# Đặc điểm và tính chất của String trong Java

1. Immutable: Một khi được tạo ra, giá trị của một đối tượng String không thể thay đổi. Nếu bạn thực hiện các thao tác như nối chuỗi hay thay đổi chuỗi, Java sẽ tạo ra một đối tượng String mới thay vì thay đổi đối tượng cũ.
2. Nhạy cảm với chữ hoa chữ thường: Các chuỗi trong Java phân biệt chữ hoa chữ thường, nghĩa là "abc" khác với "ABC".
3. Kích thước cố định: Kích thước của một đối tượng String là cố định, không thể thay đổi sau khi nó đã được khởi tạo.
4. Lưu trữ trong String pool: Java sử dụng một khu vực bộ nhớ đặc biệt gọi là String pool để lưu trữ các đối tượng String, giúp tiết kiệm bộ nhớ khi nhiều chuỗi giống nhau được sử dụng.

# Các cách để tạo một biến String

1. Sử dụng dấu nháy đơn:

String str = "Hello, World!";

1. Sử dụng từ khóa new:

String str = new String("Hello, World!");

1. Tạo từ mảng ký tự:

char[] chars = {'H', 'e', 'l', 'l', 'o'};

String str = new String(chars);

1. Tạo từ byte:

byte[] bytes = {1, 2, 3};

String str = new String(bytes);

# String Pool

* String pool là một phần của heap memory, nơi mà các đối tượng String được lưu trữ. Khi tạo một String bằng cách sử dụng dấu nháy đơn (String str = "Hello, World!";), Java sẽ kiểm tra xem có một String giống vậy đã tồn tại trong String pool chưa. Nếu có, nó sẽ trả về tham chiếu đến chuỗi đã tồn tại. Nếu không, nó sẽ tạo ra một chuỗi mới và lưu nó vào pool. Điều này giúp tiết kiệm bộ nhớ, vì các chuỗi giống nhau không cần phải được lưu trữ nhiều lần.

# So sánh hai String trong Java

1. Sử dụng phương thức equals() để so sánh giá trị 2 String:

String str1 = "Hello";

String str2 = "Hello";

boolean isEqual = str1.equals(str2); // true

1. Sử dụng ‘==’ để so sánh địa chỉ của 2 String trong bộ nhớ:

String str1 = new String("Hello");

String str2 = new String("Hello");

boolean isSameReference = str1 == str2; // false, vì str1 và str2 là 2 object khác nhau