

面向对象的C++

数据结构与算法实现

MIANXIANG DUIXIANG DE C++

SHUJU JIEGOU YU SUANFA SHIXIAN

韩珂 著



中国水利水电出版社
www.waterpub.com.cn

面向对象的C++ 数据结构与算法实现

MIANXIANG DUIXIANG DE C++
SHUJU JIEGOU YU SUANFA SHIXIAN

韩珂 著



内 容 提 要

本书以 C++ 为描述语言,系统地分析讨论了面向对象的数据结构。全书共分 8 章,主要内容包括绪论,线性表的实现及应用,栈、队列及递归思想,串及模式匹配算法,树与二叉树及算法实现,图的结构及算法实现,查找的结构及算法实现,排序算法及方法选择等。

本书可供计算机及相近专业的学生及研究人员阅读使用,也可以供从事计算机软件开发和工程应用的人员参考。

图书在版编目 (C I P) 数据

面向对象的 C++ 数据结构与算法实现 / 韩珂著. —
北京:中国水利水电出版社, 2014. 9
ISBN 978-7-5170-2413-2

I. ①面… II. ①韩… III. ①C 语言—程序设计②数
据结构③算法分析 IV. ①TP312②TP311.12

中国版本图书馆 CIP 数据核字 (2014) 第 199689 号

策划编辑:杨庆川 责任编辑:杨元泓 封面设计:崔 蕾

书 名	面向对象的 C++ 数据结构与算法实现
作 者	韩珂 著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址:www.waterpub.com.cn E-mail:mchannel@263.net(万水) sales@waterpub.com.cn
经 售	电话:(010)68367658(发行部)、82562819(万水) 北京科水图书销售中心(零售) 电话:(010)88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京鑫海胜蓝数码科技有限公司
印 刷	三河市天润建兴印务有限公司
规 格	170mm×240mm 16 开本 15.5 印张 194 千字
版 次	2015 年 4 月第 1 版 2015 年 4 月第 1 次印刷
印 数	0001—3000 册
定 价	46.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

前 言

计算机已经成为现代社会不可缺少的基本工具,近年来,计算机科学的发展日新月异,其应用深入到了人类社会的各个领域,取得了辉煌的成果,改变了而且继续改变着人们的生活。

作为计算机程序的重要组成部分,数据结构和算法实现一直是计算机领域关注的焦点之一。要从事与计算机技术尤其是应用技术相关的工作与研究,就必须具备坚实的数据结构基础,所以相关工作人员必须学会研究计算机所要加工和处理的数据的特征,掌握组织数据、存储数据和处理数据的基本方法,并能根据实际情况合理地选择数据结构和相应的算法。另外,面向对象的软件分析与设计技术是当今软件开发的主流方法,不仅如此,面向对象技术更是一种对真实世界进行抽象分析的重要思维方式,深刻理解这门技术,对于软件的开发应用以及计算机科学的发展意义重大。C++是一种支持面向对象的程序设计语言,不但实现了面向对象的程序设计要求,而且还继承了C语言的全部优点,是软件开发的有力工具,本书采用C++作为数据结构的描述语言。

在内容上,全书共分8章。主要内容包括绪论,线性表的实现及应用,栈、队列及递归思想,串及模式匹配算法,树与二叉树及算法实现,图的结构及算法实现,查找的结构及算法实现,排序算法及方法选择等。本书集先进性、科学性和实用性为一体,内容丰富、图文并茂、叙述简洁明了、可读性强,由浅入深、衔接自然、逻辑性强。

本书是作者在参考大量著作文献的基础上,结合自身多年的教学与研究经验撰写而成的,在此,向所参考文献的作者表示

真诚的感谢。另外,在撰写本书的过程中,得到了多位专家的指导和帮助,在此,同样致以真挚的感谢。限于作者水平,虽经多次修改,书中不免有不完善之处,希望同行学者和广大读者批评指正。

作者
2014年6月

目 录

前 言	(1)
第 1 章 绪论	(1)
1.1 数据结构的基本概念	(1)
1.2 算法及算法分析	(11)
第 2 章 线性表的实现及应用	(21)
2.1 线性表的定义及其基本操作	(21)
2.2 线性表的顺序存储结构	(26)
2.3 线性表的链接存储结构	(33)
2.4 线性表的应用	(49)
第 3 章 栈、队列及递归思想	(54)
3.1 栈	(54)
3.2 队列	(59)
3.3 递归	(67)
第 4 章 串及模式匹配算法	(76)
4.1 串的定义及其运算	(76)
4.2 串的顺序存储结构	(78)
4.3 串的链式存储结构	(84)
4.4 串的模式匹配算法	(88)
第 5 章 树与二叉树及算法实现	(96)
5.1 树	(96)
5.2 二叉树	(107)

5.3	遍历二叉树	(116)
5.4	线索二叉树	(125)
5.5	最优二叉树——哈夫曼树	(135)
第 6 章	图的结构及算法实现	(145)
6.1	图的存储结构	(145)
6.2	图的遍历	(155)
6.3	最小生成树	(160)
6.4	最短路径	(169)
第 7 章	查找的结构及算法实现	(176)
7.1	查找的基本概念	(176)
7.2	顺序表的查找	(178)
7.3	树表的查找	(184)
7.4	散列表的查找	(195)
第 8 章	排序算法及方法选择	(209)
8.1	排序的基本概念	(209)
8.2	插入排序	(213)
8.3	交换排序	(217)
8.4	选择排序	(224)
8.5	归并排序	(232)
8.6	基数排序	(235)
8.7	各种内排序方法的比较和选择	(236)
参考文献	(240)

第1章 绪论

随着计算机应用技术的飞速发展,计算机在人类社会的各个领域都扮演着越来越重要的角色,数据结构被称为是计算机科学的两大支柱之一,因此,要深刻了解计算机技术,就必须深入研究数据结构。目前随着计算机技术的迅速发展,计算机的运算速度已越来越快,但是这样的速度仍然跟不上人们解决实际问题的需求和数据量爆炸性的增长速度。所以程序的高效性不仅仅体现在编程技巧上,更重要的是基于合理、有效的数据组织和正确、优秀的算法。对于程序开发人员而言,掌握了程序的设计方法和技巧的同时还需要具有合理组织和安排数据的能力,才能设计出高效的程序。目前人们已在前人研究和实践的基础上总结出许多有效的数据存储结构和算法。本章我们就来分析讨论数据结构和算法。

1.1 数据结构的基本概念

1.1.1 数据结构的起源与地位

起初,人们使用计算机的主要目的是处理数值计算问题。使用计算机解决具体问题的一般步骤为:从具体问题抽象出适当的数学模型;设计或选择解决此数学模型的算法;编写程序并进行调试、测试;得出最终的解。由于最初涉及的运算对象是简单的整型、实型或布尔型数据,所以程序设计者的主要精力集中于程序设计的技巧上,而无须重视数据结构。随着计算机应用领域扩大和硬件的发展,非数值计算问题显得越来越重要。

这类问题涉及的数据结构更为复杂,数据元素之间的相互关系一般无法用数学方程式加以描述。因此,解决这类问题的关键不再是数学分析和计算方法,而是要设计出合适的数据结构。

美国计算机科学家 Donald Ervin Knuth 首次提出了数据结构和算法的概念,并在他所著的《计算机程序设计艺术第 1 卷基本算法》中首次较系统地阐述了数据的逻辑结构、存储结构及其操作。随后,数据结构越来越得到人们的重视,结构化程序设计逐渐成为当时程序设计的主流方法。著名的瑞士计算机科学家沃思(N. Wirth)教授曾提出:

算法 + 数据结构 = 程序

程序设计的实质是对实际问题设计、选择好的数据结构和好的算法,而好的算法在很大程度上取决于描述实际问题的数据结构。接下来,我们先看一个具体实例。

例 1.1 一个典型的学校行政机构如图 1-1 所示,这是一个层次结构。顶层结点“学校”代表整个系统,它的下一层结点代表这个系统的各个子系统,即部处与学院。再下一层结点代表更小的机构,如教务科、计算机系等,直到最底层一个小组或一个教研室等。在该图中,每一个结点代表一个数据元素,每一个结点的下面可能有多个结点。这样的一种结构称为层次型数据结构。

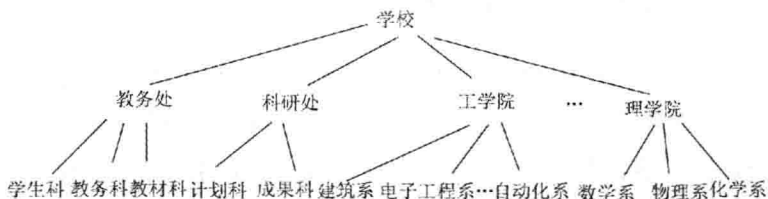


图 1-1 学校机构的“树结构”

在各种应用程序中会涉及各种各样的数据,为了组织它们、

存储它们,并且能对它们进行操作,就要考虑它们的归类以及它们之间的关系。然而,这些问题的数学模型不再是数学方程,我们需要建立相应的数据结构,并依此实现所要求的功能。例1.1中的结构为树结构,日常生活中我们还会遇到如表、图、集合之类的数据结构。

作为计算机科学的一门分支学科,数据结构主要研究非数值计算的程序设计问题中计算机的操作对象、对象之间的关系和操作等。为了编写出“好”的程序,必须分析待处理对象的特性及各对象之间的关系。

如表1-1所示,列出了数据结构的内容体系,包括3个层次的5个“要素”。其中,3个层次分别为抽象、实现与评价。通过抽象,舍弃数据元素的具体内容,就得到逻辑结构表示,通过分解将处理要求划分成各种功能,再通过抽象舍弃实现细节,就得到基本操作的定义。上述两个方面的结合将问题变换为数据结构,这是一个从具体问题到数据结构的过程,即从具体到抽象的过程。最后,通过增加对实现细节的考虑进一步得到存储结构和实现运算,从而完成设计任务,而这是一个从数据结构到具体实现的过程,即抽象到具体的过程。

表1-1 数据结构的内容体系

方面 层次	数据表示	数据处理
抽象	逻辑结构	基本操作
实现	存储结构	算法
评价	数据结构比较及算法分析	

针对要处理的问题,设计最有利于操作系统处理的数据结构时需考虑下列因素:

- ①数据的数量。
- ②数据的使用次数和方式。

- ③数据的性质是动态的还是静态的。
- ④数据结构化后需要多大的存储空间。
- ⑤存取结构化后的数据所需花费的时间。
- ⑥是否容易程序化。

数据结构已经成为计算机科学及其相关领域的重要基础,它与数学、计算机硬件和计算机软件之间有着密切的关系。在计算机科学中,数据结构不仅是一般程序设计的基础,而且是设计和实现编译程序、操作系统以及数据库系统等大型程序的基础。无论是系统软件还是应用软件都要用到各种类型的数据结构。如图 1-2 所示,表示的是数据结构在计算机领域的地位。深刻理解和研究数据结构,对计算机科学的发展意义重大。^①

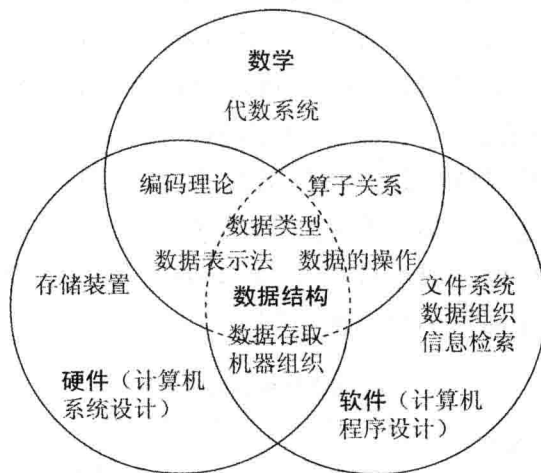


图 1-2 数据结构的地位

^① 余腊生. 数据结构:基于 C++ 模板类的实现. 北京:人民邮电出版社,2008

1.1.2 数据结构的基本概念和常用术语

1. 数据结构研究的内容

(1) 数据(data)

数据是指能够输入到计算机中,并被计算机识别和处理的符号的集合。

例如:数字、字母、汉字、图形、图像、声音都称为数据。

(2) 数据元素(data element)

数据元素是组成数据的基本单位。

数据元素是一个数据整体中相对独立的单位。但它还可以分割成若干个具有不同属性的项(字段),故不是组成数据的最小单位。

(3) 数据对象(data object)

是性质相同的数据元素组成的集合,是数据的一个子集。

例如,整数数据对象的集合可表示为

$$N = \{0, \pm 1, \pm 2, \dots\}$$

字母字符数据对象的集合可表示为

$$C = \{'A', 'B', \dots, 'Z'\}$$

(4) 结构

结构指的是数据元素之间的相互关系,即数据的组织形式。

(5) 结点

结构中的数据元素称为结点。

(6) 数据结构

对数据结构的概念,至今还没有一个标准的定义,所以只能从研究的领域来理解数据结构。数据结构是指带有结构的数据元素的集合,描述的是数据之间的相互关系,即数据的组织形式。数据结构一般包括以下三个方面的内容:

- ①数据元素之间的逻辑关系,也称为数据的逻辑结构。
- ②数据元素及其关系在计算机存储器内的表示,称为数据

的存储结构。

③数据的运算,即对数据施加的操作。

2. 数据的逻辑结构

数据的逻辑结构是从逻辑关系上描述数据的,它与数据元素的存储结构无关,是独立于计算机的。因此,数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。

一个数据结构具有两个要素,一个是数据元素的集合,另一个是关系的集合。因此在形式上数据结构可以采用一个二元组来表示。数据结构的形式定义为

$$\text{数据结构} = (D, R)$$

其中, D 是数据元素的有限集合, R 是 D 上关系的有限集合。

例如,复数可被定义为一种数据结构

$$\text{Complex} = (D, R)$$

其中, $D = \{x \mid x \text{ 是实数}\}$, $R = \{ \langle x, y \rangle \mid x, y \in D, x \text{ 称为实部}, y \text{ 称为虚部}\}$ 。

根据数据元素之间关系的不同特性,通常有下列 4 类基本结构,如图 1-3 所示。

(1)集合结构。其中所有的数据元素都属于同一个集合。集合是元素关系极为松散的一种结构,因此也可用其他结构来表示。

(2)线性结构。数据元素之间存在一对一的关系。

(3)树形结构。数据元素之间存在一对多的关系。

(4)图形结构。数据元素之间存在多对多的关系。图形结构也称作网状结构。

数据的逻辑结构描述数据元素之间的逻辑关系,分为线性结构和非线性结构。

①线性结构的逻辑特征。若结构是非空集,则有且仅有一个首元素结点和一个尾元素结点,并且所有结点都最多只有一个直接前驱和一个直接后继。

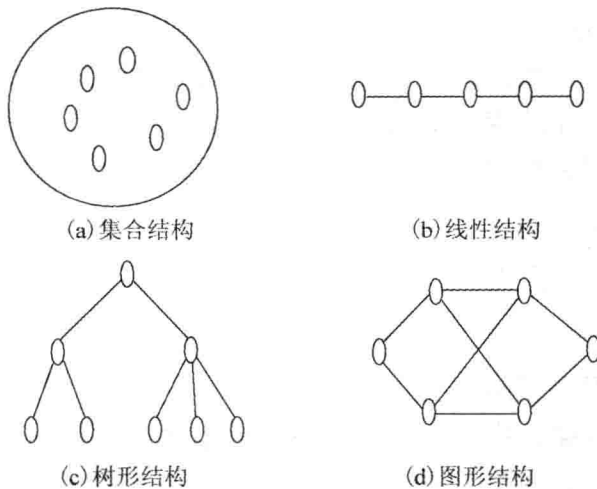


图 1-3 4类基本结构

例如,线性表、栈、队列和串等都是线性结构。

②非线性结构的逻辑特征。一个结点可能有多个直接前驱和直接后继。

例如,数组、广义表、树和图等都是非线性结构。

3. 数据的存储结构

数据的存储结构是逻辑结构用计算机语言的实现(亦称为映像),它是依赖于计算机语言的。数据的存储结构可以用以下4种基本的存储方法实现。

(1) 顺序存储方法

顺序存储方法是把逻辑上相邻的结点存储在物理位置相邻的存储单元里。由此得到的存储结构称为顺序存储结构。这通常是借助于程序设计语言的数据类型描述的。该方法主要应用于线性数据结构,非线性数据结构也可通过某种线性化的方法来实现顺序存储。

(2) 链式存储方法

在这种存储结构中,数据元素之间的逻辑关系由附加的指针来表示。因此,对数据元素的具体存放位置没有限制,只要在

存放一个数据元素的同时存储一个或多个指针,让这个或这些指针指向与本数据元素相关联的数据元素就可以了。图 1-4 给出了链式存储结构的示意图。链式存储结构的特点是链上相邻的数据元素在物理上不一定相邻,数据元素之间的逻辑关系由指针表现出来。

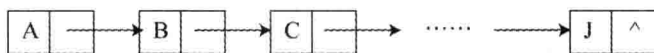


图 1-4 链式存储结构

(3) 索引存储方法

索引存储方法通常是在存储结点信息的同时,还建立附加的索引表。表中索引项的一般形式是含有关键字和地址,关键字是能唯一标识一个结点的数据项。

(4) 散列存储方法

散列存储结构又称为哈希结构,这种存储结构的本质是将数据元素的关键字带到散列函数中,通过计算得到其存储的地址。也就是说,通过散列函数在数据元素与其存储地址之间建立起直接的联系。

同一种逻辑结构采用不同的存储方法,可以得到不同的存储结构。选择何种存储结构来表示相应的逻辑结构,要视具体的应用系统要求而定,而主要考虑的还是运算方便及算法的时间和空间上的要求。

无论怎样定义数据结构,都应该将数据的逻辑结构、数据的存储结构及数据的运算这三方面看成一个整体。因此,存储结构是数据结构不可缺少的一个方面。

4. 运算

如果说数据的逻辑结构描述了数据的静态特性,那么在数据的逻辑结构上定义的一组运算给出了数据被使用的方式,即数据的动态特性。使用数据结构上定义的运算,用户可对数据结构的实例或组成实例的数据元素实施相应的操作。运算的结

果可使数据改变状态。如果一个数据结构一旦创建,其结构不发生改变,则称为静态数据结构,否则称为动态数据结构。

数据结构的最常见的运算有:

- ①创建运算。创建一个数据结构。
- ②清除运算。删除数据结构中的全部元素。
- ③插入运算。在数据结构中插入一个新元素。
- ④删除运算。将数据结构中的某个指定元素删除。
- ⑤搜索运算。在数据结构中搜索满足一定条件的元素。
- ⑥更新运算。修改数据结构中某个指定元素的值。

⑦访问运算。访问数据结构中某个元素。

⑧遍历运算。按照某种次序,系统地访问数据结构的各元素,使得每个元素恰好被访问一次。

请注意,在上面的讨论中,我们一般不明确区分数据结构和它的实例。例如,我们称创建一个数据结构,很显然是指创建某种数据结构的一个实例。

5. 数据类型

在早先的高级程序语言中都有数据类型的概念。数据类型是一组性质相同的值的集合以及定义在这个集合上的一组操作的总称。

前面我们已经提到“算法 + 数据结构 = 程序”。在早先的高级程序语言中,数据结构是通过数据类型来描述的。数据类型用于刻画操作对象的特性。每个变量、常量和表达式都属于一个确定的数据类型,例如整型、实型、字符型等,这些数据类型规定了数据可能取值的范围以及允许进行的操作。例如,C++语言中的整数类型,就给出了一个整型量的取值范围(依赖于不同的机器或编译系统),定义了对整型量可施加的加、减、乘、除和取模等算术运算。

在高级程序语言中,数据类型分为如下两种:

- ①基本类型,如整型、实型、字符型等,其取值范围和允许的

操作都是由系统预先规定的。

②组合类型,它由一些基本类型组合构造而成,如记录、数组、结构等。

基本数据类型通常是由程序语言直接提供的,而组合类型则由用户借助程序语言提供的描述机制自己定义。这些数据类型都可以看成是程序设计语言已实现了的数据结构。

6. 抽象数据类型

抽象数据类型(ADT)是指基于一个逻辑概念上的类型以及在这个类型上的一组操作。抽象数据类型主要是由于引入了面向对象的程序设计而提出的一种理论。抽象数据类型包含了一般数据类型的概念,但比一般的数据类型更广、更抽象。抽象数据类型与一般数据类型的不同之处在于与任一高级语言无关,其只考虑数据间的逻辑结构,不考虑具体存储结构和操作的具体实现,所以其带动了计算机语言的发展,而不是由计算机语言带动数据结构的发展。^①

抽象数据类型的特征有:

①数据抽象。强调本质的特征、功能和与外部的接口。

②数据封装。外部特征和其内部实现的细节分离,并且对外部用户隐藏其内部细节。

抽象数据类型的描述格式为:

```
ADT 抽象数据类型名 {  
    数据对象: <数据对象的定义 >  
    数据关系: <数据关系的定义 >  
    基本操作: <基本操作的定义 >  
} ADT 抽象数据类型名。
```

数据对象和数据关系的定义可以用伪码描述。

数据基本操作的定义格式:

^① 许乐平. 数据结构: C++ 描述. 北京: 中央广播电视大学出版社, 2006