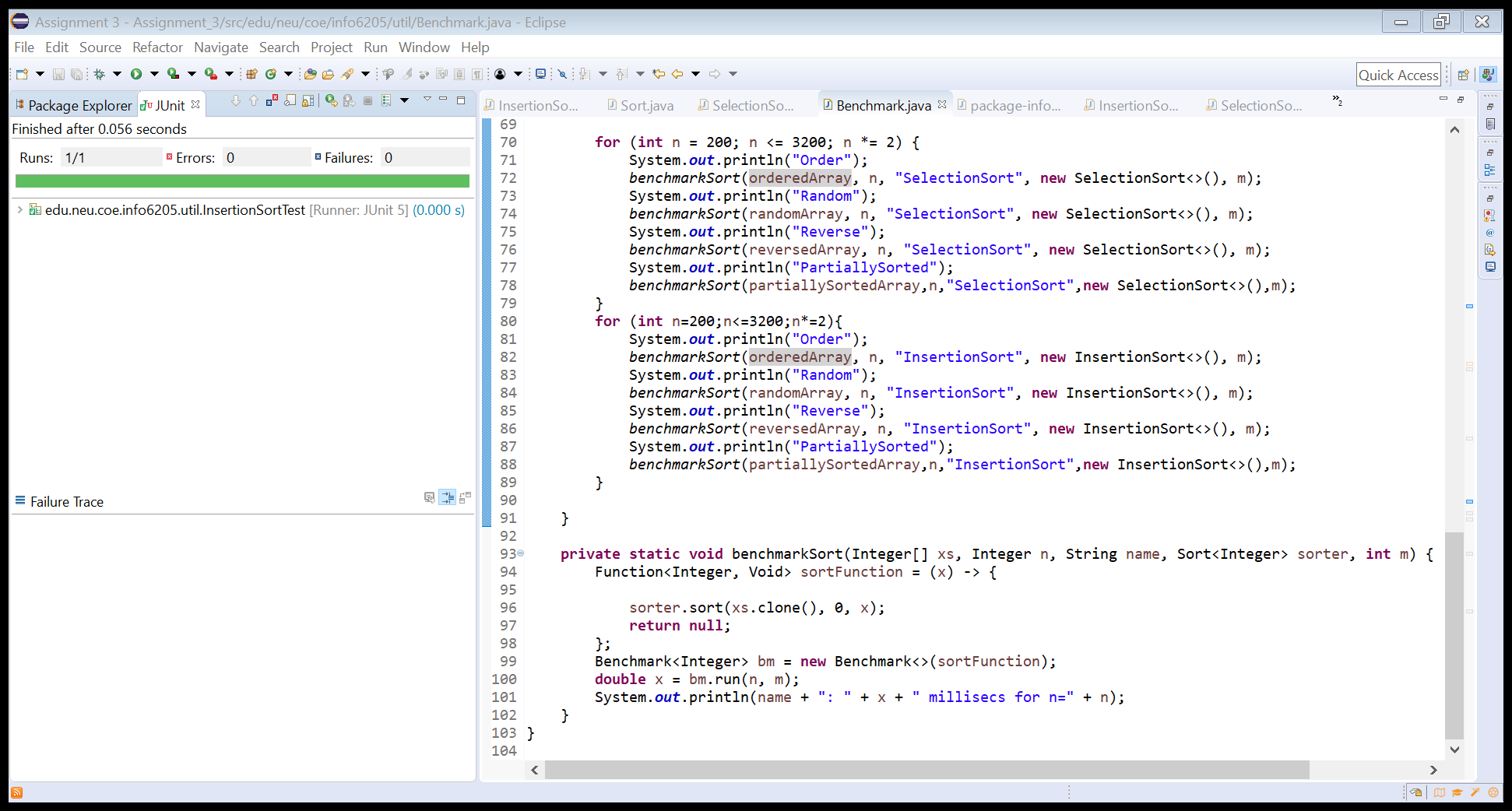
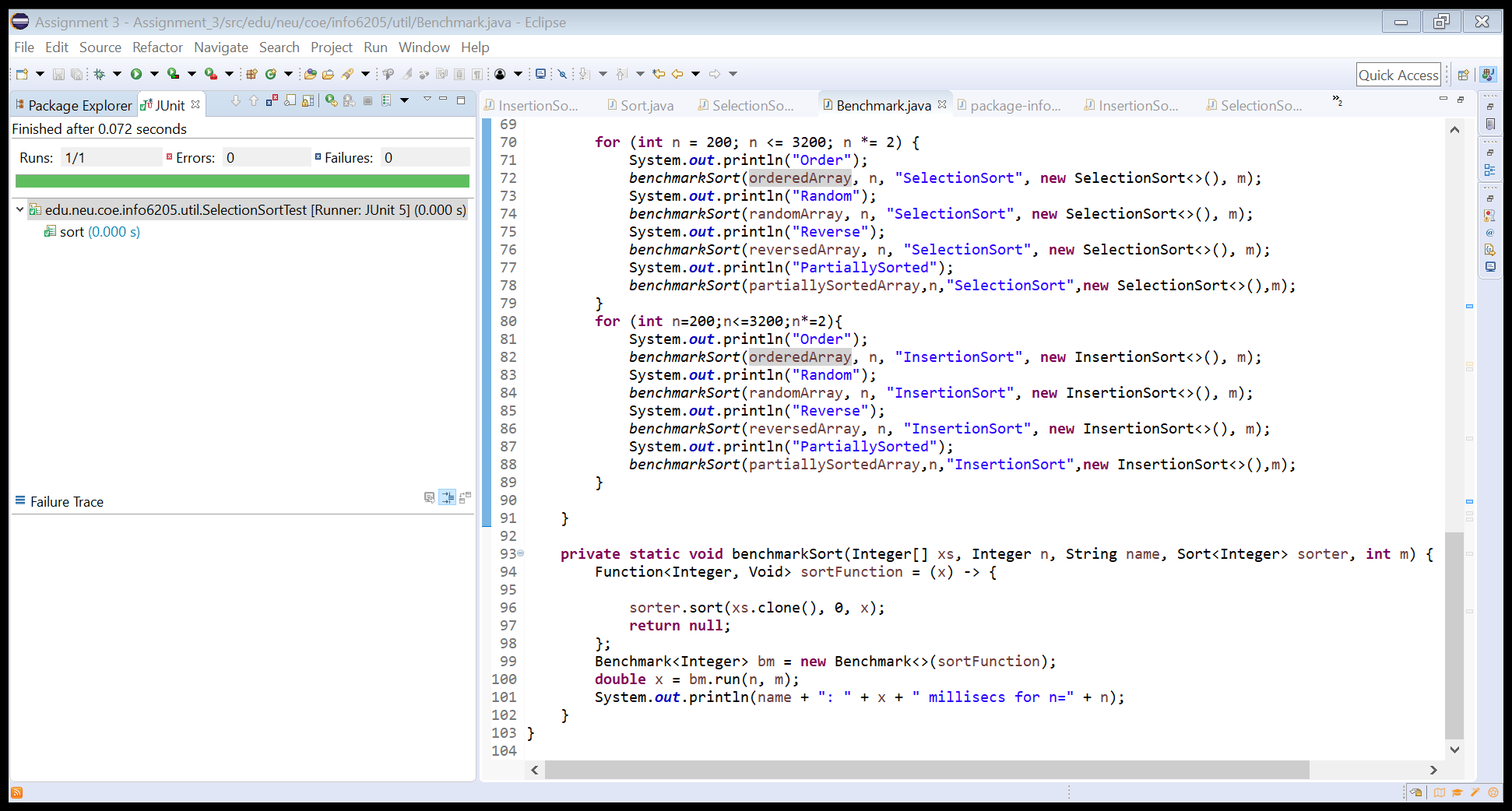
1. Run unit test

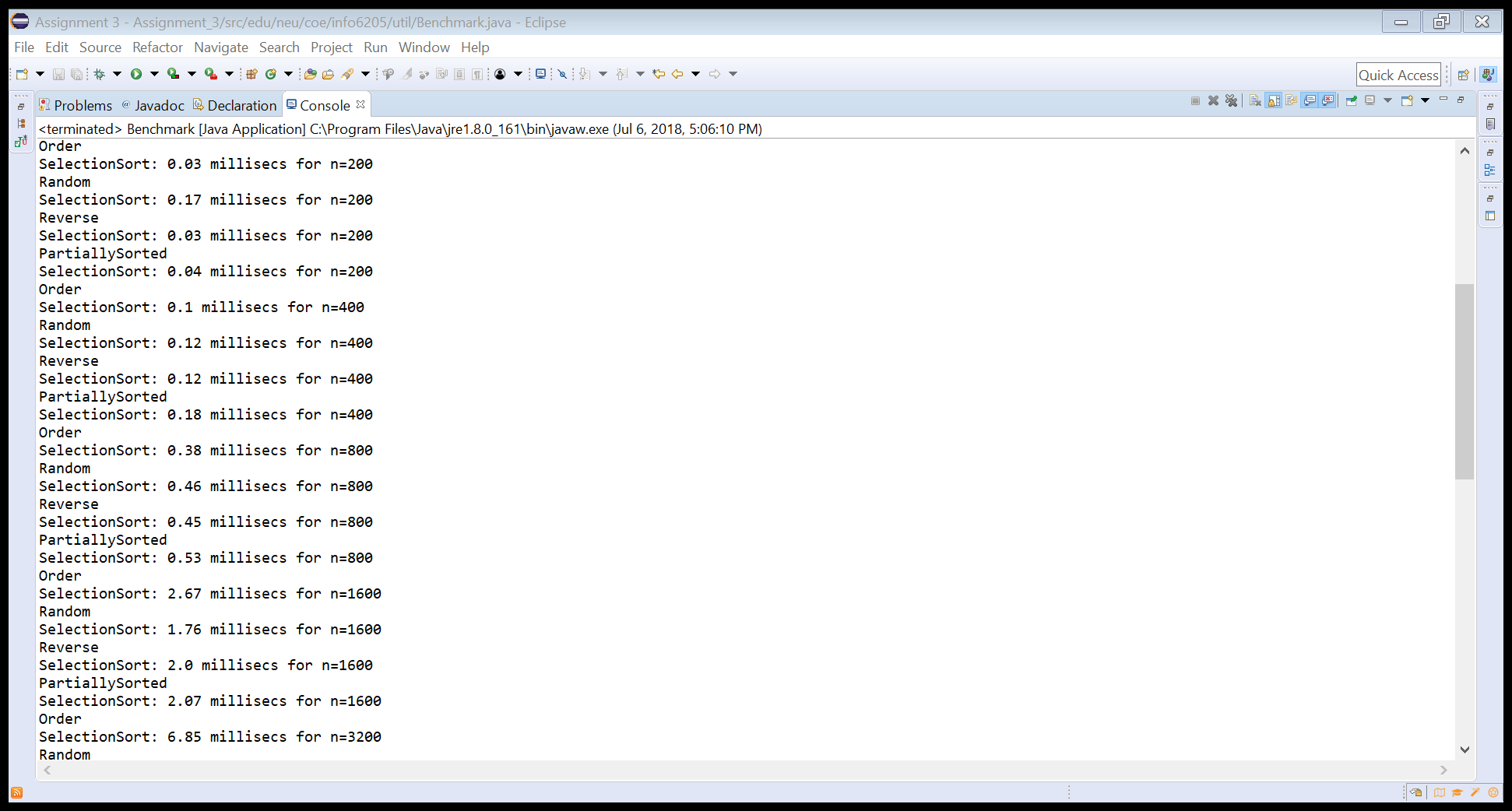
Insertion sort:

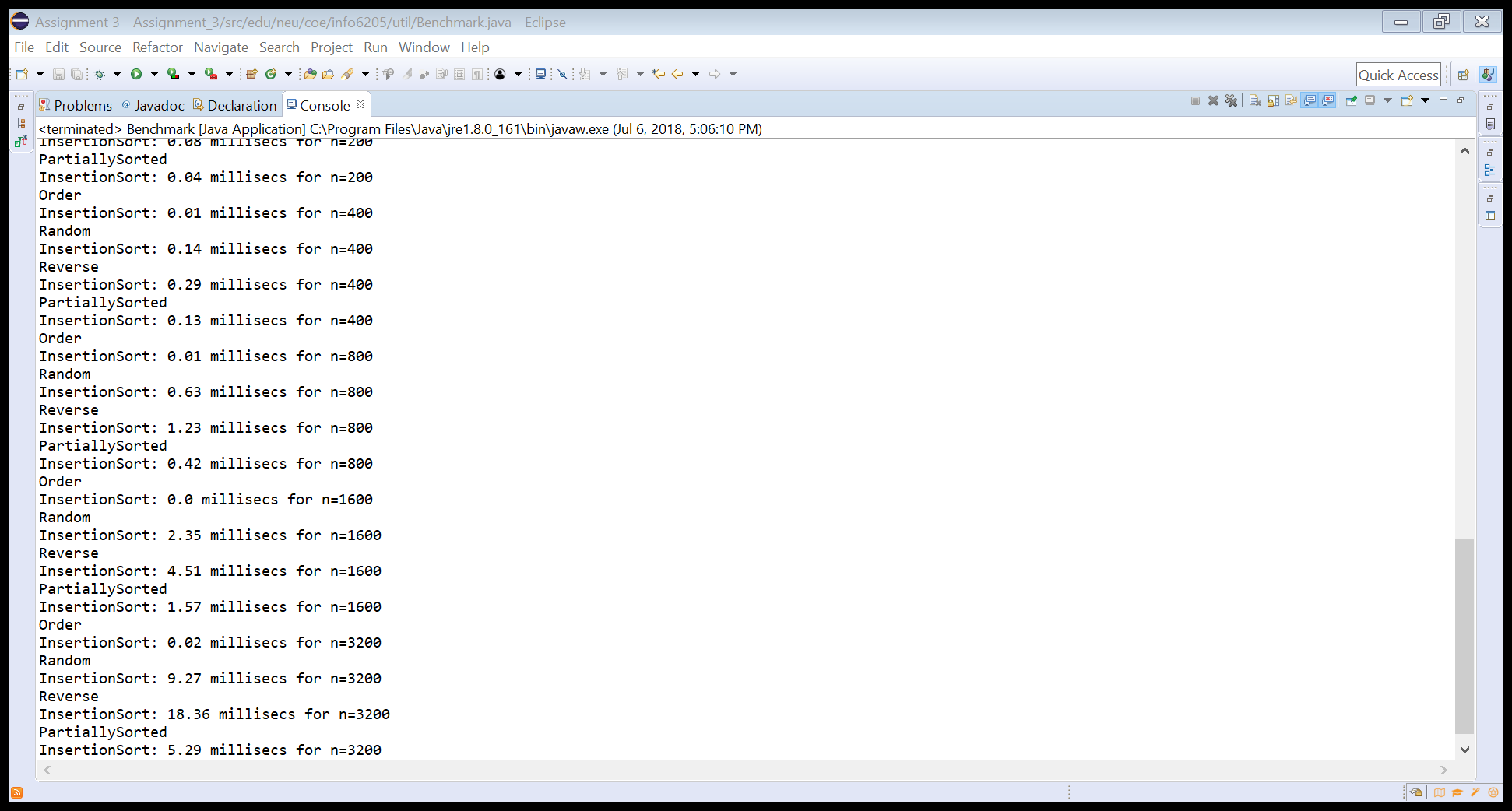


Selection sort:



1. Run main ()





1. Observation

After the system has warmed up, the data begins to be stable.

Data output sample:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n=200 | | | | n=400 | | | | n=800 | | | | n=1600 | | | | n=3200 | | | |
| O | Ra | Re | Pa | O | Ra | Re | Pa | O | Ra | Re | Pa | O | Ra | Re | Pa | O | Ra | Re | Pa |
| Selection | 0.03 | 0.17 | 0.03 | 0.04 | 0.10 | 0.25 | 0.11 | 0.13 | 0.32 | 0.61 | 0.44 | 0.45 | 1.26 | 1.84 | 1.79 | 1.70 | 4.94 | 6.74 | 8.15 | 6.51 |
| Insertion | 0.00 | 0.11 | 0.08 | 0.04 | 0.01 | 0.20 | 0.28 | 0.13 | 0.01 | 0.63 | 1.12 | 0.41 | 0.01 | 2.41 | 4.40 | 1.56 | 0.01 | 9.16 | 18.24 | 5.84 |

For the same “n”, selection sort will take similar time, even if the sequence is ordered, reverse or random. Insertion sort is affected by the sequence’s order. If data is ordered, it takes linear/very less time. If data is reversed, it takes more time(worst case). It is about 2 times than random data.

When n is larger, selection sort is faster than insertion sort when the data is reversed. As n doubles, the time increases 4 times than the former.