

# Memory

## 1. Thế nào là cấp phát tĩnh và cấp phát động ?

- **Cấp phát tĩnh (Static Memory Allocation):** Cấp phát tĩnh là quá trình cấp phát bộ nhớ cho các biến trước khi chương trình thực thi, cụ thể là trong quá trình biên dịch (compile time). Kích thước của vùng nhớ phải được xác định trước và không thể thay đổi trong suốt quá trình chạy. Vùng nhớ này được quản lý trên Stack.

- **Đặc điểm:**

- Tốc độ rất nhanh vì quá trình cấp phát và giải phóng diễn ra tự động khi một phương thức được gọi và kết thúc.
- Kích thước cố định, không thể thay đổi trong quá trình chạy.
- Thường dùng để lưu trữ các biến kiểu nguyên thủy (int, boolean, char,...) và các biến tham chiếu (chỉ lưu địa chỉ, không phải đối tượng).

**Ví dụ:** Khai báo một mảng có kích thước cố định `int arr[10];`

- **Cấp phát động (Dynamic Memory Allocation):** Cấp phát động là quá trình cấp phát bộ nhớ khi chương trình đang chạy (run time). Có thể yêu cầu cấp phát hoặc giải phóng bộ nhớ khi cần, giúp tối ưu hóa việc sử dụng tài nguyên. Vùng nhớ này được quản lý trên **Heap**.

- **Đặc điểm:**

- Tốc độ chậm hơn Stack, nhưng linh hoạt hơn.
- Kích thước có thể thay đổi trong quá trình chạy.
- Dùng để lưu trữ tất cả các đối tượng (objects) và mảng (arrays) được tạo ra bằng từ khóa `new`.
- Bộ nhớ được quản lý bởi Garbage Collector của Java, giúp tự động giải phóng bộ nhớ khi các đối tượng không còn được sử dụng.

**Ví dụ:** `Student student = new Student();`

`int[] scores = new int[5];`

## 2. Phân biệt bộ nhớ heap và bộ nhớ stack ?

### - Bộ nhớ Stack (Ngăn xếp)

Bộ nhớ Stack hoạt động theo cơ chế **LIFO (Last-In, First-Out)**, tương tự như một chồng đĩa. Dữ liệu được đưa vào sau sẽ được lấy ra trước.

- **Vị trí:** Lưu trữ các biến cục bộ, tham số của phương thức và địa chỉ trả về của các hàm.
- **Tốc độ:** Rất nhanh, vì việc cấp phát và giải phóng bộ nhớ được tự động và đơn giản.
- **Kích thước:** Nhỏ và cố định khi đầy sẽ gây lỗi `StackOverflowError`.
- **Ví dụ:** Khai báo một biến `int x = 5`; bên trong một phương thức, biến `x` và giá trị của nó sẽ được lưu trên Stack.

### - Bộ nhớ Heap

Bộ nhớ Heap là một vùng bộ nhớ lớn hơn và linh hoạt hơn. Nó không tuân theo một cơ chế thứ tự cụ thể nào.

- **Vị trí:** Lưu trữ tất cả các đối tượng (objects) và mảng (arrays) được tạo bằng từ khóa `new`.
- **Tốc độ:** Chậm hơn Stack, vì quá trình quản lý và truy cập phức tạp hơn.
- **Kích thước:** Lớn và linh hoạt, khi đầy sẽ gây lỗi `OutOfMemoryError`.
- **Ví dụ:** Tạo một đối tượng như `String name = new String("Java");`, đối tượng `String` sẽ được lưu trên Heap, trong khi biến tham chiếu `name` sẽ được lưu trên Stack.