21世纪大学计算机专业教材

C++面向对象程序设计

徐宏喆 梁 力 原 盛 编

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

21世纪大学计算机专业教材

C++面向对象程序设计

徐宏喆 梁力 原盛
姚乐 高鹤 王景彬 编

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS
·西安·

内容简介

本书是一本介绍面向对象程序设计中基本概念原理、方法与实现的教材，主要用于本科生学习面向对象程序设计课程及上机实验。本书采用C++语言为基础，VC6.0为实验环境，系统地阐述了面向对象程序设计的特点和思想，旨在使读者迅速迈入面向对象程序设计的大门，同时掌握C++程序设计的基本技能和面向对象程序设计的概念与方法，并能编写出具有良好风格的程序。

本教材共分为9章，并有3个附录。第1章绪论总的介绍面向对象程序设计和C++语言，第2章通过和传统程序设计的比较介绍面向对象程序设计的概念和特性，第3章至第9章，详细阐述了C++支持的面向对象程序设计的基本方法，包括C++语言基础、类、对象、派生与继承、多态性、I/O流、模板等。最后，在附录中介绍了相应的开发环境，并安排了综合与系统的训练，以期扩充学生的知识性，提高其编程能力。

图书在版编目(CIP)数据

C++面向对象程序设计/徐宏喆，梁力，原盛编.
西安：西安交通大学出版社，2007.2
21世纪计算机专业教材
ISBN 978-7-5605-2404-7

Ⅰ.面… Ⅱ.①徐…②梁…③原… Ⅲ.C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(2007)第002206号

书　　名	C++面向对象程序设计
编　　著	徐宏喆　梁力　原盛
出版发行	西安交通大学出版社
地　　址	西安市兴庆南路25号(邮编：710049)
电　　话	(029)82668357　82667874(发行部) (029)82668315　82669096(总编办)
网　　址	http://press.xjtu.edu.cn
电子邮箱	eibooks@163.com
印　　刷	西安市新城区兴庆印刷厂
字　　数	480千字
开　　本	787mm×1092mm　1/16
印　　张	20
版　　次	2007年2月第1版　2007年2月第1次印刷
书　　号	ISBN 978-7-5605-2404-7/TP·485
定　　价	26.00元

前　言

面向对象程序设计是不同于传统程序设计的一种新的程序设计方法，它对降低软件的复杂性，改善其通用性和维护性，提高软件的生产效率都有着十分重要的意义。

C++语言作为同时支持面向对象程序设计和传统程序设计的混合型语言，既保留了C语言灵活高效的特点，又大量引入了面向对象的思想，和独有的语法结构如类，继承，多态等等。C++语言中灵活的变量说明、新的输入/输出流、引用使用等，都大大方便了编程操作，即使在传统程序设计过程中，使用C++语言也会带来更高的效率和安全性。

本书共分为9章和3个附录。书中以准确的语言和丰富的示例程序系统地介绍了C++语言的各种语法结构及其特性。第1章绪论总的介绍面向对象程序设计和C++语言。第2章通过和传统程序设计的比较介绍面向对象程序设计的概念和特性。第3章介绍C++语言的基础语法结构，包括数据结构、表达式、语句和函数，同时还介绍了几种C++语言中特有的语法结构，如引用。第4章介绍面向对象程序设计最基础的概念——类和对象，该章节为本书的核心和基础。第5章介绍静态成员与友元的概念。第6章介绍面向对象程序设计非常重要的概念——派生类与继承，该章节内容也是本书的一个核心和难点。第7章介绍多态性的概念。第8章介绍模板的含义。第9章介绍C++语言全新的I/O流。附录一是VC++6.0的开发环境简介。附录二是综合训练，旨在提高读者的实际程序设计能力。附录三中列出了C++语言中的运算符优先级关系。

本书每一章都有综合训练、小结、练习题和小提示。综合训练包含了这一章的核心语法点，通过对实际题目的分析，编程，逐步向读者介绍使用C++语言编程的方法和经验，力求让读者能够掌握一些实际编程的方法和技巧。小结是对这一章所有内容的回顾，可以帮助读者复习。练习题作为章节内容的补充，读者可以通过完成练习题来巩固各章的内容。小提示则贯穿章节始终，向读者介绍实际编程中的一些技巧。

本书是作者集多年对本科生、研究生C++面向对象程序设计的教学经验，并参阅大量中外资料的前提下完成的。为了使读者能够更好地掌握面向对象程序设计的思想，作者根据多年在教学与科研工作中的心得体会，通过大量生动明晰的实例，运用丰富的图例、生动的语言，向读者一步步揭示和阐述面向对象程序设计的概念与思想，并在全书中运用软件工程的方法、示例，力求使本书通俗实用。

全书从初学者的视角入手，紧紧围绕着如何进行程序设计而展开，由浅入深地使读者掌握面向对象程序设计的思想与方法。生动、形象、丰富的大量实例与每章、每小节都会出现的建议、小提示等内容都是本书的一大特色。

对于本书存在的错误和不足之处，希望广大读者批评指正。

编者

目　录

第 1 章　绪论

软件开发过程是一种高密集度的脑力劳动，需要投入大量的人力、物力和财力。由于早期的软件开发模式及技术不能适应软件发展的需要，致使大量质量低劣的软件产品涌向市场，有的甚至在开发过程中就夭折了。国内外在开发一些大型软件系统时，遇到了许多困难，有的系统最终彻底失败了；有的系统则比原计划推迟了好多年，而且费用大大超过了预算；有的系统不能符合用户当初的期望；有的系统则无法进行修改维护。这些在软件开发工程中出现的情况都称之为“软件危机”。出现软件危机的原因是多方面的，如软件需求变化频繁，开发方法落后等。人们尝试从不同角度、不同层次来解决，比如严格确定软件需求、采用新的开发模型、采用计算机辅助工具等。

面向对象程序设计就是在这一大环境中产生的。在面向对象程序设计语言（以下简称为面向对象语言）产生之后，面向对象程序设计逐步成为编码的主流，其中所蕴涵的面向对象的思想不断向开发过程的上游和下游发展，形成现在的面向对象分析、面向对象设计、面向对象测试等，并一起逐步发展为面向对象软件开发方法。

1.1　软件开发的方法

关于“软件危机”，我们先来看看一些具体的案例。

案例 1：IBM 公司的 OS/360，共约 100 万条指令，花费了 5 000 个人年；经费达数亿美元，而结果却令人沮丧，错误多达 2000 个以上，系统根本无法正常运行。OS/360 系统的负责人 Brooks 这样描述开发过程的困难和混乱：“像巨兽在泥潭中作垂死挣扎，挣扎得越猛，泥浆就沾得越多，最后没有一个野兽能够逃脱淹没在泥潭中的命运。”

案例 2：1962 年 6 月，美国飞往金星的第一个空间探测器（水手Ⅰ号），因其飞行舱中计算机导航程序的一条语句出错，致使空间探测器偏离航线无法取得成功。

案例 3：阿波罗 8 号由于太空飞船的一个计算机软件错误，造成存储器的一部分信息丢失；阿波罗 14 号在飞行的 10 天中，出现了 18 个软件错误。

由以上案例我们可以清楚的知道，在软件开发的过程中一定要像其他行业那样，需要一定的方法来指导开发过程。这就是软件开发方法。软件开发方法很多，一般归结为两种：面向过程的开发方法和面向对象的开发方法。

1.1.1　面向过程的开发方法

面向过程的开发方法包含了分析、设计、实现、确认（测试）、演化（维护）等活动。典型的面向过程的开发方法有：Jackson 方法、结构化开发方法等。其中结构化的开发方法是现有的软

件开发方法中最成熟，应用最广泛的方法。结构化开发方法是一种面向数据流的开发方法，它的基本原则是功能的分解与抽象。结构化方法提出了一组提高软件结构合理性的准则，如分解和抽象、模块的独立性、信息隐蔽等。该方法的主要特点是快速，自然和方便。结构化方法总的指导思想是自顶向下、逐步求精。结构化方法的工作模型采用瀑布模型。

采用瀑布模型，就像瀑布一样，只有前期的工作完成了，才能到下一阶段的工作。即：前期阶段的工作是后期阶段的基础；而后期阶段的工作是前期阶段工作的延续和深化。但从20世纪80年代开始，逐渐发现其不足。面向过程的方法与技术构建一个系统的流程是：从需求出发，制定计划，编写代码，测试代码，维护系统。从这一过程我们可以发现，在做需求分析、制定计划阶段都需要用户的参与，它十分强调前期工作的完美性，如在此阶段有不周到的地方，将给后期的代码编写带来巨大的麻烦，往往可能由于没有弄清用户的需求而使整个开发重来。但是需求和计划有时是连用户也难以在一个具体时间内说清楚的，用户的需求可能是在使用系统时逐步提出的，也可能是在分析阶段就与程序开发人员在理解上有分歧。这样，需要不断地改写代码，但这些代码是面向过程的具体细节的，修改起来十分困难，往往会使程序开发人员陷入代码的泥潭。从思维的方式来看，这是一种自顶向下的开发方法，但是不同的人由于对系统的理解角度不同而得到的“顶”不同，由于对系统的细化程度不同而得到的“底”不同，最终导致实现过程和结果出现巨大差异。

由以上的分析可以看出：瀑布模型将软件开发分割为独立的几个阶段，不能从本质上反映软件开发过程本身的规律。此外，过分强调复审，并不能完全避免较为频繁的变动。尽管如此，瀑布模型仍然是开发软件产品的一个行之有效的工程模型。

1.1.2 面向对象的开发方法

面向对象方法的出现在一定程度上弥补了面向过程方法中的不足。面向对象技术是一种全新设计和构造软件的技术，它使计算机解决问题的方式更符合人类的思维方式，更能直接地描述客观世界，通过增加代码的可重用性、可扩充性和程序自动生成功能来提高编程效率，并且大大减少软件维护的开销，已经被越来越多的软件设计人员所接受。自20世纪80年代末以来，随着面向对象技术成为研究的热点出现了几十种支持面向对象开发方法的软件。

随着Windows操作系统和Internet的普及，面向对象程序设计技术得到越来越广泛的应用，面向对象方法和技术已成为20世纪90年代计算机领域的主流技术。面向对象方法与技术起源于面向对象的编程语言(OOPL)。20世纪80年代大批OOPL的出现和不断提高标志着OO(面向对象)技术开始走向繁荣和实用。20世纪80年代后期到90年代相继出现了一大批关于面向对象的分析与设计的论文和专著。进入20世纪90年代以来，在学术界，面向对象的方法与技术已成为最受关注的研究热点，越来越多的学术会议和学术期刊把面向对象列为主要议题之一，并且每年都有许多关于面向对象的专著出版。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技术，并以此作为提高公司形象和产品信誉的标志。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面向对象技术。这一切都向人们表明，20世纪90年代的面向对象方法已在计算机科学技术领域占据了无可争议的主流地位。

同时，虽然面向对象的方法与技术和面向过程的方法都在一定程度上缓解了“软件危机”现象，但是前者弥补了后者在软件开发中遇到的种种问题。运用面向对象的方法与技术在需

求分析阶段，如果用户对需求不十分清楚，开发人员可以从一个用户清楚的需求出发，构造实现这种需求的对象，规定其操作，由多个对象抽象为类，由类构造超类，由类派生出实例，一步步逐步构造系统。由于对象与其操作是封装的，所以在对某个对象修改时只涉及该对象、该类的细节，不影响整个系统。这是一种自底向上的开发方法，它没有过多的纸上的完美性要求，用户看到的是与系统更加贴近的内容。

提示：为了克服“软件危机”，提高软件的开发效率，人们不断地从实践经验和理论中探索软件工程的新途径。上面提及到的面向过程和面向对象开发方法是互不排斥的，而是相互补充相互促进的。我们在实践的软件开发过程中，要根据实践情况来选择适当的开发方法，这样才能促进软件的开发成功。

1.2 面向对象的概念及其程序设计

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。其基本思想是从现实世界中客观存在的事物（即对象）出发来构造系统，并在系统构造中尽可能运用人类的自然思维方式。

1.2.1 面向对象方法的基本概念

在所有的面向对象方法中有两个重要的且最为基础的概念——类与对象。在面向对象的程序设计中，“对象”是系统中的基本运行实体，是有特殊属性（数据）和行为方式（方法）的实体。即对象由两个部分构成：一组包含数据的属性和允许对属性中包含的数据进行操作的方法。也可以说，“对象”是将某些数据代码和对该数据的操作代码封装起来的模块，是有特殊属性（数据）和行为方式（方法）的逻辑实体。对象包含了数据和方法，每个对象就是一个微小的程序。

类是对具有公共的方法和一般特殊性的一组基本相同对象的描述。一个类实质上定义的是一种对象类型，由数据和方法构成，它描述了属于该类型的所有对象的性质。对象是在执行过程中由其所属的类动态生成的，一个类可以生成不同的对象。在面向对象的程序设计中，对象是构成程序的基本单位，每个对象都应该属于某一类。对象也可称为类的一个实例。

1.2.2 面向对象的程序设计

面向对象的程序设计就是用一种面向对象的编程语言（比如C++）把软件系统书写出来。在面向对象编程中，程序被看作是相互协作的对象集合，对象间的通讯是通过消息来实现的，每个对象都是某个类的实例，所有的类构成一个通过继承关系相联系的层次结构。面向对象的编程方法有四个基本特征：

（1）抽象。抽象就是忽略一个主题中与当前目标无关的那些方面，以便充分地注意与当前目标有关的方面。

（2）继承。继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。这种特征体现了一般和特殊的关系。继承很好地解决了软件的可重用性问题。

（3）封装。封装是面向对象的特征之一，是对象和类概念的主要特性。封装是把过程和

数据包围起来，对数据的访问只能通过已定义的界面。封装保证了模块具有较好的独立性，使得程序维护修改较为容易。

(4) 多态性。多态性是指允许不同类的对象对同一消息作出响应。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好地解决了应用程序的函数同名问题。

1.2.3 面向对象开发技术的优点

现在越来越多的企业和软件开发人员全部或者部分采用面向对象技术来开发他们的软件系统。因为面向对象开发技术有许多的技术优势。

(1) 与人类的思维习惯类似。面向对象技术对问题空间进行自然分割，以更接近人类思维的方式建立问题域模型，以便对客观实体进行结构模拟和行为模拟，从而使设计出的软件尽可能直接地描述现实世界。

(2) 具有良好的稳定性。运用面向对象技术，软件开发时间短，效率高，所开发的程序更稳定。由于面向对象编程的可重用性，可以在应用程序中大量采用成熟的类库，从而缩短了开发时间。

(3) 可重用性好。应用程序更易于维护、更新和升级。继承和封装使得应用程序的修改带来影响更加局部化。

面向对象技术有诸多的优点，但是，如果采用面向对象方法而没有很好的软件开发工具支持，那么就会增加开发的难度。为此，本书采用当前比较流行的C++开发工具——VC++(Microsft Visual C++)作为本书所有例子的开发工具。

1.3 面向对象语言(C++)及开发工具

C++语言是从C语言中孕育出来的。C语言是一种面向过程的语言，以其灵活和高效而著称。Bjarne Stroustrup对C语言的内核进行了必要的修改，使其满足了面向对象模型的要求。于是一种新的语言——C++语言应运而生。

C++的一个设计目标就是通过支持面向对象来增强C语言的功能。也就是说C++只是在C的基础上添加了一些面向对象概念的新语法规则，这些添加并不影响原来C语言所有的语法规则。所以C++是完全兼容C的，原来用C编程的程序可以不做修改或者做很少的修改就可以在C++环境中正确的执行。因此，也可以把C++认为是一个混合型的语言，它既支持结构化的编程(C++包含C的全部语法规则)，同时也支持面向对象的编程(C++添加了适应面向对象的语法)。

VC++是目前较为流行的C++集成开发环境(IDE)。该环境是由Microsoft公司开发的。该开发环境除了提供标准的C++语言的库函数以外，还提供了MFC(微软基础类库)，方便用户创建一些高级特性的类，在一定程度上减少了开发人员写任何一个类都要从头开始写的重复劳动。

在后续章节里，我们将可以看见在VC++里面创建一个类是很轻松的一件事情。因为我们指定了类名以后，VC++就会帮助我们自动地生成一些代码。除了自动生成代码以外，VC++还给我们提供了一个良好的编译，连接，执行功能，我们在VC++中书写完代码以后，就可以马上点击工具栏中的相应的按钮执行相应的功能。值得称道的是VC++为开发

人员提供了图形界面的调试环境，开发人员可以设定相应的断点，跟踪所有的变量的变化，从而为我们程序排错，编写出优秀的代码提供了便利的条件。具体内容见附录一。

思考与练习题

1. 软件开发方法包括哪些？
2. 试比较面向过程的开发方法和面向对象的开发方法的优缺点？
3. 试列出面向对象的编程方法的基本特征？
4. 面向对象开发技术有哪些优点？
5. 什么是 VC＋＋？

第 2 章　面向对象的程序设计

面向对象程序设计方法(Object Oriented Progvamming,OOP)的主要出发点是弥补面向过程程序设计方法中的一些缺点。OOP 把数据看作程序开发中的基本元素,并且不允许它们在系统中自由流动。它将数据和操作这些数据的函数紧密地连结在一起,并保护数据不会被外界的函数意外的改变。OOP 允许我们将问题分解为一系列"实体"——这些"实体"被称为对象(object),然后围绕这些实体建立数据和函数。

2.1　对象与类

2.1.1　对象与类的概念

面向对象不仅是一种设计编程的方法,同时也是一种看待世界和分析问题的方法。前面提到过面向对象方法思考的出发点接近人的思维方式,也就是用人类习惯的方式来描述要解决的问题。人类在认识事物时最基本的单元——对象,就成了面向对象方法中最基本的一个概念。

对象是客观世界中的一个实体。可以认为,整个世界就是由许许多多的不同的对象构成的。例如一个人,一辆车都是一个对象。应该注意的是,对象是一个具体存在的实体,不是抽象的概念。例如一辆黄色的奔驰轿车是一个对象,但"车"这个抽象的概念并不是一个对象(应该是一个类,这个在后面讲类的概念的时候会提到),只有具体化的真实存在的实体才可以被叫做对象。

对象包含以下的几个内容:

(1)对象的名字。客观世界中没有两个实体是完全一样的,同样的,当我们用对象来表示这些实体的时候,也要用不同的名字来将对象加以区分。

(2) 对象的属性。属性是对实体某一方面的描述。实体和实体的区别就在于属性的不同,这种不同表现为两种方式,一种是属性的种类不同,比如学生具有学号这个属性,老师就没有这个属性;第二种不同是属性的内容不同,例如两个学生都有学号这个属性,但学号不同。

(3) 对象的操作。对象的操作指的是对象能够进行的行为。例如一辆黄色的奔驰轿车这个对象就具有行驶这个操作。

从对象的内容我们可以看出,数据(属性)和操作被紧密的联系在一起,这就克服了结构化设计中数据和操作分离的弱点。同时这种方式也比较符合人的认识观,人在看到小汽车的时候很自然地会联想到它的功能(比如行驶),客观实体的属性和操作在人脑中本来就是被放在一起处理的,用对象这种认识问题的方法有利于程序设计者描述复杂的现实问题。

下面是一个对象的例子：

对象名称：小明

对象属性：

学历：大学

年龄：21

专业：历史系

对象操作：

上课

吃饭

人在认识世界的各个对象时，并不是把每个对象都当作一个孤立的个体的，比如一只黄猫和一只白猫是两个不同的对象，人们在认识它们的时候会认为它们都是猫，也就是说人在认识对象的时候往往能看到对象间的共性，通过这些共性对对象进行分类，这反映在面向对象方法中就是“类”的概念。

类可以说是对象的模型，用一个模型便能建立许多类似的对象。这种关系就类似于月饼和月饼模具，一旦制好了一个月饼模具，就可以成批地制作相同的月饼了。

下面是个类的例子：

类名称：学生

类属性：

学历

年龄

专业

对象操作：

上课

吃饭

我们可以把这个类的例子和前面对象的例子做个比较：

(1) 类是抽象的，类的名称并不指某个实体，例如上例中的类名称“学生”是指一个范围的人，而不是某一个人；而对象是具体的，对象的名称各不相同。

(2) 类的属性其实是一个范围，是不确定的，像上个例子中的属性年龄，只表示一个范围一般取 1～120；而对象也有年龄属性，但对象的年龄属性是个确定的值例如 21。

(3) 类和对象都有操作，且二者的操作是相同的。

从上面的比较我们可以看出，类与对象的关系就是抽象与具体的关系。类是一个框架，是对具有同类型属性和操作的对象组的抽象；对象则是类的事实例化，给类这个框架中的属性填上一组确定的值就可以得到一个对象。

2.1.2　对象的交互

在面向对象设计中，我们可以看到所有解决问题的函数都被放进了一个个类(和对象)中，也就是说功能依然是被分解了，只是和结构化的分解方法不同。面向对象把功能函数按照其所属的对象分类，也就是说，哪个对象实现这个功能，这个功能的函数就分给哪个对象(注意在面向对象中函数被称为方法)。这样的分解使得功能函数不再按实现功能的结构来排列，完成

一个功能可能要涉及到多个对象中的方法，一个对象往往不能解决问题，需要几个对象协作来完成。

程序的执行顺序。在结构化编程中，主控制程序掌握着执行的顺序，因为所有的函数都是被主函数逐级调用返回的，函数都是按照功能分模块排放的，解决一个问题所用的函数都放在一起，呈现分层的关系，控制流程很简单。但面向对象中方法是按照对象排放的，当一个方法想调用其他对象中的方法的时候，这两个对象就要发生交互，所以面向对象的控制流程相对复杂。这种复杂正是对客观世界中复杂的实体关系的反映。这就好比军队和企业。军队的管理制度就像结构化设计方法，上下级明确，下级绝对服从上级，完成上级指派的任务，最上层的领导很容易控制全局；企业的管理制度则更像面向对象的设计方法，各个部门之间大多属于平行的关系，如果一个项目要跨几个部门，就需要几个部门进行交互，没有哪个部门有绝对的主领导权，每个部门都只做好自己应做的部分，再把需要别的部门做的任务传送给相应的部门并等待回应。

对象之间的交互既然是不可避免的，那么对象之间的交互以什么样的形式进行呢？答案就是消息传送机制。但这种传递只是抽象的叫法，并不是真的发送一个完整的消息，一个对别的对象中的方法的调用也是消息的传送。这里的消息发送并不强调发送形式。消息发送只是一种通知，告诉其他对象要使用其中的方法。这种程序执行控制的方法和现实中解决一个问题的方法类似。在生活中，当我们遇到我们不能做的事情的时候也会发消息给其他人来帮我们解决。例如，我们去买火车票，我们只是给窗口工作人员发送了一个口头消息，消息包括了车票的信息，然后就等待工作人员来完成登记打印车票等一系列任务了。

可以看出，面向对象的设计靠消息发送来推动程序的进行，这种方法间的耦合关系显然远远低于结构化设计中的函数调用。但这样的执行控制形式是由对象的特性和封装性(下一节将提到)决定的。面向对象方法是使用信息发送的方式来完成对象交互，将分散在各个对象中的方法联合起来解决问题。

2.2 数据的抽象与封装

2.2.1 现实世界中的抽象与封装

在现实世界中，抽象和封装是非常普遍的一种现象。用对象的观点来看世界，各种数据和操作都被对象分隔开来。封装是对象对其内部数据和操作的遮蔽。例如：在购买火车票这个例子中，存在两个对象，购票者和工作人员。购买车票是通过这两个对象的交互来完成的。对于购票者来说，他只需要对工作人员描述清楚他对车票的要求(两个对象之间的一次消息发送)，工作人员就可以完成其他工作，返回给购票者一张正确的车票。在这个过程中购票者对工作人员进行了什么样的操作并不了解，对工作人员如何进行操作也不了解，如工作人员是否进行了车票登记，怎么登记，这些购票者是看不到的。这些操作和数据只有工作人员才知道，也就是说工作人员把他能进行的操作和数据对外隐藏了起来，这就是对象的封装性。更重要的是，购票者在发送完车票信息后就在等待一张正确车票的返回，购票者对工作人员进行了怎样的操作来完成给出一张正确车票并不关心。也是说购票者关心工作人员“能做什么”(在本例子中购票者关心工作人员具有输出一张车票的功能)，而工作人员关心“怎么做”因为工作人