

# JAVA Day 01

## # JAVA 历史

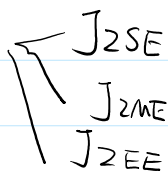
詹姆斯·高斯林

SUN 公司 → 1986 年

Oak (Java 前身)

JDK	1.0	Oak	96
	1.1		
	1.1.4	石头	97
	1.1.5	南瓜	97
	...	...	...
	1.1.8		1999
J2SE	1.2.1	无	1998
	...	...	...
	1.4.2	蜘蛛侠	2003
Java SE 5.0		老鹰	2004
Java SE 8.0		蜘蛛侠	2014
...	...	...	...
Java SE 12.0			2019

分类:



## # JRE & JDK \*

WHAT IS :

JRE: (Java Runtime Environment)

包括 JAVA 虚拟机(JVM) 和 Java 程序所需的核心类库等, 如果想要运行一个开发好的 Java 程序, 计算机中只需要安装 JRE 即可。

JRE = JVM + 类库.

JDK (Java Development Kit):

JDK 是提供给 Java 开发人员使用的, 其中包含了 Java 的开发工具, 也包括了 JRE 所以安装了 JDK, 就不用再单独安装 JRE 了。

其中的开发工具: (javac) 打包工具 (jar) 等。

JDK = JRE + JAVA 的开发工具。

# 安装。

JAVA\_HOME:

PATH:

CLASSPATH:

# JDK 目录:

/bin	开发工具
/include	头文件
/db:	自带的小型数据库
/lib:	类库
/jre:	java 运行环境
/src.zip:	java 源代码

# Java 是啥子

- 开发语言 ..
- 编程环境 ..
- 应用环境 ..
- 部署环境 ..

并 Java 有啥子特点:

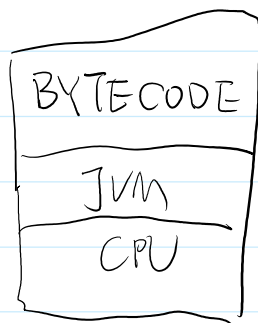
- 开源
- 简单易用
- 无指针, 不需管理内存 (GC)
- 跨平台 (JVM) (字节码文件)
- 支持多线程 (提高开发效率)
- 动态更新
- 面向对象编程
- 安全 (代码安全校验器)

动态

## 7. JVM

Java Virtual Machine.

Java 虚拟机



是一个虚构出来的计算机, 是通过在实际的计算机上仿真模拟各种计算机功能来实现的。

Java 语言使用 Java 虚拟机屏蔽了与具体平台相关的信息, 使得 Java 语言编译程序只需生成在 Java 虚拟机上运行的目标代码 (字节码), 就可在多种平台上不加修改地运行。

Java 虚拟机有较完善的硬件架构, 如处理器、堆栈、寄存器等, 还具有相应的指令系统。

Java 的跨平台, 是字节码跨平台, JVM 不跨平台。

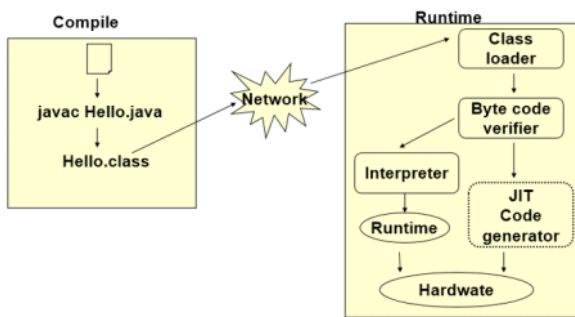
\* GC (垃圾回收器)

Java 虚拟机中, 存在自动内存管理和垃圾清扫机制。该机制对 JVM 中的内存进行标记, 并确定哪些内存需要回收, 根据一定的回收策略, 自动地回收内存, 永不倦怠地保证 JVM 中的内存空间, 防止出现内存泄露和溢出的问题。

Java GC 机制主要完成了件事:

- 确定哪些内存需要回收, ~~确定什么时候需要~~
- 确定什么时候需要执行 GC,
- 如何执行 GC

井 安全



字节码验证 BCU

代码符合 JVM 规范

不能破坏系统完整性

不能有栈溢出

数据类型要正确

类型转换要正确

JIT 即时编译器, 提高 Java 代码运行效率.

井 编程步骤 = 编

1. 编辑

2. 编译

3. 运行.

打包: package, 解决命名冲突问题, 分类管理

jar 打包.

jar -xvf my.jar

jar -cvf my.jar com

-cvf .class 打包.

-xvf -jar 解压

-tvf 看 jar 内容.

13. API: 类的使用帮助文档

包名 → 属性 方法

ZOF 2016005851