

面向对象程序设计思想在大型基坑 设计分析软件开发中的应用

李维涛 应惠清 袁波

(同济大学土木工程学院,上海 200092)

【摘要】 面向对象程序设计思想是当前一种全新的程序设计思想,它具有传统结构化程序设计思想无可比拟的优点。结合当前基坑设计分析软件的特点,提出采用面向对象程序设计思想进行大型基坑设计分析软件的开发观点,并就面向对象技术在其中的具体应用进行了探讨和分析。

【关键词】 面向对象;程序设计方法;基坑设计

【中图分类号】 TU 311.4

The Application of OOP on Development of Design and Analyses Software of Foundation Pit

Li Weitao Ying Huiqing Yuan Bo

(Construction Staff Room of Building Engineering of Tongji University, Shanghai 200092 China)

【Abstract】 The thought of OOP, which is a new design thought of program, the basic concept of OOP is introduced. Considered the character of design software of foundation pit, the point that the thought of OOP is applied to develop the software of foundation pit is brought forward, meantime the detailed application of OOP is discussed and analyzed.

【Key Words】 oriented object; design of program; design of foundation pit

0 引言

当前用于基坑设计分析的软件多为二维平面分析软件,这些软件多是基于传统的结构化程序设计(SP)思想开发的,每次只能同时对一种类型的支护结构进行分析验算。但实际上,同一个基坑往往同时采用多种不同形式支护结构进行支护。因此,当用软件进行基坑的设计分析时,通常的做法是将基坑中不同类型支护结构分离出来进行单独分析。这显然同实际情况不相符,另外,在进行不同类型支护结构分析过程中,使用者往往多次重复输入一些相同的参数信息,从而浪费了时间。因此,进行大型基坑分析软件的开发是十分必要的。

面向对象设计思想是在结构化程序设计思想基础上发展起来的一种全新的程序设计思想,它突破了传统程序设计的概念,简化了程序设计过程,增强了软件的可扩充性和重用性。采用该思想进行大型基坑设计分析软件的开发,将软件开发单元同实际工程对象相联系,概念明确,便于非专业人员的开发。同时可以大大地提高软件的开发效

率。

1 面向对象技术基本概念和特征

面向对象程序设计技术,又称 OOP 技术,是一种在分析和设计阶段独立于程序设计语言的概念化过程。传统的结构化程序设计(SP)思想将程序的数据和对数据的操作分离开来,这往往造成数据和操作程序之间的错误调用的概率,同时程序代码的通用性差,可移植性和扩展性也不好。而 OOP 技术采用数据抽象和封装技术将数据及对数据的操作放在一起作为一个整体来处理,力求做到人们认识客观事物的过程同程序实现一致^[1]。

OOP 技术中三个重要的基本概念就是:对象(Object)、类(Class)和消息(Message)。对象是自身所具有的状态特征及可以对这些特征施加的操作结合在一起所构成的对立实体。对象是 OOP 技术的核心,它由私有数据和操作这些数据的代码组成。图 1 指出了 SP 和 OOP 中数据和代码的关系;在对象里的数据只能由对象的代码来访问,数据是对象所私有,它不能被其它对象所修改。

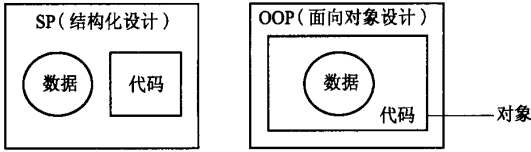


图1 SP和OOP数据代码的关系

类是一组对象的抽象,即类是一个抽象数据类型
的实现,它抽取了该种对象所具有的共同特征。根据
对象的相似性,可以把对象划分成类和子类。即相似
对象的集合称为类。OOP技术就是建立在对象和类
的基础之上的。对象除了拥有自身的特征和行为外,
还需要同其它对象发生联系。而消息就起到传递这
个联系的作用。消息是对象之间相互请求或相互协
作的途径,是要求某个对象执行其中某个功能操作
的规格的说明。OOP技术有封装性、多态性和继承性
三个基本特征。封装是将一个数据和与这个数据操
作有关的集合放在一起形成一个实体对象。对象的
所有数据、内部程序细节被限制在清楚的边界内,其
他对象不能对其进行修改。继承性表达了一种对象
之间的相交关系,它使得某类对象可以继承拥有另
一类对象的功能和特征。通过继承一方面减少了程
序代码的冗余,另一方面协调了各对象之间的关系,
减少了对象之间的接口和界面。多态性是指同一
个对象可以对不同的发送消息采取不同的行为方式
来响应,这是面向对象系统的另一个重要特征,它
大大减少了子程序和子函数的个数,方便了子程序
的管理。

2 大型基坑设计分析软件开发中的面向对象思想

2.1 基坑工程系统概念设计

采用OOP技术进行软件的开发,其软件生命周
期一般包括四个阶段:分析阶段、设计阶段、演化
阶段和维护阶段(见图2)。其中分析阶段是指在问
题域中选出词汇建立类和对象的模型;设计阶段是
对问题域的行为进行关键抽象再分解的过程;演化
阶段是进行编码、测试和集成组合的阶段;维护阶
段是指系统提交运行后的变更活动^[2,3]。

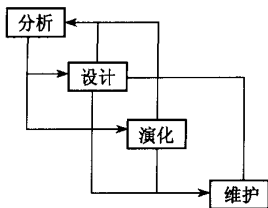


图2 OOP设计软件生命周期

软件的分析阶段又称为概念设计,即在
问题域中找出关键抽象并且建立操作他们的机制及

各对象之间的关系,这也是面向对象思想的精髓所
在。概念设计一般遵循以下顺序:首先对问题进行
分析建立起问题的概念模型,其次通过概念模型抽
象出一系列的对象及对象之间的关系,然后对抽象
出的对象进一步归纳,建立类的概念。最后以类为
基元完成整个程序的开发。基坑工程系统的概念
设计过程见图3。

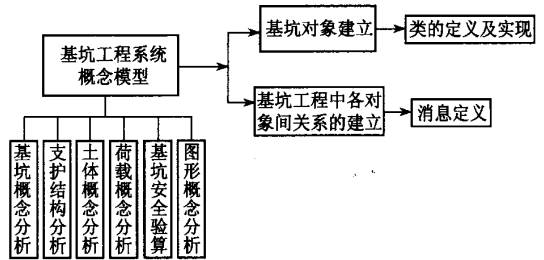


图3 基坑工程系统概念设计

2.2 基坑工程系统中的类

根据概念模型,抽象出基坑工程系统中的一些
类并采用类图表示见图4^[4]。

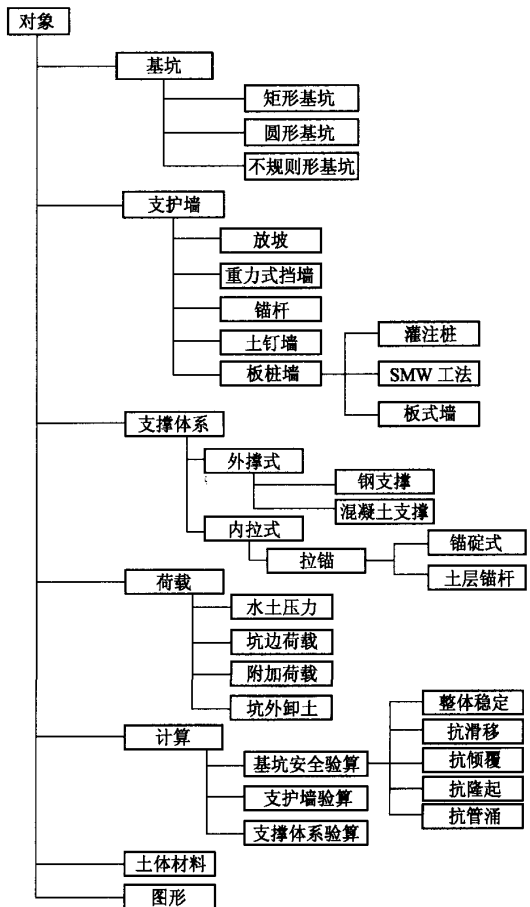


图4 基坑工程系统类图

2.3 对象关系图

对象关系见图 5。

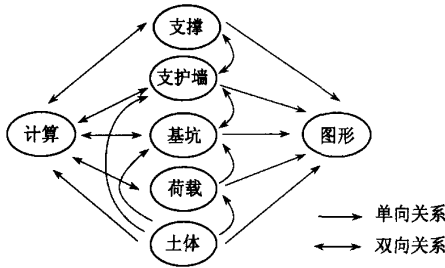


图 5 对象关系图

2.4 类的建立

OOP 技术的核心就是类,抽象出类的概念后,还须归纳各类的属性及行为,建立类的实体。由图 4 可以看出,类是有层次性的,处于第一层次上的称为基类,第一层以下的各层称为子类。子类可以由基类派生。下面列出基坑工程系统中一些基类的部分的属性和行为(见表 1)。

表 1 基类一览表

基 类	属 性	行 为
基坑	基坑平面内不同支护段的段数、各段起始点坐标、 基坑各不同支护段深度、支护类型代码等	获取支护段数、获取支护段类型代码、 删除该段支护等
支护墙	墙顶、底标高,坑内挖土深度、 墙体插入深度、墙体坡度等	获取墙体标高、计算插入深度、获取墙后 土层信息、计算墙后土压力等
墙体支撑	支撑标高、支撑的间距、支撑长度、支撑同围檩在 平面内和竖向夹角、换撑刚度及位置标高	获得支撑参数、 计算支撑内力等
荷载	荷载方向、荷载位置坐标、荷载类型代码	获取荷载类型代码、方向、坐标等
土体材料	土体重度、土体凝聚力、土体摩擦角、土层顶、 底标高,本构模型代码等	添加、删除土体,计算各层土体上压力、 获得土体本构模型代码等
图形	图形区域、图元类型、图元坐标等	定义绘图区域、修改图元等
计算	计算类型代码、计算所需各参数、计算结果	从土体、支护结构、荷载等类中 获得计算参数、安全验算等

上表归纳出基坑工程模型中的一些基类的基本情况,其子类可以在继承基类的特征及行为基础上,根据自身的情况又可以进一步拓展成为新类。下面

以支护墙基类为例,通过其下类板桩墙子类的建立说明子类生成以及子类和基类的相互关系(见图 6)。其余子类均可按照此思想进行生成。

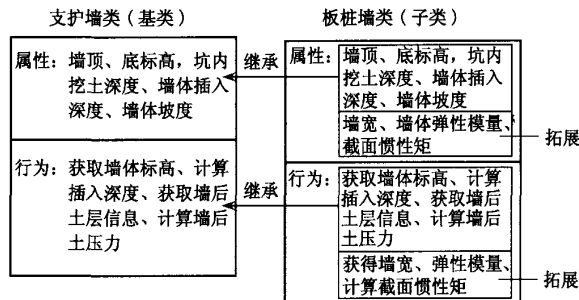


图 6 子类生成及子类和基类的相互关系

2.5 类的定义及实现

类的概念模型被抽象出以后,需要运用一定的格式进行描述类,并通过面向对象语言将其转化为机器可识别的形式,这个过程就是类的定义和实现过程。根据当前类的描述语言 CDL 格式,从格式上看,类的定义一般由五个部分组成,其意义如下:

Class:类名 //用于描述类

- {
- Superclass: 上位类表 //描述上位类,即基类
- Attribute: 属性定义 //说明类的属性
- Method: 方法名(消息表) //描述类的行为

```
Request: 请求消息表//用于向另一个类发出消息
```

上述类定义中的五个组成部分并不是必要的,其具体组成要根据类的自身情况和开发语言有所不同。目前,OOP设计开发语言有许多种,其中较为流行的当属C++家族,如Borland C++、Visual C++等等。除此之外,还用其他的一些OOP设计语言,如Visual Basic等。

下面以VC++为面向对象开发语言,给出了三个类的具体代码实现。

1) 基坑布置类

```
Class PitAr
{
    int NP;//定义基坑段
    float Lx(2), Ly(2), h, PitKind;//定义基坑段起始坐标、深度、基坑种类
public:
    PitAr();//构造设定对象初始值
    void Print();//显示基坑参数
    float GetL();//获得基坑起始坐标、深度
    int GetPit();//获得基坑种类
    ~PitAr();//析构释放已设定值
}
```

2) 支护墙类

```
Class WallSupport
{
    float top, bottom, H, HD, Angle;//定义支护墙的墙顶、底标高,坑内挖土深度、墙体插入深度、墙体坡度
public:
    WallSupport();//构造设定对象初始值
    void Print();//显示支护墙参数
    float GetPara();//获得支护墙参数
    float GetSoil();//获得土体参数
    void PressSoil( float, float, float, ... float);//
```

计算土体侧压力

```
~ WallSupport();//析构释放已设定值
}
```

3) 板桩墙类

```
Class BanWall: WallSupport//继承支护墙所有的特征、行为
{
    float BanW, E, EI;//定义板桩墙的宽度、弹性模量、惯性矩等新属性
public:
    float GetPara();//获得板桩墙的参数
    void CalEI();//计算截面惯性矩
}
```

注意,上面各类的实现代码中省去了类中行为函数的定义,具体定义形式可参照C++相关书籍。

3 结论

运用面向对象程序设计思想进行大型软件的开发具有明显的优点,其思想已被广大专业程序设计人员所接受。但对于非专业程序开发者而言,仍采用结构化的程序设计思想,由此造成开发的软件,尤其一些专业性软件存在许多固有缺陷。

采用面向对象程序设计思想进行大型基坑设计分析软件开发,其对象和类的概念使得程序建模更接近实际情况,避免了结构化程序设计中复杂的数据抽象建模过程。本文作者基于OOP技术,对其在大型基坑设计分析软件开发中的应用进行了探讨和分析,并给出了其思想的具体应用和实现,为大型基坑分析软件的开发提供了参考。

参 考 文 献

- 1 王 燕. 面向对象的理论与C++实践. 北京:清华大学出版社,1997. 27~35
- 2 孙 涌. 现代软件工程. 北京:北京希望电子出版社,2002. 110~113
- 3 史济民. 软件工程原理、方法与应用. 北京:高等教育出版社,2004. 97~104
- 4 赵志缙,应惠清. 简明深基坑工程设计施工手册. 北京:中国建筑工业出版社,2000. 70~80

收稿日期:2005-05-13