

[Home](#) » [Java](#) » Phân biệt Heap memory và Stack memory trong java

Phân biệt Heap memory và Stack memory trong java

Posted on Tháng Mười Một 1, 2018

Phân biệt Heap memory và Stack memory trong java.

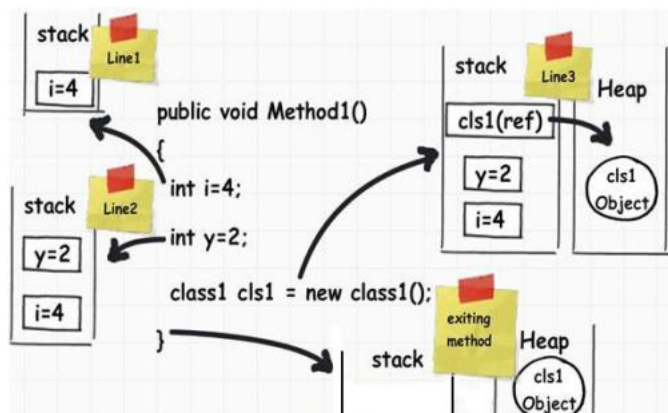
Bộ nhớ Heap – Heap Memory

- Heap memory là bộ nhớ được sử dụng bởi Java Runtime để cấp phát bộ nhớ cho các đối tượng (object) và String.
- Bất kỳ khi nào có một đối tượng được tạo, nó sẽ được tạo lưu ở bộ nhớ Heap.
- Bộ dọn rác (Garbage Collection) chạy trên heap memory để giải phóng bộ nhớ được sử dụng bởi các đối tượng có bất kỳ tham chiếu nào.

Bộ nhớ Stack – Stack Memory

- Stack memory được sử dụng cho quá trình thực thi của mỗi thread.
- Stack memory bao gồm các giá trị cụ thể của method: các biến local và các tham chiếu tới các đối tượng chứa ở trong heap memory được tham chiếu bởi method.
- Stack memory được tham chiếu theo thứ tự LIFO (Last In First Out – vào cuối cùng thì ra đầu tiên). Tức là lưu trữ kiểu ngăn xếp (stack). Khi có một method được thực thi, một block được tạo ra trong stack memory để chứa các biến nguyên thủy local và các tham chiếu tới các object. Khi method kết thúc, block đó sẽ không còn được sử dụng và được phục vụ cho method tiếp theo.
- Stack memory có kích thước rất nhỏ so với Heap memory.

Ví dụ 1:



JAVA

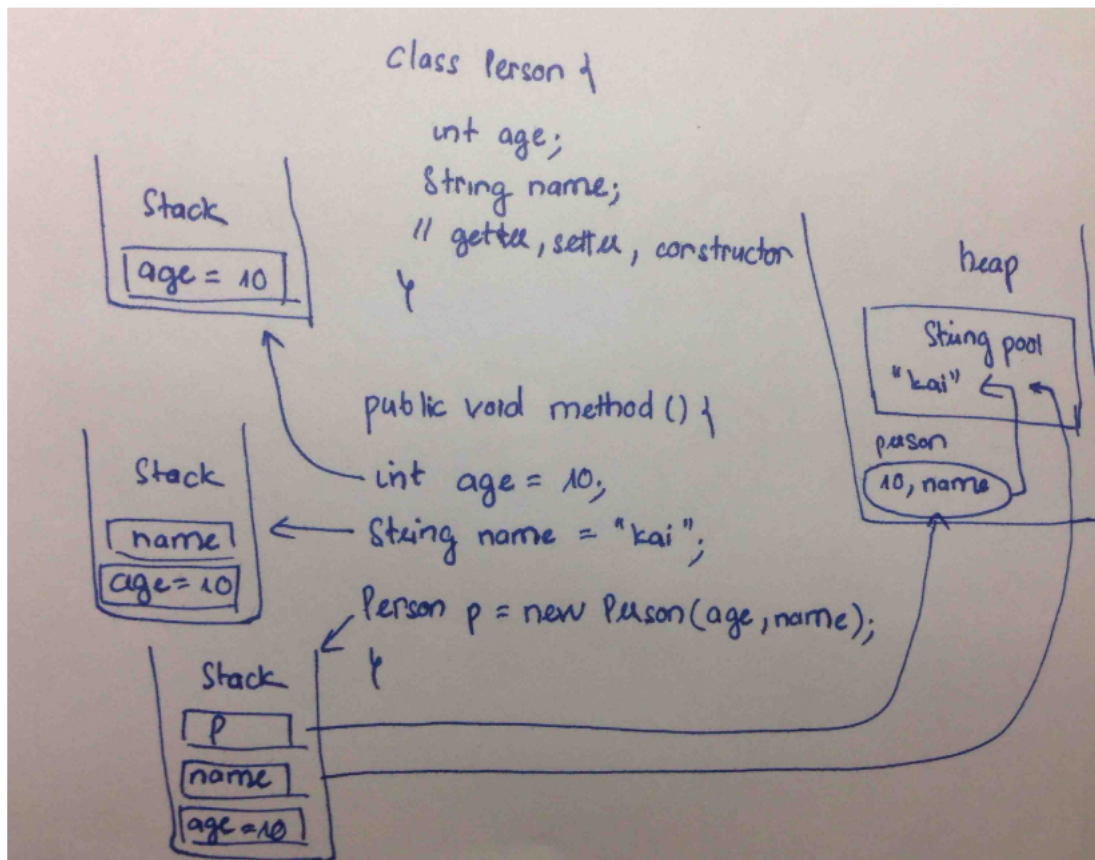
• CompletableFuture là gì? Code ví

- Dòng thứ 1: khi khai báo `int i = 4` nó sẽ đưa `i = 4` vào stack
- Dòng thứ 2: khi khai báo `y = 2` nó sẽ đưa `y = 2` vào stack (xếp trên `i = 4`)
- Dòng thứ 3: khi khai báo `class1 cls1 = new class1();` đây là kiểu đối tượng nên nó sẽ tạo đối tượng `cls1` trong heap đồng thời chưa tham chiếu của đối tượng `cls1` vào stack (xếp trên cùng trong stack)
- Sau khi kết thúc method, bộ nhớ trong stack được giải phóng, còn bộ nhớ trong heap thì chưa. Bộ nhớ trong heap phải chờ cho tới khi garbage collector (bộ dọn rác) của Java quét qua để giải phóng.

Ví dụ 2:

Tương tự như ví dụ trên nhưng với trường hợp đối tượng cụ thể:

- Điều đặc biệt ở đây là khi bạn khai báo `String name = "kai"`. Giá trị `"kai"` sẽ được lưu trong String pool và biến `name` trong stack thực hiện tham chiếu tới `"kai"`. (String là kiểu dữ liệu đặc biệt, giá trị của nó luôn được lưu trong String pool)
- Khi bạn khai báo `Person p = new Person(age, name)` nó sẽ tạo đối tượng `p` với `age`, `name` tương ứng vào bộ nhớ heap và thêm biến tham chiếu tới đối tượng `p` vào bộ nhớ tack.



Phân biệt Heap với Stack:

dự Java CompletableFuture (Java 8)

• Code ví dụ Callable, Future, Executors trong Java

• So sánh Future và CompletableFuture trong Java

• Phân biệt Heap memory và Stack memory trong java

• Hướng dẫn cài đặt Java 8 trên Windows

• Code ví dụ Java đặt lịch với Quartz Scheduler

• Code ví dụ Java đặt lịch với Timer, TimerTask

• Cài đặt Java trên Linux/Ubuntu (Oracle JDK – Open JDK)

• Cài đặt Java, JDK trên Ubuntu (Linux) bằng file tar.gz

• Code ví dụ Java SocketCluster Client, publish và subscribe channel

• Code ví dụ Java đọc RSS Feed (Java RSS Reader)

• Các cách khởi tạo đối tượng trong Java (Java Constructor)

• Code ví dụ Java Reflection, Clone Object, Copy Object

• Java Reflection là gì? Hướng dẫn Java Reflection API

• Java Serializable là gì? Serialization và Deserialization trong Java

• Các loại dữ liệu nguyên thủy trong Java (primitives variable)

Okay, Done!

(Xem thêm: Lỗi stackoverflowerror trong Java)

(Xem thêm: Điều chỉnh kích thước, heap, stack cho ứng dụng Java)

(Xem thêm: String Pool là gì? String Pool trong Java)

References:

https://docs.oracle.com/.../garbage_collect.html

<https://gurunguns.wordpress.com/.../stack-heap-value-types-reference-types-boxing-and-unboxing/>

This entry was posted in *Java*. Bookmark the *permalink*.

← [Đổi tên field trong MongoDB với \\$rename](#)

[Inject Spring Bean vào các class không phải là @Component](#)



▫ [Code ví dụ java giải nén file rar \(decompress file rar\)](#)

▫ [Package trong java là gì? Ý nghĩa Package trong Java](#)

▫ [ProcessBuilder trong Java, chạy lệnh cmd trong Java](#)

▫ [Những tính năng mới trong Java 8.](#)

▫ [Immutable là gì? Sự khác nhau giữa Immutable với Mutable](#)

▫ [Kéo thả Java Swing trên Eclipse giống như NetBean.](#)

▫ [Các tính chất hướng đối tượng của Java](#)

▫ [Sự khác nhau giữa Abstract class với Interface trong Java.](#)

▫ [HashCode với Equals trong Java](#)

▫ [Ngôn ngữ Java là gì? Đặc trưng của Java](#)

▫ [So sánh lập trình cấu trúc với hướng đối tượng](#)

▫ [SCJP, OCPJP là gì, tài liệu SCJP, OCPJP tiếng việt](#)

▫ [Code ví dụ đa ngôn ngữ với Java \(ResourceBundle, Internationalization\)](#)

▫ [Java 2 là gì? Các version/Phiên bản của Java](#)

▫ [Phân biệt giữa Java ME, Java SE và Java EE](#)

▫ [So sánh sự khác nhau giữa J2ME, J2SE và J2EE](#)

CHUYÊN MỤC

- Algorithm
- Apache
- Apache JMeter
- Apache Kafka
- AWS
- C/C++
- CDI
- Clean Code
- Demo
- Design Pattern
- Docker
- Eclipse
- Elasticsearch
- Excel
- FAQ
- Framework
- Freemaker
- FreeMarker
- Gradle
- Hibernate
- HttpClient
- HttpComponents
- Install
- IntelliJ IDEA
- Java
- Java Basic
- Java Core
- Java8
- JavaScript

- jooq
- JSF
- JSP-Servlet
- JUnit
- Library
- Linux
- Maven
- MongoDB
- MySQL
- Network Programming
- Node.js
- OOP
- PostgreSQL
- PrimeFaces
- Principle
- Python
- quartz
- Redis
- SDKMan
- Security
- SocketCluster
- Spring
- Spring Boot
- Spring Core
- Spring Data
- Spring Hibernate
- Spring JDBC
- Spring MVC
- Spring Security
- Thymeleaf

- Tomcat
- Uncategorized
- Web Service
- WebSocket
- Wordpress

