第3章 建立面向对象的编程思想

主要知识点

面向对象编程的基本思想；

面向对象编程的一般方法；

运用 Java 语言编写简单的应用程序。

学习目标

掌握面向对象编程的基本思想。

面向对象编程(Object Oriented Programming，OOP)是一套概念和想法，利用计算机程序来描述实际问题，也是一种更直观、效率更高的解决问题的方法，与面向过程的编程方法(如 C 语言)相对应。面向过程的程序设计方法从解决问题的每一个步骤入手，适合于解决比较小的简单问题。而面向对象的程序设计方法则按照现实世界的特点来管理复杂的事务，把它们抽象为对象(Object)，把每个对象的状态和行为封装在一起，通过对消息的反应来完成一定的任务。

Java 是面向对象的典型编程语言。面向对象编程方法主要解决两方面的问题。

- 程序代码的重复使用，提高共享程度，增加程序的开发速度。
- 降低维护负担，将具备独立性的代码封装起来，在修改部分程序代码时，不会影响程序的其他部分。

3.1 面向对象的思想

从现实世界中客观存在的事物(即对象)出发来构造软件系统，并在系统构造中尽可能运用人类的自然思维方式，强调直接以问题域(现实世界)中的事物为中心来思考问题，认识问题，并根据这些事物的本质特点，把它们抽象地表示为系统中的对象，作为系统的基本构成单位(而不是用一些与现实世界中的事物不太相关，并且没有对应关系的其他概念来构造系统)，可以使系统直接映射成问题域，保持问题域中事物及其相互关系的本来面貌。

3.1.1 面向对象思想的基本概念

面向对象(Object Oriented)是当今软件开发的主流方法，其概念和应用已超越了程序设计和软件开发，扩展到很宽的范围。如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。

面向对象程序设计语言必须有描述对象及其相互之间关系的语言成分。这些成分的关系是：系统中一切皆为对象；对象是属性及其操作的封装体；对象可按其性质划分为类，对象为类的实例；实例关系和继承关系是对象之间的静态关系；消息传递是对象之间动态联系的唯一形式，也是计算的唯一形式；方法是消息的序列。主要概念包括：

(1) 对象：对象是人们要进行研究的任何事物，从最简单的整数到复杂的飞机等均可看作对象，它不仅能表示具体的事物，还能表示抽象的规则、计划或事件。

(2) 对象的状态和行为：对象具有状态，一个对象用数据值来描述它的状态。对象还有操作，用于改变对象的状态，操作就是对象的行为。对象实现了数据和操作的结合，使数据和操作封装于对象的统一体中。

(3) 类：具有相同或相似性质的对象的抽象就是类。因此，对象的抽象是类，类的具体化就是对象，也可以说类的实例是对象。类具有属性，是对象状态的抽象化，用数据结构来描述类的属性。类是对象行为的抽象化，用操作名和实现该操作的方法来描述。

(4) 类的结构：在客观世界中有若干类，这些类之间有一定的结构关系。通常有两种主要的结构关系，即一般与具体、整体与部分的结构关系。一般与具体结构称为分类结构，也可以说是“或”关系，或者是“is a”关系。整体与部分结构称为组装结构，它们之间的关系是一种“与”关系，或者是“has a”关系。类中操作的实现过程叫作方法，一个方法有方法名、参数、方法体。

(5) 消息和方法：对象之间进行通信的结构叫作消息。在对象的操作中，当一个消息发送给某个对象时，消息包含接收对象去执行某种操作的信息。发送一条消息至少要包括说明接收消息的对象名、发送给该对象的消息名(即对象名、方法名)。一般还要对参数加以说明，参数可以是认识该消息的对象所知道的变量名，或者是所有对象都知道的全局变量名。

3.1.2 面向对象思想的基本特征

(1) 对象的唯一性：每个对象都有唯一的标识，通过这种标识，可找到相应的对象。在对象的整个生命期中，它的标识都不改变，不同的对象不能有相同的标识。

(2) 分类性：指将具有一致的数据结构(属性)和行为(操作)的对象抽象成类。一个类就是这样一种抽象，它反映了与应用有关的重要性质，而忽略其他一些无关内容。任何类的划分都是主观的，但必须与具体的应用有关。

(3) 继承性：使子类自动共享父类数据结构和方法的机制，这是类之间的一种关系。在定义和实现一个类的时候，可以在一个已经存在的类的基础之上来进行，把这个已经存在的类所定义的内容作为自己的内容，并加入若干新的内容。

继承性是面向对象程序设计语言不同于其他语言的最重要的特点，是其他语言所没有的。在类层次中，子类只继承一个父类的数据结构和方法，称为单重继承；子类继承了多个父类的数据结构和方法，称为多重继承。

在软件开发中，类的继承性使所建立的软件具有开放性、可扩充性，它简化了对象、类的创建工作量。采用继承性，提供了类的规范的等级结构。通过类的继承关系，使公共的特性能够共享，提高了软件的重用性。

(4) 多态性：指相同的操作或函数、过程可作用于多种类型的对象上并获得不同的结

果，不同的对象收到同一消息可以产生不同的结果。多态性允许每个对象以适合自身的方式去响应共同的消息，增强了软件的灵活性和重用性。

3.1.3　面向对象思想的基本要素

(1) 抽象：抽象是指强调实体的本质、内在的属性。在系统开发中，抽象指的是在决定如何实现对象之前的对象的意义和行为。使用抽象可以尽可能避免过早考虑一些细节，类实现了对象的数据(即状态)和行为的抽象。

(2) 封装性(信息隐藏)：封装性是保证软件部件具有优良的模块性的基础。面向对象的类是封装良好的模块，类定义将其说明(用户可见的外部接口)与实现(用户不可见的内部实现)显式地分开，其内部实现按其作用域提供保护。

对象是封装的最基本单位。封装防止了程序相互依赖性而带来的变动影响。面向对象的封装比传统语言的封装更为清晰。

(3) 共享性：面向对象技术在不同级别上促进了共享，同一类中的对象有着相同数据结构，这些对象之间是结构、行为特征的共享关系。在同一应用的类层次结构中，存在数据结构和行为的继承，使各相似子类共享共同的结构和行为，使用继承来实现代码的共享，这也是面向对象的主要优点之一。面向对象不仅允许在同一应用中共享信息，而且为未来目标的可重用设计准备了条件，通过类库这种机制和结构来实现不同应用中的信息共享。

3.2　面向对象的编程方法

面向对象编程(Object Oriented Programming，OOP)方法是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法，是建立在“对象”概念基础上的方法学。对象是由数据和允许的操作所组成的封装体，与客观实体有直接对应关系，一个对象类定义了具有相似性质的一组对象。而继承性是对具有层次关系的类的属性和操作进行共享的一种方式。所谓面向对象就是基于对象概念，以对象为中心、以类和继承为构造机制，来认识、理解、刻画客观世界和设计，构建相应的软件系统。

3.2.1　面向对象编程的基本步骤

面向对象编程通常要经过 9 个步骤：

(1) 分析确定在问题空间和解空间出现的全部对象及其属性。

(2) 确定施加于每个对象的操作，即对象固有的处理能力。

(3) 分析对象间的联系，确定对象彼此间传递的消息。

(4) 设计对象的消息模式，消息模式和处理能力共同构成对象的外部特性。

(5) 分析各个对象的外部特性，将具有相同外部特性的对象归为一类，从而确定所需要的类。

(6) 确定类间的继承关系，将各对象的公共性质放在较上层的类中描述，通过继承来共享对公共性质的描述。

(7) 设计每个类关于对象外部特性的描述。

(8) 设计每个类的内部实现(数据结构和方法)。

(9) 创建所需的对象(类的实例),实现对象间的联系(发送消息)。

3.2.2 主要概念解析

1. 对象、类和消息

对象就是变量和相关方法的集合,其中变量表明对象的状态,方法表示对象所具有的行为,一个对象的变量构成这个对象的核心,包围在它外面的方法使这个对象和其他对象分离开来。例如,可以把汽车抽象为一个对象,用变量来表示它当前的状态,如速度、油量、型号、所处的位置等,它的行为则可以有加速、刹车、换挡等。操纵汽车时,不用去考虑汽车内部各个零件如何运作的细节,而只需根据汽车可能的行为使用相应的方法即可。实际上,面向对象的程序设计实现了对对象的封装,使用者不必关心对象的行为是如何实现的这样一些细节。通过对对象的封装,实现了模块化和信息隐藏,有利于程序的可移植性和安全性,也有利于对复杂对象的管理。

对象之间必须要进行交互来实现复杂的行为。例如,要使汽车加速,必须发给它一个消息,告诉它进行何种动作(这里是加速)以及实现这种动作所需的参数(这里是需要达到的速度等)。一个消息包含三方面的内容:消息的接收者、接收对象应采用的方法和方法所需要的参数。接收消息的对象在执行相应的方法后,可能会给发送消息的对象返回一些信息,例如上例中汽车的仪表上会出现已经达到的速度等。

由于任何一个对象的所有行为都可以用方法来描述,通过消息机制可以实现对象之间的交互,同时,处于不同处理过程甚至不同主机的对象间也可以通过消息实现交互。上面所说的对象是一个具体的事物,例如每辆汽车都是一个不同的对象。但是多个对象常常具有一些共性,例如所有的汽车都有轮子、方向盘、刹车装置等。于是可以抽象出对象的共性,这就是类(Class)。典型的类是“人类”,表明人的共同性质。类中定义一类对象共有的变量和方法。把一个类实例化即生成该类的一个对象。例如可以定义一个汽车类来描述所有汽车的共性,通过类的定义可以实现代码的复用。我们不用去描述每一个对象(某辆汽车)而是通过创建类(如汽车类)的一个实例来创建该类的一个对象,这样大大减化了软件的设计量。

类是对一组具有相同特征的对象的抽象描述,所有这些对象都是这个类的实例。在程序设计语言中,类是一种数据类型,而对象是该类型的变量,变量名即是某个具体对象的标识名,即对象名。

2. 继承

通过对象、类,可以实现封装,通过子类则可以实现继承。

公共汽车、出租车、货车等都是汽车,但它们是不同的汽车,除了具有汽车的共性外,还具有自己的特点,如不同的操作方法,不同的用途等。这时可以把它们作为汽车的子类来实现,它们继承父类(汽车)的所有状态和行为,同时增加自己的状态和行为。通过父类和子类实现了类的层次,可以从最一般的类开始,逐步特殊化定义一系列的子类。同时,通过继承也实现了代码的复用,使程序的复杂性线性地增长,而不是呈几何级数增长。

Java 则只支持单一继承,降低了继承的复杂度。通过接口也能实现多重继承,接口的

概念更简单，使用接口编程更方便。

3. 抽象与接口

虽然继承别人已写好的代码的功能，使程序代码能重复使用，不过，若修改了基础类，继承基础类的扩展类是否还能正常运行呢？如果基础类是自己开发的，要修改很简单，但若基础类是别人做好了的，该如何处理呢？这就引出了抽象(abstract)的概念。抽象概念的生成是为了要降低程序版本更新后在维护方面的负担，使功能的提供者和功能的用户能够彼此分开，各自独立，互不影响。

为了达到抽象的目的，需要在功能提供者与功能使用者之间提供一个共同的规范，功能提供者与功能使用者都要按照这个规范来提供、使用这些功能。这个共用的规范就是接口(interface)，接口定义了功能数量、函数名称、函数参数、参数顺序等。它是一个能声明属性、事件和方法的编程结构，只提供定义，并不实现这些成员，留给用户自己扩充。接口定义了功能提供者与功能使用者之间的准则，因此只要接口不变，功能提供者就可以任意更改实现的程序代码，而不影响使用者。接口就好比两个以上的体系拟定的共同规范，如调用Windows API一样。

4. 多态

一个类中可以包含多个方法，是不是允许有同名呢？答案是允许。一方面，多个方法可以同名，但参数不能完全相同，否则系统无法识别。另一方面，如果在父类和子类中都有同一个方法名，也是允许的，方法体可以相同也可以不同。这就是多态，Java通过方法重载和方法覆盖来实现多态。

通过方法重载，一个类中可以有多个具有相同名字的方法，由参数来区分哪一个方法，包括参数的个数、参数的类型和参数的顺序。例如，对于一个绘图的类Graphics，它有一个draw()方法用来画图或输出文字，可以传递给它一个字符串、一个矩形、一个圆形，甚至还可以再指明绘图的初始位置、图形的颜色等，对于每一种实现，只需实现一个新的draw()方法即可，而不需要新起一个名字，简化了方法的实现和调用，程序员和用户不需要记住很多的方法名，只需要设置相应的参数即可。

通过方法覆盖，子类可以重新实现父类的某些方法，使其具有自己的特征。例如对于车类的加速方法，其子类(如赛车)中可能增加了一些新的部件来改善、提高加速性能，这时可以在赛车类中覆盖父类的加速方法。覆盖隐藏了父类的方法，使子类拥有自己的具体实现，更进一步表明了与父类相比，子类所具有的特殊性。

3.2.3　类的实现

类是组成Java程序的基本要素，封装了一类对象的状态和方法，是这一类对象的原型。在前面的例子中已经定义了一些简单的类，看下面的HelloWorldApp类：

```
public class HelloWorldApp{
  public static void main(String args[]){
    System.out.println("Hello World !");
  }
}
```

可以看出,一个类包含类声明和类体两部分内容:

```
类声明 {
  类体
}
```

1. 类的声明

一个最简单的类声明如下:

```
class 类名 {
  …
}
```

在类声明中还可以包含类的父类,类所实现的接口以及修饰符。这此内容将后面介绍。

2. 类体

类体中定义了该类所有的属性(也称为变量)和该类所支持的方法(也称为函数)。通常属性在方法前定义(但不强制),如:

```
class 类名 {
  属性声明;
  方法声明;
}
```

下例定义了一个 Point 类,并且声明了它的两个变量 x、y 坐标,同时通过 init()方法实现对 x、y 赋初值。

```
class Point {
  int x,y;
  void init(int m, int n){
    x = m;
    y = n;
  }
}
```

3. 属性

最简单的属性声明格式为:

```
类型 属性名;
```

属性的类型可以是 Java 中的任意数据类型,包括简单类型、数组、类和接口。在一个类中,属性必须是唯一的,但是属性名可以和方法名相同,例如:

```
class Point{
  int x,y;
  int x(){
    return x;
  }
}
```

其中，方法 x()和属性 x 具有相同的名字，最好是不要同名。

类的属性和在方法中所声明的局部变量是不同的，属性的作用域是整个类，而局部变量的作用域只是方法内部。对一个属性，也可以限定它的访问权限，用 static 限定它为静态变量，或者用 final 修饰符限定。

final 表示最终的，用 final 声明的变量就是最终变量，最终变量就是常量，因而它用来声明一个常量，例如：

```
class FinalVar{
  final float PAI = 3.14;
  …
}
```

例中声明了常量 PAI，并赋值为 3.14，程序中任何时候用到 PAI，均是 3.14，其值不能再变化。常量名通常用全部大写字母表示。

特别提示：一个简单的程序，可以依赖程序员的经验直接写出来，而功能强大的软件，其代码量可能达到数万条，由专门的软件设计团队，经过需求分析和整体设计，各程序员分工合作共同完成。程序员要养成良好的编程习惯，遵守行业标准和流程，并按照软件工程规范进行操作。

本章习题

1. 简答题

(1) 面向对象思想有哪些基本特征？
(2) 面向对象思想包括哪些基本要素？
(3) 面向对象编程需要哪些步骤？
(4) 什么是类？类由哪些成分构成？
(5) 解释以下概念：类、对象、继承、封装、抽象。

2. 操作题

(1) 定义一个类 Person，并设置若干成员变量和成员方法。
(2) 定义一个类 Teacher，并设置若干成员变量和成员方法，分析 Teacher 和 Person 的关系。

项目实战 1　分析“仿 QQ 聊天软件”项目

一、目的

（1）掌握面向对象的基本概念；
（2）掌握面向对象的分析与设计方法；
（3）培养良好的编码习惯和编程风格。

二、内容

（一）任务描述

在日常生活中人们经常使用 QQ 软件与亲朋好友聊天，QQ 软件的出现极大地丰富了人们的交流方式。本项目的主要目标是模仿 QQ 聊天系统进行分析设计。

（二）步骤

1. 需求分析

“仿 QQ 聊天软件”由两大部分组成，一是服务器程序，二是客户端程序。服务器程序的功能包括：
（1）处理用户登录请求，验证登录用户的用户名和密码。
（2）用户登录成功后，创建线程与该用户通信，并将用户登录信息发送给好友。
（3）处理用户退出请求，并将用户退出信息发送给好友。
服务器启动以后，用户才能使用客户端程序登录，客户端程序的功能包括：
（1）提供用户登录界面，将用户输入的用户名和密码发送给服务器进行验证。
（2）与好友聊天。
（3）查看历史聊天记录。

2. 服务器程序设计

（1）服务器界面，启动服务器后进入服务器界面，界面效果如图 P1-1 所示。
（2）启动服务界面，单击服务器界面上的“启动服务”按钮可启动服务器程序，界面效果如图 P1-2 所示。
（3）停止服务，单击服务器界面上的“停止服务”或右上角的×可停止服务。
（4）管理用户账户，用户账户信息保存在 config/user.txt 文件中，文件内容如下：

图 P1-1　服务器界面

图 P1-2　启动服务界面

```
2020001,001,张三
2020002,002,李四
2020003,003,王五
```

文件内容说明如下：

用户账号：2020001,密码：001,昵称：张三；

（5）管理好友信息，用户好友信息保存在 config/friend.txt 文件中，文件内容如下：

```
2020001,李四,2020002
2020001,王五,2020003
2020002,张三,2020001
2020002,王五,2020003
2020003,张三,2020001
2020003,李四,2020002
```

其中，2020001 为用户账号，李四为好友昵称，2020002 为好友账号，账号 2020002 是账号 2020001 的好友，这里需要双方互加好友才行。

（6）用户登录后，服务器界面中将显示用户登录信息，如图 P1-3 所示为服务器端显示用户成功登录界面。

（7）用户退出后，服务器界面中将显示用户退出信息，如图 P1-4 所示为服务器端显示用户退出登录界面。

图 P1-3　服务器端显示用户成功登录界面

图 P1-4　服务器端显示用户退出登录界面

3. 客户端程序设计

(1) 登录界面,运行客户端程序后进入登录界面,效果如图 P1-5 所示。

(2) 主界面,用户输入用户名 2020001、密码 001,单击“登录”按钮,进入主界面,效果如图 P1-6 所示。

图 P1-5 登录界面

图 P1-6 主界面

(3) 好友列表界面,双击主界面中向右的箭头,展开好友列表,界面效果如图 P1-7 所示。当前没有好友在线,再次运行客户端程序,输入账号 2020002、密码 002,单击“登录”按钮,展开好友列表,可以看到好友“张三 202001”头像已经变亮,说明好友已经登录了,界面效果如图 P1-8 所示。

图 P1-7 好友列表界面(没有好友登录)

图 P1-8 好友列表界面(有好友登录)

(4) 聊天界面,在主界面中,用户“李四”双击已经登录的好友“张三”进入聊天界面,界面效果如图 P1-9 所示,用户在聊天界面中输入聊天信息,然后单击“发送”按钮,可将聊天信息发送给好友“张三”,如图 P1-10 所示,“张三”在自己的主界面中双击好友“李四”,打开与李四的聊天界面,可查看好友“张三”发送过来的信息,如图 P1-11 所示。

(5) 消息记录界面,在聊天界面中单击消息记录按钮,可以查看聊天记录,如图 P1-12 所示。

(6) 重启聊天界面,用户“张三”与好友“李四”聊天结束,退出登录后,再次登录,用户“张三”再次打开与好友“李四”的聊天窗口,窗口中可显示上次的聊天信息,如图 P1-13 所示。

图 P1-9　聊天界面

图 P1-10　发送聊天信息界面

图 P1-11　查看聊天信息界面

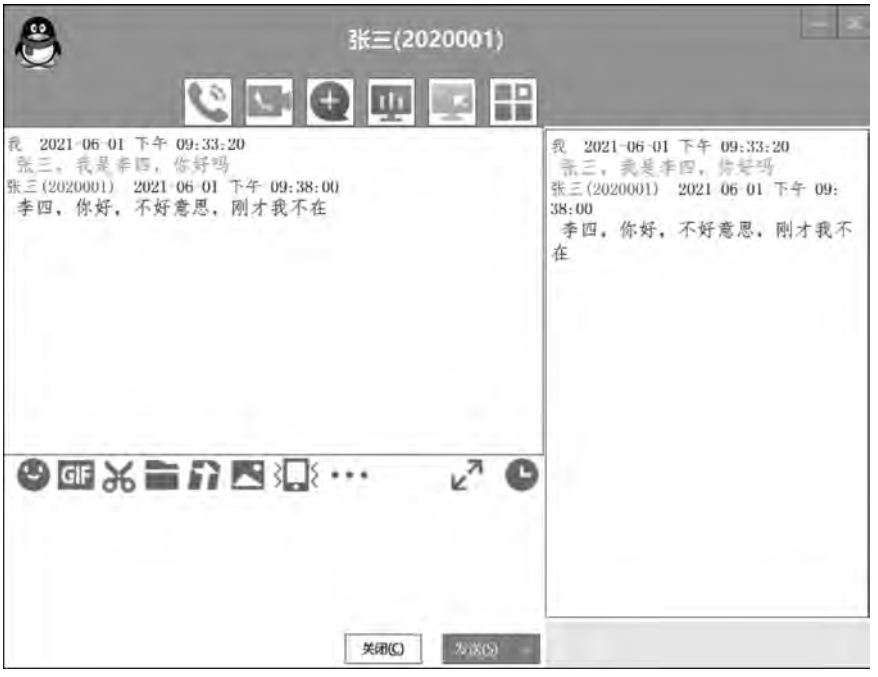

图 P1-12　消息记录界面

图 P1-13　重启聊天界面