

## **Exercise 4 - Report**

### **I. Exercises**

For this report, I was instructed to create a short report in the Assignment 4 for the course 1DV507. The purpose of this report is to show that `StringBuilder` is faster than string concatenation.

### **II. Experimental Setup**

All experiments were done on HP laptop with an Intel Core i5 processor, 8<sup>th</sup> generation with 8GB of memory. I used IntelliJ IDEA 2018.3.4 to run the code to do the testing.

I used the clock `System.currentTimeMillis()` in all the measurements

### III. Results Concatenation

String Concatenation - With 1 character

Time taken (ms)	Amount of concatenation	Length
1000	121826	121826
1000	121251	121251
1000	120699	120699
1002	119071	119071
1000	121293	121293
1008	120826	120826
1013	119790	119790
1002	120639	120639
1015	122782	122782
1015	123869	123869
<b>Average Run</b>		
1005	121204	121204

String Concatenation - With 80 characters

Time taken (ms)	Amount of concatenation	Length
1003	15572	1245760
1000	15529	1242320
1000	15367	1229360
1003	15355	1228400
1012	15307	1224560
1000	15403	1232240
1015	15301	1224080
1000	15308	1224640
1002	15354	1228320
1015	15396	1231680
<b>Average Run</b>		
1005	15389	1231136

### StringBuilder - With 1 character

Time taken (ms)	Amount of concatenation	Length
1000	224763790	224763790
999	225344295	225344295
1007	201870639	201870639
1014	228353578	228353578
1014	222109081	222109081
1007	220058918	220058918
1014	220777167	220777167
1000	215121018	215121018
1000	217233655	217233655
1008	221195687	221195687
<b>Average Run</b>		
1006	219682782	219682782

### StringBuilder - With 80 characters

Time taken (ms)	Amount of concatenation	Length
1320	8604758	688380640
1662	8598324	687865920
1555	8898722	675421451
1537	8589416	646513245
1662	8486511	644521452
1555	8321677	646514515
1662	8151565	674851485
1662	8164152	696532525
1555	8848521	674128525
<b>Average Run</b>		
1574	8518182	670525528

## IV. Results InsertionSort

Insertion sort with int Array

Run Number	Time (ms)	Array Length
1	1015	78576
2	1024	85044
3	975	82782
4	999	86200
5	988	81817
6	1017	78576
7	976	82782
8	1009	73584
<b>Average</b>	<b>1000</b>	<b>81170</b>

Insertion sort with String Array

Run Number	Time (ms)	Array Length
1	993	20692
2	1003	20968
3	1024	21802
4	994	19902
5	991	20169
6	1017	20431
7	984	20988
8	978	18411
<b>Average</b>	<b>998</b>	<b>20420</b>

## V. String Concatenation vs StringBuilder

In java, StringBuilder is going faster than String because strings are not changeable. When we do the concatenation, all the strings need to be loaded into memory. Then, a new string is created by adding the loaded strings from the memory. Then, the new string will be assigned to the variable.

For the StringBuilder, its object is changeable. The added strings will be added to the StringBuilder without extra strings in memory.

This is the reason why the StringBuilder is faster than the String Concatenation.