

So sánh xử lý xâu

1. Môi trường chạy

Phần mềm : java version “23.0.1”

Cấu hình máy :

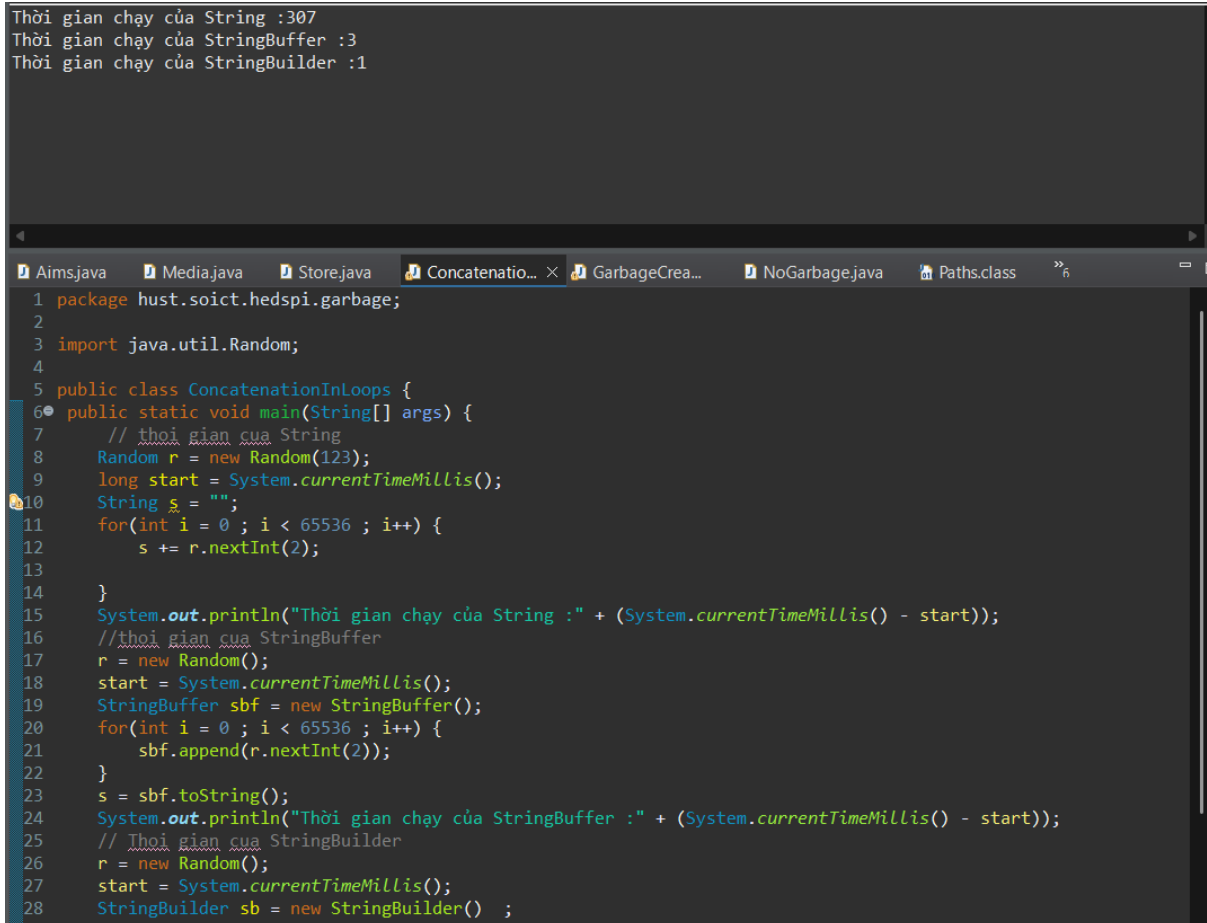
+CPU : intel core i7 11800

+RAM : 16GB

2. Report về report về running time

+random

```
Thời gian chạy của String :307
Thời gian chạy của StringBuffer :3
Thời gian chạy của StringBuilder :1
```



The screenshot shows an IDE with a dark theme. At the top, a console window displays the results of a benchmark: 'Thời gian chạy của String :307', 'Thời gian chạy của StringBuffer :3', and 'Thời gian chạy của StringBuilder :1'. Below the console, the 'Concatenatio...' file is open, showing Java code. The code defines a 'ConcatenationInLoops' class with a 'main' method. It uses a 'Random' object to generate random integers and concatenates them into a 'String', 'StringBuffer', and 'StringBuilder'. The 'String' concatenation is the slowest, taking 307 milliseconds. The 'StringBuffer' and 'StringBuilder' concatenations are much faster, taking 3 and 1 millisecond respectively. The code is as follows:

```
1 package hust.soict.hedspi.garbage;
2
3 import java.util.Random;
4
5 public class ConcatenationInLoops {
6     public static void main(String[] args) {
7         // thời gian của String
8         Random r = new Random(123);
9         long start = System.currentTimeMillis();
10        String s = "";
11        for(int i = 0 ; i < 65536 ; i++) {
12            s += r.nextInt(2);
13        }
14        System.out.println("Thời gian chạy của String :" + (System.currentTimeMillis() - start));
15        //thời gian của StringBuffer
16        r = new Random();
17        start = System.currentTimeMillis();
18        StringBuffer sbf = new StringBuffer();
19        for(int i = 0 ; i < 65536 ; i++) {
20            sbf.append(r.nextInt(2));
21        }
22        s = sbf.toString();
23        System.out.println("Thời gian chạy của StringBuffer :" + (System.currentTimeMillis() - start));
24        // Thời gian của StringBuilder
25        r = new Random();
26        start = System.currentTimeMillis();
27        StringBuilder sb = new StringBuilder() ;
28        for(int i = 0 ; i < 65536 ; i++) {
```

```
Thòi gian chạy của String :320
Thòi gian chạy của StringBuffer :2
Thòi gian chạy của StringBuilder :2

4
Aims.java Media.java Store.java Concatenatio... × GarbageCrea... NoGarbage.java Paths.class 6
1 package hust.soict.hedspi.garbage;
2
3 import java.util.Random;
4
5 public class ConcatenationInLoops {
6     public static void main(String[] args) {
7         // thòi gian của String
8         Random r = new Random(123);
9         long start = System.currentTimeMillis();
10        String s = "";
11        for(int i = 0 ; i < 65536 ; i++) {
12            s += r.nextInt(2);
13        }
14        System.out.println("Thòi gian chạy của String :" + (System.currentTimeMillis() - start));
15        //thòi gian của StringBuffer
16        r = new Random();
17        start = System.currentTimeMillis();
18        StringBuffer sbf = new StringBuffer();
19        for(int i = 0 ; i < 65536 ; i++) {
20            sbf.append(r.nextInt(2));
21        }
22        s = sbf.toString();
23        System.out.println("Thòi gian chạy của StringBuffer :" + (System.currentTimeMillis() - start));
24        // Thòi gian của StringBuilde
25        r = new Random();
26        start = System.currentTimeMillis();
27        StringBuilder sb = new StringBuilder() ;
28        for(int i = 0 ; i < 65536 ; i++) {
29            sb.append(r.nextInt(2));
30        }
31        System.out.println("Thòi gian chạy của StringBuilder :" + (System.currentTimeMillis() - start));
32    }
33}
```

+Với file txt rác :

Thời gian khi dùng String với + là :6

```
Aimsjava Media.java Store.java Concatenatio... GarbageCrea... NoGarbage.java Paths.class
1 package hust.soict.hedspi.garbage;
2
3 import java.io.IOException;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 public class GarbageCreator {
8     public static void main(String[] args) throws IOException {
9         String filename = "D:\\\\JAVA CODE\\\\String.txt";
10        byte[] inputBytes = { 0 };
11        long startTime, endTime;
12
13        inputBytes = Files.readAllBytes(Paths.get(filename));
14        startTime = System.currentTimeMillis();
15        String outputString = "";
16        for(byte b : inputBytes) {
17            outputString += (char)b;
18        }
19        endTime = System.currentTimeMillis();
20        System.out.println("Thời gian khi dùng String với + là | :"+ (endTime - startTime));
21    }
22 }
```

Thời gian của StringBuilder là : 0

```
1 package hust.soict.hedspi.garbage;
2
3 import java.io.IOException;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 public class NoGarbage {
8     public static void main(String[] args) throws IOException {
9         String filename = "D:\\\\JAVA CODE\\\\String.txt";
10        byte[] inputBytes = { 0 };
11        long startTime, endTime;
12
13        inputBytes = Files.readAllBytes(Paths.get(filename));
14        startTime = System.currentTimeMillis();
15        StringBuilder outputStringBuilder = new StringBuilder();
16        for(byte b : inputBytes) {
17            outputStringBuilder.append((char)b);
18        }
19        endTime = System.currentTimeMillis();
20        System.out.println("Thời gian của StringBuilder là : " + (endTime - startTime));
21    }
22 }
```