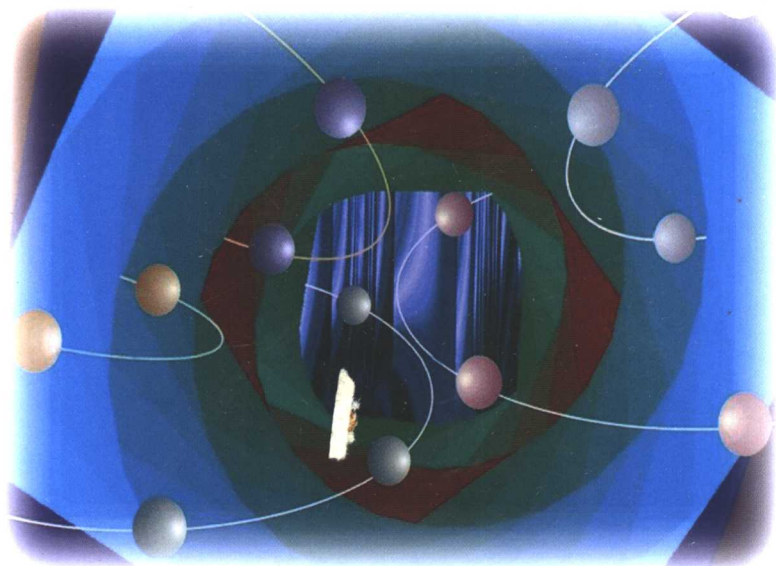




清华大学计算机基础教育课程系列教材

C++ 语言程序设计 案例教程

郑 莉 张瑞丰 李 莉 余小沛 编著



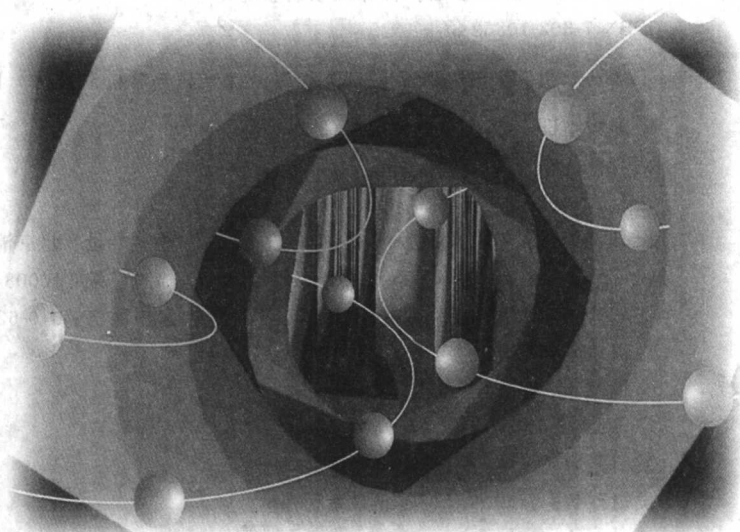
清华大学出版社



清华大学计算机基础教育课程系列教材

C++语言程序设计 案例教程

郑莉 张瑞丰 李莉 余小沛 编著



清华大学出版社
北京

内 容 简 介

本书基于 C++ 语言的基本语法设计了数百个经典语法案例,以综合性和实用性为基准设计了十几个大型综合案例。通过这些精心设计的案例深入浅出地介绍了 C++ 语言程序设计的特点,循序渐进地介绍了面向对象程序设计思想和方法。全书共分为 13 章,内容包括:C++ 程序设计和面向对象程序设计的概念,基础知识和函数,类与对象,数据的存储和访问属性,数组、指针与字符串,继承与派生,多态性,群体类和群体数据的组织,泛型程序设计与 STL,流类库与输入输出,异常处理和 Windows 程序设计特点。

本书语言精练,案例选择得当,语法讲解深入浅出。如果本书与《C++ 语言程序设计(第 3 版)》配套使用,将会得到更好的效果。该书不仅可以作为初学者学习 C++ 程序设计的入门教程,而且能够作为自学者的提高教程。

版权所有·翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C++ 语言程序设计案例教程 / 郑莉等编著. --北京:清华大学出版社,2005.7

(清华大学计算机基础教育课程系列教材)

ISBN 7-302-10852-8

I. C… II. 郑… 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(2005)第 036608 号

出版者:清华大学出版社

地 址:北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

客户服务:010-62776969

组稿编辑:张 龙

文稿编辑:霍志国

印刷者:北京四季青印刷厂

装订者:三河市春园印刷有限公司

发行者:新华书店总店北京发行所

开 本:185×260 印张:32.75 字数:771 千字

版 次:2005 年 7 月第 1 版 2005 年 7 月第 1 次印刷

书 号:ISBN 7-302-10852-8/TP·7216

印 数:1~4000

定 价:39.00 元

序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学。

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中,保持教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广：该系列教材照顾了理、工、文等各种类型专业的教材要求。

4. 立体配套：为适应教学模式、教学方法和手段的改革，该系列教材中多数都配有习题集和实验指导、多媒体电子教案，有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践，凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善，不断更新，为我国高校计算机基础教育做出新的贡献。



注：周远清，曾任教育部副部长，原清华大学副校长、计算机专业教授。

前 言

C++ 是从 C 语言发展演变而来的一种面向对象的程序设计语言。C++ 语言的主要特点表现在 3 个方面：一是全面兼容 C 语言支持面向过程的程序设计方法；二是支持面向对象的方法；三是支持范型程序设计方法。

本书是《C++ 语言程序设计(第 3 版)》系列教材的组成部分,旨在帮助读者通过实践学习和掌握 C++ 语言。本系列教材将 C++ 语言作为大学生的计算机编程入门语言,不仅详细介绍了语言本身,而且介绍了常用的数据结构和算法、面向对象的程序设计思想和方法、UML 建模语言、范型程序设计思想和方法。

本书以面向过程的程序设计为起点,以面向对象的程序设计为主线,以范型程序设计为补充。各章中的每节均包含知识点综述、疑难辨析和典型语案例析,每章最后则立足于本章的知识点设计出 1~3 个综合案例。知识点综述部分概要介绍本节涉及的主要知识点,疑难辨析部分将学生们经常遇到的疑点和难点加以详细解答和辨析,典型语案例析部分针对各个语法要点,精心设计出相应的语案例,帮助读者深入理解相关语法要点,每章最后的综合案例则集综合性、实用性于一身,结合本章主要知识点设计而成,帮助读者灵活掌握和使用本章的知识要点。本书的宗旨是不仅要使读者掌握 C++ 语言本身,而且能够对现实世界中较简单的问题和解决方案用 C++ 语言进行描述。当然,要能够描述较复杂的问题,还需要学习面向对象的软件工程课等其他课程。

本书的结构合理且实用、直观。全书的主要特色是知识点全面、系统;综合案例具有很好的启发性和引导性;通过案例对逐个知识点进行讲解,针对性强。另外,从第 4 章开始,综合案例中涉及类设计部分均给出相应的 UML 图形。

本书的章节安排基本上与《C++ 语言程序设计(第 3 版)》一致,既可以配合该书使用,也可以独立使用。

讲授 C++ 课程的教师可以将本书作为教学参考书,或用作案例课程教材。初学者通过本教程可以深刻地理解语法和基本概念,有一些编程基础的读者使用本书也有很好的参考价值。

作者郑莉在清华大学授课的教学资源(包括例题源代码、电子教案等)位于“清华大学网络学堂”,网址是: <http://learn.tsinghua.edu.cn>。请读者在“本学期课程”栏目中查找郑莉的“C++ 语言程序设计”(如果首页未列出,请单击“更多”链接),找到后单击课程名,然后以用户名“GUEST”、密码“guest”登录,即可阅读、下载教学资源。在作者不授课的学期,读者可以查找“以往课程”,下载学习资源。

本书由郑莉、张瑞丰、李莉、余小沛编写。参加本书编写工作的还有董渊、田荣牌、孟威、张文举、杨兴朋、王绚、邵毓华、孟鸿利等,在此深表感谢。

感谢读者选择使用本书,欢迎您对本书内容提出意见和建议,我们将不胜感激。作者的电子邮件地址: zhengli@mail. tsinghua. edu. cn, 来信标题请包含“C++ book”。

作 者

2005 年 5 月于清华大学

目 录

第 1 章 C++ 语言程序设计概述	1
1.1 程序设计语言	1
1.1.1 知识点综述	1
1.1.2 疑难辨析	1
1.2 面向过程与面向对象程序设计方法	3
1.2.1 知识点综述	3
1.2.2 疑难辨析	3
1.3 计算机信息存储与表示	3
1.3.1 知识点综述	3
1.3.2 疑难辨析	4
1.4 程序开发术语和过程	4
1.4.1 知识点综述	4
1.4.2 疑难辨析	5
1.5 案例分析	5
第 2 章 C++ 语言基础	8
2.1 C++ 程序概述	8
2.1.1 知识点综述	8
2.1.2 疑难辨析	9
2.1.3 典型语案例解析	9
2.2 基本数据类型和运算符	10
2.2.1 知识点综述	10
2.2.2 疑难辨析	12
2.2.3 典型语案例解析	13
2.3 流程控制	27
2.3.1 知识点综述	27
2.3.2 疑难辨析	28
2.3.3 典型语案例解析	28
2.4 自定义数据类型	43
2.4.1 知识点综述	43
2.4.2 疑难辨析	44
2.4.3 典型语案例解析	44

2.5	综合案例分析	50
第3章	函数	52
3.1	函数的声明与使用	52
3.1.1	知识点综述	52
3.1.2	疑难辨析	52
3.1.3	典型语法案例解析	53
3.2	函数的参数传递	58
3.2.1	知识点综述	58
3.2.2	疑难辨析	58
3.2.3	典型语法案例解析	58
3.3	函数的嵌套和递归调用	61
3.3.1	知识点综述	61
3.3.2	疑难辨析	61
3.3.3	典型语法案例解析	61
3.4	带默认形参值的函数	64
3.4.1	知识点综述	64
3.4.2	疑难辨析	64
3.4.3	典型语法案例解析	64
3.5	函数的重载	68
3.5.1	知识点综述	68
3.5.2	疑难辨析	68
3.5.3	典型语法案例解析	68
3.6	C++ 系统函数	73
3.6.1	知识点综述	73
3.6.2	疑难辨析	73
3.6.3	典型语法案例解析	73
3.7	综合案例分析	81
第4章	面向对象基本特点	84
4.1	面向对象基本特点	84
4.1.1	知识点综述	84
4.1.2	疑难辨析	84
4.1.3	典型语法案例解析	84
4.2	类和对象	85
4.2.1	知识点综述	85
4.2.2	疑难辨析	85
4.2.3	典型语法案例解析	86

4.3	构造函数和析构函数	89
4.3.1	知识点综述	89
4.3.2	疑难辨析	90
4.3.3	典型语法案例解析	90
4.4	类的组合	95
4.4.1	知识点综述	95
4.4.2	疑难辨析	95
4.4.3	典型语法案例解析	96
4.5	UML 图形标识	98
4.5.1	知识点综述	98
4.5.2	疑难辨析	99
4.5.3	典型语法案例解析	99
4.6	综合案例分析	99
第5章	C++ 语言存储与访问属性	102
5.1	作用域与可见性	102
5.1.1	知识点综述	102
5.1.2	疑难辨析	102
5.1.3	典型语法案例解析	102
5.2	静态生存期与动态生存期	106
5.2.1	知识点综述	106
5.2.2	疑难辨析	106
5.2.3	典型语法案例解析	106
5.3	静态数据成员和函数成员	110
5.3.1	知识点综述	110
5.3.2	疑难辨析	110
5.3.3	典型语法案例解析	110
5.4	友元函数和友元类	113
5.4.1	知识点综述	113
5.4.2	疑难辨析	113
5.4.3	典型语法案例解析	114
5.5	常量数据与共享数据的保护	120
5.5.1	知识点综述	120
5.5.2	疑难辨析	121
5.5.3	典型语法案例解析	121
5.6	编译预处理与多文件组织	125
5.6.1	知识点综述	125
5.6.2	疑难辨析	126

5.6.3 典型语案例解析	127
5.7 综合案例分析	136
第6章 数组、指针与字符串	141
6.1 数组	141
6.1.1 知识点综述	141
6.1.2 疑难辨析	141
6.1.3 典型语案例解析	142
6.2 指针和引用	155
6.2.1 知识点综述	155
6.2.2 疑难辨析	157
6.2.3 典型语案例解析	158
6.3 动态内存分配	181
6.3.1 知识点综述	181
6.3.2 疑难辨析	182
6.3.3 典型语案例解析	183
6.4 深拷贝和浅拷贝	189
6.4.1 知识点综述	189
6.4.2 疑难辨析	189
6.4.3 典型语案例解析	189
6.5 字符串	197
6.5.1 知识点综述	197
6.5.2 疑难辨析	197
6.5.3 典型语案例解析	197
6.6 综合案例分析	219
第7章 类的继承与派生	227
7.1 继承与派生	227
7.1.1 知识点综述	227
7.1.2 疑难辨析	227
7.1.3 典型语案例解析	228
7.2 构造函数和析构函数	233
7.2.1 知识点综述	233
7.2.2 疑难辨析	234
7.2.3 典型语案例解析	234
7.3 多级多继承中成员的访问	245
7.3.1 知识点综述	245
7.3.2 疑难辨析	246

7.3.3 典型语法案例解析	246
7.4 虚继承与虚基类	251
7.4.1 知识点综述	251
7.4.2 疑难辨析	251
7.4.3 典型语法案例解析	252
7.5 综合案例分析	256
第8章 多态性	263
8.1 多态性概述	263
8.1.1 知识点综述	263
8.1.2 疑难辨析	263
8.1.3 典型语法案例解析	265
8.2 成员函数和运算符重载	265
8.2.1 知识点综述	265
8.2.2 疑难辨析	266
8.2.3 典型语法案例解析	267
8.3 虚函数	281
8.3.1 知识点综述	281
8.3.2 疑难辨析	281
8.3.3 典型语法案例解析	282
8.4 纯虚函数与抽象类	289
8.4.1 知识点综述	289
8.4.2 疑难辨析	289
8.4.3 典型语法案例解析	289
8.5 综合案例分析	294
第9章 群体类和群体数据的组织	303
9.1 函数模板和类模板	303
9.1.1 知识点综述	303
9.1.2 疑难辨析	304
9.1.3 典型语法案例解析	305
9.2 简单群体类设计	312
9.2.1 知识点综述	312
9.2.2 疑难辨析	312
9.2.3 典型语法案例解析	313
9.3 综合案例分析	320

第 10 章 泛型程序设计与 C++ 标准模板库	323
10.1 命名空间、头文件命名和对组类相关概念	323
10.1.1 知识点综述	323
10.1.2 疑难辨析	323
10.1.3 典型语法案例解析	324
10.2 C++ 标准容器类	329
10.2.1 知识点综述	329
10.2.2 疑难辨析	330
10.2.3 典型语法案例解析	331
10.3 C++ 标准迭代器	340
10.3.1 知识点综述	340
10.3.2 疑难辨析	341
10.3.3 典型语法案例解析	342
10.4 C++ 标准通用算法	345
10.4.1 知识点综述	345
10.4.2 疑难辨析	349
10.4.3 典型语法案例解析	349
10.5 函数对象	386
10.5.1 知识点综述	386
10.5.2 疑难辨析	388
10.5.3 典型语法案例解析	388
10.6 综合案例分析	391
第 11 章 流类库与输入输出	398
11.1 C++ I/O 流概念	398
11.1.1 知识点综述	398
11.1.2 疑难辨析	399
11.1.3 典型语法案例解析	399
11.2 标准输入与输出	401
11.2.1 知识点综述	401
11.2.2 疑难辨析	402
11.2.3 典型语法案例解析	403
11.3 流的格式化控制与流操纵符	407
11.3.1 知识点综述	407
11.3.2 疑难辨析	407
11.3.3 典型语法案例解析	407
11.4 文件	421
11.4.1 知识点综述	421

11.4.2 疑难辨析·····	422
11.4.3 典型语法案例解析·····	422
11.5 综合案例分析·····	436
第 12 章 异常处理 ·····	447
12.1 异常处理基础·····	447
12.1.1 知识点综述·····	447
12.1.2 疑难辨析·····	447
12.1.3 典型语法案例解析·····	447
12.2 异常处理的实现·····	448
12.2.1 知识点综述·····	448
12.2.2 疑难辨析·····	449
12.2.3 典型语法案例解析·····	449
12.3 构造析构中的异常处理·····	456
12.3.1 知识点综述·····	456
12.3.2 疑难辨析·····	456
12.3.3 典型语法案例解析·····	457
12.4 标准程序库中的异常处理·····	460
12.4.1 知识点综述·····	460
12.4.2 疑难辨析·····	461
12.4.3 典型语法案例解析·····	461
12.5 综合案例分析·····	463
第 13 章 Windows 程序设计与 MFC 库 ·····	465
13.1 Windows 程序设计·····	465
13.1.1 知识点综述·····	465
13.1.2 典型案例解析·····	465
13.2 MFC 库及编程·····	474
13.2.1 知识点综述·····	474
13.3.2 典型案例解析·····	475
13.3 综合案例分析·····	487
参考文献 ·····	494
书中各章的案例索引 ·····	495

第 1 章

C++ 语言程序设计概述

1.1 程序设计语言

1.1.1 知识点综述

1. C++ 语言发展历史

(1) C++ 是从 C 语言发展演变而来的,首先是一个更好的 C。最初被称为“带类的 C”。

(2) 1983 年正式取名为 C++。

(3) 从 1989 年开始 C++ 语言的标准化工作。

(4) 于 1994 年制定了 ANSI C++ 标准草案。

(5) 于 1998 年 11 月被国际标准化组织(ISO)批准为国际标准。

2. 计算机语言

语言是一套具有语法、词法规则的系统。计算机语言是计算机可以识别的语言,用于描述解决问题的方法,供计算机阅读和执行。

3. 机器语言、汇编语言和高级语言

(1) 指令是计算机可以识别的命令。一台计算机硬件系统能够识别的所有指令的集合,称为它的指令系统。

(2) 由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。

(3) 汇编语言,将机器指令映射为一些可以被人读懂的助记符,如 ADD 和 SUB 等。

(4) 高级语言屏蔽了机器的细节,提高了语言的抽象层次,程序中可以采用具有一定含义的数据命名和容易理解的执行语句。

1.1.2 疑难辨析

(1) C++ 对 C 的继承和改进。

C++ 继承了 C,并支持面向对象的程序设计。除增加了类以外,C++ 还对 C 的其他方面做了扩充和更新:

C++ 使用 IO 流类库实现输入输出功能,C 语言的 IO 函数也仍然可以使用。

C++ 程序的源文件的扩展名为 cpp。C 语言源程序的扩展名为 c。

C++ 程序中调用某个函数之前需要声明函数的原型。

C++ 中函数的参数可以采用默认值。

C++ 增加了引用类型、布尔类型。

C++ 中常用内联函数来代替 #define 宏。

C++ 中变量和对象可以在需要的地方声明,不必全部写在程序开始的位置。

C++ 允许函数重载,C 不允许。

C++ 中用 new 操作代替 C 中 malloc() 函数,用 delete 操作代替 free() 函数。

C 中说明结构体变量需要带有关键字 struct。关键字是必不可少的。但是在 C++ 中,结构体名就是类型名,是一种自定义类型,可以省略关键字 struct 而只用结构体名来声明变量。对于枚举类型也是一样。

C++ 扩充了 C 的关键字。

在 C++ 中允许使用作用域运算符::。

在 C++ 中允许声明一个无参数的函数。

C++ 程序的主函数返回类型要求为 int,如果不写 return 语句,则自动返回 0。

(2) C++ 语言与 Java 和 C# 的差别?

C++ 是 Java 和 C# 的始祖。尽管 Java 和 C# 添加、删除和修改了许多功能,但总的来说,这 3 种语言的语法和面向对象模式十分相似。另外,3 种语言的面向对象编程风格也非常相似。如果掌握了 C++,那么很快就可以掌握 Java 和 C# 语言了。反之亦然。

C++,Java 和 C# 的主要区别在于三者运用的计算环境的类型不同。创建 C++ 是为特定类型的 CPU 和操作系统编写高性能的程序。Java 和 C# 的开发则是为了满足因特网的联机环境下特殊的编程需求。因特网连接许多不同类型的 CPU 和操作系统。因此,创建跨平台、可移植程序的能力就成为最重要的特征。第 1 种满足这种特性的语言是 Java 语言。Java 语言编写的程序可以在因特网上自由地传递。然而,Java 程序执行速度要比 C++ 慢得多,这也是可移植性好的代价。C# 也是如此。如果要创建一个高性能软件,建议使用 C++。如果需要创建一个可移植性高的软件,那么应该使用 Java 和 C#。归根结底应当注意的是,C++,Java 和 C# 是为解决不同类型问题而设计的。问题不在于哪种语言的好与坏,而在于哪种语言更加适合目前的环境。

(3) Java 和 C# 为何能够实现跨平台使用,而 C++ 不能?

Java 和 C# 可以创建跨平台、可移植的程序,而 C++ 不能,这是由于编译器生成的对象代码类型不同造成的。在 C++ 编译环境中,编译器输出的是机器代码,可以由 CPU 直接执行。因此,它受制于特定的 CPU 和操作系统。如果要在不同的系统上运行 C++ 程序,则需要当前系统下将程序重新编译成针对该环境的机器代码。

Java 和 C# 的跨平台、可移植程序是通过伪代码实现的,即以中间语言翻译来实现。在 Java 环境中,这种中间语言称为字节码。在 C# 中称为微软中间语言。伪代码都是由运行时系统执行的。对于 Java,这种运行时系统称为 Java 虚拟机(Java Virtual Machine,JVM)。C# 中称作公共语言运行时(Common Language Runtime,CLR)。因此,Java 程序可以运行在任何使用 JVM 的环境中,而 C# 程序可以运行在任何实现了 CLR 的环境中。因为 Java 和 C# 虚拟机的运行会产生一定的系统开销,而 C++ 程序无需这些开销,故 C++ 程序通常要比使用 Java 和 C# 编写的程序执行得快。

1.2 面向过程与面向对象程序设计方法

1.2.1 知识点综述

- 对象

面向对象方法中的对象是系统中用来描述客观事物的一个实体,它是用来构成系统的一个基本单位。对象由一组属性和一组行为构成。属性是用来描述对象静态特征的数据项,行为是用来描述对象动态特征的操作序列。

- 类

面向对象方法中的“类”是具有相同属性和服务的一组对象的集合。它为属于该类的全部对象提供了抽象的描述,其内部包括属性和行为两个主要部分。类与对象的关系犹如模具与铸件之间的关系,一个属于某类的对象称为该类的一个实例。

- 封装

封装是面向对象方法的一个重要原则,就是把对象的属性和服务结合成一个独立的系统单位,并尽可能隐蔽对象的内部细节。

- 继承

继承是面向对象技术能够提高软件开发效率的重要原因之一,其定义是:特殊类的对象拥有其一般类的全部属性与服务,称作特殊类对一般类的继承。

- 多态性

多态性是指在一般类中定义的属性或行为,被特殊类继承之后,可以具有不同的数据类型或表现出不同的行为。

1.2.2 疑难辨析

C++ 是一种面向对象的语言,结构化程序设计方法在 C++ 程序设计中是否仍然有用?

结构化程序设计方法始终是编写小规模程序的基础,C++ 程序中类的行为是以函数的形式抽象出来的,而编写每个函数时都要运用面向过程的结构化程序设计思想。

1.3 计算机信息存储与表示

1.3.1 知识点综述

- 计算机内部的信息可以分成两大类:计算机加工的对象是数据信息,而指挥计算机操作的是控制信息。
- 计算机的数字系统:几乎所有的计算机采用的都是二进制数系,所有的外界信息在被转化为不同的二进制数后,计算机才能对其进行传送、存储和加工处理。
- 信息的单位通常采用位、字节和字。

位(b):表示一位二进制信息。

字节(B):一个字节由 8 位二进制数字组成(1 B=8 b)。