
10.8.2	菜单的定义与使用	331
10.8.3	菜单、工具条及对话框的应用	334

习题	340
第 11 章 网络、多媒体和数据库编程	342
11.1 Java 网络编程	342
11.1.1 使用 URL	342
11.1.2 用 Java 实现底层网络通信	344
11.1.3 实现多线程服务器程序	346
11.1.4 Java 的 E-mail 编程	353
11.1.5 使用第三方的库	358
11.2 多媒体编程	358
11.2.1 在 Applet 中获取声音与图像	358
11.2.2 Java 图像编程	360
11.2.3 Java 声音编程	362
11.2.4 JMF 简介	362
11.3 Java 数据库编程	363
11.3.1 Java 访问数据库的基本步骤	363
11.3.2 使用 JTable 显示数据表	367
11.4 Java EE 及 Java ME 简介	370
11.4.1 Java EE 简介	371
11.4.2 Java ME 简介	372
习题	374
附录 A Java 语言各版本增加的重要特性	375
参考文献	376

第 1 章 Java 语言与面向对象的程序设计

Java 语言是当今流行的网络编程语言，它的面向对象、跨平台、分布应用等特点给编程人员带来了一种崭新的计算概念，使 WWW 从最初的单纯提供静态信息发展到现在的提供各种各样的动态服务，产生了巨大的变化。Java 不仅能够编写小应用程序实现嵌入网页的声音和动画功能，而且还能够应用于独立的大中型应用程序，其强大的网络功能能够把整个 Internet 作为一个统一的运行平台，极大地拓展了传统单机或 Client/Server 模式应用程序的外延和内涵。自 1995 年正式问世以来，Java 已经逐步从一种单纯的计算机高级编程语言发展为一种重要的 Internet 平台，并进而引发、带动了 Java 产业的发展壮大，成为当今计算机业界不可忽视的力量和重要的发展潮流与方向。

1.1 Java 语言简介

1.1.1 Java 语言的发展

1991 年，SUN MicroSystem 公司的 James Gosling, Bill Joe 等人，为在电视机、控制烤箱包箱等家用消费类电子产品上进行交互式操作而开发了一个名为 Oak（一种橡树的名字）的软件，但当时并没有引起人们的注意，直到 1994 年下半年，Internet 的迅猛发展，WWW 的快速增长，促进了 Java 语言研制的进展，使得它逐渐成为 Internet 上受欢迎的开发与编程语言，一些著名的计算机公司纷纷购买了 Java 语言的使用权，如 Microsoft, IBM, Netscape, Novell, Apple, DEC, SGI 等。因此，Java 的诞生对整个计算机产业产生了深远的影响，可以说，Java 为 Internet 和 WWW 开辟了一个崭新的时代。

Java 对传统的计算模型提出了新的挑战。业界不少人预言：“Java 语言的出现，将会引起一场软件革命。”这是因为传统的软件往往都是与具体的实现环境有关，而 Java 语言能在执行码（二进制码）上兼容，这样，以前所开发的软件就能运行在不同的机器上，只要所用的机器能提供 Java 语言解释器即可。

Java 语言将对未来软件的开发产生影响，可反映在如下几个方面。

（1）软件的需求分析。可将用户的需求进行动态的、可视化描述，以满足设计者更加直观的要求。Java 语言不受地区、行业、部门、爱好的限制，都可以将用户的需求描述清楚。

（2）软件的开发方法。由于 Java 语言的面向对象的特性，所以完全可以用面向对象的技术与方法来进行开发，符合最新的软件开发规范的要求。

（3）Java 语言的动态效果。就界面而言，GUI 技术达到动画效果；就数据而言，Java 能根据数据动态地提供信息。

（4）软件最终产品。用 Java 语言开发的软件可以具有可视化、可听化、可操作化的效果，其多媒体应用也十分广泛。

（5）其他。使用 Java 语言进行开发对开发效益、开发价值都有比较明显的影响。

正如 Java 的创始人之一 James Gosling 所说, Java 不仅仅只是 applets, 它能做任何事情: Java 不仅仅是一种程序设计语言, 更是现代化软件再实现的基础; Java 还是未来新型 OS 的核心; 将会出现 Java 芯片; Java 将构成各种应用软件的开发平台与实现环境, 是人们必不可少的开发工具。

因此, Java 语言有着广泛的应用前景, 包括面向对象的应用开发, 计算过程的可视化、可操作化的软件的开发, 动态画面的设计, 图形图像的调用, 交互操作的设计, Internet 的系统管理功能, 直接面向企业内部用户的软件, 与各类数据库连接查询, 等等。事实上, Java 语言已经是各类应用开发的主流语言之一。

Java 语言从 1995 年诞生以来, 经历了多次版本, 其中比较重要的版本是:

- ✎ JDK1.1.4, 1997 年 9 月发布;
- ✎ J2SE1.2, 1998 年 12 月发布 (从这时起, Java 平台又叫 Java2 平台);
- ✎ J2SE1.3, 2000 年 5 月发布;
- ✎ J2SE1.4, 2002 年 2 月发布 (这是应用很广泛的版本);
- ✎ J2SE5.0, 2004 年 9 月发布 (从这时起, Java 版本从 1.5 改称为 5.0);
- ✎ J2SE6.0, 2006 年 12 月发布 (这是现行的 Java 程序都需要的最基本版本);
- ✎ J2SE7.0, 2011 年 7 月发布 (2010 年, Oracle 宣布并购 Sun);
- ✎ J2SE8.0, 2014 年 3 月发布 (这是 Java 语言进步比较大的版本)。

本书中主要以各版本共同的语法为主, 也兼顾最新的 Java 版本中新增的语法。关于 Java 语言的规范可以参考 <http://docs.oracle.com/javase/specs/>。

主导 Java 语言特性及版本发展的主要组织是 JCP (Java Community Process), 这是一个开放组织, 其官方网址是 <http://jcp.org>。任何提议加入 Java 的功能及特性, 首先以 JSR (Java Specification Requests) 正式文件的方式提交给 JCP, 由 JCP 的执行委员会投票成为 Java 标准的一部分。比如 Java7 的规范主要是 JSR336 中体现的, Java8 的规范则包括了 JSR335 及 JSR310 等。

1.1.2 Java 三大平台

在 Java 发展过程中, 由于 Java 语言的应用越来越广, 根据不同级别的应用开发区分了不同的应用版本, Java SE (Java Standard Edition, Java 标准版), Java EE (Java Enterprise Edition, Java 企业版) 及 Java ME (Java Micro Edition, Java 微型版), 由于历史的原因, 它们也被称为 J2SE, J2EE 及 J2ME。

Java SE 是各种应用平台的基础, 编写桌面应用很方便; Java EE 是在 Java SE 基础上增加一系列服务, 适合于编写 Web 应用; Java ME 是三种平台中最小的一个, 是小型数字设备 (包括手机、PDA 等) 开发和运行的平台。本书主要介绍 Java SE。

Java SE 主要包括 JVM、Java SE API、JRE、JDK 等部分, 如图 1-1 所示。

其中 JVM, 即 Java 虚拟机 (Java Virtual Machine), 是运行 Java 字节码程序 (.class 文件) 的“操作系统”, 它在实际的操作系统 (Windows, Linux 等) 上实际装入、即时编译、并运行字节码程序的系统, 就像一个虚拟的机器。

Java SE API 是由一系列的类库 (库函数) 组成的, 它为编写程序提供了基础的 API, 使得程序员不必从最底层写代码。

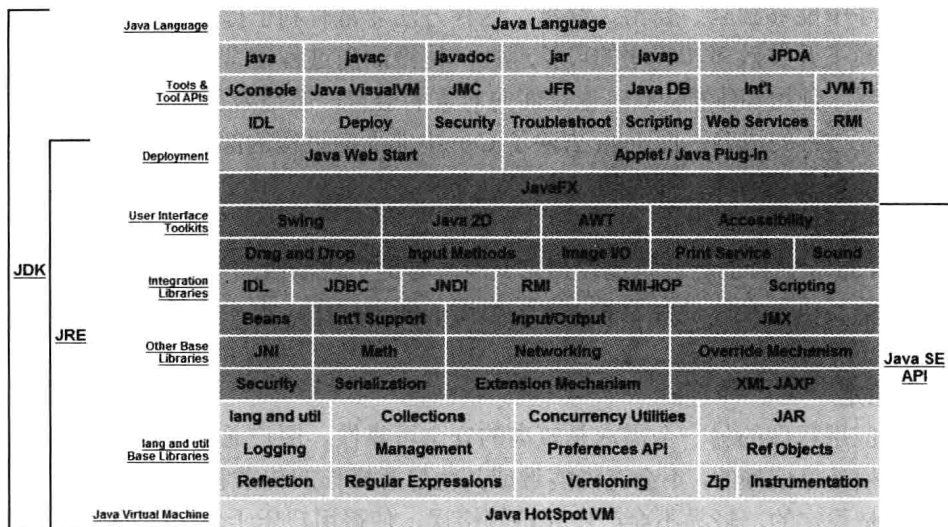


图 1-1 Java SE 的组成

JRE 即 Java 运行环境 (Java Runtime Environment)，包括了 JVM 及 Java SE API，如果要运行 Java 程序，就需要在操作系统中下载并安装 JRE。访问 <http://www.java.com> 可以进行 JRE 的下载。

JDK 即 Java 开发工具包 (Java Development Kit)。是用来开发 Java 程序的。它包括 JRE，同时还包括编译器等开发工具，可以将 Java 的源程序编译成 Java 目标程序 (字节码)。为了开发 Java 程序，需要下载 JDK，可以访问网站 <http://java.sun.com> (即 <http://www.oracle.com/technetwork/java/index.html>)。关于 JDK 的使用，第 2 章会详细讲到。

1.1.3 Java 语言的特点

简单地说，Java 是定位于网络计算的计算机语言，它的几乎所有的特点也是围绕着这一中心展开并为之服务的，这些特点使得 Java 语言特别适合作为开发网络上的应用程序。另外，作为一种问世较晚的语言，Java 也集中体现和充分利用了若干当代软件技术新成果，如面向对象、多线程等，这些也都在它的特点中有所反映。Java 的主要特点如下。

1. 简单易学

衍生自 C++ 的 Java 语言，出于安全稳定性的考虑，去除了 C++ 中不容易理解和掌握的部分，如最典型的指针操作等，降低了学习的难度；同时，Java 还有一个特点就是它的基本语法部分与 C 语言几乎一模一样。这样，无论是掌握了 Java 再学 C 语言，还是已经掌握了 C 语言再来学 Java，都会感到易于入门。

2. 面向对象

Java 是面向对象的编程语言。面向对象技术较好地解决了当今软件开发过程中新出现的种种传统面向过程语言所不能处理的问题，包括软件开发的规模扩大、升级加快、维护量增大，以及开发分工日趋细化、专业化和标准化等，是一种迅速成熟、推广的软件开发方法。面向对象技术的核心是以更接近于人类思维的方式建立计算机逻辑模型，它利用类和对象的机制将数据与其上的操作封装在一起，并通过统一的接口与外界交互，使反映现实世界实体

的各个类在程序中能够独立、自治、继承。这种方法非常有利于提高程序的可维护性和可重用性，大大提高了开发效率和程序的可管理性，使得面向过程语言难于操纵的大规模软件可以很方便地创建、使用和维护。C++也是面向对象的语言，但是为了与 C 语言兼容，其中还包含了一些面向过程的成分；而 Java 去除了 C++中非面向过程的部分，其程序编写过程就是设计、实现类，定义其属性、行为的过程。

3. 平台无关性

如前所述，Java 独特的运行机制使得它具有良好的二进制级的可移植性，利用 Java 语言，开发人员可以编写出与具体平台无关、普遍适用的应用程序，大大降低了开发、维护和管理

4. 安全稳定

对网络上应用程序的另一个需求是较高的安全可靠。用户通过网络获取并在本地运行的应用程序必须是可信赖的，不会充当病毒或其他恶意操作的传播者而攻击用户本地的资源；同时它还应该是稳定的，轻易不会产生死机等错误，使得用户乐于使用。Java 特有的机制是其安全性的保障，同时它去除了 C++中易造成错误的指针，增加了自动内存管理等措施，保证了 Java 程序运行的可靠性。

5. 支持多线程

多线程是当今软件技术的又一重要成果，已成功应用在操作系统、应用开发等多个领域。多线程技术允许同一个程序有两个执行线索，即同时做两件事情，满足了一些复杂软件的需求。Java 不但内置多线程功能，而且提供语言级的多线程支持，即定义了一些用于建立、管理多线程的类和方法，使得开发具有多线程功能的程序变得简单、容易和有效。

6. 很好地支持网络编程

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。Java 的小应用程序（Applet）是动态、安全、跨平台的网络应用程序。Java Applet 嵌入 HTML 语言，通过主页发布到 Internet。网络用户访问服务器的 Applet 时，这些 Applet 从网络上进行传输，然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制，用户一旦载入 Applet，就可以放心地生成多媒体的用户界面或完成复杂的计算而不必担心非法入侵。虽然 Applet 可以和图像、声音、动画等一样从网络上下载，但它并不同于这些多媒体的文件格式，它可以接收用户的输入，动态地进行改变，而不仅仅是动画的显示和声音的播放。

7. Java 丰富的类库

Java 提供了大量的类库以满足网络化、多线程、面向对象系统的需要。

(1) 语言包提供的支持包括字符串处理、多线程处理、例外处理、数学函数处理等，可以用它简单地实现 Java 程序的运行平台。

(2) 实用程序包提供的支持包括哈希表、堆栈、可变数组、时间和日期等。

(3) 输入输出包用统一的“流”模型来实现所有格式的 I/O，包括文件系统、网络及输入/输出设备等。

(4) 低级网络包用于实现 Socket 编程。

(5) 抽象图形用户接口包实现了不同平台的计算机的图形用户接口部件，包括窗口、菜单、滚动条和对话框等，使得 Java 可以移植到不同平台的机器。

(6) 网络包支持 Internet 的 TCP/IP 协议, 提供了与 Internet 的接口。它支持 URL 连接, WWW 的即时访问, 并且简化了客户-服务器模型的程序设计。

Java 的上述种种特性不但能适应网络应用开发的需求, 而且还体现了当今软件开发方法的若干新成果和新趋势。在以后的章节里, 将结合对 Java 语言的讲解, 分别介绍这些软件开发方法。

1.1.4 Java 和 C、C++比较

对于变量声明、参数传递、操作符、流控制等, Java 使用了和 C、C++相同的传统, 使得熟悉 C、C++的程序员能很方便地进行编程。同时, Java 为了实现其简单、健壮、安全等特性, 也摒弃了 C 和 C++中许多不合理的内容。下面选择性地讲述几点, 对于学过 C 语言或 C++语言的读者而言, 起一个快速参考的作用。对于未学过 C 语言的读者, 可以略过此节。

1. 全局变量

Java 程序中, 不能在所有类之外定义全局变量, 只能通过在一个类中定义公用、静态的变量来实现一个全局变量。Java 对全局变量进行了更好的封装。而在 C 和 C++中, 全局变量使用不当常常会造成系统的崩溃。

2. Goto 语句

Java 不支持 C、C++中的 Goto 语句, 而是通过异常处理语句 try、catch、finally 等来代替 C、C++中用 Goto 来处理遇到错误时跳转的情况, 使程序更可读且更结构化。

3. 指针

指针是 C、C++中最灵活, 也是最容易产生错误的数据类型。由指针所进行的内存地址操作常会造成不可预知的错误, 同时通过指针对某个内存地址进行显示类型转换后, 可以访问一个 C++中的私有成员, 从而破坏了安全性, 造成系统的崩溃。而 Java 对指针进行完全的控制, 程序员不能直接进行任何指针操作, 例如, 把整数转化为指针, 或者通过指针释放某一内存地址等。同时, 数组作为类在 Java 中实现, 很好地解决了数组访问越界这一在 C、C++中不做检查的错误。

4. 内存管理

在 C 中, 程序员通过库函数 malloc()和 free()来分配和释放内存, C++中则通过运算符 new 和 delete 来分配和释放内存。再次释放已释放的内存块或未被分配的内存块, 会造成系统的崩溃; 同样, 忘记释放不再使用的内存块也会逐渐耗尽系统资源。而在 Java 中, 所有的数据结构都是对象, 通过运算符 new 为它们分配内存。通过 new 得到对象的处理权, 而实际分配给对象的内存可能随程序运行而改变, Java 对此自动地进行管理并且进行垃圾收集, 有效地防止了由于程序员的误操作而导致的错误, 并且更好地利用了系统资源。

5. 数据类型的支持

在 C、C++中, 对于不同的平台, 编译器为简单数据类型, 如 int、float 等分别分配不同长度的字节数, 例如, int 在 IBM PC 中为 16 位, 在 VAX-11 中为 32 位, 这导致了代码的不可移植性, 但在 Java 中, 对于这些数据类型总是分配固定长度的位数, 如对 int 型, 它总占 32 位, 这就保证了 Java 的平台无关性。

6. 类型转换

在 C、C++中, 由于可以通过指针进行任意的类型转换, 因此常常带来不安全性; 而 Java