1. **Khi nào thì nên sử dụng biến static, phương thức static?**

Trong Java, biến và phương thức static có ý nghĩa đặc biệt và được sử dụng trong các trường hợp sau:

* Biến static:
  + Biến static được chia sẻ và duy trì giá trị chung cho tất cả các đối tượng của một lớp. Nó thuộc về lớp chứ không thuộc về các đối tượng cụ thể.
  + Sử dụng biến static khi bạn muốn lưu trữ thông tin chung cho tất cả các đối tượng của lớp hoặc khi bạn muốn có một tài nguyên được chia sẻ.
  + Ví dụ: Biến static có thể được sử dụng để đếm số lượng đối tượng được tạo từ một lớp, lưu trữ thông tin cấu hình, hoặc khởi tạo các giá trị hằng số.
* Phương thức static:
  + Phương thức static thuộc về lớp chứ không thuộc về các đối tượng cụ thể. Bạn có thể gọi phương thức static trực tiếp từ lớp mà không cần tạo đối tượng của lớp đó.
  + Sử dụng phương thức static khi bạn muốn thực hiện một hoạt động độc lập với các thuộc tính của đối tượng.
  + Ví dụ: Phương thức static có thể được sử dụng để thực hiện tính toán đơn giản, cung cấp các tiện ích chung, hoặc làm việc với các thuộc tính static.

Lưu ý rằng sử dụng quá nhiều biến và phương thức static có thể làm cho mã của bạn trở nên khó bảo trì và kiểm thử, và cũng có thể gây ra sự phụ thuộc mạnh mẽ giữa các thành phần của lớp. Do đó, hãy sử dụng biến và phương thức static một cách cân nhắc và chỉ khi nó thực sự phù hợp với thiết kế và yêu cầu của lớp.

1. **Khi nào nên sử dụng biến final , phương thức final , class final trong java trong java ?**

* Biến final:
  + Sử dụng biến final khi bạn muốn khai báo một biến chỉ đọc (read-only), tức là giá trị của biến không thể thay đổi sau khi được gán một lần.
  + Biến final phải được khởi tạo trong quá trình khai báo hoặc trong constructor của lớp (đối với biến instance) hoặc trong phương thức static initializer (đối với biến static).
  + Sử dụng biến final để đảm bảo tính không thay đổi của một giá trị, tránh việc ghi đè (overwrite) hoặc thay đổi giá trị không mong muốn.
  + Ví dụ: Biến final có thể được sử dụng để lưu trữ các hằng số, như số PI, số phi, hoặc địa chỉ cố định.
* Phương thức final:
  + Sử dụng phương thức final khi bạn không muốn phương thức được ghi đè (override) bởi các lớp con.
  + Phương thức final không thể được ghi đè và giữ nguyên hiện thực (implementation) của phương thức trong lớp gốc.
  + Sử dụng phương thức final để đảm bảo tính nhất quán và an toàn cho các phương thức quan trọng trong lớp cơ sở.
  + Ví dụ: Phương thức final có thể được sử dụng trong lớp chứa các phương thức template hoặc xử lý các thao tác quan trọng mà không được ghi đè.
* Lớp final:
  + Sử dụng lớp final khi bạn không muốn lớp con được kế thừa từ lớp đó.
  + Lớp final không thể được mở rộng và không thể có các lớp con.
  + Sử dụng lớp final để đảm bảo tính nguyên vẹn của lớp, chẳng hạn như trong trường hợp lớp chứa các phương thức final quan trọng không được ghi đè.
  + Ví dụ: Lớp final có thể được sử dụng trong các lớp tiện ích hoặc lớp cung cấp các phương thức tĩnh (static) không thay đổi.

1. **Sự khác nhau giữa Runnable và Callable ?**

Bạn có thể xem xét các hướng dẫn sau để quyết định khi nào nên sử dụng Runnable, Callable và Thread:

* Sử dụng Runnable:
  + Khi bạn chỉ cần thực hiện một nhiệm vụ đơn giản trong một luồng.
  + Khi bạn không cần nhận kết quả trả về sau khi nhiệm vụ hoàn thành.
  + Khi bạn muốn sử dụng khối lệnh run() để định nghĩa nhiệm vụ cần thực hiện.
* Sử dụng Callable:
  + Khi bạn cần thực hiện một nhiệm vụ phức tạp hơn trong một luồng.
  + Khi bạn muốn nhận kết quả trả về từ nhiệm vụ sau khi hoàn thành.
  + Khi bạn muốn sử dụng phương thức call() để định nghĩa nhiệm vụ cần thực hiện và trả về kết quả.

1. **Tài liệu về Java Colection Frameworkt rong java?**
2. **Stream in java ?**