



21世纪应用型人才培养系列教材

数据结构

(第二版)

Data Structure (Second Edition)

杨 彬 朱晓芸 编著



高等教育出版社
HIGHER EDUCATION PRESS

21 世纪应用型人才培养系列教材

数 据 结 构

(第二版)

杨 彬 朱晓芸 编著

高等教育出版社

内容简介

本书是 21 世纪应用型人才培养系列教材之一,是计算机及相关专业的专业基础课程教材。本书通过大量的程序实例,介绍数据结构的基本概念和一些常用的数据结构,阐明数据结构的内在逻辑关系,讨论它们在计算机中的存储表示,并结合典型应用介绍在各种结构上定义的运算实现方法。

本书配有基于网络的教学平台,提供了教学课件、习题及答案、模拟测试等丰富的教学资源。

本书适用于高等教育计算机及相关专业,也可作为高等职业院校计算机及相关专业的教材,亦可供广大计算机爱好者学习使用。

图书在版编目(CIP)数据

数据结构/杨彬,朱晓芸编著. —2 版. —北京:高等教育出版社,2006. 6
(21 世纪应用型人才培养系列教材)
ISBN 7-04-019490-2

I. 数... II. ①杨...②朱... III. 数据结构—高等学校—教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2006)第 048903 号

责任编辑 古 锋 封面设计 吴 昊 责任印制 潘文瑞

出版发行	高等教育出版社	购书热线	010—58581118
社 址	北京市西城区德外大街 4 号		021—56964871
邮政编码	100011	免费咨询	800—810—0598
总 机	010—58581000	网 址	http://www.hep.edu.cn
传 真	021—56965341		http://www.hep.com.cn
			http://www.hepsh.com
经 销	蓝色畅想图书发行有限公司	网上订购	http://www.landaco.com
排 版	南京理工出版信息技术有限公司		http://www.landaco.com.cn
印 刷	江苏宜兴德胜印刷有限公司	畅想教育	http://www.widedu.com
开 本	787×1092 1/16	版 次	2000 年 7 月第 1 版
印 张	15.5		2006 年 6 月第 2 版
字 数	378 000	印 次	2006 年 6 月第 1 次
		定 价	22.00 元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

物料号 19490—00

前 言

数据结构是计算机及相关专业的一门重要的专业基础课,它的主要任务是讨论数据的各种逻辑结构、物理结构以及相关的算法,使学生能够根据实际问题的需要选择合适的数据结构和设计算法,从而具有设计与编制一定规模的软件系统的能力。

本次出版已为该教材的第二版,第二版在第一版的基础上作了修改和补充,对整本书的结构进行了重新组织,主要修订内容如下:

1. 增加了“多维数组、稀疏矩阵与广义表”作为本书的第5章,其中介绍了多维数组、稀疏矩阵和广义表的基本概念及存储表示。第一版中的第5章、第6章、第7章、第8章、第9章相应地成为本书的第6章、第7章、第8章、第9章和第10章。

2. 在第4章中,增加了“递归与栈”;在第7章中,增加了“拓扑排序”;在第8章中,增加了“平衡二叉树查找”;在第9章中,增加了“基数排序”和“外部排序”,以便对排序算法进行更全面的介绍。

3. 除第1章外,每一章都增加了1~2道典型的实验题,对其进行分析,并给出了参考程序,以培养读者的实践能力。

本书共分10章,第1章讲述了数据结构与算法的基本概念,作为数据结构和算法的基础;第2章介绍了C语言中的数据结构;第3章和第4章讨论了线性数据结构中的线性表、栈、队列和串及其应用;第5章介绍了多维数组、稀疏矩阵和广义表的基本概念及存储表示;第6章和第7章介绍了树和图这两种基本的数据结构及其应用;第8章和第9章讨论了查找和排序,介绍了各种实现算法;第10章介绍了常用的文件结构。本书采用结合自然语言的类C语言作为算法描述语言,既便于理解和表达,也便于程序实现。

本书在内容组织上力求深入浅出、通俗易懂,并且注重实际的程序设计应用。书中有针对性地给出了一些示例程序,以求更好地阐明数据结构的原理和方法。另外,本书配套有基于网络的教学平台,提供了教学课件、习题及答案、模拟测试等丰富的教学资源。相关教师可登录网站(网址:www.hepsh.com)在线下载,或填写书后所附的《教学课件索取单》后来函索取。

本书由浙江大学计算机学院杨枰、朱晓芸编著,适用于高等教育计算机及相关专业,也可作为高等职业院校计算机及相关专业的教材,亦可供广大计算机爱好者学习使用。

由于编者水平有限,书中疏漏在所难免,敬请读者批评指正!

编 者

2006年4月

郑 重 声 明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》,其行为人将承担相应的民事责任和行政责任,构成犯罪的,将被依法追究刑事责任。为了维护市场秩序,保护读者的合法权益,避免读者误用盗版书造成不良后果,我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为,希望及时举报,本社将奖励举报有功人员。

反盗版举报电话:(010)58581897/58581896/58581879

传 真:(010)82086060

E - mail:dd@hep.com.cn

通信地址:北京市西城区德外大街4号

高等教育出版社打击盗版办公室

邮 编:100011

购书请拨打电话:(010)58581118

目 录

第 1 章 绪论	1	实验	59
1.1 数据结构的基本概念	1		
1.2 算法的概念与描述	3		
小结	6		
习题	6		
第 2 章 C 语言中的数据类型	7	第 4 章 栈和队列	65
2.1 C 语言中数据的存储表示	7	4.1 栈	65
2.2 基本数据类型	9	4.2 栈的应用	68
2.3 派生数据类型	16	4.3 队列	75
2.4 复合数据类型	19	4.4 队列的应用	81
小结	21	小结	85
习题	22	习题	85
实验	22	实验	85
第 3 章 线性表	25	第 5 章 多维数组、稀疏矩阵与	
3.1 线性表的定义	25	广义表	92
3.2 线性表的顺序存储结构及其		5.1 多维数组	92
运算	26	5.2 稀疏矩阵	95
3.3 线性表的链式存储结构及其		5.3 广义表	98
运算	32	小结	99
3.4 串	44	习题	100
3.5 线性表的应用	51	实验	100
小结	58	第 6 章 树	103
习题	58	6.1 树的基本概念	103
		6.2 树的存储表示	104
		6.3 二叉树	107
		6.4 二叉树的遍历	111

6.5 线索二叉树	121	实验	196
6.6 树与二叉树的转换	127		
6.7 赫夫曼树	130	第9章 排序	201
小结	135	9.1 选择排序	201
习题	136	9.2 插入排序	205
实验	136	9.3 交换排序	208
第7章 图	141	9.4 归并排序	212
7.1 图的数学基础与基本概念 ...	141	9.5 基数排序	214
7.2 图的存储表示	142	9.6 外部排序	215
7.3 图的遍历	157	小结	215
7.4 图的连通性	161	习题	216
7.5 图的最短路径	164	实验	216
7.6 拓扑排序	167	第10章 文件	220
小结	169	10.1 文件的结构	220
习题	169	10.2 文件的目录	224
实验	170	10.3 文件的访问	226
第8章 查找	173	小结	233
8.1 静态查找	173	习题	233
8.2 动态查找	178	实验	233
小结	196	参考文献	239
习题	196		

第1章 绪论

计算机是帮助人们对各种数据进行处理机器。这里所说的数据指的是计算机所能处理的一切数值、字符、图形等计算机能够识别、存储和处理的信息。

在计算机发展的早期,计算机主要用于科学计算,处理的是数值数据。此时,数据的特点是数据元素之间关系简单,数据量小,形式较一致。因此,此时的程序设计工作者重于算法,而不注重对数据的组织、性质、关系的研究。

随着计算机工业的飞速发展及计算机应用的日益普及,计算机已越来越广泛地应用于各种非数值处理问题,其应用已逐步深入到事务管理、工业控制、通信、教育等领域。这些待处理的数据形式多样,量越来越大,关系越来越复杂。要想对这些数据进行有效的处理,就必须要了解数据的性质、组织方式及相互之间的关系。这样才能采用适当的方式进行存储,选用高效的算法进行处理。这些正是数据结构所研究的内容,也就是说,数据结构是一门对计算机处理的数据进行表示、组织及操作进行研究的学科。

在计算机科学中,数据结构与算法是密不可分的,缺一不可。简单而论,算法指的是对计算机求解问题的全过程及具体步骤的描述。计算机程序正是按照算法所描述的步骤对某种结构的数据进行加工处理的。如果没有数据结构,计算机就缺少了处理的“原料”。而没有算法,计算机就缺少了求解问题的方法。因此,著名的计算机科学家 N. Wirth 在论述关于算法与数据结构之间的关系时,就曾指出:“程序=数据结构+算法”。因此,本书在讲述数据的结构关系时,同时对施加于其上的操作及其实现也作出描述。也就是说,数据结构这门课程本身也包含着算法,即计算机在处理具有结构关系的数据时涉及的基本操作及其实现的方法。

1.1 数据结构的基本概念

1. 数据

数据是程序设计的“原料”及结果。在计算机科学中,凡是能在计算机中出现的信息都是数据。数据具有不同的类型,计算机存储和处理的方法也各不相同。每种高级程序语言中都提供了不同的数据类型,以供程序设计使用。程序中,每个数据都属于某种类型,它有一定的取值范围和表示形式以及在相对于该类型的数据可执行的操作的种类。因此,数据类型指的是一个值的集合以及定义在该集合上的操作的集合。数据类型可分为固有数据类型和用户自定义数据类型两种。当使用由语言所提供的数据类型(即固有数据类型)时,语言已定义了该类型数据的取值范围、表示形式及该类型数据所允许的操作,用户只需按规定直接使用。也就是说,固有数据类型可看做是程序设计语言中已实现的数据结构。而如果用户在程序设计中自定义数据类型,则需要用户在固有数据类型的基础上,从具体应用中抽象出适当的数学模型,找出描述这个数学模型的新的数据结构,提供从结构的建立到施加在其上的各种操作的实现的算法。

简而言之,数据指的是一切可以由计算机识别、存储和处理的描述客观事物的符号,如数值、字符、图像、声音等。

数据元素是用于描述数据的基本单位。数据元素可由数据项组成。数据项是数据表示中不可分割的最小标识单位。

数据对象是具有相同性质的数据元素所组成的集合。一般情况下,数据对象中所包含的数据元素之间具有一定的组织形式和结构关系。数据结构指的是对数据元素及其相互关系的描述。也就是说,数据结构包含了两方面的内容:一方面是对数据元素的有限集合即数据对象的描述,另一方面是对数据对象中数据元素的关系的描述。

2. 数据结构的基本类型

数据结构中的这种数据关系是独立于计算机的,也就是说,是一种逻辑关系。数据结构有三种基本类型:线性数据结构、树型数据结构和图型数据结构。

(1) 线性数据结构

如表 1-1 所示的工资表,其逻辑组织结构就是一种线性数据结构。

表 1-1 线性数据结构示例

工 号	姓 名	工资(元)	奖金(元)	合计(元)
001	张三	500.00	100.00	600.00
002	李四	600.00	100.00	700.00
003	王五	550.00	80.00	630.00
004	赵六	700.00	120.00	820.00
⋮	⋮	⋮	⋮	⋮

在这张工资表中,每一行都记录了每名职工的工资收入,它是构造这张工资表的最基本的数据单位,即数据元素,也可称为结点。每一行又可分成五个独立的数据项:工号、姓名、工资、奖金和合计,这五个数据项是不可分的。整个工资表就是一个数据对象。工资表中的每个数据元素之间的组织关系是一种线性结构,即每个数据元素都有且只有一个前驱,每个数据元素也都有且只有一个后继。只有第一个数据元素没有前驱,最后一个数据元素没有后继。

(2) 树型数据结构

如图 1-1 所示是一张某公司的人事关系示意图,它是一种典型的树型数据结构。树型结构呈现的是一种层次组织关系,组成这种关系的数据元素是结点,每一个结点又有若干个数据项(在结点中称为域)组成,结点与结点之间的联系仅限于上下两层之间的一对多关系,即每一个上层结点可与多个下层结点相关联,但每一个下层结点却只能与一个上层结点相联系。在图 1-1 中,每一个人员的信息都用一个结点来描述,结点中可包括如下信息:姓名、年龄、文化程度、职位……这些信息在图中都被省略了。第一层的结点代表的是总经理,他管理着若干个部门经理,而部门经理则处于第二层上,每一个部门经理又领导着若干个职员。

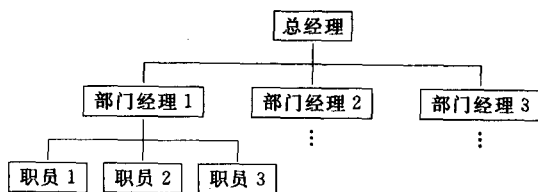


图 1-1 树型数据结构示例

(3) 图型数据结构

如图 1-2 所示是国内若干城市之间的航线图,它是一个典型的图型数据结构,组成这种数据结构的数据元素是顶点。在图中,每个顶点都可与若干个顶点相连,是一种多对多的关系。在航线图中,对每个城市的描述就可用一个顶点来实现,关于每个城市的基本信息,如城市的名称、机场的位置、航线的多少等就可由顶点中的域,亦即数据项来描述,这些描述在航线图中被省略了。

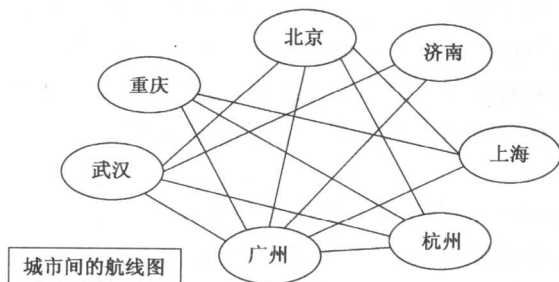


图 1-2 图型数据结构示例

3. 数据的存储结构

与数据的逻辑结构相对应的是数据的存储结构,它指的是数据元素及其逻辑关系在计算机存储器中的存储形式,它是依赖于计算机的,也称为数据的物理结构。数据的存储结构分两类:顺序存储结构与非顺序存储结构。在顺序存储结构中,逻辑上相邻的数据元素在存储器中物理位置上也相邻。也就是说,数据元素之间的逻辑关系可通过数据元素在存储器中的相对位置来体现。在非顺序存储结构中,数据的逻辑关系与其物理位置无关,在逻辑上相邻的数据元素可存放在存储器的任意位置上;数据元素之间的逻辑关系通过指针来映射。非顺序存储结构也叫链式存储结构。

1.2 算法的概念与描述

1. 算法的基本特点

算法描述的是计算机解决问题的具体步骤的序列,这个序列必须是有限的。它有五个基本特点:

(1) 确定性

算法的每一个步骤都必须有确定的含义,不能有二义性和模糊性,也就是说,对于同样的初始状态应得出相同的结果。

(2) 可行性

算法中的每一步都应在有限的执行时间内完成,否则该算法就不可能得到实现。

(3) 有穷性

算法应在有限的执行步骤里完成。

(4) 输入

每个算法都应具有零个或多个输入以描述问题的初始状态。

(5) 输出

每个算法也都应具有至少一个以上的输出作为它的结果。

2. 算法描述语言

算法总是施加在特定类型的数据结构之上的。在算法的设计过程当中,对数据的各种运算都是定义在数据的逻辑关系之上的,是处于比较抽象层次上的操作,也就是说,主要考虑的是“做什么”的问题。而究竟这些操作如何实现则依赖于数据所采用的存储结构。当然,在算法讨论中,我们对于存储结构的描述是在高级程序设计语言的基础上进行的,而不是像在机器语言当中用内存地址来直接描述数据的存储结构。由于在每种高级程序设计语言中所定义的数据类型都对应于一定的物理结构,实际上就是对数据的存储结构的一种抽象,因此,本书所讨论的存储结构指的是在 C 语言的数据类型基础上的一种较为抽象的虚拟存储结构。

本书对算法的描述,采用的是结合自然语言的一种类 C 语言的描述方法。用自然语言描述便于人们相互交流和理解,也可以更加突出重点,而不拘泥于局部的细节实现,且条理性更好,也更便于表达。同时,结合 C 程序设计语言的描述使算法的表达更加准确,也更便于最终的程序实现和对算法性能的分析。

下面是对本书中所使用的算法描述语言的简介。

(1) 数据类型

本书中所使用的算法描述语言的数据类型包括 C 语言中的所有数据类型(整型、实型、字符型、数组、指针、结构、联合等)。除了这些以外,还可以定义新的数据类型名来取代已有的数据类型名。定义形式为:

```
typedef 已有数据类型名 新的数据类型名;
```

例如:

```
typedef int integer;
```

(2) 变量和符号常量

变量的定义形式为:

```
数据类型 变量名序列;
```

符号常量的定义形式为:

```
#define 符号常量名 常量值;
```

其中,符号常量名常采用具有一定含义的标识符来命名,例如:

```
#define TRUE 1
```

(3) 数据运算

数据运算主要包括算术运算、关系运算和逻辑运算三种。

算术运算符有: +、-、*、/、%、++、--等;

关系运算符有: >、<、==、>=、<=、!=;

逻辑运算符有: !、&&、||;

另外还有指针运算符 *、&,分量运算符 .、->和下标运算符 []。

(4) 赋值语句

赋值语句的形式为:

```
变量名 = 表达式;
```

在表达式中,除了可以出现上述数据运算之外,还可以出现常用的数学符号,如 $\sum$ 、 $\cup$ 、 $\cap$ 、 $\in$ 等。

(5) 控制语句

控制语句包括:

① 选择语句,包括条件语句、多项选择语句。

条件语句的形式为:

```
if(条件表达式)
    语句块;
```

或

```
if(条件表达式)
    语句块 1;
else
    语句块 2;
```

多项选择语句的形式为:

```
switch(表达式){
    case 常量表达式 1:语句块 1; break;
    case 常量表达式 2:语句块 2; break;
    :
    default:语句块 n;
}
```

② 循环语句,包括 for 语句、while 语句和 do-while 语句。

for 语句的形式为:

```
for(循环变量初始表达式;终止条件表达式;循环变量修改表达式)
    语句块;
```

while 语句的形式为:

```
while(条件表达式)
    语句块;
```

do-while 语句的形式为:

```
do {
    语句块;
} while(条件表达式);
```

(6) 函数的定义、声明与调用

函数的定义形式为:

```
返回类型 函数名(形式参数列表){
    函数定义语句块;
};
```

函数的声明形式为:

```
返回类型 函数名(形式参数列表);
```

函数的调用形式为:

函数名(实际参数列表);

(7) 输入、输出

输入语句为:

```
scanf(格式字符串,输入变量序列);
```

输出语句为:

```
printf(格式字符串,输出表达式序列);
```

(8) 结束语句

异常结束语句 `exit`, 其形式为:

```
exit(异常代码);
```

(9) 注释

注释语句的形式为:

```
/* 注释内容 */
```

小结

本章对数据结构及算法中的基本概念进行了介绍,并且用示例的方式对本书中要介绍的数据结构类型进行了说明。

本书对算法的描述采用的是一种结合了自然语言和 C 语言特点的类 C 描述语言,并对这种语言的语法进行了说明。

习题

1. 什么是数据元素?
2. 什么是数据对象?
3. 什么是数据结构?
4. 主要的基本数据结构类型有哪些?
5. 算法有哪些基本的特点?

BCD 数字,它被称为普通的二进制编码的十进制数。其中低 4 位用于表示有效的十进制数值,而高 4 位二进制数空闲不用,其取值可以是全 0 或全 1。

表 2-1 十进制数与二进制位的对应关系

二进制位组合	十进制数值
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

如果每字节存储两个 BCD 数字,则称为紧缩的二进制编码的十进制数。这种情况经常发生在数据在通信链路上进行传输时。紧缩二进制编码的十进制数可减少用于数据传输的时间,因为每个字节可用于传送 2 个 BCD 数字。

例如,用紧缩的 BCD 格式存储的数字 56。

7 6 5 4 3 2 1 0 二进制位

+---+---+---+---+---+---+---+---+

| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

+---+---+---+---+---+---+---+---+

最高有效位 最低有效位

高 4 位组存放值 5,与此同时低 4 位组存放值 6。

2.1.4 ASCII 码

美国信息交换标准码(ASCII 码)是用于信息交换的一种标准代码,它用 7 个二进制位的 128 种组合来表示字符 A~Z(包括大小写)、特殊字符< . ? :等、数字 0~9 及用于设备控制的特殊控制码。

如表 2-2 所示是对应于每个字符的编码组合。

表 2-2 ASCII 编码表

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

计算机通常是以 8 位为单位来存储信息的,因此在 ASCII 码中的第 8 位未被用到,通常被设为 0。某些系统可以应用 8 位来表示图像或不同语言符号,如希腊的字符。控制代码使用在通信和打印机中。在 ASCII 键盘上按住 Ctrl(控制)键并且按下另一个键(A~Z, {, \, }, ^, <-)即可产生。

例如,字符串“Hello.”在 ASCII 中的表示(十六进制)如下所示:

H=48

```
e=65
l=6C
l=6C
o=6F
.=2E
```

因此,该字符串由以下字节序列表示:

```
48 65 6C 6C 6F 2E
```

2.2 基本数据类型

本节对 C 程序中所使用的各种基本数据类型进行介绍。C 语言中的基本数据类型包括整数类型(int)、字符类型(char)、实数类型(float 和 double)三种类型,它们由硬件系统直接支持实现。下面将说明数据如何被存储、访问及其在程序中的典型使用方法。

2.2.1 字符

字符用于表示非数字符号的数据,例如:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K
L M N O P Q R S T U V W X Y Z # $ % & * ( ) _ - = + | \ ' , . / ; ' [ ] }
: " < > ?
```

每个字符以 8 位存储,一个字节可表达 256 种不同的字符。

在 C 语言中,字符常量是以单引号括起来的一个字符,如'a'。计算机系统中,通常用 ASCII 代码来存储字符,即并不是把字符本身存放到内存中,而是将该字符的相应 ASCII 码值存放到存储器中。如'a'的 ASCII 码值为 97,在内存中以如下方式存放:

```
+-----+
|  97  |
+-----+
```

字符变量以如下方式定义和使用:

```
char ch1, ch2;
ch1 = 'x';
ch2 = 'y';
```

上述例子表明,变量 ch1、ch2 是字符类型,这样存储器就为它分配 8 位存储空间以存储它的值。在后面的两条赋值语句分别把字符'x'、'y'赋给字符变量 ch1 和 ch2。这就是把它们的相应的 ASCII 码值 120 和 121 放到存储单元中,如下所示:

```
+-----+
ch1 | 120 |
+-----+
+-----+
ch2 | 121 |
+-----+
```

2.2.2 字符串

字符串是字符的序列。每个字符被相继地存储,每个占用存储器的8位存储空间。字符串常量是用一对双引号括起来的字符序列。字符串常量"Hello"将被存储如下:

```

+-----+
|  48  |
+-----+
|  65  |
+-----+
|  6C  |
+-----+
|  6C  |
+-----+
|  6F  |
+-----+
|   0  |
+-----+

```

C语言中,在每个字符串的结尾加上结束标志'\0','\0'的ASCII码值为0,所以"Hello"在内存中的长度为6个字符,而不是5个字符。在C语言中,没有专门的字符串变量,字符串如果需要存放在变量中,就需要用字符数组来存放。

2.2.3 整数

数字信息用ASCII格式是不能有效地进行存储的。假设用ASCII码存储数字123 456,这就需要占用6个字节的存储空间,并且很难表示数值的符号为正或负(尽管可在它之前加+或-),更有效地存储数字的方法是使用另外的编码方案。

整数有16位(整型int、短整型short int)和32位(长整型long int)长度之分,每种长度中又有有符号和无符号之分。下面以16位整数为例来讨论其表示。

```

位
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
+---+-----+
| S |      二进制形式表示的数值      |
+---+-----+

```

S=符号位

符号位(第15位)表示该数字为正还是负,逻辑1表示负,逻辑0表示正。数字的值被转化为二进制,存储在0~14位中。

例如存储整数值+7。

① 符号位是0。