# 面向对象

## 面向对象与面向过程：

谈到面向对象就不得不提到面向过程，面向过程注重过程，比如老张开车去东北，是有顺序的，一步一步进行的，有了老张才开车，开了车才去东北。而面向对象是用对象的思维，把老张，车，东北看做三个对象，这样就不必过于注重过程。

两者比较之下，面向对象有两个优点：

1. 可以对对象进行替换，老张可以是老周，车可以是船等，提高了代码的切换灵活性。
2. 对于团队开发而言，可以使得多人进行同时开发，不必受到过程的前后顺序约束

## 面向对象的三大特征：封装，继承，多态

### 封装

1. 从数据角度，将一些基础变量复合为自定义类，这些数据不仅可以准确的描述事物，而且可以体现事物的行为。比如学生，具有学号，班级等
2. 从行为角度，调用者不必关心方法的内部代码细节，而是使用其功能。比如开拖拉机，不必关心拖拉机内部运作原理，只要能用就行
3. 从设计角度
4. 分而治之：

将一个大的需求分为许多类，比如学生管理系统，可以分为学生模型，学生管理器，学生显示器三个类

1. 封装变化：

将会产生需求变化的类进行单独封装，避免牵一发而动全身

1. 高内聚：

一个类内的各个方法都只完成一个任务

1. 低耦合：

降低类与类之间的关联性和耦合度，争取做到一个类完成一个功能

### 继承

1. 定义：继承是重用类的功能和概念 功能：子类具有父类的成员和方法，并具有自己的功能 概念：事物之间是有层次和逻辑性的，举例如火车，汽车，船，层次化抽象出交通运输工具这个概念，在需要统一管理的情况下，交通运输工具即可成为火车等的父类
2. 适用性：多个类具有统一概念，并且需要统一进行管理
3. 缺点：继承的耦合度高 切换不灵活
4. 优点：代码复用 层次化管理类

### 三：多态

1. 定义：多态是父类中的一个方法在子类中有不同的表现 比如：奔驰车宝马车都是车，继承了加油站这个父类，都具有加油这个方法，但是两个车的加油的油量不同
2. 优点：体现了子类的个性化 增强可扩展性（体现了开闭原则）

设计原则

### 六大设计原则

1. 开闭原则：对扩展开放，对修改关闭，尽量做到增加功能，内部不修改原先代码
2. 类的单一职责：一个类有且只有一个改变的原因 外界需求变化，内部对应一个类的变化
3. 依赖倒置原则：客户端代码尽量依赖抽象的组件 尽量去找父类
4. 组合复用原则：代码的复用尽量使用组合关系，而非继承 可以降低代码的耦合度
5. 里氏替换原则：（重写 子类具有父类的同名方法）父类出现的地方，子类可以进行替换，并实现相应功能
6. 迪米特法则：类与类的交换过程中，在满足基本的功能时，尽量减少传输的数据量 比如需要返回学生的学号，尽量不要返回班级，任课老师，所选课程等