栈：存放多个栈帧。一个main线程中，每个方法都有一个栈帧。

栈帧：存放线程的局部变量，操作数栈，动态链接，方法出口

局部变量：方法类的局部变量；

操作数栈：记录了一个方法执行过程中的字节码指令，他往操作数栈中进行入栈和出栈；

动态链接：**每一个栈帧**都包含一个**指向运行时常量池**中该栈帧所属的**方法的引用**，持有这个引用是为了支持方法调用过程中的**动态连接。**如果你看了[字节码文件构成](https://github.com/leosanqing/Java-Notes/blob/master/JVM/%E8%99%9A%E6%8B%9F%E6%9C%BA%E6%89%A7%E8%A1%8C%E5%AD%90%E7%B3%BB%E7%BB%9F/%E5%AD%97%E8%8A%82%E7%A0%81%E6%96%87%E4%BB%B6%E7%BB%93%E6%9E%84/%E8%83%BD%E7%9C%8B%E6%87%82%E7%9A%84%E5%AD%97%E8%8A%82%E7%A0%81-%E4%B8%8A.md)和[类加载过程](https://github.com/leosanqing/Java-Notes/blob/master/JVM/%E8%99%9A%E6%8B%9F%E6%9C%BA%E6%89%A7%E8%A1%8C%E5%AD%90%E7%B3%BB%E7%BB%9F/%E7%B1%BB%E5%8A%A0%E8%BD%BD%E8%BF%87%E7%A8%8B/%E7%B1%BB%E5%8A%A0%E8%BD%BD.md)，你应该知道，字节码文件中有很多符号引用。

这些符号引用一部分会在**类加载的解析阶段**或者**第一次使用的时候**转化为直接引用，这种转化称为**静态解析；**

方法出口：即返回地址。一个方法执行后，**只有**两种方法可以退出：

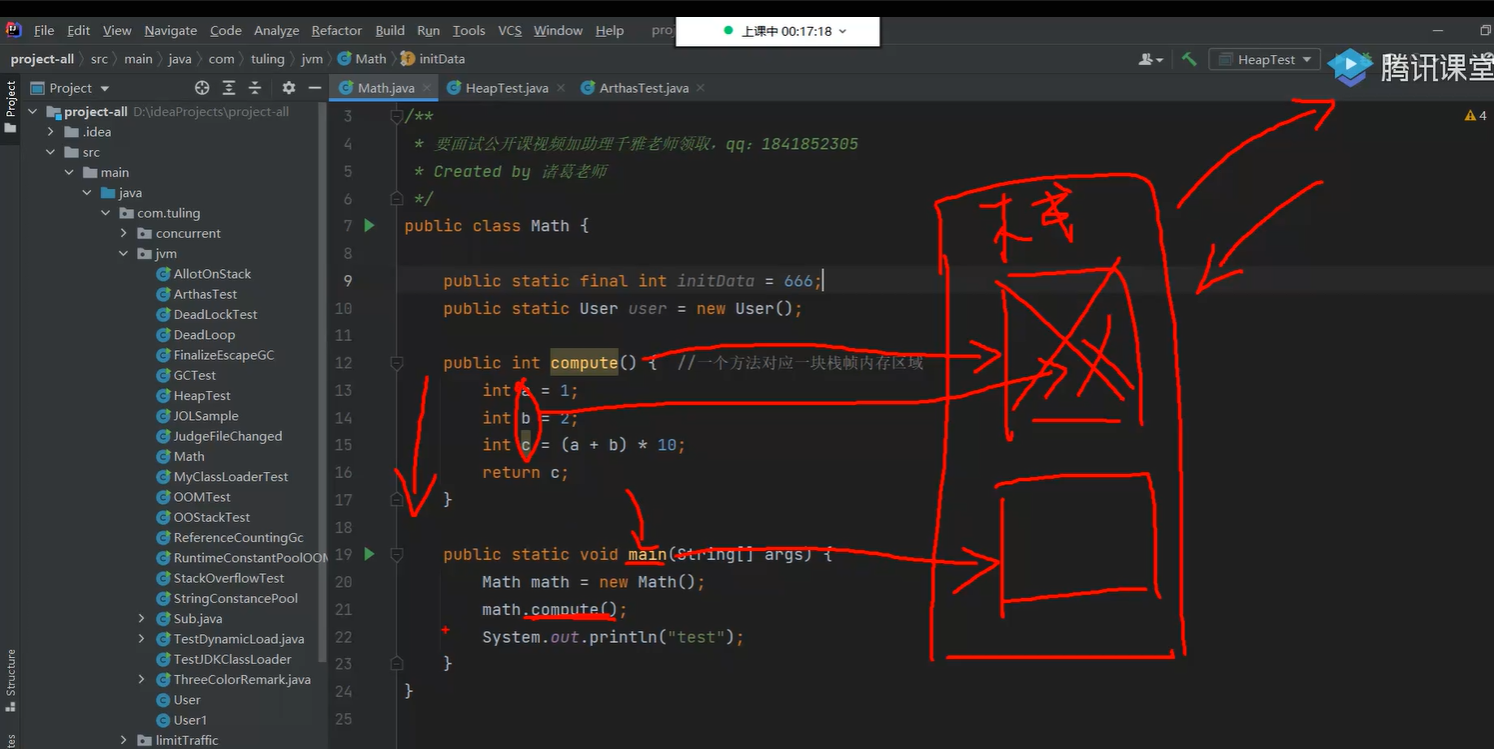
* return，正常退出
* 异常，并且不在该方法中处理

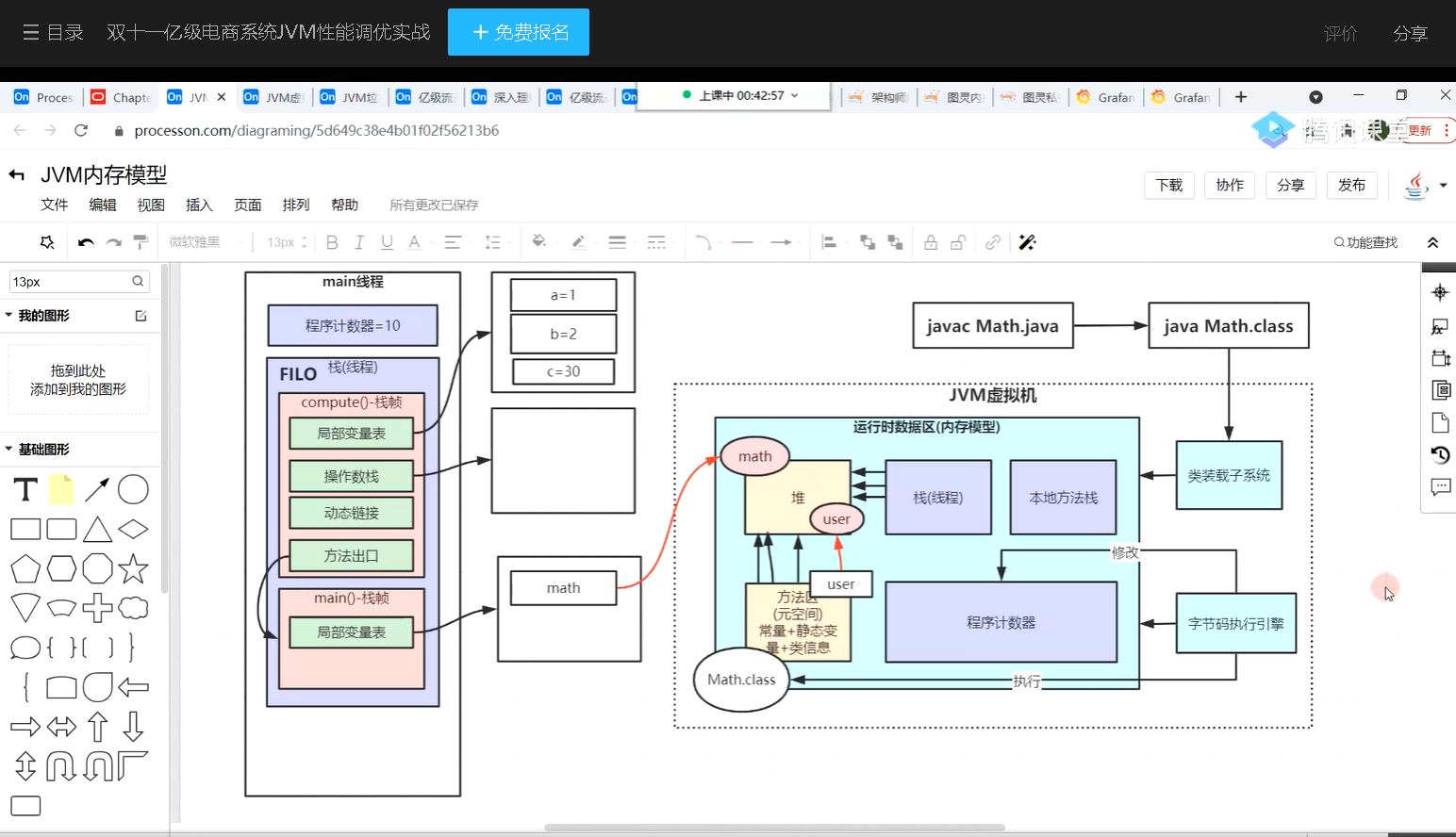
方法退出时可能的操作：

* 恢复上层方法的局部变量表和操作数栈
* 把返回值(如果有的话)压入调用者战争的操作数栈中
* 恢复PC计数器的值，以指向方法调用指令后面的一条指令

另一部分会在**运行期间**转化为直接引用，这部分称为**动态连接**

栈帧：线程里每个方法分配的内存空间





垃圾回收：

