

丛书主编 徐明华

全新改版
畅销升级

软件开发 新课堂

- ◎ 零基础初学者**入门必读**
- ◎ 知名高校**教授+培训**专家联合编写
- ◎ 从基础知识到案例开发**融会贯通**
- ◎ **知识点+应用示例+技术难点**剖析
- ◎ 体验**5个项目案例**开发全过程

C++

基础与案例开发详解

王海龙 程传鹏◎编著



DVD 教学光盘

培训专家对书中案例进行全程视频语音讲解
包括书中所有的实例源程序及相关文件



清华大学出版社

.. 014006433

TP312C

~~2236-3~~

2237

软件开发新课堂

C++基础与案例开发详解

王海龙 程传鹏 编 著



P

7P312C

2237

清华大学出版社



北航

C1693512

内 容 简 介

作为从 C 语言发展而来的一种优秀的高级程序设计语言, C++ 在计算机科学的各个领域得到了广泛的应用, 尤其适合于系统级的程序设计, 是应用最广泛的程序设计语言之一。

本书在结构安排上充分考虑到了初学者的实际情况, 在组织结构上, 针对 C++ 语言的特点, 分为三部分, 共 17 章。其中第 1~7 章为第一部分, 主要介绍了结构化程序设计, 包括 C++ 基础知识、控制流程语句、函数、数组、字符串、指针、结构体、共用体和枚举类型、链表、内存动态分配和释放等。第 8~12 章为第二部分, 主要介绍了面向对象程序设计, 包括类和对象、继承和派生、虚函数和多态性; I/O 流类以及文件操作、函数模板、类模板和标准模板库等。第 13~17 章为第三部分, 是综合案例, 依次介绍了银行卡管理系统、成绩管理系统、工资发放系统、旅店管理系统、图书管理系统等。通过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的思想和方法。

本书既可作为高等院校各专业学生学习 C++ 的基础教材, 又可作为编程爱好者的自学教程。

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++ 基础与案例开发详解/王海龙, 程传鹏编著. —北京: 清华大学出版社, 2014
(软件开发新课堂)

ISBN 978-7-302-33961-8

I. ①C… II. ①王… ②程… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 222438 号

责任编辑: 杨作梅

装帧设计: 杨玉兰

责任校对: 宋延清

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>, 010-62791865

印 装 者: 北京嘉实印刷有限公司

经 销: 全国新华书店

开 本: 190mm×260mm 印 张: 28.5 插 页: 1 字 数: 692 千字
(附 DVD1 张)

版 次: 2014 年 1 月第 1 版

印 次: 2014 年 1 月第 1 次印刷

印 数: 1~3500

定 价: 58.00 元

产品编号: 051146-01

丛书编委会

丛书主编：徐明华

编 委：(排名不分先后)

李天志 易 魏 王国胜 张石磊

王海龙 程传鹏 于 坤 李俊民

胡 波 邱加永 许焕新 孙连伟

徐 飞 韩玉民 郑彬彬 夏敏捷

张 莹 耿兴隆

丛 书 序

首先,感谢并祝贺您选择本系列丛书!《软件开发新课堂》系列是为了满足广大读者的需求,在原《软件开发课堂》系列书的基础上进行的升级和重新编辑。秉承了原系列书的精髓,通过大量的精彩实例、完整的学习视频,让您完全融入编程实战演练,从零开始,逐步精通相关知识,成为自学成才的编程高手。哪怕您没有任何编程基础,都可以轻松地实现职场的梦想和生活的愿望!

1. 丛书内容

随着软件行业的不断升温,程序员这一职业正在成为 IT 界中的佼佼者,越来越多的程序设计爱好者开始投入相关软件开发的学习中。然而很多朋友在面对大量的代码时又有些望而却步,不知从何入手。

实际上,一本好书不仅要教会读者怎样去实现书中的内容,更重要的是要教会读者如何去思考、去探究、去创新。鉴于此,我们精心编写了《软件开发新课堂》系列丛书。

本丛书涉及目前流行的各种相关编程技术,均以最常用的经典实例,来讲解软件最核心的知识点,让读者掌握最实用的内容。首次共推出 10 册:

- 《Java 基础与案例开发详解》
- 《JSP 基础与案例开发详解》
- 《Struts 2 基础与案例开发详解》
- 《JavaScript 基础与案例开发详解》
- 《ASP.NET 基础与案例开发详解》
- 《C#基础与案例开发详解》
- 《C++基础与案例开发详解》
- 《PHP 基础与案例开发详解》
- 《SQL Server 基础与案例开发详解》
- 《Oracle 数据库基础与案例开发详解》

2. 丛书特色

本丛书具有以下特色。

(1) 内容精练、实用。本着“必要的基础知识+详细的程序编写步骤”原则,摒弃琐碎的东西,指导初学者采取最有效的学习方法和获得最良好的学习途径。

(2) 过程简洁、步骤详细。尽量以可视化操作讲解,讲解步骤做到详细但不繁琐,避免直接使用大量代码占用读者的阅读时间。而对关键代码则进行详细的讲解,做到清晰和透彻。

(3) 讲解风格通俗易懂。作者均是一线工作人员及教学人员,项目经验丰富,传授知识的能力强。所选案例精练、实用,具有实战性和代表性,能够使读者快速上手。

(4) 光盘内容丰富。不仅包含书中的所有代码及实例,还包含书中主要操作步骤的视

频录像,有利于多媒体视频教学和自学,最大程度地提高了书中案例的可操作性。

3. 作者队伍

本丛书由知名培训师徐明华老师任主编,作者团队主要有北京达内科技、北京电子商务学院、郑州中原工学院、天津程序员俱乐部、徐州力行文化传媒工作室等机构和学院的专业人员及教师。正是有了他们无私的付出,本丛书才能顺利出版。

4. 读者对象

本丛书定位于初、中级读者。书中每个实例都是从零起步,初学者只需按照书中的操作步骤、图片说明,或根据多媒体视频,便可轻松地制作出实例的效果。不仅适合程序设计初学者以及普通编程爱好者使用,也可作为大、中专院校,高职高专学校,及各种社会培训机构的教材与参考书。

5. 特别感谢

本丛书从立项到写作受到广大朋友的热心支持,在此特别感谢达内科技的王利锋先生、北大青鸟的张宏先生,还有单兴华,吴慧龙、聂靖宇、刘烨、孙龙、李文清、李红霞、罗加顺、冯少波、王学锋、罗立文、郑经煜等朋友,他们对本丛书的编著提供了很好的建议。祝所有关心和支持本丛书的朋友身体健康,工作顺利。

最后还要特别感谢已故的北京传智播客教学总监张孝祥老师,感谢他在原《软件开发课堂》系列书中无私的帮助与付出。

6. 提供的服务

为了有效地解答读者在阅读过程中遇到的问题,丛书专门在 <http://bbs.022tomo.com/>开辟了论坛,以方便读者交流。

丛书编委会



前 言

C++语言是从C语言发展而来的一种优秀的高级程序设计语言，它具有全面兼容C语言并支持面向对象的特点。面向对象的设计思想是在原来结构化程序设计方法基础上的一个质的飞跃，C++完美地体现了面向对象的各种特性，支持过程化程序设计、数据抽象、面向对象程序设计、泛型程序设计等多种程序设计风格。C++以其独特的语言机制在计算机科学的各个领域得到了广泛的应用，尤其适合于系统级的程序设计，是应用最广泛的程序设计语言之一，深受广大编程爱好者的喜爱。

本书特色如下。

- 全新体系，内容翔实：本书以面向过程的程序设计为切入点，从编写简单的程序开始，循序渐进，由面向过程到面向对象，逐步深入，符合读者的认识规律。
- 概念清晰，突出算法：本书详细介绍了程序设计的基本思想、概念和技术，C++语言是作为程序设计的工具，所以我们将从语言、语法的学习，上升到程序的“设计、算法、编程、调试”的层次。
- 通俗易懂，可读性强：本书尽量用通俗易懂的语言来叙述复杂的概念，力求减少初学者学习C++的困难。
- 理论丰富，注重实践：本书每个章节中均包含多个实例，着重于强化编程实践，注重动手能力的培养。

本书共分17章，在组织结构上，针对C++语言的特点，分为三部分。其中第1~7章为第一部分，主要介绍结构化程序设计，包括C++基础知识、控制流程语句、函数、数组、字符串、指针、结构体、共用体和枚举类型、链表、内存动态分配和释放等。第8~12章为第二部分，主要介绍面向对象程序设计，包括类和对象、继承和派生、虚函数和多态性；I/O流类以及文件操作、函数模板、类模板和标准模板库等。第13~17章为第三部分，是综合案例，依次介绍银行卡管理系统、成绩管理系统、工资发放系统、旅店管理系统、图书管理系统等。通过精心设计的案例，深入浅出地讲解C++语言程序设计的特点，循序渐进地介绍面向对象程序设计的思想和方法。

本书基础知识介绍清晰明了，理论联系实际，具有很强的操作性。实例介绍知识面广，不仅温习了前面所学的内容，还扩展介绍了相关的应用技巧，从而保证读者能够更好地掌握C++程序设计语言。

本书既可作为高等院校各专业学生学习C++的基础教材，又可作为编程爱好者的自学教程。即使没有教师讲授，读者也能完全读懂本书的内容。

本书由王海龙、程传鹏老师担任主编并统稿，其中，第1~2章由郭德双编写，第5~7章由夏敏捷编写，第13~15章由程传鹏编写，其他章节由王海龙编写。此外，徐明华、尼

春雨、陈丽丽、李远、尼朋、赵丹、王国胜、伏银恋、胡文华、曹培培、孟文婷、蒋燕燕、蒋军军、张班班、张悦、张丽、张石磊等人也参与了本书的整理核对和光盘制作，在此一并表示感谢。

由于编者水平有限，书中难免有疏漏和不足之处，恳请专家和广大读者指正。

编 者



目 录

第 1 章 C++概述.....1	2.5.1 格式化输入输出 ——scanf()和 printf()27
1.1 程序设计语言的发展.....2	2.5.2 I/O 流.....29
1.1.1 机器语言阶段.....2	2.5.3 cin.....29
1.1.2 汇编语言阶段.....2	2.5.4 cout.....30
1.1.3 高级语言阶段.....3	2.5.5 输出控制符.....30
1.1.4 从 C 到 C++.....4	2.6 上机练习.....33
1.2 简单的 C++程序介绍.....4	第 3 章 C++流程控制.....35
1.3 程序开发的过程.....7	3.1 算法.....36
1.4 上机练习.....8	3.1.1 算法的概念.....36
第 2 章 数据类型、运算符和表达式.....11	3.1.2 算法的表示.....36
2.1 标识符和关键字.....12	3.2 C++语句.....37
2.1.1 标识符.....12	3.3 C++程序的三种基本结构.....38
2.1.2 关键字.....12	3.4 顺序结构程序.....39
2.2 C++的基本数据类型.....13	3.5 选择结构程序.....39
2.2.1 int 和 double 类型.....13	3.5.1 if 语句.....40
2.2.2 其他数值类型.....14	3.5.2 嵌套 if 语句.....42
2.2.3 char 类型.....15	3.5.3 switch 语句.....44
2.2.4 类型的兼容性.....15	3.6 循环结构程序.....46
2.3 变量和赋值.....16	3.6.1 while 语句.....46
2.3.1 变量.....16	3.6.2 do-while 语句.....48
2.3.2 赋值语句.....17	3.6.3 for 循环语句.....50
2.4 基本运算符和表达式.....19	3.6.4 循环的嵌套.....52
2.4.1 基本运算符和表达式的 简介.....19	3.6.5 转向语句.....55
2.4.2 算术运算符和算术表达式.....20	3.6.6 三种循环的比较.....57
2.4.3 赋值运算符和赋值表达式.....21	3.7 上机练习.....57
2.4.4 关系运算符和关系表达式.....22	第 4 章 函数和预处理.....59
2.4.5 逻辑运算符与逻辑表达式.....23	4.1 标准库函数.....60
2.4.6 位运算符和位运算表达式.....24	4.2 程序员自定义函数.....62
2.4.7 条件运算符和条件表达式.....25	4.2.1 函数定义举例.....62
2.4.8 逗号运算符和逗号表达式.....26	4.2.2 函数定义语法总结.....65
2.4.9 sizeof 运算符和表达式.....26	4.3 过程抽象.....66
2.5 简单的输入/输出实现方法.....26	4.3.1 案例分析: 购买蛋糕.....67

4.3.2 案例分析: 企业管理.....	70	5.3.2 二维数组初始化.....	110
4.4 函数的嵌套调用和递归调用.....	71	5.3.3 二维数组应用.....	111
4.4.1 函数的嵌套调用.....	71	5.4 字符数组的定义及应用.....	114
4.4.2 函数的递归调用.....	73	5.4.1 字符数组和字符串.....	114
4.5 内联函数和函数重载.....	76	5.4.2 字符串处理函数.....	117
4.5.1 内联函数.....	76	5.4.3 二维字符数组.....	120
4.5.2 函数重载.....	77	5.4.4 字符数组应用.....	121
4.6 具有默认参数值的函数.....	81	5.5 数组和函数.....	125
4.7 局部变量与全局变量.....	82	5.5.1 向函数传递一维数组.....	125
4.7.1 局部变量.....	82	5.5.2 向函数传递二维数组.....	128
4.7.2 全局变量.....	83	5.6 上机练习.....	129
4.7.3 动态存储方式与静态存储 方式.....	85	第6章 指针和引用	131
4.8 变量的存储类型.....	86	6.1 指针与指针变量.....	132
4.8.1 自动类型(auto).....	86	6.1.1 地址与指针的概念.....	132
4.8.2 寄存器类型(register).....	87	6.1.2 指针变量.....	133
4.8.3 静态类型(static).....	88	6.1.3 指针变量的运算.....	136
4.8.4 外部类型(extern).....	90	6.2 指针与数组.....	139
4.9 编译预处理.....	91	6.2.1 一维数组与指针.....	139
4.9.1 宏定义.....	92	6.2.2 二维数组与指针.....	141
4.9.2 文件包含命令.....	93	6.2.3 字符指针.....	143
4.9.3 条件编译命令.....	94	6.2.4 指针数组.....	145
4.10 程序的多文件组织.....	96	6.2.5 指向指针的指针 (二级指针).....	147
4.10.1 头文件.....	96	6.3 堆的动态分配和释放.....	147
4.10.2 多文件结构.....	96	6.3.1 new 运算符.....	148
4.10.3 多文件结构程序示例.....	97	6.3.2 delete 运算符.....	148
4.11 上机练习.....	98	6.4 指针与函数.....	150
第5章 数组	99	6.4.1 指针变量作为函数参数.....	150
5.1 数组的概念.....	100	6.4.2 函数指针.....	155
5.1.1 数组与数组元素.....	100	6.4.3 返回指针值的函数.....	157
5.1.2 数组的维数.....	100	6.5 引用.....	158
5.2 一维数组.....	100	6.5.1 引用及声明方法.....	158
5.2.1 一维数组的定义和初始化.....	100	6.5.2 用引用作为函数的参数.....	160
5.2.2 一维数组的操作.....	102	6.5.3 如何使一个被调函数同时 返回多个值.....	161
5.2.3 数组的大小.....	103	6.5.4 用引用返回函数值.....	162
5.2.4 一维数组的应用.....	103	6.6 上机练习.....	163
5.3 二维数组.....	108		
5.3.1 二维数组的定义.....	109		

第7章 结构体、共用体和枚举类型.....165

7.1 结构体的定义及使用.....166	
7.1.1 结构体的定义.....166	
7.1.2 定义结构体变量的方法.....167	
7.1.3 结构体变量的使用.....169	
7.1.4 结构体变量的初始化.....169	
7.1.5 结构体数组.....170	
7.1.6 结构体和函数.....175	
7.2 共用体的定义与使用.....176	
7.2.1 共用体的定义.....176	
7.2.2 定义共用体类型变量.....177	
7.2.3 共用体变量的使用.....177	
7.3 枚举类型.....179	
7.4 链表.....181	
7.4.1 结构体与指针.....181	
7.4.2 链表概念.....183	
7.4.3 链表的基本操作.....184	
7.5 typedef 定义类型.....193	
7.6 上机练习.....194	

第8章 类和对象.....195

8.1 面向对象程序设计基本概念.....196	
8.1.1 类.....196	
8.1.2 对象.....196	
8.1.3 封装与数据隐藏.....197	
8.1.4 继承.....197	
8.1.5 多态性.....197	
8.1.6 消息.....198	
8.2 类和对象的定义.....198	
8.2.1 定义类和成员函数.....198	
8.2.2 公共成员和私有成员.....200	
8.2.3 类和对象编程举例.....205	
8.2.4 类对象的内存分配.....210	
8.2.5 this 指针.....211	
8.3 用于初始化的构造函数.....213	
8.3.1 构造函数的定义.....213	
8.3.2 默认构造函数.....217	
8.3.3 析构函数.....218	

8.3.4 拷贝构造函数和默认拷贝

构造函数.....220

8.4 类和对象的进一步应用.....224	
8.4.1 堆对象.....224	
8.4.2 对象数组.....225	
8.4.3 类对象作为成员.....227	
8.5 静态成员.....229	
8.5.1 静态数据成员.....229	
8.5.2 静态成员函数.....232	
8.6 友元函数和友元类.....235	
8.6.1 友元函数.....235	
8.6.2 友元类.....239	
8.7 上机练习.....240	

第9章 继承与派生.....243

9.1 继承与派生的基础知识.....244	
9.2 类的继承方式.....251	
9.2.1 公有继承.....251	
9.2.2 私有继承.....253	
9.2.3 保护继承.....255	
9.2.4 继承方式的总结和比较.....255	
9.3 派生类的构造函数与析构函数.....256	
9.3.1 简单派生类的构造函数.....256	
9.3.2 析构函数.....258	
9.3.3 复杂派生类的构造函数和析构函数.....258	
9.3.4 派生友元类.....261	
9.4 基类与派生类的转换.....262	
9.5 上机练习.....264	

第10章 多态性.....265

10.1 多态性的概念.....266	
10.2 运算符重载.....266	
10.2.1 运算符重载的规则.....267	
10.2.2 运算符重载为成员函数.....267	
10.2.3 运算符重载为友元函数.....268	
10.2.4 赋值运算符重载.....270	
10.2.5 单目自增和自减的运算符重载.....272	



10.2.6	关系运算符重载.....	273	12.4.1	异常处理基础.....	324
10.2.7	下标运算符重载.....	275	12.4.2	异常处理机制.....	325
10.2.8	类型转换运算符重载.....	276	12.4.3	异常函数.....	329
10.3	联编和虚函数.....	279	12.5	上机练习.....	331
10.3.1	虚函数的作用和定义.....	279	第 13 章	银行卡管理系统.....	333
10.3.2	虚函数的使用原则.....	281	13.1	功能需求.....	334
10.3.3	虚函数的访问.....	281	13.2	总体设计.....	334
10.3.4	虚析构函数.....	282	13.2.1	总体功能模块设计.....	334
10.4	纯虚函数和抽象类.....	284	13.2.2	数据结构设计.....	335
10.5	上机练习.....	286	13.2.3	程序界面设计.....	335
第 11 章	输入和输出流类库.....	289	13.3	银行卡管理系统的实现.....	337
11.1	输入/输出流概念.....	290	13.3.1	主函数的源代码.....	337
11.2	流类库.....	291	13.3.2	创建新卡模块的源代码.....	338
11.3	格式化 I/O.....	293	13.3.3	卡号登录模块的源代码.....	339
11.3.1	使用 ios 类成员函数.....	293	13.3.4	存款操作的源代码.....	340
11.3.2	使用操纵符.....	296	13.3.5	取款操作的源代码.....	340
11.4	文件流.....	298	13.3.6	日志文件读/写的源代码.....	341
11.4.1	文件流类与文件流对象.....	299	13.4	本章小结.....	341
11.4.2	文件的打开和关闭.....	300	第 14 章	成绩管理系统.....	343
11.4.3	文本文件的输入/输出.....	302	14.1	功能需求.....	344
11.4.4	二进制文件的输入/输出.....	305	14.2	总体设计.....	344
11.4.5	文件的随机访问.....	306	14.2.1	系统功能结构.....	344
11.5	重载 I/O 输入输出运算符.....	308	14.2.2	数据结构及全局变量/函数 设计.....	345
11.5.1	重载输出运算符“<<”.....	308	14.2.3	程序界面设计.....	346
11.5.2	重载输入运算符“>>”.....	310	14.3	学生成绩管理系统的实现.....	349
11.6	上机练习.....	311	14.3.1	主函数源代码.....	349
第 12 章	模板和异常处理.....	313	14.3.2	学生信息录入函数.....	353
12.1	函数模板.....	314	14.3.3	学生信息查找函数.....	355
12.1.1	函数模板语法.....	314	14.3.4	学生信息编辑函数.....	357
12.1.2	函数模板实例化.....	315	14.3.5	学生信息统计函数.....	361
12.1.3	使用函数模板实例.....	317	14.3.6	学生成绩排序功能.....	363
12.2	类模板.....	318	14.3.7	文件读/写功能.....	368
12.2.1	类模板的语法.....	318	14.4	本章小结.....	369
12.2.2	类模板实例化.....	320	第 15 章	工资发放系统.....	371
12.2.3	类模板的默认参数.....	321	15.1	总体设计.....	372
12.3	标准模板库.....	322	15.1.1	类的设计.....	372
12.4	异常处理.....	324			

15.1.2	虚函数的分析设计.....	372	16.3.6	客房信息浏览函数.....	398
15.2	工资单系统源代码.....	373	16.3.7	信息查询函数.....	399
15.2.1	主函数源代码.....	374	16.3.8	信息删除和修改函数.....	404
15.2.2	基类 Employee 源代码.....	375	16.3.9	客房信息插入函数.....	407
15.2.3	派生类 Manager 的源代码	376	16.3.10	退出系统函数.....	408
15.2.4	派生类 CommissionWorker 的源代码.....	377	16.3.11	其他辅助函数.....	409
15.2.5	派生类 HourlyWorker 的 源代码.....	379	16.4	系统运行界面.....	412
15.2.6	派生类 PieceWorker 的 源代码.....	380	16.5	本章小结.....	415
15.3	本章小结.....	381	第 17 章	图书管理系统.....	417
第 16 章	旅店管理系统.....	383	17.1	功能设计要求.....	418
16.1	功能需求.....	384	17.2	总体设计.....	418
16.2	总体设计.....	384	17.2.1	类的设计.....	418
16.2.1	系统功能结构.....	384	17.2.2	系统功能结构.....	421
16.2.2	数据结构以及全局函数 设计.....	385	17.3	图书管理系统源代码.....	422
16.3	旅店管理系统实现.....	386	17.3.1	类的定义.....	422
16.3.1	系统头文件.....	387	17.3.2	类的成员函数实现.....	424
16.3.2	显示界面函数.....	388	17.3.3	用户界面相关函数.....	432
16.3.3	客房信息录入函数.....	392	17.3.4	读者维护模块.....	433
16.3.4	入住登记函数.....	394	17.3.5	图书维护模块.....	434
16.3.5	退房结算函数.....	396	17.3.6	系统管理员功能模块.....	435
			17.3.7	普通管理员功能.....	436
			17.3.8	系统 main 函数.....	438
			17.4	本章小结.....	439





第1章

C++概述

学前提示

C++语言是一种高效实用的程序设计语言，也是一种使用非常广泛的计算机编程语言。使用它既可进行过程化程序设计，也可进行面向对象程序设计。本章主要介绍计算机语言的发展过程、C++程序的基本结构以及程序开发步骤。

知识要点

- 程序设计语言的发展
- C++程序结构
- C++程序开发步骤

1.1 程序设计语言的发展

计算机语言通常是能完整、准确和规则地表达人们的意图，并用以指挥或控制计算机工作的“符号系统”。当使用计算机解决问题时，首先将解决问题的方法和步骤按照一定的顺序和规则用计算机语言描述出来，形成指令序列，然后由计算机执行指令，完成所需的功能。

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程。

1.1.1 机器语言阶段

机器语言是第一代计算机语言。众所周知，在计算机内部采用二进制表示信息。机器语言(Machine Language)是用二进制代码表示的计算机能直接识别和执行的一种机器指令系统。用机器语言编写的程序称为机器语言程序(又称目标程序)。每一条机器指令的格式和含义都是由设计者规定的，并按照这个规定设计制造硬件。一个计算机系统全部机器指令的总和，称为指令系统。不同类型的计算机的指令系统不同。

例如，某种计算机的指令为：

```
1011011000000000    表示进行一次加法操作
1011010100000000    表示进行一次减法操作
```

它们的前八位表示操作码，而后八位表示地址码。从上面两条指令可以看出，它们只是在操作码中从左边第0位算起的第6和第7位不同。这种机型可以包含 $256(=2^8)$ 个不同的指令。

用机器语言编写的程序，能直接在计算机上运行，运行的速度快，效率高。但机器语言难于记忆，也难于操作，代码编程繁琐、易错；而且编写的程序紧密依赖于计算机硬件，程序的可移植性差。

1.1.2 汇编语言阶段

为了克服机器语言的缺点，使语言便于记忆和理解，人们采用能反映指令功能的助记符来表达计算机语言，称为汇编语言(Assembly Language)。汇编语言采用助记符，比机器语言直观、容易记忆和理解。汇编语言也是面向机器的程序设计语言，每条汇编语言的指令对应了一条机器语言的指令，不同类型的计算机系统一般有不同的汇编语言。

例如，用汇编语言编写的程序如下：

```
MOV    AL    17D        //将十进制数 17 送往累加器
SUB    AL    10D        //从累加器中减去十进制数 10
...
```

汇编语言是第二代计算机语言。用汇编语言编写程序比用机器语言要容易得多，但计算机不能直接执行汇编语言程序，而要由一种程序将汇编语言翻译成机器语言，这种起翻译作用的程序叫汇编程序。

汇编程序是系统软件中的语言处理系统软件。汇编程序把汇编语言翻译成机器语言的过程称为汇编。汇编过程是由计算机运行汇编程序自动完成的,如图 1.1 所示。

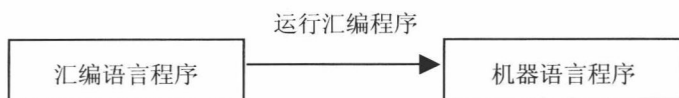


图 1.1 汇编过程

在计算机语言系统中,汇编语言仍然列入“低级语言”的范畴,它仍然依赖于计算机的硬件,移植性差。但汇编语言比起机器语言,在很多方面都有优越性,如编写容易、修改方便、阅读简单、程序清楚等。针对计算机硬件而编制的汇编语言程序,能准确地发挥计算机硬件的功能和特长,程序精炼而且质量高,所以至今仍是一种常用的程序设计语言。

1.1.3 高级语言阶段

机器语言和汇编语言都是面向机器(计算机硬件)的语言(低级语言),受机器硬件的限制,通用性差,也不容易学习,一般只适用于专业人员。后来人们逐渐意识到,应该设计一种语言:它接近于数学语言或自然语言,同时又不依赖于计算机的硬件,编出的程序能在所有的计算机上通用。这就是后来产生的高级语言(High-level Language)。高级语言是第三代计算机语言。目前广泛应用的高级语言有多种,如 BASIC、Pascal、C、C++、Java 及 C#等。

用 C++语言编写的程序片段如下:

```

int m, n, q;           //声明变量 m、n、q
cin >> m >> n;        //输入 m、n 的值
q = m + n;             //将 m 和 n 的值相加,结果赋给变量 q
cout << q              //输出相加的结果 q
  
```

使用高级语言编写程序时,不需要了解计算机的内部结构,只要告诉计算机“做什么”即可。至于计算机用什么机器指令去完成(即“怎么做”),编程者不需要关心。高级语言是面向用户的。

高级语言编写的程序叫作高级语言源程序,计算机无法直接执行,必须翻译或解释成机器语言目标程序,才能被计算机执行。翻译过程分为两步,即编译和连接,翻译过程如图 1.2 所示。



图 1.2 编译和连接过程

在图 1.2 中,高级语言经过编译后,得到目标程序(.obj),再与库程序连接,生成可执行程序(.exe)。

程序设计语言从机器语言到高级语言的抽象,带来的主要好处如下。



(1) 高级语言接近自然语言，易学、易掌握，一般工程技术人员只要经过几周时间的培训，就可以胜任程序员的工作。

(2) 高级语言为程序员提供了结构化程序设计的语法，使得设计出来的程序可读性好，可维护性强，可靠性高。

(3) 高级语言远离机器语言，与具体的计算机硬件关系不大，因而所写出来的程序可移植性好，代码重用率高。

(4) 由于把繁杂琐碎的事务交给了编译程序去做，所以自动化程度高，开发周期短，使程序员得到解脱，可以集中时间和精力去设计算法和从事更为重要的创造性劳动，以提高程序的质量。

1.1.4 从 C 到 C++

C 语言问世于 20 世纪 70 年代，发明人是 AT&T 贝尔实验室的 Dennis Ritchie。它最初用于编写和维护 Unix 操作系统(在此之前，Unix 系统程序要么只能用汇编语言来写，要么只能用 B 语言来写。B 语言由 Ken Thompson 发明，他是 Unix 的缔造者)。

C 语言是一种常规用途的语言，可用来写任何形式的程序，但 Unix 操作系统直接促成了它的成功和普及。如果你想维护 Unix 操作系统，就需要使用 C 语言。C 语言与 Unix 的配合是如此天衣无缝，以至于发明以后不久，不仅系统程序，就连 Unix 下运行的几乎所有商业程序都开始用 C 语言来编写。随着 C 语言越来越流行，逐渐出现了为其他流行操作系统编写的 C 语言版本；C 语言的应用开始不受 Unix 机器的局限。

C 语言的特殊性在于它虽是一种高级语言，但包含了低级语言的大量特点。可以说，它处于一种非常高级的语言和一种低级语言之间，优点和缺点都很突出。另一方面，C 语言又具有高级语言的许多特点，这使其比汇编语言更容易理解和编写，使 C 语言成为编写系统程序的理想选择。但对于其他程序(有时甚至包括一些系统程序)，C 语言并不像其他语言那样容易理解；另外，它不像另外一些高级语言那样提供了对自动检查的完善支持。

为了应对上述问题及 C 语言的另外一些缺陷，AT&T 贝尔实验室的 Bjarne Stroustrup 在 20 世纪 80 年代初开发了 C++ 语言。Stroustrup 将 C++ 设计成一种更好的 C。C 的大多数特性都成为 C++ 的一个子集，所以大多数 C 程序其实也是 C++ 程序(反之则不成立；许多 C++ 程序都绝对不是 C 程序)。与 C 语言不同，C++ 具备了“面向对象编程”(OOP)的能力，是一种非常强大的编程技术。

1.2 简单的 C++ 程序介绍

下面我们通过一个简单的程序，来说明 C++ 程序的基本结构。

【例 1.1】 一个简单的 C++ 程序。代码如下：

```
#include <iostream>
using namespace std;
void main()
{
    cout << "This is a C++ program."; //在屏幕上输出一行文字
```