高等职业院校

技能型紧缺人才培养培训系列教材

程序设计语言C++及开发应用

（计算机应用与软件技术专业）

主编　眭碧霞

高等教育出版社

高等职业学校
技能型紧缺人才培养培训系列教材

程序设计语言C++及开发应用

（计算机应用与软件技术专业）

主编　眭碧霞

高等教育出版社

内容简介

本书根据教育部《高等职业教育计算机应用和软件技术专业领域技能型紧缺人才培养指导方案》编写。

本书按项目教学法组织教材。全书共 11 章，第 1 章引入面向对象方法，第 2 章介绍创建对象的一般方法，第 3 章引入运算符并介绍判定结构，第 4 章介绍函数与类，第 5 章介绍构造函数和析构函数的功能、声明与应用，第 6 章介绍指针和引用，第 7 章介绍类的继承，第 8 章介绍对象的多态性，第 9 章介绍流类库与输入输出，第 10 章介绍函数模板、类模板和 C++标准模板库，第 11 章介绍 C++中的异常处理。

本书配有《C++习题与实训》，帮助学习者更好地掌握学习内容。

本书适合作为五年制高等职业院校教材，也可作为各类培训班教材。

图书在版编目(CIP)数据

程序设计语言 C++及开发应用/眭碧霞主编. —北京:高等教育出版社,2004.7

ISBN 7-04-015167-7

Ⅰ.程... Ⅱ.眭... 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(2004)第 053634 号

策划编辑 李 波 **责任编辑** 焦建虹 **封面设计** 刘晓翔 **责任绘图** 杜晓丹
版式设计 范晓红 **责任校对** 金 辉 **责任印制** 陈伟光

出版发行	高等教育出版社	**购书热线**	010-64054588
社 址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010-82028899		http://www.hep.com.cn
经 销	新华书店北京发行所		
印 刷	北京民族印刷厂		
开 本	787×1092 1/16	**版 次**	2004 年 7 月第 1 版
印 张	16.75	**印 次**	2004 年 7 月第 1 次印刷
字 数	400 000	**定 价**	20.80 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

出版说明

为了贯彻《国务院关于推进职业教育改革与发展的决定》的精神，促进职业教育更好地适应社会主义现代化建设对生产、服务第一线技能型人才的需要，教育部、劳动和社会保障部、国防科工委、信息产业部、交通部、卫生部联合发出了关于实施“职业院校制造业和现代服务业技能型紧缺人才培养培训工程”的通知。

根据“工程”的精神，教育部、信息产业部联合推出了《高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》，对职业教育教学改革提出了新的要求。即：职业教育是就业教育，要按照职业教育本身所固有的规律，在借鉴国内外成功经验的基础上，建立具有鲜明职业教育特点的课程体系。方案强调照顾学生的经验，强调合作与交流，强调多种教学方式交替使用，强调教师是学生学习过程的组织和对话伙伴。

为了帮助职业院校教师理解新的教学理念，更好地实施技能型紧缺人才培养计划，在深刻理解新的教学指导方案的基础上，高等教育出版社率先出版一套计算机应用与软件技术专业领域教材，以期帮助教师理解方案和组织教学，其特点有：

1. 借鉴国外先进的职业教育经验

研究了国外职业教育的各种模式，如英国的 BTEC 模式，印度的 NIIT 模式，澳大利亚的 TAFE 模式等，学习借鉴这些模式的优秀之处，又不拘泥于某种模式。

2. 协作式学习方式

强调以学生的团队学习为主，学生分成小组共同就某些问题进行讨论。同时认为学习与思考同等重要。在有限的时间内，使学生最大限度地掌握技能，并掌握自主学习的方法，为其今后的知识和能力拓展打下良好的基础。通过这种方法，有效地培养学生的沟通能力，如口头表达能力、书面表达能力、理解他人的能力和发表自己见解的能力。

3. 采用项目教学法组织教材

通过项目的活动过程培养学生的分析问题能力，团队精神，法律意识，沟通能力。每个项目相对较小，使学生对单个项目的学习过程不会太长，以减少学生的学习难度，提高学习兴趣。

4. 精心组织教材开发队伍

邀请教育专家、计算机专家、企业人士、职教教师共同参与项目开发，特别注意吸收双师型教师参加。

5. 根据项目特点设计课程解决方案

教材的组织是一个项目的解决方案，不是知识的细化，不以教会学生知识为目标，而以帮助学生掌握项目实施过程为目的。

6. 提供分层教学

书中实训指导、作业编排有一定梯度，以适应不同类别，不同能力学生的需要。

7. 配套完备的教学解决方案

教材出版的同时，与之配套的电子教案及与教材相关的素材将通过网站：http://sv.hep.com.cn公布，供任课教师免费下载。

通过以上方式，高等教育出版社将为职业院校师生提供精良的教学服务，有不完备的地方也欢迎广大的职业院校的师生给予批评指正。

高等教育出版社

2004年5月

前　言

为配合教育部“技能型紧缺人才培养培训工程”的实施，高等教育出版社组织教育专家、职业教育一线的骨干教师、企业的工程技术人员和培训工程师根据技能型人才培养模式的要求编写了一套适用于职业教育的教材。教材在形式上按项目进行组织，在内容上主要选择生产生活中实用的案例展开讲解，使职业技能训练与常规教学活动有机结合。教材出版的同时，与本书配套的电子教案及与教材相关的素材将通过网站：http://sv.hep.com.cn 公布，供任课教师免费下载。

面向对象技术是目前软件工业的主流，绝大多数系统、应用程序都是采用面向对象技术开发出来的。面向对象的程序设计语言常常是由几个重要的部分和概念组成的，这些概念让面向对象成为目前最流行的程序语言规格。C++语言就是一种面向对象的程序设计语言。

可以毫不夸张地说，C 及 C++语言是当今世界使用人数最多的程序设计语言之一。C 语言兼有高级语言的易于理解、易于使用、易于学习以及低级语言强大的控制功能等特点。正是这一特点，使得许多程序设计者习惯并喜欢使用 C 语言进行程序设计。而 C++则继承了 C 易学习、功能强的特点，加上其面向对象的特点，使其得到众多程序设计者的青睐。

常见的国内教材的编排体例是首先给出一个简单程序，给学生一个感性认识，然后严格按知识点排列，先是各种数据类型，接着是各种表达式、语句，再是函数、类、程序结构等。虽然逻辑严密，但在学习的过程中，难见全貌，由浅入深的渐进性不强。本书采用项目教学法，开篇介绍面向对象编程（C++）的由来、优点及过程，然后介绍一个项目背景、项目要求等。这个项目背景贯穿全书，每个章节都可看做是完成这个项目的某一部分，在完成这一部分的同时，学习相关的知识点。这些知识点又分为两类，一是书中介绍的最重要、最常用的部分，二是教材推荐、由同学自己学习掌握的部分。知识点的排列并不求全，着重是由浅入深引领学生完成项目所要求的内容。

编者在教材的编写中，遵循“实用、适当、先进”的原则，依照“通俗、精练、可操作”的编写风格，考虑到职业院校学生的特点与基础，从实际出发、从实用出发组织内容，对学生的学习有一定的指导意义。

本套教材由两本书组成，分别是《程序设计语言 C++及开发应用》和《C++习题与实训》。全书共 11 章，第 1 章引入面向对象方法，第 2 章介绍创建对象的一般方法，第 3 章引入运算符并介绍判定结构，第 4 章介绍函数与类，第 5 章介绍构造函数和析构函数的功能、声明与应用，第 6 章介绍指针和引用，第 7 章介绍类的继承，第 8 章介绍对象的多态性，第 9 章介绍流类库与输入输出，第 10 章介绍函数模板、类模板和 C++标准模板库，第 11 章介绍 C++中的异常处理。

为方便读者使用，主教材的最后有 3 个附录，分别给出了 C++中的关键字、运算符优先级和 ASCII 字符集。

本书是为高等职业院校学生编写的，也可以作为各类程序员培训班的教材。要求本书的学习者有一定的计算机操作基础，至少学习过一种编程语言。

本书由眭碧霞主编，简勇编写第1、2、3、4章，裴拯编写第5、6、7章，叶青松编写第8、9、10、11章。

赵佩华老师对本书的编写给予了大力支持，并在百忙中审阅了全书，提出了许多宝贵的意见，提高了本书的质量，在此表示衷心的感谢。

由于作者水平有限，书中不妥、错误之处在所难免，敬请读者不吝赐教。

编　者

2004年3月

目　录

第 1 章

引入面向对象方法

对每个希望学习 C++的人来说，习惯 C++需要一些时间；对于已经熟悉 C 的程序员来说，这个过程尤其令人苦恼。因为 C 是 C++的子集，几乎所有 C 中采用的技术都可以在 C++中继续使用，但很多功能用起来又不太合适。例如，C++程序员会认为指针的指针看起来很古怪，他们会问：为什么不用指针的引用来代替呢？

C 语言是一种简单的语言。它真正提供的只有宏、指针、结构、数组和函数。不论对于什么问题，C 都依靠宏、指针、结构、数组和函数来解决。C++则不然，在 C++中，宏、指针、结构、数组和函数当然还存在，此外还有私有和保护型成员、函数重载、缺省参数、构造和析构函数、自定义操作符、内联函数、引用、友元、模板、异常、名字空间等。用 C++比用 C 具有更宽广的空间，设计时可以考虑更多的选择。

在面对众多的选择时，许多 C 程序员墨守成规。一般来说，这也不是什么很大的过错，但某些 C 的习惯有悖于 C++的精神本质。也有人认为，C++语言仅仅是 C 语言的扩充，那就完全错了。C++与 C 的兼容性使它具有双重特点，但如果在使用 C++时仍然保持 C 的习惯及思维方式，那就不合适了。诚然作为 C 的超集，C++在技术上是和 C 完全兼容的，但它在概念上是和 C 完全不同的语言，因此程序员应该学会按 C++自己的方式使用它。不仅会使用 C++的编译器，更重要的要掌握 C++的思维方式、C++的设计方法和 C++的习惯。

C 语言是一种“中级语言”，既有低级语言的操作能力，又有高级语言的程序思想和设计方法，是一种结构化的程序设计语言。目前几乎所有流行的操作系统都是 C 语言的杰作。

C++语言是一种应用较广的面向对象的程序设计语言，可以用来实现面向对象的程序设计。面向对象的设计与面向过程的设计是有很大区别的，面向对象的程序设计是在面向过程的程序设计的基础上一个质的飞跃。要学会面向对象的程序设计，首先要学会一种面向对象的语言，即要学会用 VC 编程，就要先有 C++的基础，而学习 C++语言首先要认识它面向对象的特性和实现面向对象的方法。

C++语言包括过程性语言部分和类部分。对于过程性语言部分，C++与 C 没有本质的区别，只是版本提高了，语言的功能增强了；而类部分却是 C 语言中所没有的，是面向对象程序设计的主体。要学习面向对象的程序设计，首先必须具有过程性语言的基础。所以学习 C++，首先必须学习其过程性语言部分，然后再学习类部分。也就是说，先学习高版本的 C，再学习类。由于 C++和 C 都有过程性语言这一部分，所以学习 C++，不必一定从 C 开始，可以直接从 C++

开始。当然，C 语言程序设计的经验对于程序员来说非常有意义，因为 C 程序设计开发锻炼了程序员的抽象程序设计能力，这正是 C++更为抽象的概念和技术的基础。

1．C++对面向对象程序设计方法的支持和实现

（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。对象被说明为一个给定类的变量，类有些类似于 C 语言中的结构，在 C 语言中可以定义结构，但这种结构中包含数据，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，但是它没有私有或保护的成员。

（2）C++类中包含私有、公有和保护成员

C++类中可定义 3 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；一种是公有(Public)成员，类外的函数也可访问公有成员，公有成员成为该类的接口；还有一种是保护(Protected)成员，这种成员只有该类的派生类可以访问，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，前者称为友元函数，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特性。

（5）C++允许函数名和运算符重载

C++支持多态性，C++允许一个相同的标识符或运算符代表多个不同实现的函数，这称为标识符或运算符的重载，用户可以根据需要定义标识符重载或运算符重载。

（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。

以上所讲的是 C++对面向对象程序设计中一些主要特征的支持。

2．C++的词法及词法规则

（1）C++的字符集

字符是一些可以区分的最小符号。C++的字符集由大小写英文字母(a~z 和 A~Z)、数据字符

(0~9)、特殊字符(如空格、!、#、%、^、&、*、_、<、>、？、\等)组成。

（2）单词及词法规则

单词又称词法记号，它是由若干个字符组成的具有一定意义的最小词法单元。C++共有 6 种单词，分别是标识符、关键字、运算符、分隔符、常量和注释符，在编码时要特别注意这些单词的词法规则。要注意的是 C++中的空白符。C++中经常使用空白符，实际上，空白符不是一个字符，它是空格符、换行符和水平制表符的统称。注意，空白符不等于空格符，只是空白符包含空格符。还有一个空字符，要把它与空白符分开。空字符是指 ASCII 码值为 0 的那个字符。空字符在 C++中有特殊用途，用它来作为字符串的结束符。存放在内存中的字符串常量都在最后有一个结束符，即空字符，它用转义序列方法表示 '\0'。

3．C++程序结构的组成部分

（1）预处理命令

C++提供了 3 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的语句，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++程序中的语句以分号结束。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。广义地讲，对象包含了变量，即把变量也称为一种对象；狭义地讲，将对象看做是类的实例，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++中尽量都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。

几乎所有的程序设计都要讲究程序设计的风格，这既是软件工程所要求的，也是作为一个程序员所必备的素质。

4．C++程序设计的风格

（1）C++程序的书写格式

在编程时应该注意 C++程序的书写格式，基本原则是：一行通常写一条语句。短语句可以一行写多个；长语句可以一条写多行。分行的原则是不能将一个单词分开。用双引号引用的一个字符串最好不分开，如果一定要分开，有的编译系统要求在行尾加续行符("\")。

（2）C++程序的实现

C++源程序的实现与其他高级语言源程序实现的原理是一样的。一般都要经过编辑、编译和运行。其中最重要的是编译过程，C++是以编译方式实现的高级语言。C++程序的实现，必须使用某种C++语言的编译器对程序进行编译。编译器的功能是将程序的源代码转换成机器代码的形式（即目标代码），然后，再使目标代码进行连接，生成可执行文件。该过程可分为3个子过程：预处理过程、编译过程(词法分析、语法分析、符号表、错误处理程序、生成目标代码)和连接过程。

1.1 计算机程序设计语言的发展

自1946年世界上第一台电子计算机问世以来，计算机科学及其应用的发展十分迅猛，计算机被广泛地应用于人类生产、生活的各个领域，推动了社会的进步与发展。特别是随着互联网（Internet）日益深入千家万户，传统的信息收集、传输及交换方式正被革命性地改变，人们已经难以摆脱对计算机的依赖，计算机已将人类带入了一个新的时代——信息时代。新的时代的基本要求之一是：自觉地、主动地学习和掌握计算机的基本知识和基本技能，并把它作为自己应该具备的基本素质。要充分认识到，缺乏计算机知识就是信息时代的“文盲”。对于理工科的大学生而言，掌握一门高级语言及其基本的编程技能是必须的。大学学习，除了掌握本专业系统的基础知识外，科学精神的培养、思维方法的锻炼、严谨踏实科研作风的养成以及分析问题和解决问题能力的训练，都是日后工作的基础。学习计算机语言，正是一种十分有益的训练方式，而语言本身又是与计算机进行交互的有力工具。一台计算机是由硬件系统和软件系统两大部分构成的，硬件是物质基础，而软件可以说是计算机的灵魂，没有软件，计算机是一台“裸机”，是什么也不能干的，有了软件，才能活动起来，成为一台真正的“电脑”。所有的软件，都是用计算机语言编写的。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．机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数，二进制是计算机语言的基础。计算机发明之初，人们只能用计算机的语言去命令计算机，即写出一串串由“0”和“1”组成的指令序列交由计算机执行，这种语言就是机器语言。使用机器语言是十分痛苦的，特别是在程序有错而需要修改时，更是如此。而且，由于每台计算机的指令系统往往各不相同，所以在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，造成了重复工作。但由于机器语言是针对特定型号计算机的语言，故而运行效率是所有语言中最高的。机器语言是第一代计算机语言。

2．汇编语言

为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进，即用一些简洁的英文字母、符号串来替代一个特定指令的二进制串，例如用“ADD”代表加法，“MOV”代表数据传递，等等。这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，负责将这些符号翻译成二进制数表示的机器语言，这种翻译程序被称为

汇编程序。汇编语言同样十分依赖于机器硬件，移植性不好，但效率仍十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精炼而质量高，所以至今仍是一种常用的强有力的软件开发工具。

3．高级语言

从最初与计算机交流的痛苦经历中，人们意识到，应该设计这样一种语言，它接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过努力，1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了。40 多年来，相继出现几百种高级语言，有重要意义的有几十种，影响较大且使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、Java 等。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的过程。相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

20 世纪 60 年代中后期，软件越来越多，规模越来越大，而软件生产基本上是各自为战，缺乏科学规范的系统规划与测试、评估标准，其恶果是大批耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失。软件给人的感觉是越来越不可靠，以至几乎没有不出错的软件。这一切极大地震动了计算机界，历史上称为“软件危机”。人们认识到：大型程序的编制不同于编写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性，也便于验证正确性。1969 年，结构化程序设计方法被提出。1970 年，第一个结构化程序设计语言——Pascal 语言出现，它标志着结构化程序设计时期的开始。

20 世纪 80 年代初开始，在软件设计思想上，又产生了一次革命，其成果是面向对象的程序设计。在此之前的高级语言几乎都是面向过程的，程序的执行是流水线似的，在一个模块被执行完成前，人们不能干别的事，也无法动态地改变程序的执行方向。这与人们日常处理事物的方式是不一致的。人们希望发生一件事就处理一件事，也就是说，不能面向过程，而应是面向具体的应用功能，也就是对象（object）。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称之为软件集成块。它们与具体应用无关，但能相互组合，完成具体的应用功能，同时又能重复使用。对使用者来说，只关心它的接口（输入量、输出量）及能实现的功能，至于如何实现的，那是它内部的事，使用者可以完全不用关心。C++、VB、Delphi 就是面向对象程序设计语言的典型代表。

高级语言的下一个发展目标是面向应用，也就是说只需要告诉程序要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。

C++包含了整个 C，C 是建立 C++的基础。C++包括 C 的全部特征和优点，同时添加了对面向对象编程（OOP）的完全支持。1980 年，贝尔实验室的 Bjarne Stroustrup 开始对 C 进行改进和扩充，1983 年正式命名为 C++。在经历了 3 次 C++修订后，1994 年制订了 ANSI C++标准的草案，以后又经过不断完善，成为目前的 C++。

C++仍在不断发展中。美国微软公司现已准备推出 C#（C Sharp）语言，来代替 C++语言。

1.2 面向对象方法学

1.2.1 软件系统的复杂性

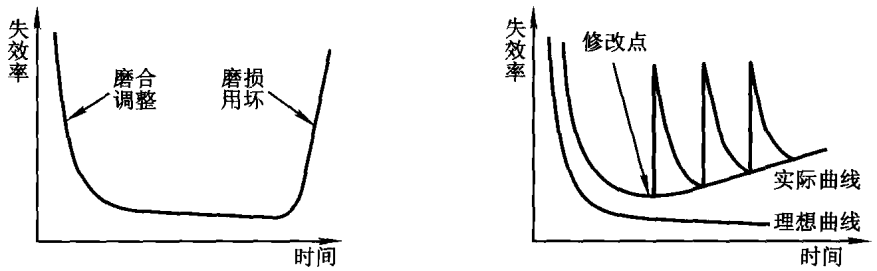
软件是计算机系统中与硬件相互依存的另一部分，它是包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文资料。

软件的特点有：

① 软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性。

② 软件的生产与硬件不同，它没有明显的制造过程。对软件的质量控制必须注重在软件开发方面下功夫。

③ 在软件的运行和使用期间，没有硬件那样的机械磨损、老化问题。任何机械、电子设备在运行和使用中，其失效率大都遵循如图 1-1（a）所示的U形曲线（即浴盆曲线）。而软件的情况与此不同，因为它不存在磨损和老化问题。然而它存在退化问题，必须要多次修改和维护，如图 1-1（b）所示。



（a）硬件失效率曲线　　（b）软件失效率曲线

图 1-1　失效率曲线

④ 软件的开发和运行常常受到计算机系统的限制，对计算机系统有着不同程度的依赖性。为了解除这种依赖性，在软件开发中提出了软件移植的问题。

⑤ 软件的开发至今尚未完全摆脱手工的开发方式。

⑥ 软件本身是复杂的。软件的复杂性可能来自它所反映的实际问题的复杂性，也可能来自程序逻辑结构的复杂性。

⑦ 软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，它的成本是比较高的。

⑧ 相当多的软件工作涉及社会因素。许多软件的开发和运行涉及机构、体制和管理方式等问题，甚至涉及人的观念和人们的心理，这些社会因素直接影响到项目的成败。

1.2.2 简化系统的复杂性

软件系统的复杂性是不可能完全摆脱的。要是人类担心构造和维护航天飞机的复杂性，那

人类就会满足于初等的滑翔机飞行了。然而，有一些办法可以保证这种复杂性不会是软件正常运行的绊脚石。简化这种复杂性的一种方法是把系统分解，分解为它的组成部件并且以某种层次结构来安排这些部件。就像一个大的跨国公司的管理，总裁下面分设若干副总裁，副总裁下面是事业部，接着是分支机构，最后是分支机构的员工，按层次结构排列。

为了简化软件系统的复杂性，1968 年在联邦德国召开的国际会议上启用了“软件工程”这个概念，陆续提出了 100 多条关于软件工程的准则或“信条”。随着时间的推移，逐渐形成几种著名的软件开发方法：结构化方法、Jackson 方法、维也纳开发方法、面向对象的开发方法。

结构化程序设计（Structured Programming，SP）方法是由 E.Dijkstra 等人于 1972 年提出来的，它建立在 Bohm、Jacopini 证明的结构定理的基础上。结构定理指出：任何程序逻辑都可以用顺序、选择和循环等 3 种基本结构来表示。

结构化程序设计方法有 2 个要点：

① 采用“自顶向下逐步求精”的设计思想。

② 程序中只采用顺序、循环和选择 3 种基本控制结构。

正是由于结构化程序设计的出现，在一定的程度上缓解了“软件危机”，对于软件工程做出了很大的贡献。但是随着软件规模越来越大，这种程序设计方法就显现出稳定性低、可修改性和可重用性差的弊端。面向对象的开发方法以其特有的先进性摈弃了上述这些弊端，能够有效地改进结构化程序设计中存在的问题，成为目前软件开发方法的主流，现在流行的编程工具几乎都是面向对象的。

1.2.3 面向对象开发方法的基础

面向对象的开发方法的基本出发点是尽可能按照人类认识世界的方法和思维方式来分析和解决问题。客观世界由许多具体的事物、事件、概念和规则组成，这些都可以看成对象。面向对象方法正是以对象作为最基本的元素，也是分析问题、解决问题的核心，它符合人类的认识规律。计算机中实现的对象与现实世界的对象有一一对应的关系，不用作任何转换。正是由于这点，使面向对象容易为人们所理解、接受和掌握。

理想的状态下，完成一个程序设计任务分为三步。第一，必须清楚地理解问题(分析，Analysis)；然后，要定义在解决方案中关键的概念（设计，Design）；最后，必须以程序的方式表达出解决方案（编程，Programming）。在大部分程序中，有一些（关键）概念是不容易通过基础类型或函数来表达的。这样一些概念，可以在程序中用类来表示。一个 C++类就是这样一种类型（Type），它指出类的实例是怎样动作的，它们怎样被创建、怎样被操作、怎样被销毁。类还可以指出对象（类的实例）怎样表现，虽然在程序设计的早期它不是主要的部分。写一个好的程序的关键是设计一个好的类，它能够清楚地表达出一个单独的概念。

面向对象的开发方法包括：面向对象分析 OOA、面向对象设计 OOD 和面向对象实现 OOP。

面向对象的程序设计方法强调直接以问题域（现实世界）中的事物为中心来思考和认识问题，并按照这些事物的本质特征把它们抽象为对象，以作为构成软件系统的基础。

（1）对象（Object）

每个对象都具有属性（Attribute）和方法（Method）这两方面的特征。对象的属性描述了