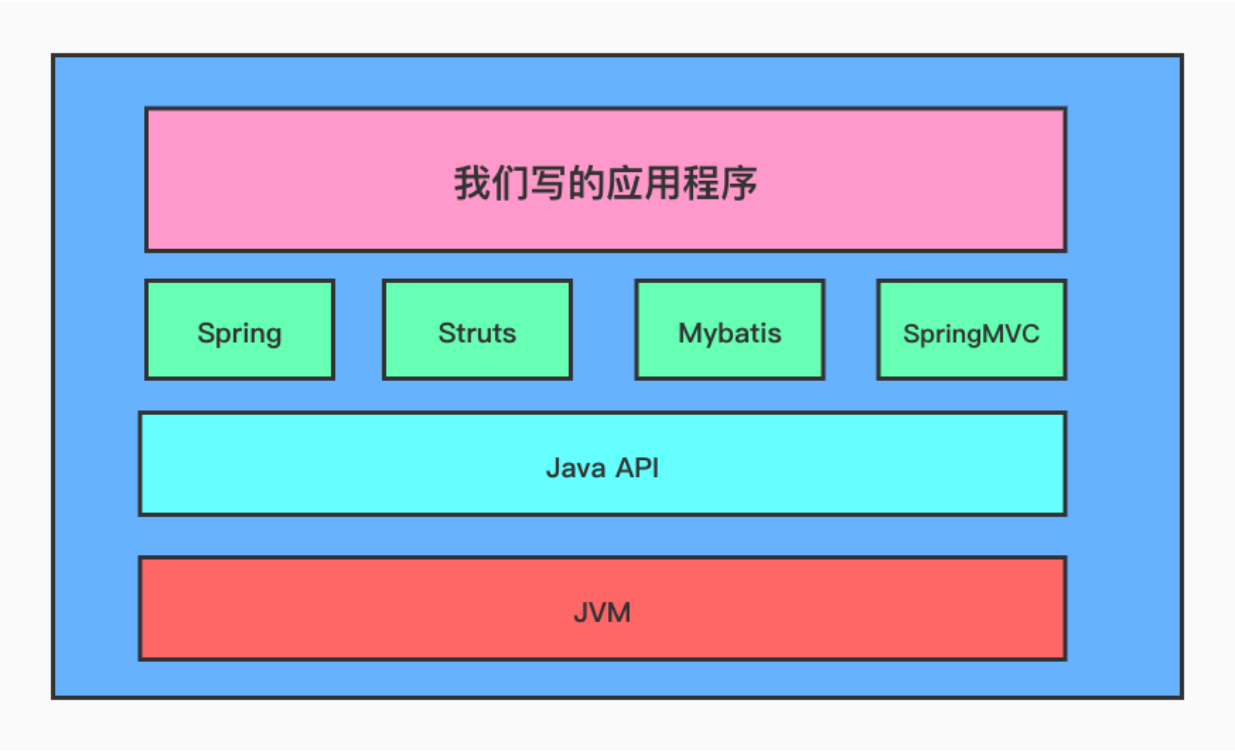
Chapter1 JVM与Java体系结构

1.前言

核心类库的API比作数学公式的话，那么Java虚拟机的知识点就是公式的推导过程。

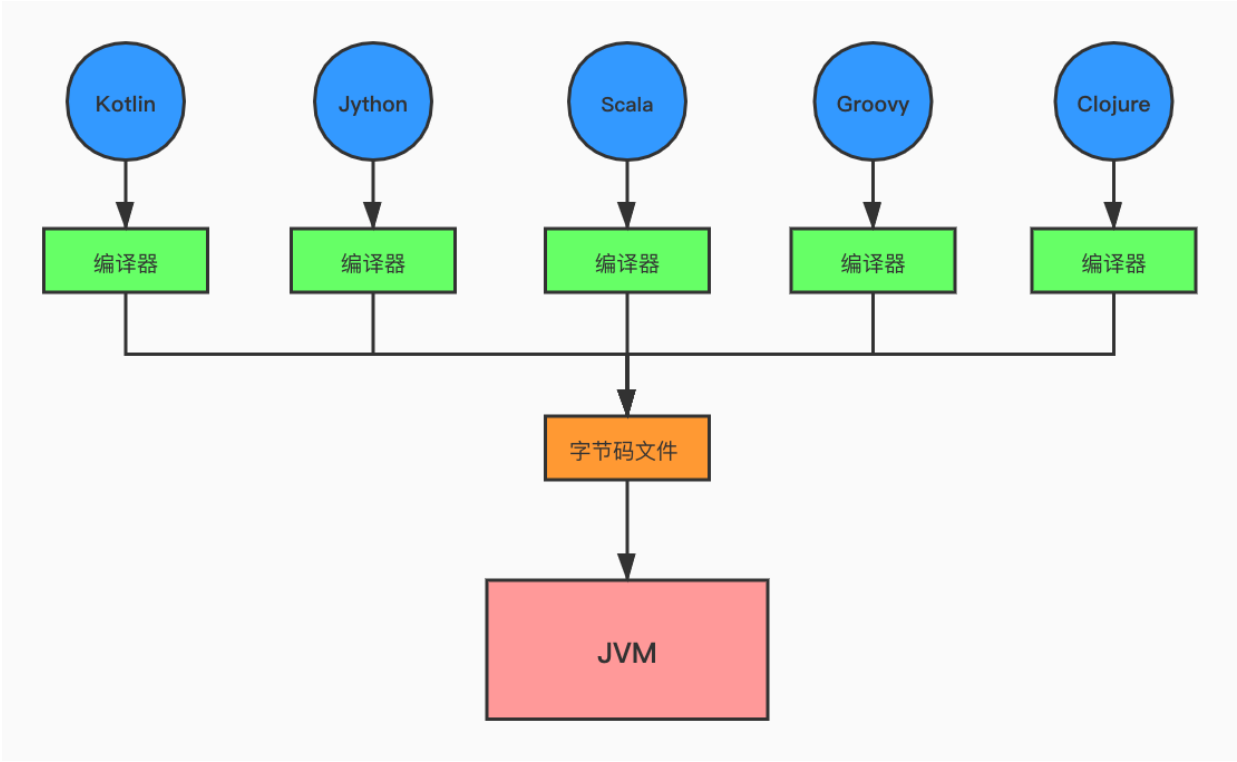


2.Java VS C++

C系语言需要自己来分配内存和回收内存，Java全部交给JVM进行分配和回收。

3.Java生态圈

随着Java7的发布，Java虚拟机的设计者们通过JSR-292规范基本实现了Java虚拟机平台上运行非Java语言编编写的程序。JVM并不会单纯与Java语言终身绑定，只要编程语言的编译结果满足JVM虚拟机的内部指令集、符号表以及其他的辅助信息，它就是有效地字节码文件，能够被虚拟机所识别被装载运行。



4.虚拟机与Java虚拟机

- 虚拟机

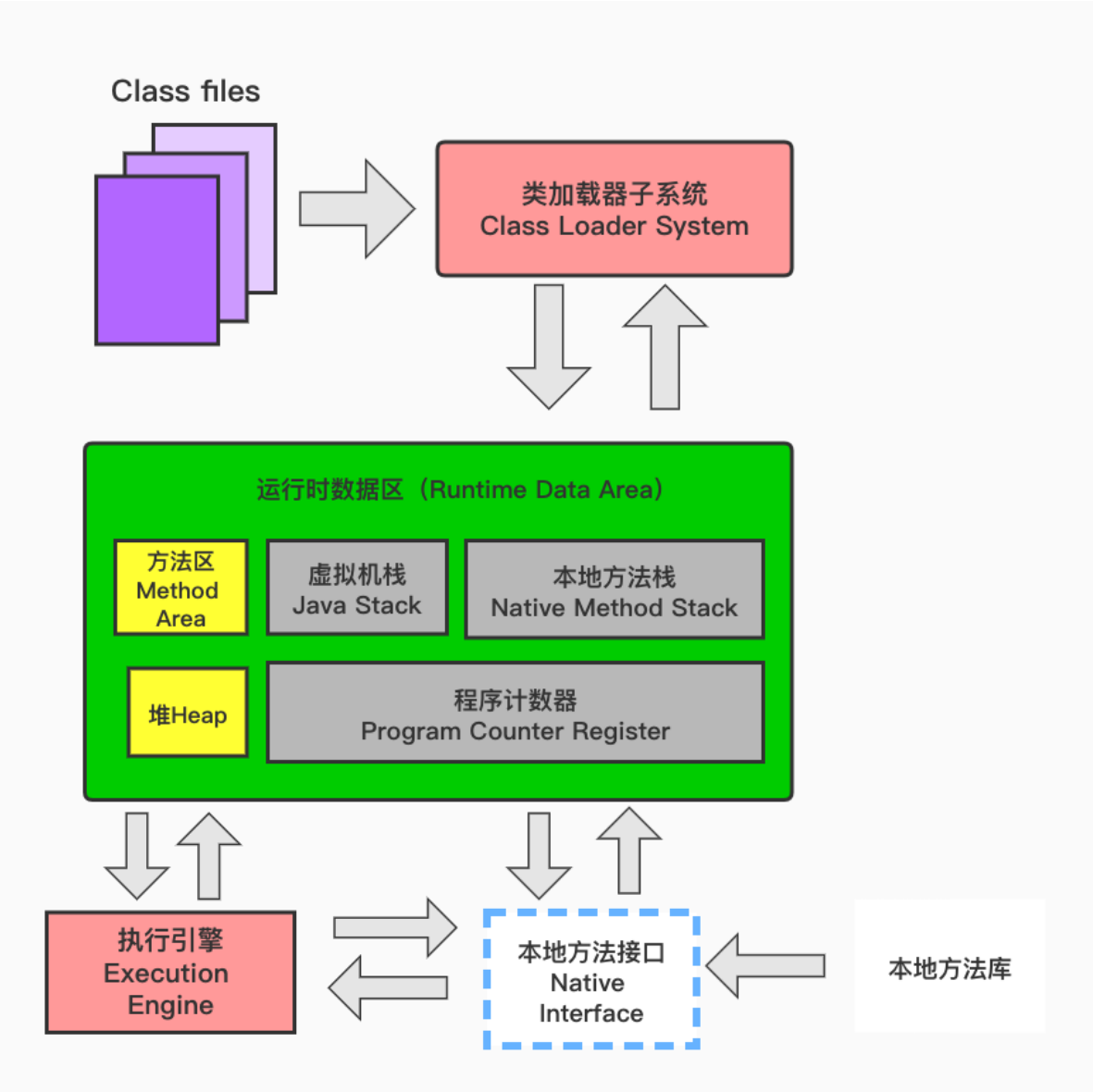
用软件来虚拟一台计算机。分为系统虚拟机和程序虚拟机。系统虚拟机如VMware，程序虚拟机如JVM。无论是系统虚拟机还是程序虚拟机，在上面运行的软件都被限制在虚拟机提供的资源中。

- Java虚拟机

跨平台性、优秀的GC垃圾回收器，以及可靠的即时编译器。

5.JVM整体结构

- HotSpot VM是市面上高性能虚拟机的代表作之一。
- 采用解释器与即时编译器并存的结构。
- 其中执行引擎包含：解释器、及时编译器、垃圾回收器。



6.JVM的架构模型

1. 基于栈的架构特点

> - 设计和实现更简单，适合资源受限的系统。
> - 避开寄存器分配难题：使用零地址指令分配。
> - 指令集中绝大多数是零地址指令，其执行过程依赖于操作栈，指令集更小，编译器更容易实现。
> - 不需要硬件实现，可移植性更好，利于跨平台。

2. 基于寄存器的架构的特点

- 典型的x86架构的二进制指令集：传统PC以及安卓的Davlik虚拟机。
- 指令集架构则完全依赖硬件，可移植性差。
- 性能好，执行高效。
- 花费更少的指令去完成一项操作。
- 绝大多数以一地址、二地址和三地址指令为主，而基于栈式架构的指令集却是以零地址为主。

7.JVM生命周期

1. 虚拟机的启动

   虚拟机的启动是通过引导类加载器(bootstrap class loader)创建一个初始类(initial class)来完成的，这个类是由虚拟机的具体实现指定的。

2. 虚拟机的执行

   一个运行中的Java虚拟机有着清晰任务，即执行Java程序。而程序开始执行时它才运行，程序结束它就停止。同时，执行一个所谓的java程序的时候，是执行一个叫做Java虚拟机的进程。

3. 虚拟机的退出

   有如下几种情况，会导致虚拟机退出

- 程序正常执行结束。
- 程序执行过程中遇到异常或错误而异常终止。
- 由于操作系统出现错误导致Java虚拟机进程终止。
- 某线程调用Runtime类或System类的exit方法，或者Runtime类的halt方法，并且Java安全管理器允许本次操作。
- 除此之外，JNI（Java Native Interface）规范描述了JNI Invocation API来加载或卸载Java虚拟机时，Java虚拟机退出的情况。