Time Measurements

This is a short report for the exercise 3 in assignment 4 in the course 1DV507 - Programing and Data Structures. We were instructed to perform a test on using the operator + is faster than using the Stringbuilder append.
It shows quite clearly that stringbuilders append is much faster than the operator +.

1. Exercises

The experiment was to check if concatenations using + or stringbuilder is faster. It was decided that we should find out how many concatenations is made under 1 seconds. Also there should be two test one with a string that is containing only one character (short string) and the other where there are 80 characters (long string).

2. Experimental Setup

All experiments was made on a dell xps 15 with an i7 processor (2.21GHz) and 16GB of ram. The JVM needed to have more memory during these experiments. So the following VM commands was used -Xmx4096m -Xmx4096m. All the other applications that was not needed for the computer to run was turned off so we can get the most accurate result. The experiment was set up so that there was a try-and-error approach to finding how many concatenations is made in one second. Than a interval was made than a close approximation was made to one second.

3. String Concat

The table below is the result for both the short string and long string. The columns nConcat is the amount of concatenations and length is the length of the final string. That could be performed in one second.

	Short string	Long string

Technique	nConcat	Length	nConcat	Length
StringBuilder	95*10^6	95*10^6	6.6*10^6	533*10^6
+ Operator	310000	310000	4.0*10^3	325*10^3

The result is quite clear Stringbuilder is much faster than using the + Operator. If you need to construct a very long string fast then stringbuilder is the way to go.
The reason that stringbuilder is so much faster than + operator is that for each concatenation str1+str2 it needs to make a new string with the length of both the string. Stringbuilder uses a

much different approach. It uses a an array that just appends the string to the end of the array. Which means it doesn't need to make a string each time it appends something.