

# Java

## 面向对象程序设计 题解与上机指导

王宏宇 贾仰理 主 编  
毋玉芝 刘 颖 白玉芹 副主编



21 世纪高职高专规划教材·计算机专业教育系列

# Java 面向对象程序设计 题解与上机指导

王宏宇 贾仰理 主 编

毋玉芝 刘 颖 白玉芹 副主编

中国人民大学出版社  
· 北京 ·

北京科海电子出版社  
[www.khp.com.cn](http://www.khp.com.cn)

## 图书在版编目(CIP)数据

Java 面向对象程序设计题解与上机指导/王宏宇, 贾仰理主编.

北京: 中国人民大学出版社, 2009

21 世纪高职高专规划教材·计算机专业教育系列

ISBN 978-7-300-10345-7

I. J…

II. ①王…②贾…

III. JAVA 语言—程序设计—高等学校—教学参考资料

IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 024558 号

21 世纪高职高专规划教材·计算机专业教育系列

Java 面向对象程序设计题解与上机指导

王宏宇 贾仰理 主编

---

出版发行 中国人民大学出版社 北京科海电子出版社

社 址 北京中关村大街 31 号 邮政编码 100080

北京市海淀区上地七街国际创业园 2 号楼 14 层 邮政编码 100085

电 话 (010) 82896442 62630320

网 址 <http://www.crup.com.cn>

<http://www.khp.com.cn> (科海图书服务网站)

经 销 新华书店

印 刷 北京市艺辉印刷有限公司

规 格 185 mm×260 mm 16 开本

版 次 2009 年 3 月第 1 版

印 张 13.75

印 次 2009 年 3 月第 1 次印刷

字 数 335 000

定 价 22.00 元

---

版权所有 侵权必究 印装差错 负责调换

# 内容提要

本书是与《Java 面向对象程序设计》配套的习题解答及上机指导教材。全书分为章节要点与习题解答，上机指导两部分。其中，章节要点与习题解答部分包括课本各章内容的要点总结、课后习题答案、补充习题及其解答；上机指导部分对 Java 程序设计上机实验的目的、步骤和方法进行介绍，提供了学习本课程应当进行的实验。

本书不仅可以作为《Java 面向对象程序设计》的配套实验指导书和参考书，而且可以作为其他 Java 面向对象程序设计学习的参考书；既适合高职高专院校师生和计算机培训机构使用，也可供报考计算机等级考试的读者使用。

# 前 言

面向对象程序设计已经取代了面向过程程序设计成为当前程序设计的主流方法。Java 是新一代面向对象的程序设计语言，它将平台无关性、面向对象、多线程、安全可靠、内嵌的网络支持等诸多特征集于一身，为软件开发人员提供了良好的编程环境，特别适用于 Internet/Intranet 上应用软件的开发，成为编写网络应用软件的首选语言。

为帮助读者学习 Java 面向对象程序设计的有关知识，使读者深入理解与掌握 Java 面向对象程序设计所涉及的概念、内涵、方法与技巧，提高程序设计能力，我们编写了本书。

本书可与《Java 面向对象程序设计》配合使用，也可供有一定基础的读者单独使用。本书包含两部分内容：第一部分是《Java 面向对象程序设计》的章节要点与习题解答，内容与课本各个章节完全对应。章节要点部分介绍了课本各章的主要内容，对各章的要点进行了总结；习题解答部分对各章课后习题进行了详细的讲解，并提供了补充练习题，给出了答案和解释。所有的编程题目都给出了完整的源程序代码，供读者参考使用。当然，很多题目是可以有多种解答的，我们这里给出一种供读者参考，并期待能启发读者给出更好的解答，希望读者在使用本书时，不要照抄照搬。

上机实验是学习任何程序设计语言必不可少的实践环节，可以提高学生的分析问题、解决问题的能力，通过实践环节理解 Java 语言的基本结构和程序设计方法。本书第二部分是 Java 面向对象程序设计上机指导，我们结合课本内容与学习 Java 面向对象程序设计的上机要求，提供了学习本课程应当进行的实验。通过上机实验，一方面可以加深对讲授内容的理解，熟悉程序开发的基本环境，学会编制和调试程序的基本方法。另一方面，可以结合课本更好地掌握常用程序设计的技巧，培养和提高动手能力。特别是 Java 语言是全新的、面向对象和网络的程序设计语言，要深入体会并掌握 Java 语言的平台无关性、面向对象、多线程、安全可靠、内嵌的网络支持等诸多特性，需要读者进行大量的编程实践。

本书内容丰富，概念清晰，实用性强，是学习 Java 语言的一本极佳参考书。它不仅可以作为《Java 面向对象程序设计》的参考书，而且可以作为其他 Java 语言学习的参考书；既适合高职高专院校师生或计算机培训机构使用，也可供报考计算机等级考试的读者使用。希望广大读者通过学习本书，在 Java 面向对象程序设计方面的知识能力能够得到较大的提高。

由于时间仓促，加之编者水平有限，不足之处在所难免，恳请广大读者指正。

编 者

2009 年 2 月

# 目 录

第一部分 章节要点与习题解答 .....	1	4.1.2 类的继承 .....	32
第 1 章 面向对象程序设计概述 .....	1	4.1.3 类的多态 .....	34
1.1 本章要点 .....	1	4.1.4 内部类 .....	34
1.1.1 程序设计方法的发展 .....	1	4.1.5 接口 .....	35
1.1.2 面向对象程序设计基本概念 .....	2	4.2 课后习题解答 .....	35
1.1.3 面向对象的基本特征 .....	2	4.3 补充习题 .....	39
1.1.4 面向对象程序设计语言 .....	3	4.4 补充习题解答 .....	41
1.2 课后习题解答 .....	3	第 5 章 包和 Java 基础类 .....	51
1.3 补充习题 .....	4	5.1 本章要点 .....	51
1.4 补充习题解答 .....	4	5.1.1 包 .....	51
第 2 章 Java 语言概述 .....	6	5.1.2 Java 类库常见包 .....	51
2.1 本章要点 .....	6	5.1.3 使用类库的方法 .....	52
2.1.1 Java 简介 .....	6	5.1.4 java.lang 包提供的常用 基础类 .....	52
2.1.2 Java 程序分类及其开发步骤 .....	7	5.1.5 java.util 包提供的常用类 .....	53
2.1.3 Java 开发工具与环境 .....	8	5.2 课后习题解答 .....	53
2.2 课后习题解答 .....	9	5.3 补充习题 .....	57
2.3 补充习题 .....	10	5.4 补充习题解答 .....	57
2.4 补充习题解答 .....	11	第 6 章 Java 异常处理 .....	62
第 3 章 Java 语言基础 .....	13	6.1 本章要点 .....	62
3.1 本章要点 .....	13	6.1.1 异常的概念 .....	62
3.1.1 Java 语言的基本组成 .....	13	6.1.2 异常处理类的层次结构 .....	62
3.1.2 基本数据类型 .....	13	6.1.3 异常处理机制 .....	63
3.1.3 运算符与表达式 .....	14	6.1.4 创建用户自己的异常 .....	64
3.1.4 Java 语句 .....	15	6.2 课后习题解答 .....	65
3.1.5 流程控制 .....	16	6.3 补充习题 .....	68
3.1.6 数组 .....	16	6.4 补充习题解答 .....	70
3.2 课后习题解答 .....	17	第 7 章 Java 图形用户界面设计 .....	74
3.3 补充习题 .....	21	7.1 本章要点 .....	74
3.4 补充习题解答 .....	25	7.1.1 Java 图形用户界面 GUI .....	74
第 4 章 类、对象和接口 .....	31	7.1.2 java.awt 包 .....	74
4.1 本章要点 .....	31	7.1.3 Swing 包 .....	77
4.1.1 类和对象 .....	31	7.1.4 AWT 图形设计 .....	77





7.2	课后习题解答	79	11.1.1	图像处理	126
7.3	补充习题	85	11.1.2	动画处理	126
7.4	补充习题解答	87	11.1.3	音频处理	127
<b>第 8 章</b>	<b>Applet 及其应用</b>	<b>93</b>	11.2	课后习题解答	128
8.1	本章要点	93	11.3	补充习题	131
8.1.1	Java Applet 基础	93	11.4	补充习题解答	131
8.1.2	Applet 的运行机制	94	<b>第 12 章</b>	<b>网络编程</b>	<b>135</b>
8.1.3	HTML 向 Applet 传递参数	95	12.1	本章要点	135
8.1.4	Applet 应用	95	12.1.1	Java 网络编程基础知识	135
8.2	课后习题解答	96	12.1.2	使用 InetAddress 类	136
8.3	补充习题	97	12.1.3	URL 网络编程	136
8.4	补充习题解答	99	12.1.4	Socket 网络编程	137
<b>第 9 章</b>	<b>I/O 系统</b>	<b>102</b>	12.1.5	UDP 网络编程	138
9.1	本章要点	102	12.2	课后习题解答	138
9.1.1	I/O 流概述	102	12.3	补充习题	148
9.1.2	字节流处理	102	12.4	补充习题解答	148
9.1.3	字符流处理	103	<b>第二部分</b>	<b>上机指导</b>	<b>152</b>
9.1.4	文件处理	103	一、	上机实验目的	152
9.2	课后习题解答	104	二、	上机实验步骤	153
9.3	补充习题	107	三、	实验基本内容	154
9.4	补充习题解答	108	<b>实验 1</b>	<b>安装、配置与应用 Java</b>	
<b>第 10 章</b>	<b>多线程编程</b>	<b>113</b>	开发工具	155	
10.1	本章要点	113	一、	实验目的	155
10.1.1	程序、进程和线程	113	二、	实验内容	155
10.1.2	线程的状态与生命周期	113	<b>实验 2</b>	<b>Java 语言基础</b>	<b>161</b>
10.1.3	多线程	114	一、	实验目的	161
10.1.4	通过继承 Thread 类创建 线程	114	二、	实验内容	161
10.1.5	实现 java.lang.Runnable 接 口创建线程	115	三、	参考程序	162
10.1.6	线程的调度和优先级	115	<b>实验 3</b>	<b>类、对象和接口</b>	<b>167</b>
10.1.7	基本的线程控制方法	115	一、	实验目的	167
10.1.8	Java 多线程的互斥与同步	116	二、	实验内容	167
10.2	课后习题解答	117	三、	参考程序	167
10.3	补充习题	121	<b>实验 4</b>	<b>Java 基础类的使用</b>	<b>175</b>
10.4	补充习题解答	122	一、	实验目的	175
<b>第 11 章</b>	<b>多媒体编程</b>	<b>126</b>	二、	实验内容	175
11.1	本章要点	126	三、	参考程序	175

实验 5 Java 异常处理.....	179	实验 9 多线程编程.....	197
一、实验目的.....	179	一、实验目的.....	197
二、实验内容.....	179	二、实验内容.....	197
三、参考程序.....	179	三、参考程序.....	197
实验 6 Java 图形用户界面设计.....	182	实验 10 多媒体编程.....	203
一、实验目的.....	182	一、实验目的.....	203
二、实验内容.....	182	二、实验内容.....	203
三、参考程序.....	182	三、参考程序.....	203
实验 7 Applet 及其应用.....	189	实验 11 网络编程.....	207
一、实验目的.....	189	一、实验目的.....	207
二、实验内容.....	189	二、实验内容.....	207
三、参考程序.....	189	三、参考程序.....	207
实验 8 I/O 系统.....	194	参考文献.....	212
一、实验目的.....	194		
二、实验内容.....	194		
三、参考程序.....	194		



# 第一部分 章节要点与习题解答

## 第 1 章 面向对象程序设计概述

### 1.1 本章要点

#### 1.1.1 程序设计方法的发展

##### 1. 早期程序设计方法和语言的发展

程序设计就是针对某一要处理的问题，按照特定的程序设计方法设计出解决该问题的计算机指令序列。进行程序设计要借助某种计算机语言来编写程序，这种计算机语言我们称为程序设计语言。

程序设计方法经历了面向机器（Machine-Oriented）、面向过程（Procedure-Oriented）和面向对象（Object-Oriented）的发展历程。面向机器的程序设计方法使用针对特定机器型号的低级语言开发程序，不利于程序的编写和维护，程序的生产效率很低，质量难以保证，可移植性差。

在面向过程程序设计中，问题被看作一系列需要完成的任务，相应的函数用于完成这些任务，这些函数是面向过程的，即函数关注如何根据规定的条件完成指定的任务。早期面向过程的高级程序设计语言有 FORTRAN、ALGOL、BASIC 等。结构化程序设计方法主要思想是采用自顶向下、逐步求精的方法，将整个程序结构划分成若干个功能相对独立的模块，模块之间的联系尽可能简单；每个模块用顺序、选择、循环三种基本结构来实现；每个模块只有一个入口和一个出口。结构化程序设计方法的代表语言是 C、Pascal、Ada 等。

##### 2. 面向对象程序设计方法

面向对象的程序设计方法以类作为构造程序的基本单位，具有封装、数据抽象、继承、多态性等特点。其基本思想是使用对象、类、消息等基本概念来进行程序设计，从现实世界中客观存在的事物（即对象）出发来构造软件系统，并且在系统构造中尽可能运用人类的自然思维方式。



常见的面向对象程序设计语言包括 Simula、Smalltalk、Eiffel、C++、Java 等。

### 1.1.2 面向对象程序设计基本概念

“面向对象”是由英文 Object Oriented 翻译而来的，简称为 OO。Java 语言是一种完全面向对象的程序设计语言。

#### 1. 对象

对象是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位，是一组属性和对这些属性进行操作的一组方法的封装体，具有唯一的名字。对象的属性即一组数据，用来描述对象的静态特征。对象的方法也称为成员方法，或称为服务、操作，它是对象动态特征（行为）的描述。每一个方法确定对象的一种行为或功能。

#### 2. 类

类是具有相同属性和操作的一组对象的集合，它为属于该类的所有对象提供了统一的抽象描述，其内部包括属性和方法两个主要部分。

在面向对象的编程语言中，类是一个独立的程序单位，描述一个类需要指明下述三个方面的内容：

- (1) 类标识：类的一个有别于其他类的名字。这是必不可少的。
- (2) 属性说明：用来描述相同对象的静态特征。
- (3) 方法说明：用来描述相同对象的动态特征。

面向对象程序设计技术采用这样一种思想：先定义类，再利用类来实例化对象。类与对象的关系就如模具和铸件的关系，类的实例化结果就是对象，而对一组对象的抽象就是类。

#### 3. 消息

对象之间利用消息进行交互。应用程序通常包含多个对象，通过这些对象的交互作用，程序可以完成相应的功能以及更为复杂的行为。消息通常包含三个方面的信息：接受消息的对象、需要执行的方法和执行方法需要的所有参数。

### 1.1.3 面向对象的基本特征

#### 1. 抽象性

抽象是人类对事物进行分类的最基本的方法和手段。面向对象程序设计中的抽象是对一类对象进行分析和认识，经过概括，抽出一类对象的公共性质，并加以描述的过程。

对一个事物的抽象一般包括两个方面：数据抽象和行为抽象。数据抽象是对对象的属性和状态的描述，使对象之间互相区别的特征量的描述。行为抽象是对数据需要进行处理的描述，它描述了一类对象的共同行为特征，使一类对象具有共同的功能，因此，又称行为抽象为代码抽象。

#### 2. 封装性

封装是把对象的数据（属性）和行为（方法）绑定在一起的一种机制，该机制保证了数

据和方法都不受外部干扰且不被误用。封装性指尽可能隐蔽对象的内部细节，它包含两个含义：把对象的全部属性和全部行为结合在一起，形成一个不可分割的独立单位（即对象）；信息隐蔽，即尽可能隐蔽对象的内部细节，对外形成一个边界，只保留有限的对外接口使之与外部发生联系。

### 3. 继承性

特殊类的对象拥有其一般类的全部属性与方法，称做特殊类对一般类的继承。一个类可以是多个一般类的特殊类，它从多个一般类中继承了属性与方法，这称为多继承。在 Java 语言中，通常我们称一般类为父类（superclass，超类），特殊类为子类（subclass）。通过父类和子类，我们实现了类的层次化，可以从最一般的类开始，逐步特殊化，定义一系列的子类。同时，通过继承也实现了代码的复用。

### 4. 多态性

多态是指一个程序中同名的不同方法共存的情况。这些方法同名的原因是它们的最终功能和目的都相同，但是由于在完成同一功能时，可能遇到不同的具体情况，所以需要定义含不同的具体内容的方法，来代表多种具体实现形式。它的目的是为了提高程序的抽象度、封闭性和简洁性，统一一个或多个相关类对外的接口。

## 1.1.4 面向对象程序设计语言

目前，比较有影响的面向对象程序设计语言有 Simula、Smalltalk、Object-c、Eiffel、C++、Java 等。Java 语言是 Sun 公司于 1995 年推出的、适用于分布网络环境的面向对象语言。它采用了与 C++ 语法基本一致的形式，将 C++ 中与面向对象无关的部分去掉，使其语义成为纯面向对象的语言。它使应用程序独立于异构网络上的多种平台，具有能编译或解释执行、连接简单、支持语言级的多线程等特点，也是一种目前广泛使用的面向对象语言。

## 1.2 课后习题解答

### 1. 什么是类？什么是对象？它们之间的关系是怎样的？

【答案】类是具有相同属性和方法的一组对象的集合，它为属于该类的所有对象提供了统一的抽象描述，其内部包括属性和操作两个主要部分。

对象是一组属性和对这些属性进行操作的一组方法的封装体，具有唯一的名字。对象的属性即一组数据，用来描述对象的静态特征。

类与对象的关系就如模具和铸件的关系，类的实例化结果就是对象，而对一组对象的抽象就是类。

### 2. 面向对象的核心特性是什么？

【答案】封装性：封装是把对象的数据（属性）和行为（方法）绑定在一起的一种机制，该机制保证了数据和方法都不受外部干扰且不被误用。封装性指尽可能隐蔽对象的内部细节，



它包含两个含义：把对象的全部属性和全部行为结合在一起，形成一个不可分割的独立单位（即对象）；信息隐蔽，即尽可能隐蔽对象的内部细节，对外形成一个边界，只保留有限的对外接口使之与外部发生联系。

**继承性：**特殊类的对象拥有其一般类的全部属性与方法，称做特殊类对一般类的继承。一个类可以是多个一般类的特殊类，它从多个一般类中继承了属性与方法，这称为多继承。子类继承父类的所有状态和行为，同时增加自己的状态和行为。通过父类和子类，我们实现了类的层次，可以从最一般的类开始，逐步特殊化，定义一系列的子类。同时，通过继承也实现了代码的复用。

**多态性：**多态是指一个程序中同名的不同方法共存的情况。这些方法同名的原因是它们的最终功能和目的都相同，但是由于在完成同一功能时，可能遇到不同的具体情况，所以需要定义含不同的具体内容的方法，来代表多种具体实现形式。它的目的是为了提高程序的抽象度、封闭性和简洁性，统一一个或多个相关类对外的接口。

3. 什么是类的继承？它有什么作用？

『答案』当定义一个新类时，不必写出全部成员变量和成员方法。只要简单地声明这个类是从一个已定义的类继承的。被继承的类称为父类或超类，这个新类就是子类。这样，子类从现有的类中产生，保留了现有类的属性和方法并可以根据需要加以修改。

继承是实现软件可重用性的一种重要方式，可以增强软件的可扩充性，提高软件的可维护性。后代类可继承并获得祖先类的成员，使祖先类的优良特性得以代代相传。如果更改祖先类中的内容，这些修改过的内容将直接作用于后代类，后代类本身无须进行维护工作。同时，后代类还可以增加自己的成员，从而不断地扩充功能，或者重写祖先类的方法以适应新的需求。

4. 什么是类的多态性？

『答案』多态即程序中同名的不同方法共存的情况。这样，一个方法可以有多个实现版本，程序运行时，系统会根据方法的参数或调用方法的对象自动选择一个方法执行，不会产生混乱和混淆。

## 1.3 补充习题

1. 面向对象程序设计与面向过程程序设计的最大区别是什么？
2. 什么是消息？它通常需要包含哪些信息？

## 1.4 补充习题解答

1. 『答案』在面向过程程序设计中，问题被看作一系列需要完成的任务，相应的函数用于完成这些任务，这些函数是面向过程的，即函数关注如何根据规定的条件完成指定的任务。

面向对象的程序设计方法以类作为构造程序的基本单位，具有封装、数据抽象、继承、多态性等特点。其基本思想是使用对象、类、消息等基本概念来进行程序设计，从现实世界中客观存在的事物（即对象）出发来构造软件系统，并且在系统构造中尽可能运用人类的自然思维方式。

2.『答案』对象之间利用消息进行交互。消息传递的实质是方法的调用。应用程序通常包含多个对象，通过这些对象的交互作用，程序可以完成相应的功能以及更为复杂的行为。消息通常包含三个方面的信息：接受消息的对象、需要执行的方法和执行方法需要的所有参数。

## 第 2 章 Java 语言概述

### 2.1 本章要点

#### 2.1.1 Java 简介

Java 语言是广泛应用的面向对象程序设计语言，它具有简单、面向对象、与平台无关、安全、高效、多线程等特性，它是在 Oak 语言基础上发展起来的。

##### 1. 简单性

Java 摒弃了 C++ 中容易引发程序错误的地方，如指针、多重继承、操作符重载等。Java 简化了内存管理和文件管理，而且提供了 C++ 中不具备的自动内存回收机制。此外，Java 还提供了丰富的类库。

##### 2. 面向对象特性

面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的，不允许定义独立于类的变量和方法，任何变量和方法都只能包含于某个类的内部。这就使程序的结构更加清晰，为继承和重用带来便利。

##### 3. 分布式

Java 的客户机/服务器模式，可以把计算任务从服务器分散到不同的客户端进行处理，从而提高整个系统的执行效率，避免瓶颈制约，增加动态可扩充性。Java 网络类库是支持 TCP/IP 协议的子例程库，目前支持 HTTP 和 FTP 等协议。

##### 4. 解释执行

Java 程序的最终执行要需经过两个步骤：编译和解释。Java 程序经过编译形成字节码，然后在虚拟机上解释执行。任何一台机器，只要配备了虚拟机，就可以运行 Java 字节码，而不管这种字节码是在何种平台上生成的。Java 通过预先把源程序编译成字节码，提高了传统解释型语言的执行效率。

##### 5. 健壮性

Java 的强类型机制、异常处理、自动内存回收等是 Java 程序健壮性的重要保证。另外，与 C/C++ 不同，Java 中不存在指针，Java 程序不可能非法访问内存，也不可能意外覆盖内存缓



冲突的区域。Java 的安全检查机制也使得 Java 更具健壮性。

## 6. 安全性

Java 摒弃 C++ 语言在安全性和稳定性方面造成很多问题的指针数据类型，同时提供了数组下标越界检查机制。在编译时，Java 会进行语法、语义的检查。链接时，Java 还要再进行一遍编译级的类型检查，消除间接对象访问。运行时，Java 的运行时系统将进行字节码检验，并记录对象的存储情况，将访问限制在安全范围之内。本地的类与远程的类分开运行，阻止远程系统对本地系统的破坏。支持 Java 的浏览器还允许用户控制 Java 软件对本地系统的访问。各种综合的措施使 Java 程序的安全性得到保证。

## 7. 平台独立性和可移植性

Java 的应用程序接口 (API) 和运行时系统是可移植性的关键。Java 为支持它的各种操作系统提供了一致的 API。在 API 界面上，所有 Java 程序将都不依赖于平台。Java 的运行时系统在解释执行程序时，将字节码转化为当前机器的机器码。程序开发人员无须考虑应用时的硬件条件和操作系统结构，用户只需有 Java 的运行时环境，就可运行编译过的字节码。

## 8. 高性能

Java 字节码可以快速地转换成为机器码来执行，而且 Java 字节码格式的设计就是针对机器码的转换，实际转换时相当简便，自动的寄存器分配与编译器对字节码的一些优化可使之生成高质量的代码。随着 Java 虚拟机的改进和“即时编译” (just in time) 技术的出现使得 Java 程序的执行速度有了更大的提高。

## 9. 多线程机制

首先，Java 环境本身就是多线程的，它可以利用系统的空闲时间来执行诸如必要的垃圾清除和一般性的系统维护等操作；其次，Java 还提供了对多线程的语言级支持，利用 Java 的多线程编程接口，编程人员在程序中可以方便地创建多个线程，各个线程执行不同的工作。为了控制各个线程的动作，Java 还提供了线程同步机制，使不同线程在访问共享资源时能够相互配合，保证数据的一致性，避免出错。Java 的多线程机制提高了程序的执行效率，大大促进了程序的动态交互性和实时性。

## 10. 动态性

Java 的动态特性是其面向对象设计方法的发展。它允许程序动态地装入运行过程中所需要的类，这是 C++ 等面向对象程序设计语言所无法实现的。

### 2.1.2 Java 程序分类及其开发步骤

Java 程序分为两类：可独立解释执行的应用程序 (Application) 和需放入 HTML 文件中依靠浏览器显示结果的小应用程序 (Applet)。

#### 1. Java Application

Java Application 是可独立解释执行的应用程序。一个 Java Application 可以有多个类，但是最多只能有一个 public 类。如果有一个类是 public 类，则源文件的名称必须与这个类



的名字完全相同。每个类可以有多个方法，任何方法中可以有多条语句，每一条语句都要以“；”结束。对于一个 Java Application，必须有一个 main() 方法，该方法标志着执行应用程序的起点。

使用 Java 语言编写 Java Application 一般分成三个步骤：建立 Java 源程序，即利用某一种文本编辑器建立 Java 源程序文件，扩展名为 .java；编译源程序，利用 Java 编译器 (javac) 编译程序，产生 .class 字节码文件；运行 Java 程序，利用解释器 (java) 解释字节码文件，完成程序的运行过程。

Java 语言具有半编译、半解释的特性，Java 程序编译后并不生成可执行文件，而是生成字节码 (byte code) 这种与平台无关的中间格式文件，然后再由 Java 虚拟机解释执行。

### 2. Java Applet

Java Applet 没有 main() 方法，不能用 Java 解释器直接执行，必须嵌在超文本文件中，并由带有 Java 解释器的 WWW 浏览器 (如 Internet Explorer、Netscape 等) 来解释执行。将 Applet 称为小程序的原因就是因为其代码较少，易于从 Internet/Intranet 下载。

Java Applet 的建立及运行步骤与 Java Application 略有不同，可分为 4 个步骤：利用文本编辑器建立 Java 源程序文件；利用 Java 编译器 (javac) 编译该 Java Applet，产生 .class 文字码文件；建立一个 HTML 文件，在其中嵌入 Java 字节码文件；用 WWW 浏览器或 appletviewer 装入该 HTML 文件，使 Applet 运行。

### 2.1.3 Java 开发工具与环境

Java 程序可以使用 Sun 公司的 Java 软件开发工具包 Java 2 SDK (Java 2 Software Development Kit) 采用命令行方式来开发，也可应用 JCreator、JBuilder 等可视化的集成开发环境来开发。

#### 1. Java 2 SDK

Java 2 SDK 又称为 JDK (Java Development Kit, Java 开发工具包)，它是一个简单的命令行工具，主要包括系统类库、编译 Java 源代码的编译器、运行 Java 字节码的解释器，以及测试 Java Applet 的 appletviewer 阅读器，还有其他一些有用的工具。它主要是通过命令行方式，在 DOS 环境下进行 Java 程序的编译和运行。虽然它的界面不如一些可视化工具友好，但却是其他各种开发工具的基础。从 Sun 公司的网站 <http://java.sun.com/j2se/1.4.2/download.html> 可以下载 Java 2 SDK 工具包。

#### 2. JCreator

JCreator 是由 Xinox 软件公司开发的一个可视化的 Java 程序集成开发环境。它是一个小巧灵活的 Java 开发工具，可将 Java 的程序的编写、编译、运行和调试集成到自身的环境中直接进行开发，且无须进行环境变量的设定。我们可以直接在 JCreator 中编辑 Java 源文件，选择相应的菜单和单击对应的按钮就可以完成 Java 程序的编译、运行等工作，十分方便。相对其他集成开发环境来说，JCreator 对系统要求较低，运行速度较快。用户可以通过访问网址 <http://www.jcreator.com/download.htm> 下载需要的 JCreator 安装程序。

## 2.2 课后习题解答

### 1. 简述 Java 语言的主要特点。

『答案』Java 是一种具有“简单、面向对象、分布式、解释型、健壮、安全、与平台无关、可移植、高性能、多线程和动态执行”等特性的语言。主要特点简述如下：

**简单性：**Java 摒弃了 C++ 中容易引发程序错误的地方，如指针等；简化了内存管理和文件管理，提供了 C++ 中不具备的自动内存回收机制；提供了丰富的类库，程序开发人员可以调用本地甚至远程的类库。

**面向对象特性：**Java 语言的设计完全是面向对象的，不允许定义独立于类的变量和方法，任何变量和方法都只能包含于某个类的内部。

**安全性：**Java 摒弃了 C++ 语言在安全性和稳定性方面造成很多问题的指针数据类型，同时提供了数组下标越界检查机制。在编译时会进行语法、语义的检查。另外，在链接和运行时，Java 也提供了安全机制。

**平台独立性和可移植性：**Java 的应用程序接口（API）和运行时系统是可移植性的关键。Java 为支持它的各种操作系统提供了一致的 API。在 API 界面上，所有 Java 程序将都不依赖于平台。用户只需有 Java 的运行时环境，就可运行编译过的字节码。

Java 还具有多线程等其他特性。

### 2. 分别简述 Java 应用程序和小应用程序的开发及运行步骤。

『答案』用 Java 2 SDK 开发 Java 应用程序步骤如下：

(1) 用任何一个文本编辑器来建立 Java 源程序文件。

(2) 用 Java 编译器把 Java 源程序编译成以 .class 为后缀的与处理器无关的二进制字节码文件，若源文件中存在错误，则编译器将指出错误位置，此时需返回上一步对源文件进行修改，修改后再进行编译，直到编译通过。

(3) Java 字节码文件由 Java 解释器（java）负责装入和执行。

Java 小应用程序的开发步骤如下：

(1) 利用文本编辑器建立 Java 源程序文件。

(2) 利用 Java 编译器（javac）编译该 Java Applet，产生 .class 文字码文件。

(3) 建立一个 HTML 文件，在其中嵌入 Java 字节码文件。

(4) 用 WWW 浏览器或 appletviewer 装入该 HTML 文件，使 Applet 运行。

### 3. Java 对源程序文件的命名规则有什么要求？源程序文件编译后生成的是什么文件？

『答案』一个 Java 程序可以有多个类，但是最多只能有一个 public 类。如果有一个类是 public 类，则源文件的名字必须与这个类的名字完全相同。文件扩展名为 .java。源程序文件编译后生成的是以 .class 为后缀的与处理器无关的二进制字节码文件。

4. 参照本章的例子，创建一个名为 NewJavaApp 的 Java Application，在屏幕上显示“I am a new Java Programmer!”。

参考程序：

```
public class NewJavaApp
{
    public static void main(String srg[ ])
    {
```