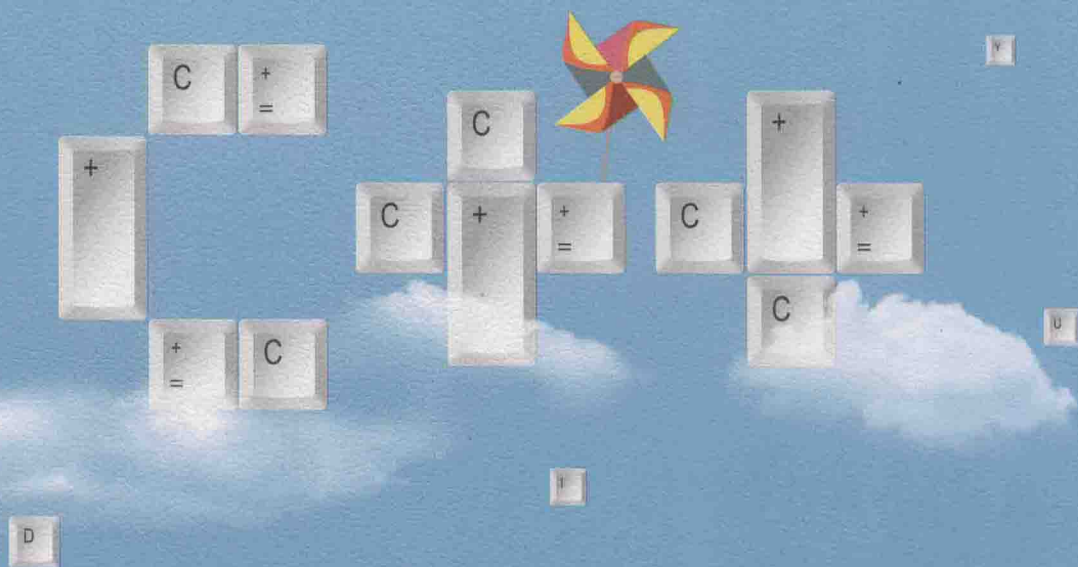




C++ 高级程序设计

(第二版)

田秀霞 主编

孙超超 张安勤 王真 彭源 副主编



基于C++标准蓝本，引用大量应用案例
组织上强调编程思维，重点或错误标注突出
语言通俗易懂，内容循序渐进

清华大学出版社



C++ 高级程序设计

(第二版)

田秀霞 主编

孙超超 张安勤 王真 彭源 副主编

清华大学出版社
北京

内 容 简 介

C++程序设计语言同时支持面向过程和面向对象编程,全书以C++标准为蓝本,以实际案例为支撑来阐释学生学习过程中晦涩难懂的概念,并将多年来教学过程中学生遇到的难以掌握的知识点以重点突出或错误解释的方式列出。整本书的语言通俗易懂,每个章节的开始都以一个短小有趣的引发好奇心的说明引导学生进入本章节内容的学习。整个内容按照循序渐进的方式展开,引领学生从C++的面向过程编程逐步过渡到C++的面向对象编程,实现学习点上的由易到难的顺利过渡。本书内容包括绪论、数据类型与基本输入输出、控制语句、函数与预处理、数组、指针与引用、自定义数据类型、类、静态成员与友元、继承与派生、多态性、输入输出流、模板、链表、异常处理和项目开发指南。

在书中每一章的最后,都有一个较为大型和完整的编程实例,涉及网络、密码学、计算机病毒、数据结构、算法设计、图形学等各门课程的一些入门知识,在巩固所学知识的同时,也让学生对这些课程有了初步的感性认识,以便后继课程的学习。

本书适用于高校计算机专业程序设计课程教学,也适用于自学成才的读者,在本书的指导下,读者可以逐步由不入门到初级程序员到高级程序员,并从中体会到学习编程的无限乐趣,为成为一名真正的程序员打下坚实的基础。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++高级程序设计/田秀霞主编.--2版.--北京:清华大学出版社,2016

ISBN 978-7-302-42980-7

I. ①C… II. ①田… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第030578号

责任编辑:付弘宇 王冰飞

封面设计:刘 键

责任校对:李建庄

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 北京嘉实印刷有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 37 字 数: 927千字

版 次: 2012年9月第1版 2016年7月第2版 印 次: 2016年7月第1次印刷

印 数: 1~2000

定 价: 69.80元

产品编号: 068063-01

前言

随着计算机技术的发展,程序设计已经成为许多理工学科研究中必不可少的一项基本技能。高级程序设计 C++ 课程是许多高校的计算机相关专业开设的一门专业基础课程,其他一些学科也将其纳入基础课的范畴。C++ 语言是一种面向对象的编程语言,该课程主要培养学生的动手编程能力和软件设计的逻辑思维能力,为后继课程的学习和工作打下基础。由于学生之前从未接触过程序设计的思想,从计算机程序设计语言的角度考虑问题存在着一定的障碍,同时这门课程内容多,对动手能力要求较高,因此学生在接受和掌握的过程中具有一定的难度。

本书以 C++ 标准为蓝本,以实际案例为支撑,来阐述和解释学生在学习过程中晦涩难懂的概念,并将多年来教学过程中学生遇到的难以掌握的知识点以重点突出或错误解释的方式列出。本书的语言通俗易懂,内容循序渐进,引领学生从 C++ 的面向过程编程逐步过渡到 C++ 的面向对象编程,实现学习上由易到难的顺利过渡。

本书的内容分为两大部分:第一部分讲授 C++ 语言基础,内容包括 C++ 及其编译环境、数据类型和表达式、程序设计基础、函数、指针和引用、自定义数据类型;第二部分讲授 C++ 的面向对象编程知识,内容包括类和对象、静态成员与友元、运算符重载、继承与派生、多态、输入流和输出流、模板、链表、异常处理等。

本书的特色如下:

1. 从熟悉的实例出发,引出抽象的概念

面向对象的程序设计方法虽然比面向过程的程序设计方法更为人性化,但对于学生而言,仍觉得十分抽象。本书对知识的描述大多是从具体的、熟悉的程序出发,先抛出问题和解决方法,然后逐步引入相应的知识点,以帮助学生更好地理解和掌握抽象的概念。

2. 强调编程思维

在书的组织上强调编程的思维,不过于强调知识点本身。针对问题,首先强调的是考虑问题的角度和解题的思路,这有助于培养学生分析问题、解决问题的能力。另外,学生在编程中经常会出现想当然的情况,书中根据作者多年来的教学经验,针对学生容易出现的问题,特别进行了讲解与提醒。通过分析出错的原因指出本质上的错误,以培养学生转变思路,从计算机理解程序的角度来考虑问题。

3. 大量的实例

大量的程序实例对学生来说非常必要,在学习语言的过程中语法规则比较重要,换个题目,学生可能就不知道该如何书写了,通过例题程序,学生可以方便地找出适合自己的程序结构,在学习的初期,实例这根“拐杖”是有必要的。本书中给出了大量的实例和课后习题供学生参考和练习,帮助学生熟悉和掌握所学的知识。

4. 与后继课程结合紧密

在书中每一章的最后都有一个较为大型和完整的编程实例,涉及网络、密码学、计算机病毒、数据结构、算法设计、图形学等各门课程的一些入门知识,在巩固所学知识的同时让学生对这些课程有了初步的感性认识,以便后继课程的学习。

本书由田秀霞老师负责组织和统稿,孙超超老师编写了第2、3、5、7、10~13章,张安勤老师编写了第4、6、14、15章,彭源老师编写了第1、8、9、16章,王真老师编写了第17章,在本书的初稿完成后,顾春华老师仔细阅读了书稿,提出了宝贵的修改建议,使得本书更加合理与完善;清华大学出版社对于本书给予了极大的关注和周到的编辑、出版安排,才使得本书在很短的时间内为广大初学者服务。

本书参考了大量的文献资料,但个别地方难免会有疏漏,敬请诸位专家、学者谅解。在此,我们向参考过的中外文献的作者表示诚挚的谢意,并对所有鼓励、支持和帮助过我们的领导、专家、朋友和读者表示真挚的感谢!

限于编者水平,书中疏漏与不当之处在所难免,敬请广大读者批评指正。

编 者

2016年5月

目 录

第 1 章 绪论	1
1.1 计算机程序设计语言	1
1.1.1 程序设计语言的演变	1
1.1.2 C++ 程序设计语言的发展	2
1.2 程序设计方法	2
1.2.1 面向过程程序设计	2
1.2.2 面向对象程序设计	3
1.3 计算机信息的存储与表示	4
1.3.1 数据在计算机中的表示	4
1.3.2 数据单位	5
1.3.3 不同形式数据的表示方法	6
1.4 C++ 程序结构与开发过程	7
1.4.1 C++ 程序的开发流程	7
1.4.2 第一个 C++ 程序	8
1.4.3 C++ 程序设计环境	11
1.4.4 带命令行参数的程序	16
1.5 C++ 的词和规则	18
1.5.1 C++ 的字符集	18
1.5.2 词与词法规则	18
1.6 编程实例——程序设计方法比较	19
习题 1	23
第 2 章 数据类型、表达式与基本输入输出	26
2.1 数据类型基本概念	27
2.1.1 C++ 数据类型的种类	27
2.1.2 整型和实型	28
2.1.3 数值的范围	28
2.1.4 有符号数和无符号数	30
2.1.5 有符号数和无符号数的范围区别	30
2.1.6 本章用到的关键字	31
2.2 C++ 基本数据类型	31
2.2.1 整型、实型和字符型	32
2.2.2 布尔型和无类型	33

2.2.3	为数据类型起别名	33
2.2.4	sizeof 的使用	34
2.3	变量	34
2.3.1	变量概述	34
2.3.2	变量的定义	34
2.3.3	赋值语句	35
2.3.4	变量的初始化	36
2.4	常量	38
2.4.1	整型常量	38
2.4.2	字符常量和字符串常量	39
2.4.3	逻辑常量	41
2.4.4	枚举常量	42
2.4.5	实型常量	44
2.4.6	地址常量	44
2.5	符号常量	45
2.5.1	为什么需要符号常量	45
2.5.2	const 与 #define 的比较	46
2.6	基本输入输出	46
2.6.1	cout 输出流	46
2.6.2	cin 输入流	47
2.7	运算符和表达式	49
2.7.1	C++ 语言运算符的种类	49
2.7.2	算术运算符	50
2.7.3	赋值运算符	53
2.7.4	关系运算符	56
2.7.5	逻辑运算符	57
2.7.6	位运算符	58
2.7.7	条件运算符	61
2.7.8	逗号运算符	62
2.7.9	运算符的优先级及结合性	63
2.8	编程实例——获取 IP 地址并计算网络号等信息	64
2.8.1	背景	65
2.8.2	源代码及程序解释	66
习题 2		69
第 3 章	语句和控制结构	74
3.1	顺序结构	75
3.1.1	表达式语句	75
3.1.2	函数调用语句	75
3.1.3	复合语句	76

3.1.4	空语句	76
3.1.5	赋值语句	76
3.2	选择结构	78
3.2.1	if...else 语句和 if 语句	78
3.2.2	多路分支	82
3.2.3	程序应用举例	91
3.3	循环结构	93
3.3.1	while 语句	93
3.3.2	do...while 语句	95
3.3.3	for 语句	97
3.3.4	3 种循环语句的选用	100
3.3.5	循环的嵌套	100
3.4	控制转移	103
3.4.1	break 语句	103
3.4.2	continue 语句	104
3.4.3	goto 语句	105
3.4.4	return 语句	106
3.5	编程实例——凯撒加/解密和维吉尼亚加/解密	106
3.5.1	背景	106
3.5.2	源代码及程序解释	106
	习题 3	111
第 4 章	函数	118
4.1	函数概述	118
4.2	预定义函数	119
4.3	自定义函数	121
4.3.1	函数的定义	121
4.3.2	函数的返回值	123
4.3.3	函数参数和传值调用	124
4.3.4	函数定义和函数声明的区别	126
4.4	函数的重载	127
4.5	带默认参数的函数	129
4.6	内联函数	131
4.7	函数的嵌套调用	133
4.8	函数的递归调用	134
4.9	局部变量与全局变量	136
4.10	变量的存储类别	139
4.10.1	自动变量	140
4.10.2	静态局部变量	141
4.10.3	寄存器变量	142

4.10.4	外部变量和静态外部变量	142
4.11	预处理命令	143
4.11.1	文件包含	143
4.11.2	条件编译	146
4.12	编程实例——Hanoi(汉诺)塔问题	148
习题 4		150
第 5 章	数组	153
5.1	一维数组	154
5.1.1	定义和引用一维数组	154
5.1.2	数组的初始化	156
5.1.3	数组下标越界	157
5.1.4	一维数组和循环	157
5.1.5	一维数组的应用	160
5.2	多维数组	162
5.2.1	定义和引用二维数组	163
5.2.2	二维数组的初始化	164
5.2.3	二维数组的应用	166
5.2.4	多维数组简单介绍	168
5.3	字符数组	169
5.3.1	定义和引用一维字符数组	169
5.3.2	一维字符数组的初始化	170
5.3.3	多维字符数组	171
5.3.4	字符数组的输入与输出	172
5.3.5	字符数组处理函数	173
5.4	string 类型	175
5.4.1	定义和初始化 string 类型变量	175
5.4.2	string 对象的使用	176
5.5	数组在函数中的应用	177
5.5.1	数组元素作为函数参数	177
5.5.2	整个数组作为函数参数	178
5.6	编程实例——大整数加减运算	180
5.6.1	设计思路	180
5.6.2	源代码和程序解释	180
习题 5		189
第 6 章	指针和引用	198
6.1	指针与指针变量	198
6.1.1	指针的概念	198
6.1.2	定义指针变量	201

6.1.3	引用指针变量	203
6.1.4	无名变量	205
6.1.5	指针作为函数参数	206
6.2	数组与指针	210
6.2.1	指向数组元素的指针	210
6.2.2	指针变量作为函数参数接收数组地址	214
6.3	字符串与指针	217
6.4	const 修饰指针	219
6.4.1	指向常量的指针	219
6.4.2	指针常量	220
6.5	引用	221
6.5.1	引用的基本概念	221
6.5.2	引用的应用	223
6.6	动态数组	227
6.7	其他指针	230
6.7.1	函数指针	230
6.7.2	指向指针的指针	231
6.7.3	指针数组	232
6.8	编程实例——选择法排序	234
	习题 6	235
第 7 章	自定义数据类型	239
7.1	结构体类型	240
7.1.1	结构体类型的声明	240
7.1.2	结构体变量的定义、初始化及使用	241
7.1.3	结构体数组的定义、初始化及使用	245
7.1.4	结构体作为函数参数	248
7.2	共用体	249
7.2.1	共用体变量的定义与初始化	250
7.2.2	共用体变量的使用	251
7.3	枚举类型	252
7.3.1	枚举类型变量的定义与初始化	252
7.3.2	枚举类型变量的使用	253
7.4	typedef 声明	254
7.5	编程实例——递归下降语法分析器	255
7.5.1	算法介绍	255
7.5.2	程序源代码	256
	习题 7	260

第 8 章 类	265
8.1 类的声明	266
8.1.1 类类型的声明	266
8.1.2 对象的定义	268
8.1.3 类的访问属性	269
8.1.4 取值函数和赋值函数	273
8.1.5 内联成员函数	276
8.1.6 this 指针	277
8.2 构造函数和析构函数	278
8.2.1 构造函数	279
8.2.2 拷贝构造函数	284
8.2.3 析构函数	289
8.3 组合对象	291
8.3.1 组合对象概述	291
8.3.2 构造函数和析构函数在组合对象中的使用	294
8.4 对象数组	297
8.4.1 对象数组的定义	298
8.4.2 对象数组的初始化	298
8.5 对象指针	299
8.5.1 对象指针的定义与使用	300
8.5.2 指向对象成员的指针	300
8.6 编程实例——学生成绩排名	301
8.6.1 算法介绍	302
8.6.2 程序源代码	303
习题 8	309
第 9 章 静态成员与友元	318
9.1 变量的使用	318
9.1.1 作用域	319
9.1.2 可见性	321
9.1.3 生存期	322
9.2 类的静态成员	323
9.2.1 静态数据成员	324
9.2.2 静态成员函数	325
9.3 类的友元	328
9.3.1 友元函数	328
9.3.2 友元类	331
9.4 const 修饰符	333
9.4.1 const 修饰变量	333

9.4.2	const 修饰引用	334
9.4.3	const 修饰对象	335
9.4.4	const 修饰类中的成员	335
9.5	编程实例——缓冲区溢出	339
9.5.1	背景	339
9.5.2	程序源代码	340
习题 9		342
第 10 章	运算符重载	352
10.1	重载二元运算符	352
10.1.1	成员函数实现运算符重载	355
10.1.2	友元函数实现运算符重载	356
10.2	重载一元运算符	357
10.3	>>和<<运算符重载	361
10.3.1	<<运算符重载	361
10.3.2	>>运算符重载	363
10.4	重载运算符限制与设计规则	365
10.5	编程实例——矩阵及向量间的运算	366
10.5.1	背景	366
10.5.2	程序源代码	367
习题 10		380
第 11 章	继承与派生	384
11.1	类的继承与派生	384
11.1.1	继承与派生的概念	385
11.1.2	派生类的声明	385
11.1.3	继承方式	388
11.1.4	派生类中的构造函数和析构函数	393
11.1.5	重定义成员函数	399
11.1.6	构造函数和析构函数的调用顺序	400
11.2	二义性问题	402
11.3	虚基类	405
11.3.1	虚基类的声明	406
11.3.2	虚基类的构造函数	407
11.4	编程实例——银行账户管理系统	409
11.4.1	需求分析与数据结构设计	409
11.4.2	程序源代码	410
习题 11		419

第 12 章 多态	426
12.1 upcast	426
12.2 后期绑定与虚函数	429
12.3 重新定义和重载	433
12.4 VPTR 和 VTABLE	434
12.5 关于运算效率的考虑	434
12.6 纯虚函数与抽象类	435
12.7 纯虚函数的定义	437
12.8 编程实例——虚函数实现求不同图形的面积	441
习题 12	444
第 13 章 输入输出流	450
13.1 流的概念	450
13.1.1 与 I/O 流类库有关的头文件	451
13.1.2 iostream 头文件中的重载运算符	452
13.2 输入输出格式控制	453
13.2.1 ios 类中的枚举常量	453
13.2.2 ios 类中的成员函数	455
13.2.3 iomanip 头文件中的控制符	457
13.3 字符 I/O	459
13.3.1 get 函数	460
13.3.2 put 函数	462
13.3.3 getline 函数	463
13.4 其他成员函数	465
13.4.1 ignore 函数	465
13.4.2 eof 函数	465
13.4.3 peek 函数	466
13.4.4 putback 函数	466
13.5 文件和文件 I/O	467
13.5.1 文件的概念	467
13.5.2 文件流类与文件流对象	468
13.5.3 文件的打开与关闭	469
13.5.4 I/O 对象的条件状态	473
13.5.5 ASCII 文件的访问	475
13.5.6 二进制文件的访问(选读)	476
13.6 编程实例——ini 文件的读/写	484
13.6.1 ini 文件结构	484
13.6.2 源代码和程序解释	485
习题 13	490

第 14 章 模板	496
14.1 函数模板	496
14.1.1 函数模板的定义	497
14.1.2 函数模板的应用	498
14.2 类模板	502
14.2.1 类模板的定义	502
14.2.2 类模板的应用	504
14.3 编程实例——复数类的定义和实现	506
习题 14	509
第 15 章 链表	512
15.1 链表概述	512
15.2 单链表类的定义	513
15.3 链表结点的访问	514
15.4 链表结点的插入	514
15.5 链表结点的删除	516
15.6 链表实例——简单的链表操作	518
习题 15	520
第 16 章 异常处理	522
16.1 C++ 错误处理	522
16.2 C++ 异常处理机制	523
16.2.1 异常处理机制的 3 个部分	523
16.2.2 异常处理的语法	525
16.2.3 自定义异常类	528
16.2.4 多个 throw 和 catch	530
16.2.5 catch(...)	533
16.2.6 异常规范	533
16.3 编程实例——病毒的自我复制	536
习题 16	539
第 17 章 项目开发指南	544
17.1 项目管理基础	544
17.1.1 项目与项目管理	544
17.1.2 项目流程	547
17.1.3 项目实施管理	550
17.2 项目开发团队	552
17.2.1 项目团队组成	552
17.2.2 项目团队要求	554

17.3 项目的研发	555
17.3.1 软件版本控制	555
17.3.2 程序编写规范	555
17.3.3 功能设计说明书的撰写	562
17.3.4 程序设计与实现	564
17.3.5 单元测试	570
17.3.6 提交代码	571
17.3.7 功能维护	572
习题 17	573

参考文献	575
------------	-----

第1章

绪论

1. 引言

C++语言是从C语言演变而来的一种以面向对象为特征的程序设计语言。本章首先从发展的角度概要介绍计算机程序设计语言的演变,然后对比了面向过程程序设计和面向对象程序设计的方法,目的是帮助读者了解面向对象编程和面向过程编程的基本特点,通过对比初步体会面向对象编程的优势所在。接着简要介绍了计算机信息存储与表示的方法,之后通过一个简单程序的引入及介绍C++程序的开发过程和集成开发环境使读者初步理解C++程序的基本结构、掌握基本编程流程及C++开发环境的简单使用,以便为后继章节的学习奠定基础。在本章的最后给出了C++语言的词和词法规则。

2. 学习目标

- (1) 了解面向过程程序设计和面向对象程序设计的过程和各自的特点。
- (2) 了解计算机中信息的存储与表示方法。
- (3) 掌握程序的开发流程。
- (4) 通过编写、运行简单程序理解程序的基本结构。
- (5) 熟练掌握C++集成开发环境的使用。

1.1 计算机程序设计语言

1.1.1 程序设计语言的演变

就像人类用语言来进行交流一样,程序设计语言是计算机与人进行交流的工具。然而计算机没有思维能力,它只能通过机器指令来实现指定的功能。

机器语言是具体的计算机上的机器指令集合,是最早的程序设计语言,也是完全面向机器的语言。机器指令是计算机能直接接收和识别的由0和1组成的指令代码。使用机器语言编写的程序能被计算机直接识别、运行速度快、占用的内存少,但难编、难读、难修改,而且机器语言因机种不同而有所不同,用机器语言编写的程序不具有通用性。

为了克服机器语言的缺点,计算机专家设计了符号化的机器语言,即汇编语言。

汇编语言用一些助记符号来表示机器指令,简单直观、便于记忆。用汇编语言编写的程序

比用机器语言编写的程序好写、好读、好修改。但计算机不能直接识别汇编语言,必须经过一个事先放在计算机内存中的翻译程序来翻译。汇编语言仍是面向机器的语言,通用性不强。

机器语言和汇编语言统称低级语言,它们的共同点就是面向机器,执行效率高,虽然现在因为CPU的规范化等原因当前机器上编译的程序可以拿到其他机器上去运行,但是难学、难记、难写、难检查的缺点仍旧无法去除,所以现在已经不是十分普及的语言了。

为了让计算机能够更方便人们使用、更普及,出现了面向人而不是面向机器的高级语言。高级语言(如C++)近乎人类的自然语言,易于学习和掌握,是面向过程的语言或面向对象的语言。用高级语言编写的程序具有易学、易懂、易修改的特点,同时具有很好的通用性,用高级语言编写的程序需要经过翻译才能被计算机识别执行。

1.1.2 C++程序设计语言的发展

C++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,但是它离硬件比较远,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),后来简化为BCPL。1970年,美国贝尔(Bell)实验室的K. Thompson以BCPL为基础设计了一种类似于BCPL的语言,取其第一个字母B命名,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,取其第二个字母C命名,故称为C语言。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”,经多次修改后取名为C++。以后又经过不断改进,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。

1.2 程序设计方法

1.2.1 面向过程程序设计

面向过程程序设计是一种以过程为中心的编程思想,就是从上往下步步求精,分析出解决问题所需要的一个个子任务,然后编写函数模块对这些子任务分别进行实现,并通过函数调用逐步实现整个功能,所以面向过程程序设计最重要的是模块化的思想方法,但是这种机制由于数据和处理这些数据的方法之间的分离导致其缺陷众多,具体表现如下。

1. 程序的开发效率低下

由于开发时需要编写大量的程序代码,并精心设计每段程序的算法,因此程序开发的工作量巨大,进而导致程序的开发效率低下。尤其是当程序规模较大、数据较多、函数关系非常复杂的时候,编写程序的过程将会变得异常困难。

2. 程序的可重用性差

由于程序和数据密切相关,因此当需要使用以前编写的某些子程序时需要精心检查和修改程序,否则很可能会出错,故这种编程方式下程序的可重用性很差。