

21

世纪

高等院校规划教材

C++ 程序设计

强调程序设计方法和思路,
引入典型程序设计案例



注重程序设计实践环节, 培养程序设计项目开发技能

杜四春 银红霞 蔡立军 编著



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校规划教材

C++程序设计

杜四春 银红霞 蔡立军 编著

中国水利水电出版社

内 容 提 要

本书全面系统地讲述了 C++ 语言的基本概念、语法和面向对象的编程方法,对 C++ 语言面向对象的基本特征:类和对象、继承性、派生类、多态性和虚函数等内容作了详尽的介绍。本书具有丰富的例题,每章后面备有形式多样的练习题。内容安排上循序渐进、深入浅出,力求通俗易懂、突出重点、侧重应用。

本书不仅可作为高等院校和培训机构 C++ 程序设计教材,也可作为 C++ 语言自学者的自学指导书和计算机工程技术人员的参考书。

图书在版编目(CIP)数据

C++程序设计/杜四春,银红霞,蔡立军编著. —北京:中国水利水电出版社, 2005

(21 世纪高等院校规划教材)

ISBN 7-5084-2311-9

I. C… II. ①杜…②银…③蔡… III. C 语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 076483 号

书 名	C++程序设计
作 者	杜四春 银红霞 蔡立军 编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机) 68331835 (营销中心) 82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 22.5 印张 549 千字
版 次	2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷
印 数	0001—5000 册
定 价	32.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

前 言

C++语言是目前使用最为广泛的一种高效程序设计语言,是从C语言发展演变而来的,是C语言的超集,它既可以进行过程化程序设计,也可以用于面向对象的程序设计。实现了类的封装、数据隐藏、继承及多态性,使得其代码容易维护且高度可重用。

本书作为一本C++语言的入门教材,不仅详细地介绍了C++语言本身,而且深入地讲述了面向对象程序设计的方法。本书的主要特点是语言流畅,简洁易懂,例题丰富,实用性强。这使得读者不仅可以学会一门程序设计语言,还能初步掌握面向对象的程序设计方法。其中丰富的例题使得初学者可以在学习的同时就开始积累编程经验,以尽快达到学以致用目的。

本书内容包括:

第1章C++简单程序设计,主要介绍C++的发展历史,面向对象程序设计的概念,C++的词法与规则,C++程序的结构与实现,数据类型,常量和变量,运算符与表达式,流控制,数据的输入输出,C++语句,顺序、分支和循环程序设计。

第2章函数,主要介绍函数的定义与声明,函数调用,内联函数和函数的作用域等内容。

第3章类和对象,主要介绍类和对象,对象的初始化,构造函数与析构函数,堆与拷贝构造函数,局部类和嵌套类,引用,常类型等内容。

第4章数组与指针,主要介绍数组与指针的概念,指针运算,指针与数组,指针与函数,指针与字符串,动态内存分配和this指针等。

第5章继承和派生,主要介绍基类和派生类,单继承、多继承和虚基类。

第6章静态成员与友元,主要介绍静态数据成员和静态成员函数,友元函数和友元类。

第7章运算符重载,主要介绍单目和双目运算符重载,包括赋值运算符重载、下标运算符重载、比较运算符重载、运算符new与delete重载、逗号运算符重载、类型转换运算符重载、->运算符的重载、函数调用运算符重载和重载I/O运算符。

第8章模板,主要介绍模板的概念,函数模板和类模板。

第9章多态性与虚函数,主要介绍普通成员函数重载,构造函数重载,派生类指针,虚函数,纯虚函数和抽象类等。

第10章流类库与输入输出,主要介绍I/O标准流类,键盘输入和屏幕输出,磁盘文件的输入和输出,字符串流等内容。

第11章异常处理,主要介绍异常的概念,基本原理,异常处理机制,异常处理方法和多路捕获。

第12章Visual C++环境下Windows程序开发概述,主要介绍Visual C++环境,Windows和MFC编程基础以及如何使用Visual C++开发Windows程序。

本书中所有例题都在 Visual C++6.0 环境下运行通过,在其他版本的编译系统下一般也都可以运行。本书作为高等院校的 C++程序设计本科教材,建议教授课时为 48 课时,上机实践课时为 24 课时,课程设计课时为 16 课时。各院校可根据教学实际情况适当增删。

本书也可作为大中专院校的程序设计课程教材和各类培训机构培训教材,还可供从事计算机应用的工程和技术人员参考。

本书编写过程中,编者参阅了许多 C++的参考书和有关资料,并阅读了一些翻译的书籍,现谨向这些书的作者和译者表示衷心的感谢。

本书由蔡立军统稿。湖南大学计算机与通信学院银红霞编写 1~5 章,杜四春编写 6~12 章。由于编者水平有限,书中不妥或错误之处在所难免,恳请专家和广大读者批评指正。

编者

2004 年 10 月于岳麓山

目 录

前言

第 1 章 C++简单程序设计	1
1.1 C++语言概述	1
1.1.1 从 C 到 C++	1
1.1.2 C++面向对象的特性	1
1.1.3 C++的词法与规则	2
1.1.4 C++程序结构的组成	4
1.1.5 C++程序的开发步骤	4
1.2 基本数据类型和表达式	5
1.2.1 数据类型概述	5
1.2.2 常量	6
1.2.3 变量	10
1.3 运算符与表达式	11
1.3.1 运算符	12
1.3.2 表达式	15
1.3.3 数据类型转换	16
1.4 程序的基本控制结构	17
1.4.1 顺序结构	17
1.4.2 选择结构	19
1.4.3 循环结构	25
1.4.4 转移语句	29
习题	31
第 2 章 函数	33
2.1 函数的定义与函数的声明	33
2.1.1 函数的定义	33
2.1.2 函数的声明	35
2.2 函数的调用	37
2.2.1 函数调用的格式	37
2.2.2 函数调用的过程	38
2.2.3 函数调用时的参数传递	38
2.2.4 带默认形参值的函数	40
2.2.5 函数的递归调用	43

2.2.6 函数的嵌套调用	48
2.3 内联函数	48
2.4 存储类型说明与作用域	51
2.4.1 存储类型说明	51
2.4.2 作用域的分类	53
2.4.3 域运算符	58
习题	59
第3章 类和对象	60
3.1 类和类的定义	60
3.1.1 类的定义	60
3.1.2 类的成员函数	63
3.1.3 类和结构	64
3.2 对象和对象的定义	66
3.2.1 对象的说明	66
3.2.2 对象的生存期	68
3.2.3 类作用域	69
3.3 构造函数与析构函数	70
3.3.1 构造函数	70
3.3.2 析构函数	73
3.3.3 缺省构造函数和缺省析构函数	75
3.3.4 带参数的构造函数	75
3.4 拷贝构造函数	76
3.5 局部类和嵌套类	81
3.6 引用	83
3.6.1 引用的概念	83
3.6.2 引用作为函数参数	84
3.6.3 对象引用作函数参数	86
3.6.4 引用返回值	86
3.7 常类型	88
3.7.1 const 和 volatile	88
3.7.2 常引用	88
3.7.3 常对象	89
3.7.4 常对象成员	90
习题	94
第4章 数组与指针	97
4.1 数组	97
4.1.1 数组的定义及初始化	97

4.1.2	二维数组	101
4.1.3	对象数组	104
4.1.4	字符串处理	105
4.2	指针	107
4.2.1	指针的定义和使用	107
4.2.2	指针运算	118
4.3	指针与字符串	122
4.3.1	通过指针访问字符	123
4.3.2	字符数组与字符指针的异同	123
4.3.3	字符指针作为函数参数	124
4.4	指针与数组	124
4.4.1	一维数组的指针表示法	124
4.4.2	二维数组的指针表示法	130
4.4.3	指针数组	131
4.5	指针与函数	135
4.5.1	函数指针	135
4.5.2	指针作为函数参数	136
4.5.3	指针型函数	143
4.5.4	main 函数中的指针参数	145
4.6	void 指针与 const 指针	147
4.6.1	void 指针	147
4.6.2	const 指针	149
4.7	类成员指针	149
4.7.1	类数据成员指针	149
4.7.2	类成员函数指针	150
4.8	this 指针	151
4.9	动态内存分配	153
4.9.1	new	153
4.9.2	delete	155
习题		159
第 5 章	继承和派生	160
5.1	基类和派生类	160
5.1.1	派生类的定义格式	160
5.1.2	派生类的三种继承方式	162
5.1.3	访问控制	165
5.1.4	基类和派生类的关系	171
5.2	单继承	171

5.2.1	成员访问权控制	171
5.2.2	派生与构造函数、析构函数	173
5.2.3	继承中构造函数的调用顺序	178
5.2.4	组合	180
5.2.5	子类型和类型适应	181
5.3	多继承	181
5.3.1	多继承的概念	181
5.3.2	多继承的构造函数	183
5.3.3	二义性和支配原则	187
5.3.4	赋值兼容规则	191
5.4	虚基类	192
5.4.1	虚基类的引入	192
5.4.2	虚基类的构造函数	197
	习题	199
第 6 章	静态成员与友元	201
6.1	静态成员	201
6.1.1	静态数据成员	201
6.1.2	静态成员函数	203
6.2	友元函数	205
6.2.1	友元函数的说明	206
6.2.2	友元函数的使用	208
6.3	友元类	209
	习题	210
第 7 章	运算符重载	214
7.1	运算符重载概述	214
7.2	运算符重载的实现	215
7.3	一元运算符重载	219
7.4	二元运算符重载	221
7.5	特殊运算符重载	224
7.5.1	赋值运算符重载	224
7.5.2	下标运算符重载	227
7.5.3	比较运算符重载	228
7.5.4	new 与 delete 运算符重载	228
7.5.5	逗号运算符重载	230
7.5.6	类型转换运算符重载	231
7.5.7	->运算符重载	232
7.5.8	函数调用运算符重载	232

7.5.9	I/O 运算符重载.....	233
习题		234
第 8 章	模板	237
8.1	模板的概念.....	237
8.2	函数模板.....	237
8.2.1	函数模板说明.....	237
8.2.2	使用函数模板.....	238
8.2.3	重载模板函数.....	239
8.3	类模板.....	239
8.3.1	类模板说明.....	240
8.3.2	使用类模板.....	242
8.3.3	类模板的友元.....	246
8.3.4	标准类模板类库.....	248
习题		249
第 9 章	多态性与虚函数	251
9.1	多态性.....	251
9.1.1	普通成员函数重载.....	251
9.1.2	构造函数重载.....	255
9.1.3	派生类指针.....	258
9.2	虚函数.....	262
9.2.1	静态联编与动态联编.....	262
9.2.2	虚函数的概念.....	263
9.2.3	动态联编与虚函数.....	271
9.2.4	虚函数的限制.....	273
9.2.5	虚函数与重载函数的比较.....	274
9.3	纯虚函数和抽象类.....	274
9.3.1	纯虚函数.....	274
9.3.2	抽象类.....	276
9.3.3	虚析构造函数.....	280
习题		281
第 10 章	流类库与输入输出	284
10.1	I/O 标准流类.....	284
10.1.1	预定义流.....	284
10.1.2	C++中的流类库.....	286
10.2	格式化 I/O.....	286
10.2.1	ios 类中的枚举常量.....	286
10.2.2	使用 ios 成员函数.....	287

10.2.3	使用 I/O 操作符	290
10.2.4	检测流操作的错误	292
10.3	文件流	292
10.3.1	文件的概念	292
10.3.2	文件的打开与关闭	294
10.3.3	文件的读写	297
10.4	字符串流	303
10.4.1	字符串流概述	303
10.4.2	istream 类的构造函数	304
10.4.3	ostream 类的构造函数	305
习题		306
第 11 章	异常处理	309
11.1	异常处理概述	309
11.2	异常处理的基本思想	309
11.3	异常处理的实现	310
11.3.1	异常处理的语法	310
11.3.2	异常处理机制	313
11.4	标准 C++ 库中的异常类	318
11.5	多路捕获	318
11.6	含有异常的程序设计	321
11.6.1	何时避免异常	321
11.6.2	异常的典型使用	322
习题		323
第 12 章	Visual C++ 环境下 Windows 程序开发概述	326
12.1	Visual C++ 可视化集成开发环境	326
12.1.1	Visual C++ 简介	326
12.1.2	项目开发过程	327
12.1.3	集成开发环境 Developer Studio	328
12.1.4	常用功能键及其意义	329
12.2	创建、组织文件、工程和工作区	330
12.2.1	新建工程	331
12.2.2	新建工作区	331
12.2.3	增加已有文件到工程中	332
12.2.4	打开工作区	332
12.2.5	设置当前工程	332
12.3	Windows 编程基础	332
12.3.1	Windows API	332

12.3.2	Windows 基础.....	332
12.3.3	Windows 消息映射和处理.....	334
12.4	Win32 开发	335
12.4.1	抢先式多任务和多线程	336
12.4.2	连续的地址空间和先进的内存管理	336
12.4.3	内存映射文件	337
12.4.4	Win32s: Windows 3.x 对 Win32 API 的支持	338
12.4.5	Win32 编程基础	338
12.5	MFC 编程.....	339
12.5.1	MFC 类库概念和组成.....	339
12.5.2	MFC 应用程序框架.....	343
12.5.3	MFC 的优点.....	344
12.5.4	MFC 对消息的管理.....	344
参考文献		348

第 1 章 C++简单程序设计

1.1 C++语言概述

C++是一门含 C 语言子集的高效程序设计语言。它比 C 语言更容易为人们学习和掌握，并且以其独特的语言机制在计算机科学领域得到了广泛的应用。它既可以进行过程化程序设计，也可用于面向对象的程序设计。

1.1.1 从 C 到 C++

C++语言源于 C 语言，而 C 语言是在 B 语言的基础上发展起来的。1960 年出现了一种面向问题的高级语言 ALGOL 60，但它离硬件比较远，不宜用来编写系统软件。1963 年英国剑桥大学推出了 CPL (Combined Programming Language) 语言，后来经简化为 BCPL 语言。1970 年美国贝尔 (Bell) 实验室的 K.Thompson 以 BCPL 语言为基础，设计了一种类似于 BCPL 的语言，取其第一字母 B，称为 B 语言。1972 年美国贝尔实验室的 Dennis M.Ritchie 为克服 B 语言的诸多不足，在 B 语言的基础上重新设计了一种语言，取其第二字母 C，故称为 C 语言。设计 C 语言的最初目的是编写操作系统。由于其简单、灵活的特点，C 语言很快就被用于编写各种不同类型的程序，从而成为世界上最流行的语言之一。但是，C 语言是一个面向过程的语言。随着软件开发技术的进步，程序员们最终发现，把数据和施加在数据上的操作结合起来，会使程序更易于理解，可读性更好，由此产生了面向对象的程序设计思想。

1980 年贝尔实验室的 Bjarne Stroustrup 对 C 语言进行了扩充，推出了“带类的 C”，多次修改后起名为 C++。以后又经过不断的改进，发展成为今天的 C++。C++改进了 C 的不足之处，支持面向对象的程序设计，在改进的同时保持了 C 的简洁性和高效性。

目前，C++越来越受到重视并已得到了广泛的应用，成为在商业软件开发中占统治地位的语言。

1.1.2 C++面向对象的特性

1. C++支持数据封装

支持数据封装就是支持数据抽象。在 C++中，类是支持数据封装的工具，对象则是数据封装的实现。面向过程的程序设计方法与面向对象的程序设计方法在对待数据和函数关系上是不同的，在面向对象的程序设计中，将数据和对该数据进行合法操作的函数封装在一起作为一个类的定义，数据将被隐藏在封装体中，该封装体通过操作接口与外界交换信息。对象被说明为具有一个给定类的变量，类类似于 C 语言中的结构，在 C 语言中可以定义结构，但这种结构只包含数据，而不包含函数。C++中的类是数据和函数的封装体。在 C++中，结构可作为一种特殊的类，它虽然可以包含函数，但是它没有私有和保护的成员。

2. C++类中包含私有、公有和保护成员

C++类中可定义三种不同访问控制权限的成员。一种是私有（Private）成员，只有在类中说明的函数才能访问该类的私有成员，而在该类外的函数不可以访问私有成员；另一种是公有（Public）成员，类外面也可访问公有成员，成为该类的接口；还有一种是保护（Protected）成员，这种成员只有该类和其派生类可以访问，其余的在这个类外不能访问。

3. C++中通过发送消息来处理对象

C++中是通过向对象发送消息来处理对象的，每个对象根据所接收到的消息的性质来决定需要采取的行动，以响应这个消息。响应这些消息是一系列的方法，方法是在类定义中使用函数来定义的，使用一种类似于函数调用的机制把消息发送到一个对象上。

4. C++中允许友元破坏封装性

类中的私有成员一般是不允许该类外面的任何函数访问的，但是友元便可打破这条禁令，它可以访问该类的私有成员（包含数据成员和成员函数）。友元可以是在类外定义的函数，也可以是在类外定义的整个类，前者称友元函数，后者称为友元类。友元打破了类的封装性，它是C++另一个面向对象的重要性。

5. C++允许函数名和运算符重载

C++允许一个相同的函数名或运算符代表多个不同实现的函数，这就称函数或运算符的重载，用户可以根据需要定义函数重载或运算符重载。

6. C++支持继承性

C++中可以允许单继承和多继承。一个类可以根据需要生成派生类。派生类继承了基类的所有方法，另外派生类自身还可以定义所需要的不包含在父类中的新方法。一个子类的每个对象包含有从父类那里继承来的数据成员以及自己所特有的数据成员。

7. C++支持动态联编

C++中可以定义虚函数，通过定义虚函数来支持动态联编。

以上是所讲的是C++对面向对象程序设计的一些主要特征的支持。

1.1.3 C++的词法与规则

1. C++的字符集

C++中含有以下字符：

- 数字：0, 1, 2, 3, 4, 5, 6, 7, 8, 9。
- 小写字母：a, b, ..., y, z。
- 大写字母：A, B, ..., Y, Z。
- 运算符：+, -, *, /, %, <, <=, =, >=, >, !=, ==, <<, >>, &, |, &&, ||, ^, ~, (), [], {}, ->, *, !, ?, ?:, ,, ;, ", #。
- 特殊字符：连字符和下划线。
- 不可印出字符：空白格（包括空格、换行和制表符）。

2. 词与词法规则

(1) 标识符。

标识符是对实体定义的一种定义符，由字母或下划线（或连字符）开头、后面跟字母或数字或下划线组成的字符序列，一般有效长度是8个字符（而ANSI C标准规定31个字符），

用来标识用户定义的常量名、变量名、函数名、文件名、数组名、数据类型名和程序等。

(2) 关键字。

关键字，也称为保留字，是具有特定含义，作为专用定义符的单词，不允许另作它用。下面列举一些常用的关键字，如表 1-1 所示。

表 1-1 C++的标准保留字

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	overload	private	protected	public
register	return	short	signed	sizeof	static	struct	switch	this	template
throw	try	typedef	union	unsigned	virtual	void	volatile	while	

(3) 运算符和分隔符。

运算符是 C++语言实现加、减等各种运算的符号。

C++语言的分隔符主要是：空格、制表和换行符。

(4) 字符串。

字符串是由双引号括起来的字符。如"China", "C++ Program"等。

(5) 常量。

C++语言中常量包括实型常量（浮点常量）和整型常量（十进制常量、八进制常量、十六进制常量）、浮点常量、字符常量和字符串常量。

(6) 注释。

注释用来帮助阅读、理解及维护程序。在编译时，注释部分被忽略，不产生目标代码。C++语言提供两种注释方式。一种是与 C 兼容的多行注释，用/*和*/分界。另一种是单行注释，以“//”开头表明本行中“//”符号后的内容是注释，如下程序：

例 1-1：一个简单的 C++程序。

```
#include <iostream.h>
void main()
{
    cout<<"This is my first C++ program.\n";    //输出 This is my first
                                                //C++ program.
    /*输出
    This is my first C++ program.*/
}
```

3. 书写格式

C++语言程序的书写格式自由度高，灵活性强，随意性大，如一行内可写一条语句，也可写几条语句；一个语句也可分写在多行内。不过应采用适当的格式书写，便于人们阅读和理解。

为了增加程序的可读性和利于理解，编写程序时按如下要点书写：

(1) 一般情况下每个语句占用一行。

(2) 不同结构层次的语句，从不同的起始位置开始，即在同一结构层次中的语句，缩进同样的字数。

(3) 表示结构层次的大括弧，写在该结构化语句第一个字母的下方，与结构化语句对齐，

并占用一行。

(4) 适当加些空格和空行。

1.1.4 C++程序结构的组成

C++程序结构的基本组成包括以下几个部分：

(1) 预处理命令，C++提供了三类预处理命令：宏定义命令、文件包含命令和条件编译命令。

(2) 输入输出，C++程序中总是少不了输入和输出语句，实现与程序内部的信息交流。特别是屏幕输出的功能，几乎每个程序都要用到，使用它把计算机的结果显示在屏幕上。

(3) 函数，C++的程序是由若干个文件组成的，每个文件又是由若干个函数组成，因此，可以认为 C++的程序就是函数串，即由若干个函数组成，函数与函数之间是相对的，并且是并行的，函数之间可以调用。在组成一个程序的若干个函数中，必须有一个 main() 函数。

(4) 语句，语句是组成程序的基本单元。函数是由若干条语句组成的。但是，空函数是没有语句的。语句由单词组成，单词间用空格符分隔，C++程序中的语句又是以分号结束的。语句除了有表达式语句和空语句之外，还有复合语句、分支语句、循环语句和转向语句等若干类。

(5) 变量，多数程序都需要说明和使用变量。广义讲，对象包含了变量，即将变量也称为一种对象，狭义讲，将对象看作是类的实例，对象是指某个类的对象。

(6) 其他，除了以上讲述的 5 个部分以外，还有其他组成部分。例如，符号常量和注释信息也是程序的一部分。C++中都尽量把常量定义为符号常量，在 C++的程序中出现的是符号常量，该符号常量代表着某个确定的常量值。

下面用一个简单的程序例子来分析 C++程序的构成。

例 1-2: C++程序构成示例。

```
/*-----C++程序构成示例----- */           //注释
#define PI 3.1415926                             //宏定义命令，定义符号常量 PI
#include <iostream.h>                             //文件包含命令
void main()                                       //主函数
{
    double length,area,radius;                 //定义变量
    cout<<"Please input radius:"<<endl;
    cin>>radius;                                //输入语句
    length=2*PI*radius;
    area=PI*radius*radius;
    cout<<"length="<<length<<endl              //输出语句
        <<"area="<<area<<endl;
}
```

1.1.5 C++程序的开发步骤

C++语言是一种高级程序设计语言，它的开发过程与其他高级语言程序开发过程类似，一般要经过编辑、编译、链接、执行四个步骤。

1. 编辑

是指把按照 C++语法规则编写的程序代码通过编辑器 (Borland C++ 5.05, Visual C++ 6.0,

Turbo C++ 3.0) 输入计算机, 并保存。在保存时, C++源文件的扩展名为.CPP。

2. 编译

将编辑好的 C++源程序通过编译器转换为目标文件 (OBJ 文件)。即生成该源文件的目标代码。

3. 链接

将用户程序生成的多个目标代码文件 (.obj) 和系统提供的库文件 (.lib) 中的某些代码连接在一起, 生成一个可执行文件 (.exe)。

4. 执行

把生成的可执行文件运行, 在屏幕上显示运行结果。用户可以根据运行结果来判断程序是否出错。

1.2 基本数据类型和表达式

数据类型是指定义了一组数据以及定义在这一组数据上的操作, 它是程序中最基本的元素。程序的基本功能是处理数据, 而数据是以变量或常量的形式存储, 每个变量或常量都有数据类型。确定了数据类型, 才能确定变量的空间大小和其上的操作。C++数据类型十分丰富, 大体上可分为基本类型、空类型、构造类型、指针类型、类类型五种。如图 1-1 所示。

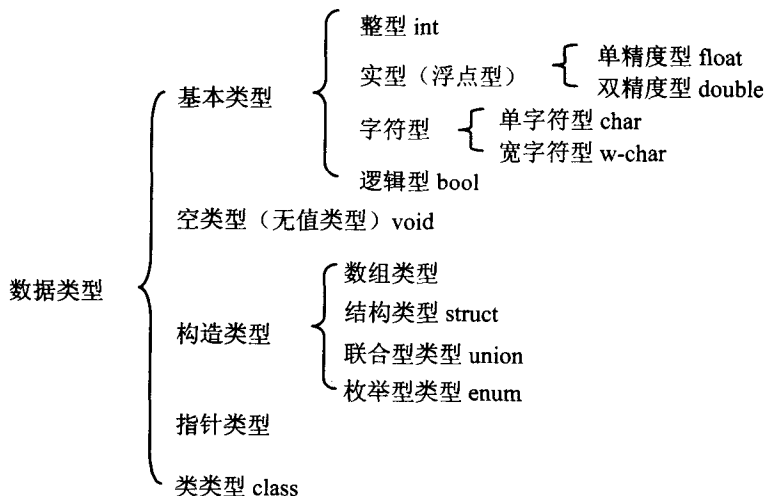


图 1-1 C++数据类型

1.2.1 数据类型概述

1. 基本数据类型

基本数据类型有 4 种: 整型 (int)、浮点型 (float)、字符型 (char)、逻辑型 (bool)。

整型数在计算机内部一般采用定点表示法, 用于存储整型量。(如 123, -7 等), 存储整数的位数依机器的不同而异。

浮点型和整型不同的地方是浮点数采用的是浮点表示法, 也就是说, 浮点数的小数点的