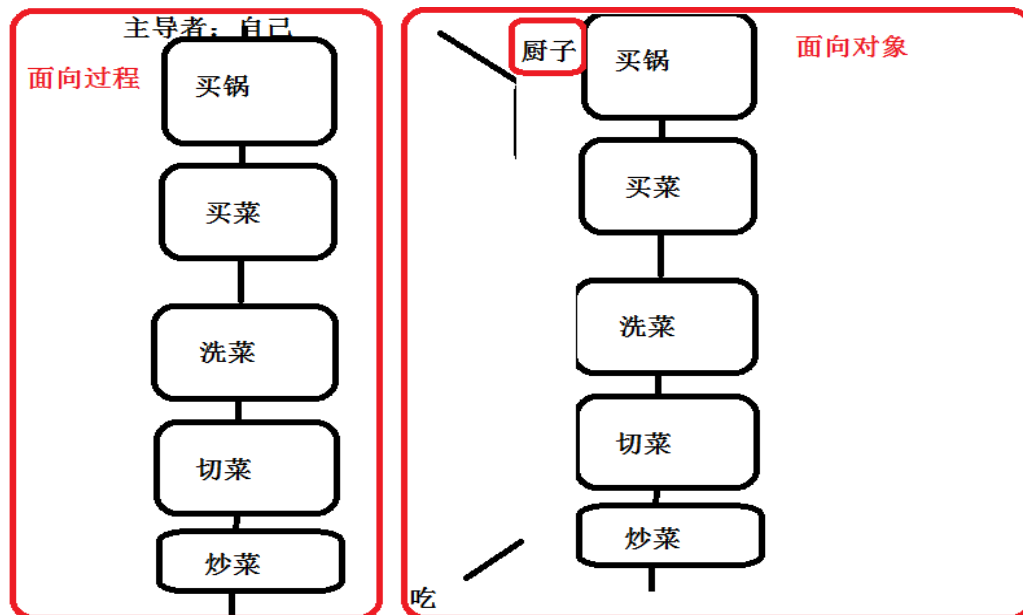


面向对象

2019年11月8日 9:43

面向过程和面向对象：



面向过程：注重过程，关注的是功能实现过程中的每一步；主导者永远都是自己

面向对象：注重对象，找到一个具备需求功能的对象，通过使用对象身上的所有的功能就可以完成整个过程；主导者变为具备功能的对象

面向对象基于面向过程。

面向对象好还是面向过程好？

面向对象：贴近人类思考问题的模式；对于复杂的大型项目，更利于进行模块开发和人员的分工合作；

面向过程：从效率上讲要比面向对象高一些；对于功能简单的项目来说，面向过程更加直接；

类和对象

类

人

name

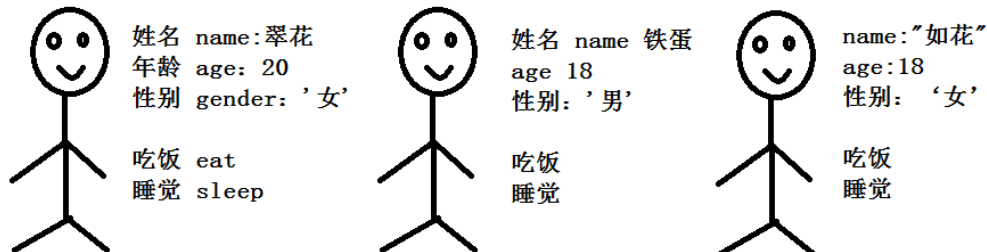
age

gender

eat

sleep

对象



类：抽象的概念；一类事物的概括；

对象：对象是抽象概念的具体化/实例化；

定义类：

格式：

```
修饰符 class 类名{
    属性;
    方法;
}
```

属性：类所包含的一系列的特征

方法：类所包含的一系列的行为

类	人	定义类：
name		class Person{
age		变量：数据类型 变量名;
gender		String name;
		int age;
		char gender;
eat		方法：修饰符 返回值类型 方法名(参数列表){
sleep		行为的完成;
		}
对象		

如何产生对象：

new关键字可以创建出类的对象

格式：

类名 变量名;

变量名 = new 类名();

类名 变量名 = new 类名();

如果获取对象的属性:

格式: 对象.属性名

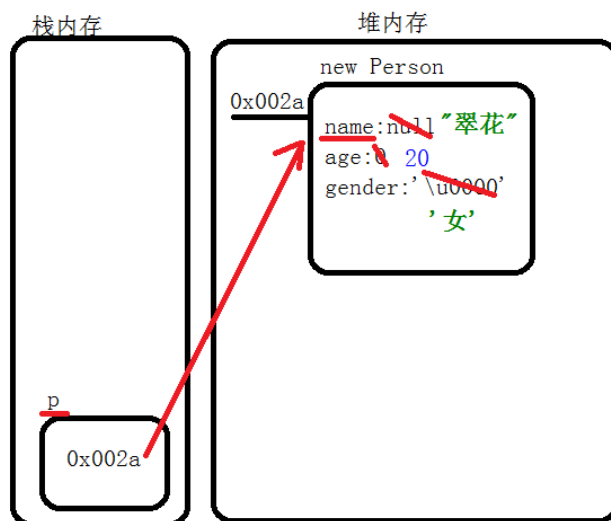
如何给对象的属性赋值:

对象.属性名 = 值;

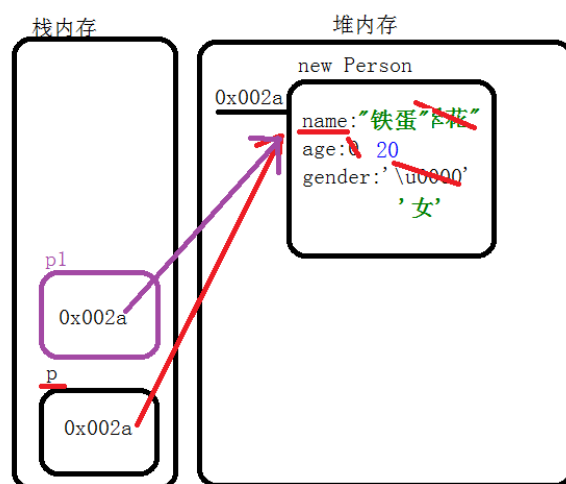
如何调用对象的方法:

格式: 对象.方法名();

对象在内存中的存储:



```
main:
Person p = new Person();
p.name = "翠花";
p.age = 20;
p.gender = '女';
Person:
class Person{
    String name;
    int age;
    char gender;
}
```



```
main:
Person p = new Person();
p.name = "翠花";
p.age = 20;
p.gender = '女';

Person p1 = p;
p1.name = "铁蛋";
System.out.println(p1.name);
System.out.println(p.name);
// "铁蛋"

Person:
class Person{
    String name;
    int age;
    char gender;
}
```

成员变量: 定义在类中代表类的属性特征

局部变量: 定义在方法内、方法上、语句中的变量

成员变量和局部变量的区别:

- 1、**定义的位置不同**：成员变量定义在类中；局部变量定义在方法内或语句内部
- 2、**作用域不同**：成员变量的作用域是整个类；局部变量作用域仅在方法中或语句内部生效
- 3、**内存中存储的位置不同**：成员变量存储在堆内存，在定义时会赋值默认值；局部变量存储在栈内存，不会被赋值默认值；
- 4、**生命周期**：成员变量随着对象的创建而创建，随着对象被垃圾回收而释放；局部变量是在方法或者语句执行时创建，当方法或语句执行完成也随之释放；

构造方法

用来创建对象的方法，通过new创建对象时执行；

格式：

```
修饰符 void 类名([参数列表]){  
    方法体;  
}
```

注意：

- 1、如果类中没有明确定义构造方法，则编译器会默认生成一个无参构造；如果已经定义了构造方法，则编译器不会再默认生成；
- 2、构造方法可以重载
- 3、构造方法中可以写return语句，避免问题；

this:

除了静态方法和静态变量其他的都需要通过对象来访问

this指代的是当前的对象；灵活的表现方式，谁调用方法，this就指代哪个对象；

应用：

- 1、this.属性名 可以给当前对象的属性直接赋值
- 2、this可以调用当前类中其他的非静态方法 可以省略this
- 3、this语句：java支持通过this语句调用当前类的其他的构造方法；
this语句必须放在首行；
- 4、this可以代表当前对象通过return作为返回值 返回

