

<<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：9787302205951

10位ISBN编号：7302205957

出版时间：2009-8

出版时间：清华大学出版社

作者：（美）路易斯（Lewis, J.），（美）蔡斯（Chase, J.）著，金名 等译

页数：393

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

本书可用作数据结构与算法课程的教材。

数据结构与算法课程常常称为CS2课程，因为它通常作为计算机专业的第二门课程。

本书涵盖了计算机专业课程设置2001（Computing Curricula 2001，CC2001）的宗旨。

从教学的角度看，本书遵循了一流的CS1教材“Java Software Solutions：Foundations of Program Design”（作者John Lewis与William Loftus）一书的风格和方法。

本书使用了在该书中受到很高评价的一些特色，比如关键概念框和完整代码示例。

作为计算机专业学生的两门或者三门导论课程，这两本书提供了一个坚实而系统的教学方案。

也就是说，本书不要求学生在以前的课程中已经使用过Java Software Solutions一书。

在CS1和CS2课程中出现的内容（比如递归或排序），本书也同样进行了阐述。

本书还给出了一些有用的参考材料，介绍了面向对象的概念以及如何在Java中实现。

我们知道，数据结构与算法在计算机专业课程中起着关键的作用，我们认为，本书能较好地满足该课程的要求。

本书第3版 在第3版中，我们对本书进行了一些重要的修改，以适应教学需要。

最重要的修改是，对本书的基本结构进行了重新设计，以使得这些内容之间的脉络更加清晰。

本版不是用一大章来复习面向对象的概念，而是把这些内容作为一个附录以供参考。

然后在全书需要的地方对这些概念进行复习，并介绍其实现策略的上下文，引用适当的参考材料。

这不仅可以在适时地把这些内容链接起来，而且还演示了特定语言结构的使用。

本版对算法分析的讨论进行了扩展，并把算法分析作为了单独的一章。

但是，这些讨论也只是停留在恰当的难度上。

本书的策略是，只对算法分析的基本概念进行介绍，而不是进行很深入的讨论。

另一个重要的结构变化是，对集合的介绍使用了一个栈作为基本示例。

在本书的上一版本中，我们以一种抽象的方法来介绍集合，即把它与核心数据结构分开，使用诸如背包或组集之类的示例。

这种方法凸显这样一个事实：从概念上来说，栈是很简单的。

使用它作为第一个示例，可以增进把集合作为一个整体的理解。

本书的上一版本用了几个章节来介绍几个大型案例研究，这些案例研究使用集合来求解实际问题。

很多教师觉得这些很有用，但他们同时觉得这会打断对核心内容的介绍。

因此，在这一版本中，我们把这些案例研究章节从书中删除了，并把它们作为辅助材料放在本书配套的网站上。

我们鼓励教师去下载这些材料，并以他们认为适合的方式使用这些材料。

最后，我们对本版进行了重新审阅，并改进了一些讨论。

我们扩展了对图的讨论，把“图”与“散列”两章的顺序进行了调换，使得脉络更清晰。

本版还添加了一章来专门讨论Set与Map集合。

我们认为，以上这些修改，都是建立在使用以前版本教学的基础上，为教师提供更多的机会和更好的灵活性来介绍本书的内容。

本书的写作风格 这种类型的图书在整体写作方法上的差别相当大。

本书的写作方法是建立在一些我们强烈推荐的重要原则之上的。

首先，我们以一种连贯叙述的方式介绍在本书中将要考察的各种集合。

其次，我们强调完美软件设计技巧的重要性。

第三，我们对本书结构加以组织以支持和强化本书的重要目标：即数据结构与算法的学习。

我们将更深入地考察这些原则。

连贯叙述 当考察某种类型的集合时，将按照以下顺序仔细解决每一问题。

1. 概念：我们会从概念上讨论集合，并构建它所提供的服务（其接口）。

2. 用途：我们将介绍一些示例，阐述在求解问题时，集合的特定属性是如何派上用场的（不用管

## &lt;&lt;Java软件结构与数据结构&gt;&gt;

集合是如何被实现的)。

3. 实现：我们将考察集合的各种实现。

4. 分析：我们对各种实现进行了比较和区分。

在恰当的时候，我们将讨论Java Collections API。

如果有必要讨论API中某个特定集合类型，那么我们将讨论该集合及其实现。

因此，我们欢迎API，但并不完全局限于它。

并且，我们会毫不犹豫地指出其缺陷。

分析步骤是一个较高的层次。

在第2章介绍的大O记法的概念将全书中应用，但这更多地是直觉上而不是数学上的分析。

完美的程序设计 全书始终将完美的软件工程实践置于一个高度优先的地位。

我们对集合实现的设计以及对使用它们的程序的设计都将遵循一致和恰当的标准。

将集合的接口与其底层实现分隔开，是非常重要的。

某一集合提供的服务，通常在Java接口中有着正式定义。

为了强化该集合为一个抽象的思想，只要合适，我们都将接口名称用作该集合的类型指定。

除了实践坚实的设计原则外，我们还在全书的论述中对其进行了强调。

我们努力尝试既通过示例又通过不断强化的方式来进行教学。

清晰的组织结构 我们对本书内容进行了精心组织，以将偏离主题带来的干扰降到最低，并对本书的总体目标进行强化。

本书的组织使得本书既可以用作数据结构与算法的教学用书，又可以作为一本颇有价值的参考用书。

本书的内容可以划分为多个部分：第一部分包括前两章，介绍了集合的概念和算法分析。

第二部分包括随后的4章，介绍了数据结构与算法以及影响它们的基本问题，并介绍线性集合（栈、队列和列表）。

第三部分介绍了递归、排序和查找的概念。

第四部分介绍了线性集合（树、堆、散列和图）。

除树之外的每一个集合类型都自成一章。

有关树的内容分布于一系列的章节中，用于考察其各个方面和各种作用。

章节划分 第1章（概述）讨论了软件质量涉及的各个方面，并对软件开发问题进行了一个全面概述。

本章用于在着手数据结构和算法设计的细节之前，建立一套正确的开发思想。

第2章（算法分析）介绍了确定算法效率的基础知识，并阐述了一个重要的标准，使得开发人员可以以正确的方式将一种算法与另一种算法进行比较。

本章的重点是理解重要的概念，而不是陷入数学或公式的漩涡。

第3章（集合）建立了集合的概念，强调了将接口和实现区分开来的必要性。

本章还从概念上介绍了栈，然后阐述了基于数组的栈的实现。

第4章（链式结构）讨论了使用引用来创建链式数据结构。

本章考察了有关链表管理的基本问题，然后使用基本链式数据结构（在第3章中已介绍），定义栈的另一种实现方法。

第5章（队列）考察了先进先出队列的概念和实现。

本章通过一个高效使用队列的示例来讨论基数排序。

本章所介绍的实现包括基本链表以及定长和环形数组。

第6章（列表）论述了3种类型的列表：有序表、无序表和索引表。

通过讨论这3种类型列表所共有的及各自独有的操作来对它们进行比较和区分。

在各种类型的列表设计中，我们将恰当地使用继承，且通过基于数组的和链式的表示两种方式实现这些列表。

第7章（递归）概要介绍了递归的概念，以及递归解决方案为什么是优美的。

本章还探讨了递归的实现细节，并讨论了递归算法分析的基本思想。

第8章（排序与查找）讨论了线性查找和二分查找算法，以及若干排序的算法（如选择排序、插

## &lt;&lt;Java软件结构与数据结构&gt;&gt;

入排序、冒泡排序、快速排序以及归并排序)。

本章着重讨论与查找和排序相关的编程问题，比如将Comparable接口用作对象比较的基础。以特定数据结构为基础的查找与排序（如堆排序）将在本书后面的适当章节讨论。

第9章（树）对树进行了概述，并构建了关键的术语和概念。

本章讨论了各种实现方式，并通过一个二叉树来表示和评估某一算数表达式。

第10章（二叉查找树）利用第9章构建的基本概念，定义了一个经典的二叉查找树。

本章先考察了一个二叉查找树的链式实现，然后讨论了树结点的平衡如何对其性能起到关键作用。这就引出了AVL和二叉查找树的红/黑实现的介绍。

第11章（优先队列与堆）探讨了堆的概念、使用和实现，尤其是与优先队列的关系。

我们通过一个堆排序来举例说明其用途。

本章还介绍了链式实现和基于数组的实现。

第12章（多路查找树）是前面章节的自然延伸。

本章讨论了2-3树、2-4树以及广义B树的概念，还讨论了各种实现。

第13章（图）探讨了无向图和有向图的概念，并构建了一些重要的术语。

本章考察了若干常用图的算法，并讨论了各种实现，包括邻接矩阵。

第14章（散列）包括散列的概念及其相关问题，比如散列函数及冲突。

本章讨论了散列的各种Java Collections API。

第15章（Set与Map集合）介绍了这两者类型的集合及其对Java Collections API的重要性。

附录A（UML）概述了统一建模语言。

UML是表示面向对象系统的事实标准表示法。

附录B（面向对象设计）为任何需要回顾面向对象的基本概念及其在Java中如何实现等内容的读者提供参考。

该章包含的概念包括抽象、类、封装、继承、多态性以及许多相关的Java语言结构，比如接口。

教 辅 资 料 本书的所有读者都可以在[www.aw.com/cssupport](http://www.aw.com/cssupport)网址上获得以下教辅资料。

\* 本书中出现的所有程序的源代码。

\* 完整的案例研究程序，包括Black Jack Game、Calculator、Family Tree Program和Web Crawler程序。

在Pearson Education的教师资源中心（网址是<http://www.pearsonhighered.com/irc>）里的以下教辅资料仅限于满足条件的教师。

请访问该网址，与你当地的Pearson Education销售代表联系，或者发送电子邮件至[computing@pearson.com](mailto:computing@pearson.com)，咨询有关资料获取的信息。

\* 本书中的部分练习题和编程项目的答案。

\* 测试题库，含有测验用的问题。

\* 讲解本书内容的PowerPoint幻灯片。

## <<Java软件结构与数据结构>>

### 内容概要

本书是著名作者John Lewis与Joseph Chase作为其一流的CSI教材“Java Software Solutions : Foundations of Program Design”的姊妹篇。

尽管《Java软件结构与数据结构(第3版)》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，《Java软件结构与数据结构(第3版)》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。

最重要的修改包括这样几个方面：（1）对《Java软件结构与数据结构(第3版)》的基本结构进行了重新设计，以使得这些内容之间的脉络更加清晰；（2）第3版把对面向对象概念的复习作为一个附录以供参考；（3）上一版给出了几个完整的Java程序设计案例和源代码，在第3版中进行了删除，并把这几个程序案例源代码放在了网上供读者下载。

译者认为，这不仅压缩了不少篇幅，而且使得《Java软件结构与数据结构(第3版)》更像是一本数据结构与算法的教材，而不是Java程序设计的教材；（4）第3版扩展了对图的讨论，把“图”与“散列”两章的顺序进行了调换，使得脉络更清晰。

本版还添加了一章来专门讨论Set与Map集合。

总之，这些修改都是建立在使用以前版本教学的基础上，为教师提供更多的机会和更好的灵活性来使用《Java软件结构与数据结构(第3版)》。

## &lt;&lt;Java软件结构与数据结构&gt;&gt;

## 书籍目录

第1章 概述 1.1 软件质量 1.1.1 正确性 1.1.2 可靠性 1.1.3 健壮性 1.1.4 可用性 1.1.5 可维护性 1.1.6 可重用性 1.1.7 可移植性 1.1.8 运行效率 1.1.9 质量问题 1.2 数据结构 1.2.1 一个物理示例 1.2.2 以集装箱作为对象 关键概念小结 自测题 练习题 自测题答案

第2章 算法分析 2.1 算法效率分析 2.2 增长函数与大O记法 2.3 增长函数的比较 2.4 时间复杂度分析 2.4.1 循环运行的复杂度分析 2.4.2 嵌套循环的复杂度分析 2.4.3 方法调用的复杂度分析 关键概念小结 自测题 练习题 自测题答案 参考文献

第3章 集合 3.1 概述 3.1.1 抽象数据类型 3.1.2 Java集合API 3.2 栈集合 3.3 主要的面向对象概念 3.3.1 继承 3.3.2 类层次 3.3.3 Object类 3.3.4 多态性 3.3.5 引用与类层次 3.3.6 泛型 3.4 栈ADT 接口 3.5 使用栈计算后缀表达式 3.6 异常 3.6.1 异常消息 3.6.2 try语句 3.6.3 异常传播 3.7 用数组实现栈 管理容量 3.8 ArrayStack类 3.8.1 构造函数 3.8.2 push操作 3.8.3 pop操作 3.8.4 peek操作 3.8.5 其他操作 关键概念小结 自测题 练习题 程序设计项目 自测题答案

第4章 链式结构 4.1 链接作为引用 4.2 管理链表 4.2.1 访问元素 4.2.2 插入结点 4.2.3 删除结点 4.2.4 哑结点 4.3 无链接的元素 双向链表 4.4 用链表实现栈 4.4.1 LinkedStack类 4.4.2 push操作 4.4.3 pop操作 4.4.4 其他操作 4.5 使用栈来穿越迷宫 4.6 java.util.Stack类实现栈 4.6.1 独有的操作 4.6.2 继承与实现 关键概念小结 自测题 练习题 程序设计项目 自测题答案

第5章 队列 第6章 列表 第7章 递归 第8章 排序与查找 第9章 树 第10章 二叉查找树 第11章 优先队列与堆 第12章 多路查找树 第13章 图 第14章 散列 第15章 Set与Map集合

附录A UML 附录B 面向对象设计

## 章节摘录

但是，在这种情景下，如果要创建一个为Map集合的有序列表，就要创建一个表示每个员工的名字的类，以及一个指向第二个类的引用。

第二个类含有员工数据的其他所有信息。

我们然后用有序列表操作把第一个类加载到我们的列表中，而第二个类的对象存在于内存中。

这里第一个类称为关键字，而第二个类称为数据。

用这种方式，当我们操作列表的元素时，只需处理关键字、名称和引用，与需要操作和员工有关的所有数据相比，此时需要的内存要少得多。

这里还有一个优点，那就是当相同的员工数据被多个Map引用时，无须使用多个副本。

因此，如果一个应用程序用栈来表示员工信息，而另一个应用程序则用有序列表来表示，那么我们就可以把关键字加载到栈中，把匹配的关键字加载到有序列表，此时实际的数据只有一个实例。

与处理别名（即多个指向相同对象的引用）的情况一样，因为对象只有一个实例，因此，当需要通过一个引用来修改某个对象时，由于该对象同时被其他引用所引用，因此必须非常小心。

在第10章中，我们讨论了TreeSet和TreeMap的Java集合API实现。

在第14章中，我们讨论了使用散列的Java集合API实现。

注意，这些Set实现违反了Set集合的代数表示。

该Set集合的元素是无序的。

除此之外，还有一些其他类实现了java.util.Set或java.util.Map接口。

这就是为什么Java语言的设计者在很多情况下使用Set和Map集合，而不是我们本书中所介绍的传统数据结构。

简单地说，Java语言不是教学语言，而是用于快速而高效地解决工业问题的语言。

因此，很多Java解决方案很适用但不那么优美。

当然，对立的问题也同样重要。

既然Java语言的设计者选择使用Set和Map，而不是本书所介绍的传统数据结构，那么我们为什么花这么多的时间来介绍传统数据结构呢？

对这个问题有很多很好的答案，但这里有一个答案可以超越其他所有答案。

尽管Java是一种很好的语言，但它并不是你的职业生涯中将遇到的唯一语言。

在我们短暂的职业生涯中，语言来来去去很多次，但数据结构的基本原则仍保持不变。

因此，我们学习这些基本原则很重要，这样我们就不会重复犯过去的错误。

关键概念小结 · Set是一种非线性集合。

在该集合中的元素之间基本上没有任何组织结构。

## <<Java软件结构与数据结构>>

### 编辑推荐

《Java软件结构与数据结构(第3版)》的写作方法是建立在一些我们强烈推荐的重要原则之上的。首先，我们以一种连贯叙述的方式介绍在《Java软件结构与数据结构(第3版)》中将要考察的各种集合。

其次，我们强调完美软件设计技巧的重要性。

第三，我们对《Java软件结构与数据结构(第3版)》结构加以组织以支持和强化《Java软件结构与数据结构(第3版)》的重要目标：即数据结构与算法的学习。

我们将更深入地考察这些原则。



<<Java软件结构与数据结构>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>