

BÁO CÁO XỬ LÝ CHUỖI: STRING, STRING BUFFER, STRINGBUILDER

1. Môi trường thực hiện

- **Hệ điều hành:** Windows 11 Home Single Language
- **Kiểu hệ thống:** 64-bit operating system
- **Bộ vi xử lý (CPU):** 13th Gen Intel(R) Core(TM) i7-1355U @ 1.70 GHz
- **Bộ nhớ RAM:** 16.0 GB
- **Ổ cứng:** SSD 512 GB
- **Java version:**
java version "22.0.2" 2024-07-16
Java(TM) SE Runtime Environment (build 22.0.2+9-70)
Java HotSpot(TM) 64-Bit Server VM (build 22.0.2+9-70, mixed mode, sharing)
- **Eclipse IDE:** 2024-12 (4.34.0)
- **JVM:** Java HotSpot(TM) 64-Bit Server VM – Version 22.0.2+9-70 – Oracle Corporation

2. So sánh thời gian chạy: String, StringBuilder, StringBuffer

a. Code:

```
// Test String "+"
Random r = new Random(123);
long start = System.currentTimeMillis();
String s = "";
for (int i = 0; i < 65536; i++){
    s += r.nextInt(2);
}
System.out.println("String (+): " + (System.currentTimeMillis() - start) + " ms");
```

```
// Test StringBuilder
r = new Random(123);
start = System.currentTimeMillis();
StringBuilder sb = new StringBuilder();
for (int i = 0; i < 65536; i++) {
    sb.append(r.nextInt(2));
}
s = sb.toString();
System.out.println("StringBuilder: " + (System.currentTimeMillis() - start) + " ms");
```

```
// Test StringBuffer
r = new Random(123);
start = System.currentTimeMillis();
StringBuffer sbf = new StringBuffer();
for (int i = 0; i < 65536; i++) {
    sbf.append(r.nextInt(2));
}
s = sbf.toString();
System.out.println("StringBuffer: " + (System.currentTimeMillis() - start) + " ms");
```

b. Kết quả thời gian chạy

```
String (+): 390 ms
StringBuilder: 2 ms
StringBuffer: 3 ms
```

c. *Nhận xét*

- **String với toán tử +:** Vì String là immutable, mỗi lần nối tạo ra một đối tượng String mới → bộ nhớ cấp phát liên tục → Tốn thời gian và bộ nhớ, tốc độ chậm.
- **StringBuilder:** Vì StringBuilder là một lớp mutable và non-synchronized nên khi nối chuỗi sử dụng bộ nhớ đệm nội bộ, không cần tạo đối tượng mới nhiều lần → Tốc độ nhanh, hiệu quả trong môi trường đơn luồng.
- **StringBuffer:** Tương tự StringBuilder nhưng được synchronized, tức thread-safe – an toàn khi có nhiều luồng truy cập cùng lúc → Tốc độ nhanh, chậm hơn StringBuilder (do phải xử lý đồng bộ), phù hợp khi xử lý đa luồng.

3. Đánh giá

	String (+)	StringBuilder	StringBuffer
<i>Hiệu năng</i>	Rất chậm	Rất nhanh	Nhanh
<i>Bộ nhớ</i>	Tốn nhiều bộ nhớ	Tiết kiệm (dùng bộ đệm)	Tiết kiệm (dùng bộ đệm)
<i>Tính bất biến</i>	Có	Không	Không
<i>Thread-safe</i>	Không	Không	Có (synchronized)
<i>Dùng khi nào</i>	Chương trình đơn giản, ít vòng lặp	Xử lý chuỗi lớn, môi trường đơn luồng	Xử lý trong môi trường đa luồng