out是一个变量还是一个内部类呢？

**不可实例化的System类**

System作为系统类，在JDK的java.lang包中，可见它也是一种java的核心语言特性。System类的构造器由private修饰，不允许被实例化。因此，类中的方法也都是static修饰的静态方法。

由此可见，System中的out和in都不是内部类，而是货真价实的**字段变量**。out是PrintStream的final static修饰的变量字段，也就是说它可以调用PrintStream类的方法。println是PrintStream的一种输出方法，所以我们通常采用System.out.println()在控制台输出内容。

# **System中常用方法**

## ****arraycopy**——**数组拷贝****

arraycopy方法五个参数，分别是被复制的数组，被复制的起始位置，复制到的数组，复制到这个数组的起始位置，复制到这个数组的结束位置。

**int**[] arr1 = {0,1,2,3,4};

**int**[] arr2 = {9,9,9,9,9};

System.*arraycopy*(arr1, 2, arr2, 0, 3);

arr1[3] = 8;

**for** (**int** i = 0; i < 5; i++) {

System.***out***.println(arr2[i] + " ");

// 2 3 4 9 9

}

## ****currentTimeMillis****——****返回毫秒数****

## ****getProperty****——****获取系统属性****

通过调用这个方法，在参数中输入键的字符串获取系统的属性。

String version = System.*getProperty*("java.version");

System.***out***.println(version);

## ****gc****——****运行垃圾回收器****

调用gc方法暗示着Java虚拟机做了一些努力来回收**未用对象**或**失去了所有引用的对象**，以便能够快速地重用这些对象当前占用的内存。当控制权从方法调用中返回时，虚拟机已经尽最大努力从所有丢弃的对象中回收了空间。

## ****exit****——****退出虚拟机****

exit(int)方法终止当前正在运行的 Java 虚拟机，参数解释为状态码。根据惯例，非 0 的状态码表示异常终止。 而且，该方法永远不会正常返回。 这是唯一一个能够退出程序并不执行finally的情况。

# 深度剖析System类源代码

**初始化**

**设置输入/输出/错误流**