

# Java GC系列（1）：Java垃圾回收簡介

 [importnew.com/13504.html](https://importnew.com/13504.html)

本文由 [ImportNew](#) - 好好先生 翻譯自 [javapapers](#)。歡迎加入[翻譯小組](#)。轉載請見文末要求。

Java的內存分配與回收全部由JVM垃圾回收進程自動完成。與C語言不同，Java開發者不需要自己編寫代碼實現垃圾回收。這是Java深受大家歡迎的眾多特性之一，能夠幫助程序員更好地編寫Java程序。

下面四篇教程是瞭解Java 垃圾回收（GC）的基礎：

1. [垃圾回收簡介](#)
2. [垃圾回收是如何工作的？](#)
3. [垃圾回收的類別](#)
4. [垃圾回收監視和分析](#)

這篇教程是系列第一部分。首先會解釋基本的術語，比如JDK、JVM、JRE和HotSpotVM。接著會介紹JVM結構和Java 堆內存結構。理解這些基礎對於理解後面的垃圾回收知識很重要。

## Java關鍵術語

- Java API：一系列幫助開發者創建Java應用程序的封裝好的庫。
- Java 開發工具包（JDK）：一系列工具幫助開發者創建Java應用程序。JDK包含工具編譯、運行、打包、分發和監視Java應用程序。
- Java 虛擬機（JVM）：JVM是一個抽象的計算機結構。Java程序根據JVM的特性編寫。JVM針對特定於操作系統並且可以將Java指令翻譯成底層系統的指令並執行。JVM確保了Java的平台無關性。
- Java 運行環境（JRE）：JRE包含JVM實現和Java API。

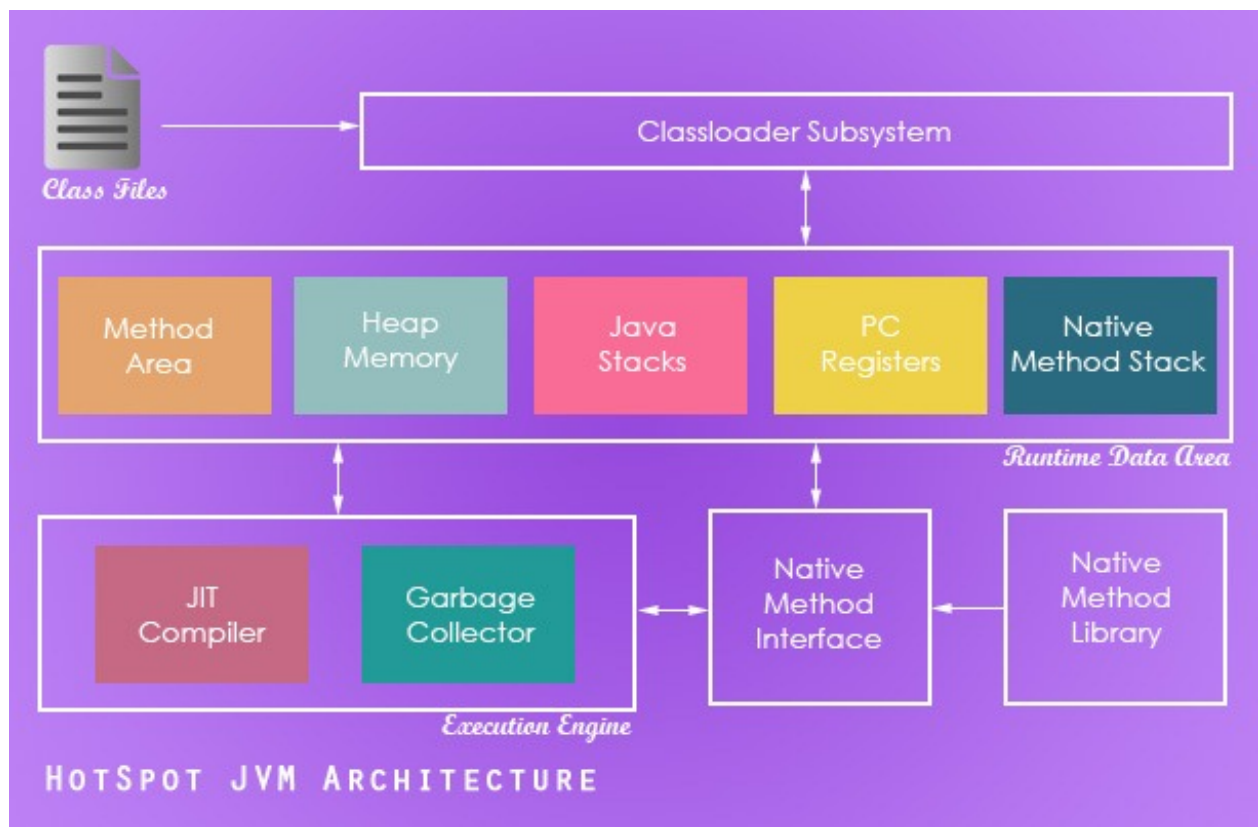
## Java HotSpot 虛擬機

每種JVM實現可能採用不同的方法實現垃圾回收機制。在收購SUN之前，Oracle使用的是JRockit JVM，收購之後使用HotSpot JVM。目前Oracle擁有兩種JVM實現並且一段時間後兩個JVM實現會合二為一。

HotSpot JVM是目前Oracle SE平台標準核心組件的一部分。在這篇垃圾回收教程中，我們將會瞭解基於HotSpot虛擬機的垃圾回收原則。

## JVM體系結構

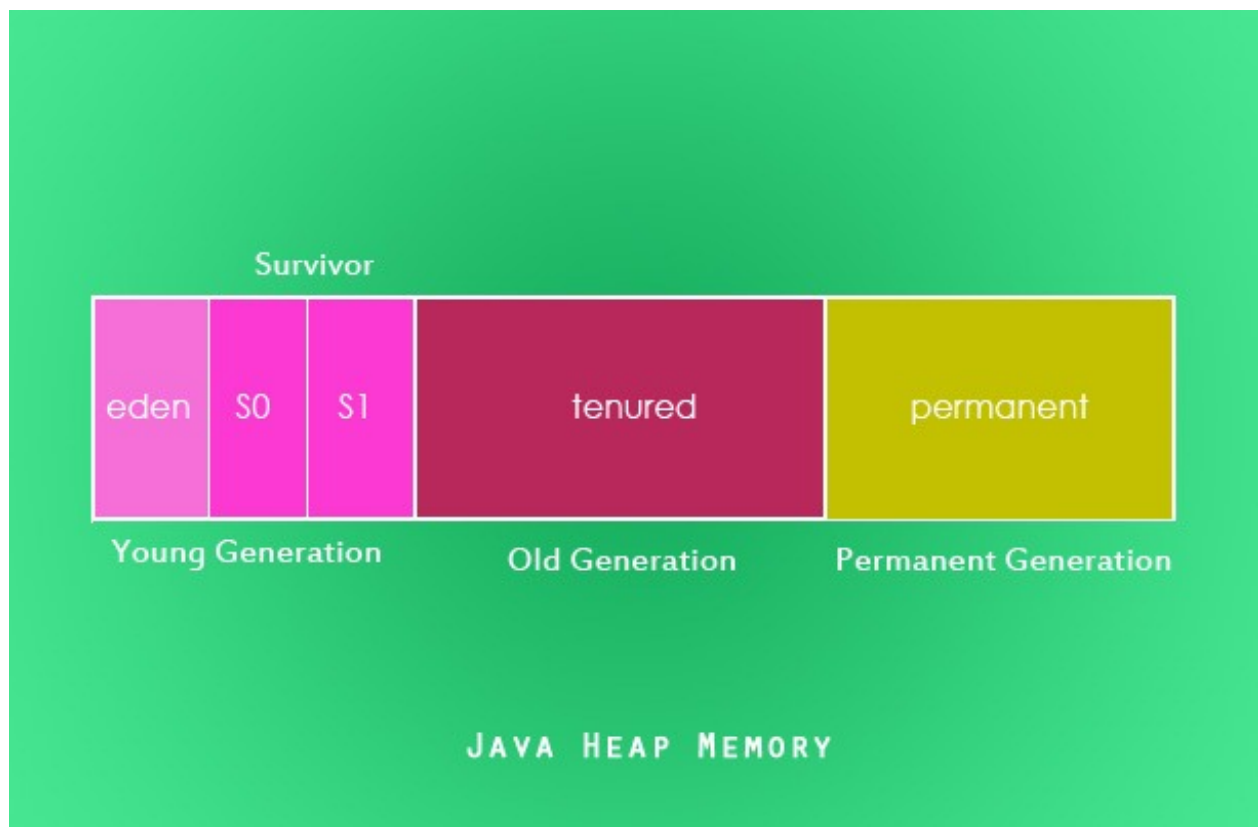
下面圖片總結了JVM的關鍵組件。在JVM體系結構中，與垃圾回收相關的兩個主要組件是堆內存和垃圾回收器。堆內存是內存數據區，用來保存運行時的對象實例。垃圾回收器也會在這裡操作。現在我們知道這些組件是如何在框架中工作的。



## Java堆內存

我們有必要瞭解堆內存在JVM內存模型的角色。在運行時，Java的實例被存放在堆內存區域。當一個對象不再被引用時，滿足條件就會從堆內存移除。在垃圾回收進程中，這些對象將會從堆內存移除並且內存空間被回收。堆內存以下三個主要區域：

1. 新生代 (Young Generation)
  - Eden空間 (Eden space, 任何實例都通過Eden空間進入運行時內存區域)
  - S0 Survivor空間 (S0 Survivor space, 存在時間長的實例將會從Eden空間移動到S0 Survivor空間)
  - S1 Survivor空間 (存在時間更長的實例將會從S0 Survivor空間移動到S1 Survivor空間)
2. 老年代 (Old Generation) 實例將從S1提升到Tenured (終身代)
3. 永久代 (Permanent Generation) 包含類、方法等細節的元信息



永久代空間在Java SE8特性中已經被移除。

在本系列的第二篇將會介紹Java垃圾回收是如何工作的。

原文鏈接：[javapapers](#) 翻譯：[ImportNew.com](#) - 好好先生

譯文鏈接：<http://www.importnew.com/13504.html>

[轉載請保留原文出處、譯者和譯文鏈接。]

關於作者：[好好先生](#)

(新浪微博：[@好好先生耶2](#))

[查看好好先生的更多文章 >>](#)