

第九章 抽象类和接口



```
abstract class 类名 {
```

```
    常量;
```

```
    变量;
```

```
    构造 ();
```

```
    访问修饰符 abstract 返回类型 方法名(); //抽象方法
```

```
    普通方法;
```

```
    静态方法 ();
```

```
}
```



□ 定义接口

```
访问修饰符 interface 接口名 {  
    常量;  
    抽象方法;  
    默认方法;  
    静态方法;  
    静态内部类型;  
}
```



□ 注意事项：

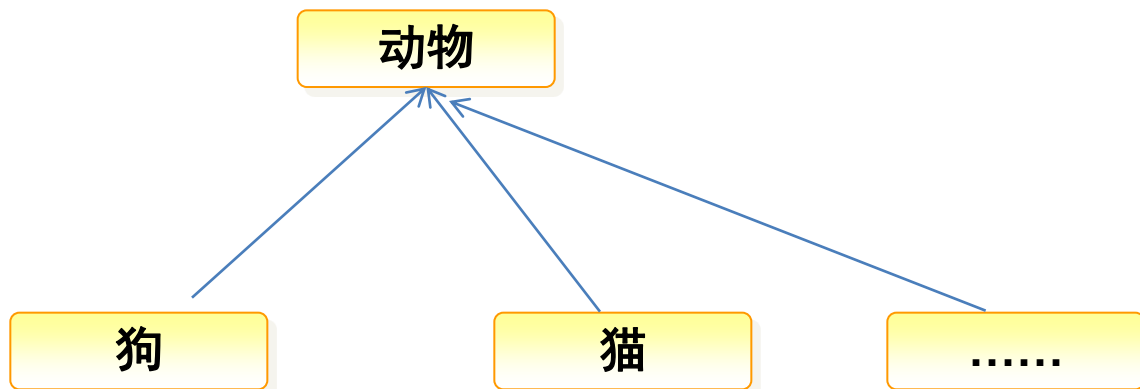
- 静态类不能被继承
- 实现多个接口方法名相同的处理

□ 接口和抽象类的异同

什么是多态



所谓多态，其实就是一种物质，多种存在形态。





□ 程序中的多态

多态：同一个引用类型，使用不同的实例而执行不同操作

```
打印机 打印机1=new 黑白打印机（）；  
打印机 打印机2=new 彩色打印机（）；
```

```
网银接口 网银1=new 京东商城（）；  
网银接口 网银2=new 淘宝商城（）；
```



对象 instanceof 类或接口

- ❑ 该运算符用来判断一个对象是否属于一个类或者实现了一个接口，结果为true或false
- ❑ 在强制类型转换之前通过instanceof运算符检查对象的真实类型，可以避免类型转换异常，从而提高代码健壮性



□ 类型转换的两种形式

➤ 向下类型转换

✓ 父类转成子类

➤ 向上类型转换

✓ 子类转成父类



- 泛化

- 实现

- 依赖

- 关联

 - 聚集

 - ✓ 组合