JVM内存：

大多数 JVM 将内存区域划分为 **Method Area（Non-Heap）（方法区）** ,**Heap（堆）** , **Program Counter Register（程序计数器）** ,   **VM Stack（虚拟机栈，也有翻译成JAVA 方法栈的）,Native Method Stack**  （ **本地方法栈** ），其中**Method Area** 和 **Heap**是线程共享的 **，VM Stack，Native Method Stack  和Program Counter Register**是非线程共享的。为什么分为 线程共享和非线程共享的呢?请继续往下看。

首先我们熟悉一下一个一般性的 Java 程序的工作过程。一个 Java 源程序文件，会被编译为字节码文件（以 class 为扩展名），每个java程序都需要运行在自己的JVM上，然后告知 JVM 程序的运行入口，再被 JVM 通过字节码解释器加载运行。那么程序开始运行后，都是如何涉及到各内存区域的呢？

概括地说来，JVM初始运行的时候都会分配好 **Method Area（方法区）** 和**Heap（堆）** ，而JVM 每遇到一个线程，就为其分配一个 **Program Counter Register（程序计数器）** ,   **VM Stack（虚拟机栈）和Native Method Stack  （本地方法栈），**当线程终止时，三者（虚拟机栈，本地方法栈和程序计数器）所占用的内存空间也会被释放掉。这也是为什么我把内存区域分为线程共享和非线程共享的原因，非线程共享的那三个区域的生命周期与所属线程相同，而线程共享的区域与JAVA程序运行的生命周期相同，所以这也是系统垃圾回收的场所只发生在线程共享的区域（实际上对大部分虚拟机来说知发生在Heap上）的原因。

Java特点：

简单性,简化了语法 取消了指针，而且有JVM（java虚拟机）操控内存，不用程序员自己操控，而java程序也是运行在虚拟机上的。

健壮性（GC垃圾回收机制）

面向对象，

安全的；

跨平台：java的跨平台是每个操作系统都可以安装对应的java虚拟机所以java程序可以运行在不同系统上

JDK(java开发工具包) JRE(java运行环境) JVM(java虚拟机)

配置环境变量：

一，path环境变量（必须）

计算机属性—系统属性—高级—环境变量。找到path变量，双击打开然后把jdk中的bin路径配置到path中，

配置好之后可以再 cmd中输入java –version 来检测

（如果想在DOS命令窗口中启动一个程序的时候就把程序的目录配置到path里面）

（神操作！！打开计算机—打开一些文件夹—在地址栏上输入cmd回车。会直接定位到当前文件夹路径）

二，JAVA\_HOME环境变量（看实际），如果需要安装Tomcat服务器就必须配置这个环境变量

三，classpath环境变量(java特有，不重要，不用配) 作用是可以让类加载器去指定的文件夹去找字节码文件

该环境变量需要新建 注意：变量名不能随意写必须交classpath

配置了该环境变量之后类加载器只会在配置的文件夹下找

Java加载和执行：java源文件是.java文件，经过编译器(javac.exe)编译成.class文件，必须是符合java语法规则才能编译通过。不符合规则编辑就不通过不会生成.class文件。

1. 先启动jvm
2. Jvm启动类加载器 去找硬盘上”类”对应的”字节码”文件
3. 找到了就会执行 找不到会报错

DOS窗口：编译java文件 用 javac java文件路径(路径分为相对路径和绝对路径，相对路径对应在当前盘符下可以使用，而绝对路径是在别的盘符下) 生成.class类文件

运行java文件 先到类文件所在的路径，然后用java 类名（java后面不能跟路径）

Java新特性 可以直接 java java源文件路径，原理一样，先编译后运行。

Main方法中的args参数可以改

Java源文件里面可以定义多个class。每个class都可以有main方法。但实际上用一个就

可以了。编译出来的是多个class文件，可以不加public；加public的类只能在同名的java

源文件里面声明

在java文件中，自己命名就是标识符，标识符命名规则：

1. 只能用数字，字母，下划线和美元符号组成
2. 不能以数字开头
3. 关键字不能做标识符
4. 严格区分大小写 大写和小写不一样
5. 没长度限制

（时刻记录编译和运行的错误，这样在犯错的时候可以有参考）

类名所有的单词首字母都是大写

方法名第一个单词首字母小写，其余的是大写（驼峰命名）

命名规范：

1. 建名知意（看到名字就知道这个方法或者类是干嘛的）
2. 驼峰
3. 类名和接口名首字母大写，后面每个单词首字母都大写
4. 变量名和方法名首字母小写，后面每个单词首字母都大写
5. 常量名全大写，单词名用下划线区分

61