

实例讲解
实训强化
培养技能
面向就业



国家高职示范院校核心课程教材

全国高等职业教育计算机类规划教材 · 实例与实训教程系列

C++开发实例教程

◎ 刘畅 主编

◎ 刘辉 刘苗苗 胡艳梅 副主编



- ◆ 知识结构合理，案例适用
- ◆ 项目驱动方式讲解C++开发的各知识点
- ◆ 所有程序运行结果采用截屏方式，程序更准确
- ◆ 配有教学计划，电子教案，所有实例、练习题和课后编程题的源代码



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

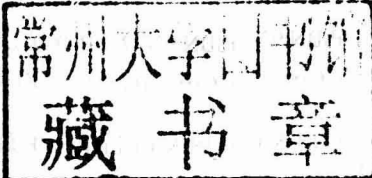
全国高等职业教育计算机类规划教材·实例与实训教程系列

国家高职示范院校核心课程教材

C++开发实例教程

刘 畅 主编

刘 辉 刘苗苗 胡艳梅 副主编



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统介绍了 C++面向对象程序设计的基本概念和编程方法,共分 12 章:第一至六章主要介绍了 C++的数据类型、程序结构化设计、函数、C++的构造类型、程序的结构等基础知识;第七至九章主要介绍了类和对象、继承和派生、单继承和多继承、虚函数和抽象类等多态的知识;第十至十一章主要介绍了 C++的 I/O 标准流、数据输入和输出、C++异常的处理等知识;第十二章是一个 C++的综合实例。

本书针对各个知识点,首先举出一个案例(实例),然后讲解相关知识点(小讲堂),再通过一两个实例(练一练)来加深对该知识点的印象,最后通过“想一想”提出本讲的一些重点内容或下一讲的相关思考题,对读者进行启发。本书注重基础,突出应用;对初学者容易混淆的内容进行了重点提示和讲解;针对高等职业教育的教学目标,提高读者的动手编程能力,以便于读者能够更好地理解 C++面向对象编程的知识。

本书适合作为高职高专类各相关专业的程序设计教材,也适合编程开发人员的培训、自学使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C++开发实例教程 / 刘畅主编. —北京:电子工业出版社, 2009.12

全国高等职业教育计算机类规划教材. 实例与实训教程系列. 国家高职示范院校核心课程教材

ISBN 978-7-121-09912-0

I. C… II. 刘… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 211749 号

策划编辑: 左 雅

责任编辑: 徐云鹏 特约编辑: 宋兆武

印 刷: 北京丰源印刷厂

装 订: 涿州市桃园装订有限公司

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: 22.75 字数: 582 千字

印 次: 2010 年 1 月第 1 次印刷

印 数: 4 000 册 定价: 33.00 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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

序

20世纪90年代以来,以计算机和通信技术为推动力的信息产业在我国获得前所未有的发展,全国各企事业单位对信息技术人才求贤若渴,高等教育计算机及相关专业毕业生供不应求。随后几年,我国各高等院校及众多培训机构相继开设计算机及相关专业,积极扩大招生规模,不久即出现了计算机及相关专业毕业生供大于求的局面。纵观近十年的就业市场变化,计算机专业毕业生经历了“一夜成名、求之不得”的宠幸,也遭遇了“千呼百应、尽失风流”的冷落。

这个时代深深地镌刻着信息的烙印,这个时代是信息技术人才尽情展示才能的舞台。目前我国劳动力市场,求职人数过剩,但满足企业要求的专业人才又很稀缺。这种结构性的人才市场供求矛盾是我国高等教育亟待解决的问题,更是“以人为本,面向人人”为目标的职业教育不可推卸的责任。

电子工业出版社作为我国出版职业教育教材最早的出版社之一,是计算机及相关专业高等职业教材重要的出版基地。多年来,我们一直在教材领域为战斗在职业教育第一线的广大职业院校教育工作者贡献着我们的力量,积累了丰富的职业教材出版经验。目前,计算机专业高等教育正处于发展中的关键时期,我们有义务、有能力协同全国各高等职业院校,共同探寻适合社会发展需要的人才培养模式,建设满足高等职业教育需求的教学资源——这是我们出版“全国高等职业教育计算机类规划教材·实例与实训教程系列”的初衷。

关于本系列教材的出版,我们力求做到以下几点:

(1) 面向社会人才市场需求,以培养学生技能为目标。工学结合、校企合作是职业教育发展的客观要求,面向就业是职业教育的根本落脚点。本系列教材内容体系的制定是广大高职教育专家、一线高职教师共同智慧的结晶。我们力求教材内容丰富而不臃肿、精简而不残缺,实用为主、够用为度。

(2) 面向高职学校教师,以方便教学为宗旨。针对每个课程的教学特点和授课方法,我们为其配备相应的实训指导、习题解答、电子教案、教学素材、阅读资料、程序源代码、电子课件、网站支持等一系列教学资源,广大教师均可从华信教育资源网(www.huaxin.edu.cn)免费获得。

(3) 面向高职学校学生,以易学、乐学为标准。以实例讲述理论、以项目驱动教学是本系列教材的显著特色。这符合现阶段我国高职学生的认知规律,能够提高他们的学习兴趣,增强他们的学习效果。

这是一个崭新的开始,但永远没有尽头。高等职业教育教材的建设离不开广大职业教育工作者的支持,尤其离不开众多高等职业院校教师的支持。我们诚挚欢迎致力于职业教育事业发展的有识之士、致力于高等职业教材建设的有才之士加入到我们的队伍中来,多批评,勤点拨,广结友,共繁荣,为我国高等职业教育的发展贡献我们最大的力量!

电子工业出版社高等职业教育分社

前 言

面向对象的程序设计方法把数据和处理数据的过程当成一个整体,具有封闭和数据隐藏、继承和重用及多态性的特点,成为开发大型软件所采用的主要方法。C++是面向对象的程序设计中应用最广泛的一种,也是现在各大专院校计算机专业必开的专业课程之一。

本书作为 C++ 程序设计的入门与应用教材,共分 12 章。主要内容结构如下:

第一章 C++ 编程简介: 主要介绍程序设计的基本概念, C++ 的特点、开发过程、程序结构, 以及 ++ 的运行环境介绍, 为以后学习编程知识打下基础。

第二章 数据类型与表达式: 主要介绍了 C++ 的基础知识, 包括数据类型、常量、变量、数据类型的转换等。

第三章 控制结构: 主要介绍了 C++ 各种语句、顺序、选择和循环结构程序设计方法。

第四章 函数: 主要介绍了 C++ 的函数的定义与声明、函数调用、函数重载和内联函数等内容。

第五章 构造数据类型: 主要介绍了 C++ 的数组、指针、结构、联合、枚举等构造类型的定义及使用方法。

第六章 C++ 程序的结构: 主要介绍了变量的存储和作用域, 以及程序的文件结构及编译预处理命令。

第七章 类和对象: 主要介绍了 C++ 的类和对象的概念, 构造函数与析构函数, 如何抽象、封装对象, 以及静态成员及友元的概念。

第八章 继承和派生: 主要介绍了 C++ 的基类和派生类, 单继承、多继承等继承方法, 二义性和虚基类等。

第九章 多态性和虚函数: 主要介绍了 C++ 的虚函数和抽象类等概念, 各种运算符重载的方法。

第十章 C++ 输入/输出流: 主要介绍了 C++ 的 I/O 标准流类、键盘输入、屏幕输出, 磁盘文件的输入和输出等。

第十一章 异常处理: 主要介绍了 C++ 的异常的概念、基本原理、异常处理方法和多路捕获。

第十二章 C++ 程序设计综合实例: 讲解了一个完整的 C++ 实例, 以提高读者实际编程能力。

本书的特点有以下几点:

第一, 内容的结构合理。

学习 C++ 不一定必须学习面向对象编程, 也可以通过 C++ 来学习面向过程的程序设计, 所以本书前后分成两大部分, 前面主要介绍了面向过程的程序设计, 后面完整介绍了面向对象的编程方法。没有 C 语言基础的读者可从头学起, 而学过 C 语言有基础的读者, 可以将前面知识迅速读完 (但要注意里面与 C 语言的不同之处), 再从第七章开始仔细学习有关面向对象的知识。读者学习完后, 就可以对 C++ 有个总体了解, 能够编写出各种功能的实用程序。

第二，让读者掌握计算机编程的数据存储及变化过程。

本书从常量、变量在内存的存放、函数的调用等知识都讲解了数据在内存中处理的信息，让读者更容易掌握所学知识。

第三，本书的编写方式采用了项目驱动方式讲解 C 语言的各知识点。

针对各个知识点，首先举出一个案例（实例×-×），然后讲解本案例的相关知识点（小讲堂），再通过一二个实例（练一练）来加深对该知识点的印象，最后通过“想一想”提出本讲的一些重点内容或下一讲的相关思考题，对读者进行启发。

第四，书中所有程序运行结果采用截屏方式，增加程序运行结果准确性。

以前很多教材的运行结果是用文字表示的，不直观。本书将各个源程序运行结果屏幕化，不可能出错，读者看起来更直观。

本书易教易学、学以致用、注重能力，对初学者容易混淆的内容进行了重点提示和讲解。针对高职的教学目标，提高读者的动手编程能力，以便于读者能够更好地理解 C++ 面向对象编程的知识，提高实际编程能力。本书的课后习题选用了一些典型的国家二级等级考试试题，方便读者加深所学重点知识的印象。本书适合作为高职高专类各相关专业的程序设计教材，也适合编程开发人员作培训、自学使用。

本书由电子工业出版社专门的平台华信教育资源网 (<http://www.hxedu.com.cn>) 为读者提供服务，可免费下载配套资源。配套资源主要有本书配套电子课件，书中所有实例题、练习题和课后编程题的程序源代码及参考的教学计划，方便教师授课和读者自学，本书中的所有源程序均在 Visual C++6.0 环境下通过上机调试。

本书由刘畅担任主编。其中，第三、四、七章及附录 B、附录 C 由刘畅编写，第八至十二章由刘辉编写，第一、二章由刘苗苗编写，第五、六章由胡艳梅编写。袁鸿雁、叶宾、张旭辉、冯书晶也参加了本书的审校和编辑工作，在此表示感谢。全书由刘畅负责统稿审校。

由于作者水平有限，经验不足，编写时间仓促，书中难免存在缺点及不足之处，恳请广大读者批评指正。

编 者

目 录

第一章 C++编程简介	(1)
第一讲 C++基础知识	(1)
一、计算机程序设计语言概述	(1)
二、C++概述	(4)
三、C++程序的开发过程	(5)
四、一个 C++程序实例	(6)
五、名字空间	(11)
六、无格式的输入/输出流	(12)
练一练	(13)
想一想	(14)
本章小结	(14)
课后习题一	(14)
第二章 数据类型与表达式	(16)
第二讲 C++的数据类型、常量与变量	(16)
一、C++的数据类型	(16)
二、变量和常量	(18)
练一练	(23)
本讲小结	(24)
想一想	(24)
第三讲 运算符与表达式	(24)
一、算术、关系、逻辑及赋值运算符及其表达式	(24)
二、其他运算符	(28)
三、类型转换	(34)
练一练	(35)
本讲小结	(35)
想一想	(36)
本章小结	(36)
课后习题二	(36)
第三章 C++的控制结构	(38)
第四讲 顺序结构	(38)
一、表达式语句及复合语句	(38)
二、输入/输出流的格式控制	(39)
练一练	(43)
本讲小结	(44)
想一想	(44)
第五讲 选择结构	(44)

一、标准 if 语句	(44)
二、复合 if 语句	(46)
三、if 语句的嵌套	(48)
四、switch 语句	(48)
练一练	(50)
本讲小结	(51)
想一想	(51)
第六讲 循环语句	(51)
一、while 循环	(51)
二、do-while 语句	(52)
三、for 循环语句	(54)
练一练	(55)
本讲小结	(56)
想一想	(56)
第七讲 嵌套循环及流程转向语句	(56)
一、循环语句的嵌套	(57)
二、goto 语句	(58)
三、break 语句	(59)
四、continue 语句	(60)
练一练	(61)
本讲小结	(62)
想一想	(62)
本章小结	(62)
课后习题三	(62)
第四章 函数	(67)
第八讲 函数的定义、调用、返回语句及函数声明	(67)
一、函数的定义和调用	(67)
二、函数参数的按值传递	(72)
练一练	(73)
本讲小结	(75)
想一想	(75)
第九讲 函数的嵌套调用和递归调用	(75)
一、嵌套调用	(75)
二、递归调用	(76)
练一练	(78)
本讲小结	(80)
想一想	(80)
第十讲 内联函数、带默认形参的函数、函数重载及函数模板	(81)
一、内联函数	(81)
二、带默认形参值的函数	(83)

三、函数重载	(85)
四、函数模板	(87)
五、C++系统函数	(89)
练一练	(90)
本讲小结	(91)
想一想	(91)
本章小结	(91)
课后习题四	(91)
第五章 构造数据类型	(94)
第十一讲 数组	(94)
一、一维数组的定义、初始化、数据赋值、数组越界	(94)
二、二维数组	(97)
三、数组作为函数参数	(99)
练一练	(101)
本讲小结	(101)
想一想	(102)
第十二讲 指针基础知识	(102)
一、指针的定义、初始化、运算及 const 指针	(102)
二、一维、二维数组中的指针使用	(106)
三、指向一维数组的指针	(109)
练一练	(110)
本讲小结	(111)
想一想	(111)
第十三讲 字符指针、指针数组、指向指针的指针和常用字符串处理函数	(111)
一、字符指针和字符串	(111)
二、指针数组和指向指针的指针	(113)
三、常用字符串处理函数	(115)
练一练	(117)
本讲小结	(118)
想一想	(118)
第十四讲 指针和函数	(118)
一、指针作为函数参数	(118)
二、返回指针的函数	(120)
三、指向函数的指针	(121)
练一练	(123)
本讲小结	(124)
想一想	(124)
第十五讲 动态存储分配、void 指针和引用	(124)
一、动态存储分配和 void 指针	(124)
二、引用	(127)

练一练	(130)
本讲小结	(131)
想一想	(132)
第十六讲 结构、联合与枚举类型	(132)
一、结构类型	(132)
二、联合类型	(138)
三、枚举类型	(140)
练一练	(141)
本讲小结	(142)
想一想	(142)
本章小结	(142)
课后习题五	(143)
第六章 C++程序的结构	(147)
第十七讲 作用域与存储类型	(147)
一、全局变量与局部变量	(147)
二、变量的作用域	(151)
三、域运算符	(153)
四、名字空间的定义与使用	(153)
练一练	(155)
本讲小结	(156)
想一想	(156)
第十八讲 程序的文件结构与编译预处理命令	(156)
一、文件包含命令	(157)
二、宏定义	(158)
三、条件编译	(161)
练一练	(163)
本讲小结	(164)
想一想	(164)
本章小结	(164)
课后习题六	(164)
第七章 类和对象	(168)
第十九讲 类和对象基础知识	(168)
一、类的声明、成员函数及访问权限, 对象的定义和引用及对象的赋值	(168)
练一练	(173)
本讲小结	(174)
想一想	(175)
第二十讲 构造函数和析构函数	(175)
一、构造函数和析构函数	(175)
练一练	(182)
本讲小结	(184)

想一想	(184)
第二十一讲 对象的使用	(184)
一、对象指针和动态对象	(184)
二、对象数组	(187)
三、this 指针	(189)
练一练	(191)
本讲小结	(192)
想一想	(192)
第二十二讲 静态成员和友元	(192)
一、静态数据成员	(192)
二、静态成员函数	(194)
三、友元函数	(196)
四、友元类	(198)
练一练	(200)
本讲小结	(201)
想一想	(202)
本章小结	(202)
课后习题七	(202)
第八章 继承和派生	(207)
第二十三讲 基类和派生类	(207)
一、继承的概念和定义	(207)
二、派生类的三种继承方式	(211)
练一练	(215)
本讲小结	(217)
想一想	(217)
第二十四讲 派生类的构造函数和析构函数	(217)
一、派生类的构造函数和析构函数	(217)
练一练	(222)
本讲小结	(223)
想一想	(223)
第二十五讲 二义性和虚基类	(224)
一、二义性及解决方法	(224)
二、虚基类和虚继承	(226)
三、虚基类的构造函数	(228)
练一练	(230)
本讲小结	(232)
想一想	(232)
本章小结	(232)
课后习题八	(233)

第九章 多态和虚函数	(234)
第二十六讲 虚函数	(234)
一、使用基类对象访问派生类对象的成员	(234)
二、虚函数	(238)
练一练	(241)
本讲小结	(243)
想一想	(243)
第二十七讲 运算符重载	(243)
一、运算符重载	(243)
二、运算符重载为类的成员函数	(245)
三、重载为类的友元函数	(247)
练一练	(249)
本讲小结	(251)
想一想	(251)
第二十八讲 典型运算符重载	(251)
一、单目运算符重载	(251)
二、赋值运算符重载	(254)
三、下标运算符重载	(256)
练一练	(257)
本讲小结	(259)
想一想	(259)
本章小结	(259)
课后习题九	(259)
第十章 C++输入/输出流	(261)
第二十九讲 C++输入/输出概述	(261)
一、C++中流的概念	(261)
二、C++中标准输入/输出流	(264)
三、ios 类中控制输出格式的成员函数	(266)
四、格式控制操作符	(271)
练一练	(272)
本讲小结	(273)
想一想	(274)
第三十讲 文件系统概述	(274)
一、文件系统基础知识及数据流的错误检测	(274)
二、文件的打开/关闭操作	(276)
练一练	(281)
本讲小结	(282)
想一想	(282)
第三十一讲 文件输入/输出	(282)
一、文本文件的读写	(282)

二、二进制文件的读写	(285)
三、文件指针	(287)
练一练	(291)
本讲小结	(292)
想一想	(293)
本章小结	(293)
课后习题十	(293)
第十一章 异常处理	(295)
第三十二讲 异常处理基础知识	(295)
一、异常处理机制	(295)
二、异常类的定义	(300)
练一练	(304)
本讲小结	(305)
想一想	(305)
本章小结	(305)
课后习题十一	(306)
第十二章 C++程序设计综合实例	(307)
第三十三讲 实例讲述	(307)
一、问题描述	(307)
二、算法分析	(307)
三、数据说明	(308)
四、功能说明	(309)
五、主函数的完整程序	(313)
附录 A 课后习题参考答案	(315)
课后习题一参考答案	(315)
课后习题二参考答案	(315)
课后习题三参考答案	(317)
课后习题四参考答案	(320)
课后习题五参考答案	(322)
课后习题六参考答案	(327)
课后习题七参考答案	(328)
课后习题八参考答案	(332)
课后习题九参考答案	(337)
课后习题十参考答案	(339)
课后习题十一参考答案	(342)
附录 B 常用字符与 ASCII 码对照表	(344)
附录 C 运算符的优先级、结合性	(346)
参考文献	(347)

第一章 C++编程简介

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。C++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。

本章读者主要学习以下内容：

- ◆ 了解不同类型的编程语言；
- ◆ 了解面向过程的结构化程序设计语言与面向对象程序设计语言的特点；
- ◆ 了解面向对象程序设计语言的基本概念；
- ◆ 熟悉C++程序的开发过程；
- ◆ 能够编写一个简单的程序，输入数据，输出结果。

第一讲 C++基础知识

学习目标

- 了解C++的发展历史；
- 掌握C++程序的开发过程；
- 掌握C++的词法与规则；
- 掌握面向对象的程序设计方法；
- 掌握C++程序的构成。

一、计算机程序设计语言概述

小讲堂

1. 机器语言与汇编语言

机器语言（Machine Language）是计算机自身的语言，是计算机直接理解执行的语言，是人类使用的第一种编程语言。机器语言由一系列指令组成，指令是计算机可识别的逐个执行的基本命令，由二进制的0和1构成。例如：将100与200相加的机器语言程序由下列两条指令实现：

1101	1000	0110	0100	0000	0000	(B86400)
0000	0101	1100	1000	0000	0000	(05C800)

虽然机器语言能为计算机直接识别和执行，但对于人类来说却十分晦涩难懂，更难记忆与编程。在计算机发展的初期，程序员只能用机器语言编程，在这一阶段计算机编程语言与人类的自然语言之间存在着巨大的差异，软件开发难度大、周期长、修改维护困难。

为了解决机器语言编程的困难，程序员使用类似英文缩写的助记符来表示指令，这些助记符构成了汇编语言（Assembly Language）。如使用 ADD、SUB 助记符分别表示加、减运算指令。例如，将上面 100 与 200 相加的机器语言改为汇编语言，程序实现为：

```
MOV  AX, 100
ADD  AX, 200
```

汇编语言由称为汇编器（Assembler）的翻译程序将它转换成机器语言程序。此时，虽然汇编语言比机器语言提高了一步，汇编语言与人类自然语言之间的差异略有缩小，但仍然与人类的思维相差甚远。由于汇编语言的抽象层次太低，一个简单的任务需要大量的语句来实现，并且程序员还需考虑大量的机器细节，故使用汇编语言编程难度大。

小讲堂

2. 高级语言

汇编语言实质上是机器语言，与人的自然语言习惯差别大，依赖于具体的硬件系统，因此，同样属于低级语言。为了进一步方便编程，人们开发了更加接近人类的自然语言习惯的高级语言，它屏蔽了机器的细节，不依赖于具体的硬件系统，提高了语言的抽象层次，程序中采用更有意义和容易理解的语句，使程序更容易描述具体的事物与过程，编程效率大大提高。例如，还是将 100 与 200 相加，用下面的一条高级语言即可实现：

```
100+200
```

高级语言由称为编译器（Compiler）的翻译程序将它变成相应的机器语言程序。BASIC、FORTRAN、COBOL、ALGOL 均属于高级语言，它们具有较好的语言表达能力，可广泛用于科学计算和商业事务处理中。

早期的计算机多用于数学计算，为了完成计算，必须设计出一个计算方法或解决问题的过程，因此早期的高级语言是一种面向过程的语言。随着计算机处理的问题日益复杂，程序也越来越庞大，许多大型软件的开发遇到了困难，甚至陷入“软件危机”的状态。20 世纪 60 年代产生了结构化的程序设计思想，为用面向过程的方法解决复杂的问题提供了有力手段。结构化的程序设计是软件发展中继子程序与高级语言后的第三个影响更为深远的里程碑。

结构化程序设计方法主要内容有：自顶向下，逐步求精；程序按功能划分为若干模块，这些模块形成调用的层次树状结构；模块间的关联尽可能的简单，模块的功能相对独立；每个模块均由顺序、选择和循环三种基本结构组成。

由于结构化程序设计采用了模块分解与功能抽象以及自顶向下、分而治之的方法，从而将一个复杂的系统分解成易于实现和控制的子任务，显著减少了软件开发的复杂性，提高软件的可靠性、可测试性和可维护性。正是由于结构化程序设计方法的这一系列的特点和优点，使 20 世纪 70 年代时结构化语言获得蓬勃发展并得到广泛应用，以致计算机应用人员将 20 世纪 70 年代誉为“结构化语言年代”。Pascal 语言、C 语言属于结构化语言的代表。

结构化编程语言进一步提高了语言的层次。结构化数据、结构化语句、数据抽象、过程抽象概念使程序更便于体现客观事物的结构和逻辑含义。这使得编程语言与人类的自然语言更加接近。

虽然结构化语言有很多优点，但其本质仍然是一种面向过程的语言。它把数据和数据处

理过程分离成相互独立的实体,当数据结构改变时,所有相关的处理过程都要进行相应的修改,程序的可重用性差。同时,由于图形用户界面(Graphics User Interface, GUI)的使用,应用软件应随时响应用户的各种操作,因此软件的功能很难用过程来描述与实现,如果仍然使用面向过程的方法与语言,软件的开发与维护将面临困难。

为了解决软件开发过程中的困难,消除结构化编程语言的局限,自20世纪80年代以来,出现了面向对象的程序设计语言(Object-Oriented Program Language, OOP)。

小讲堂

3. 面向对象的程序设计语言

面向对象的程序设计语言与以往的各种编程语言的根本不同点在于,它设计的出发点就是为了能更加直接地描述客观世界中存在的事物和对象,以及它们之间的关系。

开发一个软件是为了解决现实世界中的问题,这些问题所涉及的业务范围称为该软件的问题域。面向对象的程序设计语言将现实世界中的客观事物描述成具有属性和行为(或称为服务)的对象,通过抽象找出同一类对象的共同属性(静态特征)和行为(动态特征),形成类。

类通过一个简单的外部接口与外界发生关系,对象与对象之间通过消息进行通信。这样,程序模块间的关系更为简单,程序模块的独立性、数据的安全性就有了良好的保障。类的继承与多态性可以很方便地实现代码的重用,大大提高了程序的可重用性,缩短了软件的开发周期,并使软件风格统一。

因此,面向对象的编程语言使程序能够比较直接反映问题域的本来面目,软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。

面向对象的语言的发展经历了一个很长的发展阶段,早在20世纪60年代中后期,Simula-67语言中使用了对象的概念并开始使用数据封装。20世纪70年代Xerox公司经过对Smalltalk 72, 74, 76语言的不断研究、改进之后,在1980年推出商品化的Smalltalk-80。Smalltalk-80是第一个真正的面向对象的语言。其后,许多语言中也广泛引入了面向对象的特征。1985年AT&T公司贝尔实验室开发了C++。1995年5月SUN公司发布了Java语言。

小讲堂

4. 面向对象的基本概念

面向对象的程序设计是在吸取结构化程序设计的一切优点的基础上发展起来的一种新的程序设计方法。它的本质是把数据和处理数据的过程当成一个整体——对象。

面向对象是C++概念中的主要概念,在学习C++之前首先要了解这些概念。

(1) 对象。从一般意义上讲,对象是现实世界中一个实际存在的事物,它可以是有形的(比如一架飞机),也可以是无形的(比如一个项目、一个计划)。对象是构成世界的一个独立单位,具有自己的静态特征(状态)和动态特征(操作)。静态特征是可以由某种数据来描述的特征,动态特征是对象所表现的行为或对象所具有的功能。

面向对象的语言把状态和操作封装于对象体之中,并提供一种访问机制,使对象的“私有数据”仅能由这个对象的操作来执行。用户只能通过允许公开的操作提出要求(消息),才能查询和修改对象的状态。

(2) 类。把众多的事物归纳,划分成一些类,把具有共性的事物划为一类,得出一个抽

象的概念，是人类认识世界经常采用的思维方法。类的概念使人们能对属于该类的全部个体事物进行统一的描述。

类是面向对象的语言必须提供的用户定义的数据类型，它将具有相同状态、操作和访问机制的多个对象抽象成为一个对象类。

类与对象的关系如同汽车与具体一辆车的关系。汽车能跑，有四个轮子，所有的汽车组成了一个类，具体到一辆车上，它具有类的全部特性（能跑，有四个轮子），是汽车类的一个子集或元素。类给出了属于该类的全部对象的抽象定义，而对象则是符合这种定义的一个实体。所以，一个对象又称做类的一个实例。

（3）封装。封装是面向对象的方法的一个重要原则。它有两个含义：第一个含义是把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位（即对象）；第二个含义也称做“信息隐蔽”，即尽可能隐蔽对象的内部细节，对外形成一个边界（或者说形成一道屏障），只保留有限的对外接口使之与外部发生联系。这主要是指对象的外部不能直接地存取对象的属性，只能通过几个允许外部使用的服务与对象发生联系。

（4）继承。继承是面向对象的语言的另一特性。类与类之间可以组成继承层次，一个类的定义（子类）可以定义在另一个已定义类（父类）的基础上。子类可以继承父类中的属性和操作，也可以定义自己的属性和操作。C++支持单继承和多继承，因而具有继承这一特性所带来的优势，大大增加了程序的重要性。

（5）消息。消息是向对象发出的服务请求，通过消息进行对象之间的通信，消息包含提供服务的对象标识、服务标识、输入信息、回答信息。

消息在面向对象的程序设计语言中用函数调用实现。

（6）多态性。对象的多态性是指在类中定义的属性或操作被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或操作名在一般类及其各个特殊类中具有不同的语义。

二、C++概述



5. C++的发展历史

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门新的语言，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程序设计语言，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，会使程序员更易于理解，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