现代计算机教育系列教材（英文版）——国外著名大学教授鼎力之作
丛书主编 金兰

Java Programming

Java 程序设计

Danny Poo
（新加坡）潘祥春 编著

清华大学出版社
Tsinghua University Press

Java Programming

Java 程序设计

Danny Poo

（新加坡）潘祥春　编著

清华大学出版社

Tsinghua University Press

内 容 简 介

本书使用 Java 语言讲解面向对象的程序开发方法。全书内容共 15 章，内容包括：Java 编程环境；Java 组成；表达式、语句和运算符；程序流程控制机制；数组；方法；类和对象；Java 应用程序界面（API）；输入和输出；文件控制；单类继承；封装；多态性；抽象；排序、搜索和递归。本书内容深入浅出，循序渐进，对编程过程的每一步均给出详细的指导，每个范例均提供完整的源代码，非常适合于没有编程知识的初学者学习 Java 语言编程方法。

本书可作为高等院校相关专业本科生 Java 语言程序设计课程教材，也可作为软件开发设计人员学习 Java 面向对象编程方法的自学参考书。

图书在版编目（CIP）数据

Java 程序设计 = Java Programming：英文/（新加坡）潘祥春（Danny Poo）编著. —北京：清华大学出版社，2010.1
（现代计算机教育系列教材（英文版）——国外著名大学教授鼎力之作/金兰主编）
ISBN 978-7-302-20722-1

Ⅰ. J…　Ⅱ. 潘…　Ⅲ. JAVA 语言－程序设计－教材－英文　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 146243 号

责任编辑：战晓雷　薛　阳
责任校对：时翠兰
责任印制：杨　艳

出版发行：清华大学出版社　　**地　　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编：**100084
社　总　机：010-62770175　　**邮　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：北京市清华园胶印厂
经　　销：全国新华书店
开　　本：185×230　　**印　张：**19.5　　**字　数：**409 千字
版　　次：2010 年 1 月第 1 版　　**印　次：**2010 年 1 月第 1 次印刷
印　　数：1～3000
定　　价：35.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　　产品编号：033347-01

About the Author

Dr. Danny Poo graduated with a BSc (Hons), MSc and PhD in Computer Science from the University of Manchester Institute of Science and Technology (UMIST), England. He is currently a tenured Associate Professor in the Department of Information Systems, National University of Singapore. He has taught courses in Systems Analysis and Design, Enterprise Systems Development, Object-Oriented Software Engineering, and Information Technology Project Management.

He is a Steering Committee Member of the Asia-Pacific Software Engineering Conference and founder and director of Cicada Cube Pte Ltd, an NUS spin-off company specializing in Enterprise-level Search and Retrieval Solutions. A well-known speaker in seminars, he has conducted numerous in-house training and consultancy for organizations both locally and regionally.

Dr. Poo is the author of 5 books: "Object-Oriented Programming and Java", 2nd edition, Springer-Verlag, 2007; **"Developing Systems Using J2EE"**, PrenticeHall, 2004, **"Learn To Program Java"**, 4th edition, Cengage Learning, 2009; **"Learn To Program Java User Interface"**, Thomson Learning, 2006; and **"Learn To Program Enterprise JavaBeans 3.0"**, 3rd edition, Cengage Learning, 2009.

Preface

Who Should Read This Book

This book teaches Java programming with a focus on object-oriented programming using the Java programming language. Designed for readers with no prior knowledge in computer programming, this book is unassuming and is most suitable for teaching Java at the undergraduate level. Complete source code is provided in every example and where applicable, screen-shots are used to as an added help for readers in their practical exercises.

Topic Covered in This Book

Topics covered include: the Java programming environment; the Java language components which include variables; data structures; data types and their declaration; expressions, statements, and operators; program flow control mechanisms; arrays; methods; inputs and outputs; class and objects; file handling; single class inheritance; encapsulation; polymorphism; abstract class, abstract method, inner class, method overriding, multiple class inheritance and interface. This book also covers the Java Application Programming Interface (API)—a rich Java class library.

How This Book Is Organized

This book is organized into fourteen chapters with each chapter building on what have been covered in the previous chapters. Each chapter is designed *workshops* and *exercises*. Workshops are practices with complete solutions provided while exercises are intended for readers to try out writing object-oriented programs in Java on their own.

Chapter 1: **The Java Programming Environment**

This chapter prepares the readers for Java programming with instructions on how to set

up the Java programming environment.

Chapter 2: **The Java Language Components**

This chapter defines the vocabulary of the Java language and explains how the words and symbols can be put together to form the basic program structure in Java.

Chapter 3: **Expressions and Statements**

This chapter explains the structure of a Java expression and how it can be used in conjunction with a semi-colon to form statements, the basic executable component of Java.

Chapter 4: **Program Flow Controls**

This chapter explains how to add program flow control mechanisms in a Java program. Statements can be put together and executed sequentially using the Sequence construct, selectively using the Selection construct, and repetitively using the Iteration construct.

Chapter 5: **Arrays**

This chapter explains the concept of an array as a data structure for storing and manipulating data of the same type as a collection. Details on how to manipulate a one-dimensional and two-dimensional array are included in this chapter.

Chapter 6: **Methods**

This chapter explains how a Java program can be more structured with the use of method—a collection of data and statements for performing a task.

Chapter 7: **Class and Objects**

Java is an object-oriented programming language. This chapter explains the concept of class and object and how to write Java programs using the object-oriented programming paradigm.

Chapter 8: **The Java Application Programming Interface (API)**

Java is a very small language with only 46 reserved keywords but it is extended with a library of Java Application Programming Interfaces (Java APIs). It is from the Java APIs that the power of Java is felt. This chapter introduces the Java API and its accompanying documentation and explains how the Java API can be used to solve complicated problems in Java.

Chapter 9: **Inputs and Outputs**

This chapter explains how data input and output can be carried out in Java.

Chapter 10: **File Handling**

This chapter discusses how to read and write data from text and binary files.

Chapter 11: **Inheritance**

Inheritance is an object-oriented mechanism for realizing software reuse. A subclass in single class inheritance can inherit properties from a superclass. When a subclass inherits

properties from more than one superclass, we have multiple class inheritance. This chapter discusses concepts related to single class inheritance.

Chapter 12: **Encapsulation**

Encapsulation is the bringing together of data fields and methods into an object definition with the effect of hiding the internal workings of the data fields and methods from the users of the object. Any direct access and updates to the object's constituents is not permissible and changes to the data fields can only be carried out indirectly via a set of publicly available methods. This chapter focuses on data field encapsulation and class encapsulation.

Chapter 13: **Polymorphism**

The ability of objects of different subclass definition to respond to the same message is *polymorphism*. Polymorphism is only possible with dynamic binding—the capability of determining which method implementation to use for a method at runtime. This chapter explains the concept of polymorphism and its peripheral object-oreinted programming concepts.

Chapter 14: **Interface**

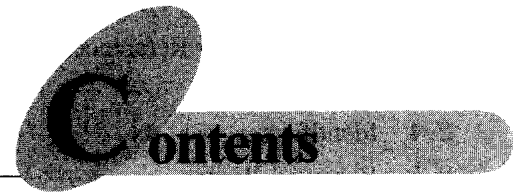
Sometimes it is necessary to derive a subclass from several classes but the Java extends keyword does not allow for more than one parent class. With interfaces, multiple class inheritance is possible. This chapter explains and shows how to use the Java interface construct to realize multiple class inheritance.

Chapter 15: **Sorting, Searching and Recursion**

This chapter discusses three topics commonly found in Java programming. They include: Sorting, Searching, and Recursion.

Enjoy!

Danny



CHAPTER *1* The Java Programming Environment

CHAPTER *2* The Java Language Components

CHAPTER 3 Expressions, Statements and Operators

CHAPTER 4 Program Flow Controls

CHAPTER 5 Arrays

CHAPTER 6 Methods

CHAPTER 7 Class and Objects

CHAPTER *11* Inheritance

CHAPTER *12* Encapsulation

CHAPTER *13* Polymorphism

CHAPTER *14* Interface

CHAPTER *15* Sorting, Searching, and Recursion

CHAPTER 1

The Java Programming Environment

Java is a programming language①. Much like English is a spoken language that we use to communicate with one another, Java is used for communicating our intentions as a programmer to a computer.

A computer is a device that can be configured to understand the instructions written in Java. The set of instructions is commonly known as a *program*. So, when we speak of Java programming, we are involved in writing programs in the Java language that a computer can understand and execute according to what the set of instructions dictates.

The Java programming language② is textual and is expressed using English. It has a set of restricted vocabulary that we can use to express our intentions in a program.

Can a computer understand English and how does it know how to execute the program? A computer technically does not understand English as it is. But, it does understand code specially crafted for machines. Such code is typically represented in 0s and 1s which we human being finds it extremely difficult to comprehend.

How do we then bridge the gap between a high-level English-written program and a set of low-level binary-code-based machine instructions? The solution lies in having an interpreter. An interpreter is a software program that takes in a high-level program and translates it into a low-level binary-code-based program that a computer can understand. In the case of Java, the output of the interpretation process is a set of *byte code*.

Byte code, though low-level, is still not at the lowest level. Machine code, or *object*

---

① There are other programming languages available e. g. C, C++, C#, FORTRAN, COBOL, MUMPS, SNOBOL, Pascal, Python, etc.

② Java bears some similarities to the C programming language, as Java was created from C.