



JAVA面向对象

类和对象

类

对象

属性和方法

单一职责原则

拆分后的类如何交互

对象实例化

栈

堆

执行流程

注意事项

构造方法

this 关键字

类和对象

- 万物皆对象
- 类是模子，确定对象将会拥有的特征（属性）和行为（方法）
- 对象是类的实例表现


类

- 抽象的概念
- 模板

对象

- 类实例产生的具体体现

属性和方法

- 属性：对象具有的各种静态特征（对象有什么）
- 方法：对象具有的各种动态行为（对象能做什么）
- 当对类的属性进行定义但未赋值时，属性具有默认值
 - String类型默认值 `null`
 - int类型默认值 `0`
 - double类型默认值 `0.0`
- 方法可重载（见  [JAVA方法](#)）

单一职责原则

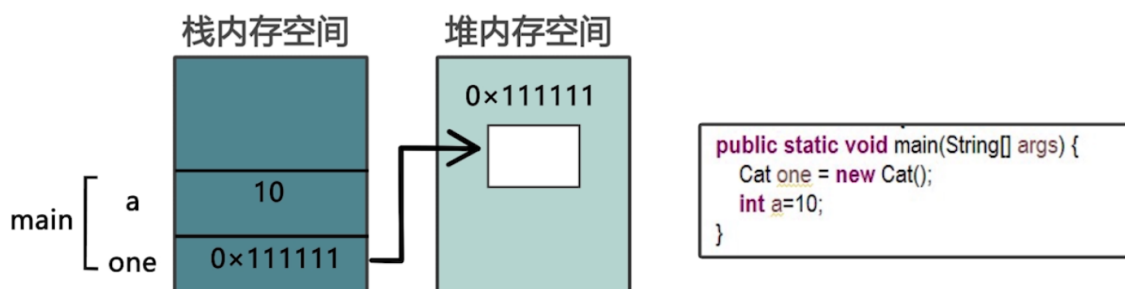
- 也叫单一功能原则
- 一个类有且只有一个引起功能变化的原因
- 承载的功能越多——交融、耦合性高，复用性低

拆分后的类如何交互

- 优先寻找同个包内允许访问的类、成员和方法

对象实例化

- 实例化对象的过程分为两部分：
 - 声明对象：开辟内存栈空间，未初始化
 - 实例化对象：开辟内存堆空间
- JAVA程序-JVM-操作系统



- 内存分配由JVM执行

栈

- 每个方法（Method）在执行时，都会创建一个栈帧，用于存储局部变量表、操作数、动态链接、方法出口信息等
- 栈中所存储，多用于保存局部信息的值，譬如：方法体中的基本数据类型定义的变量、对象的引用（也称为对象实例）等。当局部作用范围结束时，栈内信息立即自动释放
- 当存储内容是由基本数据类型（byte、short、int、long、float、double、char、boolean）声明的局部变量时，在栈中存储的是他们对应的具体数值
- 当存储的是局部的对象的引用（定义在方法体中的引用类型的变量），存储的是具体对象在堆中的地址。如果对象的引用没有指向具体的空间，则是null

堆

- 用来存放动态产生的数据，比如new出来的对象。当对象使用结束，并确定已无实例引用指向堆空间时，JVM才会依据相关垃圾回收机制进行资源回收，完成堆内资源释放，也就是说，并不是方法结束，方法内涉及到的堆空间就会立即释放
- 此时创建出来的对象只包含属于各自的成员变量，并不包括成员方法。因为同一个类拥有各自的成员变量，存储在堆中的不同位置，但是同一个类不同实例之间共享该类的方法，并不是每创建一个对象就把成员方法复制一次

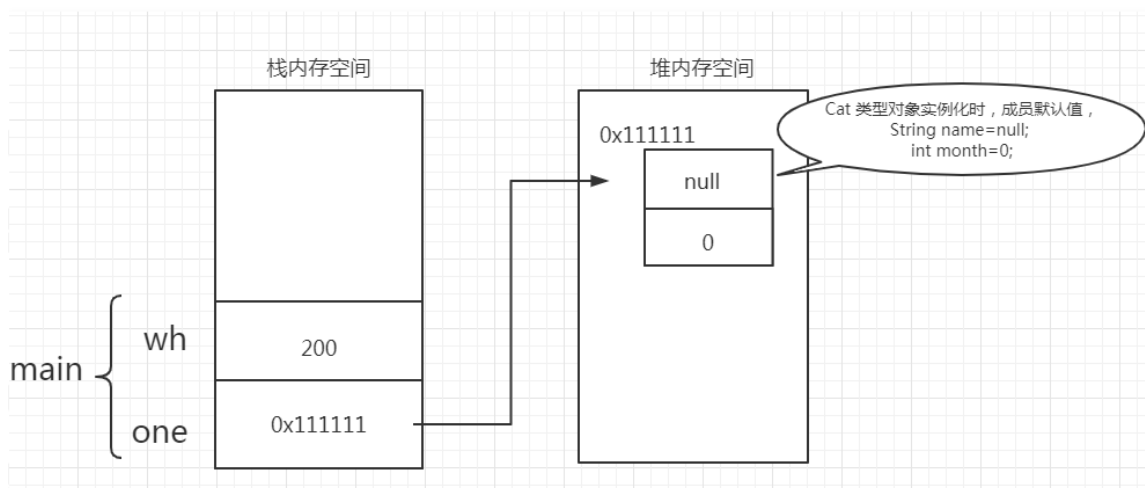
执行流程

```

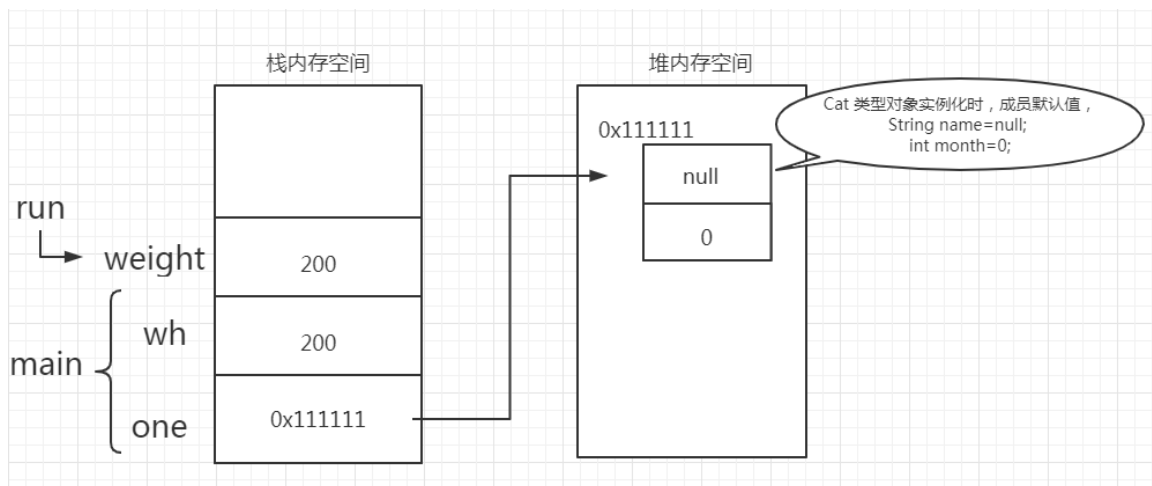
public class CatTest {
    public static void main(String[] args) {
        // 对象实例化
        Cat one = new Cat();
        double wh=200;
        one.run(wh);
    }
}

```

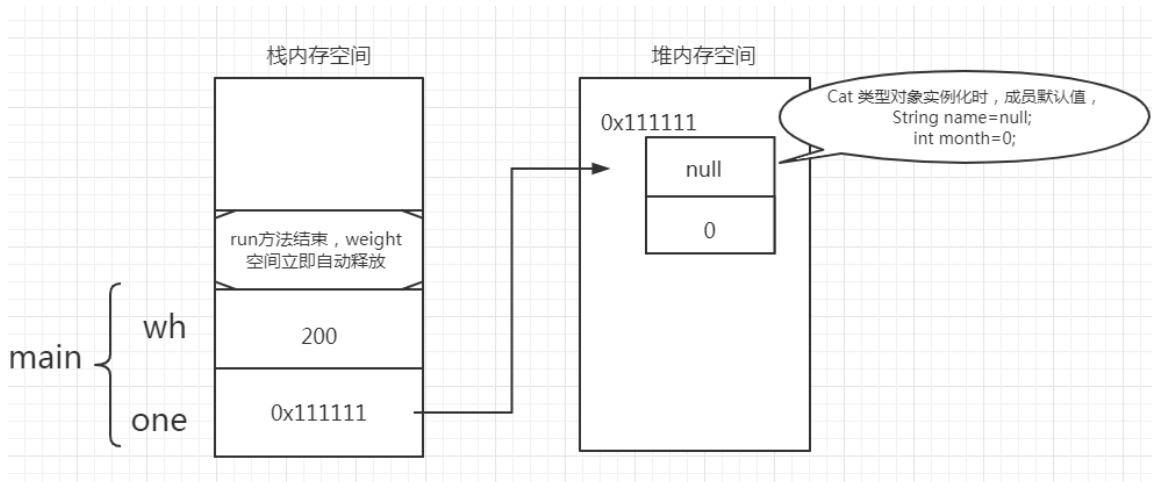
1. JVM自动寻找main方法，执行第一句代码，创建一个Cat类的实例one，在栈中分配一块内存，存放一个指向堆区对象的地址譬如0x111111
2. 创建一个double类型的变量wh，由于是基础数据类型，直接在栈中存放double对应的值200



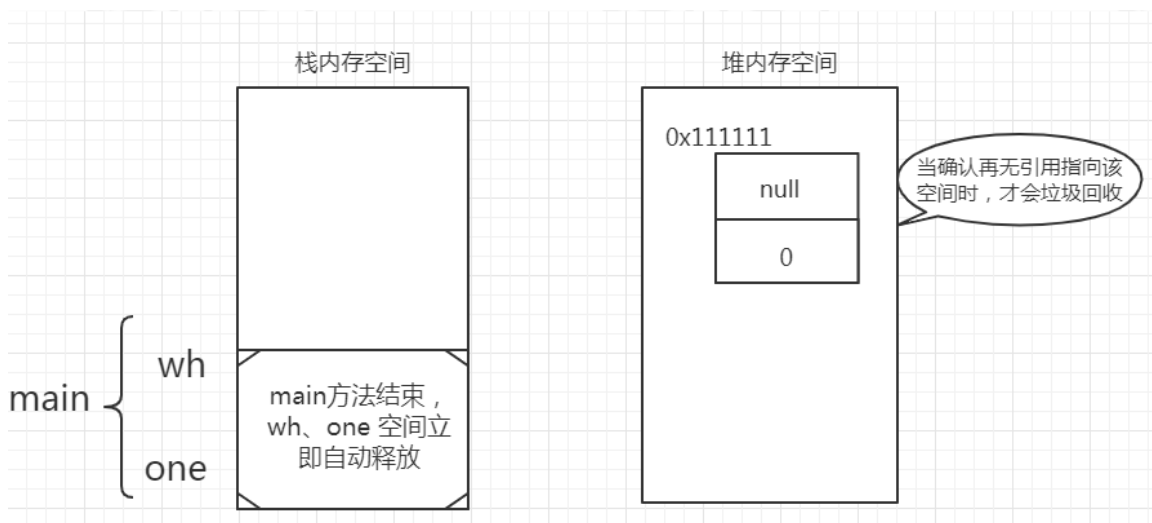
3. 当调用one对象的run方法，并以wh为参数传入方法时，JVM检测到方法参数weight，作为方法局部变量，也会放入栈中并将wh的值200复制给weight



4. 当完成run方法调用后，立即释放局部变量weight所占用的栈空间



5. main方法运行结束，立即释放栈中one，wh的空间，而堆中对象空间，则会当确定无引用指向后，由垃圾回收机制进行回收，不会立即释放资源



注意事项

- 需要多次访问同一对象时，必须进行声明
- 只需访问一次时，可使用匿名对象进行方法调用

```
public class CatTest{  
    public static void main(String[] args){  
        new Cat().run();  
    }  
}
```

- 同一作用范围内，不能定义同名对象
- 可以声明多个引用，用逗号分隔（不推荐）

- `Cat one, two;`
- `Cat three = new Cat(), four = new Cat();`

构造方法

- 也称为构造函数、构造器、Constructor
- 只能与 `new` 一起进行调用，不能被对象单独调用
- 与类同名且没有返回值，且不能用 `void` 修饰
- 构造方法的语句格式
 - `public` 访问修饰符
 - 没有返回值类型
 - 方法名与类名相同
 - `()` 内可以指定参数

```
public 构造方法名(){  
    //初始化代码  
}
```

- 只能在对象实例化时调用
- 当没有指定构造方法时，系统会自动添加无参的构造方法

- 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
- 一个类中可以有多多个构造方法
- 可以通过 `this()` 关键字在构造方法中调用别的构造方法，必须放在构造方法首行
- 一个构造方法内只能调用一个构造方法（首行）

`this` 关键字

- 代表当前对象本身，指向当前对象的一个引用
- 可用于调用成员属性、成员方法、构造方法，也可以当做参数进行方法传参以及方法返回值