

## ③ 面向对象程序设计与结构化程序设计的比较

379-382

刘晓霞

TP311.56

(西北大学计算机科学系, 710069, 西安; 31岁, 女, 讲师)

**A 摘要** 利用面向对象程序设计(OOP)方法开发复杂系统具有较好的可靠性、可重用性、可扩展性和可维护性。比较了OOP与SPP(结构化程序设计), 讨论了OOP的优点和不利因素, 以及如何进行面向对象程序设计方法及由SPP转向OOP方法。

**关键词** 面向对象程序设计; 结构化程序设计; 对象; 类; 封装; 继承

**分类号** TP311.56

## 1 面向对象基本概念

面向对象基于以下4个重要概念, 即对象、类(封装和信息隐蔽)、继承(包括属性和行为)和多态。

一个对象(object)是基本的运行实体, 它既包括属性, 又包括行为。因此, 一个对象把属性和行为封装成为一个整体。从程序设计者的观点来看, 对象是一个程序模块; 从用户观点来看, 对象为他们提供了所希望的行为。在对象内的操作通常叫方法(method)或成员函数(member function)。

类(class)是用来描述一组对象的共同行为和属性的, 就像一个模具。类具体化就可生成对象。类具有层次结构, 即一个类可以有父类, 也可以有子类。

继承(inheritance)是父类和子类之间共享数据和方法的机制。

对象和对象之间的通讯是按照一定的协议通过消息传递来完成的。在收到消息时, 对象要予以响应。不同的对象收到同一消息可能产生完全不同的结果, 这一现象称为多态(polymorphic)。

图1, 图2用银行储蓄实例来说明上述一些基本概念。

```
class csavings_account
{ public: // 成员函数说明
    csavings_account(float initbalance, float intrate);
        // 构造函数
    ~csavings_account(); // 析构函数
    float get_balancefn();
    float get_intratefn();
    void set_intratefn(float newrate);
    float depositfn(float newamount);
    float withdrawfn(float newamount);
    void compute_intfn();
private: // 数据成员说明
    float balance;
    float interestrate;
}; // 类 csavings_account 说明结束
```

图1 对象类 csavings\_account 的说明

Fig.1 Specification of Object Class csavings\_account

由图 1 知,对象类的说明,不仅包括属性,如 `balance`, `interesterate`, 而且还包括其上的操作即成员函数,如 `public` 中所说明的。数据隐藏是通过说明成员(数据成员和成员函数)的类型来实现的。按照访问权限,成员函数有 3 种类型,即公有的(`public`), 私有的(`private`)和被保护的(`protected`)。任何函数都可访问公有成员,只有类的成员才能访问私有成员,当某个成员是被保护的,它只能被基类的成员以及由它导出的类的成员访问而不能被非成员函数访问。

图 2 中, `current_account` 和 `esavings_account` 都是 `bank_account` 的子类。`especial_account` 是 `current_account` 和 `esavings_account` 的子类。称 `current_account` 和 `esavings_account` 是单一继承,因它们都只有一个父类,而 `especial_account` 是多重继承,因其有两个父类。

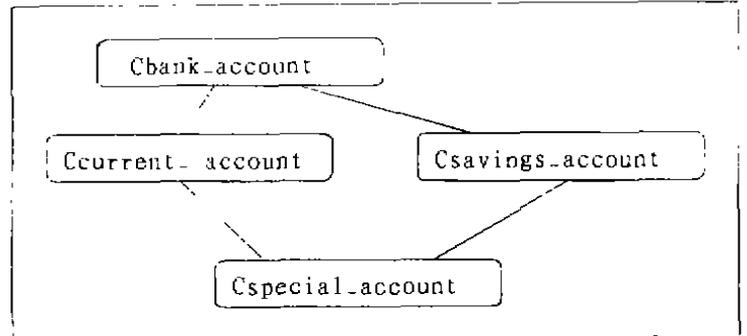


图 2 类层次结构

Fig. 2 Class Hierarchies

不同于 SPP, OOP 使用消息激活方法来改变对象的状态。一个消息是使对象进行某个操作或改变对象状态的命令。

在图 1 中, `compute_intfn()` 是使对象计算利息并把结果加到结余上的操作。而 `set_intratefn()` 是使对象状态发生改变的函数。`csavings_account` 和 `especial_account` 都可调用 `compute_intfn()` 函数,但其运行结果显然不同,这就是多态。

## 2 SPP 方法和 OOP 方法的比较

结构化方法强调功能抽象和模块性,它将解决问题的过程看作是一个处理过程。而面向对象方法则综合了功能抽象和数据抽象,它将解决问题看作是一个分类演绎过程。

面向对象方法模拟了人类认识问题的较高、较广层次的过程,即分类过程;结构化方法则模拟了人类的思维过程,处于人类认识问题的较深层次过程。

### 2.1 模块与对象

结构化方法中模块是对功能的抽象,每个模块都是一个处理单位,它有输入和输出。而对象是包括数据和操作的整体,是对数据和功能的抽象和统一。可以说,对象包括了模块概念。

### 2.2 过程调用与消息传递

在结构化方法中,过程为一独立实体,显式地为它的使用者所见。而在 OOP 方法中,方法是隶属于对象的,它不是独立存在的实体,而是对象的功能体现。另外,过程调用所涉及到的过程及函数都属于同一个程序实体,无论它们是以一个文件还是以多个文件存放。而不同对象间的消息传递则是不同程序实体之间的交互与协作。可以这样说,面向对象方法中的程序实体是松耦合的,结构化方法中的程序实体是紧耦合的。

由图 3 可看出, SPP 使用过程调用, OOP 使用消息传递。但一对象的方法是基本结构和 SPP 中使用的基本结构是一样的,无非是顺序结构,选择结构和循环结构。

### 2.3 类型与类

类型与类都是数据和操作的抽象,即定义了一组具有共同特征的数据和可以作用于其上的一组操作,但类型是偏重于操作抽象的,而类是集成数据抽象和操作抽象的,二者缺一不可。同时,类引入了继承性质,实现了可扩充性。

### 2.4 静态连接与动态连接

从程序设计发展来看,用户对灵活性和方便性的要求不断增强,所以动态连接代替静态连接是必然趋势。

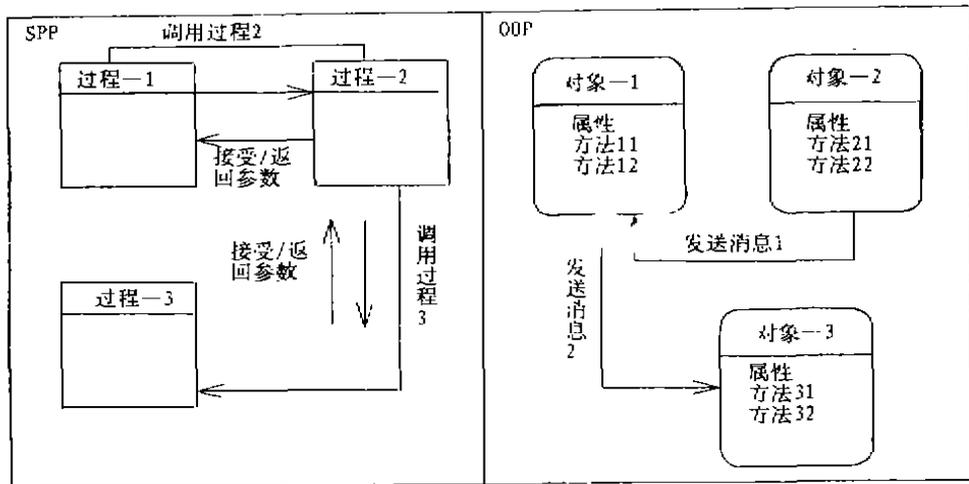


图 3 模块间通讯方式的比较

Fig. 3 Comparison of the Communication between Modulos in SPP and OOP

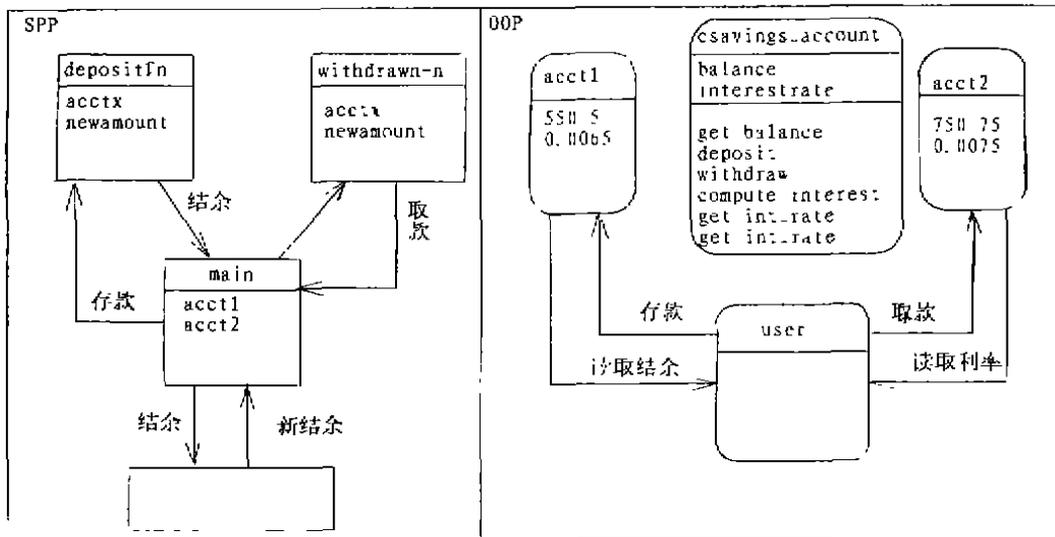


图 4 银行储蓄问题具体实现的比较

Fig. 4 A Comparison of SPP and OOP for the Bank Account Problem

### 3 如何进行面向对象程序设计

一般来说,开发一面向对象程序需经过以下步骤:①用简单的描述性语言描述用户需求;②确定对象及其属性(数据成员),同时确定与对象类相关的数据操作(成员函数);③把系统划分成模块,每个模块由一个或多个对象类组成;④通过封装属性和操作说明每一个对象类;⑤根据每个对象的要求确定相互作用的对象间的消息传递;⑥根据对象类间的依赖关系确定类继承关系和类层次关系;⑦建立系统的OO逻辑模型来表示对象间的相互作用;⑧对对象的成员函数写出其算法;⑨使用面向对象程序设计语言对每一个模块进行代码实现;⑩确定测试、维护计划。

### 4 OOP 的优点

以下这些优点是 OOP 所特有的,使用 SPP 是无法实现的。

- OOP 能较好地模拟现实世界,如图 4 右所示。
  - 抽象数据类型概念以及与其相关的封装,信息隐藏原理,增强了软件的可靠性及可扩展性。
  - 多态性和动态联编增加了代码的灵活性和可重用性。
  - 继承性允许代码是可扩展和可重用的,因为不用修改原代码,通过创建子类就可增加新的属性和操作。
  - OOP 允许用软件的可重用部件构造新的应用软件。
- 为了很好地利用 OOP 以上这些优点,必须克服以下障碍:
- OOP 软件开发及测试工具还不够成熟,有待于标准化,维护环境有待于提高。
  - SPP 中使用的 ERD(Entity-Relationship Diagram)和 DFD(Data Flow Diagram)需要改进以适应 OO 的类和对象分解。
  - 在实际的应用中,OO 类层次是非常复杂且难以维护的。
  - 多态性及动态联编需较长的处理时间。
- 这些问题是现实世界的反映,随着对面向对象技术(OT)研究的深入,相信会逐步解决这些问题的。

## 5 结束语

OOP 将是以后软件设计的主流方法,它与 SPP 方法结合应用是目前软件开发与设计的最好方法。OOP 思想应加强推广应用。

### 参 考 文 献

- 1 蔡希尧,陈平.面向对象技术.西安:西安电子科技大学出版社,1993
- 2 Booch G, Object \_Oriented Development. IEEE Trans. Software Eng. , 1986(2):211~221
- 3 Wilde N, Matthews P, Huitt R. Maintaining Objat \_Oriented Software. IEEE Software. Jan. 1993:75~80
- 4 Wirfs-Brock B J, Johnson R E. Surveging Current Research in Objected-Oriented vesign. Comm. ACM. 1990,33(9):Sept,104~124
- 5 Henderson-Sellers B, Edwards J M. The Objat-oriented Systems Life Cycle. Comm. ACM. 1990,33(9):142~159

责任编辑 张素敏

## A Comparison of SPP and OOP Approaches

Liu Xiaoxia

(Department of Computer Science, Northwest University, 710069, Xi'an)

**Abstract** The object-oriented programming (OOP) approach has the advantage of producing better structured and more reliable software for complex systems, greater reusability, more extensibility, and easier maintainability. Structured procedural programming (SPP), on the other hand, is well understood, well defined, and widely used for developping software. The two paradigms are compared, and how the SPP methodology migrates to OT.

**Key words** Object-oriented programming; structured procedural programming; object; class encapsulation; inheritance