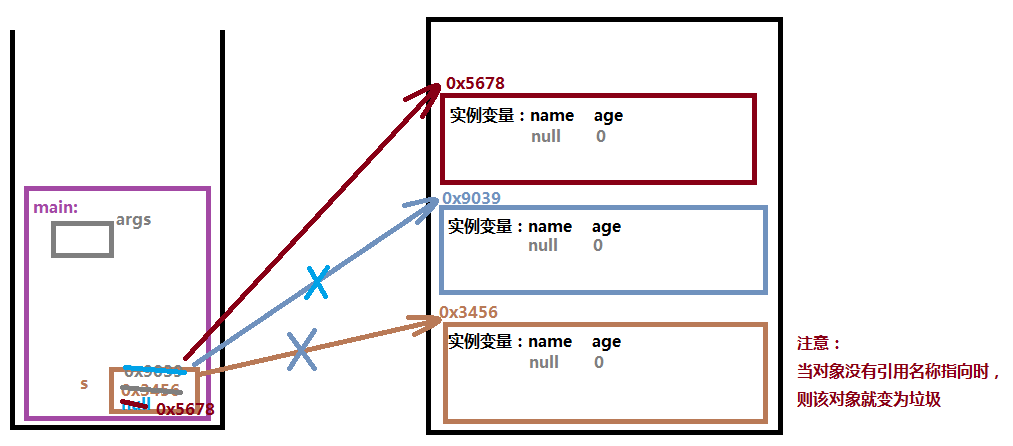
## Day08

### 8-1 变量按数据类型分为

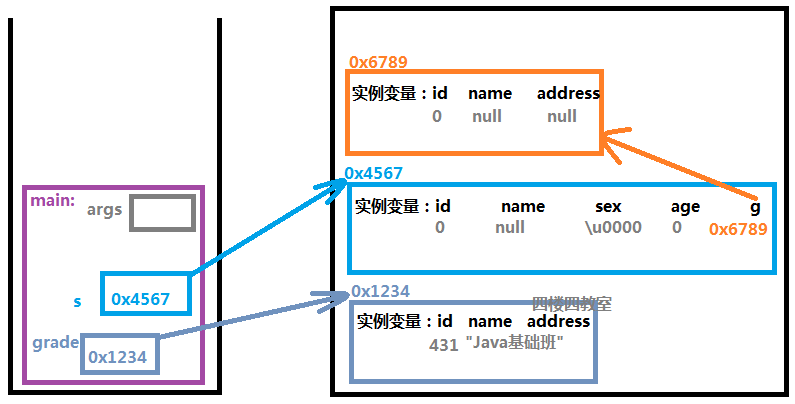
* 基本数据类型：在内存中存放的是真正的值
  + 4大类，共计8种
* 引用数据类型：在内存中存放的是地址(引用)
  + String
  + 用户自定义类型，如：Student Computer……..
* **对引用数据类型赋值的方式：**
  + 第一种：将对象赋给引用名称
    - 如：s = new Student();
  + 第二种：将null值赋给引用名称
    - 如：s = null;

### 8-2 空指针异常

* 属于运行错误，java.lang.NullPointerException
* 原因：当引用名称的值为null时，就不能方法某个对象中的属性或方法，如果非要访问则就出现空指针异常
* 解决办法：在访问某个对象中的属性或方法之前必须保证该引用名称中存放的是对象的地址(引用)



### 8-3 当一个对象中的属性是引用数据类型



### 8-4 封装

* 面向对象有三大特征：
  + 封装、继承、多态
* 封装
  + 定义：隐藏类的内部信息，不允许外部程序直接访问，而是通过方法进行操作
  + 封装操作步骤

第一步：将属性设置为私有的private,只能在本类中使用

**private int age;**

第二步：编写对应属性的赋值setXxx和取值getXxx方法

**public void setAge(int age){**

**//第三步：根据情况编写判断语句**

**//将局部变量的值赋给成员变量,年龄：0---100，否则为18**

**if(age >= 0 && age <= 100){**

**this.age = age;**

**}else{**

**this.age = 18;**

**}**

**}**

**public int getAge(){**

**return age; //表示返回成员变量age的值**

**}**

### 8-5 this关键字

* this表示当前这个对象，也就是说当前谁调用这个方法，则这个对象就是谁
* this关键字可以**访问本类中**的实例变量
  + 当局部变量和成员变量没有同名时，则编写属性名等价于this.属性名，如：id 等价于 this.id
  + 当局部变量和成员变量同名时，并且访问成员变量则必须使用this.
* this关键字可以**访问本类中的**实例方法
  + 方法名称([参数列表]); 等价于 this.方法名称([参数列表]);
* this关键字还可以访问本类中的其他构造方法
  + this(); //访问本类无参构造方法

this([实参列表]);

* + 当使用this访问本类构造方法时，则只能编写在构造方法中，并且是第一条语句
  + 构造方法不能出现递归调用