

## <<Java程序设计教程>>

### 图书基本信息

书名：<<Java程序设计教程>>

13位ISBN编号：9787113117917

10位ISBN编号：7113117910

出版时间：2010-9

出版时间：左晓英、徐翠娟、李德有 中国铁道出版社 (2010-09出版)

作者：左晓英，徐翠娟，李德有著

页数：180

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;Java程序设计教程&gt;&gt;

## 前言

Java是由Sun Microsystems公司于1995年5月推出的Java程序设计语言和Java平台的总称。用Java实现的Hot Java浏览器（支持Java Applet）显示了Java的魅力：跨平台、动态的web、Internet计算。

从此，Java被广泛接受并推动了Web的迅速发展，常用的浏览器现在均支持Java Applet。

2003年，有5.5亿台桌面计算机上运行Java程序，75%的开发人员将Java作为主要的开发工具。

至今还保持着编程语言第一的市场份额，可以毫不夸张地说，Java在软件界占据半壁江山。

每一种语言都会有出生、成长、壮大、衰落几个阶段，Java也不例外，既然有它的黄金时期，就有它的衰落时期。

不过我们大可放心，因为Java不只是一门语言，更是一门技术，Java所代表的这一类语言将不会消失。

人们需要跨平台、开源的技术，人们需要功能强大但使用简单的技术。

我们认为未来高级编程语言将会更清晰的分为三大类：一类适合底层开发，追求效率，就像C语言；一类适合傻瓜式的开发，目标是使人人都能够编程，就像Delphi；第三类则是在效率与简单之间寻找平衡，就像Java。

本书共分6章。

第1章介绍Java基础的起源与发展、Java的特征以及Java技术的体系结构；第2章以及第3章介绍Java语言基础语法，包括标示符、关键字、数据类型、运算符、表达式、语句等；第4章介绍Java的数据类型中常用的字符串及数组；第5章开始介绍Java面向对象的特征，包括封装、继承、多态等；第6章详细介绍了面向对象中继承的概念及应用等。

该书是作者多年的教学经验及项目实战经验积累的结晶，是对Java应用技术及学生项目实战演练中出现的问题进行分析、总结的成果。

分析结果表明：学生在学习Java技术中出现问题根本原因是由于找不到一本便于理解的、由浅入深出的教材，本套书有两册：《Java程序设计教程（入门篇）》和《Java程序设计教程（进阶篇）》。

前者主要介绍Java核心基础技术，使读者了解Java语言的一些关键特性；后者主要介绍编程人员在开发时需要用到的高级应用技术。

因此，两者搭配使用能够使读者快速掌握使用Java语言进行软件开发的技术。

在这里我们将竭诚为Java爱好者提供帮助。

如果您是一个有着丰富经验的开发人员，并且能够灵活应用像枚举和泛型这样的高级语言特性，那么您就不必学习完“入门篇”再学习“进阶篇”，不过在“进阶篇”中根据内容需要适当参考“入门篇”中的有关内容。

两本书配合使用能满足读者学习Java语言的需求，使得读者不必再徘徊在Java技术的大门外，使其能够走进Java语言的美妙世界。

## <<Java程序设计教程>>

### 内容概要

《Java程序设计教程（入门篇）》在介绍Java基础语法的同时，由浅入深地引入面向对象编程的概念，能够让读者快速入门，理解Java程序设计语言。

《Java程序设计教程（入门篇）》共分6章。

第1章介绍了Java基础的起源与发展、Java的特征以及Java技术的体系结构。

第2、3章介绍了Java语言基础语法，包括标识符、关键字、数据类型、运算符、表达式、语句等。

第4章介绍了字符串及数组。

第5章介绍Java面向对象的特征，包括封装、继承、多态等。

第6章详细介绍了面向对象中继承的概念及应用等。

《Java程序设计教程（入门篇）》中提供了大量的经典实例，从而达到快速理解知识点的目的。

《Java程序设计教程（入门篇）》层次清晰，结构严谨，便于理解，注重应用，既可以作为高等院校相关专业的教材，也可作为从事软件开发工作的专业技术人员的参考书。

## &lt;&lt;Java程序设计教程&gt;&gt;

## 书籍目录

第1章 Java课程概述1.1 Java的由来1.2 Java的历史1.3 Java的特性1.3.1 Java语言的优点1.3.2 Java语言的关键特性1.4 Java程序运行机制1.5 Java虚拟机1.6 JDK的安装与配置1.7 编写第一个Java程序1.8 程序的检测1.8.1 语法错误1.8.2 语义错误1.9 提高程序的可读性本章小结第2章 标识符、关键字、数据类型2.1 标识符与关键字2.1.1 标识符2.1.2 关键字2.2 变量2.3 基本数据类型概述2.3.1 基本数据类型2.3.2 整型类型2.3.3 字符类型2.3.4 单精度浮点型与双精度浮点型2.3.5 布尔类型2.3.6 基本数据类型的默认值2.3.7 数据类型的转换本章小结第3章 运算符、表达式、语句3.1 运算符、表达式与语句3.1.1 运算符与表达式3.1.2 赋值运算符3.1.3 运算符的优先级3.2 表达式3.3 循环与选择性语句3.3.1 程序的结构设计3.3.2 量择结构3.3.3 多重选择语——switch语句3.3.4 while循环3.3.5 do...while循环3.3.6 for循环3.3.7 循环嵌套3.3.8 循环的跳出3.3.9 局部变量本章小结第4章 字符串和数组4.1 String、String Buffer和String Builder4.1.1 String类4.1.2 string Buffer类4.1.3 String Builder类4.2 数组4.2.1 一维数组4.2.2 二维数组4.3 多维数组本章小结第5章 类的基本形式5.1 面向对象程序设计的基本概念5.1.1 对象5.1.2 类5.1.3 封装性5.1.4 继承性5.1.5 多态性5.2 类与对象5.2.1 类的声明5.2.2 创建新的对象5.2.3 创建多个新对象5.3 类的封装性5.4 在类内部调用方法5.5 引用数据类型的传递5.6 匿名对象5.7 构造方法5.8 对象的比较5.9 this关键字的使用5.10 static关键字的使用5.10.1 静态变量5.10.2 静态方法5.10.3 理解main ( ) 方法5.10.4 静态代码块5.11 私有构造方法5.12 对象数组的使用5.13 内部类5.13.1 在类外部引用内部类5.13.2 在方法中定义内部类5.14 Java文档注释5.14.1 java.doc标记5.14.2 文档注释的一般形式5.14.3 java.doc的输出本章小结第6章 类的继承6.1 继承的基本概念6.1.1 子类对象的实例化过程6.1.2 super关键字的使用6.1.3 限制子类的访问6.1.4 覆写6.2 抽象类6.3 Object类6.4 final关键字6.5 接口 ( interface ) 6.6 对象多态性6.6.1 instanceof关键字的使用6.6.2 覆写Object类中的equals ( ) 方法6.6.3 接口对象的实例化6.7 匿名内部类本章小结

## &lt;&lt;Java程序设计教程&gt;&gt;

## 章节摘录

插图：1.简洁有效Java语言是一种相当简洁的“面向对象”程序设计语言。

Java语言省略了C++语言中所有的难以理解、容易混淆的特性，例如头文件、指针、结构、单元、运算符重载、虚拟基础类等。

它更加严谨、简洁。

2.可移植性对于一个程序员而言，写出来的程序如果无须修改就能够同时在Windows、Macos、UNIX等平台上运行，简直就是美梦成真的好事！

而Java语言就让这个原本遥不可及的事越来越接近现实了。

使用Java语言编写的程序，只要做较少的修改，甚至有时根本无须修改就可以在不同平台上运行。

3.面向对象可以这么说，“面向对象”是软件工程学的一次革命，大大提升了人类的软件开发能力，是一个伟大的进步，是软件发展的一个重大的里程碑。

在过去的30年间，“面向对象”有了长足的发展，充分体现了其自身的价值，到现在已经形成了一个包含“面向对象的系统分析”、“面向对象的系统设计”、“面向对象的程序设计”的完整体系。

所以作为一种现代编程语言，是不能够偏离这一方向的，Java语言也不例外。

4.解释型Java语言是一种解释型语言，相对于C / C++语言来说，用Java语言写出来的程序效率低，执行速度慢。

但它正是通过在不同平台上运行Java解释器，对Java代码进行解释，来实现“一次编写，到处运行”的宏伟目标。

为了达到目标，牺牲效率还是值得的，况且，现在的计算机技术日新月异，运算速度也越来越快，用户是不会感到太慢的。

5.适合分布式计算Java语言具有强大的、易于使用的联网能力，非常适合开发分布式计算程序。

Java应用程序可以像本地文件系统一样通过URL访问远程对象。

使用Java语言编写Socket通信程序十分简单，使用它比使用任何其他语言都简单。

而且它还十分适用于公共网关接口（CGI）脚本的开发，另外还可以使用Java小应用程序（Java Applet）、Java服务器页面（Java Server Page, JSP）、Server等手段来构建更丰富的网页。

6.拥有较好的性能正如前面所述，由于Java是一种解释型语言，所以它的执行效率相对就会慢一些，但由于Java语言采用了两种手段，使得其性能还是不错的。

Java语言源程序编写完成后，先使用Java伪编译器进行伪编译，将其转换为中间码（也称为字节码），然后再进行解释。

提供了一种“准实时”（Just-in-time, JIT）编译器，当需要更快的速度时，可以使用JIT编译器将字节码转换成机器码，然后将其进行缓冲，这样速度就会更快。

## <<Java程序设计教程>>

### 编辑推荐

《Java程序设计教程(入门篇)》：高等教育应用型人才培养系列教材·科瑞国际企合作指定教材。

<<Java程序设计教程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>