<<Java软件结构与数据结构>>

图书基本信息

书名：<<Java软件结构与数据结构>>

13位ISBN编号：9787302207306

10位ISBN编号：7302207305

出版时间：2009-9

出版时间：清华大学出版社

作者：（美）刘易斯，（美）切斯　著

页数：526

<<Java软件结构与数据结构>>

前言

This book is designed to serve as a text for a course on data structures and algo-rithms. This course is typically referred to as the CS2 course because it is oftentaken as the second course in a computing curriculum. We have designed thisbook to embrace the tenets of Computing Curricula 2001 （CC2001）. Pedagogically, this book follows the style and approach of the leading CS1book Java Software Solutions: Foundations of Program Design, by John Lewisand William Loftus. Our book uses many of the highly regarded features of thatbook, such as the Key Concept boxes and complete code examples. Together,these two books support a solid and consistent approach to either a two-courseor three-course introductory sequence for computing students. That said, thisbook does not assume that students have used Java Sofware Solutions in a previ-ous course. Material that might be presented in either course （such as recursion or sorting）is presented in this book as well. We also include strong reference material pro-viding an overview of object-oriented concepts and how they are realized in Java. We understand the crucial role that the data structures and algorithms courseplays in a curriculum and we think this book serves the needs of that course well.The Third EditionWe have made some key modifications in this third edition to enhance its peda-gogy. The most important-change is a fundamental reorganization of material thatis designed to create a cleaner flow of topics. Instead of having an early, largechapter to review obiect-oriented concepts, weve included that material as anappendix for reference. Then we review concepts as needed and appropriate in thecontext of the implementation strategies discussed throughout the book and citethe appropriate reference material. This not only links the topics in a timely fash-ion but also demonstrates the usefulness of particular language constructs. Weve expanded the discussion of Analysis of Algorithms, and given it its ownchapter. The discussion, however, stays at an appropriately moderate level. Ourstrategy is to motivate the concepts involved in the analysis of algorithms, layinga solid foundation, rather than get embroiled in too much formality.

<<Java软件结构与数据结构>>

内容概要

本书是著名作者John Lewis与William Loftus作为其一流的CS1教材“ Java Software Solutions：Foundations of Program Design ”的姊妹篇。
尽管本书的英文名为“ Java Software Structures: Designing and Using Data Structures ”，但正如作者在前言中所说的那样，本书其实是一本可作为“ 数据结构与算法 ”课程的教材。
根据使用了前两版的教师和学生的反馈，作者在第3版中进行了重大修改，以适应教学的需要。
本书关注的是数据结构和算法背后的核心设计问题。
在展现每种集合时，本书都是先探讨该集合的一般概念，接着再讨论该集合在问题求解中的用法，最后讨论了各种候选实现方案。
因此，本书是“ 数据结构与算法 ”Java语言描述课程的理想教材。

<<Java软件结构与数据结构>>

书籍目录

章节摘录

Abstraction is another important software engineering concept. In large soft-ware systems, it is virtually impossible for any one person to grasp all of the de-tails of the system at once. Instead, the system is divided into abstract subsystemssuch that the purpose of and the interactions among those subsystems can bespecified. Subsystems may then be assigned to different developers or groups ofdevelopers that will develop the subsystem to meet its specification. An object is the perfect mechanism for creating a collection because, if it is de-signed correctly, the internal workings of an object are encapsulated from the restof the system. In almost all cases, the instance variables defined in a class shouldbe declared with private visibility. Therefore, only the methods of that class canaccess and modify them. The only interaction a user has with an object should bethrough its public methods, which represent the services that the object provides. As we progress through our exploration of collections, we will always stressthe idea of separating the interface from the implementation. Therefore, for everycollection that we examine, we should consider the following: How does the collection operate, conceptually ?

How do we formally define the interface to the collection ?

What kinds of problems does the collection help us solve ?

In which various ways might we implement the collection ?

What are the benefits and costs of each implementation ?

Before we continue, lets carefully define some other terms related to the explo-ration of collections. A data type is a group of values and the operations definedon those values. The primitive data types defined in Java are the primary exam-pies. For example, the integer data type defines a set of numeric values and theoperations （addition, subtraction, etc.） that can be used on them. An abstract data type （ADT） is a data type whose values and operations arenot inherently defined within a programming language. It is abstract only in thatthe details of its implementation must be defined and should be hidden from theuser. A collection, therefore, is an abstract data type.

<<Java软件结构与数据结构>>

版权说明