* class Circle {

private double radius;

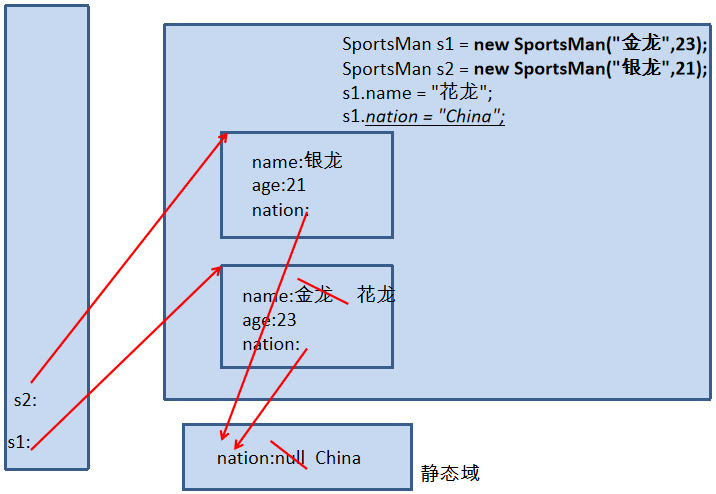
public Circle(double radius) {this.radius = radius;}

public double findArea() {return Math.PI\*radius\*radius;}

}

* **创建两个Circle对象**
* Circle c1=new Circle(2.0); //c1.radius=2.0
* Circle c2=new Circle(3.0); //c2.radius=3.0
* **Circle类中的变量radius是一个实例变量（instance variable），它属于类的每一个对象，不能被同一个类的不同对象所共享。**
* **上例中c1的radius独立于c2的radius，存储在不同的空间。c1中的radius变化不会影响c2的radius，反之亦然。**

**如果想让一个类的所有实例共享数据，就用类变量。**



* **类属性、类方法的设计思想**
* 类属性作为该类各个对象之间共享的变量。**在设计类时，分析哪些类属性不因对象的不同而改变，将这些属性设置为类属性。相应的方法设置为类方法**。
* **如果方法与调用者无关，则这样的方法通常被声明为类方法，由于不需要创建对象就可以调用类方法，从而简化了方法的调用**。
* **使用范围：**
* 在Java类中，可用static修饰**属性**、**方法**、**代码块**、**内部类**。
* **类变量**
* 被修饰后的属性（类变量）具备以下特点：
* **随着类的加载而加载**。
* **优先于对象存在**。
* 修饰的成员，被所有对象所**共享**。
* 访问权限允许时，可不创建对象，**直接被类调用**。
* **类方法**
* 没有对象的实例时，可以用**类名.方法名()**的形式访问由static标记的类方法。
* 在**static方法内部只能访问类的static属性**，不能访问类的非static属性。
* 因为不需要实例就可以访问static方法，因此**static方法内部不能有this。（也不能有super）**
* 重载的方法需要同时为static的或非static的。