**对面向对象OOP的理解**

所谓对象就是**由一组数据结构和处理它们的方法组成的**，划重点 “数据” 包括对象的特性、状态等的静态信息； “方法” 也就是行为，包括该对象的对数据的操作、功能等能动信息。把相同行为的对象归纳为类，类是一个抽象的概念，对象是类的具体。简单点说：**对象就是类的实例**。

面向对象的目的：解决软件系统的可扩展性，可维护性和可重用性；

**·面向对象的三大特性：封装、多态和继承**

**（1）封装**（对应可扩展性）：**隐藏对象的属性和实现细节，仅对外公开接口**,控制在程序中属性的读和修改的访问级别。封装是**通过访问控制符（public protected private）来实现**。一个类就可看成一个封装。

**pay1:封装好处**

A.避免使用非法数据赋值

B.保证数据的完整性

C.避免类内部发生修改的时候，导致整个程序的修改

**（2）继承**（重用性和扩展性）：子类继承父类，**可以继承父类的方法和属性**。可以对父类方向进行覆盖（实现了多态）。但是继承破坏了封装，因为他是对子类开放的，修改父类会导致所有子类的改变，因此继承一定程度上又破坏了系统的可扩展性，所以继承需要慎用，只有明确的IS-A关系才能使用。所以继承要慎用，尽量优先使用组合。

**pay2:继承好处**

A.继承模拟了现实世界的关系，OOP中强调**一切皆对象**，这符合我们面向对象的编程思考方向

B.继承实现了**代码的复用**，这在例子中我们已经有所体会，合理地使用继承，会使我们的**代码更加简洁**

C.继承使得**程序结构清晰**，子类和父类的层次结构更清晰，最终的目的是是子类只关注子类的相关行为和状态，无须关注父类的行为和状态.例如，学员只需要管理学号、爱好这种属性，而公共的姓名、年龄、性别属性交给父类管理

**（3）多态**（可维护性和可扩展性）：**接口的不同实现方式即为多态**。接口是**对行为的抽象**，刚才在封装提到，找到变化部分并封装起来，但是封装起来后，怎么适应接下来的变化？这正是接口的作用，接口的主要目的是**为不相关的类提供通用的处理服务**,我们可以想象一下。比如鸟会飞，但是超人也会飞，通过飞这个接口，我们可以让鸟和超人，都实现这个接口。

**pay3:多态**

A.相同的两个或者是**多个属于不同类的对象，对于同一个消息(方法的调用)做出不同响应**的行为方式

**pay4:接口的好处**

A.接口是对继承单根性的扩展：接口可以说是从某种程度上实现了**多重继承**的功能。对子类来说，一个类虽然不能实现多重继承，但是一个类可以实现多个接口。通过实现多个接口可以说是变相的实现了类的多重继承。

B.接口是一种**规范和标准**：一个类如果是实现了一个接口，意味着这个类遵循了某种规范和约定。其它类可以通过这些规范和约定与她通信

C.接口**屏蔽了实现的细节**：接口可以让接口的使用者不必关心接口的实现者是如何实现USB接口标准，就可以正常使用它了。同样在程序中一个类 如果实现了IComparable接口，便可以让这个类的对象进行排序。

D.接口的使用**方便团队的协助开发**：例如一个系统，加入要实现对象的排序功能，实现接口的专门实现接口，而做排序功能的专门做排序，如果一个系统都能够做到这种设计，便可以更为方便的使用开发团队进行的分工明确的协作开发！

pay5:序列化和反射

A.我们经常需要将对象的字段值保存到磁盘中，并在以后检索此数据。尽管不使用序列化也能完成这项工作，但这种方法用代码行来衡量开发进度，无异于用重量来衡量制造飞机的进度。