java 关键字 static 的一点理解

作者：oldcaptain
原文链接：https://ld246.com/article/1389689404248
来源网站：链滴

<h2>前言：</h2>
      在看 http://blog.csdn.net/mazhimazh/article/details/16805061 这篇文章时候，对于作者对于static的方法在继承的时候，感到不能完全理解。随在网友的帮助下整理以下这文章。
<h2>正文：</h2>
<h3>static关键字的特性：</h3>
    1、被static修饰的变量，方法不依赖类特定的实例，被类的所有实例享
    2、当类被虚拟机加载时，按照声明顺序先后初始化static成员段和static语句块
    3、 在static所修饰的方法和语句块中不能用非static成员字段。
<h3>static变量 ： </h3>
     1、被static修饰的叫静态变量，类变量。没有被static修饰的实例变量。
     2、在jvm分配内存的时候，被static修饰的，只分配一次，被类的所有实例共享。实例变量每new 一个实例，创建一次实例变量，互不影响。
<h3>static方法：</h3>
     1、静态方法，可以被类直接调用，也可以被实例调用不能直接访问所属类的实例变量和方法。
     注：因此静态方法中不能用this和super关键（自己写了一个类，继承了一个父类，子类、父类各有静态方法，但是子类的方法不能使用super，调父类的方法，编译器不过，但是不理解这是为什么？）
    个人猜测： “我得理解 继承，覆写 些全是与实例方法相关的，静态方法不进入方法表，也就是这个class编译的时候就已经确定了要调用个方法了，也就和继承没啥关系了 （来自于网友@总舵主讨论，目前觉得还比较有说服力，坐等@一向北 ,@网警110,@clong@朱@流浪中の狮子@臭猫男等等来做详细解答，排名不先后）”
<h3>static代码块 ： </h3>
     static代码块也叫静态代码块，是在类中独立于成员的static语句块，可以有多个，位置可以随便放，它不在任何的方法体内，JVM加载类时会执行些静态的代码块，如果static代码块有多个，JVM将按照它们在类中出现的先后顺序依次执行它们，个代码块只会被执行一次。大多数用来做初始化。
<h3>static和jvm：</h3>
     1、非静态方法有一个隐含的传入参数，该参是JVM给它的，和我们怎么写代码无关，这个隐含的参数就是对象实例在stack中的地址指针。因此静态方法（在stack中的指令代码）总是可以找到自己的专用数据（在heap 中的对象属性值）。当然非静态方法也必须获得隐含参数，因此非静态方法在调用前，必须先new一个对象实例，获得stack中的地址指针，否则JV将无法将隐含参数传给非静态方法。

     2、而静态方法无此隐含参数，因此也不需要new对象，只要class文件被ClassLoader load进入JVM stack，该静态方法即可被调用。当然此时静态方法是存取不到<spn>heap 中的对象属性的。 </>
    3、前面提到对象实例以及动态属性都是保存heap 中的，而heap 必须通过stack中的地址指才能够被指令（类的方法）访问到。因此可以推断出：静态属性是保存在stack中的，而不同于动态性保存在heap 中。正为都是在stack中，而stack中指令和数据都是定长的，因此很容易算出偏移量，也因此不管什么指令类的方法），都可以访问到类的静态属性。也正因为静态属性被保存在stack中，所以具有了全局属。

   总结一下：静态属性保存在stack指令内

区，动态属性保存在heap 数据内存区
这一小段整理于 http://blog.csdn.net/itm_hadf/article/details/7383241&nbs;精华
<h3>疑问及猜测：</h3>
       猜测：static修饰的保存在stack指令内存区，在类加载到内存中即可进行调用，非static的属性或法，是动态的，用到的时候需要new，如果在静态方法中使用this，或者super，则会编译不过。</spn>
       疑问：坐等@一路向北 ,@网警110,@clong@朱@流浪中の狮子@臭猫男等等来做详细解答，排名不分先后</spa>

原文链接：java 关键字 static 的一点理解