

- 一、内存是什么？
- 二、java语言特性
 - 1.简单性
 - 2.面向对象
 - 3.健壮性
 - 4.完全支持多线程并发
 - 5.可移植性/跨平台
- 三、Java的加载与执行
 - Java非常重要的两个阶段：
 - Java程序从开发到运行经历的过程：
 - Java HelloWorld的执行过程和原理
 - 文件指令
- 四、重点术语总结
 - 1.Java体系的技术被划分为三大块
 - 2.安装JDK之后
 - 3.JDK、JRE、JVM的关系
- 五、问题汇总
 - 1.去外地跑项目，需要安装JDK么？
 - 2.为什么安装JDK自带JRE？
 - 3.编译阶段和运行阶段可以在不同操作系统上完成么？
 - 4.Java代码一旦编译，源代码可以删除么？只留下字节码可以么？
 - 5.字节码是二进制文件么？
 - 6.path环境变量的作用是什么？
 - 7.path环境变量时Java中的机制还是windows操作系统中的机制？path环境变量时隶属于Java的么？path环境变量和Java有关么？

一、内存是什么？

对于计算机来说，

CPU：中央处理器，负责发送并执行指令，负责计算和运算

内存：程序运行过程中当中的临时数据存储空间，断电之后或者关机之后 内存的数据就消失了。
cpu计算过程中的临时存储空间

硬盘：持久化硬盘，硬盘上的数据不会断电而丢失

主板：相当于人类的躯干，是一个载体：将电脑的CPU、内存条、硬盘等连接起来

二、java语言特性

1.简单性

在java语言当中**真正操作内存**的是：**JVM(java虚拟机)**

所有Java程序都是运行在java虚拟机当中的

对于c或者c++来说程序员都是直接**通过指针操作内存**的

C或者C++更灵活，可以直接程序员操作内存，要求技术更精湛，有更驾驭感

java语言屏蔽了指针概念，不能直接操作指针，或者程序员不能直接操作内存

优点：不容易导致内存泄漏

缺点：效率问题较低

java语言层是C++,所以JVM使用C++语言写好的一个虚拟电脑，**安装JDK后，JVM就代表安装好了。**

2.面向对象

java是完全面向对象，更容易让人理解，采用面向对象的方式可以让复杂问题简单化

3.健壮性

主要是因为java有一种机制：**自动垃圾回收机制**（GC机制）

java不容易导致内存的泄漏，比C具有健壮性

C或者C++使用不当，易导致内存泄漏，没有健壮性

JVM负责调度GC机制。程序员不需要干扰

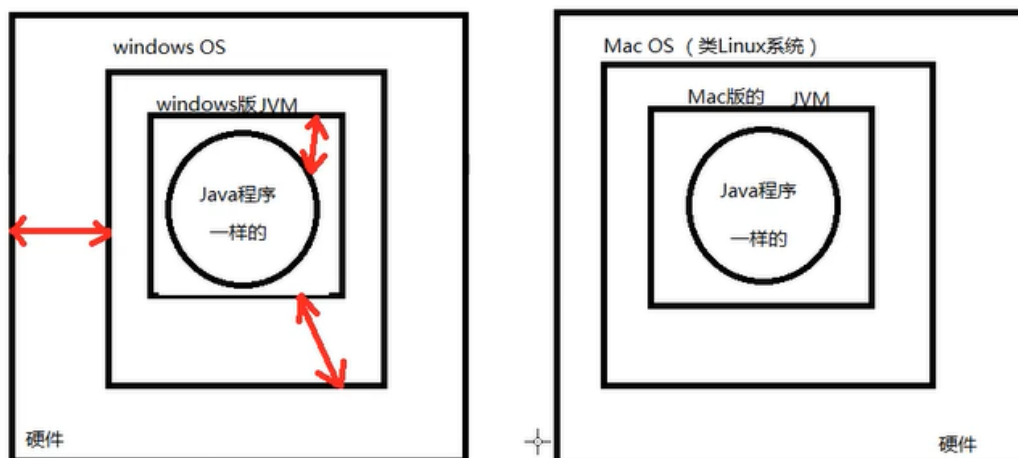
4.完全支持多线程并发

5.可移植性/跨平台

一次编写，到处运行！

例如：java编写完成后，可以运行在windows操作系统上，不需要做任何改动可以直接运行在Linux操作系统上，同样也可以运行到Mac os上面

JVM：Java虚拟机，用C++语言实现的虚拟的计算机。java程序实际上并没有直接和操作系统打交道。



JVM屏蔽了操作系统之间的差异

优点：一次到处运行，可以跨平台

缺点：麻烦，对于运行java程序来水必须先有JVM

三、Java的加载与执行

Java非常重要的两个阶段：

- 编译阶段

java程序员直接编写的java代码（普通文本）是无法执行被JVM识别的。Java代码必须经过一个编译，将这个普通文本变成**字节码**的过程，称为：编译。

Java代码被称为Java源代码，编译生成JVM可以识别的字节码。

- 运行阶段

Java程序从开发到运行经历的过程：

第一步：在硬盘的某个位置（随意），新建一个xxx.java文件

第二步：使用记事本或者其它文本编辑器例如EditPlus打开xxx.java文件

第三步：在xxx.java文件中编写“符合java语法规则的”源代码。

第四步：保存一定要将xxx.java文件保存一下

第五步：使用编译器(javac【JDK安装后自带】)对xxx.java文件进行编译。

第六步：如果xxx.java文件中编写的源代码是符合语法规则的，编译通过，如果xxx.java文件中编写的源代码违背了语法规则，那么编译器会报错，编译器报错之后class文件是不会生成的，只有编译通过了才会生成class字节码文件。并且一个java源文件是可以生成多个class文件的（编译实质上是检查语法）

运行期(JRE在起作用)：

第七步：如果是在Linux上运行，需要将windows上生成的class文件拷贝过去不需要拷贝源代码，真正运行的是字节码。

第八步：使用JDK自带的一个命令/工具：java（负责运行的命令/工具）执行字节码

第九步：往下的步骤就全部交给JVM了，就不需要程序员干涉了。JVM会将字节码文件装载进去，然后JVM对字节码进行解释（解释器负责将字节码解释为1010101010..等的二进制）

第十步：JVM会将生成的二进制码交给os操作系统，操作系统会执行二进制码和硬件进行交互

Java HelloWorld的执行过程和原理

java命令切记只能跟类名，带路径的都不对

第一步：会先启动JVM

第二步：JVM启动之后，JVM会去启动“类加载器classloader”

类加载器作用：加载类。本质上类加载器会负责去硬盘上找“类”对应的“字节码”文件

第三步：类加载器如果在硬盘上找不到会报错：找不到或无法加载主类

类加载器如果在硬盘上找到了对应的字节码文件，类加载器会将该字节码文件装载到JVM中，JVM启动“解释器”将字节码解释为“101010000....”这种二进制码，操作系统执行二进制码和硬件交互

文件指令

classpath是一个环境变量，给类加载器指路的

不属于windows操作系统，隶属于Java，是Java特有的，需要新建

Java文件后缀：.java

字节码文件后缀：.class

JDK的bin目录文件下：

javac命令：负责编译

java命令：负责运行

注意：一个Java文件可以生成多个class文件，最终运行的是class文件

四、重点术语总结

1.Java体系的技术被划分为三大块

JavaSE：标准版

JavaEE：企业版

JavaME：微型版

2.安装JDK之后

JDK： java开发工具箱

JRE： java运行环境

JVM： java虚拟机

3.JDK、JRE、JVM的关系

JRE包含JRE，JRE包含JVM

JRE和JDK都可以独立安装，都有独立的安装包。

安装**JDK**时，**JRE**就自动安装了，同时**JRE**内部的**JVM**也就自动安装了

安装**JRE**时，**JVM**也就自动安装了

JDK13安装的时候内置了一个**JRE**

JDK8安装时不仅内置了一个**JRE**，还在**JDK目录之外**独立生存一个单独的**JRE**

五、问题汇总

1.去外地跑项目，需要安装JDK么？

不需要跑项目只需要安装**JRE**就行，体积小，安装非常便捷快速

2.为什么安装JDK自带JRE？

因为java程序员开发完程序的时候，要测试这个程序，让程序运行，需要**JRE**

3.编译阶段和运行阶段可以在不同操作系统上完成么？

在windows上编译，编译生成之后生成字节码，把字节码放到linux上运行完全可以，因为Java是跨平台的

4.Java代码一旦编译，源代码可以删除么？只留下字节码可以么？

完全可以执行，因为源代码不参与程序的执行过程，参与程序执行过程的是字节码，但是最好不要删除源代码，因为如果执行结果是错误的，可能需要重新修改

5.字节码是二进制文件么？

不是，如果是二进制文件，就不需要JVM了。因为操作系统可以直接执行二进制

6.path环境变量的作用是什么？

path环境变量的作用就是给windows操作系统指路的

告诉windows操作系统去哪里找这个命令文件

7.path环境变量时Java中的机制还是windows操作系统中的机制？ path环境变量时隶属于Java的么？ path环境变量和Java有关么？

path环境变量时隶属于windows的,Java只是用了一下而已