Interface

定义：

interface 接口名{}（只能用public修饰）

类实现接口：

格式：

class 类名 impments 接口名{}

接口的实现类：

1. 抽象类：不常用，没有什么意义。
2. 具体类：实现接口中所有的抽象方法
3. 接口实现类的命名规范：类名Impl

注意:

1. 接口是否能实例化？

不能实例化

1. 接口可以包含哪些内容？

成员属性：只可以定义静态常量public static final 修饰

成员方法：抽象方法 public abstract

Jdk1.7(及之前版本)

非抽象方法：（jdk1.8开始）static修饰

调用格式：接口名.方法名();

支持重载，不支持重写，子类中不显示。

构造方法：没有构造方法；接口不能理解为类

1. 接口能否体现多态？----- 可以，最常用的多态的使用形式

面向接口开发：

向上造型：创建接口实现类的对象赋值给一个接口类型的引用。

AnimalTrain a=new Dog();编译看左边，运行看右边，也可以强制转化

1. 类，接口之间的关系

类和类：继承关系，java是单继承，多重继承（传递）

类和接口：实现关系，多实现;

格式：class 类名 implements 接口1 ,接口2.。。。。{}

注意：接口多实现也会产生方法调用的混乱。

接口和接口：继承关系,多继承

1. 抽象类和接口比较？

相同点：抽象类和接口中都可以定义抽象方法；

抽象类和接口都是多态的体现，

抽象类：复用性；

接口：提高了java程序多态的灵活性 扩展性；

抽象类和接口都不能直接被实例化

不同点：

1. 成员：

抽象类可以包含那些内容：

成员属性：变量可以存在 常量可以存在

成员方法：抽象方法可以 非抽象方法可以

构造方法：可以有，也支持构造方法重载；提供给子类使用

接口可以包含哪些内容：

成员属性：只可以定义静态常量public static final 修饰

成员方法：抽象方法 public abstract

Jdk1.7(及之前版本)

非抽象方法：（jdk1.8开始）static修饰

调用格式：接口名.方法名();

支持重载，不支持重写，子类中不显示。

构造方法：没有构造方法；接口不能理解为类

1. 类和接口之间的关系

类和类：继承关系，java是单继承，多重继承（传递）

类和接口：实现关系，多实现;

格式：class 类名 implements 接口1 ,接口2.。。。。{}

注意：接口多实现也会产生方法调用的混乱。

接口和接口：继承关系,多继承

1. 设计思想

抽象类：is a，共性的属性和方法

接口：like a,扩展；模板或者是一种约束

案例：

