面向对象三大特征：

**封装：**

将数据和方法包装进类中，加上具体实现的隐藏。

增强安全性，隐藏对象的实现细节：只需要知道类实现后的结果，不必关心如何实现。

分而治之，封装变化：让每个类处理一个独立的功能。当需求发生变化时，只需要修改这个类,而不用影响外面的类。

**继承：**

继承为子类具有父类的特征（属性和方法）。并在此基础上进行扩展。

父类拥有所有子类所共有的方法。

优点：提高代码重用性，增强了代码的扩展性。

缺点：耦合度过高，牵一发而动全身。

**多态：**

多态就是父类作用于不同的子类时，一些属性与方法是父类没有，而子类所独有的，从而导致了执行时的结果各不相同。

优点：增强程序扩展性，体现开闭原则。

子类实现了父类中相同(名称、参数)的方法，在调用该方法时，实际执行的是子类的方法。

面向对象六大原则：减小耦合

**开闭原则**

对扩展开放，对修改关闭。

在出现问题或修改需求时，尽量去扩展原来的代码，而不要去修改，以免引起更多的问题。

**类的单一职责**

一个类有且只有一个改变的原因。一个类负责一件事情。

优点：

1. 可以降低类的复杂度
2. 提高类的可读性
3. 当发生变化的时候，能将变化的影响降到最小

**依赖倒置**

子类尽量依赖父类，因为父类时抽象的，也是不可变的，子类是细节的，也是不稳定的。

这样可以降低代码的耦合度，提高系统稳定性。

**组合复用原则(少用继承)**

如果仅仅为了代码复用优先选择组合关系，而非继承关系。

组合的耦合度低于继承，灵活度高于继承。

组合复用是将所有类的共同对象封装为一个单独的类，直接调用，修改时，对其他类没有影响。

**里氏替换(重写注意事项)**

所有引用基类的地方必须能够透明的使用其子类的对象。

子类可以去扩展父类的功能，但是不能改变父类原有的功能

由于继承在一定程度上违反了封装的原则，父类可以影响所有的子类。当需求发生变化，修改父类时，其他子类都会有影响，所以提出里式替换，使得父类对子类的影响降到最小。

里式替换一定程度上与开闭原则是相同的。

**迪米特法则(低耦合)**

一个对象应该对其他对象保持最小的了解。

类与类交互时，在满足功能的基础上，传递的数据量越少越好。

尽量让一个对象只与其接触最为直接的对象通信。