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

金百东　刘德山　刘 丹　主　编

Java程序设计

学习指导与习题解答

JAVA CHENGXU SHEJI
XUEXI ZHIDAO YU XITI JIEDA

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 程序设计学习指导与习题解答

金百东　刘德山　刘　丹　主　编

科学出版社
北　京

内 容 简 介

本书是《Java 程序设计》一书的学习与实验指导配套教材。全书共分13章，按照Java SE基本内容自成体系，按照“源于主教材、高于主教材”的原则对主教材中的内容进行扩展。第1～11章结合主教材介绍了其主要内容、典型例题分析、问题与解答、实验与指导、习题与参考解答5个部分，案例设计由浅入深，讲解详尽，突出应用性和指导性，以知识点、范例为基础，精心设计了实验题目和章节习题。第12章设置有面向能力提升和自我测试的综合实验，第13章精选了3套模拟试题，努力提高学生独立解决问题的能力。

本书适合作为“Java 程序设计”课程的学习指导及实验教材，尤其适用于学生自学时使用。

图书在版编目(CIP)数据

Java程序设计学习指导与习题解答/金百东，刘德山，刘丹主编. —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 978-7-03-034461-8

Ⅰ. ①J… Ⅱ. ①金… ②刘… ③刘… Ⅲ. ①Java语言-程序设计-高等学校-教学参考资料 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2012）第107815号

策 划：隽青龙

责任编辑：隽青龙 / 责任校对：王万红

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

新科印刷有限公司 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：14 3/4

字数：335 000

定价：27.00元

（如有印装质量问题，我社负责调换〈新科〉）

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

前　言

本书是《Java 程序设计》的配套教学用书，是对主教材教学内容的补充和实践环节的完善。全书共分 13 章，第 1～11 章针对主教材，介绍了其主要内容、典型例题分析、问题与解答、实验与指导、习题与参考解答 5 个部分。

（1）主要内容

以知识点的方式总结本章的主要内容，包括基本的概念、方法和应用。对于部分较难掌握的内容，通过示例进行深入分析。

（2）典型例题分析

按照“源于主教材、高于主教材”的原则对主教材中的示例进行扩展。对一部分基础、重要、易混淆的示例进行细致讲解；对有代表性和可扩展的示例，进行了由浅入深的一系列解答，有助于读者学习和理解；对于部分较难理解的内容和因版本更新而增加的知识，以示例的形式进行了补充。

（3）问题与解答

最基本和最典型的知识点以问题的形式提出，并进行解答。

（4）实验与指导

实验与指导包括实验内容和实验指导两部分。实验内容提出需要完成的实验题目，实验指导根据各章节内容不同各有侧重。教材的前几章强调程序的运行环境和常见的语法错误；后面的章节多在程序的算法实现或程序框架方面给出指导，帮助学生完成实验。

（5）习题与参考解答

提供大量习题，读者完成习题后，可以检查对课程内容的掌握程度。对习题进行解答，对照参考解答，读者可以发现问题，有助于掌握所学知识。

第 12 章精心设计了综合实验，第 13 章精选了 3 套模拟试题。

本书由金百东、刘德山、刘丹任主编，参加编写的还有刘江平、张秋生。教材中的程序代码均运行通过，部分程序可能有多种实现方法。

由于编者水平有限，书中难免存在错误与不妥之处，欢迎读者和同行专家批评指正。

目　录

第 1 章　Java 语言概述

1.1　主 要 内 容

1. Java 语言的特点

Java 语言是一种典型的面向对象的程序设计语言，具有平台无关性、可移植性好、高度的稳定性和安全性等特点。平台无关性是 Java 较重要的特点之一，Java 程序的半编译、半解释的特点和虚拟机机制是实现平台无关性的重要原因。图 1-1 表现了 Java 程序从编写到运行的过程。

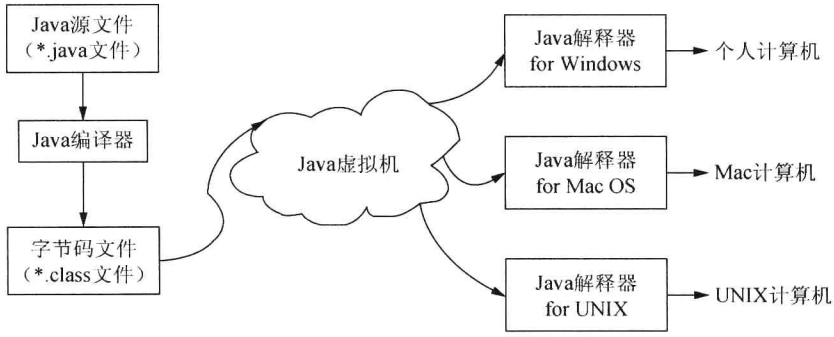

图 1-1　Java 程序的编译运行过程

编译器：对源代码进行半编译，生成与平台无关的字节码文件（.class）。

解释器：分布在不同的操作系统平台上，用于解释执行字节码文件，不同平台上的解释器是不同的。这些分布在不同平台上的解释器，又称虚拟机。

所以，通常把 Java 程序的翻译过程称为半编译、半解释，其他语言程序要么是全编译，要么是全解释。正因为如此，Java 程序才具有“write once ,run anywhere”的特点。

2. Java 程序及其结构

Java 程序分为两大类：一种是 Java Application，称为 Java 应用程序，是一种在 Java 虚拟机（缩写为 JVM）中独立运行的完整程序；另一种是 Java Applet，称为 Java 小程序，需要在浏览器中运行，是一种非独立的程序。

（1）Java Application

Java Application 有如下特点。

- 一个源文件可以由若干个类组成，但一个文件中只能有一个用 public 声明的类（全局类），并且，文件中如果包含了用 public 声明的类，那么源程序文件名一定和这个 public 类的类名相同。

- main()方法是 Java Application 的入口，一个 Java Application 只能有一个 main()方法。其严格定义格式如下。

```
public static void main(String args[]) {
}
```

（2）Java Applet

Java Applet 的特点如下。

- Applet 可以由若干个类组成，但其起始类必须是 javax.swing.JApplet 或 java.awt.Applet，即其起始类必须继承上面两个类中的一个。
- Java Applet 起始类必须被定义为 public，否则 Java 虚拟机无法创建这个类的实例。
- Java Applet 不像 Java Application 那样作为独立程序能够被直接执行，它必须由支持 Java 的浏览器加载执行。在执行 Java Applet 之前，必须先建立一个 HTML 文件，使用 HTML 标记把 Applet 嵌入其中。将 Applet 嵌入到 HTML 文件中使用的标记形式如下。

```
<applet code="…" width=… height=…>
</applet>
```

其中，<applet>和</applet>说明在此插入一个 Applet，字节码文件的名称放在“code=”后（扩展名.class 一定要有），而 width 和 height 参数说明了此 Applet 占用的宽度和高度。

实际上，Applet 可以被视为在 WWW 浏览器中执行的程序。Applet 运行的完整过程是，首先由 Java 编译器将其源代码编译生成通用的字节码文件，然后再编写一个 HTML 文件将该字节码嵌入其中。注意，应将此字节码文件和 HTML 文件保存在服务器的特定路径下。当 WWW 浏览器下载此 HTML 文件并显示时，它会自动下载其中指定的 Applet 字节码，然后调用内置在浏览器中的 Java 解释器来解释执行下载到本机的字节码程序。

Applet 除了在浏览器中运行，JDK（Java Development Kit）的小程序查看器 appletviewer 也可用来运行 HTML 文件中的 Applet。

3. Java 程序开发环境

（1）JDK

JDK，即 Java 开发包，是 Sun 公司推出的一种 Java 程序开发环境。开发者可以利用它来编译、运行和调试 Java 程序。JDK 主要包括 Java 虚拟机、Java 核心类库、命令行开发工具。

（2）Java 程序编辑环境

Java 程序编辑环境可以使用记事本，也可以使用 UltraEdit、EditPlus 等程序编辑器。

（3）集成开发环境

典型的集成开发环境包括 Eclipse、NetBean、JBuilder 等。其中，Eclipse 是一个开

放源代码的、基于 Java 的可扩展开发平台，提供了程序编译、程序调试、代码导入等功能。NetBean、JBuilder 和 Eclipse 功能类似。

一般在学习 Java 程序设计初期，建议使用 JDK 来开发，便于初学者掌握基本的语法和常用类。在学习后续课程或进行项目开发时，再使用集成开发环境来提高开发效率。

1.2　典型例题分析

1．阅读下面的程序，回答问题。

1）下面的两个类保存在一个 Java 源文件中，文件命名有什么要求？

2）源文件编译后，产生的字节码文件的文件名是什么？

3）说明程序的执行过程。

```
class Greeting {
    private String salutation;
    public Greeting(String s) {
        salutation = s;
    }
    public void greet(String whom){
        System.out.println(salutation + " " + whom);
    }
}
public class TestGreeting {
    public static void main(String[] args){
        Greeting g = new Greeting("Hello");
        g.greet("World");
    }
}
```

解析：

1）上面的两个类可以保存在同一个 Java 源文件中，但由于文件中包括了 public 声明的类，所以源文件名必须被命名为 public 的类名，即 TestGreeting.java。需要说明的是，一个源文件中只能有一个 public 类。

当然，上面的两个类也可以分别保存在两个文件中。

2）源文件编译后，源文件包括两个类，所以产生两个字节码文件，文件名分别是 Greeting.class 和 TestGreeting.class。

3）程序从 main()方法开始执行。

① 执行语句 Greeting g = new Greeting("Hello") ，调用构造方法 Greeting(String s)，同时将"Hello"传递给参数 s，然后将 s 的值送给 salutation，构造方法 Greeting(String s) 执行结束。

② 执行语句 g.greet("World")，调用方法 greet(string whom)，将"World"传递给参数 whom，然后输出 salutation+""+whom，即"Hello World "。

③ greet(string whom)方法执行结束，返回。main()方法执行结束，程序结束。

2．阅读下面的程序，回答问题。

1）程序前两行的导入语句有什么功能？

2）程序中有一个方法 paint(Graphics g)，可以用其他方法名来替代吗？为什么？

3）查阅 drawString()方法的 JDK 文档，了解这个方法中参数的意义。

```
import java.awt.Graphics;
import javax.swing.*;
public class Test extends JApplet {
    String str;
    public void init() {
        str = getParameter("city");
    }
    public void paint(Graphics g) {
        g.drawString("Hello applet", 50, 100);
        g.drawString(str, 50, 150);
    }
}
```

解析：

1）import java.awt.Graphics 用来导入 java.awt 包中的 Graphics 类。该类提供了若干种绘图的方法。本程序中，在 Applet 上显示了一个字符串。import javax.swing.* 用来导入 JApplet 类。

2）paint(Graphics g)方法不可以用其他方法名来替代，因为这个方法是重写 JApplet 类的方法，重写要求方法名、返回值、参数类型完全一样。

3）略。

3．为了运行前面的 Applet 程序，需要书写下面的 HTML 文件，并回答问题。

1）HTML 文件中各标签的含义是什么？

2）如何将 HTML 文件中的参数传递给 Applet？

3）说明 HTML 文件及 Applet 的执行过程。

```
<HTML>
<BODY>
<applet code="Test.class" width="400" height="400" >
    <param name="city" value="beijing">
</applet>
</BODY>
</HTML>
```

解析：

1）<HTML>和</HTML>表示 HTML 文件（网页文件）的开始和结束。所有 HTML 的其他标记都应置于这两个标记之间。

<BODY>和</BODY>是 HTML 文件的主体部分，所有需要在浏览器中显示的信息都应写在这两个标记之间。

<applet>和</applet>用于标识一个 Java Applet 程序，指明 Applet 的起始类。

2）可以在<applet>和</applet>之间嵌入 param 标记，以便从 HTML 文件向 Applet

传递参数。每个 param 标记包含两个属性，分别指定参数名（name）和取值（value）。本例中，参数名是“city”，取值是“beijing”。

3）本程序的执行过程如下。

① 在浏览器中打开 HTML 文件，遇到<applet>标签后，在当前文件夹下寻找“Test.class”文件并加载。

② Test.class 继承了 JApplet 类，首先自动执行 init()方法。在该方法中，调用了 JApplet 类的 getParameter("city")方法，该方法自动从 HTML 文件中读取 param，标记 name 为"city"的属性对应的 value 值，将 value 值“beijing”传递给变量 str。

③ init()方法执行后，浏览器自动调用 paint(Graphics g)方法，在浏览器中输出相应的信息。

4．分析输出“Hello Java!”的不同应用程序，了解 Java 程序的基本结构。

1）“Hello Java!”是一个字符串常量。

```
class HelloJava1 {
    public static void main(String[] args) {
        System.out.println("Hello Java!");    //输出字符串常量
    }
}
```

2）“Hello Java!”在局部变量 str 中。

```
class HelloJava2 {
    public static void main(String[] args) {
        String str = "Hello Java!"
System.out.println(str);    //输出字符串局部变量 str
    }
}
```

3）“Hello Java!”在静态成员变量 str 中。

```
class HelloJava3 {
    static String str;
    public static void main(String[] args) {
        str = "Hello Java!";
System.out.println(str);    //输出静态成员变量 str
    }
}
```

4）“Hello Java!”在命令行参数中。

```
class HelloJava4 {
    public static void main(String[] args) {
        System.out.println(args[0]);    //输出命令行参数
    }
}
```

5）交互式输入、输出“Hello Java!”。

```
class HelloJava5 {
```

```
        public static void main(String[] args) {
            java.util.Scanner sc = new java.util.Scanner(System.in);  //
构造 Scanner 类对象
            System.out.print("Please enter a word:");
            System.out.println(sc.next());                    //从控制台读
                                                              //取并输出数据
        }
    }
```

解析：

1）上面的 5 个例子中，第 1 个例子直接输出字符串常量"Hello Java!"，是最基本的输出方式。方法 System.out.println()功能是输出信息后换行，类似的方法 System.out.print()功能是不换行输出。

2）第 2 个例子将"Hello Java!"保存在局部变量 str 中，使用 System.out.println(str)来输出局部变量的值。

3）第 3 个例子将"Hello Java!"保存在静态成员变量 str 中，使用 System.out. println (str)来输出静态成员变量的值。与第 2 个例子的区别是，如果静态成员变量不赋值，程序编译会通过，输出的是默认值，这里的默认值是一个空串，如果第 2 个例子的局部变量没有赋值，程序编译时会报错。

4）第 4 个例子使用了命令行参数 args[]。命令行参数以 String 数组的形式传递给 main()方法，由参数 args[]接收。程序执行时，第 1 个参数传递给 args[0]，第 2 个参数传递给 args[1]……

在这个例子中，程序运行时，执行命令：`java Hello Java4  Hello Java`

其中，Hello Java4 是类名，类名后面的"Hello Java"即命令行参数。这里只向 args[0]传递了一个参数“Hello Java”，所以程序的输出结果为"Hello Java"。

5）命令行参数提供了向 main()方法传递数据的一个手段，实际上，更为常见的是，使用交互式命令来实现数据输入，第 5 个例子利用 Scanner 类来接收用户的输入。

Scanner 是 java.util 包中的一个类，可以从控制台获取输入数据。具体的做法是，先创建一个 Scanner 类的一个对象。例如，`java.util.Scanner sc = new java.util.Scanner (System.in);`其中，new 运算符使用标准输入对象 System.in 作为参数，然后再调用 Scanner 类对象的有关方法获取输入数据，主要方法包括以下几种。

- next()　　//获得一个字符串
- nextInt()　　//获得一个 int 型整数
- nextDouble()　　//获得一个 double 型浮点数
- nextBoolean()　　//获得一个 boolean 型数据

这个程序运行时，提示用户输入一个字符串，输入完成后，程序将输出输入的字符串。

1.3　问题与解答

1．什么是 Java 虚拟机？

2．什么是 JDK？

3．Java Application 和 Java Applet 的区别是什么？

4．什么是 Java 字节码文件？

解答：

1．简单地说，Java 虚拟机是一种软件系统，它可以翻译和运行 Java 的字节码文件。

2．JDK 是 Java Development Kit 的缩写，是 Sun 公司免费提供的开发、运行 Java 程序的系统软件。它包括了 Java 编译器和 Java API。

3．Java Application 是可以在 Java 虚拟机中独立运行的程序。Java Applet 是需要借助浏览器中的 Java 虚拟机来运行的程序。

4．字节码文件是 Java 编译器在编译 Java 源文件时生成的，扩展名是.class，可以在不同的 Java 虚拟机上运行。

1.4　实验与指导

1．实验内容

1）从 http://java.sun.com 下载 JDK，并安装和配置环境变量。

2）安装一种 Java 程序编辑环境，如 UltraEdit 或 EditPlus。

3）完成一个 Java Application，输出“Hello Java!”。

4）完成一个 Java Applet，输出“Hello Applet!”。

5）完成一个 Java Application，输出 1 到 100 内的偶数和。

6）完成一个 Java Applet，输出 10 的阶乘。

2．实验指导

刚开始学习 J2SE，建议使用文本编辑器和 MS-DOS 环境来编辑和运行 Java 程序，学习 Java SE 基本结束时，再使用集成开发环境，这样，有利于学生掌握 Java 基本类库和 Java SE 的基本知识。刚学习编辑和运行 Java 文件时，常见的错误如下。

（1）文件的保存路径

文件保存在 E:\java 文件夹下，编译时如果没有进入到包含 Java 源文件的文件夹，出现提示如图 1-2 所示。只有正确进入指定包含源文件的文件夹，才能找到文件进行编译。

图 1-2　找不到文件编译错误提示

（2）文件的扩展名

在 MS-DOS 状态下，编译源文件 HelloJava 时的命令如下。

```
Javac HelloJava.java
```

该命令需要指定扩展名.java。运行 HelloJava 时的命令如下。

```
Java HelloJava
```

这个命令不可以指明扩展名。同时需要注意，main()方法一定要存在于 HelloJava 这个类中。

为了很好地查看文件的扩展名，避免不必要的错误，建议编写 Java 程序前，将 Windows 操作系统设置为显示文件扩展名状态。具体操作如下。

双击文件夹图标，执行“工具→文件夹选项→查看”命令，弹出“文件夹选项”对话框，如图 1-3 所示。取消勾选“隐藏已知文件类型的扩展名”复选框。

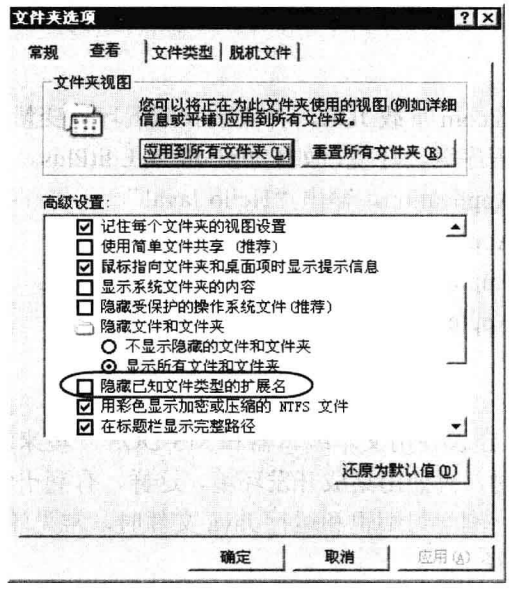

图 1-3　“文件夹选项”对话框

（3）文件名和标识符严格区分大小写的问题

Java 的文件名和标识符是严格区分大小写的，如果在输入程序代码或保存文件时，没有注意大小写的问题，将出现编译错误。中英文标点的问题和大小写的问题类似，尤其是从一些网络上下载的共享代码，中英文标点的错误可能很多。

（4）path 或 classpath 设置错误

JDK 平台提供的 Java 编译器（javac.exe）和 Java 解释器（java.exe）存在于 Java 安装目录的 bin 文件夹中，为了能在任何文件夹中使用编译器和解释器，需要设置 path 环境变量。同时，JDK 安装目录的 jre 文件夹中包含着 Java 应用程序运行时所需的 Java

基本类库，这些类库被包含在\jre\lib 文件夹中的压缩文件 rt.jar 中，使用各种 Java 开发工具时，为了能使用其中的类，需要设置 classpath 环境变量。

假设 JDK 的安装目录是 C:\Java\jdk1.6.0_23，则 path 变量应设置为 C:\Java\jdk1.6.0_23\bin；classpath 变量应设置为 C:\Java\jdk1.6.0_23\jre\lib\rt.jar。除了按照教材中的方法在 Windows 操作系统下设置外，也可以使用 DOS 命令来设置。

1.5　习题与参考解答

1．选择题。

1）关于 Java 语言特点的描述中，下列说法中错误的是（　　）。

A．支持多线程操作　　B．Java 程序的平台无关性

C．支持方法的重载和覆盖　　D．支持多继承和单继承

2）关于 Java Application 的叙述中，下列说法错误的是（　　）。

A．Java 程序是由一个或多个类构成的

B．Java Application 的若干个类可以保存在一个文件中，也可以保存在多个文件中

C．Java Application 的文件名要和类名统一

D．一个 Java 源程序文件最多包含一个 public 类

3）关于 Java 和 C++的比较，下列说法中错误的是（　　）。

A．Java 语言中取消了 goto 语句，但 goto 仍可视为 Java 的保留字

B．Java 语言中取消了指针的概念

C．Java 语言支持方法重载，但没有运算符重载

D．Java 语言保留了 C++中的结构的概念

4）下列关于 Java 源文件的说法中，正确的是（　　）。

A．一个源文件中可以有一个以上的公共类

B．一个源文件只能供一个程序使用

C．一个源文件只能有一个方法

D．一个程序可以包括多个源文件

5）关于 main()方法，下列说法中正确的是（　　）。

A．一个类可以没有 main() 方法

B．所有对象的创建都必须放在 main()方法中

C．main() 方法必须放在 public 类中

D．main()方法的声明可以根据情况修改

2．编程题。

1）设计一个 Applet 程序，在浏览器是绘制一个圆，圆心位置为（100，100），半径为 50。了解查 API 文档的基本方法。

2）设计并测试一个求长方体的体积和表面积的类。

参考解答：

1. D　C　C　D　A

2. 1）Applet 程序代码如下。

```
import java.awt.*;
import javax.swing.*;
public class TestApplet extends JApplet {
    public void paint(Graphics g) {
        g.setColor(Color.blue);          //圆的颜色
        g.fillOval(50,50, 100,100);      //圆心为(100, 100),半径为 50
        g.setColor(Color.red);           //圆心颜色
        g.drawString("*",100, 100);      //圆心
    }
}
```

HTML 文件如下。

```
<HTML>
<BODY BGCOLOR="#FFFFFF">
<applet code="TestApplet.class" width = 200 height=200>
</applet>
</BODY>
</HTML>
```

2）程序代码如下。

```
class TestCuboid {
    int longer,width,height;
    public static void main(String[] args) {
        Cuboid c = new Cuboid(3,3,4);
        System.out.println(c.getVolume());
        System.out.println(c.getSurfaceArea());
    }
}
class Cuboid {
    int longer,width,height;
    public Cuboid(int longer, int width, int height) {
        this.longer = longer;
        this.width = width;
        this.height = height;
    }

    int getVolume() {
        return longer*width*height;
```