**一、类和对象**

1.类（自行车的设计图）

对象（一辆辆的自行车）

2.为什么要设计类和对象的机制（面向对象的编程)

状态 + 行为

属性 方法

3.面向对象的三大特性：

1.封装 2.继承 3.多态

4.语法

1）定义类的语法

修饰符（modifier）class 类名称{
1.属性
2.方法
3.构造方法
}

2）属性的定义

修饰符（modifier）变量类型 变量名称 = ;
默认值：变量类型的默认值

3）方法的定义

修饰符（modifier）返回值类型 方法名称（形参列表）{
实现
}

3）构造方法的定义

修饰符（modifier）类名称（形参列表）{
}

4）构造对象的语法（对象实例化的语法）

类名称 变量名称 = new 类名称（实参列表）
(引用类型——类引用)
匿名对象： new 类名称（实参列表）

5）

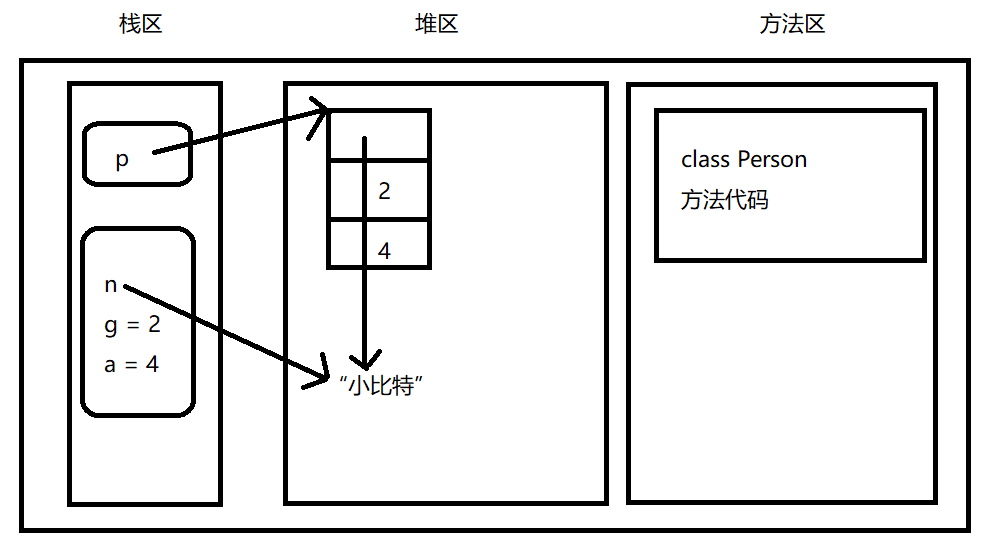
class Person{
public String name = null;
public int gender = 0;//0f 1m
public int age = 0;
public String toString(){
return String.format("Person:(%s,%d,%d)",name,gender,age);
}
//构造方法
public Person(String n,int g,int a){ //n为引用类型，g和a是基本类型
name = n;
gender = g;
age = a;
}
}
public class java0720{
public static void main(String[] args){
Person p = new Person("xiaobite",2,4);
System.out.println(p.toString());
}
}

(1）一个文件中只能有一个public的类

(2）new：计算对象需要的内存大小，开辟空间，初始化为0

(3）调用构造方法，对属性初始化

(4）构造方法调用结束，返回一个指向新对象的引用



6）如何访问对象的属性（类的内部与类的外部）

内部：

属性 直接通过属性名称访问即可

方法 直接通过方法名称访问即可

外部：

属性 引用 . 属性名称

方法 引用 . 方法名称

类的内部
class Person{
int a;
int b;
public void methodA(){
a=10;
System.out.println(a);
methodB();
}
public void methodB(){
}
}
类的外部
class Main(){
Person p = new Person();
p.a;
p.methodA();
}

7）嵌套调用

Sysetm.out.println(Arrays.toString(rotate(a,3)));
//
int[] tmp1 = a[];
int[] tmp2 = rotate(tmp1,3);
String tmp3 = Arrays.toString(tmp2);
System.out.println(tmp3);

8)链式调用

class A{
public A create(){
return new A;
//A tmp1=new A;
//return tmp1;
}
}
class B{
public static void main(){
new A().crate().create().create();
//A tmp1 = new A();
//A tmp2 = tmp1.create();
//A tmp3 = tmp2.create();
//A tmp4 = tmp3.create();
}
}

Person p = null;

定义了一个没有指向任何对象的Person类型的引用（类类型的引用）

Person p = new Person(...);

1）定义了一个Person类型的对象

2）定义了一个指向Person类型的对象的Person类型的引用（类类型引用）

引用类型和指向的对象类型是俩回事。

9）this关键字的使用

（1）在构造方法调用其他构造方法。调用语句必须出现在第一行。

class Person(){
String name;
public Person(String n){
name = n;
}
public Person(){
this("陌生人")；
//name = "陌生人";
}
}

（2）通过this访问属性或方法（一般出现在发生了命名遮挡（name shadowing）的时候）

class Person(){
String name;
public Person(String name){
name = this.name;
this.toString();
}
public String toString(){
...
}
}

(3)this代表的就是当前对象（current object）的引用

class Person{
String name;
public Person(String name){
this.name=name;
}
public String toString(){
return this.name;
}
public boolean equals(Person p){
return p.name = this.name;
}
public void test(){
this.equals(this);
}
}

重点：普通属性和普通方法，其实都绑定着一个隐含的对象。

对象就是放在堆中一块区域，主要保存的就是属性的值，其次还有找到在方法区中保存类的信息。

Person p = new Person();

p是一个Person类型的引用，指向一个Person类型的对象。