从JVM运行角解释，为什么类的成员变量有默认初始值，而方法内的局部变量却没有初始值

导读：此问题涉及到JVM类加载和字节码执行的两个阶段，这两个阶段是依次执行的。JVM类加载是JVM利用类加载器将class文件加载到JVM的过程，涉及“加载”、“验证”、“准备”、“解析”和“初始化”。

1. 类的成员变量初始化 ---> 在JVM类加载阶段完成

类的成员变量又分为静态成员变量和非静态成员变量。

**静态成员变量**

静态成员变量会被初始化两次，第一次在“准备”阶段（也可以理解为“编译”阶段），先进行一次初始化，系统附上默认值；第二次在“初始化”阶段，根据代码中的赋值情况再进行一次初始化。

例如：

public static int i = 3;

第一次初始化后i的值为0，第二次初始化后，值猜为3.

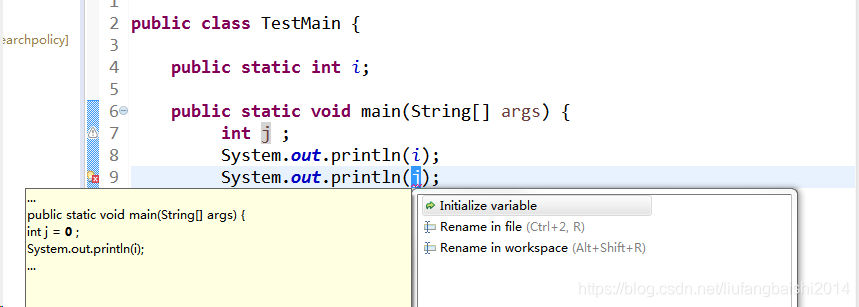
数据最终存放在方法区中。

**非静态成员变量**

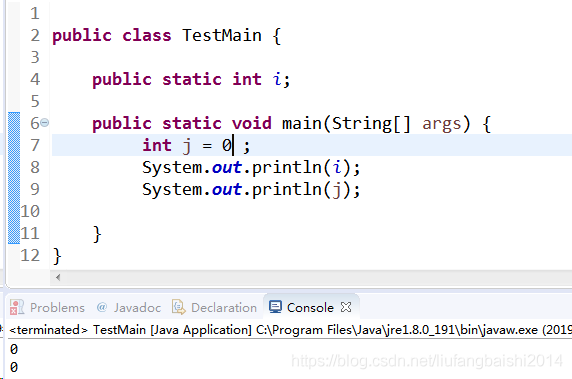
仅“初始化”阶段赋值。根据代码中的赋值情况，代码不赋值直接赋默认值，有赋值则等于代码中的赋值。对象实例化后，该变量随java对象分配到java堆中。

1. 方法区的局部变量没有初始化

而方法中的局部变量没有初始化步骤



需要在代码中进行初始化



总结：在类方法中的代码，是在字节码执行的时候，才会被运行，此时局部变量是存储在虚拟机栈 - 栈帧中的局部变量表中。局部变量定义了但没有赋值是不能使用的。

代码块

@Test  
 */\*\*  
 类方法中的成员变量默认值  
  
 总结：  
 1. 当类中成员是封装类/包装类类型时，它们的默认值都为null  
 2. 当类中成员是基本数据类型时，它们的默认值如下  
 short 、int、long ---> 0  
 float、double ---> 0.0  
 char ---> 什么都不输出(看起来)  
 boolean ---> false  
 \*/* **public void** test3(){  
 JVMTest jvmTest = **new** JVMTest();  
  
 System.*out*.println(jvmTest);  
  
 System.*out*.println(jvmTest.aboolean); // false  
 System.*out*.println(jvmTest.aBoolean); // null  
 System.*out*.println(jvmTest.abyte); // 0  
 System.*out*.println(jvmTest.aByte); // null  
 System.*out*.println(jvmTest.achar); // 0x0000 ---> 什么都不输出  
 System.*out*.println(jvmTest.aChar); // null  
 System.*out*.println(jvmTest.adouble); // 0.0  
 System.*out*.println(jvmTest.aDouble); // null  
 System.*out*.println(jvmTest.afloat); // 0.0  
 System.*out*.println(jvmTest.aFloat); // null  
 System.*out*.println(jvmTest.aint); // 0  
 System.*out*.println(jvmTest.aInteger); // null  
 System.*out*.println(jvmTest.along); // 0  
 System.*out*.println(jvmTest.aLong); // null  
 System.*out*.println(jvmTest.ashort); // 0  
 System.*out*.println(jvmTest.aShort); // null  
  
 }  
}  
  
**class** JVMTest{  
 **byte** abyte;  
 Byte aByte;  
 **short** ashort;  
 Short aShort;  
 **int** aint ;  
 Integer aInteger;  
 **long** along;  
 Long aLong;  
 **double** adouble;  
 Double aDouble;  
 **float** afloat;  
 Float aFloat;  
 **char** achar;  
 Character aChar;  
 **boolean** aboolean;  
 Boolean aBoolean;  
}