

Linnaeus University

1DV507 – Lab4

Student: Robin Stempa

Student ID: rs222nv@student.lnu.se



In exercise two we are supposed to write a short report about our time experiments.

The experiments had to do with time efficiency with string concatenation vs string builder class. In Java there are two ways of concatenating strings, one by using the + sign to increase the length of the string by adding another one to it. The second way is by using the class String builder to append more strings and then calling the toString method.

The task was to find the fastest approach by measuring how many concatenations, and the length of the final string, each of them can do in one second, by adding short strings together containing only one character and then by adding long strings together containing 80 characters.

This experiment was done by using two different methods, using both the + operator and string builder. Inside each method a while loop was used for doing concatenation and appending as long time, measured in Nano seconds was less or equal to one second.

Method	Length of String	Number of concatenations
Concatenate short string	68820	68820
Concatenate long string	475629	5467
String builder short string	19712195	19712195
String builder long string	661983870	7609010

As we can see using string builder has better results with more concatenations and longer strings. This means that string builder is a more effective way of concatenating strings in a Java program, if it has to be done for many concatenations in a loop or so.

This is because string builder does not need to create a new string object every time a concatenation is done.

In assignment three a task was to implement four different sorting algorithms. One version of insertion sort for integers and one for strings and the same for merge sort. This time the task was to use these implementations to see how many strings and integers could be sorted in one second.

This experiment was done by setting up two different arrays of length ten by adding random integers and random strings. This so that the arrays are not sorted from the beginning.

This time four while loops are used and executed as long as the time was less or equal to one second, measured in Nano seconds.

Sorting algorithm	Number of elements sorted
Merge sort integers	31040700
Merge sort strings	26037410
Insertion sort integers	617466560
Insertion sort strings	419874990

From the results of this experiment the number of elements sorted by insertion sort are many more than with merge sort. Although if the size of the array being sorted was made larger, around the length of 100 000 merge sort was much faster than selection sort.