

JVM Basic CheatSheet

JVM Essentials

JVM interprets and compiles Java code

Acts as a bridge between code and hardware

Compilation

Java code compiles to bytecode

JVM interprets and executes bytecode

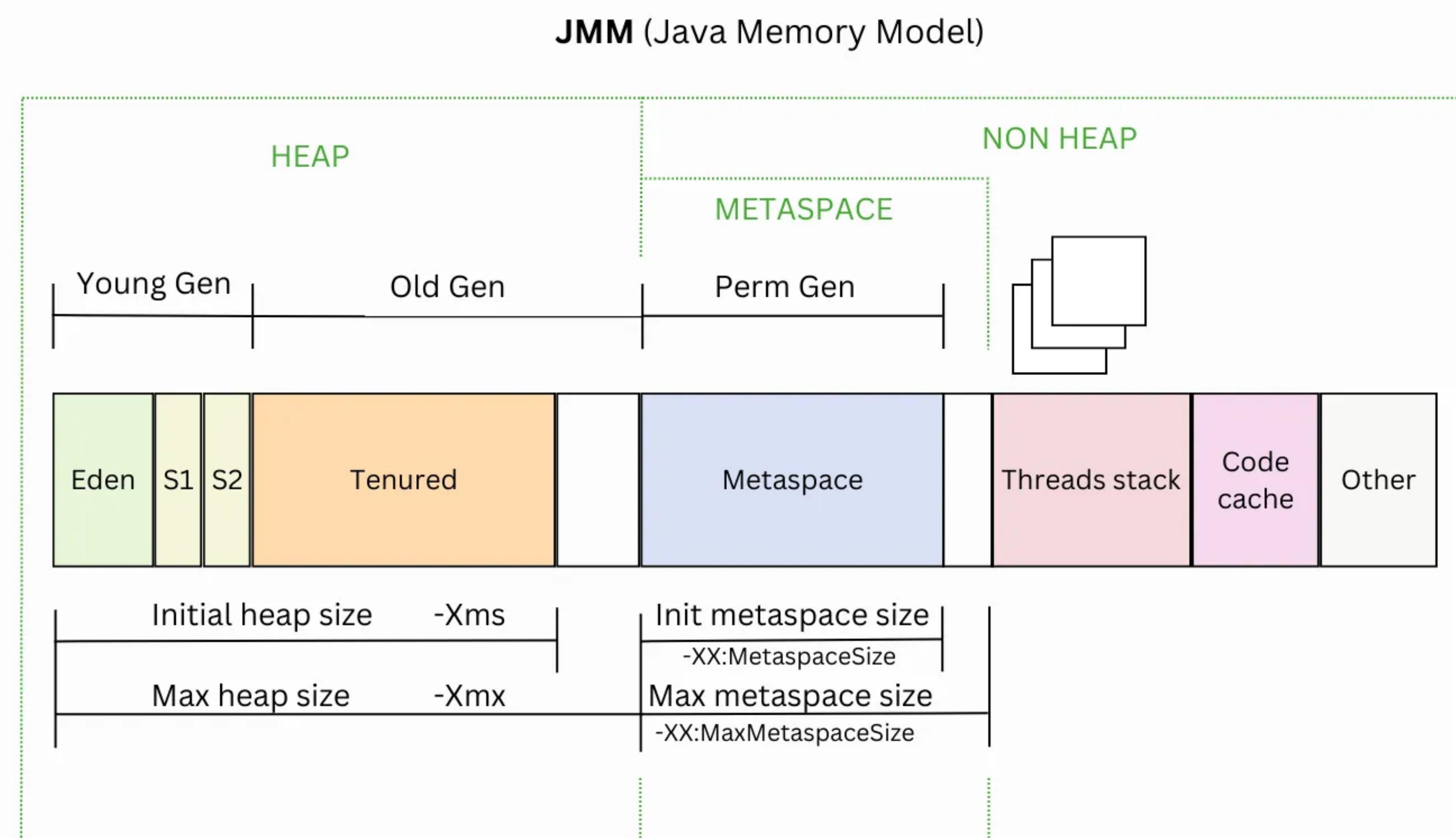
Memory Management

Method area stores class data

Heap holds objects (young and old generations)

Stacks manage method call info

Registers track executing instructions



Garbage Collection

Young gen: Short-lived objects

Old gen: Long-lived objects

JVM cleans memory through collections

Pre-G1 Garbage Collectors memory structure (**Serial, Parallel, CMS**)



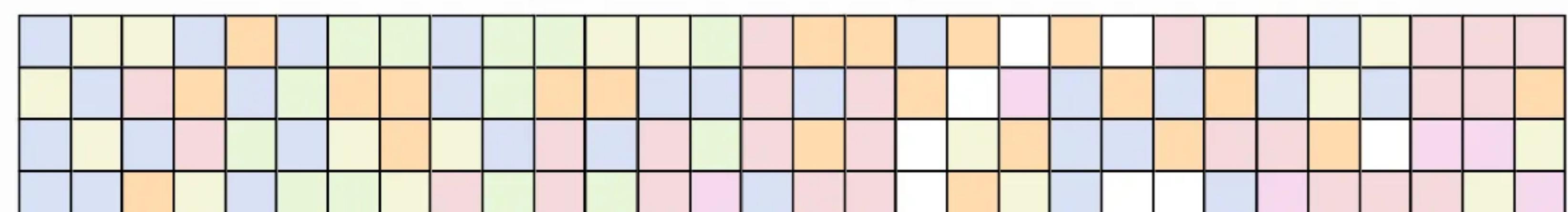
GC Algorithms

Serial: Single-threaded, simple

Parallel: Multi-threaded, high throughput

G1: Balanced approach, low pause times (default since Java 8)

G1 Garbage Collector memory structure
memory areas not grouped, cells and regions are allocated randomly



JIT Compilation

JIT compiles bytecode to native machine code

Speeds up execution

Languages on JVM

Kotlin: Concise, null safety

Scala: Object-oriented, functional

Groovy: Dynamic, Java-compatible

Performance Optimisation

Customise GC, profiling, JIT settings

Monitor memory, threads, CPU usage

Tuning Commands

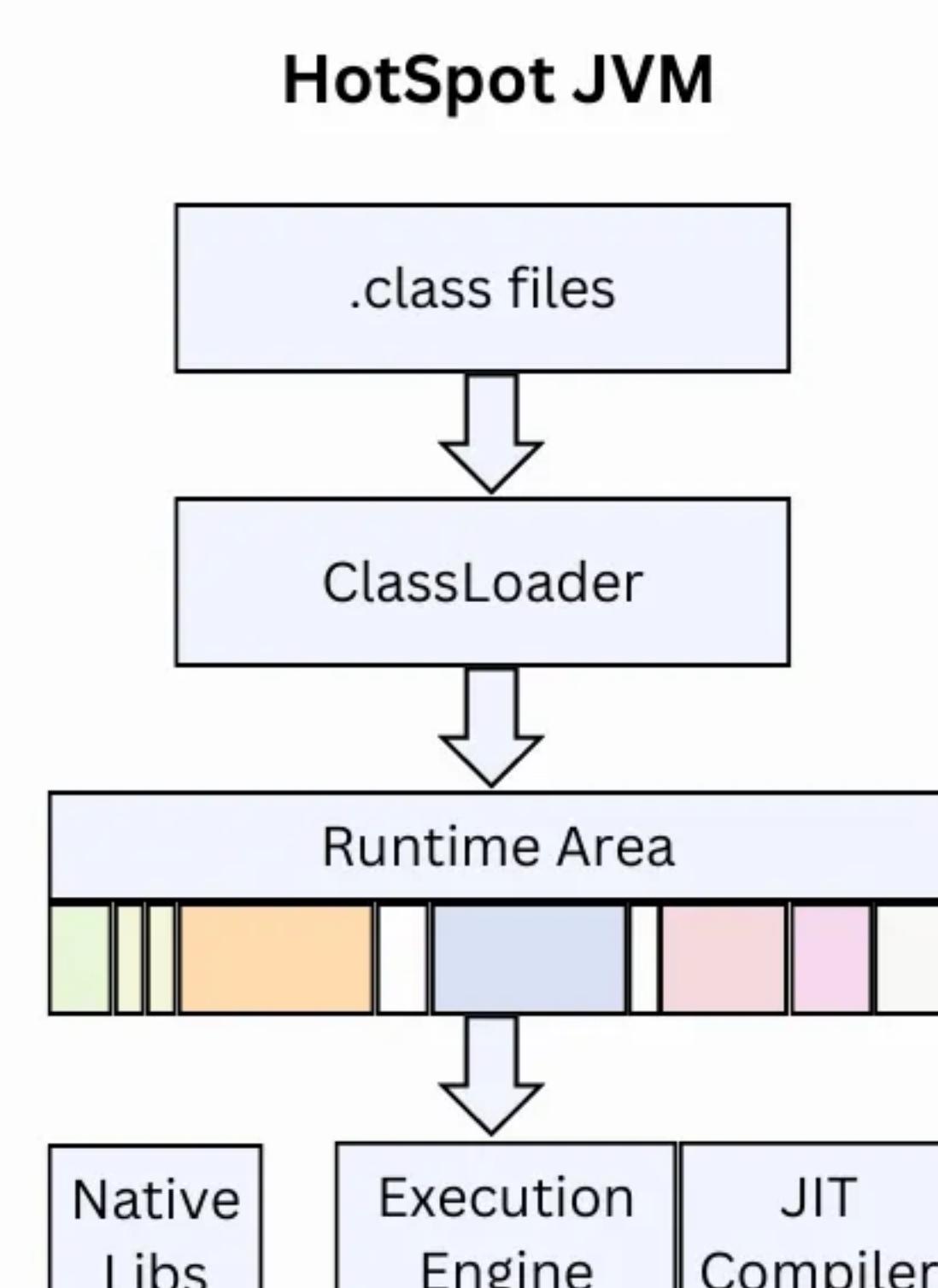
Set heap size: -Xms, -Xmx

Choose GC: -XX:+UseG1GC

Threads: -XX:ParallelGCThreads

Profiling: -XX:+PrintGCDetails

JIT: -XX:CompileThreshold



JVM Providers

Oracle OpenJDK AdoptOpenJDK Amazon Corretto IBM	Azul Systems SAP Machine Red Hat Microsoft
---	---

Class Loading

