类与对象

1. 创建一个类以及对象：

public class a{

public static void main(String[]args){

Scanner myscanner = new Scanner(System.in);

Cat cat1 = new Cat();//cat1指向由new新建的对象cat

cat1.name ="甘甘"；

cat1.age = 114514；

cat1.color = "yellow";

Cat cat2 = new Cat();

cat1.name ="坤坤"；

cat1.age = 1145141919；

cat1.color = "blue";

}

}

class cat{//定义一个猫类

String name;

int age;

String color;

}

1. 对象的内存布局（引用类型，有点像数组的存放方式），当新建一个对象会存放一个地址在栈中，地址指向堆区，在堆区中存放对象中的属性。 属性中的字符串类型会指向方法区的常量池中
2. 内存分配补充：类信息在方法区中加载，新建对象在堆区中，栈中一个变量指向对象的地址，对象的string类型会存储在方法区的常量池中（string是引用类型），在常量区中的string类型会把他的地址再给堆区中的对象。
3. 成员变量 == 属性 == field
4. 访问修饰符：控制方法的访问范围。
5. public：
6. protected：
7. 默认：
8. Private：
9. 方法，方法中可以执行很多操作，类似循环，创建成员方法时需要public void xxx()；

可以给方法传参（感觉就是在类里面写一个函数）

class cat{

public void sum(){

int sum=0;

for(int i=0;i<1000;i++)sum+=i;

System.out.print(sum);

}

}

1. 方法调用机制
2. 首先，主方法Main会在栈中开辟一块空间。
3. 加载类信息，new之后会在堆区新建一个对象由新建的变量指向这个对象。
4. 当程序在执行对象的方法时，会在栈区中新建一个独立的空间，被执行方法的语句所指向。
5. 被调用的方法执行到return语句后，会返回到调用方法的位置并销毁他在栈区中的空间。