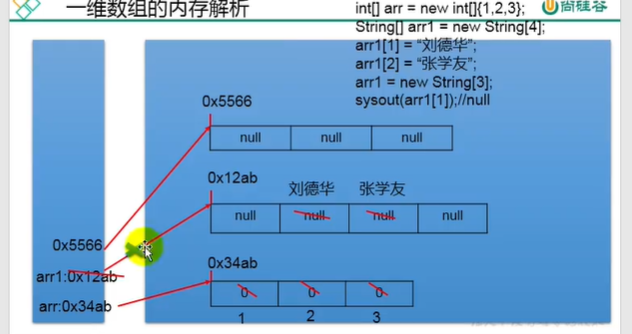
Ctrl加鼠标左键点变量名可跳转, al加方向键回去 左再回去



不实例化调用类无法更改属性，为静态调用

Public void无返回值public static voi sort（） public double xxx必须有返回值且类型为doube public static double random(){ returnxxx;}

权限修饰符 //关键字 返回值类型 方法名（//形参列表）方法体{ 可写return ;表示方法结束类似break;后不可再加语句 }

Returnxxx之后更新xxx属性,当前对象的!!!新声明对象无关

局部变量放在栈,属性放在堆(非静态)

方法A中调用方法A 递归调用

常见错误 Sack overflow栈溢出 形参或者局部变量等无限生成

一类/\*\* 注释\*/name 二类用name时鼠标停留可显示注释

数组的二次赋值 arr[i]=new String(0; arr1={。。， 。。， 。。， 。。.};

匿名对象仅调用一次 new S()。Xxxx 可直接点加属性方法



相当于new Phone()一个匿名对象传递到show方法中的新phone对象,间接性有了对象名

函数重载,方法名一样参数不一样,以便不同类型的变量传入使用

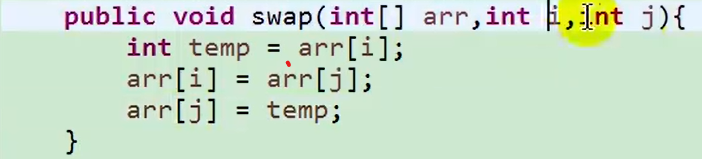
Public void a(String … b){} 可传入多个或零个String类型变量A.a(“aa”,”bb”,”cc”),也有legth等跟数组一样的属性,同一方法的参数列表中放在最后面,不可于数组对象一同使用(String[] s),不可构成重载 –A.a(new String[]{12,3})

引用类型的变量只有两种值,null或地址,String Scaaner或者类名等如类为A ,对象为a与b.。A a=b：将a的地址给b，若a本身为null，则将null给b；而且b的属性等修改会影响a

基本数据类型的变量就是简单的赋值，地址都各自自动有，int a=b；b=10，a还是a

修改形参来达到修改实参的效果可以传入地址，即传入一个对象，引用类型对象储存的就是地址，改形参也相当于修改实参。方法结束后形参销毁，形参的指针销毁，然而并不是只有形参一个地址指向堆空间，实参地址也指向堆空间，所以指针仍然存在，并不能当作垃圾回收堆空间。

之所以形参无法达到修改实参的效果是因为，实参传递过来后只是复制了一份值给形参，并没有复制堆空间，而形参在方法执行结束后都会销毁。形参指向的堆空间也会销毁。



System.exit(0)立即终止程序

递归最后的时候一定是递归到return一个已知的值,这样才能逐级再往上return直到最开始的时候

String是比较特殊的类,定义的对象赋值时储存在字符串常量池中,一旦定赋值了就不能改变, 