1. 面向对象
2. 面向对象三大特性

**（1）封装性**

封装是面向对象的核心思想，将对象的属性和行为封装起来，不需要让外界知道具体实现细节，这就是封装思想。例如，用户使用电脑，只需要使用手指敲键盘就可以了，无需知道电脑内部是如何工作的，即使用户可能碰巧知道电脑的工作原理，但在使用时，并不完全依赖电脑工作原理这些细节。

**（2）继承性**

继承性主要描述的是类与类之间的关系，通过继承，可以在无需重新编写原有类的情况下，对原有类的功能进行扩展。例如，有一个汽车的类，该类中描述了汽车的普通特性和功能，而轿车的类中不仅应该包含汽车的特性和功能，还应该增加轿车特有的功能，这时，可以让轿车类继承汽车类，在轿车类中单独添加轿车特性的方法就可以了。继承不仅增强了代码的复用性、提高开发效率，还为程序的维护补充提供了便利。

**（3）多态性**

多态性指的是在程序中允许出现重名现象，它指在一个类中定义的属性和方法被其它类继承后，它们可以具有不同的数据类型或表现出不同的行为，这使得同一个属性和方法在不同的类中具有不同的语义。例如，当听到“Cut” 这个单词时，理发师的行为是剪发，演员的行为表现是停止表演，不同的对象，所表现的行为是不一样的。

1. 垃圾回收机制

在Java中，当一个对象成为垃圾后仍会占用内存空间，时间一长，就会导致内存空间的不足。针对这种情况，Java中引入了垃圾回收机制。有了这种机制，程序员不需要过多关心垃圾对象回收的问题，Java虚拟机会自动回收垃圾对象所占用的内存空间。

一个对象在成为垃圾后会暂时地保留在内存中，当这样的垃圾堆积到一定程度时，Java虚拟机就会启动垃圾回收器将这些垃圾对象从内存中释放，从而使程序获得更多可用的内存空间。除了等待Java虚拟机进行自动垃圾回收外，还可以通过调用System.gc()方法来通知Java虚拟机立即进行垃圾回收。当一个对象在内存中被释放时，它的finalize()方法会被自动调用，因此可以在类中通过定义finalize()方法来观察对象何时被释放。

1. project、public、private的区别

1、public：public表明该数据成员、成员函数是对所有用户开放的，所有用户都可以直接进行调用。

2、private：private表示私有，私有的意思就是除了class自己之外，任何人都不可以直接使用。

3、protected：protected对于子女、朋友来说，就是public的，可以自由使用，没有任何限制，而对于其他的外部class，protected就变成private

1. 类的初始化过程

Student s = new Student();在内存中做了哪些事情?

（1）加载Student.class文件进内存

（2）在栈内存为s开辟空间

（3）在堆内存为学生对象开辟空间

（4）对学生对象的成员变量进行默认初始化

（5）对学生对象的成员变量进行显示初始化

（6）通过构造方法对学生对象的成员变量赋值

（7）学生对象初始化完毕，把对象地址赋值给s变量

1. 什么时抽象类？

在面向对象的概念中，所有的对象都是通过类来描绘的，但是反过来，并不是所有的类都是用来描绘对象的，如果一个类中没有包含足够的信息来描绘一个具体的对象，这样的类就是抽象类。

抽象类除了不能实例化对象之外，类的其它功能依然存在，成员变量、成员方法和构造方法的访问方式和普通类一样。

由于抽象类不能实例化对象，所以抽象类必须被继承，才能被使用。也是因为这个原因，通常在设计阶段决定要不要设计抽象类。

父类包含了子类集合的常见的方法，但是由于父类本身是抽象的，所以不能使用这些方法。

在Java中抽象类表示的是一种继承关系，一个类只能继承一个抽象类，而一个类却可以实现多个接口。

1. 什么是多态？

在设计一个方法时，通常希望该方法具备一定的通用性。例如要实现一个动物叫的方法，由于每种动物的叫声是不同的，因此可以在方法中接收一个动物类型的参数，当传入猫类对象时就发出猫类的叫声，传入犬类对象时就发出犬类的叫声。在同一个方法中，这种由于参数类型不同而导致执行效果各异的现象就是多态。继承是多态得以实现的基础。多态性简单说就是一个对象对应着不同类型。

前提：

1. 需要存在继承或者实现关系。
2. 需要有覆盖操作。

好处：提高了代码的扩展性，前期定义的代码可以使用后期的内容。

弊端：前期定义的内容不能使用（调用）后期子类的特有内容。

1. 抽象类和接口

不同点：

抽象类需要被继承，而且只能单继承。接口需要被实现，而且可以多实现。

抽象类中可以定义抽象方法和非抽象方法，子类继承后，可以直接使用非抽象方法。接口中只能定义抽象方法，必须由子类去实现。

抽象类的继承，是is a关系，定义该体系的基本共性内容。接口的实现是like a关系。