Memory

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

**a. Cấp phát tĩnh (Static Memory Allocation):**

* **Định nghĩa**: Là quá trình cấp phát bộ nhớ tại **thời điểm biên dịch**. Kích thước bộ nhớ được xác định trước và không thay đổi trong suốt chương trình.
* **Đặc điểm**:
  + Kích thước của biến phải biết trước khi chương trình chạy.
  + Bộ nhớ được cấp phát tự động và giải phóng khi chương trình kết thúc.
  + Không cần giải phóng bộ nhớ thủ công.
  + Thường được lưu trữ trong **stack memory**.

Vd:

Int num = 10;

**b. Cấp phát động (Dynamic Memory Allocation):**

* **Định nghĩa**: Là quá trình cấp phát bộ nhớ tại **thời điểm chạy chương trình**. Kích thước bộ nhớ có thể thay đổi khi chương trình đang chạy.
* **Đặc điểm**:
  + Bộ nhớ được cấp phát tại thời điểm chạy chương trình (runtime).
  + Cần giải phóng bộ nhớ để tránh tình trạng rò rỉ bộ nhớ.
  + Thường được lưu trữ trong **heap memory**.

Vd:

Integer num = new Integer(10);

**2. Phân biệt bộ nhớ Heap và Stack**

**a. Bộ nhớ Stack:**

* **Định nghĩa**: Là vùng bộ nhớ được quản lý theo cấu trúc **LIFO** (Last In First Out). Dùng để lưu trữ biến cục bộ và lời gọi hàm.
* **Đặc điểm**:
  + Kích thước nhỏ và cố định.
  + Bộ nhớ được cấp phát và giải phóng tự động khi phương thức kết thúc.
  + Nhanh chóng và hiệu quả trong việc truy xuất.
  + Quản lý bởi trình biên dịch (compiler).

Đây là nội dung đã được viết lại để bạn dễ nhập vào file .docx:

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

**a. Cấp phát tĩnh (Static Memory Allocation):**

* **Định nghĩa**: Là quá trình cấp phát bộ nhớ tại **thời điểm biên dịch**. Kích thước bộ nhớ được xác định trước và không thay đổi trong suốt chương trình.
* **Đặc điểm**:
  + Kích thước của biến phải biết trước khi chương trình chạy.
  + Bộ nhớ được cấp phát tự động và giải phóng khi chương trình kết thúc.
  + Không cần giải phóng bộ nhớ thủ công.
  + Thường được lưu trữ trong **stack memory**.

Ví dụ:

java

Copy code

int num = 10; // Biến này được cấp phát tĩnh

**b. Cấp phát động (Dynamic Memory Allocation):**

* **Định nghĩa**: Là quá trình cấp phát bộ nhớ tại **thời điểm chạy chương trình**. Kích thước bộ nhớ có thể thay đổi khi chương trình đang chạy.
* **Đặc điểm**:
  + Bộ nhớ được cấp phát tại thời điểm chạy chương trình (runtime).
  + Cần giải phóng bộ nhớ để tránh tình trạng rò rỉ bộ nhớ.
  + Thường được lưu trữ trong **heap memory**.

Ví dụ:

java

Copy code

Integer num = new Integer(10); // Cấp phát động

**2. Phân biệt bộ nhớ Heap và Stack**

**a. Bộ nhớ Stack:**

* **Định nghĩa**: Là vùng bộ nhớ được quản lý theo cấu trúc **LIFO** (Last In First Out). Dùng để lưu trữ biến cục bộ và lời gọi hàm.
* **Đặc điểm**:
  + Kích thước nhỏ và cố định.
  + Bộ nhớ được cấp phát và giải phóng tự động khi phương thức kết thúc.
  + Nhanh chóng và hiệu quả trong việc truy xuất.
  + Quản lý bởi trình biên dịch (compiler).

Ví dụ:

java

Copy code

public static void method() {

int x = 5; // x lưu trữ trong stack memory

}

**b. Bộ nhớ Heap:**

* **Định nghĩa**: Là vùng bộ nhớ cấp phát động, dùng để lưu trữ các đối tượng và các biến cấp phát động.
* **Đặc điểm**:
  + Bộ nhớ được quản lý bởi **Garbage Collector** trong Java.
  + Dùng để lưu trữ các đối tượng lớn.
  + Tốc độ truy cập chậm hơn Stack do quản lý phức tạp hơn.
  + Không có quy tắc chặt chẽ về thứ tự cấp phát bộ nhớ.