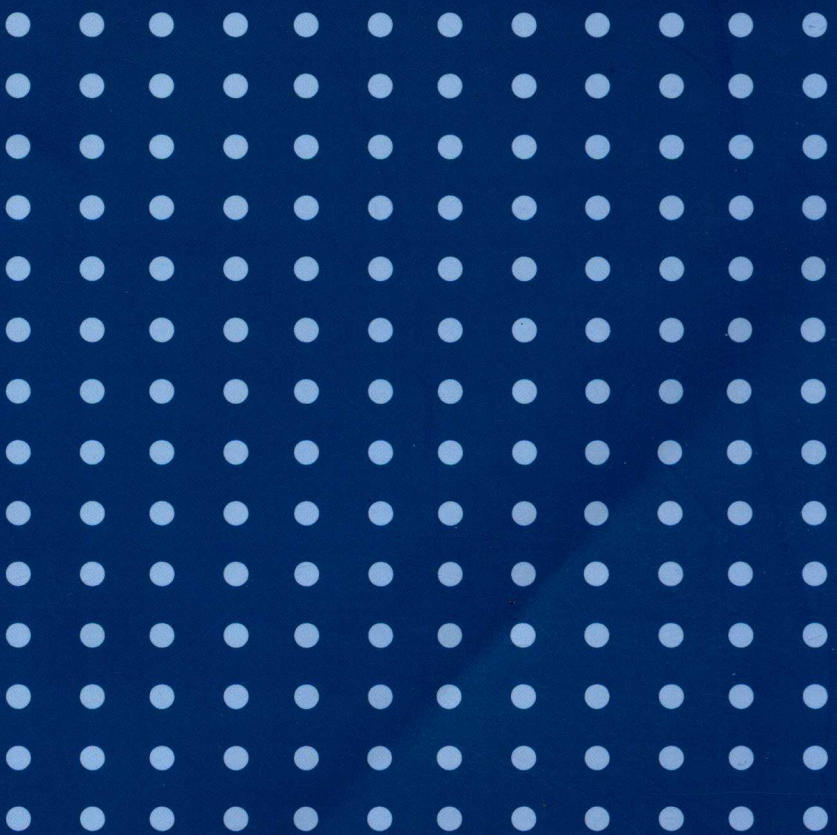
重点大学计算机专业系列教材

面向对象程序设计与VC++实践

揣锦华 袁琪 编著

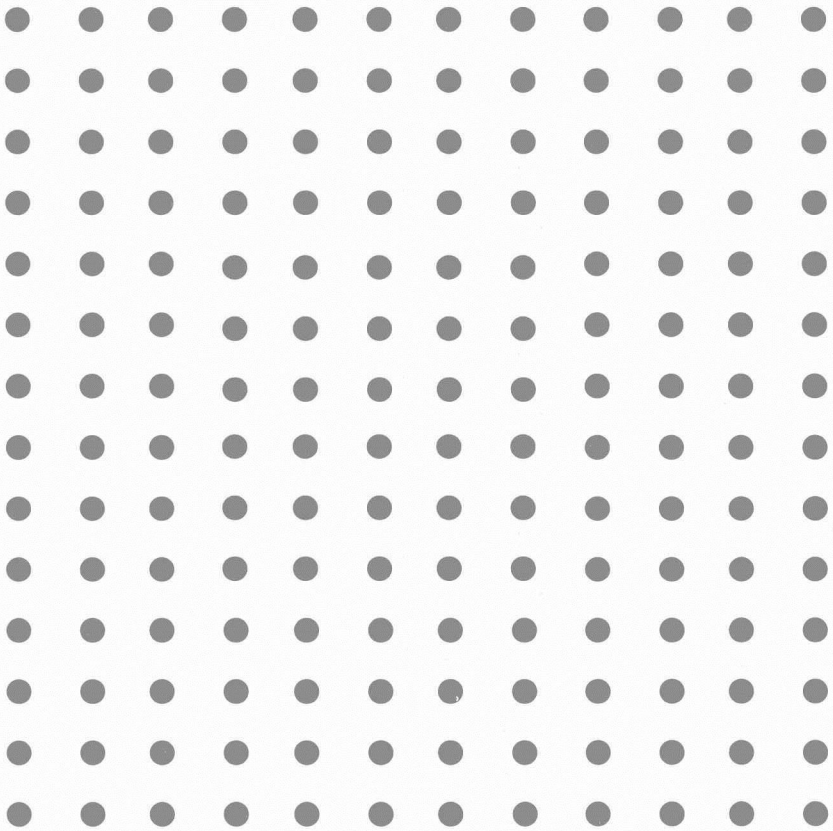


清华大学出版社

重点大学计算机专业系列教材

面向对象程序设计与VC++实践

揣锦华 袁琪 编著



清华大学出版社
北京

内 容 简 介

本书以面向对象程序设计思想贯穿始终，结合应用实例，强调实用，以通俗易懂的语言将面向对象程序设计的方法和 VC++的具体应用展示给读者。全书共分为三部分：第一部分介绍 C++面向对象程序设计的基础知识；第二部分介绍 Windows 编程基础以及 Windows 程序的基本特征；第三部分介绍使用 MFC 进行 Windows 应用程序设计的方法，包括 MFC 的一些高级应用。同时，各章都配备有相应的应用程序实例和大量习题，既方便教师安排教学，又便于读者综合运用所学知识。

本书既可作为高等院校计算机专业或非计算机专业的面向对象程序设计及 VC++的教学用书，也可作为程序设计人员的自学参考用书。

图书在版编目(CIP)数据

面向对象程序设计与 VC++实践/揣锦华，袁琪编著. —北京：清华大学出版社，2016
重点大学计算机专业系列教材
ISBN 978-7-302-42645-5

Ⅰ. ①面… Ⅱ. ①揣… ②袁… Ⅲ. ①C 语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 013816 号

责任编辑：郑寅堃　李　晔
封面设计：常雪影
责任校对：梁　毅
责任印制：刘海龙

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，010-62795954

印 装 者：三河市中晟雅豪印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：19.25　　**字　　数**：483 千字
版　　次：2016 年 2 月第 1 版　　**印　　次**：2016 年 2 月第1 次印刷
印　　数：1～2000
定　　价：39.50 元

产品编号：059375-01

INTRODUCTION

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大，社会对计算机专业人才的需求不仅体现在数量的增加上，而且体现在质量要求的提高上，培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前，我国共有16个国家重点学科、20个博士点一级学科、28个博士点二级学科集中在教育部部属重点大学，这些高校在计算机教学和科研方面具有一定优势，并且大多以国际著名大学计算机教育为参照系，具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系，形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础，其中专业教材的使用和建设则是这种氛围的重要组成部分，一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势，特别是专业教材建设上的优势，同时配合各重点大学的计算机学科建设和专业课程教学需要，在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下，清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”，同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿，适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础，反映基本理论和原理的综合应用，重视实践和应用环节。

2. 反映教学需要，促进教学发展。教材要能适应多样化的教学需要，正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略，突出重点，保证质量。规划教材建设的重点依然是专业基础课和专业主干课；特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本，合理配套。专业基础课和专业主干课教材要配套，同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系；基本教材与辅助教材以及教学参考书的关系；文字教材与软件教材的关系，实现教材系列资源配套。

5. 依靠专家，择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主编。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

FOREWORD

前言

面向对象的程序设计技术是程序设计的一种新方法，它汲取了结构化程序设计中最为精华的部分，它是软件开发的第二次变革。计算机可视化技术以其新颖的图形用户界面、卓越的多任务操作系统性能、高层次可视化的软件开发平台迅速风靡全球。

Visual C++集成开发环境(IDE)，提供了涵盖项目管理、资源配置、代码编辑、编译、调试、维护和图形用户界面设计的开发环境。本书以 Visual Studio 开发工具包为例介绍了基于 IDE 的项目开发过程。该环境可引导用户生成 Visual C++项目应用程序框架，包括简单的 Win32 控制台程序、Windows 环境下的 Win32 项目和基于 MFC 的具有图形用户界面的应用程序，还可以开发类和动态链接库。用户只需要在框架下按需求增加功能代码，大大提高了开发效率。除此之外，微软基础类库 MFC 的利用使应用程序框架、类框架及函数框架的自动生成更加快捷方便，大大提高了开发效率和程序质量。

本书以面向对象程序设计思想贯穿始终，以应用实例诠释设计和实现方法，通俗易懂，强调实用性。全书共分为三部分。第一部分以 VC++语言为基础，介绍面向对象程序设计的基本思想。第二部分介绍 Windows 编程的基础知识以及 Windows 程序的基本特征，并介绍应用程序编程接口(API)及其在 Windows 绘图程序设计中的应用。第三部分介绍 MFC 基础知识以及基于 MFC 的 Windows 应用程序开发，具体包括 MFC 应用程序框架和启动流程；MFC 应用程序消息映射机制；基于常用 MFC 控件如编辑框、按钮、菜单、滚动条、列表框的用户界面开发；以及文档 / 视图结构和加速键、工具条、状态条等用户界面工具的使用；最后，介绍了 MFC 的一些高级应用，如动态链接库、多线程等。为了使读者能更深入地了解 VC++，在每章中都给出了应用实例，以便读者参考，同时书中各章均配有思考题和习题，方便读者练习和自我考核。

本书注重培养读者分析问题和解决问题的能力，强化动手实践，内容由浅入深，循序渐进，适合作为高等学校计算机程序设计相关课程的教材和教学参考书，也可作为学习程序设计人员的培训或自学教材。

本书的第 1、2、3、5、6 章由长安大学的揣锦华编写，第 4、11 章由长安大学的袁琪编写，第 7～10 章由袁琪和揣锦华共同编写。在本书的编写过程中，查阅和参考了大量文献，在此对书后所列出的“参考文献”的作者表示感谢。另外，对广大读者和师生对本书诚恳的建议和意见也表示衷心的感谢。由于作者水平有限，书中难免有不足和欠妥之处，恳请读者批评指正。

编　者

2015 年 9 月

CONTENTS

目录

CHAPTER 1

第1章 面向对象程序设计

1.1 面向对象程序设计思想

面向对象程序设计(Object Oriented Programming,OOP)是一种计算机编程架构。面向对象程序设计的基本原则是计算机程序由单个能够起到子程序作用的单元或对象组合而成。面向对象程序设计达到了软件工程的三个主要目标：重用性、灵活性和扩展性。为了实现整体运算，每个对象都能够接收信息、处理数据和向其他对象发送信息。

1.1.1 结构化程序设计的不足

结构化程序设计的基本思想是自顶向下，逐步求精，即将复杂的大问题层层分解为许多简单的小问题的组合。在具体程序设计时，整个程序被划分成多个功能模块。

结构化程序设计的特点是“程序＝数据结构＋算法”。其中，用于保存数据的数据结构与各种类型的变量相对应，完成具体功能的算法与函数相对应。在结构化程序设计中，算法和数据结构是分离的，没有直观的手段能够说明一个算法操作了哪些数据结构，一个数据结构又由哪些算法来操作。当数据结构的设计发生变化时，分散在程序各处的所有操作该数据结构的算法都需要修改。结构化程序设计也没有提供手段来限制数据结构可被操作的范围，任何算法都可以操作任何数据结构，很容易造成算法由于编写失误对关键数据结构进行错误的操作而导致程序出现严重错误(bug)，甚至崩溃。

结构化程序设计也称为面向过程的程序设计，过程是通过函数来实现的。因此，结构化程序设计归根结底要解决的是如何将整个程序分解为一个个的函数，并且要决定哪些函数之间要互相调用，以及每个函数内部将如何实现。当软件的规模变大时，程序中大量的函数、变量之间的关系也会错综复杂。

结构化程序设计的不足主要表现在：

(1) 结构化程序难以理解和维护。

(2) 结构化的程序不利于修改和扩充。

(3) 结构化的程序不利于代码重用。

总之，随着软件规模的不断扩大，结构化程序设计难以适应软件开发的需要，此时，面向对象程序设计就应运而生了。

1.1.2 从结构化程序设计到面向对象程序设计

在结构化程序设计中，确定所采用的数据结构实际就是数据抽象，而设计实现算法的函数实际就是过程抽象。下面举例从数据抽象和过程抽象的角度来说明结构化程序设计和面向对象程序设计的不同。

例如，学生数据包括学号、姓名以及计算机、英语、高数三门课的成绩，编写程序实现以下功能：

(1) 计算每个学生的总成绩。

(2) 输出学生信息。

1. 数据抽象与过程抽象分离的结构化程序设计

```
//数据抽象
struct student
{
    int no;
    char *name;
    float computer;
    float english;
    float math;
};
//过程抽象
float sum(student stu)
{
    return stu.computer + stu.english + stu.math;
}
void disp(student stu)
{
    cout < stu.no << endl;
    cout < stu.name << endl;
    cout < stu.computer << endl;
    cout < stu.english << endl;
    cout < stu.math << endl;
}

int main()
{
    student stu1 = {1,"Wutao",19,87,90};
    cout << sum(stu1)<< endl;
    disp(stu1);
    return 0;
}
```

2. 在数据抽象内部组织过程抽象的面向对象程序设计

```
//在数据抽象内部组织过程抽象
class CStudent
{
    int no;
    char * name;
    float computer;
    float english;
    float math;
public:
    CStudent(){…}
    float sum()
    {
        return computer + english + math;
    }
    void disp()
    {
        cout < no << endl;
        cout < name << endl;
        cout < computer << endl;
        cout < english << endl;
        cout < math << endl;
    }
};

int main()
{
    CStudent stu2{1,"Wutao",19,87,90};
    cout << stu2.sum()<< endl;
    stu2.disp();
    return 0;
}
```

1.1.3　面向对象的概念和方法

要了解面向对象的概念，首先要知道什么是对象。对象在现实世界中是一个实体或者一种事物的概念。现实世界中的任何一个系统都是由若干具体的对象构成，作为系统的一个组成部分，对象为其所在的系统提供一定的功能，担当一定的角色。所以可以将对象看作是一种具有自身属性和功能的构件。

我们在使用一个对象时，并不关心其内部结构及实现方法，仅仅关心它的功能和它的使用方法，也就是该对象提供给用户的接口。例如，对电视机这个对象来说，我们并不关心电视机的内部结构或其实现原理，只关心如何通过按钮来使用它，这些按钮就是电视机提供给用户的接口，至于电视机内部结构原理，对用户来说是隐藏的。分析一个系统，也就是分析系统由哪些对象构成，以及这些对象间的相互关系。

在面向对象方法中，我们采用与现实世界相一致的方式，将对象定义为一组数据及其相关操作的结合体，其中数据描述了对象的属性，对数据进行处理的操作则描述了对象的功

能。所以可以将对象看作是一种具有自身属性和功能的构件，对象将其属性和操作的一部分对外界开放，作为它的对外接口，而将大部分的实现细节隐藏，这就是对象的封装性，外界只能使用上述定义的接口与对象交互。

面向对象方法中引入了类的概念。所谓类，就是同样类型对象的抽象描述，对象是类的实例。类是面向对象方法的核心，对相关的类进行分析，抽取这些类的共同特性，形成基类，通过继承，派生类可以包含基类的所有属性和操作，还可以增加属于自己的一些特性。同时，通过继承，可以将原来一个个孤立的类联系起来，形成清晰的层次结构关系，称为类族。

一个系统由多个对象组成，其中复杂对象可以由简单对象组合而成，称之为聚合。对象之间存在着依存关系，一个对象可以向另一个对象发送消息，也可以从其他对象接收消息，对象之间通过消息彼此联系，彼此协作，对象以及对象之间的这种相互作用构成了软件系统的结构。

综上所述，面向对象的方法就是利用抽象、封装等机制，借助于对象、类、继承、多态、消息传递等概念进行软件系统构造的软件开发方法。

1.1.4 面向对象程序设计的特点

1. 抽象性

抽象，一般是指从具体的实例中抽取出共同的性质并加以描述的过程。在面向对象方法中，抽象是通过对一个系统进行分析和认识，强调系统中某些本质的特性，而对系统进行的简化描述。

一般，对问题的抽象包括两个方面：数据抽象和行为抽象。数据抽象为程序员提供了对对象的属性和状态的描述，行为抽象则是对这些数据所需要的操作的抽象。

抽象的过程是通过模块化来实现的，即通过分析将一个复杂的系统分解为若干个模块，每个模块是对整个系统结构的某一部分的一个自包含的和完整的描述。同时，对模块中的细节部分进行信息隐藏，用户只能通过一个受保护的接口来访问模块中的数据，这个接口由一些操作组成，定义了该模块的行为。

下面举例说明。假设我们需要在计算机上实现一个绘制圆形的程序。通过对这个图形的分析，可以看出需要三个数据来描述该圆的位置和大小，即圆心的横、纵坐标以及圆的半径，这就是对该圆形的数据抽象。另外，该图形应该具有设置圆心坐标、设置半径大小、绘制圆形等功能，这就是对它的行为抽象。下面给出该图形的C++语言描述。

圆形(CCircle)的数据抽象：

```
double  x,y,r;
```

圆形(CCircle)的行为抽象：

```
get_x( );
get_y( );
get_area( );
draw( );
```

抽象是面向对象方法的核心。

2. 封装性

封装是面向对象方法重要的原则。所谓封装，就是将一个事物包装起来，使外界不了解