Testing report of Project 4

Testing the class of Double linked list and its Iterator.

In the java file DoubleLinkedAndIteratorListTester (Yes, I know it is extremely long, but what can I do?) It tests the method of the DL list and its iterator together. Since the tester was provided by Prof during recitation, I only add the additional methods that I added, and they are: hasNext, hasPrevious, next, previous and add()

I test hasNext, hasPrevious, next, and previous together in one test program, and each is tested with first, middle, last rule, and the form of testing can be seen in the comment.

Then I test the add method with similar form of professor’s testing of remove method. It also follows the first, middle, last rule.

Test job class

First, test the constructor that the input changes the field, then test that the field after setter method is equal to the result of the getter method, and they are in the method called testGetSetConstructor.

Then the testComparable class test three comparable methods, and they are tested as input equal to the “this.”, smaller and greater to cover all possible situation of comparing.

Test compoundJob class

In this class, since that there are only two methods and one constructor that I created, they are in the same test class and it is CompoundJobTester class. The only method in it tests that the profit is properly calculated ( which is the total possible earn profit if all sub job is finished). Then test that the class’s constructor categories and stores all the input variable. Then, test the getter methods works properly.

Then I test the sortAllJob method, since it will only be called when I use schedule as Early/Late as possible, then I verify that each subjob’s start time is successfully set up. Since it tests that compound job can be added by start early/ late method, no more testing of add compound job will be provided.

Test ScheduleSlot class

In the ScheduleSlotTester class, it tests its getter setter and constructor.

Test ScheduleAsLateAsPossible, and ScheduleAsEarlyAsPossible class.

It is testing that for situation like no element/ only front/ only back in the schedule variable.

Then test that it can successfully detect some job can be added to the class and some cannot.

The test of adding CompoundJob is included at the sortAllJob tester.

Just look at JobScheduleTimeTester class, the explanation notes are detailed enough for any human being, and all possible situation of If are tested, and they all work properly, and if you are confused about any special code, please look at the outprint in the interaction pane, it includes detailed situation of each test trial.

JobScheduler

Since it only includes schedule one method, it is tested with schedule JobScheduleTimeTester class.

Since it is only about schedule a job according the input parameter, I will just test that it output whatever we want, instead of anything else.

Testing the main method:

I just input a job file and expect that one schedule early schedule and one schedule late schedule is printed, and the one that earns more profit is at first, then I walk trough each job to see that it is schedule in the early/late manner, and each one’s scheduled start and finish time are in between the earliest start time and the deadline, then I calculate the profit to see it is correct.