Java核心技术

**面向对象的特点**

可以分为4点，封装、继承、多态、封装。

先说抽象，抽象就是忽略一个主题中与当前业务无关的那些方面，把我们需要的特征抽象出来。以便更加专注于我们的业务。比如一个学生，我们业务只需要查看学生的成绩、班级、姓名。那么这个学生的身高、体重这些就不需要我们去关注。

然后是封装，封装就是把我们抽象出来的数据进行包装，对外提供接口，隐藏内部细节。也可以决定数据的对外的可见性。从而可以降低外界对程序的修改后的影响。也是增强了程序的健壮性、安全性。

然后是继承，继承就是一个新类可以从现有的类中派生的过程就是类的继承。这个新类被称之为子类，而原始的类被称为父类。子类包含了父类共用的特征和行为。并且子类可以对父类的行为进行覆盖。通过继承让我们代码能得到很好的重用。就好比一个图形类，这个图形类有个画图的方法，通过这个图形类派生出一个三角形类，这个三角形类可以覆盖父类的画图方法，画出一个三角形。你想要个圆形，也可用继承这个图形类，然后覆盖画图方法就可以实现不同的行为。这样也可以看出，通过继承可以让我们的代码更加的灵活。如果有用户需求需要改变，不需要去改原来的代码，只需要继承父类重写方法就可以实现新的业务。

最后就是多态，多态是指不同的类做出不同的行为，就像刚才的那个图形例子一样。不同的图形有不同的行为，而多态就是程序在运行时，做的一个动态的调用。通过这个动态调用，让我们在程序运行期间让程序展示不同的行为，提高了程序的灵活性。

**重载和重写的区别**

重载是指同一个类中的多个方法具有相同的名字,但这些方法具有不同的参数列表。

重写是存在子父类之间的,子类定义的方法与父类中的方法具有相同的方法名字,相同的参数表和相同的返回类型 。

重载在本类中完成，重写在子类中完成。

**抽象类和接口的区别**

可以从“语法”和“设计”两个层面来讲

从语法上来说呢：

1. 抽象类可以包含普通方法，而接口只能抽象方法
2. 抽象类可以有变量和常量，而接口只能有常量
3. 抽象类能包含静态方法、静态代码块，而接口不行。
4. 抽象类是通过继承的，而接口是用来实现的。
5. 抽象类有构造器，接口中没有构造器
6. 。。。。。。。。

这是从语法上有这些区别，从抽象类和接口的设计上来说，抽象类更像是对一种事物的抽象，而接口则是一种行为抽象，或者说是一种标准。接口就好比我们的国标，它规定一个插排的尺寸。那么各各厂商必须根据这个规定去制造。那么国标就是个接口，各各插排的厂商都实现了这个接口。而抽象类就好比是一只鸟，鸟是个模糊的概念，世界上有很多鸟，不同的鸟有不同的行为。这个鸟就是一个抽象类，像麻雀、八哥这些具体的鸟就是这个抽象类的具体实现。

如果需要存储一些数据或存在一些默认的行为，那么就可以选择使用抽象类，反之就可以选择接口，个人更喜欢用接口一点，因为java中只能继承一个类，但可以实现多个接口，更灵活。

**String和StringBuffer的区别？StringBuffer和StringBuilder区别？**

　String是不可变的，对String类的任何改变都会返回一个新的String 对象。

　　StringBuffer是可变的，对StringBuffer中的内容修改都是当前这个对象。

　　String重写了equals方法和hashCode方法，StringBuffer没有重写equals方 法。

String是final的类。StringBuffer不是。

String创建的字符串是在常量池中，创建的变量初始化一次，如果再对该字符串改变会产生新的字符串地址值，StringBuffer是在堆中创建对象，当对字符串改变时不会产生新的字符串地址值，如果对字符串进行频繁修改的话建议使用StringBuffer，以节省内存。

StringBuffer和StringBuilder，StringBuffer是线程安全的，StringBulider是线程不安全的。当不考虑并发问题时候，请使用StringBulider。

**HashMap和HashTable的区别**

主要是两方面，1：HashMap是线程不安全的，而HashTable是线程安全的。2：HashMap和HashTable都实现了Map接口，但HashTable的key不能存在null值，HashMap可以存在一个null值。

**ArrayList和LinkedList的区别**

主要是两方面的区别。1：ArrayList底层是数组的形式来存储数据，而LinkedList是以链表的形式来存储。2：ArrayList

是数组实现的，所以在中间插入数据或删除的时候效率比较低，而LinkedList则相反，数据添加删除效率高。但是数据的遍历效率不如ArrayList。

**ArrayList和Vector的区别**

主要两大区别：

1、Vector是线程安全的，用synchronized实现线程安全，而ArrayList是线程不安全的。如果不需要考虑线程安全，则可以使用ArrayList。

2、当存储的元素的个数超过了容量时，就会自动扩充容量，Vector增长原来的一倍，ArrayList增加原来的0.5倍。

总结：

1. 回答任何问题都要给出解释。为什么是这个答案。
2. 问你一个东西的理解时，首先就说这东西是什么？干什么用的？优缺点？
3. 说话不要急，自然点，不要像背书一样，适当加一些自己的语气词。