面向对象

1、什么是对象
2、什么是面向对象
3、什么是类
4、什么是属性
5、什么是方法
6、类和对象的关系

什么是对象

- 万物皆对象、客观存在的事物
- 对象：用来描述客观事物的一个实体，由一组属性和方法构成

什么是面向对象

- 人关注对象
- 人关注事物信息

类

- 类是模子，确定对象将会拥有的特征（属性）和行为（方法）
- 类的特点
  - 类是对象的类型
  - 具有相同属性和方法的一组对象的集合

什么是对象的属性和方法

- 属性：对象具有的各种静态特征
- “有什么”
- 方法：对象具有的各种动态行为
- “能做什么”

类和对象的关系

- 类是抽象的概念，仅仅是模板
- 对象是一个你能够看得到、摸得着的具体实体
- 类是对象的类型
- 对象是特定类型的数据

- 具体开发过程中，先定义类再实例化对象

单一职责原则

单一职责原则

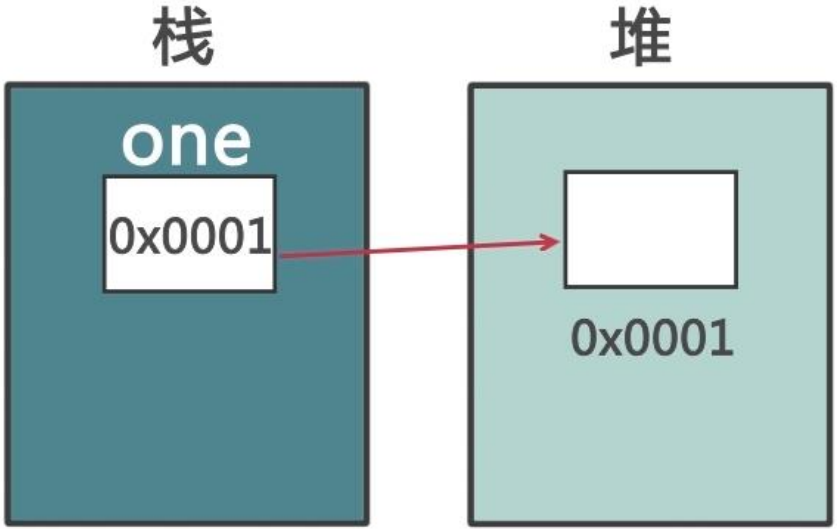
- 单一职责原则，也称为单一功能原则
- 英文Single Responsibility Principle 缩写SRP
- 是面向对象设计中的一个重要原则
- 一个类，应该有且只有一个引起变化的原因

- 在程序设计中，尽量把不同的职责，放在不同的职责中，即把不同的变化原因，封装到不同的类中。

对象实例化需要了解的

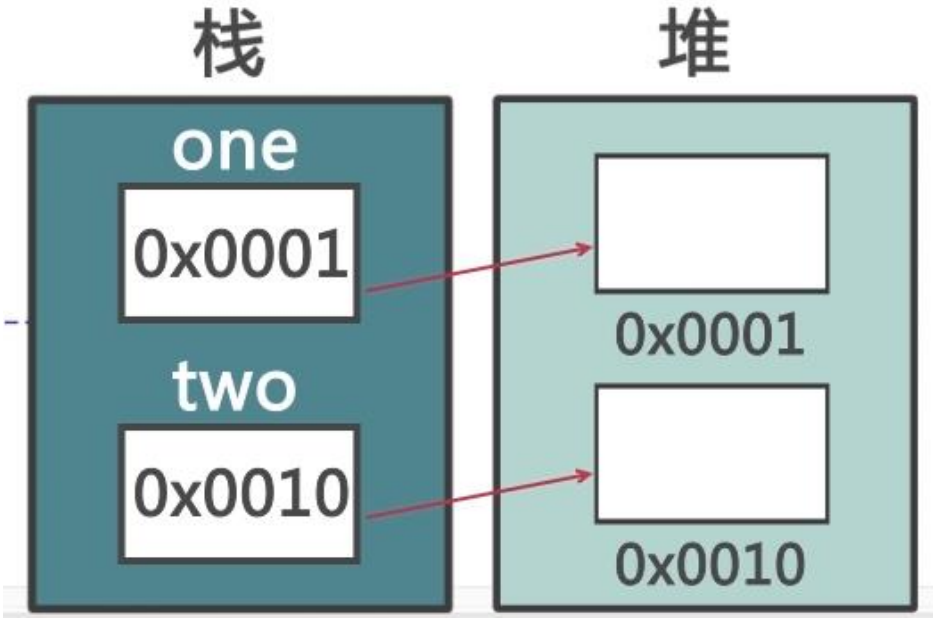
对象实例化

- 实例化对象的过程可以分为两部分：
  - 声明对象　Cat one
  - 实例化对象　new Cat();
  - Cat one=new Cat();



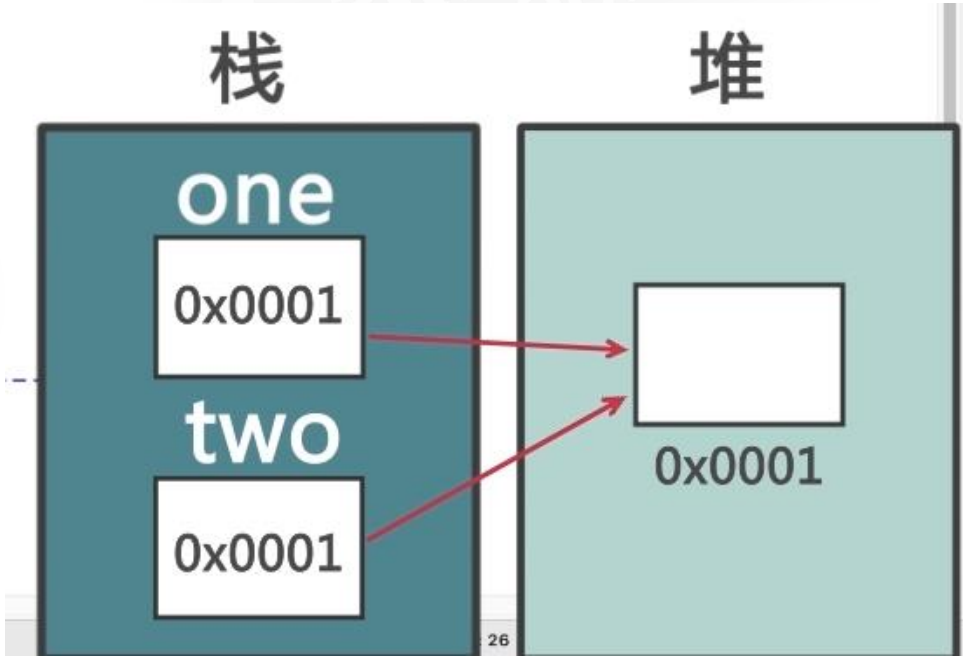
对象实例化

- 每次new对象会产生新的实例化对象
  - Cat one=new Cat();
  - Cat two=new Cat();



对象实例化

- 多个对象可以指向同一块实例化空间
  - Cat one=new Cat();
  - Cat two=one;



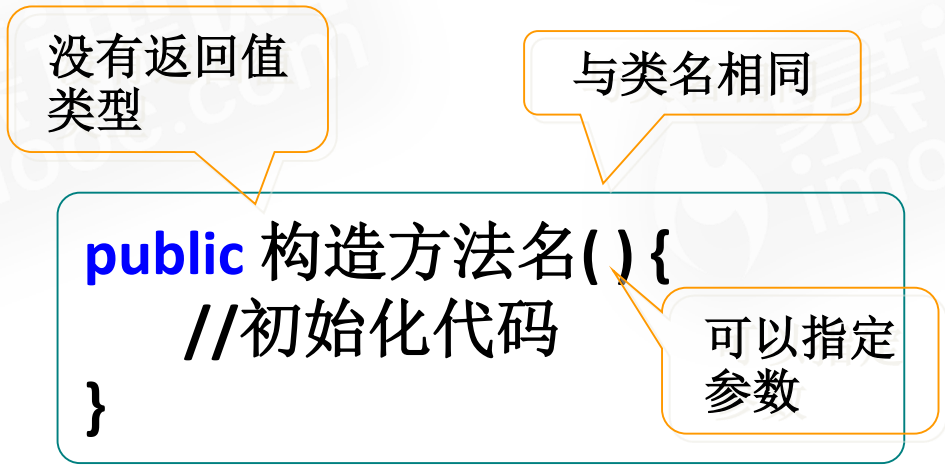
对象实例化

- 对象必须被实例化之后才能使用
- 对象间的引用传递，实际上传递的是堆内存空间的使用权

构造方法

构造方法

1、构造方法与类同名且没有返回值
2、构造方法的语句格式
3、只能在对象实例化的时候调用



构造方法

4、当没有指定构造方法时，系统会自动添加无参的构造方法

5、当有指定构造方法，无论是有参、无参的构造方法，都不会自动添加无参的构造方法

6、一个类中可以有多个构造方法

this关键字

this关键字

- this：当前对象的默认引用
- this的使用
  - 调用成员变量，解决成员属性和局部变量同名冲突

```
// 成员属性：昵称、年龄、体重、品种
String name;// 昵称
int month;// 年龄

public Cat(String name) {
    this.name=name;
    System.out.println("我是单参构造");
}
```

  - 调用成员方法

```
// 成员方法：跑动、吃东西
// 跑动的方法
public void run() {
    this.eat();
    System.out.println("小猫快跑");
}

// 吃东西的方法
public void eat() {
    System.out.println("小猫吃鱼");
}
```

this关键字

- this的使用
  - 调用重载的构造方法

```
public Cat(){
    System.out.println("我是无参构造");
}

public Cat(String name) {
    this();
    this.name=name;
    System.out.println("我是单参构造");
}
```

必须是构造方法第一条语句