

实战JVM-基础篇



黑马程序员
www.itheima.com

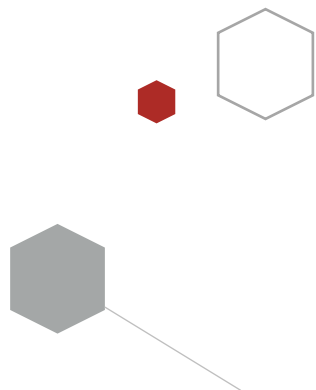
传智教育旗下
高端IT教育品牌



目录

Contents

- ◆ 初识 JVM
- ◆ 字节码文件详解
- ◆ JVM 的内存区域
- ◆ JVM 的垃圾回收



初识 JVM

- 什么是 JVM
- JVM 的功能
- 常见的 JVM

什么是JVM

JVM 全称是 Java Virtual Machine，中文译名 Java虚拟机。



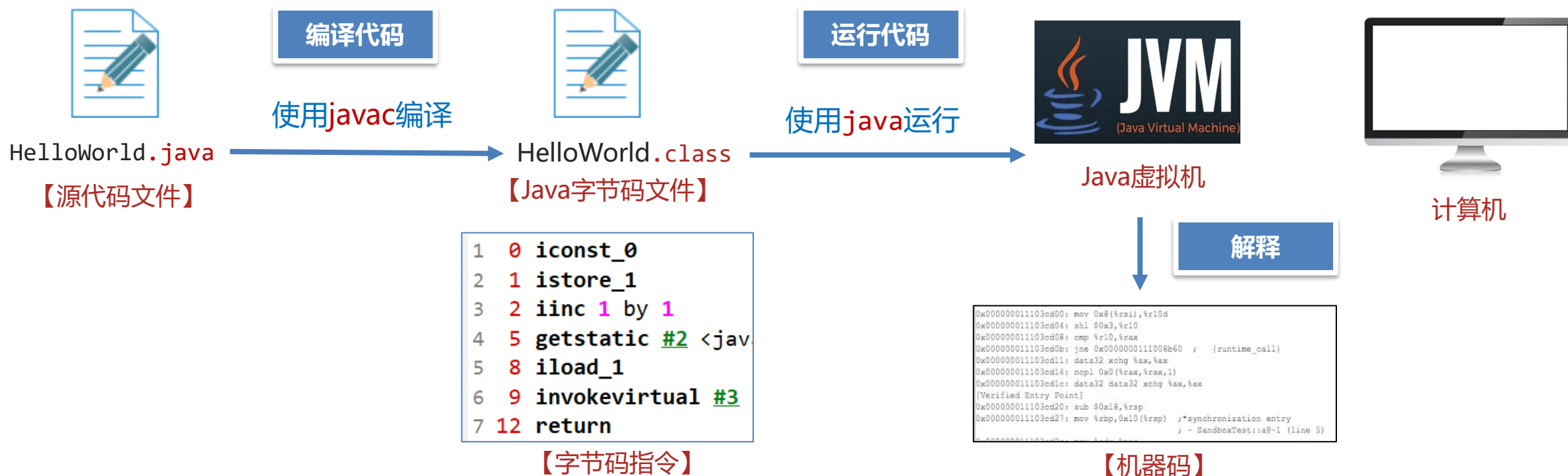
什么是JVM

JVM 本质上是一个运行在计算机上的程序，他的职责是运行Java字节码文件。



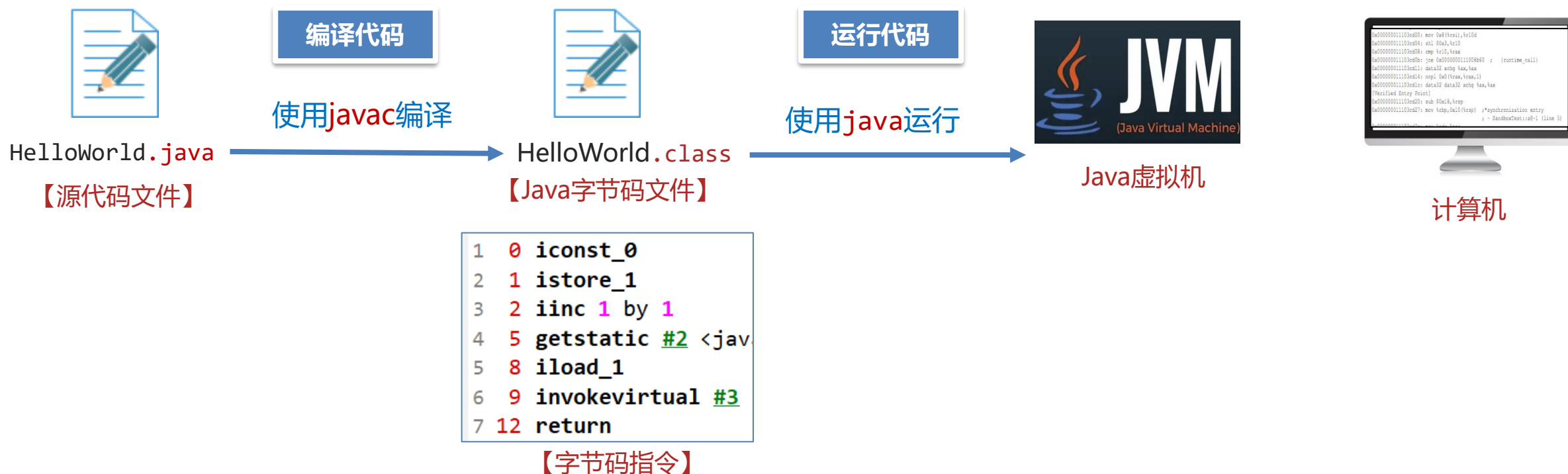
什么是JVM

JVM 本质上是一个运行在计算机上的程序，他的职责是运行Java字节码文件。



什么是JVM

JVM 本质上是一个运行在计算机上的程序，他的职责是运行Java字节码文件。



JVM的功能

01

解释和运行

- 对字节码文件中的指令，实时的解释成机器码，让计算机执行

02

内存管理

- 自动为对象、方法等分配内存
- 自动的垃圾回收机制，回收不再使用的对象

03

即时编译

- 对热点代码进行优化，提升执行效率

JVM的功能 - 即时编译

- Java语言如果不做任何优化，性能不如C、C++等语言。

Java语言

```
1 0 iconst_0
2 1 istore_1
3 2 iinc 1 by 1
4 5 getstatic #2 <java.lang.Object.class>
5 8 iload_1
6 9 invokevirtual #3 <java.lang.Object.toString()Ljava/lang/String;V>
7 12 return
```

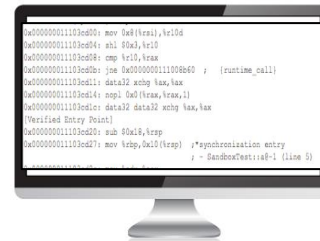
【字节码指令】

实时解释



```
0x00000000: mov 0x0(%rsi),%r10d
0x00000001: shl $0x3,%r10
0x00000002: cmp %r10,%rax
0x00000003: jne 0x0000000011008b60 ; {runtime_call}
0x00000004: data32 xchg %ax,%ax
0x00000005: nopl 0x0(%rax,%rax,1)
0x00000006: data32 data32 xchg %ax,%ax
[Verified Entry Point]
0x00000007: sub $0x18,%rsp
0x00000008: mov %rbp,0x10(%rsp) ;*synchronization entry
; - SandboxTest::a@-1 (line 5)
```

【机器码】



C/C++语言



main.c

【源代码文件】

编译和链接



main.exe

【可执行文件(机器码)】



JVM的功能 - 即时编译

- Java需要实时解释，主要是为了支持跨平台特性。

Java语言

```
1 0 iconst_0
2 1 istore_1
3 2 iinc 1 by 1
4 5 getstatic #2 <jav
5 8 iload_1
6 9 invokevirtual #3
7 12 return
```

【字节码指令】

实时解释



```
0x000000001103cd00: mov 0x8(%rsi),%r10d
0x000000001103cd04: shl $0x3,%r10
0x000000001103cd08: cmp %r10,%rax
0x000000001103cd0b: jne 0x0000000011008b60 ; {runtime_call}
0x000000001103cd11: data32 xchg %ax,%ax
0x000000001103cd14: nopl 0x0(%rax,%rax,1)
0x000000001103cd1c: data32 data32 xchg %ax,%ax
[Verified Entry Point]
0x000000001103cd20: sub $0x18,%rsp
0x000000001103cd27: mov %rbp,0x10(%rsp) ;*synchronization entry
; - SandboxTest::a@-1 (line 5)
```

【windows机器码】



实时解释



```
0x000000001103cd00: mov 0x8(%rsi),%r10d
0x000000001103cd04: shl $0x3,%r10
0x000000001103cd08: cmp %r10,%rax
0x000000001103cd0b: jne 0x0000000011008b60 ; {runtime_call}
0x000000001103cd11: data32 xchg %ax,%ax
0x000000001103cd14: nopl 0x0(%rax,%rax,1)
0x000000001103cd1c: data32 data32 xchg %ax,%ax
[Verified Entry Point]
0x000000001103cd20: sub $0x18,%rsp
0x000000001103cd27: mov %rbp,0x10(%rsp) ;*synchronization entry
; - SandboxTest::a@-1 (line 5)
```

【linux机器码】



JVM的功能 - 即时编译

- 由于JVM需要实时解释虚拟机指令，不做任何优化性能不如直接运行机器码的C、C++等语言。



C++



Java

JVM的功能 - 即时编译



HelloWorld.class
【Java字节码文件】

```
1 0 iconst_0
2 1 istore_1
3 2 iinc 1 by 1
4 5 getstatic #2 <java.lang.String
5 8 iload_1
6 9 invokevirtual #3 <java.lang.String
7 12 return
```

【热点代码字节码指令】

解释并优化

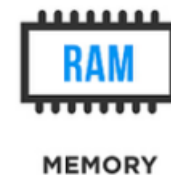
```
0x000000001103cd00: mov 0x8(%rsi),%r10d
0x000000001103cd04: shl $0x3,%r10
0x000000001103cd08: cmp %r10,%rax
0x000000001103cd0b: jne 0x0000000011008b60 ; {runtime_call}
0x000000001103cd11: data32 xchg %ax,%ax
0x000000001103cd14: nopl 0x0(%rax,%rax,1)
0x000000001103cd1c: data32 xchg %ax,%ax
[Verified Entry Point]
0x000000001103cd20: sub $0x18,%rap
0x000000001103cd27: mov %rbp,0x10(%rap) ; *synchronization entry
; - SandboxTest::a@-1 (line 5)
0x000000001103cd2e: mov %edx,%eax
```

【汇编和机器码】

保存至内存

再次执行

直接调用



内存

JVM的功能 - 即时编译

- JVM提供了**即时编译** (Just-In-Time 简称JIT) 进行性能的优化，最终能达到接近C、C++语言的运行性能

甚至在特定场景下实现超越。



C++



Java

JVM的功能

01

解释和运行

- 对字节码文件中的指令，实时的解释成机器码，让计算机执行

02

内存管理

- 自动为对象、方法等分配内存
- 自动的垃圾回收机制，回收不再使用的对象

03

即时编译

- 对热点代码进行优化，提升执行效率

常见的JVM

名称	作者	支持版本	社区活跃度 (github star)	特性	适用场景
HotSpot (Oracle JDK版)	Oracle	所有版本	高(闭源)	使用最广泛，稳定可靠，社区活跃 JIT支持 Oracle JDK默认虚拟机	默认
HotSpot (Open JDK版)	Oracle	所有版本	中(16.1k)	同上 开源，Open JDK默认虚拟机	默认 对JDK有二次开发需求
GraalVM	Oracle	11, 17, 19 企业版支持8	高 (18.7k)	多语言支持 高性能、JIT、AOT支持	微服务、云原生架构 需要多语言混合编程
Dragonwell JDK 龙井	Alibaba	标准版 8,11,17 扩展版11,17	低(3.9k)	基于OpenJDK的增强 高性能、bug修复、安全性提升 JWarmup、ElasticHeap、Wisp特性支持	电商、物流、金融领域 对性能要求比较高
Eclipse OpenJ9 (原 IBM J9)	IBM	8,11,17,19,20	低(3.1k)	高性能、可扩展 JIT、AOT特性支持	微服务、云原生架构

常见的JVM: Java虚拟机规范

- 《Java虚拟机规范》由Oracle制定，内容主要包含了Java虚拟机在设计和实现时需要遵守的规范，主要包含class字节码文件的定义、类和接口的加载和初始化、指令集等内容。
- 《Java虚拟机规范》是对虚拟机设计的要求，而不是对Java设计的要求，也就是说虚拟机可以运行在其他的语言比如Groovy、Scala生成的class字节码文件之上。
- 官网地址：<https://docs.oracle.com/javase/specs/index.html>

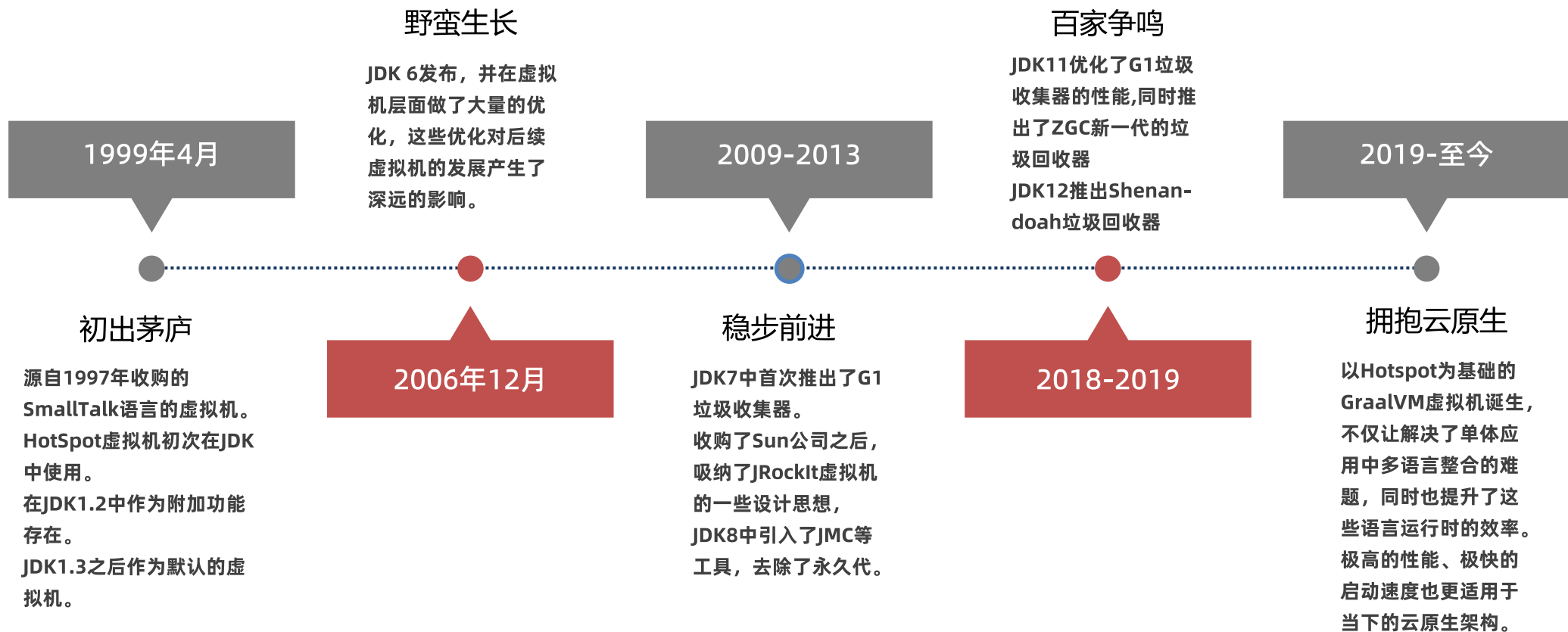
The Java® Virtual
Machine Specification
Java SE 8 Edition

Tim Lindholm
Frank Yellin
Gilad Bracha
Alex Buckley

常见的JVM

名称	作者	支持版本	社区活跃度 (github star)	特性	适用场景
HotSpot (Oracle JDK版)	Oracle	所有版本	高(闭源)	使用最广泛，稳定可靠，社区活跃 JIT支持 Oracle JDK默认虚拟机	默认
HotSpot (Open JDK版)	Oracle	所有版本	中(16.1k)	同上 开源，Open JDK默认虚拟机	默认 对JDK有二次开发需求
GraalVM	Oracle	11, 17, 19 企业版支持8	高 (18.7k)	多语言支持Ruby、Python、C++等 高性能、JIT、AOT支持	微服务、云原生架构 需要多语言混合编程
Dragonwell JDK 龙井	Alibaba	标准版 8, 11, 17 扩展版 11, 17	低(3.9k)	基于OpenJDK的增强 高性能、bug修复、安全性提升 JWarmup、ElasticHeap、Wisp特性支持	电商、物流、金融领域 对性能要求比较高
Eclipse OpenJ9 (原 IBM J9)	IBM	8, 11, 17, 19, 20	低(3.1k)	高性能、可扩展 JIT、AOT特性支持	微服务、云原生架构

常见的JVM - HotSpot的发展历程





总结

1、JVM到底是什么？

JVM 全称是 **Java Virtual Machine**，中文译名 **Java虚拟机**，是一个运行在计算机上的程序，他的职责是运行**Java字节码文件**。

2、JVM的三大核心功能是什么？

JVM 包含内存管理、解释执行虚拟机指令、即时编译三大功能。

3、常见的JVM虚拟机有哪些？

常见的JVM有HotSpot、GraalVM、OpenJ9等，另外DragonWell龙井JDK也提供了一款功能增强版的JVM。其中使用最广泛的是HotSpot虚拟机。



传智教育旗下高端IT教育品牌