

21 世纪高等学校计算机专业核心课程规划教材

# Visual C++ 面向对象程序设计教程与实验 (第 3 版)

## 学习指导与习题解答

温秀梅 高丽婷 孟凡兴 编著

清华大学出版社

THE UNIVERSITY OF CHICAGO PRESS

# Global Call to Action

Volume 3  
of the  
Global Call to Action

2014-2015

21 世纪高等学校计算机专业核心课程规划教材

# Visual C++ 面向对象程序设计教程与 实验(第 3 版)学习指导与习题解答

温秀梅 高丽婷 孟凡兴 编著

清华大学出版社  
北京

## 内 容 简 介

C++面向对象程序设计是一门实践性非常强的课程,为了更好地培养学生的程序设计能力和综合应用能力,并配合“C++面向对象程序设计”课程的教学需要,我们编写了本书。

本书内容包括两部分。第1部分为学习指导与课后习题参考答案。按照教材的结构,对应教材的每一章节,分析了重点和难点,归纳了各章的知识点,对一些经典例题进行解答指导,使学生掌握解题的技巧和基本方法,用于巩固和强化基本概念和重点知识。对教材中每一章节的课后习题给出了参考答案。第2部分为实验参考答案,给出了教材附录C中各实验内容的参考答案,用以提高学生的实践能力和应用能力。

本书适合作为计算机科学与技术及相关专业的“C++面向对象程序设计”课程的辅导教材,也可供读者自学使用和参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

Visual C++面向对象程序设计教程与实验(第3版)学习指导与习题解答/温秀梅等编著. --北京:清华大学出版社,2014

21世纪高等学校计算机专业核心课程规划教材

ISBN 978-7-302-36751-2

I. ①V… II. ①温… III. ①C语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第124156号

责任编辑:魏江江

封面设计:杨 兮

责任校对:时翠兰

责任印制:李红英

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:14.75 字 数:355千字

版 次:2014年10月第1版 印 次:2014年10月第1次印刷

印 数:1~2000

定 价:29.50元



# 前言

本教材以 C++ 面向对象程序设计的思想和方法作为重点,结合例题进行详细的分析解释。该门课程对于培养计算机专业学生的计算机应用能力起了重要的作用。为了更好地培养学生的程序设计能力和综合应用能力,并配合“C++ 面向对象程序设计”课程的教学需要,我们编写了本书。

为了简单起见,本书中的所有参考程序均做成了单文件形式。学生在上机实践过程中,各个习题最好做成多文件结构,多文件结构参见教材 3.1.2 节。每个习题和实验均有很多种解题方法,本书给出的解法只起抛砖引玉之作用,读者可在学习的过程中,尝试做更深入、广泛的研习。

本书中的所有程序都是在 Visual C++ 6.0 环境下编译调试通过的,各部分内容相互配合。使用本书,对于面向对象程序设计课程的学习具有重要的促进作用,对于分析问题、解决问题能力的提高不无裨益。

本书在编写过程中,编者参阅了许多 C++ 参考书和有关资料,并阅读了一些外文教材,现谨向这些书的作者表示衷心的感谢。

本书由河北建筑工程学院温秀梅、高丽婷、孟凡兴编著,司亚超、任亮(石家庄二中)任副主编,全书由温秀梅教授进行审校并统稿。

本书在编写和出版过程中得到了清华大学出版社的大力支持和帮助,在此表示诚挚的感谢!

由于作者水平有限,书中难免有疏漏、错误和不妥之处,恳请广大专家和读者批评指正,我们将不胜感激。

编者

2014 年 3 月



# 目 录

## 第 1 部分 学习指导与习题参考答案

<b>第 1 章 绪论</b> .....	2
1.1 学习要求 .....	2
1.1.1 基本要求.....	2
1.1.2 基本知识点.....	2
1.1.3 重点和难点.....	2
1.2 内容概要 .....	2
1.2.1 结构化程序设计与面向对象程序设计.....	2
1.2.2 面向对象程序设计的基本概念.....	4
1.2.3 面向对象系统的四大特性.....	5
1.2.4 面向对象的系统开发步骤.....	5
1.3 典型例题解析 .....	6
1.4 课后习题解答 .....	6
<b>第 2 章 C++ 语言对 C 语言的扩充</b> .....	9
2.1 学习要求 .....	9
2.1.1 基本要求.....	9
2.1.2 基本知识点.....	9
2.1.3 重点和难点.....	9
2.2 内容概要.....	10
2.2.1 名字空间 .....	10
2.2.2 动态内存的分配与释放 .....	10
2.2.3 引用 .....	10
2.2.4 C++ 语言中函数的新特性 .....	11
2.3 典型例题解析.....	11
2.4 课后习题解答.....	12



<b>第3章 类和对象</b> .....	15
3.1 学习要求 .....	15
3.1.1 基本要求 .....	15
3.1.2 基本知识点 .....	15
3.1.3 重点和难点 .....	15
3.2 内容概要 .....	16
3.2.1 类和成员函数的定义 .....	16
3.2.2 C++的多文件结构 .....	17
3.2.3 对象 .....	17
3.2.4 构造函数和析构函数 .....	17
3.2.5 类的聚集——对象成员 .....	19
3.2.6 静态成员 .....	20
3.3 典型例题解析 .....	20
3.4 课后习题解答 .....	23
<b>第4章 友元</b> .....	36
4.1 学习要求 .....	36
4.1.1 基本要求 .....	36
4.1.2 基本知识点 .....	36
4.1.3 重点和难点 .....	36
4.2 内容概要 .....	36
4.2.1 友元的概念和定义 .....	36
4.2.2 友元函数 .....	37
4.2.3 友元成员 .....	37
4.2.4 友元类 .....	37
4.3 典型例题解析 .....	37
4.4 课后习题解答 .....	39
<b>第5章 继承与派生</b> .....	41
5.1 学习要求 .....	41
5.1.1 基本要求 .....	41
5.1.2 基本知识点 .....	41
5.1.3 重点和难点 .....	41
5.2 内容概要 .....	42
5.2.1 单一继承 .....	42
5.2.2 多重继承 .....	43
5.2.3 虚基类 .....	44
5.2.4 类模板 .....	44

5.3	典型例题解析	45
5.4	课后习题解答	46
<b>第6章</b>	<b>多态性和虚函数</b>	<b>66</b>
6.1	学习要求	66
6.1.1	基本要求	66
6.1.2	基本知识点	66
6.1.3	重点和难点	66
6.2	内容概要	67
6.2.1	运算符重载	67
6.2.2	虚函数	69
6.2.3	纯虚函数和抽象类	70
6.2.4	虚析构函数	71
6.3	典型例题解析	71
6.4	课后习题解答	74
<b>第7章</b>	<b>C++语言的输入输出流库</b>	<b>94</b>
7.1	学习要求	94
7.1.1	基本要求	94
7.1.2	基本知识点	94
7.1.3	重点和难点	94
7.2	内容概要	94
7.2.1	C++语言输入输出流	94
7.2.2	文件输入输出流	95
7.3	典型例题解析	96
7.4	课后习题解答	97
<b>第8章</b>	<b>异常处理</b>	<b>105</b>
8.1	学习要求	105
8.1.1	基本要求	105
8.1.2	基本知识点	105
8.1.3	重点和难点	105
8.2	内容概要	105
8.3	典型例题解析	107
8.4	课后习题解答	109
<b>第9章</b>	<b>Windows编程基础和MFC编程基础</b>	<b>112</b>
9.1	学习要求	112
9.1.1	基本要求	112





9.1.2  基础知识点	112
9.1.3  重点和难点	112
9.2  知识点归纳	112
9.3  课后习题解答	113
<b>第10章 对话框和控件</b>	<b>119</b>
10.1  学习要求	119
10.1.1  基本要求	119
10.1.2  基础知识点	119
10.1.3  重点和难点	119
10.2  知识点归纳	119
10.3  课后习题解答	120
<b>第11章 菜单和文档/视图结构</b>	<b>131</b>
11.1  学习要求	131
11.1.1  基本要求	131
11.1.2  基础知识点	131
11.1.3  重点和难点	131
11.2  知识点归纳	131
11.3  课后习题解答	132
<b>第12章 图形设备接口</b>	<b>134</b>
12.1  学习要求	134
12.1.1  基本要求	134
12.1.2  基础知识点	134
12.1.3  重点和难点	134
12.2  知识点归纳	134
12.3  课后习题解答	135

## 第2部分 实验参考答案

<b>实验1 熟悉实验环境</b>	<b>144</b>
实验目的	144
实验内容	144
<b>实验2 C++对C的扩充</b>	<b>147</b>
实验目的	147
实验内容	147

<b>实验3 类和对象</b> .....	154
实验目的 .....	154
实验内容 .....	154
<b>实验4 友元</b> .....	193
实验目的 .....	193
实验内容 .....	193
<b>实验5 继承和派生</b> .....	203
实验目的 .....	203
实验内容 .....	203
<b>实验6 多态性和虚函数</b> .....	210
实验目的 .....	210
实验内容 .....	210
<b>实验7 I/O 流库和异常处理</b> .....	220
实验目的 .....	220
实验内容 .....	220
<b>实验8 对话框和控件</b> .....	221
实验目的 .....	221
实验内容 .....	221
<b>实验9 菜单和文档/视图结构、图形设备接口</b> .....	222
实验目的 .....	222
实验内容 .....	222
<b>参考文献</b> .....	223



**第 1 部分**

**学习指导与  
习题参考答案**

## 1.1 学习要求

### 1.1.1 基本要求

- (1) 了解面向对象方法的起源；
- (2) 掌握什么是面向对象、面向对象程序设计；
- (3) 掌握结构化程序设计和面向对象程序设计的区别；
- (4) 掌握面向对象的基本概念：对象、类、消息和方法；
- (5) 掌握面向对象系统的四大特性：抽象性、封装性、继承性和多态性；
- (6) 了解面向对象的系统开发方法和过程。

### 1.1.2 基本知识点

- (1) 结构化程序设计和面向对象程序设计的区别；
- (2) 对象、类、消息和方法的基本概念；
- (3) 对象的特性；
- (4) 面向对象系统的四大特性：抽象性、封装性、继承性和多态性；
- (5) 面向对象的系统开发方法和过程。

### 1.1.3 重点和难点

#### 【重点】

- (1) 面向对象程序设计与结构化程序设计的区别；
- (2) 面向对象的基本概念；
- (3) 面向对象系统的四大特性。

#### 【难点】

面向对象的系统开发步骤。

## 1.2 内容概要

### 1.2.1 结构化程序设计与面向对象程序设计

#### 1. 结构化程序设计

结构化程序设计的设计思想是：自顶向下、逐步求精；其程序结构按功能划分为若干

个基本模块,这些模块形成一个树状结构;各模块之间的关系尽可能简单,在功能上相对独立;每一模块内部均由顺序、选择和循环三种基本结构组成;其模块化实现的具体方法是使用子程序、过程或函数。

程序的任务是描述问题并解决问题,在结构化程序设计中可以用下面的式子表示程序:

$$\text{程序} = \text{数据结构} + \text{算法} + \text{程序设计语言} + \text{语言环境}$$

结构化程序设计中程序的结构如图 1-1-1 所示。

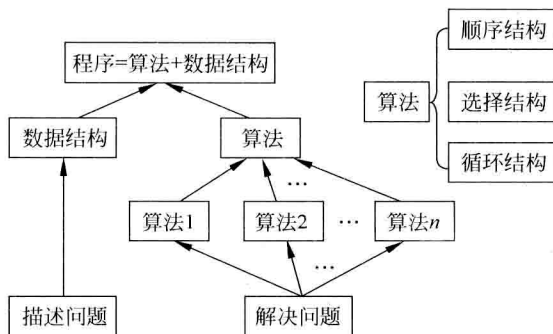


图 1-1-1 结构化程序设计中程序的结构

## 2. 面向对象程序设计——程序设计的新思维

面向对象方法是从系统的组成上来进行分解的,对问题进行自然分割,以更接近人类思维的方式简化问题域模型,从而使设计出的软件尽可能直接地描述现实世界。

面向对象程序设计将数据及对数据的操作放在一起,作为一个相互依存、不可分割的整体来处理,它采用了数据抽象和信息隐藏技术。在面向对象程序设计中可以用下面的式子表示程序:

$$\text{程序} = \text{对象} + \text{对象} + \dots + \text{对象}$$

$$\text{对象} = \text{算法} + \text{数据结构} + \text{程序设计语言} + \text{语言环境}$$

面向对象程序设计中程序的结构如图 1-1-2 所示。

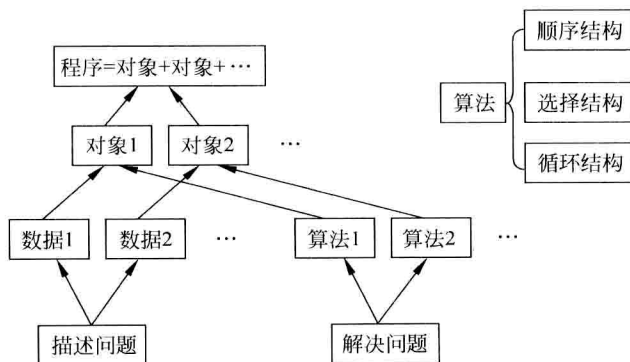


图 1-1-2 面向对象程序设计中程序的结构

## 1.2.2 面向对象程序设计的基本概念

### 1. 类

在面向对象系统中,类(class)是一种用户自定义的数据类型,在类中封装了程序中的数据以及对数据进行的操作。类是对象的抽象及描述,是具有共同属性和操作的多个对象的相似特性的统一描述体。

### 2. 对象

在面向对象系统中,对象(object)是类类型的变量,是内存中的一块区域。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。对象具有如下特性。

- (1) 具有唯一标识名,可以区别于其他对象;
- (2) 具有一个状态,由与其相关联的属性值集合所表征;
- (3) 有一组操作方法即服务,每个操作决定对象的一种行为;
- (4) 一个对象的成员可以是另一个对象;

(5) 模块独立性:模块内部状态不因外界的干扰而改变,也不会涉及其他模块;模块间的依赖性极小或几乎没有;各模块可独立地被系统所组合选用,也可被程序员重用,不必担心破坏其他模块;

(6) 动态连接性:在面向对象程序设计中,通过消息机制,把对象之间动态连接在一起,使整个机体运转起来,称为对象的动态连接性;

(7) 易维护性:由于对象的修改、完善功能及其实现的细节都被局限于该对象的内部,不会涉及外部,这就使得对象和整个系统变得非常容易维护。

### 3. 消息

消息(message)是面向对象系统中实现对象间的通信和请求任务的操作,是要求某个对象执行其中某个功能操作的规格说明。发送消息的对象称为发送者,接收消息的对象称为接收者。对象间的联系,只能通过消息来进行。对象在接收到消息时才被激活。

消息具有以下三个性质。

- (1) 同一对象可接收不同形式的多个消息,产生不同的响应;
- (2) 相同形式的消息可以送给不同对象,所做出的响应可以是截然不同的;
- (3) 消息的发送可以不考虑具体的接收者,对象可以响应消息,也可以对消息不予理会,对消息的响应并不是必需的。

对象之间传送的消息一般由三部分组成:接收对象名、调用操作名和必要的参数。

### 4. 方法

在面向对象程序设计中,要求某一对象做某一操作时,就向对象发送一个相应的消息,当对象接收到发向它的消息时,就调用有关的方法(method),执行相应的操作。方法就是对象所能执行的操作。方法包括界面和方法体两部分。方法的界面就是消息的模式,它给出了方法的调用协议;方法体则是实现某种操作的一系列计算步骤,也就是一段程序。消息和方法的关系是:对象根据接收到的消息,调用相应的方法;反过来,有了方法,对象才能响应相应的消息。所以消息模式与方法界面应该是一致的。同时,只要方法界面保持不变,方法体的改动不会影响方法的调用。在C++语言中方法是通过函数来实现的,称为成员函数。

### 1.2.3 面向对象系统的四大特性

#### 1. 抽象性

面向对象程序设计鼓励程序员以抽象的观点看待程序,即程序是由一组对象组成的。可以将一组对象的共同特征进一步抽象出来,从而形成“类”的概念。抽象(abstract)是一种从一般的观点看待事物的方法,它要求程序员集中于事物的本质特征,而不是具体细节或具体实现。类的概念来自人们认识自然、认识社会的过程。在这一过程中,人们主要使用两种方法:从特殊到一般的归纳法和从一般到特殊的演绎法。在归纳的过程中,从一个个具体的事物中把共同的特征抽取出来,形成一个一般的概念,这就是“归类”;在演绎的过程中,又把同类的事物,根据不同的特征分成不同的小类,这就是“分类”。对于一个具体的类,它有许多具体的个体,称这些个体为“对象”。

#### 2. 封装性

所谓数据封装(encapsulation)就是指一组数据和与这组数据有关的操作集合组装在一起,形成一个能动的实体,也就是对象。

#### 3. 继承性

从已有的对象类型出发建立一种新的对象类型,使它继承原对象的特点和功能,这种思想是面向对象设计方法的主要贡献。继承(inheritance)是对许多问题中分层特性的一种自然描述,因而也是类的具体化和被重新利用的一种手段,它所表达的就是一种对象类之间的相交关系。它使得某类对象可以继承另外一类对象的特征和能力。继承所具有的作用有两个方面:一方面可以减少代码冗余;另一方面可以通过协调性来减少相互之间的接口和界面。

从继承源上划分继承可分为单一继承和多重继承。子类对单个直接父类的继承叫做单一继承。子类对多个直接父类的继承叫做多重继承。父类也叫基类或超类,子类也叫派生类。

#### 4. 多态性

不同的对象接收到相同的消息时产生多种完全不同的行为的现象称为多态性(polymorphism)。C++语言支持两种多态性,即编译时的多态性和运行时的多态性。编译时的多态性通过重载函数实现,而运行时的多态性通过继承性和虚函数实现。

### 1.2.4 面向对象的系统开发步骤

面向对象的系统开发分为以下四个步骤。

(1) 系统调查和需求分析:对应用系统将要实现的功能以及用户对系统开发的需求进行调查研究。这是所有开发方法都必须进行的。

(2) 面向对象分析(Object Oriented Analyse, OOA):分析问题的性质和求解问题,在繁杂的问题域中抽象地识别出对象及其行为、结构、属性、方法等。

(3) 面向对象设计(Object Oriented Design, OOD):整理问题,即对分析的结果作进一步的抽象、归类、整理,最终以规范的形式描述对象和类。

(4) 面向对象程序设计(Object Oriented Programming, OOP):程序实现,即用面向对象的程序设计语言将上一步整理的对象和类的描述映射为应用程序软件。

## 1.3 典型例题解析

**【例 1-1-1】** 面向对象系统的四大特性分别是抽象性、封装性、继承性和多态性。

**【例 1-1-2】** 简述类和对象的关系。

**【答】** 类和对象之间的关系是抽象与具体的关系。类是一种用户自定义的数据类型,是具有相同数据结构和相同操作的一类对象的描述。对象是类的实例,是类类型的变量,是计算机内存区域中的一块空间。

**【例 1-1-3】** C++语言是纯粹的面向对象语言吗?

**【解析】** C++语言是一种混合型的语言,既支持结构化程序设计,又支持面向对象程序设计。

**【答】** 不是,至少它还含有 main() 这个全局函数。

## 1.4 课后习题解答

1. 什么是面向对象程序设计? 它与传统的结构化程序设计有什么不同?

**【解】** 面向对象程序设计既吸取了结构化程序设计的一切优点,又考虑了现实世界与面向对象解空间的映射关系,它所追求的目标是将现实世界的问题求解尽可能简单化。

面向对象程序设计将数据及对数据的操作放在一起,作为一个相互依存、不可分割的整体来处理,它采用了数据抽象和信息隐藏技术。它将对象及对对象的操作抽象成一种新的数据类型——类,并且考虑不同对象之间的联系和对象所在类的重要性。

面向对象程序设计优于传统的结构化程序设计。其优越性表现在,它有希望解决软件工程的两个主要的问题——软件复杂性控制和软件生产率的提高,此外它还符合人类的思维习惯,能够自然地表现现实世界的实体和问题,对软件开发过程具有重要的意义。

在面向对象程序设计中可以用下面的式子表示程序:

$$\text{程序} = \text{对象} + \text{对象} + \dots + \text{对象}$$
$$\text{对象} = \text{算法} + \text{数据结构} + \text{程序设计语言} + \text{语言环境}$$

在结构化程序设计中可以用下面的式子表示程序:

$$\text{程序} = \text{数据结构} + \text{算法} + \text{程序设计语言} + \text{语言环境}$$

2. 面向对象程序设计语言有哪几类?

**【解】**

(1) LISP 家族

LISP 是 20 世纪 50 年代开发出来的一种语言,它以表处理为特色,是一种人工智能语言,20 世纪 70 年代以来,在 LISP 基础上开发了很多 LISP 家族的面向对象语言。

(2) Simula

Simula 语言是 20 世纪 60 年代开发出来的,在 Simula 中引入了几个面向对象程序设计语言中最重要的概念和特性,即数据抽象、类和继承性机制。Simula 67 是它具有代表性的一个版本,20 世纪 70 年代发展起来的 CLU, Ada, Modula-2 等语言也是在其基础上发展起来的。



### (3) Smalltalk

Smalltalk 是第一个真正的面向对象程序设计语言,起源于 Simula 语言。它体现了纯粹的 OOP 设计思想,是最纯粹的 OOP 语言。尽管 Smalltalk 不断完善,但在那个时期,面向对象程序设计语言并没有得到广泛的重视,程序设计的主流仍是结构化程序设计。

### (4) C 家族

在 20 世纪 80 年代,C 语言成为一种极其流行、应用非常广泛的语言。C++ 是在 C 语言的基础上进行扩充,并增加了类似 Smalltalk 语言中相应的对象机制。它将“类”看作是用户定义类型,使其扩充比较自然。C++ 以其高效的执行效率赢得了广大程序设计员的青睐,在 C++ 中提供了对 C 语言的兼容性,因此,很多已有的 C 程序稍加改造甚至不加改造就可以重用,许多有效的算法也可以重新利用。它是一种混合型的面向对象程序设计语言,由于它的出现,才使面向对象的程序设计语言越来越得到重视和广泛的应用。

Java 语言是一种适用于分布式计算的新型面向对象程序设计语言,可以看作是 C++ 语言的派生,它从 C++ 语言中继承了大量的语言成分,抛弃了 C++ 语言中冗余的、容易引起问题的功能,增加了多线程、异常处理、网络程序设计等方面的支持,掌握了 C++ 语言,可以很快学会 Java 语言。

## 3. 面向对象系统有哪些特性? 分别加以解释。

### 【解】

#### (1) 抽象性

抽象(abstract)是一种从一般的观点看待事物的方法,它要求程序员集中于事物的本质特征,而不是具体细节或具体实现。类的概念来自人们认识自然、认识社会的过程。在这一过程中,人们主要使用两种方法:从特殊到一般的归纳法和从一般到特殊的演绎法。在归纳的过程中,从一个个具体的事物中把共同的特征抽取出来,形成一个一般的概念,这就是“归类”;在演绎的过程中,又可以把同类的事物,根据不同的特征分成不同的小类,这就是“分类”。对于一个具体的类,它有许多具体的个体,称这些个体为“对象”。

#### (2) 封装性

所谓数据封装(encapsulation)就是指一组数据和与这组数据有关的操作集合组装在一起,形成一个能动的实体,也就是对象。数据封装就是给数据提供了与外界联系的标准接口,无论是谁,只有通过这些接口,使用规范的方式,才能访问这些数据。数据封装是软件工程发展的必然产物,使得程序员在设计程序时可以专注于自己的对象,同时也切断了不同模块之间数据的非法使用,减少了出错的可能性。

#### (3) 继承性

从已有的对象类型出发建立一种新的对象类型,使它继承原对象的特点和功能,这种思想是面向对象设计方法的主要贡献。继承(inheritance)是对许多问题中分层特性的一种自然描述,因而也是类的具体化和被重新利用的一种手段,它所表达的就是一种对象类之间的相交关系。它使得某类对象可以继承另外一类对象的特征和能力。继承所具有的作用有两个方面:一方面可以减少代码冗余;另一方面可以通过协调性来减少相互之间的接口和界面。

#### (4) 多态性

不同的对象接收到相同的消息时产生多种完全不同的行为的现象称为多态性(polymorphism)。C++ 语言支持两种多态性即编译时的多态性和运行时的多态性。编译时