国外经典初学者入门教程系列

Mc Graw Hill Education

Java: A Beginner's Guide

Java 实用教程

（第3版）

使用最新的 J2SE 5版本

Herbert Schildt 著
马海军 景丽 等译

Mc Graw Hill

清华大学出版社

Java 实用教程

（第 3 版）

Herbert Schildt 著

马海军 景丽 等 译

清华大学出版社

北 京

Herbert Schildt
Java：A Beginner's Guide，Third Edition
EISBN: 0-07-223189-0

图书在版编目（CIP）数据

Java 实用教程：第 3 版/（美）赫伯特（Herbert, S.）著；马海军，景丽等译. —北京：清华大学出版社，2005.12
书名原文：Java: A Beginner's Guide, Third Edition
ISBN 7-302-12066-8

I. J…　II. ①赫…　②马…　③景…　III. JAVA 语言-程序设计-教材　IV. TP312

中国版本图书馆 CIP 数据核字（2005）第 127512 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
责任编辑：冯志强
印 刷 者：北京市清华园胶印厂
装 订 者：三河市新茂装订有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×230　**印张**：36　**字数**：828 千字
版　　次：2005 年 12 月第 1 版　2005 年 12 月第1 次印刷
书　　号：ISBN 7-302-12066-8/TP・7811
印　　数：1～3000
定　　价：64.00 元

前　言

在过去短短的几年中，Java 从鲜有问津迅速成为一种最为重要的 Internet 语言。今天要想成为一名专业的 Web 开发者，必须熟练掌握 Java。因此，如果你将来要从事基于 Internet 的编程工作，应该选择正确的语言学习——本书正是为了帮助你学习 Java 而编写的。

本书介绍了 Java 编程的基础知识。本书采用分步骤的教学方法，安排了许多示例、自我测试和编程练习。本书不需要读者具备编程经验，从最基本的基础知识，诸如如何编译并运行一个 Java 程序开始讲起。接下来讨论了每一个 Java 关键词，还介绍了 Java 的一些最重要的高级功能，如多线程编程、泛化和创建 applets。最终读者将会牢固地掌握 Java 编程精髓。

值得一提的是，本书只是学习 Java 的起点。Java 远不止是一些定义语言的元素，它还包括了扩展的库和工具来帮助开发程序。而且，Java 还提供了一组复杂的库来处理浏览器用户界面。只有成为顶尖的 Java 程序员才能掌握这些领域的奥秘。希望读者在学习完本书之后，继续学习 Java 的其他知识。

0.1　Java 的发展历程

很少有几种语言能够对计算机程序设计的发展带来全方位的深刻影响。可以毫不夸张地说，1995 年 Sun 公司发布的 Java 1.0 给计算机程序设计领域带来了一场变革。这场变革迅速地把 Web 带入了一个高度交互的环境，也给计算机语言设计设置了一个新标准。

多年以来，Java 不断地发展、演化和修订。和其他语言加入新功能的动作迟缓不同，Java 一直站在计算机程序设计语言的前沿，部分原因是其变革的文化，部分原因是它所面对的变化。Java 已经做过或大或小的多次升级。

第一次主要的更新是 Java 1.1 版，这次更新比较大，加入了很多新的库元素，修订了处理事件的方式，重新配置了 1.0 版本的库中的许多功能。

第二个主要的版本是 Java 2，它表示 Java 的第二代，标志着 Java 的现代时代的到来。Java 2 第一个发布的版本号是 1.2，该号码最初指 Java 库的内部版本号，后来就泛指整个版本号了。Java 2 被 Sun 包装为 J2SE（Java 2 Platform Standard Edition），并且开始把版本号应用于该产品。

Java 的下一次升级是 J2SE 1.3，它增强了一些已有的功能，并且紧凑了开发环境。J2SE 1.4 进一步增强了 Java。该版本包括一些重要的新功能，如链式异常、基于 I/O 的通道，以及 assert 关键字。

Java 的最新版是 J2SE 5，它无论从深度、广度等各方面看，都是一次具有重要意义的升级。

0.2 J2SE 5 是 Java 的第二次变革

J2SE 5 添加了许多新功能，从根本上改变了 Java 语言的特性，增强了其性能和适用范围，并且会完全改变 Java 代码的编写方式。

下面是 J2SE 的一些主要新功能：

- 泛化
- 自动封包/自动解包
- 增强型 for-each 形式的 for 循环
- 可变长度变元
- 静态导入
- 元数据

仅从这些项目还看不出这次升级的意义。其中，泛化、增强型 for 循环和可变长度变元引入了新的语法元素；自动封包和自动解包修改了语法规则；元数据增加了一种全新的编程注释方法。

值得说明的是，从版本号的变化方式看，这一版本的 Java 应该是 1.5。由于新功能和变革如此之多，常规的版本号无法标识实际的变化，所以 Sun 决定使用版本号 5，因此，当前的版本叫做 J2SE 5，开发工具包叫做 JDK 5。同时，为了维持和以前的一致性，Sun 在内部使用内部版本号 1.5，即外部版本号为 5，内部版本号为 1.5。用户在使用编译器显示版本号，以及搜索在线文档时，使用的是内部版本号 1.5。

0.3 本书的组织结构

本书采用教程式的组织结构，每一章都建立在前面的基础之上。本书共分 14 章（模块），每一章讨论一个有关 Java 的议题。本书的特色就在于它包含了许多便于读者学习的特色内容。

重要技能

每一章都包括一些重要技能，并且在各章中以节标识。

思考与练习

每一章都有思考与练习，测试读者学习到的知识。答案在附录中提供。

学习检查

每一节后面都有一个“学习检查”，问题的答案在底注中。

专家问答

每一章中都有一些“专家问答”，以一问一答的形式介绍补充知识和要点。

练习

每一章中都包含 1~2 个实习项目，帮助读者将学习的知识应用到实践中去。

0.4 本书不需要读者具备编程经验

本书假定读者没有任何编程经验。如果你没有任何编程经验，正好阅读本书。当然，许多读者都已经或多或少有了一些编程经验，对于大多数读者而言，这些编程经验就是 C++。实际上 C++和 Java 具有一些相似性。因此，如果你已经学习过 C++，学习 Java 会非常容易。由于许多读者都已经学习过 C++，因此，书中会经常指出 C++和 Java 之间的相似之处。

0.5 本书需要的软件环境

要想编译和运行本书的程序，需要获得最新版本的 Sun 的 Java Software Developers Kit (SDK)，本书使用的是 Java 2，第 5 版（J2SE 5）。本书在第 1 章介绍了如何获得 Java SDK。

0.6 不要忘记 Web 上的代码

别忘了，本书所有示例和编程项目的源代码都可以免费从 Web 网址 **www.osborne.com** 获得。

目　录

第 1 章　Java 基础 …… 1
1.1　Java 的起源 …… 2
1.1.1　Java 与 C 和 C++ 的关系 …… 3
1.1.2　Java 与 C#的关系 …… 4
1.2　Java 对 Internet 的贡献 …… 4
1.2.1　Java applets …… 5
1.2.2　安全 …… 5
1.2.3　可移植性 …… 5
1.3　Java 的魔力在于字节码 …… 5
1.4　Java 的关键术语 …… 6
1.5　面向对象程序设计 …… 7
1.5.1　封装 …… 8
1.5.2　多态性 …… 9
1.5.3　继承 …… 9
1.6　第一个简单的程序 …… 11
1.6.1　输入程序 …… 11
1.6.2　编译程序 …… 12
1.6.3　逐行分析第一个程序 …… 12
1.7　第二个简单程序 …… 16
练习 1-1　将加仑换算为升 …… 20
1.8　两个控制语句 …… 21
1.8.1　if 语句 …… 21
1.8.2　for 循环语句 …… 23
1.9　创建代码块 …… 25
1.10　分号和定位 …… 26
练习 1-2　改进从加仑到升的转换程序 …… 27
1.11　Java 的保留关键词 …… 30
1.12　Java 的标识符 …… 30
1.13　思考与练习 …… 31

第 2 章　数据类型与运算符 …… 33
2.1　Java 的原语类型 …… 34
2.1.1　整数类型 …… 34
2.1.2　浮点型 …… 36
2.1.3　字符型 …… 37
2.1.4　boolean 类型 …… 38
练习 2-1　闪电有多远 …… 39
2.2　字面值 …… 40
2.2.1　十六进制与八进制常量 …… 41
2.2.2　字符转义序列 …… 41
2.2.3　字符串字面值 …… 42
2.3　变量详解 …… 43
2.3.1　初始化变量 …… 43
2.3.2　动态初始化 …… 44
2.4　作用域和变量的生命期 …… 44
2.5　算术运算符 …… 48
2.6　关系运算符和逻辑运算符 …… 50
2.7　赋值运算符 …… 54
2.8　速记赋值 …… 54
2.9　赋值中的类型转换 …… 55
2.10　不兼容类型的强制转换 …… 56
练习 2-2　显示逻辑运算符的真值表 …… 59

2.11 表达式 …… 60
2.11.1 表达式中的类型转换 …… 61
2.11.2 间距和圆括号 …… 63
2.12 思考与练习 …… 63

第 3 章 程序控制语句 …… 65
3.1 从键盘输入字符 …… 66
3.2 if 语句 …… 67
3.2.1 嵌套 if 语句 …… 69
3.2.2 if-else-if 阶梯状结构 … 70
3.3 switch 语句 …… 71
练习 3-1 建立一个 Java 帮助系统 …… 76
3.4 for 循环 …… 79
3.4.1 for 循环的一些变体 … 80
3.4.2 缺失部分要素的 for 循环 …… 81
3.4.3 无限循环 …… 83
3.4.4 没有循环体的循环 …… 83
3.4.5 在 for 循环内部声明循环控制变量 …… 84
3.4.6 增强型 for 循环 …… 85
3.5 while 循环 …… 85
3.6 do-while 循环 …… 87
练习 3-2 改进 Java 帮助系统 …… 89
3.7 使用 break 退出循环 …… 92
3.8 将 break 语句作为一种 goto 语句使用 …… 94
3.9 使用 continue …… 99
练习 3-3 完成 Java 帮助系统 …… 101
3.10 嵌套的循环 …… 105
3.11 思考与练习 …… 106

第 4 章 类、对象和方法 …… 108
4.1 类的基础知识 …… 109
4.1.1 类的基本形式 …… 109
4.1.2 定义类 …… 110
4.2 如何创建对象 …… 114
4.3 引用变量和赋值 …… 114
4.4 方法 …… 115
4.5 从方法返回值 …… 118
4.6 返回值 …… 119
4.7 使用参数 …… 121
练习 4-1 创建 Help 类 …… 125
4.8 构造函数 …… 132
4.9 带参数的构造函数 …… 133
4.10 深入介绍 new 运算符 …… 136
4.11 垃圾回收与终止器 …… 136
练习 4-2 演示垃圾回收 … 138
4.12 this 关键词 …… 140
4.13 思考与练习 …… 142

第 5 章 其他数据类型与运算符 …… 144
5.1 数组 …… 145
练习 5-1 排序数组 …… 149
5.2 多维数组 …… 151
5.3 不规则数组 …… 152
5.3.1 三维或更多维的数组 …… 154
5.3.2 初始化多维数组 …… 154
5.4 另一种声明数组的语法 …… 156
5.5 数组引用赋值 …… 156
5.6 使用 length 成员 …… 158
练习 5-2 Queue 类 …… 160
5.7 for-each 形式的循环 …… 165
5.8 字符串 …… 170
5.8.1 构造字符串 …… 171

5.8.2 操作字符串 …… 172
5.8.3 字符串数组 …… 174
5.8.4 字符串是不可变的 … 175
5.9 使用命令行变元 …… 176
5.10 位运算符 …… 178
5.10.1 位运算符的与、或、异或和非 …… 178
5.10.2 位移运算符 …… 183
5.10.3 位运算符赋值速记符 …… 185
练习 5-3 ShowBits 类 …… 186
5.11 “?”运算符 …… 189
5.12 思考与练习 …… 191

第 6 章 方法和类剖析 …… 193
6.1 控制对类成员的访问 …… 194
6.2 Java 的访问指示符 …… 194
练习 6-1 改进 Queue 类 …… 199
6.3 向方法传递对象 …… 201
6.4 返回对象 …… 205
6.5 方法重载 …… 207
6.6 重载构造函数 …… 213
练习 6-2 重载 Queue 构造函数 …… 216
6.7 递归 …… 219
6.8 理解 static 关键词 …… 222
练习 6-3 快速排序 …… 226
6.9 嵌套类和内部类 …… 229
6.10 可变长度变元（varargs） … 233
6.10.1 varargs 基础 …… 234
6.10.2 重载 varargs 方法 … 237
6.10.3 varargs 和歧义 …… 239
6.11 思考与练习 …… 240

第 7 章 继承 …… 242
7.1 继承的基础知识 …… 243
7.2 构造函数和继承 …… 249
7.3 使用 super 调用超类构造函数 …… 251
7.4 使用 super 访问超类成员 …… 256
练习 7-1 扩展 Vehicle 类 …… 257
7.5 创建多级层次结构 …… 261
7.6 何时调用构造函数 …… 264
7.7 超类引用和子类对象 …… 266
7.8 方法重写 …… 271
7.9 重写的方法支持多态性 …… 274
7.9.1 为何使用重写的方法 …… 276
7.9.2 在 TwoDShape 中应用方法重写 …… 276
7.10 使用抽象类 …… 281
7.11 使用 final …… 286
7.11.1 使用 final 防止重写 …… 286
7.11.2 使用 final 防止继承 …… 287
7.11.3 对数据成员使用 final …… 287
7.12 Object 类 …… 289
7.13 思考与练习 …… 290

第 8 章 包和接口 …… 291
8.1 包 …… 292
8.1.1 定义包 …… 292
8.1.2 寻找包和 CLASSPATH …… 293
8.1.3 一个简短的包的示例 …… 294
8.2 包和成员访问 …… 295
8.3 理解被保护的成员 …… 298

8.4 导入包 …… 301
8.5 Java 的类库位于包中 …… 302
8.6 接口 …… 303
8.7 实现接口 …… 304
8.8 使用接口引用 …… 308
练习 8-1 创建队列接口 ……311
8.9 接口中的变量 …… 317
8.10 接口能够扩展 …… 318
8.11 思考与练习 …… 320

第 9 章 异常处理 …… 321
9.1 异常的层次结构 …… 322
9.2 异常处理基础 …… 322
9.2.1 使用 try 和 catch …… 323
9.2.2 一个简单的异常示例 …… 324
9.3 未捕获异常的结果 …… 326
9.4 使用多个 catch 语句 …… 329
9.5 捕获子类异常 …… 330
9.6 嵌套 try 代码块 …… 332
9.7 抛出异常 …… 333
9.8 Throwable 详解 …… 336
9.9 使用 finally …… 338
9.10 使用 throws …… 340
9.11 Java 的内置异常 …… 342
9.12 创建异常子类 …… 344
练习 9-1 向排序类添加异常 …… 346
9.13 思考与练习 …… 350

第 10 章 使用 I/O …… 352
10.1 Java 的 I/O 基于流 …… 353
10.2 字节流和字符流 …… 353
10.3 字节流类 ……354
10.4 字符流类 ……354
10.5 预定义流 ……355
10.6 使用字节流 ……356
10.6.1 读取控制台输入 ….357
10.6.2 写入控制台输出 ….358
10.7 使用字节流读写文件 ……359
10.7.1 从文件输入 ……359
10.7.2 写入文件 ……361
10.8 读写二进制数据 ……363
练习 10-1 文件比较程序 ……367
10.9 随机访问文件 ……369
10.10 使用 Java 字符流 ……372
10.10.1 基于字符流的控制台输入 ……373
10.10.2 使用字符流的控制台输出 ……376
10.11 使用字符流的文件 I/O（OK） ……377
10.11.1 使用 FileWriter …377
10.11.2 使用 FileReader …379
10.12 使用 Java 类型包装器转换数值字符串 ……380
练习 10-2 创建一个基于的帮助系统 ….383
10.13 思考与练习 ……390

第 11 章 多线程程序设计 ……391
11.1 多线程基本原理 ……392
11.2 Thread 类和 Runnable 接口 ……392
11.3 创建一个线程 ……393

练习 11-1 扩展 Thread …… 399
11.4 创建多重线程 …… 402
11.5 确定线程何时结束 …… 405
11.6 线程的优先级 …… 409
11.7 同步化 …… 412
11.8 使用同步化方法 …… 412
11.9 同步化语句 …… 416
11.10 使用 notify( )、wait( )和 notifyAll( )的线程通信 …… 418
11.11 线程的挂起、继续执行和停止 …… 424
练习 11-2 使用主线程 … 428
11.12 思考与练习 …… 430

第 12 章 枚举、自动封包和静态导入 …… 432
12.1 枚举 …… 433
12.2 Java 的枚举是类类型 …… 436
12.3 values( )和 valueOf( )方法 …… 436
12.4 构造函数、方法、实例变量和枚举 …… 438
12.5 枚举继承 Enum …… 440
练习 12-1 计算机控制的交通指示灯 …… 442
12.6 类型包装器 …… 448
12.7 自动封包基础 …… 450
12.8 自动封包和方法 …… 451
12.9 发生在表达式中的自动封包/自动解包 …… 452
12.10 静态导入 …… 454
12.11 元数据 …… 457
12.12 思考与练习 …… 460

第 13 章 泛化 …… 461
13.1 泛化基础 …… 462
13.2 一个简单的泛化示例 …… 463
13.3 约束类型 …… 469
13.4 使用通配符变元 …… 473
13.5 约束通配符 …… 476
13.6 泛化方法 …… 480
13.7 泛化构造函数 …… 482
13.8 泛化接口 …… 483
练习 13.1 创建一个泛化队列 …… 486
13.9 原类型和遗留代码 …… 490
13.10 Erasure …… 493
13.11 歧义错误 …… 495
13.12 一些泛化限制 …… 496
13.13 思考与练习 …… 498

第 14 章 applet、事件和其他议题 …… 500
14.1 applet 基础 …… 501
14.2 applet 层次结构 …… 504
14.3 一个完整的 applet 框架 …… 505
14.4 applet 初始化与终止 …… 506
14.5 请求重绘 …… 507
练习 14-1 一个简单的广告条 applet …… 509
14.6 使用状态窗口 …… 513
14.7 向 applet 传递参数 …… 514
14.8 Applet 类 …… 516
14.9 委派事件模型 …… 517
14.9.1 事件 …… 518
14.9.2 事件源 …… 518
14.9.3 事件侦听者 …… 518
14.9.4 事件类（Event Class） …… 519
14.9.5 事件侦听者接口 … 519
14.10 使用委派事件模型 …… 520

14.10.1 处理鼠标事件…… 520
14.10.2 一个简单的鼠标事件 applet …… 521
14.11 其他 Java 关键词 …… 524
14.11.1 transient 和 volatile 修饰符 …… 525
14.11.2 instanceof …… 525
14.11.3 strictfp ……525
14.11.4 assert ……526
14.11.5 native 方法……526
14.12 思考与练习……527
附录 思考与练习答案……529

第 1 章

Java基础

1.1 了解 Java 的历史和基本原理
1.2 理解 Java 对 Internet 的贡献
1.3 理解字节码的重要性
1.4 了解 Java 的术语
1.5 理解面向对象程序设计的基本原理
1.6 创建、编译和运行一个简单的 Java 程序
1.7 使用变量
1.8 使用 if 和 for 控制语句
1.9 创建代码块
1.10 理解如何定位、缩进和终止语句
1.11 了解 Java 关键词
1.12 理解 Java 标识符的规则

Internet 和 World Wide Web 的兴起从根本上改变了计算机的处理方式。短短数年前，电脑界还是由孤立的 PC 机所统治，而今天，几乎所有的 PC 机都接入了 Internet。最初，Internet 本身只是用于提供一种共享文件和信息的捷径，但是今天，它已经演变成了一个浩瀚的分布式计算空间。这些改变引发了一种新的编程方法的产生，这就是 Java。

Java 是一种卓越的 Internet 语言，而且不仅如此，它还使程序设计产生了革命，改变了我们考虑程序形式与功能的方式。今天，要成为职业的程序员就意味着要具备使用 Java 编程的能力，这一点非常重要。在本书的课程中，你将学习到掌握 Java 的必要技巧。

本章的目的是向你介绍 Java，包括它的历史、设计原理和一些最重要的特性。目前，学习程序设计语言最大的难点是各部分之间不是相互孤立的。相反，语言各个组成部分的运作是相互关联的。这种相互的关联性在 Java 中尤为突出。事实上，只讨论 Java 的一个方面，而不涉及其他部分是非常困难的。为了有助于克服这一困难，本章对 Java 的几个特性进行了简单的概述，其中包括 Java 程序的基本形式、一些基本的控制结构和操作符。对于这些内容我们并不进行深入讨论，只是关注一下 Java 程序共有的一些概念。

1.1 Java的起源

促使计算机语言革新的因素有两个：程序设计技术的改进和计算环境的改变。Java 也不例外。在大量继承 C 和 C++的基础之上，Java 还增加了体现当前程序设计技术状态的功能与精华。针对在线环境的蓬勃发展，Java 为高度的分布式体系结构提供了流水线程序设计功能。

Java 是 1991 年由 Sun Microsystems 的 James Gosling、Patrick Naughton、Chris Warth、Ed Frank 和 Mike Sheridan 共同构想的成果。这个语言起初名为“Oak”，于 1995 年更名为“Java”。多少有些让人吃惊的是，设计 Java 的最初动力并不是源于 Internet，而是出于创建一种独立于平台的语言，使其能够用于创建内嵌于不同消费类电子产品，如烤箱、微波炉和遥控器的软件的需要。正如你可能猜想到的，不同类型的 CPU 都可以作为控制器使用。麻烦在于多数的计算机语言只能用于一个特定的目标。例如，C++即是如此。

虽然任何类型的 CPU 或许都能编译 C++程序，然而这需要 CPU 有一个完整的 C++编译器。而编译器又非常昂贵，且其开发也很耗时。为了找到更好的解决方法，Gosling 和其他人尝试开发一种可移植的交叉平台语言，使该语言生成的代码可以在不同环境下的不同 CPU 上运行。这一努力最终导致了 Java 的诞生。

大概就是在快要设计出 Java 的细节的时候，另一个在 Java 未来中扮演关键角色的更重要的因素出现了。这个第二动力就是 World Wide Web。在 Web 还未成形的同时，Java 出炉了。对于消费类电子产品的程序设计而言，Java 可能是一个有用但却晦涩难懂的语

言，然而随着 Web 的出现，Java 被推到了计算机语言设计的前端，因为 Web 也需要可移值程序。

大多数程序员在其早期生涯都了解可移植程序是既令人期待，也让人难以捉摸的。在对创建高效可移植（独立于平台）程序的需要几乎与程序设计学科一样陈旧的同时，由于诸多问题，可移植程序退到了后台。然而，Internet 和 Web 的出现使原有的可移植性问题重新摆上了桌面。因为，Internet 毕竟是一个由许多类型的计算机、操作系统和 CPU 组成的多样化的分布式空间。曾经恼人心绪，却也无关紧要的问题也就成为了要急需解决的问题。

到 1993 年，Java 设计团队的成员十分明显地发现，在创建嵌入式代码时经常遇到的问题同样也出现在创建 Internet 的代码中。了解到这一点以后，Java 的重点从消费类电子产品转移到了 Internet 程序设计。因此，尽管开发中性体系结构的程序设计语言的初衷提供了最初的星星之火，然而却是 Internet 最终促成了 Java 的燎原之势。

1.1.1 Java 与 C 和 C++的关系

Java 与 C 和 C++直接相关。Java 继承了 C 的语法，Java 的对象模型从 C++改编而来。Java 与 C 和 C++关系之所以重要是出于以下几个原因。

第一，许多程序员都熟悉 C/C++语法。这样对于他们而言，学习 Java 就简单了。同样，对于 Java 程序员学习 C/C++也是很简单的。

第二，Java 设计者并没有重开炉灶。相反，他们进一步对成功的程序设计范式进行了提炼。现代程序设计始于 C，而后过渡到 C++，现在则是 Java。通过大量的继承，Java 提供了一个强大的、可以更好利用已有成果的逻辑一致的程序设计环境，并且增加了在线环境需求的新功能。然而，或许最重要的一点在于，由于它们的相似性，C、C++和 Java 为专业程序员定义了一个统一的概念架构。程序员从其中一种语言转到另一种语言时，不会遇到太大的困难。

C 和 C++的主要设计原理之一就是程序员的控制。Java 也继承了这一原理。除了 Internet 环境施加的约束以外，Java 还为程序员提供了完全的控制。如果你的程序编得好，程序就会体现出来，而如果不好，同样也会体现出来。不同的是，Java 并不是一种教学式语言，它是为专业程序员准备的语言。

Java 还有一个与 C 和 C++共有的属性：即它是由真正的程序员设计、测试和修改的。它与设计者的需求和经验紧密结合。因此，再没有比这更好的方法来创建如此一流的专业程序设计语言了。

因为 Java 与 C++的相似性，特别是它们对面向对象程序设计的支持，使得 Java 被简单地认为是“C++的 Internet 版”。然而，这种观点是错误的。因为 Java 在实际应用以及基本原理上与 C++有显著的不同。尽管 Java 受到 C++的影响，但是它绝不是 C++的增强版。例如，Java 不提供对 C++的向上或向下兼容。当然，Java 与 C++的相似是十分明

显的，而且如果你是一名 C++程序员，那么在你使用 Java 时会有驾轻就熟的感觉。另外，Java 不是为替代 C++而设计的，而是为了用来解决一系列特定问题而出现的。C++则是用来解决另外一系列不同问题的。两者将在未来几年中共存。

1.1.2 Java 与 C#的关系

近来，一种名为 C#的新语言浮出了水面。由 Microsoft 创建的用以支持其.NET 架构的 C#，与 Java 也密切相关。事实上，C#的许多功能都是直接从 Java 改编来的。Java 和 C#共享相同的 C++语法风格，都支持分布式程序设计，使用相同的对象模型。它们之间当然有不同之处，但就整体外观而言，两者极为相似。这就意味着，如果你已经了解了 C#，那么学习 Java 就很简单，反之，如果你将来要学的是 C#，那么现在学到的有关 Java 的知识也会给你将来带来帮助。

鉴于 Java 与 C#两者的相似性，自然有人要问，“C#会替代 Java 吗？”。Java 和 C#是对两种不同类型计算环境的优化，正如 C++会和 Java 长期共存一样，C#和 Java 也会长期共存。

1. Java 对 Internet 而言用途很大是因为它可以生成_____程序。①
2. Java 是由什么语言直接派生出来的？②

1.2 Java对Internet的贡献

Internet 帮助 Java 走到了程序设计的前台，而 Java 也对 Internet 产生了深远的影响。对于这一点，原因很简单，Java 扩展了对象的生存空间，使其可以在电脑空间中自由穿梭。在网络中，往来于服务器与个人计算机之间有两大类对象：即被动数据和动态主动程序。例如，你阅读电子邮件时，你浏览的就是被动数据，甚至在你下载程序时，程序代码运行之前也是被动数据。然而，传输到计算机的第二种对象，则是动态、自执行的程序。尽管这样的程序是由服务器初始化，但它是客户计算机上的一个主动代理。例如，一个可能由服务器提供的用于正确显示传输数据的程序就是主动程序。

动态网络程序给人们带来渴望的同时，也带来了安全和可移植方面的严重问题。因为在 Java 之前，电脑空间对现在一半的对象而言是封闭的。正如你所看到的，Java 带来了这些问题，并且定义了一种新的程序形式：applet。

① 可移植。
② C 和 C++。

1.2.1 Java applets

applets 是一种特殊的 Java 程序，用于在 Internet 上传输，由兼容 Java 的 Web 浏览器执行。而且，applets 可以像图像、声音和视频文件一样根据用户的需要下载。与普通多媒体文件的不同之处在于，applets 是一种智能程序。也就是说，applets 能够反映用户的输入并且动态进行变化，而不是像动画或者声音文件那样反复地播放。

尽管 applets 令人激动，但是前提是 Java 必须解决两个与 applets 相关的基本问题：即安全性与可移植性问题。在继续学习之前，我们将定义一下与 Internet 相关的安全与可移植性的概念。

1.2.2 安全

正如你可能意识到的，每次下载一个“普通”程序时都可能会感染病毒。在 Java 出现之前，大多数用户并不经常下载可执行程序，如果下载了这些程序，他们也会在执行程序之前查杀病毒。即便如此，许多用户仍然担心自己的系统可能感染了病毒，或者有恶意的程序在自己的系统中横行（恶意程序可能会通过查找你的计算机的本地文件系统来收集你的私人信息，如信用卡号、银行账户收支情况和密码等）。Java 通过在网络应用程序和计算机之间提供“防火墙”解除了这些顾虑。

使用兼容 Java 的 Web 浏览器，可以安全地下载 Java applets，而无需害怕感染病毒。Java 实现这一功能的方法是将 Java 程序限制于 Java 执行环境中，不允许它访问计算机的其他部分（稍后，你会看到这是如何实现的）。坦率地讲，下载 applets，而且确保对客户计算机无害的功能是 Java 最重要的特征。

1.2.3 可移植性

正如前面所讨论的，与 Internet 连接的计算机和操作系统有多种类型。对于被动态下载到不同类型平台的程序而言，必须具备生成可移植的可执行代码的方法。正如你稍后会看到的，确保安全的机制同时也有助于确保创建可移植代码。Java 对安全性和可移植性问题的解决方法可谓既简洁又有效。

1.3 Java的魔力在于字节码

Java 能够解决前面提到的安全问题和可移植问题的关键在于 Java 编译器的编译结果不是可执行代码，而是字节码（bytecode）。字节码是设计用来由名为 Java 虚拟机（Java Virtual Machine，JVM）的 Java 运行时系统执行的高度优化的一系列指令。用专业语言讲，Java 虚拟机是一个字节码解释器。这可能会让你有些吃惊。因为如你所知，多数现代语言，如 C++是设计用来被编译，而不是被解释的——这多半也是由于性能方面的原