Java Garbage Collection Introduction

In Java, allocation and de-allocation of memory space for objects are done by the garbage collection process in an automated way by the JVM. Unlike C language the developers need not write code for garbage collection in Java. This is one among the many features that made Java popular and helps programmers write better Java applications.

This is a **four part tutorial series** to know about the basics of garbage collection in Java,

1. Java Garbage Collection Introduction
2. How Java Garbage Collection Works?
3. Types of Java Garbage Collectors
4. Monitoring and Analyzing Java Garbage Collection

This tutorial is the first part in the series. It will explain the basic terminologies like JDK, JVM, JRE, HotSpot VM then understand the JVM architecture and Java heap memory structure. It is important to understand these basic things before going into the Garbage collection tutorial.

Key Java Terminologies

- Java API – is a collection of packaged libraries that helps developers to create Java applications.

- Java Development Kit (JDK) – is a set of tools that enables a developer to create Java applications. JDK includes tools to compile, run, package, distribute and monitor Java applications.
- Java Virtual Machine (JVM) – JVM is an abstract computing machine. Java programs are written against the JVM specification. JVM is specific for OS platform and they translate the Java instructions to the underlying platform specific instructions and execute them. JVM enables the Java programs to be platform independent.
- Java Runtime Environment (JRE) – JRE comprises the JVM implementation and the Java API.

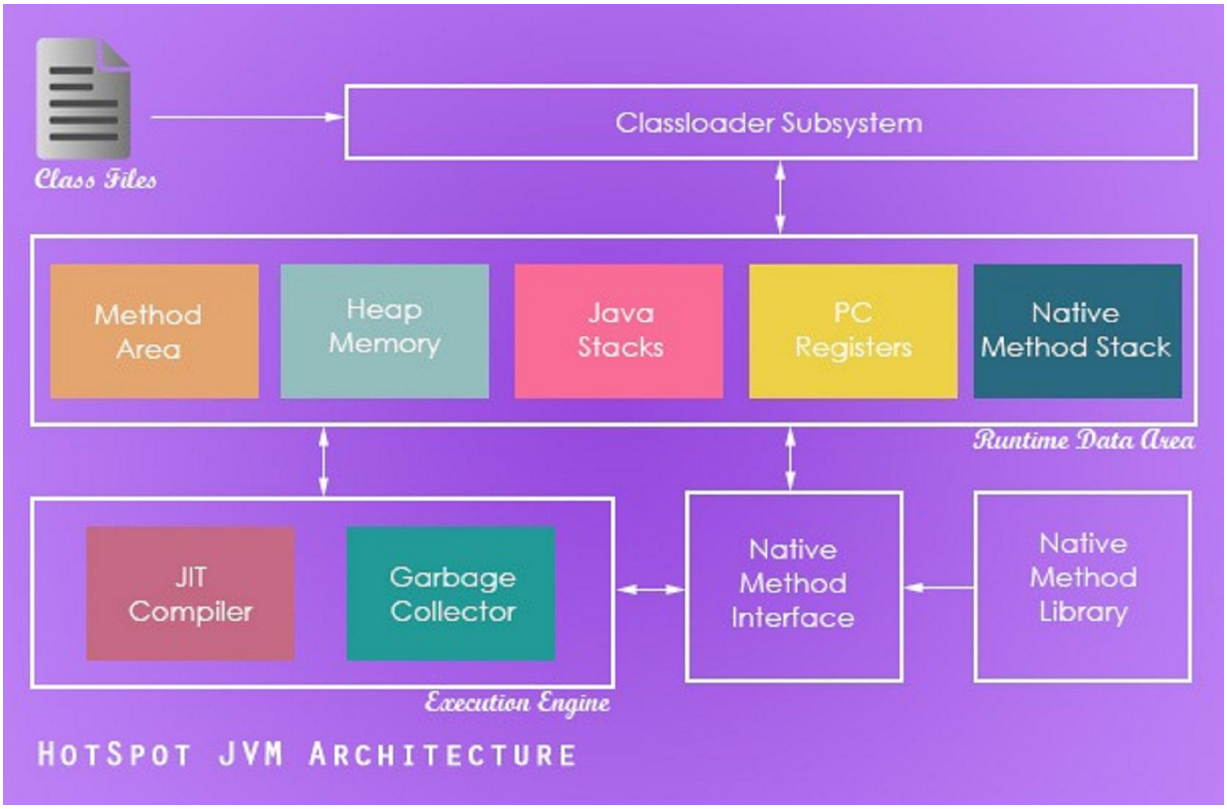
Java HotSpot Virtual Machine

Each JVM implementation may be different in the way the garbage collection principles are implemented. Prior to SUN takeover Oracle had JRockit JVM and after takeover it got the HotSpot JVM. Presently Oracle maintains both the JVM implementations and it has declared that over a period these two JVM implementations will be converged to one.

HotSpot JVM is available as a core component as part of the standard Oracle SE platform. In this garbage collection tutorial, we will see the garbage collection principles based on the HotSpot Virtual Machine.

JVM Architecture

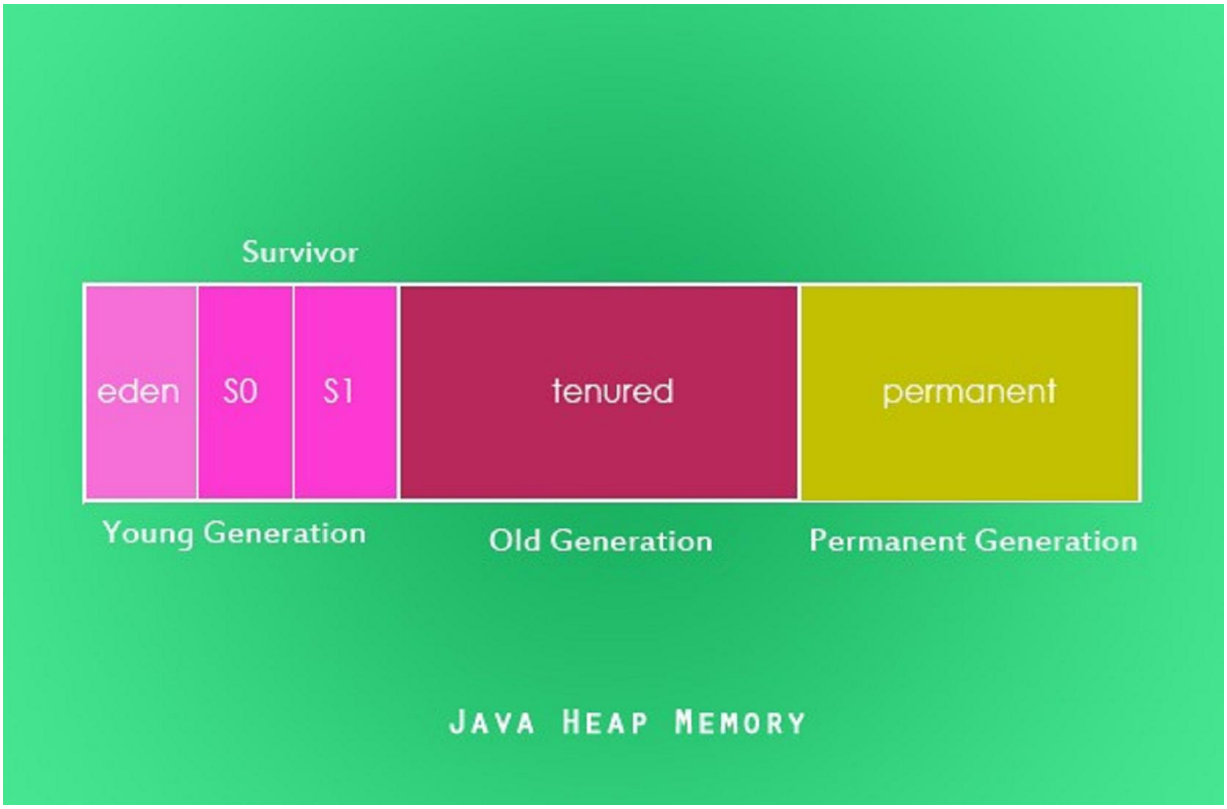
Following diagram summarizes the key components in a JVM. In the JVM architecture, two main components that are related to garbage collection are heap memory and garbage collector. Heap memory is the runtime data area where the instances will be store and the garbage collector will operate on. Now we know how these things fit in the larger scheme.



Java Heap Memory

It is essential to understand the role of heap memory in JVM memory model. At runtime the Java instances are stored in the heap memory area. When an object is not referenced anymore it becomes eligible for eviction from heap memory. During garbage collection process, those objects are evicted from heap memory and the space is reclaimed. Heap memory has three major areas,

1. Young Generation
    1. Eden Space (any instance enters the runtime memory area through eden)
    2. S0 Survivor Space (older instances moved from eden to S0)
    3. S1 Survivor Space (older instances moved from S0 to S1)
2. Old Generation (instances promoted from S1 to tenured)
3. Permanent Generation (contains meta information like class, method detail)



Update: Permanent Generation (Permgen) space is removed from Java SE 8 features.

In this next part of this tutorial series we will see about how garbage collection works in Java.

This Java tutorial was added on 12/10/2014.

Share This Tutorial:

Twitter Facebook Google+

« Getting Started With Hibernate

How Java Garbage Collection Works? »

Comments on "Java Garbage Collection Introduction" Tutorial:

How Java Garbage Collection Works? says: *12/10/2014 at 10:32 pm*

[...] works. This is the second part in the garbage collection tutorial series. Hope you have read introduction to Java garbage collection, which is the first [...]

Reply

Types of Java Garbage Collectors says: *12/10/2014 at 11:22 pm*

[...] works in Java, it is an interesting read and I recommend you to go through it. In the part 1 introduction to Java garbage collection, we saw about the JVM architecture, heap memory model and surrounding Java [...]

Reply

Rajkumar says: *14/10/2014 at 10:37 am*

I am eagerly waiting for your “Monitoring and Analyzing Java Garbage Collection” thread.

Reply

Java Garbage Collection Monitoring and Analysis says: *19/10/2014 at 5:05 pm*

[...] In this Java garbage collection tutorial series let us look about the tools available for garbage collection monitoring and analysis. Then use a tool and monitor an example Java application for garbage collection process. If you are a beginner it is better for you to go through this series of tutorials. You can start with introduction for Java garbage collection. [...]

Reply

Your Comment

Name

Email

Post Comment

Java

↑ Go to top

Introduction

Java Basics

Java and OOPS

Java String

Exception Handling

Java Serialization

Java Collections Framework

Java Generics

Java Util

Concurrent Util

Java JDBC

Java Internals

Java Garbage Collection

Java Garbage Collection Introduction

How Java Garbage Collection Works?

Types of Java Garbage Collectors

Java Garbage Collection Monitoring and Analysis

Java 8

Java Puzzles

Third Party

Tools

Java Interview Questions

Others

Java Gallery

Android

Design Patterns

Hibernate

Spring

Web Services

Servlet

JavaPapers is a tutorials site and **Joe** *runs it passionately.*

Say hi: joe@javapapers.com

Recommended Tutorials

- Java History
- Overloading Vs Overriding In Java
- Eclipse Shortcuts
- Java Serialization
- Difference Between Interface And Abstract Class
- Java Hashtable
- Difference Between Forward And SendRedirect
- Differentiate JVM JRE JDK JIT
- Java (JVM) Memory Types
- Why Multiple Inheritance Is Not Supported In Java

Web Services Servlet

Site Map