教育部大学计算机课程改革项目成果
国家级优秀教学团队成果

算法与高级语言程序设计

高飞 白霞 主编
薛艳明 聂青 副主编

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

教育部大学计算机课程改革项目成果
国家级优秀教学团队成果

算法与高级语言程序设计

高 飞 白 霞 主 编
薛艳明 聂 青 副主编

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书是教育部高等教育司大学计算机课程改革项目“理工类高校计算思维与大学计算机课程研究与教材建设”的成果之一。本书兼顾程序设计语言和算法的学习，在介绍C++语言的程序设计方法的基础上，采用C++程序设计语言描述算法。

全书共 12 章，分为上下篇。上篇讲述高级语言程序设计基础；下篇在介绍算法设计及算法的性能度量后，介绍 C++语言描述的典型数据结构和经典算法的设计与分析技术。

本书内容由浅入深、循序渐进、案例丰富、通俗易懂、实用性强，可作为高等学校理工类计算机及相关专业学生的教材，也可供从事程序设计的工程人员参考。

图书在版编目(CIP)数据

算法与高级语言程序设计 / 高飞，白霞主编. —北京：电子工业出版社，2015.1
ISBN 978-7-121-24712-5
Ⅰ. ①算… Ⅱ. ①高…②白… Ⅲ. ①电子计算机－算法理论－高等学校－教材②C 语言－程序设计－高等学校－教材 Ⅳ. ①TP301.6②TP312
中国版本图书馆 CIP 数据核字（2014）第 265774 号

策划编辑：章海涛
责任编辑：章海涛　　特约编辑：何　雄
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
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
开　　本：787×1092　1/16　　印张：17.75　　字数：500 千字
版　　次：2015 年 1 月第 1 版
印　　次：2015 年 1 月第 1 次印刷
定　　价：39.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

本书是教育部高等教育司大学计算机课程改革项目“理工类高校计算思维与大学计算机课程研究与教材建设”的成果之一。本书编写团队是“计算思维导论”项目的参研者，也是国家级优秀教学团队和北京市优秀教学团队“计算机公共课教学团队”的主要成员。

在参加由哈尔滨工业大学战德臣教授主持的大学计算机课程改革项目“理工类高校计算思维与大学计算机课程研究与教材建设”的研究中，本书编写团队逐步对计算思维的精髓和本质加深了理解。我们认为，计算机解题的核心是算法设计，算法设计是在良好数学素养的基础上，对实际问题进行抽象和形式化，从而使用计算机来解决实际问题，而这恰恰正是培养计算思维的过程。

在计算学科领域，问题求解的核心是算法和系统。算法类问题强调的是数学建模及算法设计和分析。而程序设计语言则是实现算法的载体，语言只有满足算法实现的需求，才能被认识和掌握，算法只有通过程序语言才能实现。因此，本书兼顾了程序设计语言和算法的学习。

本书在介绍 C++语言的程序设计方法的基础上，采用 C++语言描述算法。C++语言是一种既支持面向过程程序设计，又支持面向对象程序设计的混合型语言，它独特的面向对象特征，可以为面向对象技术提供全面支持，是描述算法的一种较为理想的语言。采用面向对象程序设计语言描述算法，不仅有利于学习算法设计和面向对象技术，也为上机实践提高高级语言程序设计水平提供了方便。

全书共 12 章，分为上下篇。

上篇为高级语言程序设计基础，包括第 1～8 章。第 1～3 章，介绍 C++简单程序设计，没有学习过“C 语言程序设计”的读者可通过选学这部分内容，掌握 C++程序设计基础。第 4～8 章系统介绍 C++语言的封装性、继承性、多态性，以及模板的概念和使用。

下篇为算法分析与实现，包括第 9～12 章。第 9 章是算法导论，主要介绍算法的基本概念、算法设计以及算法的性能度量所需要的数学基础。第 10 章主要介绍 C++语言描述的典型数据结构，包括线性表、栈和队列、树以及哈希表等基本概念和应用。第 11 章介绍经典算法的设计和分析技术，主要包括递归与分治法、贪心算法和动态规划等的基本思想及应用。第 12 章介绍图算法，

主要包括基本的图算法、最小生成树和单源最短路径等。

本书由高飞设计总体架构。第1、2、3章由薛艳明编写，第4、5、6、7、8章由白霞编写，第9、10章由高飞编写，第11、12章由聂青编写。全书由高飞、薛艳明统稿、定稿。

感谢哈尔滨工业大学战德臣教授在计算思维研究中的贡献，正是他所领导的研究团队的研究成果给予了我们非同凡响的启发和影响，这些智慧的结晶对本书的编写具有重要意义的指导作用。感谢“理工类高校计算思维与大学计算机课程研究与教材建设”项目组所有参与学校的老师们对于本书的建议和帮助。感谢北京理工大学“计算机公共课教学团队”中从事教学工作多年的各位同仁提出的宝贵建议。感谢电子工业出版社对本书的编写给予热情支持。

由于计算机算法和程序设计技术发展迅速，作者水平有限，书中的疏漏与不足在所难免，敬请广大读者和同仁不吝赐教，拨冗指正。

欢迎读者对本书提出意见和建议。作者 E-mail：gaofei@bit.edu.cn。

本书为教师提供相关教学资源，有需要者，请登录华信教育资源网 http://www.hxedu.com.cn，注册之后进行下载。

作　者

目　录

上篇　高级语言程序设计基础

下篇 算法分析与设计

上 篇

高级语言
程序设计基础

第 1

高级语言

程序设计基础

第1章

C++语言概述

1.1 C++语言简介

本书采用 C++语言作为载体来讲授高级语言程序设计方法。计算机的工作是通过程序来控制的，程序是指令的集合，指令则是计算机可以识别的命令。由于计算机硬件只能识别由二进制指令构成的机器语言，所以在计算机诞生之初，只能使用机器指令编程，进而出现了将机器指令映射为助记符的汇编语言。1954 年诞生了用于科学计算的高级语言 FORTRAN，之后又先后出现了 BASIC、ALGOL、COBOL 和 C 等高级语言。高级语言屏蔽了机器的细节，提高了语言的抽象层次。采用高级语言编写的程序由具有一定含义的标识和容易理解的执行语句构成，为更多的人使用计算机提供了很大的方便。

在众多的计算机程序设计语言中，诞生于 20 世纪 70 年代的 C 语言在描述、开发系统软件和应用软件中具有重要地位。最初它是编写 UNIX 操作系统的工具，随着 UNIX 操作系统的广泛应用，C 语言迅速得到推广。后来经过不断完善改进，C 语言具有了功能丰富、表达能力强、使用灵活方便、应用范围广、目标程序效率高、可移植性高等特点。更重要的是，C 语言是介于高级语言与低级语言之间的“中间语言”，既具有高级语言的结构化和模块化的特点，又具有低级语言的控制性和灵活性的特点。因此，C 语言被列为程序设计课程的首选语言，深刻影响了几代计算机工作者。

然而，C 语言也存在不足。例如，与其他高级语言相比，其语法限制不严格。从应用的角度看，C 语言比其他高级语言较难掌握。而且，由于 C 语言是面向过程的结构化和模块化的程序设计语言，当处理的问题比较复杂、程序规模较大时，就显得力不从心。为了更好地开发大型软件，20 世纪 80 年代提出了面向对象的程序设计方法，因此需要设计出能够支持面向对象的程序设计方法的新语言。由于 C 语言应用广泛，深入人心，AT&T Bell 实验室的 Bjarne Stroustrup 等人在 C 的基础上开发了它的增强版，即 C++语言。C++语言在保留 C 语言所有优点的基础上，增加了适用于面向对象的程序设计的类类型，因此 C++语言是一种既支持面向过程程序设计又支持面向对象程序设计的混合型程序语言。C++语言是 C 语言的超集，与 C 语言兼容。用 C 语言编写的程序基本上可以不加修改就可以用于 C++程序。C++语言不仅在 C 语言的基础上增加了面向对象的机制，而且对 C 语言的功能也做了不少的扩充。

面向过程程序设计与面向对象程序设计是两种用途不同、互为补充的程序设计方法。面向对象程序中的成员函数就是用结构化程序设计方法实现的，任何程序的具体的操作过程都是面向过程的。计算机硬件直接操作的问题往往需要直接用面向过程的方法来解决。

C++语言是由 C 语言发展而来的，学习 C++程序设计方法，既要学会用 C/C++语言进行面向过程的结构化程序设计方法，也要学会用 C++语言进行面向对象的程序设计方法。

1.2 C++语言的基本组成

计算机程序设计语言与自然语言一样具有自己对字符、单词、特殊符号、语句、语法的使用规则。在C++语言中，所涉及的规定主要包括基本字符集、标识符、语句与标准函数库等。

1.2.1 基本字符集

C++的基本字符集包括以下几部分：

- ⊙ 大小英文字母：A～Z，a～z。
- ⊙ 数字字符：0～9。
- ⊙ 运算符：+ - * / % = < > <= >= != == << >> & | && || ^ ~ () [] . -> , ;。
- ⊙ 特殊字符：空格 _（下划线）及各种转义字符。

1.2.2 词法记号

C++的词法记号是通过其关键字和标识符体现的。

（1）关键字

C++预定义单词。用户定义的标识符不能和系统的关键字同名，以下是48个C++语言的标准关键字：

asm	auto	break	case	catch	char
class	const	continue	default	delete	do
double	else	enum	extern	float	for
friend	goto	if	inline	int	long
new	operator	private	protected	public	register
return	short	signed	sizeof	static	struct
switch	template	this	throw	try	typedef
union	unsigned	virtual	void	volatile	while

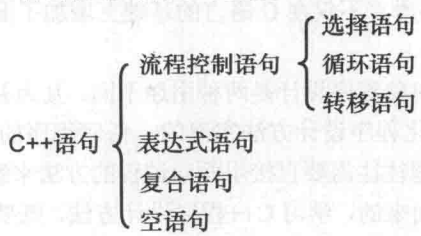
（2）标识符

标识符是程序员声明的单词，命名程序中的一些实体。

标识符的构成规则为，以大小写英文字母或下划线（_）开头，可以由大写英文字母、小写英文字母、下划线（_）或数字0～9组成。大写英文字母和小写英文字母表示不同的标识符。

1.2.3 语句

语句是组成程序的基本单位。所有程序设计语言都提供了满足编写程序要求的一系列语句，它们都有确定的形式和功能。C++语言的语句有以下几类：



1.2.4 标准函数库

C 语言程序中各种功能基本上都是由函数实现的。标准函数是由 C 编译系统提供的一些非常有用的功能函数。例如，C 语言没有输入、输出语句，也没有直接处理字符串语句，而一般的 C 语言编译系统提供了完成这些功能的函数。

调用函数库中的函数需要程序的开头包含某些头文件。头文件中声明了很多函数原型，如头文件“stdio.h”声明了 printf 函数原型。因此，在系统预定义函数 printf()时，必须在程序开始处加上**#include <stdio.h>**这样的预处理命令。有了包含命令，编译器将打开头文件“stdio.h”，并将其内容添加在该行位置替换原先的包含命令。

C 语言在发展的过程中建立了功能丰富的函数库，由于 C++语言是从 C 语言发展而来，因此，在 C++程序中可以使用 C 语言的函数库。在 C++程序中使用这些头文件有两种方法。

① 遵循 C 语言规定的方法。C 语言的头文件以 .h 为后缀，如 stdio.h、stdlib.h 等。由于 C 语言没有命名空间，头文件不可能存在于命名空间中，因此 C++程序中在用到以 .h 为后缀的头文件时，不必使用命名空间，只需像 C 语言程序那样，将头文件包含即可，如“#include <stdio.h>”。

② 用 C++语言的方法。按 C++语言标准要求，由系统提供的头文件不带后缀 .h，但要与“using namespace std;”配合使用。例如：

```
#include <iostream>
using namespace std;
```

为了与 C 语言的头文件既有联系又有区别，C++程序所用的头文件名是在 C 程序相应的头文件名（去掉后缀 .h）最前面加字符 c。

用户自己编制的头文件可以有后缀 .h。在 C++程序中，也可以使用 C 语言编译系统提供的带后缀 .h 的头文件，如“#include <math.h>”，此时不需使用“using namespace std;”语句。例如，C 语言中的 stdio.h 在 C++语言中为 cstdio；C 语言中的 math.h，在 C++语言中为 cmath。因为大多数 C++语言编译系统既提供了 C++语言的方法，又保留了 C 语言的方法，所以上述两种使用头文件的方法可以任选。

1.3 数 据 类 型

计算机处理的数据是以某种特定的形式存在的，如整数、浮点数、字符等形式。C++语言可以使用的数据类型如图 1.3.1 所示。C++语言没有统一规定各类数据的精度、数值范围和在内存中所占的字节数，计算机所能表示的实际数据范围根据编译器和计算机系统结构不同而不同。

整型数据以二进制形式存储，如十进制数 65 的二进制形式为 01000001，在内存中的存储形式如图 1.3.2 所示。整型数据和字符型数据都有带符号和无符号两种形式，分别由修饰符 signed 和 unsigned 表示。如果指定为 signed，则数值以补码形式存放，存储单元最高位表示数值的符号。如果指定为 unsigned，则数值没有符号，全部二进制都用来表示数值本身。

浮点数有单精度（float）、双精度（double）和长双精度（long double）之分。

1.3.1 常量

常量是指程序在运行时其值不能改变的量，C++语言中大量使用常量。常量包括数值常量和字符常量。数值常量包括整型常量和实型常量，字符常量中还包括字符串常量。

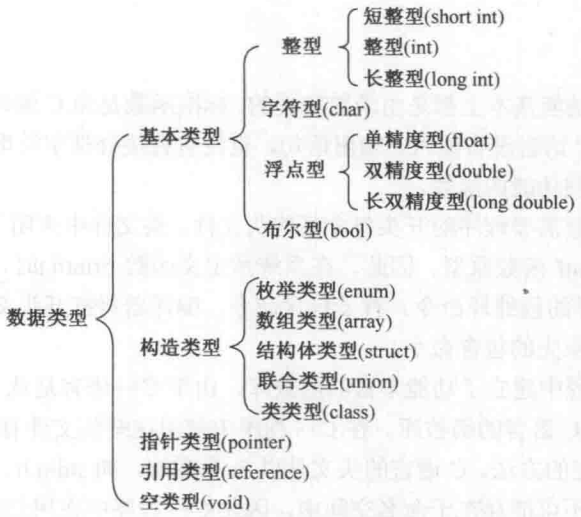


图 1.3.1　C++语言中的数据类型

0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

图 1.3.2　整型数据的存储形式

1. 数值常量

数值常量就是常数。在C++语言中，数值常量有类型之分，数值常量是直接以自身的存在形式体现其值和类型的。

（1）整型常量

整型常量也是直接以自身的存在形式体现其值和类型的。整型常量可以用以下 3 种表示方式。

① 十进制整数，如 123、-123、0 等。在一个整型常量后面加上字母 l 或 L，则认为它是 long int 型常量，如 123L、-123L、0L 等。

② 八进制整数。在常数开头加数字 0，表示这是以八进制整数形式表示的常数，如 020、030 等。

③ 十六进制整数。在常数开头加数字 0 和一个英文字母 X（或 x），表示这是以十六进制整数形式表示的常数，如 0X20、0x30 等。

（2）实型常量

一个实型数可以用十进制小数和指数两种形式表示。

① 十进制小数形式。一般由整数和小数两部分构成，如 12.345、-6.789 等。可以省略其中一部分，如 12.、.345 等，但两部分不能同时省略。

② 指数（浮点）形式。实型数可以表示成指数（浮点）形式。例如，3.1415926 可以写成 0.031415926×10^{2}、0.31415926×10^{1}、3.1415926×10^{0}、31.415926×10^{-1}、314.15926×10^{-2} 等形式。这种形式在 C++语言程序中应写成 0.31415926e1 或 314.15926e-2 等。归纳起来，实型数的指数（浮点）表示的一般形式为：

数字符号　数字部分　指数符号　指数部分

在程序中，不论把实型数表示成小数形式还是指数（浮点）形式，在计算机内存中都是以规格化浮点数形式存储的。所谓规格化的浮点数，就是数字部分必须是小于 1 的小数，小数点后面的第一位数字必须是非 0 的数字。存储单元分为两部分：一部分存放数字部分，另一部分存放指数部分。例如，0.31415926×101 的存储形式如图 1.3.3 所示。

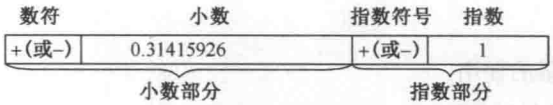


图 1.3.3　实型数在计算机中的存放形式

2. 字符常量

字符常量是括在一对单引号内的一个字符。例如，'a'、'B'、'%'、'?'、'6'、' '（表示空格字符）等都是字符常量。字符常量只能是一个字符，如'XY'是非法的。字符常量区分大小写字母，如'F'和'f'是两个不同的字符常量。这里的单引号仅起字符常量的边界符的作用，并不是字符常量的一部分。

除上述的字符常量外，对于常用的但难以用一般形式表示的不能显示的字符，C++语言提供了一种特殊的字符常量，即以转义标识符“\”开头的字符序列。常用的转义字符序列如表 1-3-1 所示。

例如，'\n'中的“n”已经不表示字符常量'n'了，由于其前面有转义标识符“\”，所以表示换行。这是一种“控制字符”，在屏幕上是不能显示的，在程序中也无法用一个一般形式的字符表示，所以只能用特殊形式来表示。'\015'则是'\ddd'形式的转义字符，其中“015”是八进制字符串，表示 ASCII 码为十进制 13 的字符，即回车。转义字符'\x18'是'\xdd'形式的转义字符，其中“18”是十六进制字符串，表示 ASCII 码编码为十进制 24 的字符，即↑。

表 1-3-1　转义字符序列及其功能

转义字符	功　能	ASCII 码
\0	空字符	0
\b	退格，将当前位置移到前一列	8
\t	水平跳格，跳到下一个 tab 位置	9
\n	换行，将当前位置移到下一行首	10
\f	换页，将当前位置移到下一页首	12
\r	回车	13
\"	双引号字符	34
\'	单引号字符	39
\\	反斜杠字符'\'	92
\0	空字符	0
\ddd	1～3 位八进制表示的字符	
\xhh	1～2 位十六进制表示的字符	

将一个字符常量存储在内存单元时，实际上并不是把该字符本身存到内存单元中，而是将该字符对应的 ASCII 码存储在内存单元中，一个字符占 1 字节。因为 ASCII 码是 0～255 之间的整数，所以 C++语言中的字符型数据和整型数据可以通用。例如，字符'C'的 ASCII 码值为 67，其二进制形式为 1000011，在计算机中的存储如图 1.3.4 所示。

3. 字符串常量

字符串常量是用一对双引号括起来的字符序列。这里的双引号仅起字符串常量的边界符的作用，它并不是字符串常量的一部分。例如，"ABC"、"a"、"x+y"、" "等都是合法的字符串。

注意，字符常量和字符串常量是完全不同的，如"c"和'c'是两种完全不同的数据。从书写形式上看，字符常量是用单引号括起来的单个字符，字符串常量是用双引号括起来的一个或一串字符。从存储方式上看，字符常量在内存中只占 1 字节，字符串常量除每个字符各占一个字符空间外，编译系统在字符串最后自动加一个'\0'作为结束标志，该结束标志也要占 1 字节的存储空间。'c'和"c"的存储情况如图 1.3.5 所示。

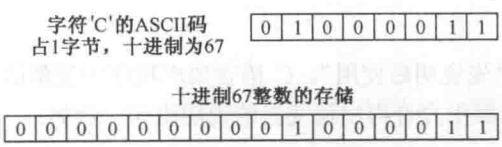


图 1.3.4　字符型数据的存储示意

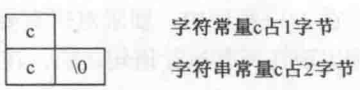


图 1.3.5　字符与字符串的存储情况

4. 符号常量

在 C++程序中，常量除了以自身的存在形式直接表示外，还可以用标识符表示常量，这就是符

号常量。

【例 1-3-1】 符号常量的使用。

```
#define  PI  3.1415926
#include <iostream>
using namespace std;
void main()
{
    float  r, c, s;
    cin >> r;
    c=2*PI*r;      s=PI*r*r;
    cout <<"c="<< c;
    cout <<"s="<< s << endl;
}
```

程序中使用预处理命令#define 指定 PI 在该程序中代表 3.1415926，此后在该程序中出现的所有 PI 都代表 3.1415926。在该程序中，符号常量 PI 与一般的常量一样参与运算，但不能被赋值。

在 C++语言中可以用常量说明符 const 代替#define 宏类定义常量。例如，“#define PI 3.1415926”可用如下方法替代：

```
const  PI = 3.1415926;
```

1.3.2 变量

变量是在程序运行期间其值可以发生变化的量。在程序设计语言中，可以把变量看成一个存储数据的容器，其功能是存储数据。对变量有两个基本的操作：一是向变量中存储数据，即赋值操作；二是取得变量的当前值，即取值操作。因为要对变量进行赋值和取值的操作，所以一个变量应该有一个名字，并在内存中占据一定的存储单元。注意，变量名和变量值是两个不同的概念，图 1.3.6 表明这两者是不同的。

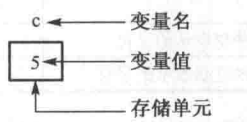


图 1.3.6 变量名与变量值的关系

变量名代表内存中的一个存储单元，在对程序编译连接时系统给每个变量分配一个地址。在程序中取得变量的值实际上是通过变量名找到相应的内存单元，从中读取数据的。而程序中的赋值操作则是通过变量名找到相应的内存单元，向其中存储数据。

（1）变量命名规则

变量名是变量的标识，其命名规则符合标识符的所有规定，即应该由大小写英文字母、数字和下划线构成，但不能以数字开头。以下是合法的变量名：

```
a1  m_1  _sum  r123  sum_12_3  stu_name
```

在 C++语言中，大写字母和小写字母被认为是两种不同的字符，因此 A1 和 a1 是两个不同的变量名。习惯上，人们把符号常量用大写字母表示，把变量名用小写字母表示。注意，变量名不能与 C++语言的关键字和系统函数名相同。

（2）变量的说明

在 C++程序中，要求对所有要用的变量必须“先说明后使用”。C 语言要求程序中变量的说明必须出现在所有执行语句之前，在正常执行语句后面不允许再出现变量的说明语句。例如：

```
int x, y;
x=20;                    //执行语句
int z;                   //出错!
```

然而，在 C++程序中，只要求在第一次使用之前进行说明即可，变量的说明可以出现在任何位