

第8章 面向对象程序设计

目录

CONTENTS

8.1	面向对象程序设计基础
8.2	类和对象
8.3	类的继承和多态
8.4	面向对象应用实例

8.1 面向对象程序设计基础

现实生活中的每一个相对独立的事物都可以看做一个对象。例如，一个人，一辆车，一台电脑等。
对象是具有某些特性和功能的具体事物的抽象。每个对象都具有描述其特征的**属性**及附属于它的**行为**。

面向对象程序设计是一种计算机编程架构，
它具有以下3个基本特性。

(1) 封装性 (Encapsulation)

(2) 继承性 (Inheritance)

(3) 多态性 (Polymorphism)

(1) 封装性 (Encapsulation)

封装性就是将一个数据和与这个数据有关的操作集合放在一起，形成一个实体——对象。用户不必知道对象行为的实现细节，只需要根据对象提供的外部特性接口访问对象即可。目的在于将对象的用户与设计者分开，用户不必知道对象行为的细节，只需用设计者提供的协议命令对象去做就可以。

(2) 继承性 (Inheritance)

在面向对象程序设计中，根据既有类（基类）派生出新类（派生类）的现象称为类的继承机制，亦称为继承性。

派生类无需重新定义在父类（基类）中已经定义的属性和行为，而是自动地拥有其父类的全部属性与行为。派生类既具有继承下来的属性和行为，又具有自己新定义的属性和行为。当派生类又被它更下层的子类继承时，它继承的及自身定义的属性和行为又被下一级子类继承下去。面向对象程序设计的继承机制实现了代码重用，有效地缩短了程序的开发周期。

(3) 多态性 (Polymorphism)

面向对象程序设计的多态性是指基类中定义的属性或行为，被派生类继承之后，可以具有不同的数据类型或表现出不同的行为特性，使得同样的消息可以根据发送消息对象的不同而采用多种不同的行为方式。

Python完全采用了面向对象程序设计的思想，是真正面向对象的高级动态编程语言，完全支持面向对象的基本功能，如封装、继承、多态。但与其他面向对象程序设计语言不同的是，Python中对象的概念很广泛，Python中的一切内容都可以称为对象。例如，字符串、列表、字典、元组等内置数据类型都具有和类完全相似的语法和用法。创建类时用变量形式表示的对象属性称为数据成员或成员属性，用函数形式表示的对象行为称为成员函数或成员方法，成员属性和成员方法统称为类的成员。



总结



总结

深刻理解面向对象程序设计的3个基本特征：封装性、继承性、多态性。



THANKS
