Linnéuniversitetet
Kalmar Växjö

Report

Time

Författare: Elise Anjel
Termin: VT2018
Ämne: Programming and Data Structures
Kurskod: 1DV507

Linnéuniversitetet
Kalmar Växjö

Innehållsförteckning

Innehållsförteckning **2**

Exercise 1 **3**

Method 3

Results 3

Exercise 2 **4**

Method 4

Results 4

String concatenations vs StringBuilder **5**

Exercise 1

This exercise was to measure how many string concatenations and StringBuilder appends can be performed in one second.

Method

The experiment was done by writing an application that performs string concatenations until one second has passed storing the results and then performs StringBuilder appends until one second has passed and storing the results. The application was designed to perform multiple runs and calculate the average results.

The results were measured and the average was calculated of 5 runs.

The application was run on a machine with a Intel Core i5-2500K CPU @ 4.0 GHz running Windows 10.

Results

	Concats/Appends	Final string length
Concat short	88333	88333
Concat long	7291	583328
Append short	89989906	89989906
Append Long	19468680	1557494400

Exercise 2

The exercise was to measure the size of an array that can be sorted by the author of this report's insertion sort algorithms (Integer and String).

Method

An application was written that creates integer arrays and sorts them, the size changes until an array that takes ≈1 second to sort is produced. This is repeated for string arrays.

The results were measured and the average was calculated of 5 runs.

The application was run on a machine with a Intel Core i5-2500K CPU @ 4.0 GHz running Windows 10.

Results

Insertion sort	**Array size**	**Time (seconds)**
Integer array	55638	0.9984
String array	20507	1

String concatenations vs StringBuilder

StringBuilder is faster because in java, strings are unchangeable.
When a concatenation occurs, all the strings are loaded into memory, a new string is created by adding the loaded strings from memory and then the new string is assigned to the variable. The old string is marked for the garbage collector.
The StringBuilder object is changeable, therefore the added string will be added to the StringBuilder, without extra strings in memory that the garbage collector has to deal with.