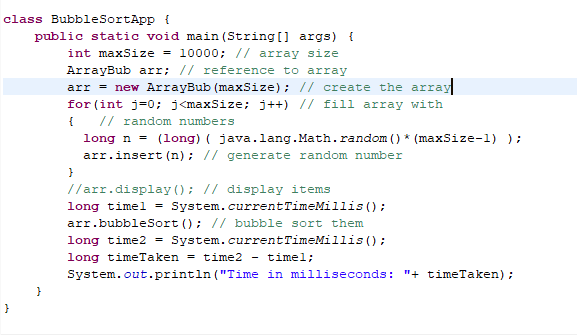
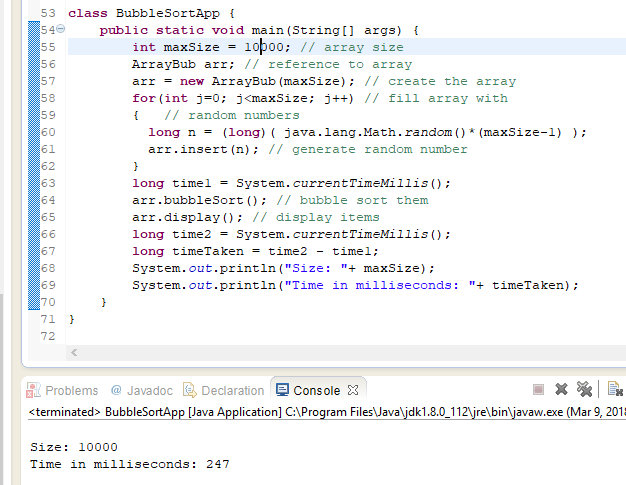
**BUBBLE SORT**

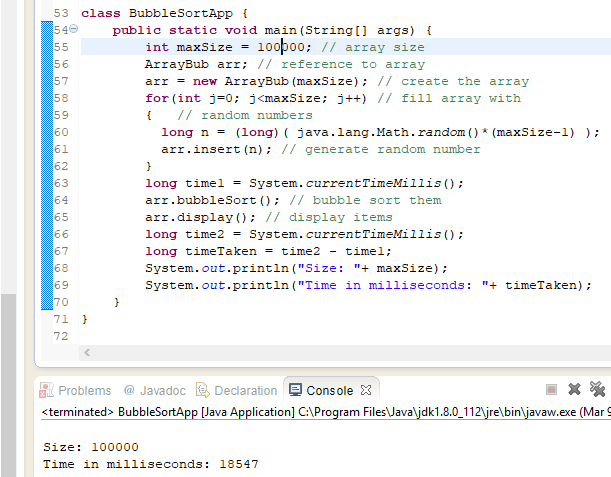
1. **Paragraph and Screenshot Number-1:** Before and after the sorting algorithm the random numbers sorted well but because of member size its not shown up on console. Bubble sort swaps elements over and over if they are not in right order.



1. **Paragraph and Screenshot Number-2**: Here, I worked with current time millisecond method to get the sorting time of specific number of elements This screenshot portraits the total time taken to sort 10,000 elements. The time is 247 milliseconds which is .247 seconds. Where 1 second equal 1000 milliseconds.

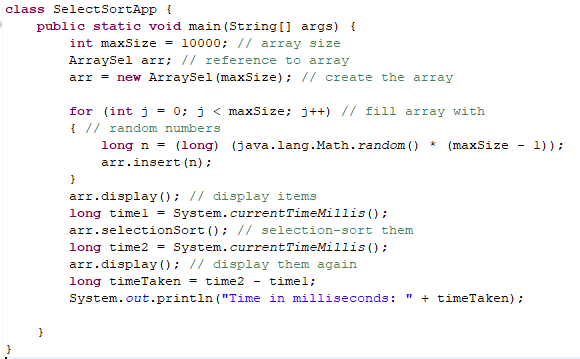


1. **Paragraph and Screenshot Number-3**: This screenshot shows the total time taken to execute 100,000 elements. Since you said it should take less than 30 seconds. The result is 18547 milliseconds. Which is almost 19 seconds. This is good compare to run small number of elements.

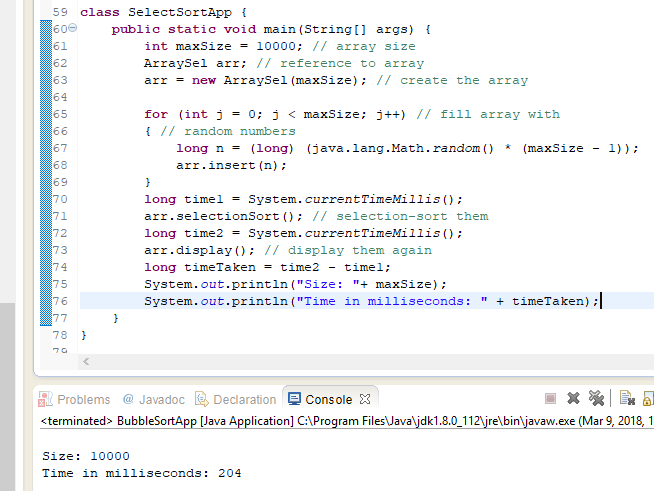


**SELECTION SORT**

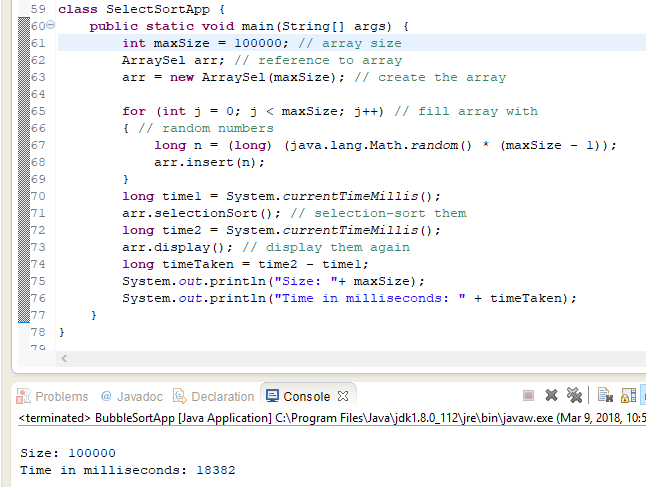
1. **Paragraph and Screenshot Number-1:** Before and after the sorting algorithm the random numbers sorted well but because of big member size, it’s not shown up on console. Selection sorts finds minimum element order from unsorted parts and putting that element at the beginning.



1. **Paragraph and Screenshot Number-2**: Here, I worked with current time millisecond method to get the sorting time of specific number of elements This screenshot portraits the total time taken to sort 10,000 elements. The time is 204 milliseconds which is .204 seconds. Where 1 second equal 1,000 milliseconds. This is faster compare to the bubble sort.

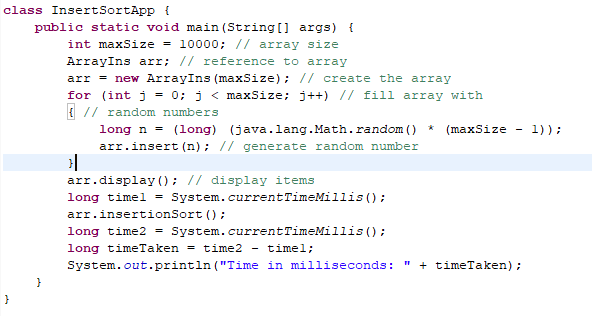


1. **Paragraph and Screenshot Number-3**: This screenshot shows the total time taken to execute 100,000 elements. Since you said it should take less than 30 seconds. The execution time is 18382 milliseconds. Which is almost close to 18 seconds. This faster than bubble sort compares to handling same number of items.

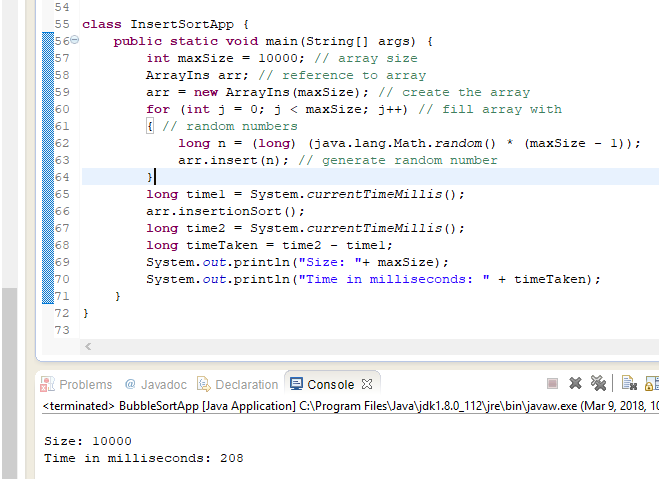


**INSERTION SORT**

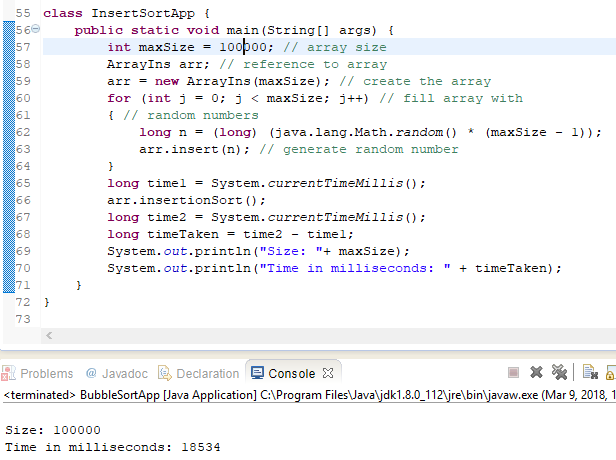
1. **Paragraph and Screenshot Number-1:** Before and after the sorting algorithm the random numbers sorted well but because of big member size, it’s not shown up on console. Insertion sort iterates through the list by consuming one input element at each repetition and growing a sorted output list. On a repetition, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remain.



1. **Paragraph and Screenshot Number-2**: Here, I worked with current time millisecond method to get the sorting time of specific number of elements This screenshot portraits the total time taken to sort 10,000 elements. The time is 208 milliseconds which is .208 seconds. Where 1 second equal 1,000 milliseconds. This is similar to the selection sort.



1. **Paragraph and Screenshot Number-3**: This screenshot represents the total time taken to execute 100,000 elements. Since you said it should take less than 30 seconds. The execution time is 18534 milliseconds. Which is almost close to 18.5 seconds. This faster than bubble sort compares to handling same number of elements or data.



**SUMMARY AND CONCLUSION**

1. **Paragraph Comparing Bubble, Insertion and Selection Sort Algorithms:** In comparison of these three sorting in-term of algorithms they are kind of same in big O explanation. But things are different in term of doing or sorting the elements. Among these three-insertion sort much simpler to understand and fast enough to handle large amount of data.
2. **Paragraph Comparing Bubble, Insertion and Selection Sort Timing:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Bubble** | **Selection** | **Insertion** |
| 1,000 | .32 Seconds | .15 Seconds | .15 Seconds |
| 10,000 | .247 Seconds | .204 Seconds | .208 Seconds |
| 100,000 | 19 Seconds | 18 Seconds | 18 Seconds |

1. **Paragraph Discussing the necessity to look at 10,000 and even larger arrays when testing timing:**

Finally. I think insertion sort the most efficient to handle large or small amount data compare to other sorting. If we look at the compared table, we will realize the insertion sorting is the more fastest among the three.