**Phần 5. Memory**

1. *Cấp phát tĩnh*

Cấp phát tĩnh (Static Allocation) là quá trình phân bổ bộ nhớ cố định cho các biến, mảng, hoặc đối tượng trước khi chương trình chạy. Các biến này được cấp phát bộ nhớ tại thời điểm biên dịch (compile-time), và không thay đổi trong suốt thời gian chương trình hoạt động.

* **Ưu điểm**:
  + Tối ưu hóa tốc độ, vì bộ nhớ đã được cấp phát sẵn, không cần phải cấp phát hoặc giải phóng khi đang chạy.
  + Dễ quản lý, vì bộ nhớ cố định trong suốt chương trình.
* **Nhược điểm**:
  + Không linh hoạt vì dung lượng bộ nhớ đã cố định.
  + Có thể gây lãng phí bộ nhớ nếu kích thước không phù hợp.

*Ví Dụ*

*int arr[10]; // Mảng có kích thước cố định là 10 phần tử*

1. *Cấp phát động*

Cấp phát động (Dynamic Allocation) là quá trình phân bổ bộ nhớ tại thời điểm chạy (runtime) khi chương trình yêu cầu. Điều này cho phép linh hoạt thay đổi kích thước bộ nhớ theo nhu cầu.

* **Ưu điểm**:
  + Linh hoạt, có thể thay đổi kích thước bộ nhớ khi chương trình chạy.
  + Giảm lãng phí bộ nhớ vì chỉ cấp phát khi cần thiết.
* **Nhược điểm**:
  + Chậm hơn cấp phát tĩnh vì mất thời gian cấp phát và giải phóng.
  + Dễ gặp lỗi như tràn bộ nhớ (memory leak) nếu không giải phóng bộ nhớ đã cấp phát.

*Ví Dụ*

*int[] arr = new int[n]; // Mảng có kích thước được quyết định tại thời điểm runtime*

\* Bộ nhớ Stack và bộ nhớ Heap

1. Heap:

Heap là nơi lưu trữ các đối tượng Java. Khi tạo một đối tượng bằng cách sử dụng từ khóa "new", đối tượng sẽ được lưu trữ trong Heap. Vùng nhớ Heap có thể tồn tại đến khi không có tham chiếu đến đối tượng đó nữa hoặc cho đến khi chương trình kết thúc.

2. Stack:

Stack là nơi lưu trữ các biến cục bộ (các biến bên trong phương thức) và các tham số của phương thức . Mỗi lần một phương thúc được gọi, một frame mới được tạo ra trên Stack để lưu trữ các biến cục bộ và tham số của phương thức đó. Khi phương trình hoàn thành, frame được loại bỏ khỏi Stack và bộ nhớ được giải phóng.

*Note: Frame là một đơn vị dữ liệu được tạo ra khi một phương thức được gọi trong chương trình Java. Mỗi frame chứa các thông tin về biến cục bộ, tham số của phương thức và các giá trị trả về. Mỗi khi phương thức được gọi frame đẩy được tạo và đẩy vào đỉnh của stack, khi kết thúc hàm frame trên stack sẽ được loại bỏ.*

1. Cách lưu trữ bộ nhớ trên Heap và Stack khác nhau:

* Trong Heap, bộ nhớ được cấp phát động bởi JVM khi các đối tượng được tạo ra và bộ nhớ được giải phóng khi không có tham chiếu nào trỏ đến đối tượng đó nữa.
* Trong Stack, bộ nhớ được cấp phát tĩnh khi một phương thức được gọi và bộ nhớ được giải phóng khi phương thức hoàn thành.