

Tôi Học Code

Đừng chỉ là coder, hãy là một developer

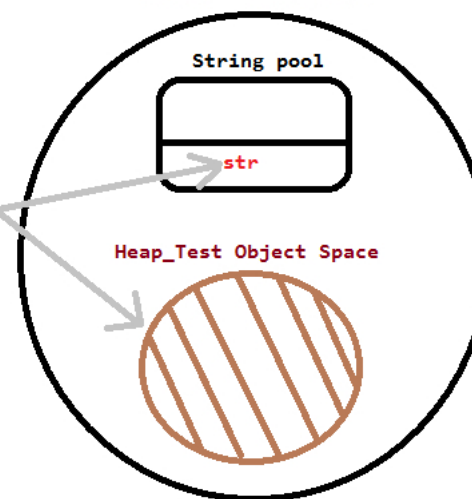
HOME / NHẬT KÝ CODING / NHẬT KÝ CUỘC SỐNG / CHUYỆN LINH TINH



String Pool là gì, hiểu thế nào cho đúng?

```
public class Heap_Test {  
    public static void main(String[] args)  
    {  
        Heap_Test reff = new Heap_Test();  
        reff.foo();  
    }  
    void foo() {  
        String str = "Heap memory space";  
        System.out.println(param);  
    }  
}
```

HEAP MEMORY



Trong ứng dụng Java, ở vùng nhớ Perm, nằm trong vùng nhớ Heap – có một nơi gọi là String Pool –

nơi này dùng để lưu trữ các biến được khai báo theo kiểu String.

Sở dĩ Java nó tạo ra vùng nhớ String Pool để tối ưu hóa việc lưu trữ và sử dụng vùng nhớ khi khai báo biến String, việc này giúp hạn chế tình trạng tràn bộ nhớ Java Heap Space.

Vì sao String Pool lại làm được như vậy ?

Chúng ta cùng tìm hiểu cách thức hoạt động của nó như thế nào nhé.

Đầu tiên chúng ta cùng dạo qua 1 tí về String.

Tóm tắt về String

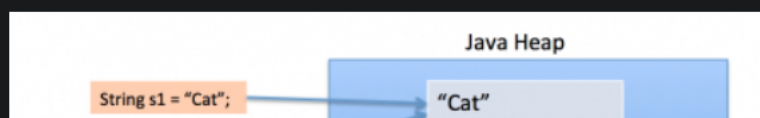
Trong lập trình java, chúng ta thường khai báo và sử dụng String. Khác với các biến nguyên thủy (Primitive type) khác như: char, byte, boolean, int, long, float, double, ... String nó là một kiểu biến tham chiếu. (Reference type)

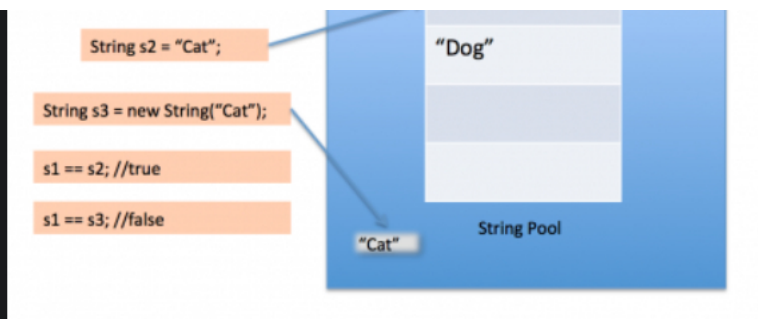
Như chúng ta đã biết, về cơ bản thì:

1. String là một immutable class (là một lớp bất biến, giá trị và vùng nhớ của biến được khai báo theo kiểu String sẽ không thay đổi). Vì sao String là một immutable class, mình sẽ giải thích cho các bạn ở bài sau nhé 😊
 2. Biến khai báo theo kiểu String khi được tạo ra sẽ được lưu vào String Pool ở bộ nhớ Perm của **Java Heap Space**
 3. Có 2 cách để khai báo 1 biến kiểu String
 - a. Sử dụng dấu ngoặc kép double quotes như khai báo các biến nguyên thủy thông thường:
String value = "Hello Java";
 - b. Sử dụng toán tử new: String value = new String("Hello Java");
- 2 kiểu khai báo trên đều tạo ra đối tượng String, nhưng giữa chúng có gì khác nhau ?

String Pool – cơ chế làm việc

Dưới đây là hình ảnh mô tả String pool ở trong bộ nhớ Java Heap Space và sự khác nhau của 2 cách khai báo:





- Khi sử dụng double quotes: Java sẽ truy cập vào String Pool, rồi tìm ở trong Pool ô nhớ nào có cùng giá trị với nó, nếu tìm thấy thì sẽ tham chiếu đến địa chỉ của ô nhớ đó, còn không thì nó sẽ tạo ô nhớ mới ở trong Pool rồi sẽ thực hiện việc tham chiếu.

Ở ví dụ trên, khai báo biến `String s1 = "Cat";` nó sẽ tạo mới trong String Pool một ô nhớ có giá trị là "Cat" vì lúc này ở trong Pool chưa có ô nhớ nào giá trị đó.

Lúc khai báo `String s2 = "Cat";` nó tìm thấy trong Pool giá trị một ô nhớ có giá trị là "Cat", s2 chỉ việc tham chiếu đến địa chỉ của ô nhớ này. Như vậy s2 đã có giá trị là "Cat" nhờ vào việc tham chiếu mà không cần phải tạo 1 ô nhớ khác.

Vì s1 và s2 cùng tham chiếu đến cùng một địa chỉ ô nhớ vùng nhớ heap nên `s1 == s2`

- Khi sử dụng toán tử new, thì Java sẽ không tạo ô nhớ mới ở bộ nhớ String Pool mà sẽ tạo ở Java Heap Space. Và khi đó nó sẽ **luôn luôn** tạo ô nhớ mới cho dù đã có sẵn những ô nhớ khác có cùng giá trị.

Ở ví dụ trên: `String s3 = new String("s3");` dòng này sẽ tạo 1 ô nhớ ở Java Heap Space có giá trị là "s3", biến s3 này sẽ tham chiếu địa chỉ của ô nhớ đó.

Và nếu có: `String s4 = new String("s3");` tương tự dòng này cũng sẽ tạo 1 ô nhớ khác ở Java Heap Space cũng có giá trị là "s3" (không giống ở trên là sẽ tìm ô nhớ nào có giá trị như vậy rồi tham chiếu đến nó), biến s4 này sẽ tham chiếu đến địa chỉ của ô nhớ mới.

Và vì s3 và s4 ko tham chiếu đến cùng địa chỉ ô nhớ ở vùng nhớ heap nên `s3 != s4`

=> Khai báo theo cách sử dụng toán tử new sẽ nhanh hơn cách dùng double quotes vì nó ko kiểm tra sự tồn tại của ô nhớ trong Heap space để tham chiếu đến như dùng double quotes. Thay vì đó nó sẽ luôn tạo những ô nhớ mới việc này dẫn đến dư thừa, nếu ko tính ý để quản lý thì sẽ dễ dàng tràn bộ nhớ ở Heap space.

Vấn đề tràn bộ nhớ Java Heap Space thì chỉ có ở máy tính có RAM thấp thôi, bây giờ máy đời mới rồi

vấn đề tràn bộ nhớ Java Heap Space thì chỉ có ở máy tính có bộ nhớ thấp thôi, bây giờ máy đời mới rồi nên cũng hiếm gặp lắm, đối với developer khi làm việc với vấn đề tạo ra rất nhiều nhiều String đối khi sẽ gặp phải vấn đề này. Link [này](#) sẽ hướng dẫn các bạn cách để giải quyết vấn đề tràn bộ nhớ

Vâng, đó là tất cả những gì mình muốn chia sẻ với các bạn về String Pool. Chắc hẳn các bạn đã nắm được cơ bản nó là gì, nguyên tắc hoạt động thế nào.

Đây là bài viết đầu tiên của mình, xin nhận gạch đá ạ 😊

Hì vọng những bài viết này sẽ mang đến cái nhìn rõ ràng và dễ hiểu nhất cho các bạn. Rất mong nhận được sự góp ý, thảo luận chân thành từ các bạn để những bài viết sau sẽ trở nên tốt hơn với mình và các bạn

Nguồn tham khảo:


1. [Java – Java Heap Space or Java Heap Memory](#)
2. [Java String Pool – Journal Dev blog](#)

Advertisements



SHARE THIS:

[Tweet](#) [Share 8](#)


[★ Like](#) 


One blogger likes this.




Quang Nguyễn

Chào các bạn! Mình là Quang - một developer. Từ thuở chập chững bắt đầu vào học code, mình đã có nhiều trăn trở và thắc mắc về kiến thức lập trình, về con đường nghề nghiệp, nhưng không có ai

 March 19, 2016

 **Nhật Ký Coding**

 **Chia Sẻ, double quotes, Heap Space, Java heap space, Java String Pool, Kinh**

giải đáp. Blog này là nơi mình chia sẻ kiến thức - kinh nghiệm mà bản thân mình đạt được trong quá trình học tập, làm việc và trải nghiệm. Hi vọng nó sẽ là người đồng hành, giải đáp phần nào những khúc mắc cho những bạn mới học code như mình ngày xưa. Luôn mong muốn sẽ giúp đỡ các bạn nhiều hơn, không chỉ ở phạm vi blog này. Thân ái!



new space, Java String, new operator, String Pool, Thủ Thuật

➔ Next post

7 thoughts on “String Pool là gì, hiểu thế nào cho đúng?”

[ADD YOURS](#)

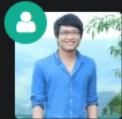


Khanh

February 11, 2018 at 7:31 pm

Cảm ơn bạn, bài viết rất hay và dễ hiểu 😊

★ Like



Quang Nguyễn

February 11, 2018 at 7:39 pm

Cảm ơn bạn

★ Like

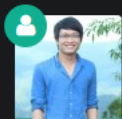


Quyet

September 13, 2018 at 4:40 am

Cảm ơn bạn nha, hôm nay đi phỏng vấn mới biết cái này, lúc đấy không trả lời được. 😊

★ Like



Quang Nguyễn

September 13, 2018 at 6:35 am

Quan trọng là bạn học được gì sau khi đi phỏng vấn nữa 😊

★ Like

Phan Huy

December 22, 2018 at 8:10 am

Cảm ơn bạn nhiều nha

★ Like



Tạ Anh Tú

December 26, 2018 at 8:31 am

Hay!

★ Like



Nguyễn D. Phương

July 7, 2019 at 2:05 pm

Bài viết của anh hay và dễ hiểu!! cảm ơn a

★ Like

Leave a Reply

Enter your comment here...

Search ...



CURRENT POSTS

- » Cơ chế lưu trữ dữ liệu của một chương trình Java
May 8, 2016
- » [Design Pattern] Bạn đã triển khai mẫu Singleton
như thế nào? April 17, 2016
- » Kiểu khai báo mảng – Mảng một chiều March 26,
2016
- » String Pool là gì, hiểu thế nào cho đúng? March
19, 2016

COMMENTS



Nguyễn D. Phương on String Pool là gì, hiểu
thế này...



Tạ Anh Tú on String Pool là gì, hiểu thế
này...

Phan Huy on String Pool là gì, hiểu thế này...



Quang Nguyễn on String Pool là gì, hiểu thế
này...



Quyet on String Pool là gì, hiểu thế này...

Singleton – Ng... on [Design Pattern] Bạn
đã triển...



Quang Nguyễn on String Pool là gì, hiểu thế
này...



Khanh on String Pool là gì, hiểu thế này...

[Design Pattern] Bạn... on [Design Pattern]
Bạn đã triển...

GẦN ĐÂY

- » Cơ chế lưu trữ dữ liệu của một chương trình Java
May 8, 2016
- » [Design Pattern] Bạn đã triển khai mẫu Singleton
như thế nào? April 17, 2016
- » Kiểu khai báo mảng – Mảng một chiều March 26,
2016
- » String Pool là gì, hiểu thế nào cho đúng? March
19, 2016

LƯU TRỮ

Select Month



CHUYÊN MỤC

Select Category ▼

AUTHORS



Quang Nguyễn

Cơ chế lưu trữ dữ liệu của một chương trình Java

[Design Pattern] Bạn đã triển khai mẫu Singleton như thế nào?

Kiểu khai báo mảng – Mảng một chiều

String Pool là gì, hiểu thế nào cho đúng?

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept