JAVA的核心技术

——面向对象编程

古新生 编著

西安交通大学出版社

TP312
956

JAVA的核心技术
——面向对象编程

古新生　编著

BBP09/04

西安交通大学出版社
·西 安·

内容简介

Java 语言是一种纯面向对象的具有高性能、分布式且独立于平台的网络编程语言。Java 是开发 Internet/Intranet 应用程序的最佳工具，必然会成为计算机类课程的重要教学内容。作者长期从事面向对象方法的研究并且近年来着重从事 Java 的教学与研究工作，本书以剖析 Java 的面向对象技术为主干，结合大量有说服力并经具体实现的 Applet(网上使用最多的 Java 小应用程序)编程实例来介绍 Java 语言。

本书不仅注重理论的正确，还十分重视上机编程，每章都有相应的实例，并都成功运行；每章的习题亦注意到理论的理解与实际编程能力的培养。因此，可作为计算机专业大、专学生的教学用书，并可作为从事网络应用与网页设计的技术人员、研究生，从事与此相关的教学工作人员学习面向对象技术和 Java 编程技术的教材或教学参考书。对所有理工科高等学校相关专业的师生及想学习和掌握面向对象方法的工程技术人员也是一本很好的入门参考书。

图书在版编目(CIP)数据

JAVA 的核心技术：面向对象编程／古新生编著. —西安：西安交通大学出版社，2002.8
ISBN 7-5605-1518-5

Ⅰ.J… Ⅱ.古… Ⅲ.JAVA 语言—程序设计
Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 015038 号

*

西安交通大学出版社出版发行
(西安市兴庆南路 25 号 邮政编码：710049 电话：(029)2668315)
陕西省轻工印刷厂印装
各地新华书店经销

*

开本：787mm×1092mm 1/16 印张：9 字数：209 千字
2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷
印数：0001～5 000 定价：13.00 元

发行科电话：(029)2668357，2667874

前言

Foreword

新世纪之初，面向对象技术正在逐步取代面向过程（结构化）的开发方法，成为计算机应用开发领域的主流。究其原因，是由于面向对象技术具有对现实世界最直接最自然的描述，极强的抽象能力和表达能力，并做到将数据与对数据的操作封装在一起成为一个“软件的组件”。更具特色的是面向对象方法有着独特的继承性，以及与此紧密相联的、并可提高抽象、封装程度的多态性。面向对象的开发方法大大地减轻与简化了软件的开发工作，优化了程序结构。所有这些特点适应了当代大规模软件开发与高新技术研制的要求。

面向对象方法与技术（Object - Oriented Paradigm and Technology）起源于面向对象的编程语言（OOPL），而 Java 语言正是一种纯面向对象的，在 Internet 网上最流行的编程语言。Java 语言诞生于 1995 年，仅用了短短的几年时间就风行全球。它除了具有面向对象特点之外，还具有与平台无关，可内嵌于 HTML 或 XML，高可靠、多线程和异常处理等等宝贵而优越的特性。正是由于 Java 具有的众多优异特性，才使它成为网络编程的首选工具之一。因此，本书选取 Java 语言作为学习面向对象技术的最佳实例语言。

本书特色

1. 将面向对象编程与当前最为流行的 Java 语言结合起来，以纯面向对象的 Java 语言为实例，深入学习面向对象的核心概念。由于 Java 较其他的面向对象语言更简单、易学且用途更广，容易引起学习的兴趣并增强教学效果。

2. 本书是作者多年从事面向对象技术研究与 Java 教学的结晶。书中的理论部分都经作者理解和深入思考，并充分考虑学生学习中容易理解错或上机操作易犯的错误等情况，尽量注意引导；做到说理清楚、逻辑严密、层次分明、通俗易懂；例子选取既与理论部分联系紧密，又切合实际，并经过上机验证。

3. 为了引导读者自学，除了注意内容的安排与叙述做到思路清晰、条理分明、由浅入深之外，每章前有扼要的综述，每章结束有小结，每章最后有复习题与练习题；其中，复习题侧重概念的理解，练习题侧重于上机实践。

4. 注意强调关键和核心，采用字体加粗，或用注意、小窍门，或用图形或用表格等等方法，帮助读者更易于接受。

5. 为了帮助读者掌握这门课的专业英语词汇，本书的章节标题都加上对应的英语，旨在促进读者参考有关的英语资料。

感谢

这本书得以和读者见面，我要衷心感谢所有帮助我完成写书的领导、同事、上过课的学生以及国内外的家人。也要感谢西安交通大学出版社本书的责任编辑赵丽萍老师和叶涛老师，还有负责校对与印制本书的同志们。

希望

希望本书的出版有助于广大读者学习面向对象技术和 Java 语言。由于作者水平所限，时间较为仓促，书中难免有不足之处，敬请本领域的专家、教授、学者、学习本课程的同学们和广大读者提出宝贵意见。

编者

2002 年 6 月

本书导读

Reading Guide

本书的第1章是面向对象技术与Java语言的概述。首先，从软件开发的角度综述了过程化(也称为结构化)开发方法与面向对象开发方法的比较，进而介绍面向对象的基本概念和Java语言的特点，以求为全面深入学习本书奠定基础。

第2章进入Java的编程学习。首先介绍Java程序的两大类型，这是由于在后面的Java程序设计中都离不开这两种类型，尤其是能在网络上运行的小应用程序(Applet)会作为本书举例的重点。

第3章和第4章是面向对象编程基本知识的重点与核心。第3章从Java编程的角度阐明了类和对象，从而更深入地理解抽象与封装这两个重要特性。第4章则介绍类层次，深入剖析面向对象方法的最重要特色——继承性，以及与继承性紧密相连的多态性。

第5章讨论接口和包，接口是为了克服Java单继承性的不足并进一步增强其灵活性提出的，包是为了更好地组织与管理Java的程序和有关的资源。

第6章与第7章介绍内容十分丰富多彩的Java的图形用户界面(具体介绍AWT)。其中第6章首先阐明Java的抽象窗口的概念，然后说明Java的最常用的图形用户界面；第7章则是Java图形用户界面较为深入的内容，在阐明事件处理机制概念之后、深入探讨了基本组件、容器及相应的事件。并叙述了各种布局管理器。这两章的学习除了是对这部分重要内容的学习之外，它又是进一步巩固和加深对面向对象方法重要内容(第3与第4章)的深入理解与具体应用。由于这两章的内容较多，在教学过程中可选择部分作为自学或结合前面的教学自编程时的实例参考。

第8章介绍Java编程的又一特色——多线程，Java正是有了多线程的支持，才能使网页变得生动、活泼。最后一章介绍Java的异常处理机制，有了这种机制才使Java的程序更加安全。

本书的读者

本书是为已具备面向过程编程语言而要进一步学习面向对象技术的读者而写的，它是学习面向对象技术和Java语言的一本很好的自学入门书。对有过编程实践的读者进一步加深面向对象方法的认识和提高Java编程能力也很有用。

本书也可以作为大学本、专科学生学习面向对象方法的教材或教学参考书。书中对内容的编排、选取都基于多年教学与面向对象方法研究与实践，经过精心的考虑，力求做到由浅入深、循序渐进。学习本书之前应对计算机操作有一定的掌握，但不必有软件开发经验。

学好面向对象方法和Java语言的建议

1. 注意认真学习与领会书中的基本概念

对初学者应按本书顺序学习，在掌握了前面知识的基础上再向后面的章节学习。对于已学过面向对象方法但尚理解不深的读者，可以在迅速扫描全书各章节的基础上找出自己学习的弱点，重点深入学习有关章节。

2. 注重上机编程实践是学好本课程的关键

面向对象编程的学习离不开上机编程实践。这是计算机学科是强调应用与实践的学科所决定的，要想成为优秀的面向对象编程人员，要想真正掌握Java编程的技术就一定是大量地上机编程、调试与运行。

3. 加强对学好本课程的兴趣

培养自己学好一门专业课的小窍门就是培养自己对学习这门课的兴趣，因为"兴趣是最好的老师！"。有了兴趣就会增强学习的积极性和克服困难的勇气与毅力。

试读结束，需要全本PDF请购买 www.ertongbook.com

目　录

第 1 章

面向对象方法与 Java 的概述
(Chapter 1. Introduction to Object-Oriented Paradigm and Java)

本章介绍面向对象方法的基本概念以及纯面向对象的编程语言 Java 的概述。在扼要地对比结构化开发方法与面向对象开发方法的基础上,着重阐明面向对象的系统分析、设计与编程,进而介绍了面向对象的基本概念,为后面的进一步学习奠定基础,最后介绍 Java 语言的特色。

1.1 面向对象方法概述(Introduction to Object - Oriented Paradigm)

面向对象方法已成为当今计算机软件领域的主流技术,并被许多著名专家、学者誉为"新世纪很有发展前途的**新**方法",也是研究高、新技术的**好**方法。

从软件开发的方法来看,将面向对象方法称为新方法是相对于传统的结构化的旧方法。它们之间差异见表1.1。

表1.1 两种软件开发方法的对照

传统的方法(Traditional Methods)		面向对象的新方法 (New Methods)
方法名称	结构化方法 (Structured Method)	面向对象方法 (O-O Paradigm)
软件开发 (Software Developing)	结构化系统分析(S.S.A) ↓ 结构化系统设计(S.S.D) ↓ 结构化编程 (S.P)	面向对象系统分析方法(O-OA) ↓ 面向对象系统设计方法(O-OD) ↓ 面向对象的编程方法(O-OP)
高级编程语言的发展历程 (Development of Program Language)	ALGOL 语言 (经历 50 多年) ↓ Fortran 语言 ↓ Pascal 语言 ↓ C 语言	Simula 语言(经历 30 多年) ↓ Smalltalk 语言 ↓ C++ 语言 ↓ Java 语言
优点	技术成熟,已全面标准化并推向实用	针对高技术的复杂难题而研制,实践证明能胜任此重任
缺点	对高技术领域复杂的非规则信息无法处理、显得力不从心。	面向对象的系统分析与设计尚未完全标准化,应用尚不够广泛。

关于结构化方法已在许多软件工程的书籍中有详细介绍,本书着重对面向对象方法在软件开发方面的应用作简要说明。

面向对象方法的软件开发主要有如下三个阶段:

1. 面向对象的系统分析(Object - Oriented System Analysis,以下简称 OOA)

作为系统分析的主要任务是经过对用户需求分析确定系统的整体功能,即**系统要做什么**(What to do)?

OOA 强调直接针对要开发的系统中客观存在的各种事物建立 OOA 模型:系统中有那些值得考虑的事物,OOA 模型就有那些对象,也就是说,客观世界与面向对象分析方法存在着一一对应的**自然而直接的映射**关系。为了更深入地描述客观事物,面向对象分析方法还用属性描述事物的静态(状态)特性;用方法描述事物的动态行为。图 1-1 表示汽车对象和描述这个

对象的状态和行为特性。

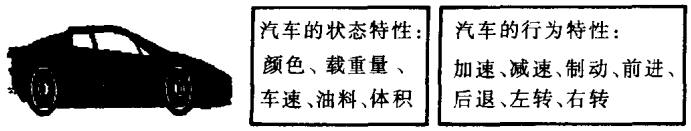

图 1-1　汽车对象的描述

事实上，客观世界存在着各种各样的汽车，从分析问题的角度，往往将这些具有相同特性的对象归结为类；并用分类、组装等结构描述类之间相互关系，如图 1-2 所示。实质上，分类反映出客观世界事物之间普遍存在的一般与特殊的关系，例如汽车具有较高的概括，手动车只是汽车的一种，而跑车则是手动车的一种，图 1-2(a)中层次越高抽象程度就越强，反之抽象程度就越低、越具体；组装则反映客观世界事物之间普遍存在的整体与局部的关系。从这两种最重要、最普遍的关系中，可以充分体会到 **OOA 强调对象与客观事物的一致性，尽量做到自然而直接地反映客观世界。这是 OOA 优于传统的结构化方法的显著特色之一。**

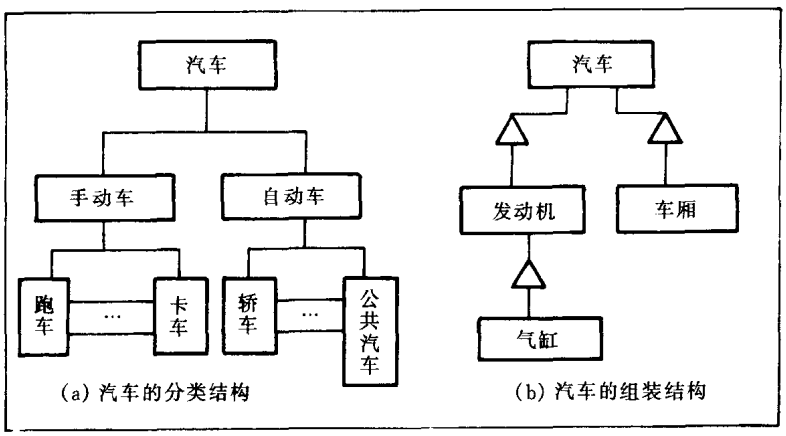

图 1-2　分类与组装的结构

2. 面向对象的系统设计(Object-Oriented System Design，以下简称 OOD)

系统设计的核心是确定**系统应怎样做**(How to do)?

OOD 应当在已建立的 OOA 模型的基础上，具体考虑系统实现的各种有关因素，例如采用什么图形用户界面，数据应如何管理，采取何种编程语言等。因此 OOD 应包括两步：一是**将 OOA 模型直接引入而不必转换，只作部分的修正与补充**；然后针对具体实现中的人机界面、数据存储、任务管理等因素运用 OO 方法。

OOA 与 OOD 采用一致的表示方法是面向对象方法优于结构化系统分析与设计方法的重要特色之二。

3. 面向对象的编程(Object-Oriented Programming，以下简称 OOP 或 OOI)

1) 面向对象程序设计与面向过程的程序设计的区别

类和对象是面向对象程序设计的核心，一个问题的求解过程在程序设计上可以看作是定义类和对象的过程。通过面向对象的系统分析和系统设计，抽象出类及类之间的联系，然后将

试读结束，需要全本PDF请购买 www.ertongbook.com

类实例化为若干对象。属于同一类或不同类的对象通过自己的方法改变状态，并通过相互发送消息实现对象之间的交互作用。最终通过对象的交互来实现对问题的求解。

而面向过程的程序设计则强调功能模块实现的算法及所用到的数据，它除了与结构化系统分析与系统设计之间往往存在着“濠沟”之外，在具体实现时由于函数与数据的分离而给软件的开发与维护都带来较大的不便。在图 1－3 中，展示面向过程的程序设计(如图(a))，面向对象的程序设计(如图(b))。

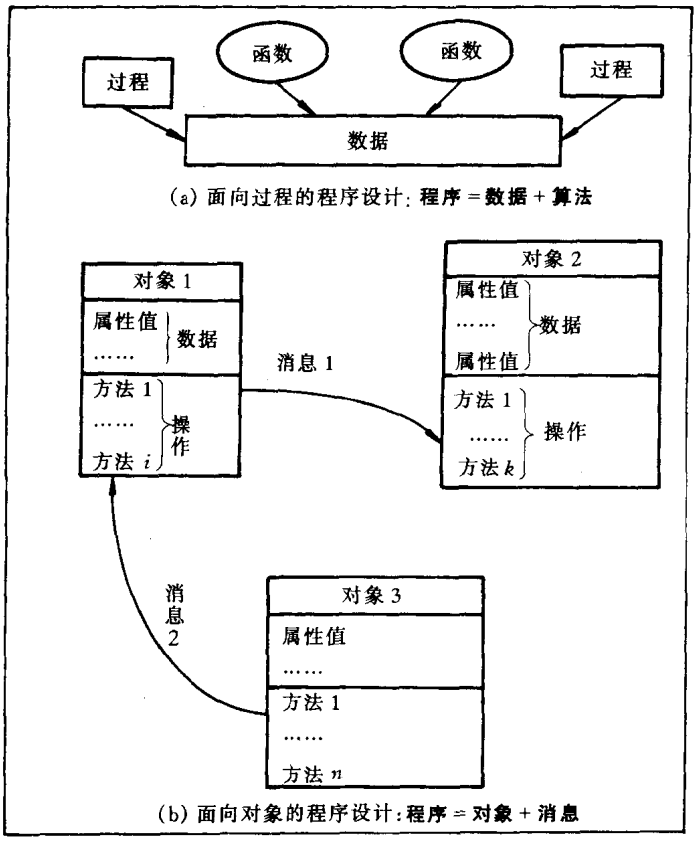

图 1－3　面向对象程序设计与面向过程程序设计的区别

2) 面向对象编程的具体内容

OOP 是面向对象方法的软件开发最终得以实现的重要阶段。其核心是用一种面向对象的编程语言来实现 OOD 模型。具体应有下面几方面：

(1) 选择一种合适的面向对象编程语言，如 C ++ 、Java 等等；

(2) 用选定的语言，编程实现详细设计时的公式、图表、说明和规则等；

(3) 将写好的类代码模块按类的相互关系集成；

(4) 利用开发人员或用户提供的测试样例分别检验各个模块和整个系统。在面向对象开发过程中，测试工作不是在最后才进行的，而是随着整个实现阶段工作的深入同步完成，也就是说，每个模块(类实现)完成之后可立即加入到整个系统框架中，模块的修改和细化也可在框架内完成。

面向对象的软件开发还包括面向对象的测试和维护。它们在软件生命周期中也占了很大

的分量,是非常复杂、烦琐的工作。但是,在面向对象的软件开发工作中,同于采取了对象这个灵活、可扩展的概念,维护与运行工作将大大简化。

1.2 面向对象的基本概念(Basic Concepts of Object-Oriented)

1.2.1 对象(Object)

对象是面向对象方法的**核心**,首先从下面几方面理解这一重要概念。

1. 对象具有很强的表达能力和描述功能

人们要进行研究的任何**事**和**物**统称为**对象**。

1) 对象可以是看得见、摸得着的有形的实体,如计算机、飞机、汽车等等;

2) 对象也可以表示人或组织所起的作用,如治病、教学、生产等;

3) 对象还可以表示事件,如各种事故、演出、战斗、开会等;

4) 对象更可以表示规则,如各种专家规则、约束条件等。

综上所述,**对象既能表示最简单的结构化的数据,又能表示非结构化的、抽象的、复杂的事件和规则**。

从系统分析和系统设计的角度来看,对象是构成系统的最基本的单位,也是描述客观事物的独立的实体。

2. 描述对象的两个要素:属性和方法

1) **属性** 用于描述对象静态特性(结构特性)的一个数据项。如描述一个人可用姓名、性别、身分证号等属性。

2) **方法**(也称服务) 用于描述对象动态特性(行为特性)的一个操作系列。如每个人都具有工作、学习等行为特性。

3. 对象体现了封装性能——实现了数据与操作的结合

对象就是一组属性和相关的方法的集合。这是面向对象方法与结构化方法的重大区别之一。

对象把它全部的属性和全部的方法结合在一起,形成一个独立的基本单位,如图 1-4 所示。也就是说,对象将数据和施加于数据上的操作打包成一个不可分割的最基本的模块,使得仅有该类的有限个方法才可以操纵、改变这些数据。这样,当数据出错时,只需检查有限的方法就可以。这是面向对象方法的封装性的一个重要表现。

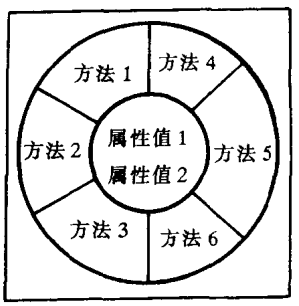

图 1-4 对象的封装性

4. 对象具有唯一标识(以下简称 OID)

当创建一个对象时,可由系统或用户授予该对象的 OID。对象一旦获取了 OID,就在它的整个生命周期中一直使用,作为唯一的标识。对象清除后,OID 可由系统回收,再分配给其它新对象。

5. 具体的对象必须参与一个或一个以上的类

在实际应用中,具体的对象应当参与类并作为该类的一个实例。在具体应用时,类实例与

对象是同义词。

6. 消息(Message)

对象之间是通过的发送与接收消息来建立联系的。

对象通过它对外提供的服务(方法)在系统中发挥它的作用。当系统中其它对象要求这个对象执行某个服务时,面向对象方法就把向这个对象发送的服务请求称为消息。通过消息进行对象之间的通信,如图 1-5 所示。这也是面向对象方法的一个原则:消息通信为对象之间联系提供了唯一合法的动态联系途径,从而确保对象的封装性得到很好的实现。对象之间既能够互相配合、又能够各自独立、互不干扰。

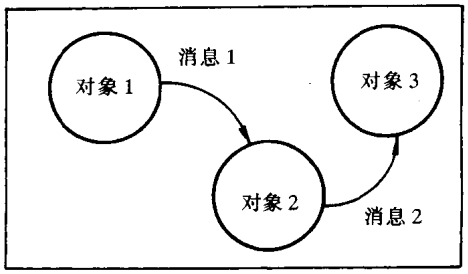

图 1-5 对象之间的消息发送与接收

7. 对象具有高度的抽象性(Abstraction)

在面向对象的方法中,对象可有最高的概括和抽象能力:它可作为面向对象编程语言的类的最高的父类——**根**。如 Java 或 C ++ 语言都把最高层类都定为**抽象的对象(Object)**。

1.2.2 类(Class)

类也是面向对象方法的一个非常重要的概念。把众多的事物归纳、划分成类是我们在认识世界时经常采用的方法。分类所依据的原则是抽象,也就是忽略事物的非本质特征,只注意那些与当前目标有关的本质特征,从而找出事物的共性并将具有共同性质的事物划分为同一个类。例如:人、树、山等等都属于高度抽象后的类。在面向对象方法中,也采用了类的概念,并给出下面的定义。

1. 类是具有相同语义特性的对象的集合

所谓相同的语义特性是指:

1) 同一类中的对象有相同的属性(也称为成员变量,它描述该类的静态特性或结构特性)

2) 同一类中的对象有相同的方法(也称服务,它描述该类的行为特性或动态特性)

3) 同一类中的对象遵守相同的语义规则。(共同的约束或规则)

2. 为什么要有"类"的概念?

有了类的概念之后,在具体应用时就会大大地简化编程并提高效率:成千上万个对象如果都具有相同的特征,只需**统一地定义一个类**,然后**用这个类作为创建对象的模板**来建立众多的对象(类实例)就可。因此,往往将类形象地比喻为一个制造它的实例(对象)的工厂,如图1-6所示。

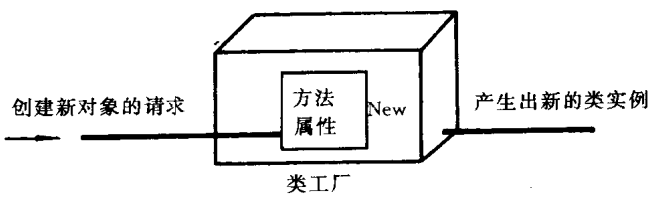

图 1-6 类是生成新的类实例的制造工厂

1.3 Java 概述(Introduction to Java)

1. Java 的故事(The Story of Java)

受 Sun MicroSystem 公司之托,1991 年由 Jame Gosling 等为消费性家电产品的控制自行开发了 Oak(橡树)的语言,当时尚未出名。但到 1994 年,随着国际互联网(Internet 是 Interconnection Network 缩写)的迅速发展,尤其是万维网 WWW (World Wide Web)的广泛应用,Sun 公司发现 Oak 的跨平台能力非常适合于网络应用。于是在 1995 年由 Sun 公司将原来的 Oak 命名为 Java。Java 语言一经推出,立即引起人们广泛的注意,很快就成为 Internet 上最受欢迎的编程语言,并迅速获得微软、IBM、DEC 等大公司的承认与使用。1996 年 Sun 成立 Javasoft 分公司来专门改进、推广与维护 Java 语言。

2. Java 语言的特色(The Characteristics of Java)

Java 语言在短短的几年内能得到如此广泛和迅猛的发展与应用,是因为它具有如下显著的特点:

1) 简单(Simple)

Java 语言与当前流行的 C ++ 等编程语言极为相似,但却去除了许多其它高级语言中不必要的特性。如 Java 不支持 C ++ 语言中有关指针的操作,不支持隐含强制类型转换、结构、联合数据类型、操作符重载、模板、头文件和多重继承等。

2) 面向对象(Object - Oriented)

面向对象是现实世界模型的自然延伸。现实世界中任何感兴趣的事物都可看作是对象。对象之间通过消息相互作用。Java 具有面向对象编程语言以对象为中心,以消息为驱动的共同特点。用公式表示:

面向对象编程语言用公式表示:程序 = 对象 + 消息。

结构化的过程式编程语言用公式表示:程序 = 算法 + 数据。

3) 分布式(Distribution)

Java 可在网络编程中使用的重要原因在于它的分布式特性。分布式包括数据分布和操作分布。数据分布是指数据可分散在网络不同的主机上,操作分布是指把一个处理分散在不同的主机上。此外,Java 还提供了一整套网络类库,开发人员可利用类库进行网络程序设计,更方便实现 Java 的分布式特性。

4) 与平台无关(Independent with Platform)

Java 是创建 Internet 应用程序的十分出色的工具,这是其它语言无法比拟的。除了它的分布式特性外,还因为 Java 具有与平台无关或平台独立的特点。所谓与平台无关是指用 Java 语言编写的应用程序可不用修改就可在不同的硬件平台上运行,做到"一次编写、到处运行"。

5) 安全性(Security)

由于 Java 语言能在网络/分布环境下运行,安全性的要求特别高,而 Java 在这方面也作了很多的考虑:既可避免计算机病毒的侵害,也能够防止对系统意外的损害。这是通过在编译和运行时采取了防护措施:编译程序保证源程序不违反安全规则;运行时系统校验字节代码等。也就是说,Java 的编译和运行环境为 Java 代码筑起两道坚实的防护屏障。

6）健壮性(Robust)

Java 具有很高的可靠性，它消除了 C ++ 语言许多不可靠的因素，此外，它还强调对错误的检查，包括早期对可能问题的检查，后期(运行时)对错误的动态检查以及防范可能产生的错误等到。

此外，Java 还具有**可移植性、高效率、编译型、强类型、易学易懂、多线程、动态可扩充性**等等特色，在后面的学习中再结合具体内容详细介绍。

1.4 本章小结(Summary)

本章首先介绍了面向对象的软件开发方法，主要包括面向对象的系统分析、面向对象的系统设计、面向对象的编程。进而阐明面向对象方法的最重要的基本概念，即对象与类，描述类的属性、方法，类之间的分类与组装关系，对象之间的消息发送等。最后，对纯面向对象语言 Java 的特点作了简要的说明。

复习题

1-1 简述结构化方法与面向对象方法在软件开发上的异同。

1-2 面向对象的系统开发方法应包括哪几个阶段？每个阶段的核心是什么？

1-3 如何描述对象的静态与动态性质？

1-4 对象之间是如何进行通信的？

1-5 类与对象有何关系？如何理解类是生产对象的工厂？

1-6 类与类之间又有何关系？什么是分类关系？什么是组装关系？

1-7 为什么说 Java 语言是纯面向对象的编程语言？

1-8 为什么说 Java 语言与平台无关？

1-9 Java 语言作为一个 Internet 语言，它最大的两个优点是什么？

第 2 章

Java 程序的两种类型
(Chapter 2. Two Types of Java Program)

本章介绍 Java 程序的两种类型:按程序结构组成和运行环境的不同,Java 程序可分为两大类,即 Java 的独立应用程序(Java Application)和 Java 的小应用程序(Java Applet)。Java 独立应用程序是完整的程序,需要独立的解释器来解释就可独立地运行;而 Java 的小应用程序则是嵌入到 HTML(超文本标记语言)或 XML(扩展的超文本标记语言),是编写网页很有用的、非独立的程序,通常由小应用程序的浏览器(APPLETVIEWER)或 Web 浏览器(如 IE、NETSCAPE 等)内含的 Java 解释器来解释运行。本章通过实例着重阐明它们的编辑、编译与运行。

试读结束,需要全本PDF请购买 www.ertongbook.com