丰富的图示与范例
诠释数据结构

图解数据结构
——使用Java

丰富的图示解释
复杂的数据结构概念

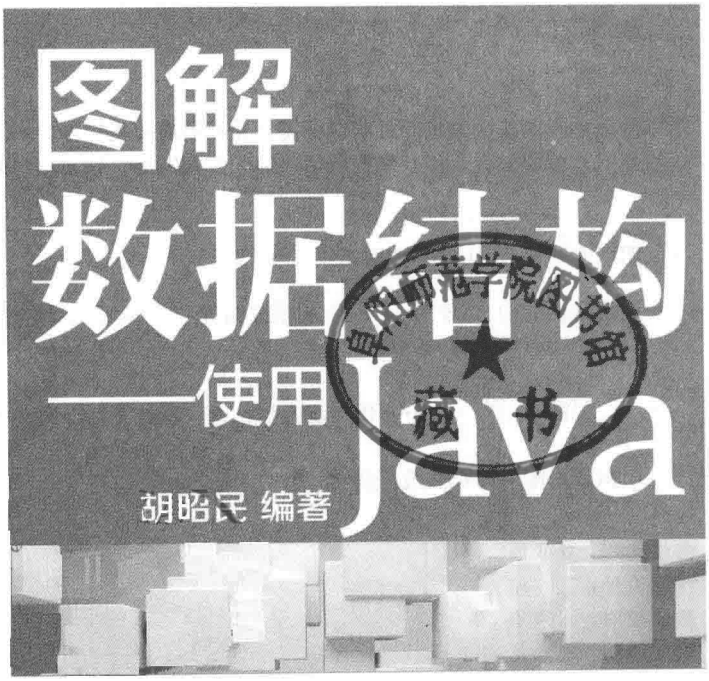
以范例程序说明
数据结构的内涵

以Java语言实现
数据结构中的重要理论

提供网络资源下载

胡昭民 编著

清华大学出版社



清华大学出版社
北京

本书版权登记号：图字 01-2015-2540

内 容 简 介

这是一本以 Java 程序语言实战来解说数据结构概念的教材。全书内容浅显易懂，利用大量且丰富的图示与范例，详解复杂的抽象理论，从最基本的数据结构概念开始说明，再以 Java 工具加以诠释阵列结构、堆栈、链表、队列、排序、查找等重要的概念，引领读者抓住重点轻松进入数据结构的学习领域。

本书每章重要理论均有范例实现，书中收录了精华的演算法及程序的执行过程，在线阅读或下载附有完整的范例程序源代码，读者可以依照学习进度做练习。除此之外，还有配合各章教学内容的练习题目，以便读者测试自己的学习效果。

本书内容架构完整，逻辑清楚，采用丰富的图例来阐述基本概念及应用，有效提升可读性。以 Java 程序语言实现数据结构中的重要理论，以范例程序说明数据结构的内涵。采用“Eclipse”Java ID 工具，整合编译、执行、测试及除错功能。强调边做边学，结合下载文件，给予最完整的支援。

图书在版编目（CIP）数据

图解数据结构 ：使用 Java / 胡昭民编著. -- 北京 ：清华大学出版社，2015
ISBN 978-7-302-40299-2

Ⅰ. ①图… Ⅱ. ①胡… Ⅲ. ①数据结构－图解②JAVA 语言－程序设计 Ⅳ. ①TP311.12-64②TP312

中国版本图书馆 CIP 数据核字(2015)第 113813 号

责任编辑：夏非彼
封面设计：王　翔
责任校对：闫秀华
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**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：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张**：23.5　　**字　　数**：601 千字
版　　次：2015 年 8 月第 1 版　　**印　　次**：2015 年 8 月第1 次印刷
印　　数：1～3500
定　　价：49.00 元

产品编号：063194-01

序

数据结构一直是计算机科学领域非常重要的基础课程，它除了是各大专院校信息工程、计算机工程、软件工程、应用数学以及计算机科学等信息相关专业的必修科目外，近年来包括机电、电子或一些商务管理系也列入选修课程。同时，一些信息相关专业的转学考试、研究所考试等，也将数据结构列入必考科目。由此可见，不论从考试的角度，或者研究信息科学的角度，数据结构确实是有志于从事信息工作的专业人员不得不重视的一门基础课程。

要学好数据结构的关键点在于能否找到一本最容易阅读，并将数据结构中各种重要理论、算法等做最详实的解释及举例的图书。市面上以 Java 来实现数据结构理论的书籍比较缺乏，本书是一本如何将数据结构概念用 Java 程序语言实现的重要著作。为了方便学习，书中程序代码都是完整的，可以避免片断学习程序的困扰。另外，本书下载文件中包括提供范例完整的程序代码，方便练习及教学之用。

本书的主要特色在于将比较复杂的理论以图文并茂的形式进行说明，并以最简单的表达方式，将这些数据结构理论加以解释。为了避免在教学及阅读上的不顺畅感，书中的算法尽量不以伪码进行说明，而以 Java 程序语言来展现。同时，书中提到的重要理论，尽量搭配完整的范例程序，便于读者了解以 Java 语言实现这些算法的注意事项。

另外，为了检验读者各章的学习成果，在书中安排了大量的习题，这些题目包含重要考试的考题，以便读者更加灵活应用各种知识。

在附录中提供 Java 的开发环境的简介，本书编辑环境是采用 Eclipse 软件，它是一套 Open Source 的 Java IDE 工具，Eclipse 整合编译、运行、测试及除错功能。

一本好的理论书籍除了内容的完整专业外，更需要清晰易懂的架构安排及表达方式。在仔细阅读本书之后，相信读者会体会笔者的用心，也希望用户能对这门基础学科有更深入、更完整的认识。

本书配套源代码下载地址（注意数字与字母大小写）：http://pan.baidu.com/s/1bnqCuHD，若下载有问题，请电子邮件联系 booksaga@126.com，邮件标题为“求代码，图解数据结构：使用 Java”。

作者敬笔

目　录

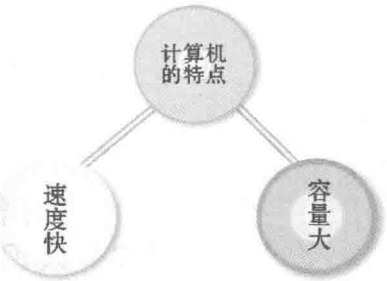
第1章
数据结构导论

计算机（Computer），或者有人称为计算器（Calculator），是一种具备数据处理与计算的电子化设备。它可以接收人类所设计的指令或程序设计语言，经过运算处理后，输出所期待的结果。对于一个有志于从事计算机专业领域的人员来说，数据结构（Data Structure）是一门和计算机硬件与软件都相关的学科，其中包含算法（Algorithm）、数据存储架构、排序、查找、程序设计的概念与哈希函数。

数据结构的研究重点是计算机的程序设计领域，如何将计算机中相关数据的组合，以某种方式组织而成，然后在这样的定义下，就可以探讨各种有意义的操作与关系，以便提高程序的执行效率。

1.1 数据结构简介

大家可以将数据结构看成是在数据处理过程中一种分析、组织数据的方法与逻辑，它考虑到了数据间的特性与相互关系。在现代社会中，计算机与信息是息息相关的，因为计算机具有处理速度快与存储容量大两大特点，在数据处理的角色上更为举足轻重，如下图所示。



数据结构无疑就是数据进入计算机内处理的一套完整逻辑，就像程序设计师必须选择一种数据结构来进行数据的新增、修改、删除、存储等操作。因此当我们要求计算机为我们解决问题时，必须以计算机所能接受的模式来确认问题，而安排适当的算法去处理数据，就是数据结构要讨论的重点。

1.1.1 数据与信息

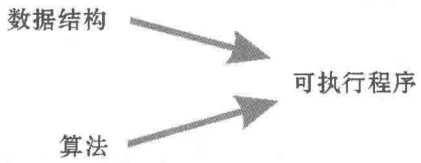
谈到数据结构，首先必须了解何谓数据（Data）与信息（Information）。从字面上来看，所谓数据（Data），指的就是一种未经处理的原始文字（Word）、数字（Number）、符号（Symbol）或图形（Graph）等，它所表达出来的只是一种没有评估价值的基本元素或项目。例如姓名或我们常看到的课表、通讯录等都可泛称是一种“数据”（Data）。

当数据经过处理（Process），例如以特定的方式系统地整理、归纳甚至进行分析后，就成为“信息”（Information）。而这样处理的过程就称为“数据处理”（Data Processing）。从严谨的角度来形容“数据处理”，就是用人力或机器设备，对数据进行系统的整理，如记录、排序、合并、整合、计算、统计等，以使原始的数据符合需求，而成为有用的信息。

不过各位可能会有疑问：“那么数据和信息的角色是否绝对一成不变？”。这倒也不一定，同一份文件可能在某种状况下为数据，而在另一种状况下则为信息。例如美伊战争的某场战役死伤人数报告，对你我这些平民百姓而言，当然只是一份不痛不痒的“数据”，不过对于英美联军指挥官而言，这份情报可就是弥足珍贵的“信息”。

1.1.2 算法

数据结构与算法是程序设计实践中最基本的内涵。程序能否快速而有效地完成预定的任务，取决于是否选对了数据结构，而程序是否能清楚而正确地把问题解决，则取决于算法。所以各位可以这么认为：“数据结构加上算法等于可执行的程序。”



不过在韦氏辞典中算法却定义为：“在有限步骤内解决数学问题的程序。”如果运用在计算机领域中，我们也可以把算法定义成：“为了解决某项工作或某个问题，所需要有限数目的机械性或重复性指令与计算步骤。”其实日常生活中有许多工作都可以利用算法来描述，例如员工的工作报告、宠物的饲养过程、学生的课程表等。

1.1.3 算法的条件

当认识了算法的定义后，我们还要说明算法所必须符合的 5 个条件，如下表所示。

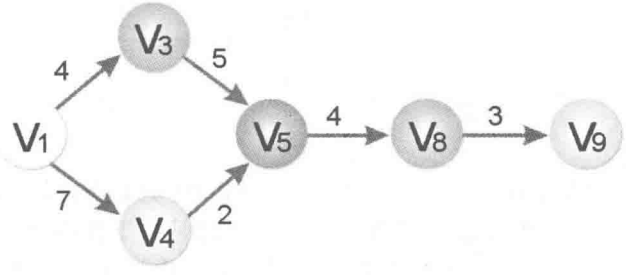
算法特性	内容与说明
输入（Input）	0 个或多个输入数据，这些输入必须有清楚的描述或定义
输出（Output）	至少会有一个输出结果，不可以没有输出结果
明确性（Definiteness）	每一个指令或步骤必须是简洁明确的
有限性（Finiteness）	在有限步骤后一定会结束，不会产生无限循环
有效性（Effectiveness）	步骤清楚且可行，能让用户用纸笔计算而求出答案

接着各位要来思考：该用什么方法来表达算法最为适当。其实算法的主要目的在于为人们提供阅读了解所执行的工作流程与步骤，只要能清楚表现算法的 5 项特性即可。常用的算法如下。

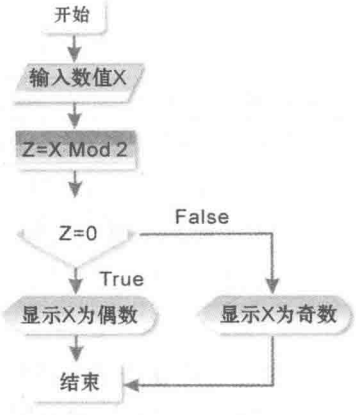
- 一般文字叙述：中文、英文、数字等。文字叙述法的特色在于使用文字来说明演算步骤。
- 伪语言（Pseudo-Language）：接近高级程序设计语言的写法，也是一种不能直接放进计算机中执行的语言。一般都需要一种特定的预处理器（Preprocessor），或者用手写转换成真正的计算机语言，经常使用的有 SPARKS、Pascal-LIKE 等语言。以下是用 SPARKS 写成的链表反转的算法：

```
Procedure Invert(x)
        P←x;Q←Nil;
        WHILE P≠NIL do
    r←q;q←p;
    p←LINK(p);
    LINK(q)←r;
        END
    x←q;
END
```

- 表格或图形：如数组、树形图、矩阵图等。



- 流程图（Flow Diagram）：是一种通用的图型符号表示法。例如请您输入一个数值，并判断是奇数还是偶数。



- 程序设计语言：目前算法也能够直接以可读性高的高级语言来表示，例如 Visual Basic、C、C++、Java。

至此，还要特别为各位说明一点，程序（Program）和算法是有区别的。程序中可以允许无限循环的存在，如一般操作系统中的“作业调度器”（Job Scheduler），在启动后，除非关机或产生例外状况，不然会一直处于执行等待循环。但算法却必须是有限的，这是两者之间的最大不同。

1.2 认识程序设计

在数据结构中，所探讨的目标就是将算法朝有效率、可读性高的程序设计方向努力。简单地说，数据结构与算法必须通过程序（Program）的转换，才能真正由计算机系统来执行。而程序设计的目的就是通过程序的编写与执行来达到用户的需求。或许各位读者认为程序设计的主要目的只是要“执行”出正确的结果，而忽略了执行效率或者日后的维护成本，其实这是不清楚程序设计真正意义的表现。

1.2.1 程序开发流程

至于程序设计时必须利用何种程序设计语言，通常可根据主客观环境的需要确定，并无特别规定。一般评判程序设计语言好坏的四项原则如下。

- 可读性（Readability）高：阅读与理解都相当容易。
- 平均成本低：成本考虑不局限于编码的成本，还包括执行、编译、维护、学习、调试与日后更新等成本。
- 可靠度高：所编写出来的程序代码稳定性高，不容易产生边际错误（Side Effect）。
- 可编写性高：对于针对需求所编写的程序相对容易。

对于程序设计领域的学习方向而言，无疑就是以有效率、可读性高的程序设计成果为目标。一个程序的产生过程，则可分为以下 5 个设计步骤。

步骤 1：需求认识（Requirements）：了解程序所要解决的问题是什么，有哪些输入及输出等。

步骤 2：设计规划（Design and Plan）：根据需求选择适合的数据结构，并以任何的表示方式写一个算法以解决问题。

步骤 3：分析讨论（Analysis and Discussion）：思考其他可能适合的算法及数据结构，最后再选出最适当的目标。

步骤 4：编写程序（Coding）：把分析的结论写成初步的程序代码。

步骤 5：测试检验（Verification）：最后必须确认程序的输出是否符合需求，这个步骤需要执行程序并进行许多的相关测试。

程序设计的 5 大步骤

1.2.2 数据类型简介

当您进行程序设计时，除了算法的设计外，首先必须挑选一种程序设计语言。要了解程序设计语言的重点，除了语法及语义外，最重要的就是数据类型（Data Type）。所谓数据类型就是程序设计语言的变量（variable）所能表示的数据种类。又因为存储层次上的不同可分为 3 种。

基本数据类型（atomic data type）

基本数据类型或称为物理数据类型（physical data type），也就是一个基本的数据实体，例如一般程序设计语言中的整数、实数、字符等。基本上，每种语言都拥有略微不同的基本数据类型，例如 C 语言的基本数据类型为整数（int）、字符（char）、单精度浮点数（float）与双精度浮点数（double）。

结构型数据类型（structure data type）

结构型数据类型或称为虚拟数据类型（virtual data type），比物理数据类型更高级，是指一个数据结构包含其他的数据类型，例如字符串（string）、集合（set）、数组（array）。

抽象数据类型（Abstract Data Type，ADT）

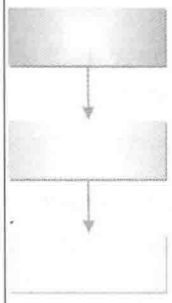
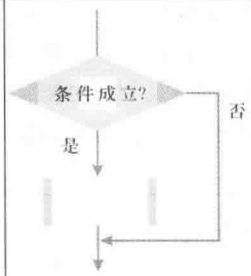
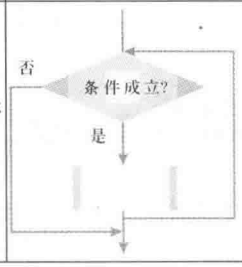
抽象数据类型比结构型数据类型更高级，ADT 是指定义一些结构型数据类型所具备的数学运算关系，用户无须考虑 ADT 的制作细节，只要知道如何使用即可。也就是说，只针对数据的运算，而非数据本身的性质，例如某个数据对象可以插入一个列表，或在列表中增删，而不需关心这个对象的类型是字符串、整数、实数还是逻辑值。通常出现在面向对象程序设计语言（OOP）中的堆栈（stack）或队列（queue）就是一种很典型的 ADT 模式。

1.2.3 结构化程序设计

在传统程序设计的方法中，主要以“由下而上”与“由上而下”方法为主。所谓“由下而上”是指程序设计师将整个程序需求中最容易的部分先编写，再逐步扩大来完成整个程序。而“由上而下”则是将整个程序需求从上而下、由大到小逐步分解成较小的单元，或称为“模块”（Module），这样使得程序设计师可针对各模块分别开发，不但可减轻设计者负担、可读性较高，也便于日后维护。而结构化程序设计的核心精神，就是“由上而下设计”与“模块化设计”。例如在 Pascal 语言中，这些模块称为“过程”（Procedure），而 Java 中称为“函数”（Function）。

每一个模块会完成特定的功能，主程序则组合每个模块后，完成最后要求的功能。不过一旦主程序要求的功能变动时，则可能许多模块内的数据与算法都需要同步变动，而这也是面向过程的设计无法有效使用程序代码的主要原因。

通常“结构化程序设计”具有以下三种控制流程，对于一个结构化程序，不管其结构如何复杂，都可利用以下的基本控制流程来加以表达。

流程结构名称	概念示意图
顺序结构：逐步编写程序语句	
选择结构：根据某些条件做逻辑判断	
重复结构：根据某些条件决定是否重复执行某些程序语句	

1.2.4 面向对象程序设计

“面向对象程序设计”（Object-Oriented Programming，OOP）是近年来相当流行的一种新兴程序设计理念。它主要让我们在设计程序时，能以一种更生活化、可读性更高的设计概念来进行，并且所开发出来的程序也较容易扩充、修改及维护。例如在现实生活中充满了形形色色的物体，每个物体都可视为一种对象。我们可以通过对象的外部行为（behavior）及内部状态（state）模式，来进行详细的描述。行为代表此对象对外所显示出来的运行方法，状态则代表对象内部各种特征的目前状况，如下图所示。