

高等学校理工科
计算机类课程

学习辅导丛书

数据结构与算法

学习辅导及习题详解

张乃孝 编 著

► 教学重点和难点

► 习题解答 ► 上机辅导

学习的帮手 考研的参谋



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校理工科计算机类课程学习辅导丛书

数据结构与算法 学习辅导及习题详解

张乃孝 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书综合“数据结构与算法”的知识梳理、习题解答及上机辅导等于一身；精心挑选了覆盖教学大纲的五百多道题目，并且提供所有题目的参考答案；对于较难的算法和上机题，给出了详细的分析和说明；对于学习的重点和难点、易犯的错误、题目的难易和重要性，以及国内教材的差异等都给出了必要的说明。

本书可给使用各种教材讲授和学习“数据结构与算法”（或者“数据结构”）的师生参考，是系统复习该课程和准备应考计算机专业研究生的理想辅导书，也是与《算法与数据结构——C语言描述》配套的教学辅导书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目(CIP)数据

数据结构与算法学习辅导及习题详解/张乃孝编著. —北京：电子工业出版社，2004.10

ISBN 7-121-00419-4

I. 数… II. 张… III. ①数据结构—高等学校—教学参考资料②算法分析—高等学校—教学参考资料 IV. TP311.12

中国版本图书馆 CIP 数据核字（2004）第 100614 号

责任编辑：李岩

印 刷：北京大中印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销：各地新华书店

开 本：787×980 1/16 印张：31.75 字数：708 千字 插页：1

印 次：2004 年 10 月第 1 次印刷

印 数：5 000 册 定价：45.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：(010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

1978 年,北大计算机系成立。参照国外的教学计划,把数据结构列为本科教学的基础课,分配给我们教研组负责。在十分困难的条件下,我组织大家查找资料,起草教学大纲,经过一年多的团结奋斗,编写出北京大学第一本(在国内也是第一批)数据结构讲义。大约是 1979 年深秋,用这个讲义的初稿,在浙江省计算中心,我第一次系统讲述了数据结构。

在其后 25 年的时间里,我坚持在教学的第一线讲授数据结构,并不断地学习和研究数据结构的教學方法,更新教材的内容。1998 年,北大对全校主干基础课设置主持人,我连续 3 届被校长聘为该课程主持人。六年来,组织理科各院、系主讲教师制定教学大纲,编写实用教材,交流考题,整理题库,努力为提高全校算法与数据结构课程的教学质量贡献力量。

我参加编写的数据结构教材有四本。

第一本《数据结构》^[1]在 1979 年起草的数据结构讲义的基础上,经校内外广泛使用,多次修改,最终到 1987 年才得以正式出版,前后共花费了八年时间。这本教材是综合了[5]和[6]两本书的优点编写而成的,内容比较丰富;同时把各种数据结构按逻辑结构、存储结构和运算三要素逐层展开,使整个教材结构比较清晰。这在同类教材中还是属于首次。书中所有算法采用我们自己定义的伪代码表示。

第二本《数据结构基础》^[2]是应北京市自学考试委员会的要求,为计算机专业的考试编写的教材。作为数据结构课程的考试委员,我承担制定本课程教学大纲、编写教材和每年两次的出题/阅卷任务。这本教材吸取了[7]中抽象数据类型的观点,为使教材适合自学,压缩了内容、突出了重点;书中所有算法采用 Pascal 语言描述。

第三本《数据结构——C++与面向对象的途径》^[3]于面向对象编程思想得到广泛应用的 20 世纪 90 年代问世。从研究的兴趣出发,我们根据一个数据结构的分层模型,组织了全书的结构。该书的最大特点就是数据结构与面向对象思想的紧密结合。

第四本《算法与数据结构——C 语言描述》^[4]是为北大同名主干课编写的教材。根据综合大学的特点,针对北大信息、数理、生化和医学等专业学生的不同要求,在编写时既要突出重点,还要灵活实用。全书以数据结构为主线、以算法为辅线,强调数据结构与算法在问题求解中的地位 and 作用,重点讲解数据结构的核心内容和基本要求,也区分了对于算法的不同要求。

另外,在我发表的论文中,大约有十分之一直接和数据结构与算法有联系。在 20 世纪 80 年代关于“五代机”的研究^[27, 28, 29]和 90 年代“面向语言方法学”的研究^[30, 31, 32]中都离不开树的表示和算法。应该说,算法与数据结构的基本原理对于我科研思想的影响

是十分深远的。

编写本书的动机

数据结构与算法在计算机教育中的核心地位与重要作用是公认的，学习数据结构与算法的困难也是广大师生所共知的。多年来，许多老师和同学都曾向我表示：十分需要数据结构与算法的教学指导、实习指导、学习辅导和习题解答之类的书。

目前国内外出版了大量数据结构的教材。它们之间的主要区别，有的是结构不同，有的是深浅不同，有的是语言不同；但是在基本概念、数据结构和算法的设计与实现上基本一致。这说明本课程的主要内容和方法已经基本取得共识，为编写公用的学习辅导和习题解答提供了良好的基础。

根据著者多年的教学经验，学习数据结构的难点在做题。学生在做题中经常出现的错误，通常也是教材中没有重点解释的问题。提高学生的知识水平光靠听课和看书是不够的，在学生做题的过程中，可能会发现许多教学中忽视的问题。如果挑选一批对于理解数据结构的概念和思想有益的题目让学生练习，可以有助于学生学得活、理解得更深；在习题解答中标明学生经常出现的错误，并且适当加以解释，供学生在自己独立思考后参考，更能达到事半功倍的效果。

算法是人类知识的结晶。在数据结构的习题中，最难的是算法题。算法题的答案不是惟一的，所以无法给出一个标准答案。编写习题解答的重点和难点是算法。设计算法的关键在于设计的思路，不同的思路产生出不同的算法。面对一个算法题，学生最需要的是如何开始分析这个问题和逐步解决这个问题。所以对每个复杂的算法题，不应该仅仅给出一个程序，而需要首先给出问题的分析、算法的思路，再给出程序和算法分析。对存在多个解法的题目，在给出一个算法的同时，还可以指出答案的缺点和改进之处。这样，学生可以从中体会到算法的真谛，从而真正有利于提高学生的算法设计与分析能力。

应用数据结构与算法的知识解决实际问题，并上机实现，是学习数据结构与算法不可缺少的环节。作为上机的指导用书，不但要选择适当的问题，提出实习的具体要求和规定上机报告的内容和形式；同时更需要给学生以帮助。实习题与算法题有类似之处，但通常更加复杂。我搜集了常见教材上的各种应用问题，并给出求解的样例，其目的是有助于组织好这一教学环节。

另外在目前众多的相关教材中，虽然主要内容和方法已经基本一致，但总有一些概念（例如树的高度，B树的定义，满二叉树的定义等）不完全一致。作为基础课的教材，这样必然容易造成误解，同时也影响交流。我常常看到，许多学生为了准备考研，总是重新购买一本所报考学校的数据结构教材，从头看起，毫无重点，浪费了许多宝贵时间。能够在一本书中把这些区别解释清楚，也是本书的一个目标。

经过几年的努力,我深深体会到,编写这种辅导书要比编写一本普通教材困难得多。但愿我的上述理想,在本书中能够得以体现。

本书的组织

本书继承了《算法与数据结构——C语言描述》^[4]的主要编写思想,并采用了与之类似的结构;但是根据广大考研读者的需要,补充了原书缺省的内容,使之成为一本比较完整的包括教、学、习题和上机的辅导资料。

全书共分9章,每章又至少分为5节。各节的内容安排如下。

第1节是“主要内容与方法”。它参考[4]的内容,给出本章的主要教学内容和教学方法;指出本章的重点、难点和注意事项。各章的第1节均以电子课件的形式给出。它是后面习题解答的基础,可以作为学生复习的大纲,也可以作为老师讲课的胶片。

从第2节开始是习题和参考答案。从国内外重点教材[1~11]的习题和各种考试题目中,精心挑选了五百多道题目,这些题目覆盖了全部教学大纲的要求,可以分为以下五类。

- “简单题”(各章的第2节)。主要对本章的基本知识和概念通过简单再现(是非、选择、填空和简答等)的形式帮助学生进行复习和检查。
- “表示题”(各章的第3节)。主要通过图、表或者文字方式来表述答题。例如,对具体数据的逻辑结构、存储结构加以说明,或者是对具体算法用实例模拟运行过程,等等。在有些章中,表示题比较丰富,所以又按照知识点分为小节。表示题是考试的重点,通常的考卷中这部分占一半以上。
- “算法题”(各章的第4节)。主要是锻炼和提高学生的算法设计和分析能力。这通常是学习的难点,也是决定考试能否取得好成绩的关键。本书对于这些题目的解答给出了详细的分析。
- “应用与上机题”(各章的第5节)。包括经常出现在各种教材中的应用问题。本书的这一节希望在上机实习的教学中起到一定的指导作用。在第3章的解答中,还提供一个上机报告,希望对学生有参考价值。
- 每章的最后,可能有用“**”标记的第6节(甚至第7节)是“扩充题”。补充了与本章相关,而在教材[4]中没有讨论的内容。例如与文件、存储管理、稀疏矩阵、广义表、优先队列、跳跃表和集合等相关的习题和解答,专门提供给准备考研的读者参考。

除了简单题,其他各类题目中的重点题都用“☆”标记,掌握这些题目,有助于学生理解数据结构与算法的本质,希望读者认真对待;用“▲”标记的是难题,供考研的读者练习;没有做标记的是选做题。

读者对象与使用

本书可以提供给使用各种教材讲授“数据结构”相关课程的老师参考,也可以提供给参加相关课程各种层次学习和考试的学生参考。特别对于以下两类读者实用价值更强。

(1) 报考计算机及相关专业研究生的考生。无论基础如何,选择本书复习应考,将会达到事半功倍的效果。建议考研的学生,认真理解每章第1节的内容,把已经掌握的知识加以总结;完成各章所有简单题、带“☆”的表示题、算法题和补充题;然后在时间许可的情况下,尽可能选做带“▲”的难题和其他题。

(2) 以《算法与数据结构——C语言描述》^[4]为相关课程教材的教师和学生。本书包含原教材的精华和全部的习题解答。建议大学本科(或者专科)的学生,使用每章的第1节作为复习大纲;选择第2节的简单题和第3、4节带“☆”的表示题和算法题巩固学到的基本知识;有条件并希望深造的同学可以参考第一条。大专(或者非理工科本科)学生,使用本书复习时,可以跳过书中所有用“*”标记的章节。

另外,本书收编的大都是基本的习题,为了更充分地复习应考,读者可以自己对其进行灵活的变化和补充,方法如下。

(1) 各章的“主要内容与方法”里,选择重要的概念,采用简单题的任何一种形式(是非、选择、填空和简答)就可以得到大量对应的简单考题。

(2) 把所有表示题的实际对象,更换成新的数据或者算法,就可以得到不同的表示型考题。例如,用不同的散列函数、不同的处理碰撞方法,处理不同数据就得到不同的散列表考题。

(3) 书中每个算法习题,变化为“读算法理解功能”、“算法关键部位填空”、“算法复杂性分析”、“算法思想陈述”或“算法的改进”等形式。这也是实际考卷中经常使用的方法。

本书中各章的第1节“主要内容与方法”、所有算法题和上机题解答的程序代码、部分数据结构试卷样本和部分与本课程相关的(著者发表的)论文均可在电子工业出版社华信教育资源网(www.hxedu.com.cn)上获得。

感谢

首先要感谢的是我所有的学生,没有他们的认真听课和积极且广泛的发问,没有他们提交的丰富多彩的作业、上机报告和考卷,就没有编写本书的源泉和动力;特别要感谢的,是那些在我讲课期间,反反复复帮助我选题,做题,改题和上机编程、调试的本科生、研究生和助教们,是他们的原作给本书提供了坚实的基础,本书的每个题目至少都经过他们三人以上的处理;我真诚地感谢所有支持我编写此书老师、同事、朋友和亲人,感谢他们数十年来对我的关心、理解、爱护和帮助;最后要提前感

谢本书的所有读者，你们的宝贵意见将是使本书不断提高和完善的主要保证。

本书虽是著者多年教学经验的积累，但整理出版与读者共享是不久前的决定。是电子工业出版社和李岩博士的大力支持，才使得本书能在较短的时间里与读者见面。时间仓促、工作量大，错误和疏忽之处在所难免，衷心希望广大读者批评指正。

张乃孝

znx@pku.edu.cn

2004年8月于北大

目 录

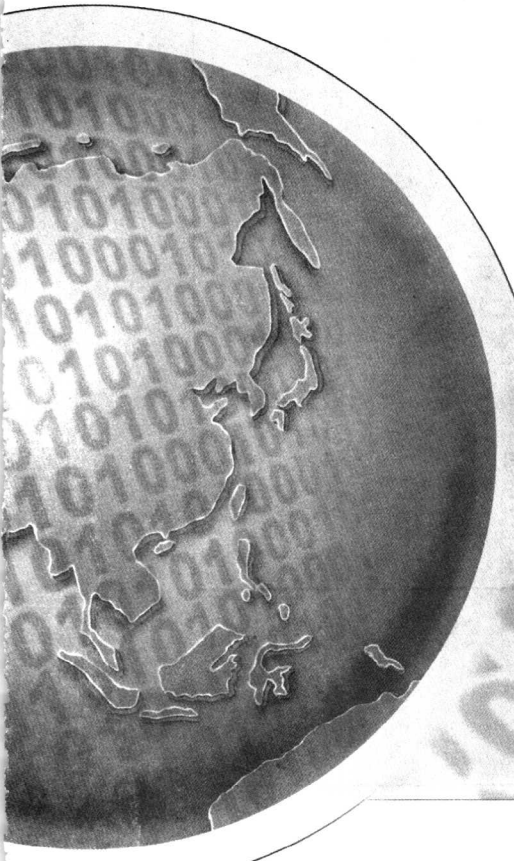
第 1 章 绪论	1
1.1 主要内容与方法	1
1.1.1 问题求解	2
1.1.2 抽象数据类型	2
1.1.3 数据结构	2
1.1.4 算法	3
1.1.5 讨论	4
1.2 简单题	6
1.2.1 是非题	6
1.2.2 选择题	6
1.2.3 填空题	7
1.2.4 简答题	7
1.3 数据结构题	7
1.4 算法题	13
1.5 问题求解题	16
1.6 文件**	23
第 2 章 线性表	26
2.1 主要内容与方法	26
2.1.1 线性表的概念	27
2.1.2 顺序表	27
2.1.3 单链表	29
2.1.4 讨论	32
2.2 简单题	33
2.2.1 选择题	33
2.2.2 填空题	33
2.2.3 简答题	34
2.3 表示题	34
2.4 算法题	35
2.5 应用与上机题	56
2.6 动态存储管理**	65
第 3 章 串*	70
3.1 主要内容与方法	70

3.1.1	字符串及其运算	71
3.1.2	字符串的表示	71
3.1.3	模式匹配	72
3.1.4	讨论	75
3.2	简单题	76
3.3	表示题	76
3.4	算法题	77
3.5	应用与上机题	87
第 4 章	栈与队列	95
4.1	主要内容与方法	95
4.1.1	栈	96
4.1.2	栈的实现	96
4.1.3	栈与递归	98
4.1.4	队列	99
4.1.5	队列的实现	99
4.1.6	讨论	101
4.2	简单题	102
4.2.1	是非题	102
4.2.2	选择题	102
4.2.3	填空题	102
4.2.4	简答题	103
4.3	表示题	103
4.4	算法题	106
4.5	应用与上机题	119
第 5 章	树与二叉树	150
5.1	主要内容与方法	150
5.1.1	树的概念	151
5.1.2	树的实现	152
5.1.3	二叉树的概念	155
5.1.4	二叉树与树(林)的对应	157
5.1.5	二叉树的实现	158
5.1.6	最优二叉树	159
5.1.7	讨论	161
5.2	简单题	162
5.2.1	是非题	162

5.2.2	选择题	162
5.2.3	填空题	165
5.2.4	简答题	166
5.3	表示题	167
5.3.1	逻辑	167
5.3.2	周游	173
5.3.3	存储	178
5.3.4	转换	181
5.3.5	哈夫曼树	183
5.3.6	表达式树*	187
5.4	算法题	189
5.5	应用与上机题	216
第 6 章	字典与检索	226
6.1	主要内容与方法	226
6.1.1	基本概念	227
6.1.2	顺序表示	227
6.1.3	散列表示	228
6.1.4	二叉排序树表示	230
6.1.5	AVL 树表示	233
6.1.6	索引表示	235
6.1.7	讨论	239
6.2	简单题	240
6.2.1	是非题	240
6.2.2	选择题	240
6.2.3	填空题	242
6.2.4	简答题	242
6.3	表示题	244
6.3.1	顺序表示	244
6.3.2	散列表示	244
6.3.3	二叉排序树表示	258
6.3.4	AVL 树表示*	263
6.3.5	索引表示*	270
6.4	算法题	281
6.5	应用与上机题	294
6.6	集合**	303

第7章 排序	326
7.1 主要内容与方法	326
7.1.1 基本概念	327
7.1.2 插入排序	327
7.1.3 选择排序	329
7.1.4 交换排序	330
7.1.5 分配排序	332
7.1.6 归并排序	333
7.1.7 讨论	334
7.2 简单题	336
7.2.1 选择题	336
7.2.2 填空题	339
7.2.3 简答题	339
7.3 表示题	342
7.3.1 排序实例	342
7.3.2 堆*	348
7.4 算法题	354
7.5 应用与上机题	372
7.6 优先队列**	377
第8章 图	387
8.1 主要内容与方法	387
8.1.1 基本概念	388
8.1.2 图的基本运算	388
8.1.3 图的周游	388
8.1.4 图的存储	390
8.1.5 最小生成树	392
8.1.6 最短路径	394
8.1.7 AOV 与 AOE 网	396
8.1.8 讨论	397
8.2 简单题	399
8.2.1 是非题	399
8.2.2 选择题	399
8.2.3 填空题	400
8.2.4 简答题	400
8.3 表示题	401

8.3.1	逻辑	401
8.3.2	存储	403
8.3.3	周游	410
8.3.4	最小生成树	414
8.3.5	最短路径	417
8.3.6	AOV 与 AOE 网*	423
8.4	算法题	425
8.5	应用与上机题	437
8.6	广义表**	449
8.7	稀疏矩阵**	453
第 9 章	算法分析与设计*	458
9.1	主要内容与方法	458
9.1.1	算法分析技术	459
9.1.2	算法设计技术	460
9.1.3	讨论	463
9.2	简单题	465
9.2.1	选择题	465
9.2.2	填空题	465
9.3	算法分析题	466
9.4	算法设计题	472
9.5	应用与上机题	482
参考文献		492



第 1 章

绪 论

学习要点

- 掌握数据结构和算法的基本概念和方法，理解它们在问题求解中的作用。
- 在用计算机解题过程中，算法和数据结构是相辅相成、缺一不可的两个方面。数据结构是算法处理的对象，也是实现算法的基础。
- 一个具体问题的数据在计算机中往往可以采用多种不同的表示方式，一个问题的解决又常常有多种可用的算法。
- 选择什么样的算法和数据结构是程序设计中最重要课题。
- 文件的基本概念作为本章的扩充在最后一节给出。

1.1 主要内容与方法

1.1.1 问题求解

用计算机解决实际问题的过程，实质上就是在计算机中建立一个解决问题的模型。在这个模型中，使用计算机能够理解的数据表示实际问题的对象，用计算机能够执行的算法分析和处理这些数据，实现对于实际问题的分析和处理过程。

问题求解的四个阶段

1. 分析阶段
2. 设计阶段 (算法 + 数据结构)
3. 编码阶段
4. 测试和维护阶段

1.1.3 数据结构

数据结构：用计算机表示（存储）的具有一定逻辑关系和行为特征的一组数据的集合。

这个集合中的每一个元素是这个数据结构的**一个实例（对象）**。

组成数据结构的（基本）数据元素称为这个数据结构的**结点**。

根据面向对象的观点，**数据结构可以看成抽象数据类型的具体实现**。

根据这个观点，ADT的数学模型对应数据的**逻辑结构**，模型的具体表示称为数据的**存储结构**，ADT中操作的具体实现就是结构的**行为特征**。

1.1.2 抽象数据类型

类型（Type）：一组值（或者对象）的集合。

数据类型（Data Type）：计算机（通常是指程序设计语言）中可以直接表示和处理的类型。

抽象数据类型（Abstract Data Type，简称为ADT）：定义了一组操作的一个抽象（数学）模型。

集合与集合的并、交、差运算就可定义为一个抽象数据类型。

数据结构分类的三个要素

(1) 逻辑结构

表示数据元素之间的逻辑关系 $B = (K, R)$ 。
例如，集合、线性表、树形结构、图。

(2) 物理（存储）结构

数据在计算机存储器中的表示方式。
例如，顺序表示、链接表示、散列表示、索引表示。

(3) 结构的行为特征

可作用于数据上的运算和这些运算间的关系。
例如，检索、插入、删除等操作和后进先出、先进先出等关系。

1.1.4 算法

算法 (Algorithm) 是由有穷规则构成的, 为解决某一特定类型问题而确定的一个运算序列。

一个算法可以有若干输入, 这些输入是在算法开始时给出的初始值或条件。

一个算法通常又有若干个输出, 它们是同输入有某种关系的计算结果。

一个算法就是将其输入转换到输出的良好定义的 (还应该具有有穷性、确定性和可行性) 计算过程。

5

主要算法设计思想

- 贪心法
- 分治法
- 回溯法
- 动态规划法
- 分限界法

设计算法通常采用“自顶向下, 逐步求精”的方法。

7

算法具有以下性质

- **有穷性:** 一个算法必须在执行有穷步之后结束。在某些领域也需要研究不终止的算法, 但这不属于数据结构教材讨论的范畴。
- **确定性:** 算法的每一步, 必须有确切的定义。也就是说, 对于每一步需要执行的动作必须严格地清楚地给出规定。
- **可行性:** 算法是可行的意味着原则上都是能够由机器或人完成的。整个算法好像一个解决问题的“工作序列”, 其中的每一步都是我们力所能及的一个动作。

6

算法的复杂性

算法的空间代价 (或称空间复杂性): 当被解决的问题的规模由 1 增至 n 时, 解决该问题的算法所需占用的空间也以某种单位由 $f(1)$ 增至 $f(n)$, 这时称该算法的空间代价是 $f(n)$ 。

算法的时间代价 (或称时间复杂性): 当问题规模以某种单位由 1 增至 n 时, 对应算法所耗费的时间也以某种单位由 $g(1)$ 增至 $g(n)$, 这时称算法的时间代价是 $g(n)$ 。

8

大“O”表示法

某个算法的时间代价(或者空间代价)为 $O(f(n))$, 则表示存在正的常数 c 和 n_0 , 当问题的规模 $n > n_0$ 后, 该算法的时间(或空间)代价 $T(n) \leq c \cdot f(n)$ 。

这种说法意味着: 当 n 充分大时, 该算法的复杂性不大于 $f(n)$ 的一个常数倍。

1.1.5 讨论

主要概念

- 抽象数据类型。
- 数据的逻辑结构, 存储结构, 行为特征。
- 算法的设计与分析, 空间代价和时间代价。
- 线性表, 字符串, 堆栈, 队列, 树, 二叉树, 字典, 图。
- 贪心法、分治法、回溯法、动态规划法、分支限界法。
- 自顶向下, 逐步求精。

算法代价的计算

(1) 加法规则

$$T(n) = T_1(n) + T_2(n) = O(f_1(n)) + O(f_2(n)) \\ = O(\max(f_1(n), f_2(n)))$$

(2) 乘法规则

$$T(n) = T_1(n) \times T_2(n) = O(f_1(n)) \times O(f_2(n)) \\ = O(f_1(n) \times f_2(n))$$

$c < \log n < n < n^2 < 2^n$
常数, 对数, 线性, 二次, 指数 (增长率)。

ADT的进一步讨论

- 抽象数据类型是定义了一组操作的数学模型。
- 抽象数据类型隐藏了数据的表示方式和运算的实现方法, 是模块化的思想的发展, 它为模块的划分提供了理论依据。
- 抽象数据类型的引入, 提高了程序设计的抽象层次, 也为数据结构和算法的研究提供了一种分层的模型。
- 按照面向对象程序设计观点, 一种数据结构就是一种抽象数据类型的具体实现。
- 在面向对象程序设计语言中, 一种数据结构通常使用一个类 (Class) 进行封装。
- 在非面向对象程序设计语言中, 通常用结构 (Structure) 封装数据的存储结构, 用函数 (Function) 封装一个运算的实现。