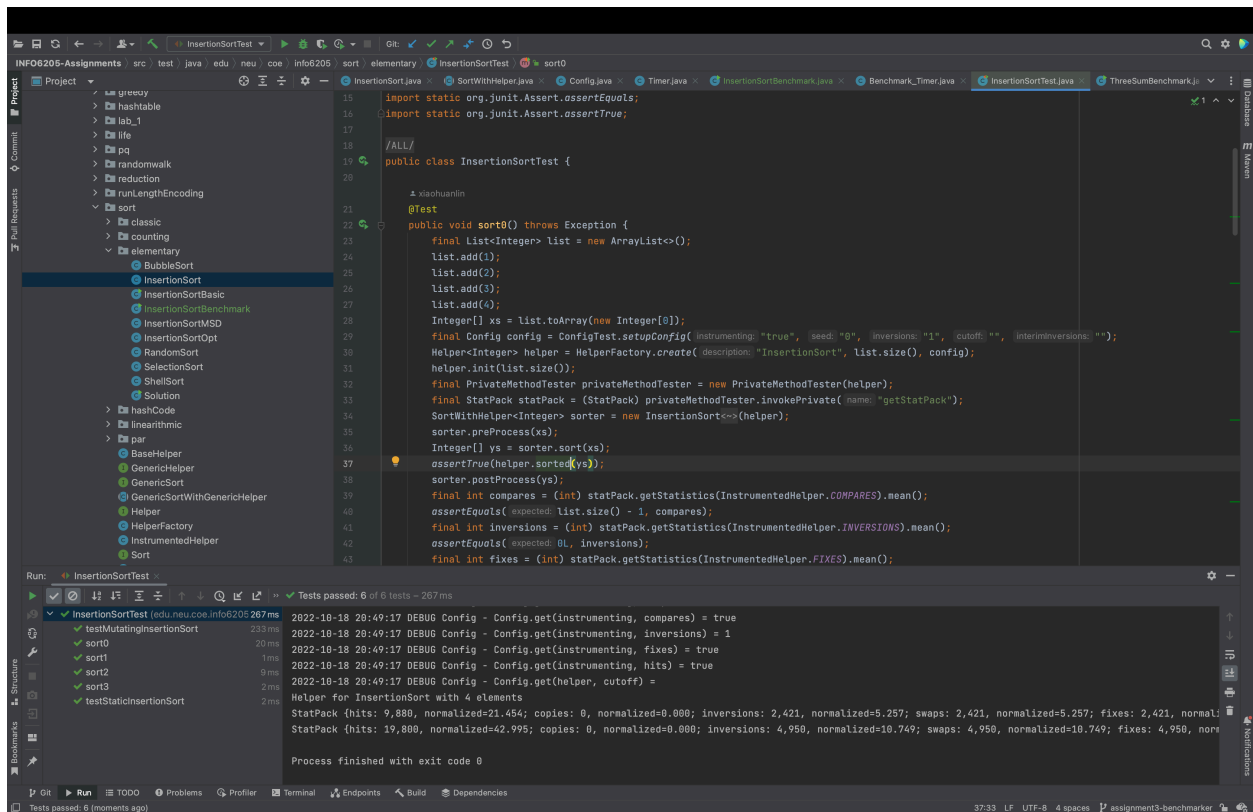


InsertionSort unit tests:



```
15 import static org.junit.Assert.assertEquals;
16 import static org.junit.Assert.assertTrue;
17
18 //ALL/
19 public class InsertionSortTest {
20
21     //xiaohuanlin
22     @Test
23     public void sort0() throws Exception {
24         final List<Integer> list = new ArrayList<>();
25         list.add(1);
26         list.add(2);
27         list.add(3);
28         list.add(4);
29         Integer[] xs = list.toArray(new Integer[0]);
30         final Config config = ConfigTest.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "", interinversions: "");
31         Helper<Integer> helper = HelperFactory.create( description: "InsertionSort", list.size(), config);
32         helper.init(list.size());
33         final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
34         final StatPack statPack = (StatPack) privateMethodTester.invokePrivate( name: "getStatPack");
35         SortWithHelper<Integer> sorter = new InsertionSort<>(helper);
36         sorter.preProcess(xs);
37         Integer[] ys = sorter.sort(xs);
38         assertTrue(helper.sorted(ys));
39         sorter.postProcess(ys);
40         final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
41         assertEquals( expected: list.size() - 1, compares);
42         final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
43         assertEquals( expected: 0L, inversions);
44         final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();
```

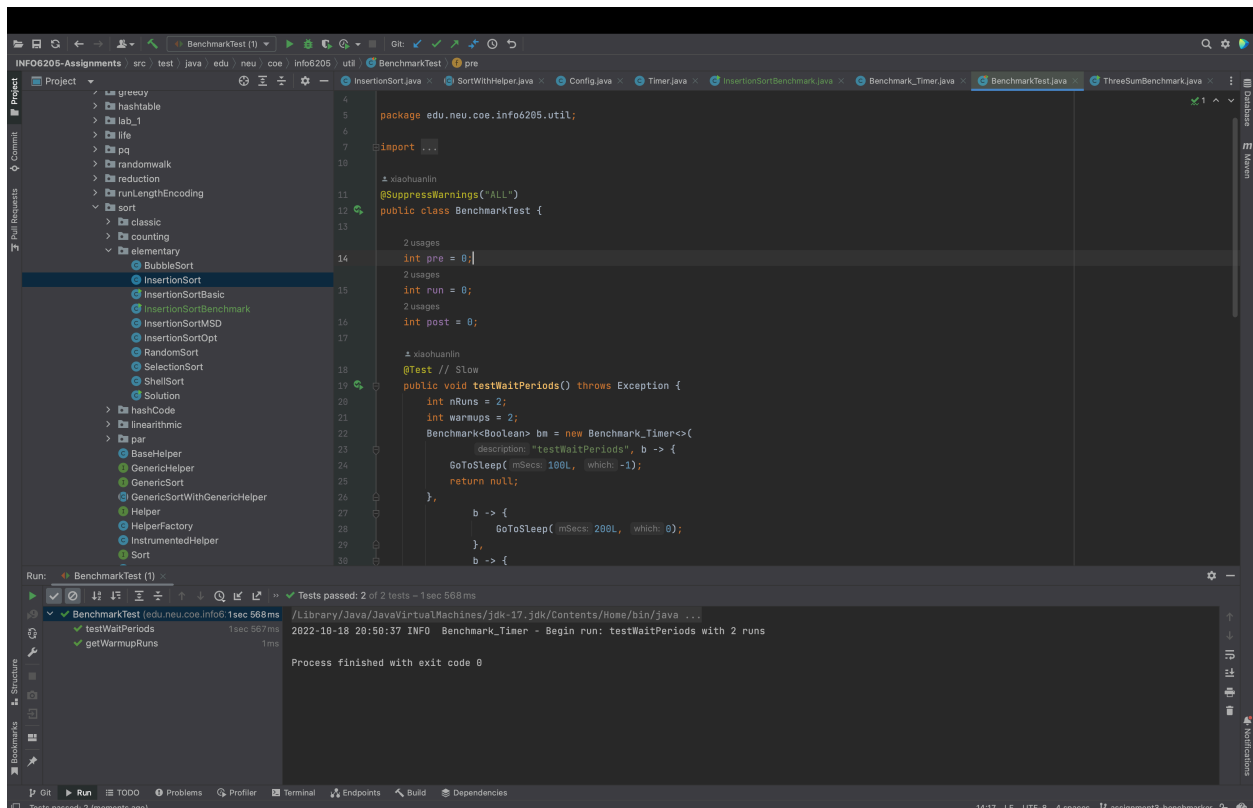
Run: InsertionSortTest

Tests passed: 6 of 6 tests - 267ms

- InsertionSortTest (edu.neu.coe.info6205) 267ms
- testMutatingInsertionSort 233ms
- sort0 20ms
- sort1 1ms
- sort2 9ms
- sort3 2ms
- testStaticInsertionSort 2ms

Process finished with exit code 0

Benchmark unit tests:



```
4 package edu.neu.coe.info6205.util;
5
6 //ALL/
7 import ...
8
9 //xiaohuanlin
10 @SuppressWarnings("ALL")
11 public class BenchmarkTest {
12
13     //2 usages
14     int pre = 0;
15     int run = 0;
16     int post = 0;
17
18     //xiaohuanlin
19     @Test // Slow
20     public void testWaitPeriods() throws Exception {
21         int nRuns = 2;
22         int warmups = 2;
23         Benchmark<Boolean> bm = new Benchmark_Timer<>(
24             description: "testWaitPeriods", b -> {
25                 GoToSleep( mSec: 100L, which: -1);
26                 return null;
27             },
28             b -> {
29                 GoToSleep( mSec: 200L, which: 0);
30             },
31             b -> {}
32         );
```

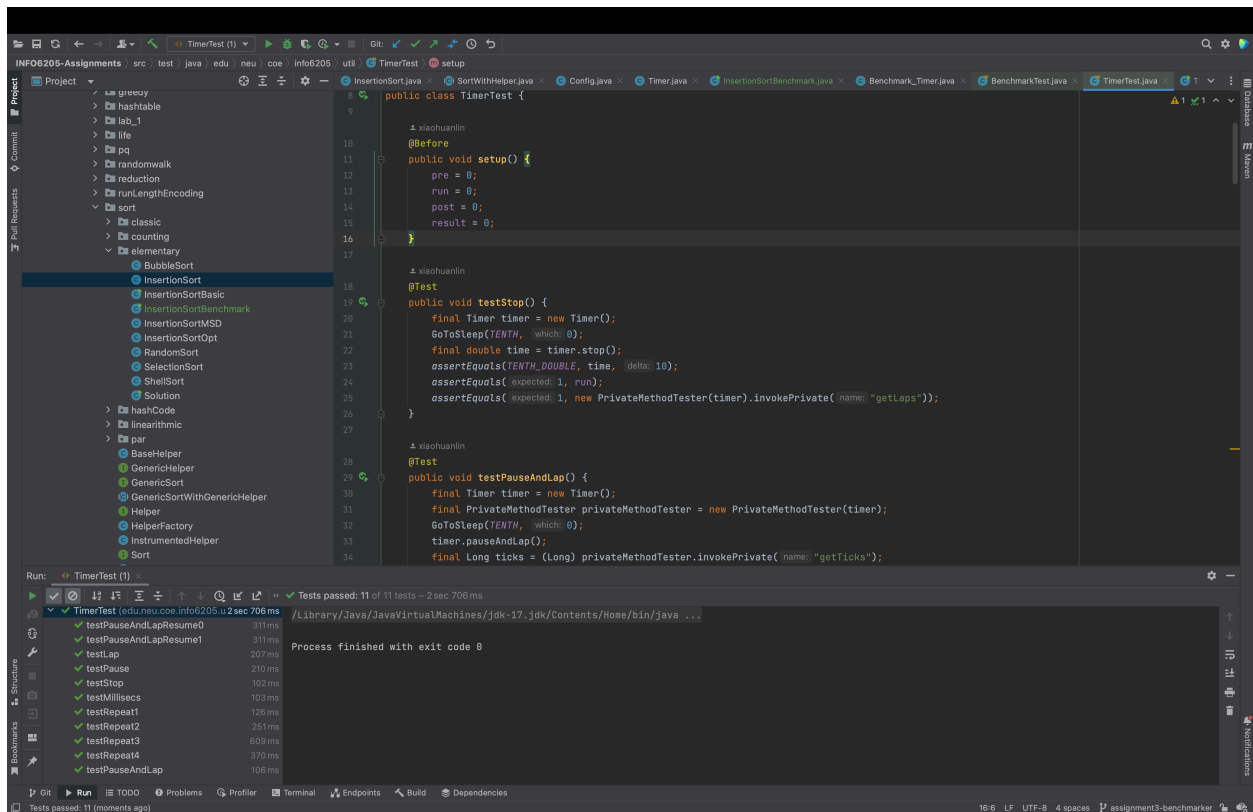
Run: BenchmarkTest (1)

Tests passed: 2 of 2 tests - 1sec 568ms

- BenchmarkTest (edu.neu.coe.info6) 1sec 568ms
- testWaitPeriods 1sec 567ms
- getWarmupRuns 1ms

Process finished with exit code 0

Timer unit tests:



Time Observation:

