教育部产学合作协同育人项目

高等学校计算机课程规划教材

面向对象程序设计

杨巨成　于　洋　主编
孙　迪　于秀丽　侯　琳　李建荣　副主编

清华大学出版社

教育部产学合作协同育人项目

高等学校计算机课程规划教材

面向对象程序设计

杨巨成　于　洋　主编
孙　迪　于秀丽　侯小琳　李建荣　副主编

清华大学出版社
北京

内 容 简 介

本书共10章，分别介绍面向对象、类、Visual Studio 2015环境，以及面向对象程序的结构、函数、数组、指针、继承、派生、多态性、流类库、输入输出和异常处理。

C++面向对象程序设计涵盖面向过程的C语言，是学习C类语言的基础，学好C++面向对象程序设计可为今后用C#开发智能软硬件系统以及用Objective-C在iOS和MAC OS系统中进行移动端程序设计打好坚实的基础。

本书是高级语言程序设计的入门教程，完全适合零起点的学生，可作为计算机类相关课程的基础课教材，也可作为面向对象程序设计爱好者的参考书。为方便读者学习，本书提供作者自主开发的电子课件和视频。

图书在版编目(CIP)数据

面向对象程序设计/杨巨成，于洋主编. —北京：清华大学出版社，2018
(高等学校计算机课程规划教材)
ISBN 978-7-302-48931-3

Ⅰ. ①面… Ⅱ. ①杨… ②于… Ⅲ. ①面向对象语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2017)第287287号

责任编辑：汪汉友
封面设计：傅瑞学
责任校对：李建庄
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：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：16.25　　**字　　数**：395千字
版　　次：2018年2月第1版　　**印　　次**：2018年2月第1次印刷
印　　数：1～1500
定　　价：39.00元

产品编号：077890-01

出版说明

信息时代早已显现其诱人魅力，当前几乎每个人随身都携有多个媒体、信息和通信设备，享受其带来的快乐和便宜。

我国高等教育早已进入大众化教育时代，而且计算机技术发展很快，知识更新速度也在快速增长，社会对计算机专业学生的专业能力要求也在不断翻新，这就使得我国目前的计算机教育面临严峻挑战。我们必须更新教育观念——弱化知识培养目的，强化对学生兴趣的培养，加强培养学生理论学习、快速学习的能力，强调培养学生的实践能力、动手能力、研究能力和创新能力。

教育观念的更新，必然伴随教材的更新。一流的计算机人才需要一流的名师指导，而一流的名师需要精品教材的辅助，而精品教材也将有助于催生更多一流名师。名师们在长期的一线教学改革实践中，总结出了一整套面向学生的独特的教法、经验、教学内容等。本套丛书的目的就是推广他们的经验，并促使广大教育工作者更新教育观念。

在教育部相关教学指导委员会专家的帮助和指导下，在各大学计算机院系领导的协助下，清华大学出版社规划并出版了本系列教材，以满足计算机课程群建设和课程教学的需要，并将各重点大学的优势专业学科的教育优势充分发挥出来。

本系列教材行文注重趣味性，立足课程改革和教材创新，广纳全国高校计算机优秀一线专业名师参与，从中精选出佳作予以出版。

本系列教材具有以下特点。

1. 有的放矢

针对计算机专业学生并站在计算机类课程建设、技术市场需求、创新人才培养的高度，规划相关课程群内各门课程的教学关系，以达到教学内容互相衔接、补充、相互贯穿和相互促进的目的。各门课程功能定位明确，并去掉课程中相互重复的部分，使学生既能够掌握这些课程的实质部分，又能节约一些课时，为开设社会需求的新技术课程准备条件。

2. 内容趣味性强

按照教学需求组织教学材料，注重教学内容的趣味性，在培养学习观念、学习兴趣的同时，注重创新教育，加强“创新思维”“创新能力”的培养和训练；强调实践，案例选题注重实际和兴趣度，大部分课程各模块的内容分为基本、加深和拓宽内容 3 个层次。

3. 名师精品多

广罗名师参与，对于名师精品，予以重点扶持，教辅、教参、教案、PPT、实验大纲和实验指导等配套齐全，资源丰富。同一门课程，不同名师分出多个版本，方便选用。

4. 一线教师亲力

专家咨询指导，一线教师亲力；内容组织以教学需求为线索；注重理论知识学习，注重学

习能力培养，强调案例分析，注重工程技术能力锻炼。

经济要发展，国力要增强，教育必须先行。教育要靠教师和教材，因此建立一支高水平的教材编写队伍是社会发展的关键，特希望有志于教材建设的教师能够加入到本团队。通过本系列教材的辐射，培养一批热心为读者奉献的编写教师团队。

清华大学出版社

前　言

C++ 在 2017 年的编程语言排行榜中，仍然排在第三的位置，是世界上最稳定，使用范围最广的计算机编程语言。这是 C++ 相对于其他语言的优势所在。

针对普通高等学校计算机类专业“面向对象程序设计”课程教学的实际情况，以培养应用型人才为主要目标，结合多年的一线教学经验积累，我们特别编写了本教材。本教材可供 64 学时以上的“面向对象程序设计(C++)”专业基础课程和“面向对象课程设计”等实验实践类课程使用，是自主学习和课堂教学的必备的资料。

本套教材的内容具有以下特点：

(1) 从最基础的底层知识开始讲解，适合初次学习面向对象编程语言的高等学校学生学习。

(2) 本书内容翔实，基本覆盖了 C++ 程序语言的各个知识点，为初学者以后的深入学习奠定了坚实的基础。

(3) 本书从基本概念、基本理论、基本应用出发，结合实际，对每一个程序都进行了调试、编译和运行，保证了程序的正确性、可读性和良好的可移植性。

(4) 本书语言通俗易懂，使枯燥的理论学习更加易于接受。

(5) C++ 面向对象程序设计涵盖面向过程的 C 语言，以 C 语言为基础，为今后使用 C# 开发智能软硬件系统以及使用 Objective-C 在 iOS 和 MAC OS 系统中进行移动端程序设计打好坚实的基础。

为了便于读者学习，本书的编排形式如下。

(1) 知识点。在每节的开始位置都会有准确、清晰的知识点介绍，让读者在第一时间了解相关概念，便于后面的学习。

(2) 示例。在每节知识点之后都会列举完整的示例，并以章节顺序编号，便于检索和案例的衔接。示例代码与示例编号一一对应，层次清楚、语言简洁、注释丰富，体现了代码优美的原则，有利于读者养成良好的代码编写习惯。对于大段程序，均在每行代码前设定编号便于学习，并加以详尽的规范的注释。每段程序和例题均给出运行结果，帮助读者更直观地理解实例。

(3) 习题。每章最后都会提供专门的测试习题，供读者检验所学知识是否牢固掌握。

随着高校计算机基础教育的发展，本教材也将不断更新、调整，希望各位专家、教师和读者不断提出宝贵的意见和建议，我们将根据大家的意见不断改进内容，更新编程环境的版本。

作为“面向对象程序设计”的国家级双语示范精品课程主讲教师，本书的主编杨巨成教授在编写的同时还给予其他年轻教师大力帮助，提供了大量的有价值的第一手材料，他对教材严格把关，给出了很多中肯的意见和建议。参与本教材编写的教师有于洋、孙迪、于秀丽、侯琳、李建荣、梁琨、蔡润身和张翼英老师。在教材编写过程中，参与程序测试的学生团队成员包括陈晗頔、任哲、程宇辉、王凯、周柯奇、谭进、季晓曈、康钰莹和陈序。这些具有丰富竞

赛经验的在校本科生从读者的角度，对教材的易用性等方面进行了科学、客观的测试，本书每一段程序都由他们调试通过、整理，为教材的编写出版做出了巨大的贡献。本教材以实用为出发点，力求为我国高校计算机基础教育的教材建设和人才培养做出贡献。

为了便于学习，本书可以配有自主开发的电子课件、视频教程，以及基于 Visual Studio 2015 开发的综合实例源代码。读者可从清华大学出版社网站本书相应页面下载。

编　者

2017 年 10 月于天津科技大学

目　录

第1章　面向对象和类

【本章内容】

- 面向对象和类的基本知识；
- 面向对象程序设计；
- 对象和类；
- 构造函数和析构函数；
- 静态成员；
- 友元；
- 类模板；
- 习题。

1.1　面向对象程序设计

1.1.1　什么是面向对象程序设计

1. 面向对象的思想

面向对象是相对于面向过程而言的一种编程思想，它是通过操作对象实现具体的功能，即将功能封装进对象，用对象实现具体的细节。这种思想以数据为中心、方法（算法）居其次，是对数据的一种优化，实现起来非常方便、简单。

C++是一种面向对象的程序设计语言，使用它可以进行面向对象程序设计，在介绍面向对象程序设计的特性之前，必须先要了解它的方法和特点。

2. 面向对象程序设计

面向对象程序设计（Object Oriented Programming，OOP）既是一种程序设计范型，又是一种程序开发方法。这里的对象指的是类的实例，即客观世界中存在的对象。它们既互不相同、拥有各自的特点，具有唯一性，又彼此相互联系、相互作用。面向对象程序设计是将对象作为构成软件系统的基本单元，并从相同类型的对象中抽象出一种新型的数据结构——类。

类是一种特殊的类型，其成员中不仅包含描述类对象属性的数据，还包含对这些数据进行处理的程序代码，这些程序代码称为对象的行为（或操作）。将对象的属性和行为封装在一起后，可使内部的大部分实现细节被隐藏，仅通过一个可控的接口与外界交互。

面向对象程序设计是完成程序设计任务的一种新方法，它汲取了结构化程序设计思想中最为精华的部分。面向对象程序设计是一种结构化程序设计，是软件开发的第二次变革，是程序结构的统一理论。

3. 面向对象程序设计的基本特点

（1）抽象性。抽象是指从具体的实例中抽取共同的性质并加以描述，忽略次要的和

非本质的特征。面向对象程序设计比面向过程程序设计更加强调抽象性。在面向对象方法中，抽象是从系统层面进行分析和认识的，强调的是实体的本质和内在属性，是从一般的观点来观察事物，集中研究某一性质，忽略其他与此无关的部分，对系统进行简化描述。

对于问题的抽象一般包括两个方面：数据抽象和行为抽象。数据抽象是针对对象属性实现数据封装的，仅为程序员提供了对象的属性和状态的描述，而对象的属性和状态在类外不能被访问。行为抽象是对这些数据所需要的操作进行的抽象。

抽象的过程是以模块化的形式实现的，即通过分析，将一个复杂的系统分解为若干个模块，每个模块都是对整个系统结构中某一部分进行的自包含和完整的描述。由于对模块中的细节进行了信息隐藏，所以使用者只能通过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。

例如，想要在计算机上绘制一个圆形时，通过对这个图形进行分析，可以看出绘制这个图形需要 3 个数据来描述圆的位置（圆心的横坐标、纵坐标）以及圆的大小（半径），这就是对该圆形的数据抽象。因此，绘制圆形时应该具有设置圆心坐标、半径等功能，这就是对它的行为抽象。

用 C++ 语言可以将该图形描述如下：

```
圆形(circle);
数据抽象:
double x,y,r;
行为抽象:
setx();sety();setr(); draw();
```

再例如，要建立一个学生类，学生类中包含学号、姓名、性别、专业等特征。这些特征是学生的属性。学生会进行上课、吃饭、学习等行为。在定义学生类时可以将这些行为特征抽象为方法。

用 C++ 语言可以将上述行为特征描述如下：

```
学生(student)
数据抽象:
int num;
char name;
char sex;
char major;
行为抽象:
have lessons();
eating();
study();
```

（2）封装性。封装是一种信息隐蔽技术，是面向对象方法的重要法则。封装具有两个作用，一是将不同的小对象封装成一个大对象，二是把一部分内部属性和功能对外界隐蔽。

例如，一台计算机是一个大对象，它由主机、显示器、键盘、鼠标等小对象组成，在设计时可以先对这些小对象进行设计，确定各自的属性，然后在它们之间建立相互联系，最后就可以拼装成一台计算机。封装是将事物的属性和行为包装到对象的内部，形成一个独立模块单位，使外界不了解它的详细内情。在面向对象方法中，某些相关的代码和数据被结合在一起，成为一个数据和操作的封装体，这个封装体向外提供一个可以控制的接口，其内部大部分的实现细节对外隐蔽，达到对数据访问权限的合理控制的目的。

封装的目的在于把对象的设计者和使用者分开，使用者可以不必要知道行为实现的细节，只需使用设计者提供的消息来访问该对象。信息隐蔽技术的具体实现是，函数的调用者只需要了解函数的接口信息来正确地使用函数，无须了解函数的具体实现，即函数接口与具体实现是独立的。

例如，有个学生信息管理系统，系统实现中定义了一个学生类，向用户提供了输入学生信息 Input()、输出学生信息 Output()、查询学生学号 Searchnum()、查询学生姓名 Searchname() 4 个接口，而将所用的函数的具体实现和 num、name 等数据隐藏起来，实现数据的封装和隐藏。封装使得程序中各部分之间的相互影响达到最小，提高了程序的安全性，简化了代码的编写工作。

对象是面向对象程序语言中支持并实现封装的机制。对象中既包含数据(即属性)又包含对这些数据进行处理的操作代码(即行为)，它们都称为对象的成员。对象中的成员可以定义为公有成员或者私有成员。

私有成员即在对象中被隐藏的部分，不能被该对象以外的程序访问。

公有成员则提供对象与外界的接口，外界只能通过这个借口与对象发生联系。可以看到，对象有效地实现了封装的两个目标——对数据和行为的包装和信息隐藏。

封装防止了系统间相互依赖而带来的变动影响。面向对象的封装比传统语言的封装更加清晰、有力。面向对象的类是封装良好的模板，类定义将其说明(外部接口)与实现(内部实现)显式地分开，其内部实现按其具体定义的作用域提供保护。封装防止了程序相互依赖性而带来的变动影响。

(3) 继承性。继承是软件复用的一种方式，通过继承，一个对象可以获得另一个对象的属性。继承反映的是对象之间的相互关系，它允许一个新类从现有类中派生而出，新类能够继承现有类的属性和行为，修改或增加新的属性和行为，增添一些自己特有的性质，成为一个功能更强大、更满足应用需求的类。

如图 1-1 所示，东北虎、华南虎、孟加拉虎都属于虎，金钱豹、印度豹都属于豹，家猫、野猫都属于猫，而虎、豹、猫均属于猫科动物，如果不使用层次概念，每个对象都需要明确定义各自的特征。如果通过继承的方式进行描述，一个对象只需要在它的类中定义一些使它成为唯一的属性，其他的通用属性可以从父类中继承。正是由于这种继承机制，才可以使得一个对象可以成为一个通用类的特定实例。一个深度继承的子类将继承它在类层次中的每个祖先的所有属性。

在特殊类中，人们不必考虑继承来的属性和行为，只需着重研究它所特有的性质，这就好像在现实世界中，人们已知房子是建筑物这一概念的继承，则房子这一概念具有建筑物的

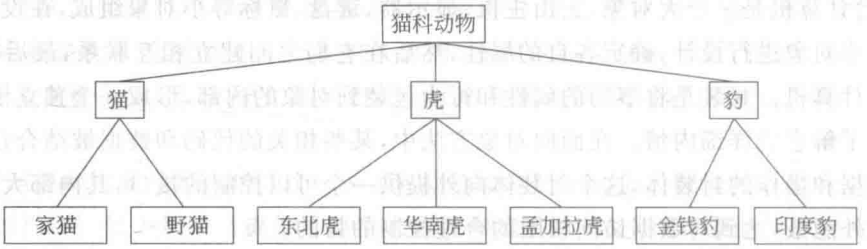

图 1-1　类的继承

所有特点，还同时包含有它自身所特有的一些属性。一个类也可以继承多个一般类的特性，被称为多继承，即每个子类可以有多个父类。

C++ 提供了继承机制，采用继承的方法可以很方便地利用一个已有的类建立一个新的类，这就可以重用已有软件中部分甚至大部分代码，大大节省了编程的工作量，这就是软件重用思想。这样既可以使用自己以前建立的类，又可以使用别人建立的类或是存放在类库中的类。而且只需要对这些类做适当的加工即可使用，大大地缩短了软件开发的周期，对于大型软件的开发具有重要意义。

(4) 多态性。多态是面向对象程序设计的一个重要特征，同一消息被不同的对象接收，可产生完全不同的行为，即“一个接口，多种形态”，这种现象称为多态性。多态性表现为同一属性或操作在一般类或特殊类中具有不同的语义，从一个基类派生出的各个对象具有同一个的接口，因而能响应同一种格式的信息，但是不同类型的对象对该信息的响应方式不尽相同，可产生完全不同的行为。这里所说的消息一般是指对类的成员函数的调用，不同的行为要用不同的函数进行实现。

如果有几个相似而不完全相同的对象，当向它们发出同一信息时，它们的反应会各不相同，分别执行不同的操作。例如，在编制绘图程序时，不同图形的绘制方式是不同的，要先声明一个“几何图形”基类，在该类中定义一个“绘图”行为，并定义该类的“直线”“椭圆”“多边形”等派生类，这些类都继承了基类中的“绘图”行为。在基类的“绘图”行为中，由于图形类型尚未确定，所以并不明确定义如何绘制一个图形的方法而是在各派生类中，根据具体需要对“绘图”重新定义。这样，当对不同的对象发出同一“绘图”命令时，不同对象调用自己的“绘图”程序可绘制出不同的图形。

C++ 的多态性指的是，由继承而产生的相关类中的对象对同一消息会做出不同的响应。多态性是面向对象程序设计的一个重要特征，增加了软件的灵活性和重用性。

1.1.2　为什么要用面向对象程序设计

学习过 C 语言的人都知道，C 语言是面向过程的结构化语言。既然已经有了面向过程程序设计，为什么还要开发面向对象的程序呢？这是因为面向过程程序设计是结构化的程序设计，是以解决问题为基础和重点的，因此在方法上存在不足。在结构化程序设计中，程序被定义为“数据结构＋算法”，数据与处理这些数据的过程是分离的，这样在对不同格式的数据进行相同的处理或者对相同的数据进行不同的处理时，都要用不同的程序模块来实现，这使得程序的可复用性不高。同时，由于过程和数据分离，数据可能同时被多个模块使用和

修改，因此很难保证数据的安全性和一致性。

面向过程程序设计是以数据为中心的，面向过程的功能分解法属于传统的结构化分析法。现实世界中，分析者将对象系统看作一个大的处理系统，然后将其分解为若干个子处理过程，最后解决系统的总体控制问题。在分析过程中，使用数据描述各子处理过程之间的联系，整理各子处理过程的执行顺序。

面向过程程序设计的稳定性、可修改性和可重用性都比较差，对数据类型的检查机制比较弱，语言结构不支持代码重用，程序员在大规模程序开发中很难控制程序的复杂度。随着软件的规模和复杂度的不断增加，面向过程程序设计的结构化程序设计方法已经难以满足大型软件的开发要求，于是在此基础上发展、扩充到对象领域，诞生了 C++ 面向对象程序设计语言。

面向对象程序设计充分体现了分解、抽象、模块化、信息隐蔽等思想，有效地提高了软件生产率，缩短了软件开发时间，提高了软件质量，是控制复杂度的有效途径。

面向对象程序设计的代码易于修改和维护、复用性高、易于扩展、效率更高，能更好地满足用户需求。

1.2 类和对象

1.2.1 类的声明

类的声明就是类的定义。与结构的声明类似，声明一个类所用语法的一般形式如下：

```
class <类名>
{
private:
  <私有成员函数和数据成员的说明>
public:
  <公有成员函数和数据成员的说明>
  };
  <各个成员函数的实现>
```

其中，class 是声明类的关键字；<类名>是标识符，表示声明的类的名字；花括号内部包含的语句块是该类的类体，在类体中可对该类的成员进行定义和说明。类的声明以分号结束。

类声明体内的函数和变量称为这个类的成员，分别称为成员函数和数据成员。数据成员通常是变量或对象的说明语句。成员函数是指函数的定义语句或说明语句。

public、private 等关键字用来说明类的成员的访问控制属性。“public：”之后的成员为公有成员，在类外可以直接引用，提供了类与外界的接口。通常，类的成员函数会被全部或部分定义为公有成员。“private：”之后的成员为私有成员，只能被本类定义的函数引用。通常，将数据成员定义为私有成员，如果在类的定义中既不指定 private，也不指定 public，则系统就默认为是私有的。一个类体中，关键字 public 和 private 可以分别出现多次，即一个