MIANXIANGDUIXIANG CHENGXUSHEJIYUC++SHIXIAN

21世纪高等院校创新教材

面向对象程序设计与C++实现

刘晋萍 编著

MIANXIANGDUIXIANG CHENGXUSHEJIYUC++SHIXIAN

科学出版社
www.sciencep.com

·21世纪高等院校创新教材·

面向对象程序设计与C++实现

刘晋萍　编著

科学出版社

北京

内 容 简 介

本书介绍了关于面向对象的基本概念，并以这些概念为主线介绍了面向对象程序设计的方法和步骤。本书以 C++作为语言依托，介绍了 C++语言实现面向对象程序设计的各种机制，通过大量有针对性的实例，给读者提供了如何将面向对象程序设计的方法步骤与 C++各种语言机制结合的依据。

本书可以作为高等院校计算机、信息及相关专业本、专科学生的教材，同时可供软件开发人员参考。

图书在版编目（CIP）数据

面向对象程序设计与 C++实现/刘晋萍编著. －北京：科学出版社，2006
（21 世纪高等院校创新教材）
ISBN 7-03-017993-5

Ⅰ.面… Ⅱ.刘… Ⅲ.①面向对象语言－程序设计－高等学校－教材
②C 语言－程序设计－高等学校－教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2006）第 104824 号

责任编辑：江 兰／责任校对：王望容
责任印制：高 嵘／封面设计：宝 典

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
湖北京山德新印刷有限公司印刷
科学出版社发行 各地新华书店经销
*
2006 年 8 月第 一 版 开本：787×1092 1/16
2006 年 8 月第一次印刷 印张：19 1/2
印数：1~3 000 字数：480 000
定价：28.80 元
（如有印装质量问题，我社负责调换）

前　言

面向对象成为计算机领域的主流技术已经成为无可置疑的事实。面向对象的出现是程序设计方法学方面的一场实质性革命。在产业界，越来越多的公司从传统的软件开发技术转向面向对象的开发技术，特别是在一些发达国家，几乎所有的新软件开发都全面或部分地采用了面向对象方法和技术。在教育界，面向对象方法和技术以及面向对象程序设计被越来越多的大学列为学生的必修课，社会上各种面向对象技术的培训也大量涌现。在出版方面，也有大量关于面向对象方法和技术特别是面向对象程序设计语言方面的书籍问世。这一切都向人们表明，面向对象在计算机科学技术领域占据了无可争议的主流地位。

面向对象程序设计是面向对象方法从诞生、发展到走向成熟的第一片领地，也是使面向对象的软件开发最终落实的重要阶段。在软件开发的过程中，面向对象程序设计是在面向对象分析和设计模型的基础上进行的，这时，面向对象程序设计的工作比较简单，就是利用相应的面向对象程序设计语言，对面向对象设计模型进行编码。但是，目前在大学开设的关于面向对象方法的有关课程都是从面向对象程序设计开始的。如何在没有面向对象分析和面向对象设计模型的前提下进行面向对象程序设计，即从问题出发直接用某种语言写出面向对象程序，是学习面向对象方法必须经历的过程。对于程序员而言，必须要在系统学习如何对所开发的软件系统进行分析和设计的方法之前，掌握直接对不太复杂的问题进行分析与设计并完成面向对象程序设计的方法，这是本书的出发点。

在众多关于面向对象程序设计的书籍中，绝大多数都是介绍具体语言的，比如关于C++的书籍，都是以语法为主线。而从面向对象的概念出发，以一种面向对象语言为依托，学习面向对象程序设计方法的书籍并不很多。根据本书作者长期从事计算机语言程序设计特别是C和C++语言教学工作的经验，认为走概念和方法相结合的教学和学习路线可以达到更好的效果。当学习和实践者面对不同的面向对象程序设计语言时，可以很轻松地进行语言过渡，因为语言只是实现概念和方法的工具，工具可以变，但概念和方法是一致的。

全书共分为12章，内容结构概括为：面向对象概念——面向对象程序设计的方法和步骤——C++语言结构——C++程序结构——C++面向过程程序设计——C++面向对象程序设计。

第1章概要介绍：面向对象、面向对象方法以及面向对象技术的概念；面向对象程序设计和面向对象程序设计语言的概念以及它们之间的关系；面向对象程序设计在面向对象软件开发过程中的地位和面向对象程序设计语言中的面向对象机制。

第2章介绍面向对象程序设计的步骤、方法路线以及如何用C++实现面向对象程序设计。

这两章是全书的主线，也是全书的纲领。

第3章~第6章，是C++面向过程的语言机制，也是C++面向对象语言机制的基础。对于有学习C语言经验的读者而言，可以加快这几章的学习节奏，重点注意对C的扩充部分。而对于初学者而言，则可按本书安排的顺序进行系统的学习。

第7章~第11章是本书的重点，也是核心，集中介绍了如何体现面向对象的四大基本特性(即抽象性、封装性、继承性和多态性)，如何利用C++实现这四大特性的机制，实现面向对象程序设计的方法。

第 12 章中有与具体语言关系密切的内容，也是程序中必须具备的内容。这一章本书没有过多地介绍与具体语言系统有关的内容，因为这些内容读者可以通过查阅有关文献获取。本书从应用的角度出发，通过具体实例对基本问题涉及的基本方法和语言机制进行了介绍。

概括起来，本书的特点可从两个方面来看：总体特点是以概念和方法为主线，以“怎样的概念和方法对应怎样的语言机制”为思路展开全书内容；具体内容结构清晰、目标明确、循序渐进、实例引导、逐步展开、操作性强。

希望通过本书能使读者明确面向对象程序设计的基本思路、基本方法、基本步骤，在学习面向对象程序设计语言时能有明确的切入点和深入的路线。

本书中所有程序作者都在 Turbo C++ for Windows Version 3.1 版本的编译系统下上机调试通过。

由于时间仓促，书中难免有错漏之处，敬请读者给予批评指正。

编　者

2006 年 7 月

目　录

第 1 章　面向对象概论

面向对象在计算机软件程序设计中是以一种新的思想引入的。在这种思想的影响下，计算机软件的程序设计能够以一种反映人类解决问题的思维方法的过程进行。本章概要介绍：面向对象、面向对象方法以及面向对象技术的概念；面向对象程序设计和面向对象程序设计语言的概念以及它们之间的关系；面向对象程序设计在面向对象软件开发过程中的地位和面向对象程序设计语言中的面向对象机制。这些内容将为后续章节介绍具体的面向对象程序设计方法和技术打下基础。

1.1　面 向 对 象

如果在 20 世纪 80 年代初期以前讨论面向对象这个问题，可以说它是一种新兴的程序设计方法或者是一种新兴的程序设计范型。但自 80 年代以来，面向对象已不仅仅局限在程序设计方法、程序设计语言和编程技术了，而是在此基础上逐步深入到计算机软件领域的几乎所有分支，成为软件系统开发的新的方法论——面向对象方法。随着面向对象作为一种新兴的程序设计方法，软件系统开发的新的方法论地位的确认，面向对象作为一种新的技术被更广泛地应用于计算机软件以外的领域，如人工智能和计算机体系结构等。

综上所述：面向对象是新的程序设计范型——面向对象程序设计方法；面向对象是新的方法论——面向对象方法；面向对象是新的技术——面向对象技术。

1.2　面向对象方法

在计算机软件领域，面向对象方法是从程序设计方法扩大和深入到软件开发层面的新的软件开发方法。尽管面向对象方法还涉及了计算机软件以外的领域，但就目前来看，面向对象方法最主要的应用范围仍然是软件开发，对软件开发周期的各个阶段(包括分析、设计、编程、测试与维护)以及它们所涉及的各个领域(如人机界面、数据库、软件复用、形式化方法、CASE 工具与环境等等)，都已形成或正在形成面向对象的理论与技术体系。本节介绍的面向对象方法，仍然是在计算机软件这一领域范围内。

1.2.1　面向对象方法的定义

在计算机软件领域，面向对象是一整套关于如何看待软件系统与现实世界的关系，以什么观点来研究问题并进行求解，以及如何进行系统构造的软件方法学。这种软件方法学被称为面向对象方法。

面向对象方法的基本思想是：从现实世界中客观存在的事物(即对象)出发来构造软件系统，并在系统构造中尽可能运用人类的自然思维方式，例如抽象、封装、分类、继承、聚合等等。

面向对象方法的基本思想反映出面向对象方法的两个基本点：一是构造软件的出发点和

基本单位；二是构造软件时运用的思维方式。

具体有如下要点：

① 从问题域客观存在的事物出发来构造软件系统，用对象来抽象表示这些事物，并以对象作为系统的基本构成单位。

② 对象有属性和服务两个方面。属性表示事物的静态特征(即可以用一些数据来表达的特征)，服务表示事物的动态特征(即事物的行为)。

③ 封装对象的属性和服务，即将对象的属性和服务结合成为一个独立的实体，对外屏蔽其内部细节。

④ 对事物进行分类。把与事物对应的具有相同属性和相同行为的对象归为一类。

⑤ 运用抽象的原则(或多或少地忽略事物之间的差异)建立类之间的一般-特殊关系。特殊类继承一般类的属性与服务。

⑥ 复杂对象可以用简单对象作为其构成部分，这种构成也称做聚合。

⑦ 对象之间的动态联系通过消息通信来实现。

⑧ 对象之间的静态关系通过关联来表达。

通过以上要点可以看到，在用面向对象方法开发的系统中，对象是系统的基本单位。对象以类的形式进行描述，以类的实例存在于系统中。对象和问题域中的各个事物对应，对象内部的属性与服务刻画了事物的静态特征和动态特征。问题域中事物之间实际存在的各种关系通过类之间的继承关系、聚合关系、消息和关联如实地表达。因此，无论是系统的构成成分，还是通过这些成分之间的关系而体现的系统结构，都可以直接地映射问题域。

综上所述，可对“面向对象方法”作如下定义：面向对象方法是一种运用对象、类、继承、封装、聚合、消息传送、多态性等概念来构造系统的软件开发的方法。

1.2.2 面向对象方法对软件开发的意义

软件开发的实质是对事物的认识与描述。

软件开发是对问题求解的过程。从软件工程学的角度来分析软件开发过程，包括分析、设计、编程、测试和维护等主要活动；从认识论的角度来分析软件开发过程，则归结为两项主要活动，即认识与描述。认识是人们对所要解决的问题及其相关事物的认识，描述则是基于这种认识用一种“语言”将“认识”表示出来。

“认识”主要是“分析、设计”阶段的活动，但编程阶段也包括一定的认识和理解活动，“描述”则是三个阶段都有的工作。分析、设计阶段的描述是指按一定的软件开发模型中分析与设计阶段的表示方法产生分析文档和设计文档；编程阶段的描述是使用一种能够被机器读得懂的语言描述这些分析文档和设计文档，也就是通常所说的编程。

然而，认识是人类的一种思维活动，思维活动都是借助于人们所熟悉的某种自然语言进行的，而描述最终是要用计算机语言进行的，这两种语言之间有很大的距离。软件开发人员必须跨越这个距离，即从思维语言过渡到描述语言。这种过渡如果没有一套可靠的方法，往往要耗费大量精力，并且会造成许多错误。

编程语言的发展大大缩短了这个距离。从低级到高级，编程语言经历了机器语言、汇编语言、高级语言和面向对象语言等阶段。

机器语言是人类学会的第一种编程语言，整个语言只包括两种符号，即“0”和“1”。这种语言离机器最近，机器能直接执行它，然而它却离人类的思维最远，即与思维语言的距离

最大。

汇编语言中使用比较容易理解和记忆的符号构成程序，它比机器语言提高了一步，稍稍符合人类的形象思维，但离机器仍然最近，与思维语言仍然相差很远。

高级语言的出现是计算机编程语言的一大进步，它屏蔽了机器的细节，提高了语言的抽象层次，程序中可以采用具有一定涵义的数据命名和容易理解的执行语句。这使得在书写和阅读程序时可以联系到所描述的具体事物，也使得描述语言向思维语言靠近。

结构化编程语言的出现进一步提高了语言的层次。结构化数据、结构化语句、数据抽象、过程抽象等概念使程序更便于体现客观事物的结构和逻辑涵义，这使得编程语言与人类的自然语言更接近。但二者之间仍有不少差距，主要是程序中的数据和操作不能有效地构成与问题域中具体事物紧密对应的程序成分。

面向对象语言与以往各种语言的根本不同在于它的设计出发点是为了能更直接地描述问题域中客观存在的事物(即对象)以及它们之间的关系。用面向对象语言编写的程序能够比较直观地反映客观世界的本来面目，并且使软件开发人员能够运用人类认识事物所采用的一般思维方法来进行软件开发。这时，面向对象语言和人类认识、理解客观世界所使用的自然语言之间的距离缩小，但二者之间仍有一定的差距。

值得注意的是，缩小或消除语言之间的距离只是解决了描述问题。然而，人们借助自然语言也不一定都能正确地认识客观世界，因为这需要有正确的思维方法。再者，在软件开发过程中，要求人们对问题域的理解比人们日常生活中对它的理解更深刻、更准确。这些就不是编程语言所能解决的问题，而是软件工程学所要解决的问题。软件工程学提供了包括分析、设计、编程、测试、维护在内的一整套软件工程理论与技术体系。

在认识方面，分析阶段提供了一些对问题域的分析认识方法；在描述方面，分析和设计阶段提供了一些从问题域逐步过渡到编程语言的描述手段。这些方法和手段为语言之间的距离铺设了一些平坦的路段。不同的软件工程方法铺设的路段有所不同。传统的软件工程方法中，这些路段并不连续，也就是说，语言之间的距离并没有完全消除。而面向对象的软件工程方法中，从面向对象的分析到面向对象的设计，再到面向对象的编程、面向对象的测试都是紧密衔接的，消除了语言之间的距离。

在面向对象的软件工程方法中，面向对象分析(OOA)阶段以问题域中的事物(即对象)为中心，最大程度地采用与问题域中的事物一致的概念和术语来反映这些对象之间在问题域中原有的各种关系，这使得 OOA 阶段的结果能够直接映射问题域。

在面向对象设计(OOD)阶段，将 OOA 直接搬到 OOD(不经过转换，仅根据实现的需要，作些修改和调整)，作为 OOD 的一部分，其余的部分是针对具体实现中的各种因素补充的一些与实现有关的内容，这些部分采用与 OOA 相同的表示法和模型结构。

面向对象程序设计(OOP)是面向对象方法从诞生、发展到走向成熟的第一片领地，也是面向对象的软件开发最终落实的重要阶段。在 OOA→OOD→OOP 这一软件工程的过程系列中，认识问题域与设计系统成分的工作在 OOA 和 OOD 阶段完成，OOP 工作就是用一种面向对象语言将 OOD 模型中的每个成分写出来。OOP 阶段产生的程序能够紧密地对应 OOD 模型；OOD 模型中的一部分对象类对应 OOA 模型，其余部分的对象类对应与实现有关的因素；OOA 模型中的全部类及对象都对应问题域中的事物。这样的映射关系不但提高了开发工作的效率和质量，而且对开发以后的维护工作具有更长远的意义。由此可见，面向对象的软件工程方法消除了语言之间的距离。

1.2.3 面向对象方法中的主要概念

通过前面的介绍可知，面向对象方法是一种运用对象、类、继承、封装、聚合、消息传送、多态性等概念来构造系统的软件开发方法。下面先介绍与这些概念有关的术语，主要有：

① 对象——属性、服务；
② 类——抽象、实例、一般类、特殊类；
③ 封装——信息隐藏；
④ 继承——单继承、多继承；
⑤ 消息——服务请求；
⑥ 结构与连接——一般-特殊结构、整体-部分结构、实例连接、消息连接；
⑦ 多态性。

1. 对 象

从一般意义上讲，对象是现实世界中一个客观存在的事物，它可以是有形的，比如一辆车、一棵树，也可以是无形的，比如一项功能、一次活动。它是构成世界的一个独立单位，具有自己的静态特征和动态特征。静态特征是可以用某种数据来描述的特征，动态特征是对象表现的行为或对象具有的功能。

在开发一个系统时，通常只是在一定的范围(问题域)内考虑和认识与系统目标有关的事物，并用系统中的对象抽象地表示它们，所以面向对象方法在提到“对象”这个术语时，既可能泛指现实世界中的某些事物，也可能专指它们在系统中的抽象表示，即系统中的对象。这里，主要针对系统中的对象来介绍对象的概念。

对象是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务构成。

属性和服务是构成对象的两个主要因素，其定义为：属性是用来描述对象静态特征的一个数据项，是对对象的一个总体描述；服务是用来描述对象动态特征(行为)的一个操作序列，是对象自身与外界联系的操作。

为了更清楚地说明对象的概念，再给出一个对“对象”下的定义：对象是问题域或实现域中某些事物的一个抽象，它反映该事物在系统中需要保存的信息和发挥的作用，它是一组属性和有权对这些属性进行操作的一组服务的封装体。

例如，可以把球看做一个对象，它的静态属性有球的大小、球的定位点、类别等。动态服务可以有求大小、移动位置、显示类别等。

要说明的是，一个对象的特征是很丰富的，有内在的，也有外在的；有客观的，也有主观的。在确定系统中对象的特征时，只考虑客观事物本质的、与系统目标有关的特征，而不考虑那些非本质的、与系统目标无关的特征。比如，在学生成绩管理系统中，主要考虑学生的成绩特征，而其身高是与成绩无关的特征。

2. 类

类是分类产生的结果。分类是人类在认识客观世界时经常采用的思维方法，就是对众多的事物进行归纳，依据抽象的原则，忽略事物的个别的、非本质性的特征，找出那些与当前目标有关的本质特征，从而确定事物的共性。把具有共同性质的事物划分为一类，得出一个

抽象的概念，例如：水果、树木等都是一些抽象概念，它们是一些具有共同特征的事物的集合，可以被称做类。类的概念使我们能对属于该类的全部个体事物进行统一描述，例如，“树具有树根、树干、树枝和树叶，它能进行光合作用”这个描述适合所有的树，而不必对每棵具体的树都进行一次这样的描述。

因此，类是具有相同属性和服务的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和服务两个主要部分。

类与对象的关系可以看成是抽象与具体的关系。类给出了属于该类的全部对象的抽象定义，而对象则是符合这种定义的一个实体。所以，一个对象又称做类的一个实例。

如果在某个类的范围内考虑定义这个类时舍去的某些特殊性，则在这个类中只有一部分对象具有这些舍去的特殊性，而这些对象彼此是共同的，于是得到一个新的类。它是前一个类的特殊类，而前一个类称做这个新类的一般类。例如：考虑水果这个类，是从众多具体的水果中，舍去了各种水果不同的部分，而保留了共同部分所得到的。如果在水果的范围内考虑定义这个类时舍去的某些特殊性，比如含籽和不含籽，则可举出苹果是含籽的水果，而香蕉是不含籽的水果。苹果和香蕉只是水果类中的一部分对象，而所有苹果彼此是共同的，同样所有香蕉也是共同的，于是就可得到新的类：苹果类和香蕉类。苹果类是水果类的特殊类，香蕉类也是水果类的特殊类，而水果则称为苹果和香蕉的一般类。

一般地，如果类 A 具有类 B 的全部属性和全部服务，而且具有自己特有的某些属性或服务，则 A 叫做 B 的特殊类，B 叫做 A 的一般类。

3. 封　装

封装就是把对象的属性和服务结合成一个独立的系统单位，并尽可能隐蔽对象的内部细节。封装的概念具有两个涵义：第一是把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位。第二是尽可能隐蔽对象的细节，对外形成一个边界，只保留有限的对外接口使之与外部发生联系，这一涵义也叫做“信息隐蔽”。

封装反映了这样一个基本事实:事物的静态特征和动态特征是事物不可分割的两个侧面。系统中把对象看成它的属性和服务的结合体，就使对象能够集中而完整地描述并对应一个具体的事物。封装的信息隐藏反映了事物的相对独立性。当以对象之外的角度观察一个对象时，只需要注意它对外呈现什么行为(做什么)，而不必关心它的内部细节(怎么做)。一旦规定了它的职责后，就不应该随意从外部去改动它的内部信息或干预它的工作。

4. 继　承

由一般类和特殊类的概念可以看到,特殊类的对象自动拥有其一般类的全部属性与服务，而不必重新定义已在一般类中定义过的属性和服务。也就是说，只要两个类的特殊与一般关系成立，特殊类就拥有了一般类的全部属性与服务。这种特性就称为类的继承性。

一个类只是另一个一般类的特殊类时，它只从一个一般类继承属性和服务，这种继承模式叫做单继承模式。然而，一个类还可以是多个一般类的特殊类，它从多个一般类中继承属性与服务，这种继承模式叫做多继承模式。多继承是单继承的推广。

继承简化了人们对事物的认识和描述。比如在认识了轮船的特征之后，再考虑货轮时，因为知道货轮也是轮船，于是可以认为它理所当然地具有轮船的全部一般特征，从而只需要把精力用于发现和描述货轮独有的那些特征上。

继承是实现软件复用的有效途径之一。使特殊类继承一般类，本身就是软件复用。如果将开发好的类作为构件放到构件库中，在开发新系统时便可以直接或继承使用。

5. 消　息

从对象的概念中可知，服务是对象自身与外界联系的一个操作序列。在系统中，对象是通过它对外提供的服务在系统中发挥自己作用的。当系统中的其他对象(或其他系统成分)请求这个对象执行某项服务时，它就响应这个请求，完成指定的服务所应完成的职责。这种向对象发出的服务请求称做消息。

由于对象的封装特性，使对象在系统中成为一些各司其职、互不干扰的独立单位，对象之间通信的唯一合法的动态联系途径就是消息。消息通信使系统中对象的行为能够互相配合，构成一个有机的运动的系统。因此也可以说，因为有了封装，才有了消息。

例如，“冷饮亭”对象描述现实中的一个冷饮亭。它的属性是亭内的各种冰棍、饮料(名称、定价)和钱箱(总金额)，它有两个服务——冷饮零售和款货清点。封装意味着这些属性和服务结合成一个不可分的整体——冷饮亭对象。它对外有一道边界，即亭子的隔板，并留有一个接口，即出售冷饮的窗口，在这里提供冷饮零售服务。顾客只能从这个窗口要求提供服务，而不能自己伸手到亭内拿冷饮和找零钱。货款清点是亭内的一个服务，不向顾客开放，如图 1-1 所示。

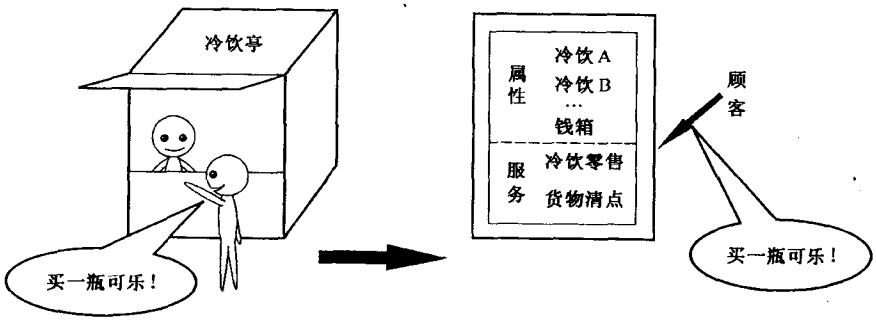

图 1-1　封装和消息

当顾客对着一个冷饮亭说“买一瓶可乐!”，这就是一条消息，它是顾客向冷饮亭发出的服务请求。顾客没有伸手到亭内取可乐，而是通过一个消息达到买可乐的目的。冷饮亭接收到这个消息就执行一次对外提供的服务——冷饮零售。这条消息包含下述信息：接受者，即某个冷饮亭；要求的服务，即冷饮零售；输入信息，即要买的冷饮种类、份数和递进去的钱；回答信息，即买到的冷饮和找回的零钱。

过去商场的售货方式都是这种封闭式的方式，顾客和售货员(商场的代表)之间都是通过消息来达到购货和售货目的的。而现在商场几乎都采用了超市售货方式，打破了封闭的柜台，顾客再不用通过消息来达到购货目的了。

综上所述，消息就是向对象发出的服务请求，它应该含有下述信息：提供服务的对象标识、服务标识、输入信息和回答信息。

值得一提的是，消息的接收者是提供服务的对象，而消息的发送者可以是要求服务的对象或其他系统成分(例如 C++中的 main 函数)。在编程语言中，消息就是函数(或过程)的调用。