# 面向对象三大特征

## 一、三大特征：（根据具体需求具体分析）

### 封装：

三类：数据的封装，行为的封装，属性的封装？

根据需求分解成多个类。

对外提供必要的功能，而不必考虑其执行细节。

（例如标签）

把同一事物进行操作的方式和相关的方法及数据放在同一个类中；

优势：内部可实现自由修改，具有清晰的对外接口。

规则：分而治之（例如MVC）：确定类的功能与职责，每个类只负责单一一件事。

封装变化（张三开车回家）：找到变化点进行封装。

高内聚： 功能的内聚，单一职责。

低耦合： 模块与模块间尽可能使其独立存在，尽可能使用关联和依赖。

体现的是个性。

### 继承：

主要分离变化。

重用现有类的概念和功能。

体现了依赖倒置原则。

多个父类情况下继承遵循C3算法

可以让某个类型的对象获取另一个对象的属性或者数据的方法。

优势：使用现有类的功能，在无需编写原有类的情况下对这些功能进行扩展，提高了可复用性及可扩展性。

是多态的前提。

缺点：降低了代码的灵活度。子类必须拥有父类属性及方法，让子类多了约束，提高了代码耦合度。父类变量修改要考虑子类。

体现的是共性。

### 多态：

在继承的基础上，同一操作对于不同的对象产生不同效果。

分动态多态和静态多态：

静态多态体现在对方法的重载

动态多态体现在对方法的重写

一个类的实例相同方法在不同子类有不同的表现形式；

优势：增强了代码的扩展性及维护性。

面向对象优点：高维护（结构清晰）、高拓展（开闭原则），高复用（代码简洁）

## 开闭原则（目标）：

（补丁）

（员工换岗：）对扩展开放，对修改关闭；

增加新功能，不修改原代码（重写、装饰器的体现）

**依赖倒置**：

（人打电话依赖通讯工具）调用者和被调用者都依赖于抽象的对象，两者没有直接关联，降低两者耦合度

**单一原则**

（活字印刷）（积木）：一个类实现一个功能; 对外代码灵活度高，对内高内聚。

一个类只有一个改变的原因。（换电脑部件）

**迪米特法则（**低耦合一个对象尽可能与其他对象间关联度低；

低耦合（降低变量间的耦合性）；不跟陌生人说话。

**组合复用：**

泛化、关联、依赖：继承、属性、方法

优先选用关联 ，尽量不使用继承。

**里氏替换**：（重写，函数传参）抽象和多态将设计中的静态结构改为动态结构