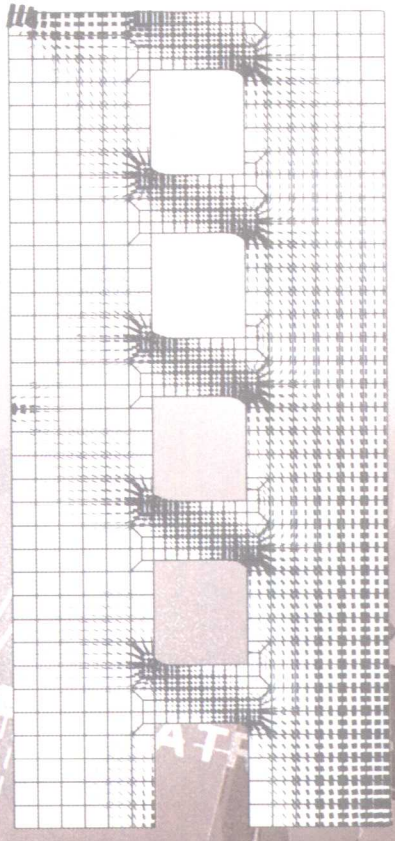
吴晓涵　编著

高等院校土木工程系列教材

面向对象结构分析程序设计



科学出版社

内 容 简 介

面向对象技术正在成为当前程序设计的主流。本书系统地讲述了采用面向对象编程技术进行结构分析程序设计的方法，内容包括面向对象技术的基本概念、C++在面向对象编程中的运用技巧、Windows 应用程序的面向对象可视化编程的基本方法、结构对象的分析和设计、面向对象结构静力和动力分析程序设计，以及将结构对象嵌入 Windows 应用程序框架，建立基于 Windows 的结构分析程序过程。书中还介绍了运用 Windows 窗口进行结构分析前后处理的方法。

本书可作为高等院校的力学、土建、水利、岩土、机械等专业的本科生和研究生的教材或参考书，也可供上述专业的教师和工程技术人员参考。

图书在版编目(CIP)数据

面向对象结构分析程序设计/吴晓涵编著. —北京：科学出版社，2002
ISBN 7-03-010162-6

Ⅰ.面… Ⅱ.吴… Ⅲ.面向对象语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 008922 号

科学出版社 出版
北京东黄城根北街16号
邮政编码:100717
http://www.sciencep.com
源海印刷厂 印刷
科学出版社发行 各地新华书店经销
*
2002 年 4 月第 一 版 开本:720×1000 B5
2002 年 4 月第一次印刷 印张:28
印数:1—3 000 字数:536 000

定价:42.00 元

(如有印装质量问题，我社负责调换〈兰各〉)

前　言

结构计算分析在工程应用和研究中占有相当大的比重，随着计算机技术的快速发展，结构的计算分析技术有了相当大的提高。目前，国内外已研制出许多功能强大的结构分析程序，如 ANSYS、SAP、ADINA、ABAQUS、STRAND 等。但是，这些结构分析软件不可能解决所有问题，而且，随着工程技术的发展和研究的深入，还会不断涌现出新的待解问题。因此，专业技术人员掌握一定的结构分析程序设计的原理和方法还是很有必要的。

然而，结构分析程序所涉及的计算内容繁多，以传统的结构化编程方式编制结构分析程序需花费大量的工时，在可扩充性和代码可重用性方面受到许多制约。面向对象编程技术的出现，使软件编程摆脱了这些不利因素。由于采用了统一的对象分类方法，编程过程将类似采用标准部件组装设备一样，由对象拼装成完整的程序，编程人员可通过继承方式派生出新的对象类，在新对象类中添加新的操作功能，建立满足自己特定需求的程序。由于具有诸多的优点，面向对象编程方法正在成为程序设计的主流，本书就是在这一发展要求的前提下进行编著的。目前，已经有许多成熟的结构对象类库，读者在掌握了面向对象编程技术后，可利用这些类库，方便快捷地建立结构分析程序。

全书共分七章。

第一章概述了面向对象技术的基本概念，对象类的基本特点，面向对象的结构程序设计方法，介绍了面向对象程序的编程语言和开发工具。

第二章介绍了 C＋＋语言的相关技术在面向对象程序设计中的运用，以及矩阵类的设计方法。

第三章讲述了 Windows 应用程序的基本结构、运行原理和设计方法，介绍了 Visual C＋＋的基本使用方法。

第四章论述了结构对象的分析和设计方法，详细介绍了各结构对象类，以及通过对象进行结构分析的步骤。讲述了运用这些结构对象和 Windows 应用程序框架，建立基于 Windows 的面向对象结构分析程序 OOPFE 的方法和过程。

第五章深入讲解了面向对象技术在结构分析程序设计中的应用，包括通过继承建立新的单元类，运用多态性为 OOPFE 添加新的单元类等。

第六章介绍了结构分析的前后处理，包括 Windows 窗口的结构图形和数据显示、结构图形的 AutoCAD 图形交换文件格式的 DXF 文件输出。

第七章讲述了运用面向对象技术进行结构动力分析程序设计的方法，内容包括单元和总体结构质量矩阵的生成、结构模态分析，以及结构动力响应分析的程序

实现方法。

本教材在阐述基本概念、原理和方法的同时，注重实践和应用，所述内容均附有代码实例和分析实例，并提供了大量插图，以方便阅读理解。

教材中的基于 Windows 的面向对象结构分析程序 OOPFE 利用了 Windows 较强的图形绘制功能，对分析的结果采用图形形式予以直观反映，图形显示内容包括：杆件内力、平面单元主应力、结构变形和结构振动动态反应等，可直接用于结构分析课程的教学演示，或作为课程学习辅导软件。为方便结构建模，教材中还提供和介绍了 AutoCAD 环境下的结构辅助建模程序 FEEST。

教材中的程序代码、分析实例数据以及辅助建模程序 FEEST 可通过同济大学结构工程与防灾研究所网页 http://www.tongji.edu.cn/~risedr/免费下载，或联系作者的电子信箱 wuxhtj@sina.com 获取。

教材中的结构对象类的分析、抽象和程序建模是作者在工作实践和研究应用基础上建立起来的。读者可结合自己的分析要求、编程环境和嵌入平台，对教材中的结构对象模型进行调整，或选择其他的结构对象类库，建立更适合自己需要的结构对象类。

本教材可作为高等院校的力学、土建、水利、岩土、机械等专业的本科生和研究生的教材或参考书，也可供教师和工程技术人员参考。

本书的编写工作得到了吕西林教授、周德源教授和钱江教授的大力支持和帮助，特在此表示衷心的感谢。

由于本教材内容涉及面较广，程序 OOPFE 的代码量较大，难免会存在一些不足之处，有待进一步改进和优化，望读者批评指正和提供宝贵意见。

目　录

试读结束：需要全本请在线购买：　www.ertongbook.com

第一章 绪 论

最早的结构分析程序设计采用的是面向过程的方法，即针对某一分析对象，通过流程、顺序实现之。面向过程的程序设计要求程序设计员编写和熟悉程序内部的各个细节，编程方式好比以往的小农经济，每项生活和生产资料都靠自己制作生产。后来提出的结构化方法，将分析对象按功能分解成模块，由功能模块来实现具体的细节，而个个功能模块可由许多程序设计员分别编写，最终通过功能调用完成分析过程。结构化程序设计的分工与合作体现了社会化大生产的精神。虽然程序设计结构化从一定程度上提高了程序开发效率和可维护性，但对于程序的可再用性、可扩充性仍然提高不大，这主要是由两个方面的原因所造成的。首先，结构化方法采用了“面向任务”的指导思想，要在针对某一特定任务设计的程序中增加新的“任务”的话，程序的修改会涉及原有程序的方方面面，从而造成程序开发效率低下；其次，结构化程序中数据与功能模块相分离，程序在增加新的功能模块同时，要添加新的数据结构，模块的功能越强，数据量越大，两者的协调关系就越复杂。针对某一问题的数据结构，因其特殊性，一般很难不作修改就用于相近的问题。数据与功能模块相分离也使得模块的功能限制在较小的范围内，程序员只能用一些小的零部件进行程序装配，虽然避免了对每个零部件制造过程的了解，但需关心的程序结构层面相对还是较低的，这就要求程序员在整个程序的各个方面都是比较专业的。面向对象(object oriented)技术的出现克服了上述结构化程序设计的不足，为功能模块的集成化、可再用及程序扩充的灵活性提供了有利条件。

1.1 面向对象的概念

1.1.1 对象组成

面向对象是一种试图模仿现实世界的方法，它遵循认识方法学的基本原理，运用我们已形成了的对现实世界的概念、分类和抽象方法，将现实世界分成不同的对象类(class)，如“人”、“汽车”、“建筑”等，不同的对象的组合及其相互作用构成了我们要研究分析和构造的客观系统。每一对象类均享有一组属性(attribute)或行为特征，比如人的年龄、身高、体重和学历、性情等，结构构件的长度、截面积、组成材料等。这些对象的属性可根据施加在对象上的各种因素而变更。对象类(class)和对象(object)两者不是等同的，对象类是对具体对象的抽象(abstraction)，对象是对象类的实例化(instance)，比如，你、我、他是人类，我们每个人是人类这一对象类的单

个实例。结构分析模型中将杆件分成一类，在具体计算时将其实例化，有多少根杆件，就有多少个杆件实例，每根杆件有长短、截面尺寸和所属材料等各自的属性。对象的属性域可以是简单类型的数据，也可以是另一些对象。这种情况在现实世界里普遍存在，例如，一辆汽车就包括了动力、传动、刹车、油路和电路等多个子系统。包含其他对象作为其组成成分的对象称为复合对象或容器对象，相对于简单对象，复合对象能表达更复杂的属性数据。除了所包含的属性以外，对象还包括了各项操作(operation)，通过操作来变更和调节属性以及实现对象的各种功能。比如，调节汽车的调速度档，可改变传动速率；踩下油门，可使汽车开动等。

1.1.2 对象相互作用

现实世界的事物是相互联系产生作用的，模拟现实的对象也一样，系统运行时对象之间的作用是通过消息(message)来传递进行的。一个消息是由发送对象向接收对象发出的调用某个对象操作的请求信息，必要时还包括适当的数据参数传送信息。接受对象收到请求消息后，按请求信息中指定的操作内容和数据参数来执行操作。对象操作执行完成后的结果有时还要返回给发送对象。比如，你拿了一卷拍摄好的胶卷到洗印店去冲印，你给了店家要求冲印的信息并将胶卷传送给对方，待店家操作完以后，再将冲印好的胶卷或照片给你。在程序中，对象间传送的消息由三部分组成：①接收消息的对象名，好比上例中的相片洗印店；②调用操作名，好比上例中你要求店家是冲印胶卷还是提供其他服务；③必要的数据参数，如上例的胶卷。有一些对象操作是用来对对象内部属性进行调整和修改。比如，一台电视机上的许多按钮，可调节电视屏幕的亮度、对比度、色彩及电视频道等。对象实例的一组属性数据值构成了对象的一个状态，对象操作的执行可能会改变一个或几个对象属性值，也就是改变了对象状态。系统运行时一个对象从初始状态开始，在对象操作下对象状态会不断发生变化。但不管对象状态如何变化，对象的根本不会改变，这是由对象内部的构成所决定的。

1.2 对象和类的基本特点

1.2.1 封装(encapsulation)

封装是描述把对象的属性数据值和功能操作包装在一起构成一个具有类类型的对象的术语。封装的类有一个类名，程序中由类名说明对象实例。由于对象的封装性，外界只能见到对象界面上的消息，对象内部对外界是隐蔽的，其目的是将对象的使用者和对象的设计者分开，使用者不必知道对象功能实现的细节，而只需用设计者提供的消息来访问对象。就像你去洗印店印相片，你不用知道具体的印相过程；你选电视频道，而不用了解电视机内的变换过程。封装的另一个好处是类的内

部的改动不会影响外界其他部分，比如相片洗印店内印相工艺流程的改动或改进对顾客没有影响，顾客得到的还是相片。

1.2.2 继承(inheritance)

在现实世界中，许多实体或概念不是孤立的，它们具有共同的特征，但也有细微的差别，人们使用层次分类的方法来描述这些实体或概念之间的相似点和不同点。例如图 1.1 表示昆虫学家对昆虫进行分类之后形成的一个分类树。在这个分类树中，最高层是最普通的，每一层都比它之上的层更具体，并且低层包含有高层的特征。例如，一旦确定一个昆虫为蝴蝶目中的一个，我们就不必指出这个蝴蝶有一对翅膀，蝴蝶从它所属的类中继承了这一特征。

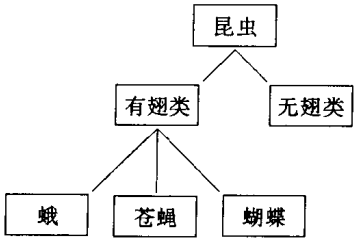


图 1.1 昆虫类

在对象类的关系中，继承类似于昆虫分类之间的关系，只是它是针对类而言的，类可以从另一个类中继承特征，从而实现已有功能模块的可重用性。

一个类从另一个类继承特征，称为派生一个类，所派生的类称派生类或子类(subclass)。其上一层的那个类称为基类或父类(superclass)。类的派生过程可以无限继续下去。派生类在继承基类的特征和功能的同时，加入其自身具有的特征和功能，从而实现功能模块的可扩充性。

1.2.3 多态性(polymorphism)

多态性就是同种操作具有多种形态，在 C++中，多态性又被直观地称为“一个名字，多个函数”。例如，常用的加法，整数相加得到整数，复数相加得到复数，矩阵相加的结果是矩阵。同样是加法操作，针对不同的对象可有不同的操作形态而产生不同的执行结果。这一种多态性在 C++编程中称为操作重载(overloading)。从同一父类派生而来的子类，具有名称相同的，但不完全相同的某种操作，则是另一种多态性。例如，昆虫类的飞行，苍蝇和蝴蝶是不一样的；几何类图形求面积，三角形和圆形要采用各自的计算公式。

在 C++中，多态性的实现与联编这一概念相关。将一个操作函数调用链接上相应函数体的代码这一过程被称为函数联编(简称联编)。C++中，有两种联编形式：静态联编和动态联编。静态联编对应于上述的前一种多态，动态联编则对应于上述的后一种多态。静态联编在程序被编译时进行，而动态联编直到程序运行时才

试读结束：需要全本请在线购买： www.ertongbook.com

能确定调用哪个函数。

静态联编所支持的多态性称为编译时多态，对调用重载函数的代码进行编译时，编译器根据调用重载函数所使用的实参类型，在编译时就确定下来应该调用哪个函数。

动态联编所支持的多态性称为运行时多态性，这需要通过继承和虚函数来支持，程序运行中将根据不同的子类来确定应该调用哪个操作函数。

1.3 面向对象的结构分析程序设计

近 30 年来结构分析技术有了突破性的发展，具体的标志是有限元法的出现与成熟。由于它的发展使长期困扰固体力学、结构力学的大量问题得以解决。目前的结构计算分析问题大都是围绕着有限元所展开。有限元方法在按层次分类和聚合方面与面向对象方法有许多相似之处，由于具有了面向对象方法的特点，有限元方法同样具有可重用性和可扩展性，通过运用相同的几类单元，可组合成多种不同类型和不同规模的结构体。面向对象结构分析程序较有限元方法有着更高层次的抽象和分类，因此它的可重用性和可扩展性内容更为广泛和深入，为扩展计算分析功能和增加单元模型带来了方便。

面向对象方法可以分为三个部分：面向对象分析、面向对象设计和面向对象程序实现。

1.3.1 面向对象分析

面向对象分析的目的是完成对问题空间的分析和建立系统模型。其具体任务是确定和描述系统中的对象、对象的静态特性和动态特性、对象间的关系及对象的行为约束等。其主要内容归结为静态结构分析和动态行为分析。

1. 静态结构分析

静态结构描述对象、类以及类间的相互关系。静态结构分析首先要确定对象和对象类。对象是一种具有简明界面及应用意义的实体的抽象。所有对象都具有惟一的标识，可予以区分。对象类描述的是一组具有共同属性、操作和语义特征的对象，其中属性是对象的性质，操作是对象的行为。

根据静态结构分析方法，我们可以初步将一个结构分成图 1.2 所示的构成形式。整个结构由一整体结构类进行描述，整体结构类包含构件单元对象、荷载对象、结点对象、材料对象等属性数据以及多项操作。

在确定了对象和对象类的基础上，还要确定对象类之间的关系。对象类之间的关系主要有：

(1) 层次关系

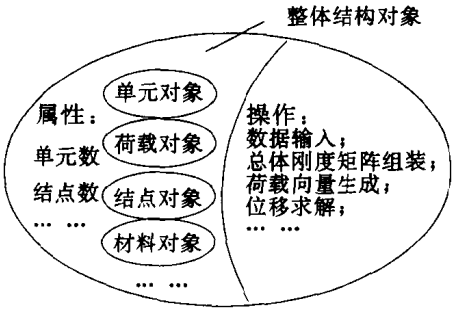


图1.2　整体结构对象组成

在分析过程中根据类的共性及个性将类组织成不同层次。高层次的类表达共性形成父类；低层次的类表达个性，形成子类。子类通过继承机制来获得父类的属性和操作。图 1.3 表示了构件单元类的层次关系，最高层次的基本单元包括了各单元的共性，如结点数和结点编号等，在第二层次上，根据单元几何形状分成杆单元、平面单元、板壳单元和块体单元等。杆单元类中根据构件不同的受力特点分成梁单元和连杆单元。再根据不同的构成材料或受力特性分成混凝土梁和钢梁等。层次关系分类的标准不是惟一的，也可将按材料构成来划分层次。还有一些单元，需根据应用情况继承不同的父类。如钢筋和预应力钢筋单元，在嵌入平面单元中时，按平面单元描述，需从平面单元派生而来；嵌入杆单元中时，按杆单元描述，需从杆单元派生而来。

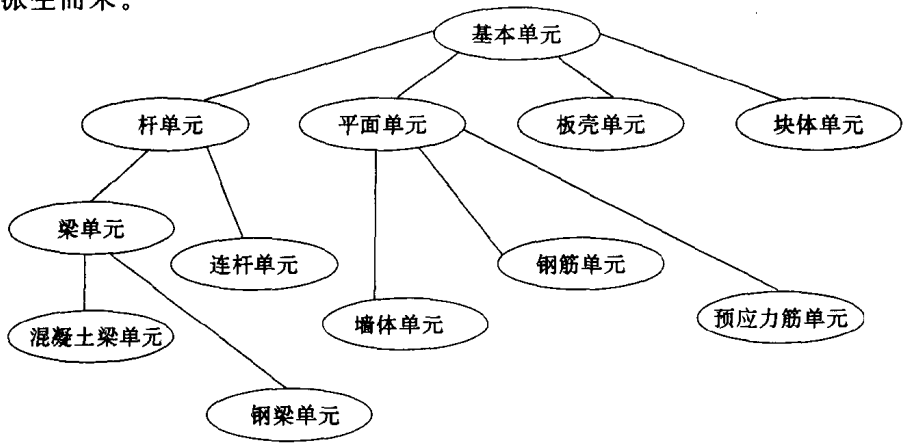


图1.3　单元类的层次关系

(2) 聚合关系

聚合关系是对象之间的组合构造关系。对象间的组合根据一定层次关系来进行，高层次的对象是容器对象，称为聚合对象；低层次的对象是内含对象，称为组成对象。聚合对象通过组成对象的操作来实现自身的操作。如图 1.2 所示的整体结构对象即为高层次的容器对象，其组成对象为各种构件单元对象和荷载对象等。在第四章对单元类的描述中将可以看到，作为组成对象的构件单元对象也包括了诸如变形矩阵、刚度矩阵等组成对象。因此，构件单元对象又是次一层次的容器对象。

2. 动态行为分析

动态行为描述了系统中对象的合法状态序列。对象的动态行为通常用动态模型来表示，它包括两个方面的内容：一是单个对象自身的生命周期演化；二是整个对象系统中对象间的消息传递和协同工作。

对象生命周期演化主要包括三个组成部分：一是对象在生命期中可能的状态；二是对象发生状态转换时要执行的动作，动作的效果不仅依赖于对象的操作而且依赖于对象所处的状态；三是导致对象从一个状态到另一个状态的转换的事件，事件是控制状态转换的条件。这三个部分一起构成了对象生命期中的状态转换图描述对象内部的动态行为。

对象的动态行为可以通过继承关系由其子对象继承。子对象状态图是父对象状态图的细化。

在对象系统中，对象之间通过消息传递来协同工作。对应于系统的每一项任务，都有一组对象上的一组消息传递及动作来完成该任务。每个对象系统都要完成一组任务，而每个任务都有一组事件序列与之对应。因此，系统中对象协作的动态性质可以由一组事件序列来描述。结构分析中各对象动态行为的具体分析将在第四章中予以叙述。

1.3.2 面向对象设计

面向对象的设计是确定问题的解决方案的过程。面向对象设计的涉及面有大有小，结构分析软件的设计涉及用户界面、系统数据结构、系统资源分配、数据交换存储、图形显示等许多方面。综合起来，面向对象设计包括对象设计和系统设计两方面的内容。

1. 对象设计

对象设计以对象分析为基础，分析过程已给出对象模型。在设计对象时可根据需要进行扩展，增加和优化属性数据及功能操作，对分析中未考虑周全的类关系进行调整，尽可能利用继承的优点，提高代码的复用程度。

2. 系统设计

系统设计是为实现需求目标而对程序的系统结构进行的总体设计。通过程序系统地调用各对象功能操作，达到解答问题的目标。通常系统结构本身也包含在对象内部。系统结构实现策略取决于具体的应用目标。面向对象结构分析系统设计主要包括以下内容：

(1) 数据的存取和处理设计

结构分析中会涉及大量的数据。首先是结构模型的描述数据，较简便的方法是

通过数据文件直接读入。随着软件技术的发展，通过用户界面人机交互进行输入的技术也已经相当成熟。结构计算分析过程中和完成后，有大量的数据产生，可按一定的数据格式进行储存或连接到数据库上。在计算分析的同时或后期通过图形和数据显示分析结果。

(2) 用户接口及相应过程设计

面向对象方法采用消息传递的方式进行，由用户在用户界面上发出一条消息，程序按消息调用某个对象的操作，完成用户所需的功能操作。相应过程设计就是建立这一消息响应操作的程序流程。

(3) 系统组织设计

系统组织是指程序中对变量和对象的管理。前面的对象动态行为分析中提到，对象存在生命周期，一个对象从创立开始，在程序运行中不断变化属性，发挥一定的功能，完成其使命后消亡。一些对象始于程序首，终于程序末，一些对象只在一段程序中发挥作用。在以往计算机硬件资源相对较小的时候，为节约资源，采用频繁的动态资源分配，将暂时不用的对象删除，然后再建。虽然现在的计算机硬件有了很大发展，但严谨的系统组织管理对设计的效率、以后的维护和功能扩充是非常有益的。由于各种结构的构成不同，构件单元对象需要动态创建。某些对象虽然是局部的，由于需频繁调用，则可采用静态创建，以提高程序运行效率。

1.4 面向对象程序编程和开发工具

1.4.1 面向对象程序编程语言和环境

面向对象的程序设计语言有许多，主要有 Smalltalk、Effel、C++和 Java 等。其中的C++因其诸多优点、强而灵活的功能而被广泛采用。许多程序的开发和已有程序的二次开发都采用了C++。比如，Windows 程序的编写，AutoCAD 二次开发工具 ObjectARX 等。还有许多成熟的和商品化的函数库和类库可供选择。

C++由 C 改进扩充而来，C++包含 C 的全部特征，添加了对面向对象编程的全面支持。主要包括：可进行类定义；为类提供了数据操作的函数及操作符；为类提供了构造方法及析构方法；对类内部的数据结构进行了封装性的划分；提供了对继承关系及多重继承的支持；提供了对多态性及函数重载的支持。

面向对象的结构分析程序由程序语言实现后的完成品是封装起来的各个对象，可采用以往的主程序调用子程序的方式，通过调用对象的功能操作完成计算分析过程。也可将封装的对象嵌入其他程序，如编入具有窗口环境的 Windows 框架程序中，或嵌入具有较强图形功能的软件中，如 AutoCAD 等。在 Windows 的窗口环境下，用户通过窗口界面向结构分析程序发出消息来完成用户所需的操作，操作内容应包括前后处理、各项结构分析计算等。窗口界面在分析过程中能随时将各种

信息反馈给用户对象。这种人机交互的分析处理过程更符合面向对象的方法模式。

在采用面向对象方法编程以前，Windows 编程是一件非常繁琐的事情，编程人员要关心窗口建立、消息传递、各种函数调用等各个细节，使人望而生畏。采用面向对象编程技术后，Windows 程序的开发从形式到内容均发生了很大变化，Borland 公司和 Microsoft 公司相继推出了用于 Windows 编程的类库，这些类对 Windows API 函数进行了封装，从而大大简化了 Windows 的编程工作。使得不是很专业的人员也能编写 Windows 程序。运用开发工具中的向导模板，较标准的 Windows 界面程序可在一二分钟内完成。

1.4.2 开发工具 Visual C ++

C++程序开发工具有许多种，从 20 世纪 90 年代初的集成开发环境，如 Turbo C++、Borland C++，发展到现在的可视化开发环境，如 Visual C++、C++ Builder 等。可视化开发环境能清晰地了解系统中各部分的关系，消息的传递方式、类的继承关系、类的封装内容等得以直观地表示，帮助编程人员快速浏览类的属性变量和操作函数，可视化的调试过程能动态显示变量的数值变化、追踪各对象的访问过程等。

Visual C++开发环境 Microsoft Visual Studio 由 Windows 下的一套集成工具所组成，包括输入程序源代码的文本编辑器、用户窗口界面设计的资源编辑器、跟踪程序源文件和建立项目配置的项目管理器、建立并运行程序的优化编译器以及能进行可视化调试的集成调试器等。

Visual C++既可用于 Windows 编程，也可编写类似以往的 Dos 状态下的程序。功能强大的向导工具 AppWizard 和 ClassWizard 使 Windows 编程大大简化。AppWizard 用于生成基于 MFC 类库的各种较常规的 Windows 窗口的源文件和资源文件。在创建应用程序的基本框架后，使用类向导 ClassWizard 来定义类的消息及其处理过程、重载虚函数、派生或创建新的类、定义新的类属性数据等，最终建立起完整的应用程序。

第二章　C＋＋相关技术运用与矩阵类

C＋＋以其高效实用性被应用于程序设计的众多领域，功能强大的开发环境和许多技术成熟的应用类库使C＋＋编程效率大大提高。本章首先讲述面向对象编程中一些主要的C＋＋相关技术运用方法，其中还包括了数据处理功能较强的MFC (Microsoft Foundation Class Library)的数组类的应用。

矩阵运算在科研计算中被广泛使用，采用矩阵对象进行矩阵运算可使程序变得简洁明了。目前，已经有不少现成的矩阵类库可供选择。本章的2.2节将讲述矩阵类的实现方法。

2.1　C＋＋相关技术运用

2.1.1　匈牙利表示法

匈牙利表示法是指一种应用程序变量的标志符记法，它是由Microsoft公司的Charles Simonyi首先采用的。因他是匈牙利人，所以该记法被称为匈牙利表示法。

匈牙利表示法中对变量进行描述的变量标志符是以大写字母开头的英文词组成，如Student、ClassName等。需两个以上英文词对变量进行描述时各单词均以大写字母开头，以便阅读。采用这种变量描述法，在编程开始时感到麻烦些，但它便于回顾，对程序的调试和维护是很有益的。在变量标志符的前面加上前缀表明变量的类型，比如：i表示整型、d表示双精度浮点数、a表示数组、p表示指针、C表示类等。当变量为对象的成员变量时，再在前缀前加上前导m_，以便在成员函数中区分对象的成员变量和成员函数的局部变量。

根据上述匈牙利表示法，下列中的第一变量名表示该变量是某对象的成员变量，变量的类型是双精度数组，用来记录某些位移。第二个变量名表示指向名为Element对象的指针数组。

```
m_adDisplacement
m_apElement
```

表2.1中列出了本书中用到的变量前导和前缀。大家也可根据自己的要求或习惯，添加或变更这些前导和前缀。

试读结束：需要全本请在线购买：　www.ertongbook.com