



## 总结

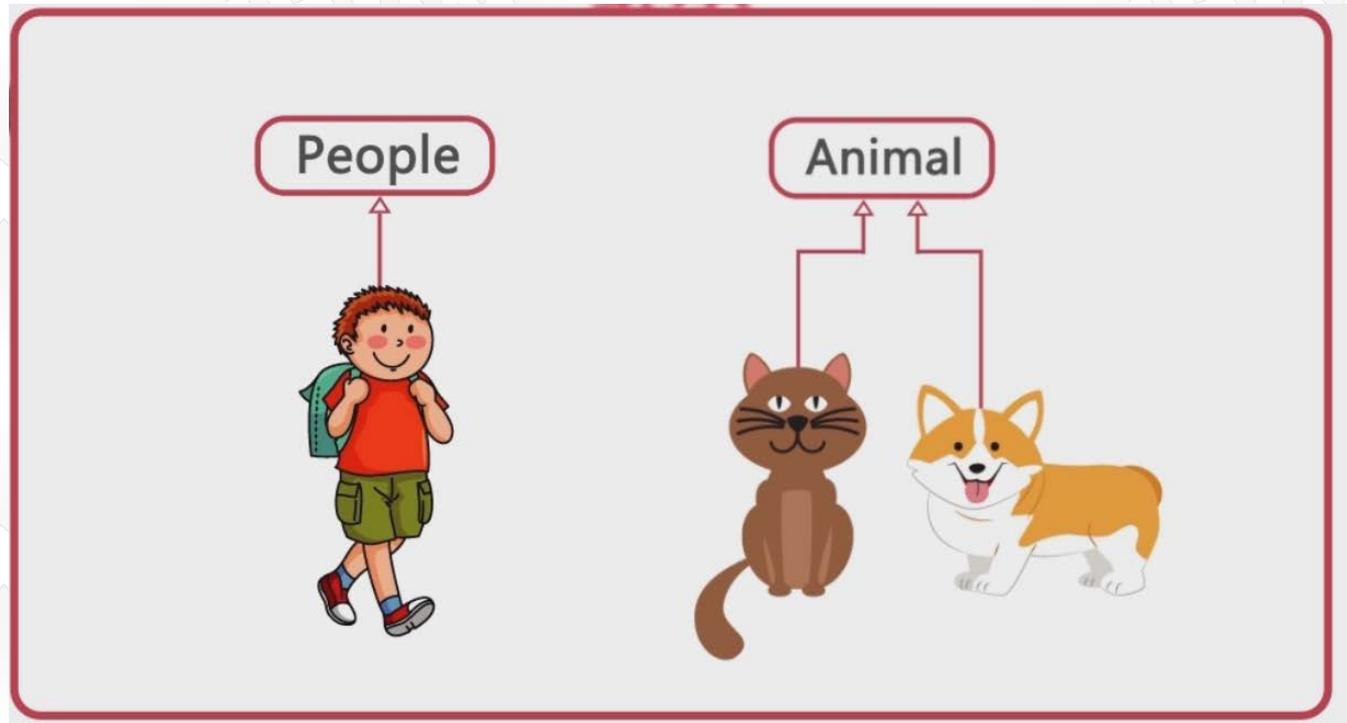


# 继承

概念：

- 1、一种类与类之间的关系
- 2、使用已存在的类的定义作为基础建立新类
- 3、新类的定义可以增加新的数据或新的功能父类的功能，但不能选择性地继承父类
- 4、满足“A is a B”的关系

# 继承



# 继承

特点：

- 1、利于代码复
- 2、缩短开发周期

# 继承

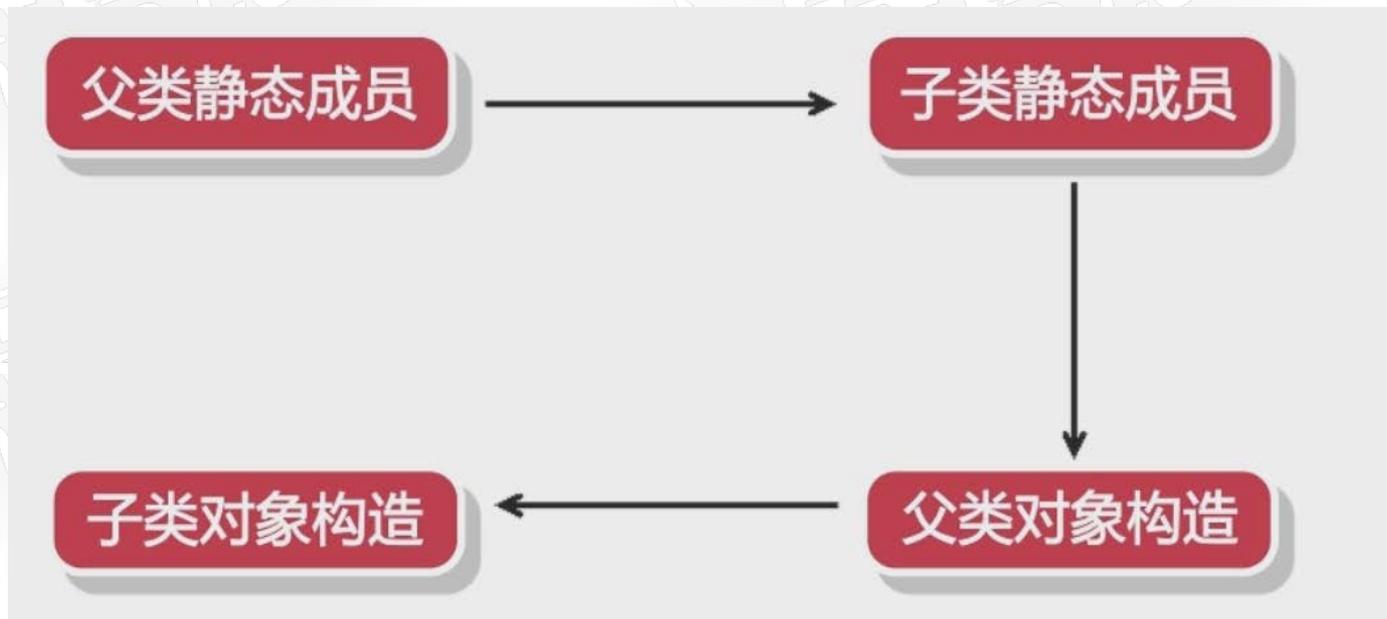
语法：

- 1、使用extends实现继承
- 2、单一继承，只能有一个父类



# 继承

## 继承后的初始化顺序



# super

## 继承后的初始化顺序

### – 访问父类成员方法

```
super.print();
```

### – 访问父类属性

```
super.name;
```

### – 访问父类构造方法

```
super();
```

# super

- 子类的构造的过程中必须调用其父类的构造方法,默认调用无参的构造方法。
- 如果子类构造方法中既没有显示标注,而父类又没有无参的构造方法,则编译出错
- 使用super调用父类指定构造方法,必须在子类的构造方法的第一行

# super pk this



## this: 当前类对象的引用

- 访问当前类的成员方法
- 访问当前类的成员属性
- 访问当前类的构造方法
- 不能在静态方法中使用



## super: 父类对象的引用

- 访问父类的成员方法
- 访问父类的成员属性
- 访问父类的构造方法
- 不能在静态方法中使用

# super pk this

构造方法调用时，super和this不能同时出现

```
25 public Cat(String name,int month){
26     /* 子类构造默认调用父类无参构造方法
27     * 可以通过super()调用父类允许被访问的其他构造方法
28     * super()必须放在子类构造方法有效代码第一行
29     */
30     this();
31     super(name,month); //this
32
33     System.out.println("我是子类的带参构造方法");
34 }
```

# 方法重写 PK 方法重载

- **方法重写：**

- 在满足继承关系的子类中
- 方法名、参数个数、顺序、类型与父类、返回值相同
- 访问修饰符的限定范围大于等于父类方法

- **方法重载：**

- 在同一个类中
- 方法名相同
- 参数个数、顺序、类型不同
- 返回值类型、访问修饰符任意

# 访问修饰符

访问修饰符	本类	同包	子类	其他
<b>private</b>	√			
默认	√	√		
<b>protected</b>	√	√	√	
<b>public</b>	√	√	√	√