# Day06 面向对象

## 面向对象与面向过程的比较

### 面向过程：

注重的是程序执行的过程，强调的是动作

### 面向对象：

注重的是一个对象，只要有了这个对象，也就掌握了这个对象 所具有的所有的方法

举一个现实中的例子，当我们饿了，我们可以选择自己去买菜，洗菜，做饭， 当进行这个过程的时候，我们清楚买菜的过程，洗菜的过程，做饭的过程，这 就是面向过程，此外，我们还有另一种选择，那就是去饭店里吃，去饭店吃的 话，只要你告诉厨师你要吃什么，其他的过程你就不用管了，不考虑买菜洗菜 做饭的过程，而是只要等着厨师把饭做好就可以了，你所关注的，就是厨师做 没做好这就是面向对象的过程。

注意：当我们要做一些比较简单的场景的时候，面向过程比较好，而做一些比 较复杂的过程的时候，面向对象比较好

## 类与对象的关系

### 类的定义：

将一些共有的属性和行为抽取出来，就形成了一个类，举例来说：人，人一般都有名字，性别，年龄这些属性，以及吃饭，说话等一些行为，将这些属性和行为抽取出来，就可以表示为人类。

### 对象的定义：

将类这个宽泛的定义具体化，例如，人类可以泛指很多人，对象则指的是具体的某一个人。所指的这个人就是人类的一个对象。

### 类和对象的关系：

根据一类对象的共有的特征和行为进行抽取，将这类事物用类来表示，将特征抽取成了属性，将行为抽取成方法---类是对象的抽取 /概括

利用new关键字来创建对应的类的对象/实例---对象是类的具体化

## 成员变量和局部变量的关系

### 定义位置：

成员变量（属性）是指定义在类内方法外的变量；定义在了方法或者语句内的变量就是局部变量。

### 作用范围：

成员变量是作用在整个类内；局部变量只能作用于对应的方法或者语句中。

### 内存位置：

成员变量是存储在堆内存中，并且自动的赋予了一个默认值；局部变量存在栈内存中，不会自动给默认值。

### 生命周期：

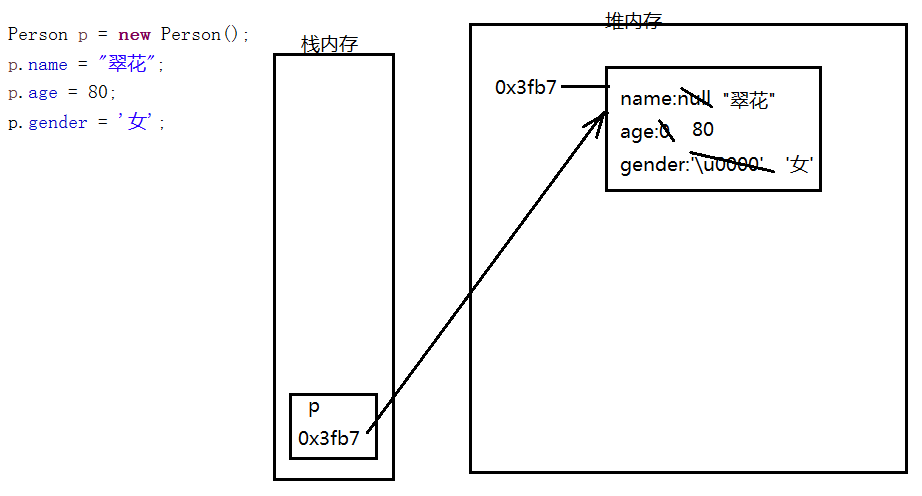
成员变量随着对象的创建而存储到堆内存中，随着对象的回收而被销毁；局部变量在方法或者语句执行完成之后立即回收。

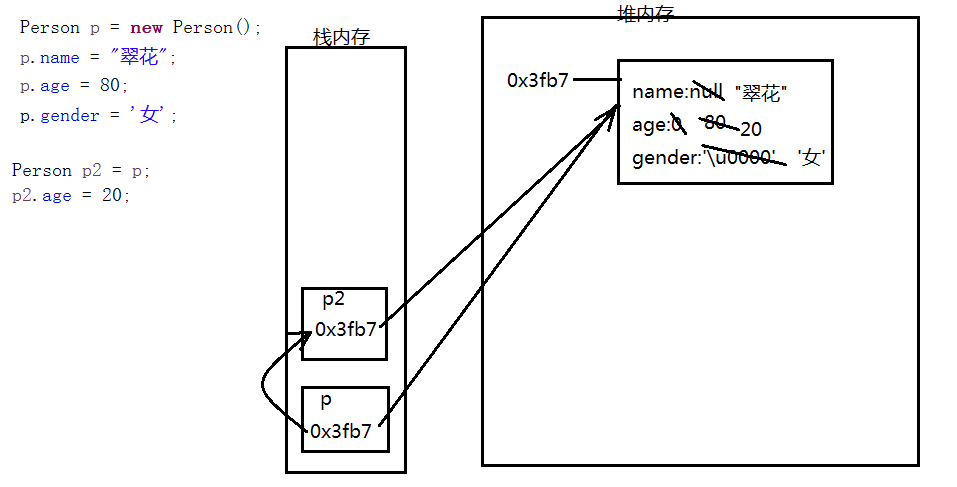
## 对象在内存中的存储

### 对象的存储位置：

对象是存储在堆内存中的

### 图形表示：





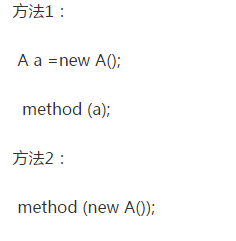
## 匿名对象

### 匿名对象的定义：

根据名字我们可以知道，匿名对象，就是没有名字的对 象，因为他没有名字，所以我们只能在创建它的时候使用一次。

### 使用场景：

适用于那些我们只需要使用一次，不需要重复利用此对象的 情况，例如：



当我们写一方法，参数需要传入某个对象的时候，这个对象我们只需要用一次， 就可以使用匿名对象的方式。

## 构造函数

### 定义：

在类中与类同名而没有返回值类型的方法叫做构造方法。

### 构造方法的作用：

当创建对象的时候需要调用目标类的构造方法。

### 知识点：

在类中，如果没有手动添加构造方法，那么Java会在编译的时 候添加一个默认的无参构造，如果手动添加了构造方法，Java就不再添加默认的构造方法了。

构造方法可以重载。

## This

### this的定义：

this代表当前在活动的对象---可以认为是一个虚拟对象---用于在类内调 用本类中的非静态方法和非静态属性的。

因为在java中所有的非静态的方法和属性都是通过对象来调用的，而在本类 中如果想调用本类中的方法和属性，并没有本类的对象，所以要用到this来 代表当前类的对象，调用属性和方法

注意：this语句不能放在main函数中使用

### this语句：this（）；

This语句放在构造函数的第一行，表示用来在本类中调用其他的构造方法。

## 局部代码块

### 局部代码块的定义：

在方法或者语句内部用{}括起来的一段代码，

例如：for循环

For（int i=0;i<5;i++）{

代码块；

}

用代码括起来的代码块就叫做局部代码块

### 局部代码块的意义：

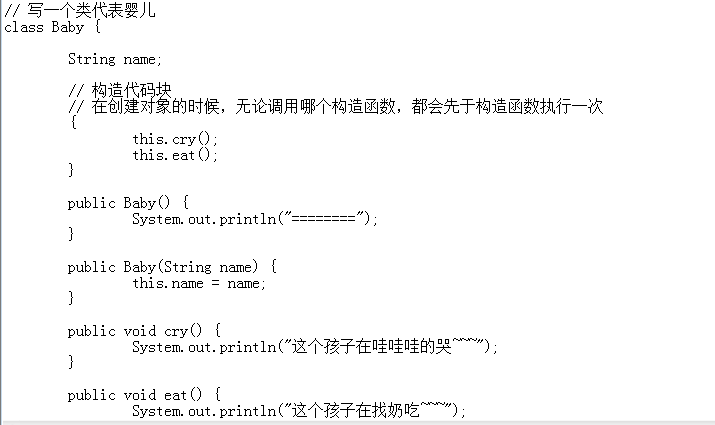
限制变量的使用范围和生命周期，以提高内存的利用率

## 构造代码块

### 构造代码块的定义：

在类内方法外用{}括起来的一段代码，也叫做初始化代码块

例如：



### 构造代码块的意义：

因为构造代码块是先于构造函数执行的，所以我们可以通过构造函数来实现一 些需要在对象创建以前所需要的初始化的工作。