

String, StringBuilder and StringBuffer

1. Kiến thức

- String

- + Là immutable (bất biến): mỗi lần thay đổi nội dung chuỗi, Java tạo ra một object mới trong vùng nhớ.
- + Mỗi phép nối + là tạo object mới → tốn nhiều thời gian và bộ nhớ nếu lặp nhiều lần.
- + Nên dùng cho chuỗi ngắn, không lặp.

- StringBuilder

- + Là mutable: có thể thay đổi nội dung chuỗi mà không tạo ra object mới.
- + Không thread-safe → dùng khi chỉ có 1 luồng truy cập.
- + Nhanh hơn String rất nhiều trong vòng lặp lớn.
- + Sử dụng khi thao tác chuỗi nhiều, chỉ 1 luồng.

- StringBuffer

- + Giống StringBuilder nhưng thread-safe (các phương thức được đồng bộ hóa).
- + Vì đồng bộ nên chậm hơn StringBuilder, nhưng an toàn khi dùng với nhiều luồng (multithread).
- + Sử dụng khi thao tác chuỗi nhiều, đa luồng.

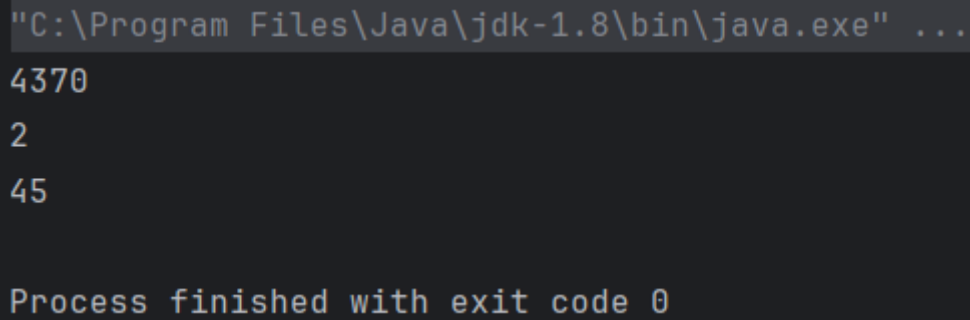
2. Thử nghiệm

- So sánh String và StringBuilder:

```
1 import java.util.Random;
2 public class ConcatenationInLoops {
3     public static void main(String[] args){
4         Random r = new Random(123);
5         long start = System.currentTimeMillis();
6         String s = "";
7         for(int i =0; i < 65536; i++)
8             s += r.nextInt(2);
9         System.out.println(System.currentTimeMillis() -
10 start);
11
12         r = new Random(123);
13         start = System.currentTimeMillis();
14         StringBuilder sb = new StringBuilder();
15         for (int i = 0; i < 65536; i++)
16             sb.append(r.nextInt(2));
17         s = sb.toString();
18         System.out.println(System.currentTimeMillis() -
19 start);
```

```
20
21     r = new Random(123);
22     start = System.currentTimeMillis();
23     StringBuffer sbf = new StringBuffer();
24     for (int i = 0; i < 65536; i++)
25         sbf.append(r.nextInt(2));
26     s = sbf.toString();
27     System.out.println(System.currentTimeMillis() - start);
28 }
```

+Kết quả:



```
"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
4370
2
45

Process finished with exit code 0
```

- ⇒ Thời gian chạy khi dùng String lớn hơn rất nhiều so với StringBuilder do String liên tục tạo Object mới còn StringBuilder tiếp tục nối vào chuỗi ban đầu mà không tạo Object mới.
- ⇒ StringBuilder thao tác chuỗi nhiều, đa luồng nên chạy lâu hơn StringBuilder một chút nhưng vẫn nhanh hơn rất nhiều so với String.