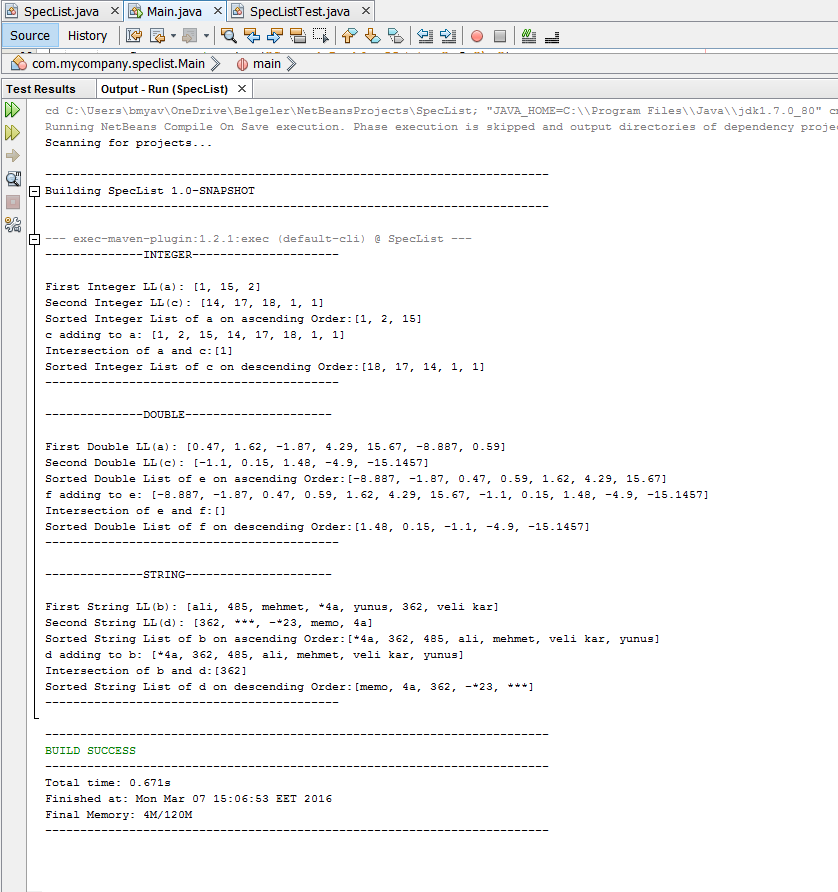
*REPORT HW3-SpecList*

* Homework Description: Implementation of specific list class called SpecList.
  + It has three method that are addAllAtHead, getIntersect, SortList
  + Calculation running time of this methods
* Class definition: public class SpecList<E extends Comparable <E> > extends LinkedList<E> so that I made E generic type to comparable to use compareTo method using with Comparable interface.
* I used the cocktail shaker sort algorithm pseudocode link that given in homework file for sortList method.
* In the addAllAtHead method implemented using with addAll(0,c) that Linked List method
* In the getIntersectList method I took an element from this then I compared with collections elements one by one. At the same time if there is same elements that added to intersect list then I don’t put same element to intersect list because it has to be unique list.
* Class Diagram:

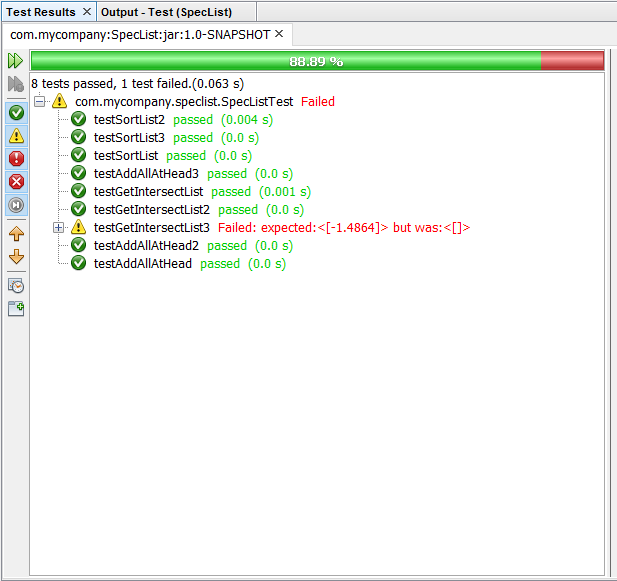
Each method was tested three different types such as String, Integers and Doubles. Result are shown below;



Junit Tests

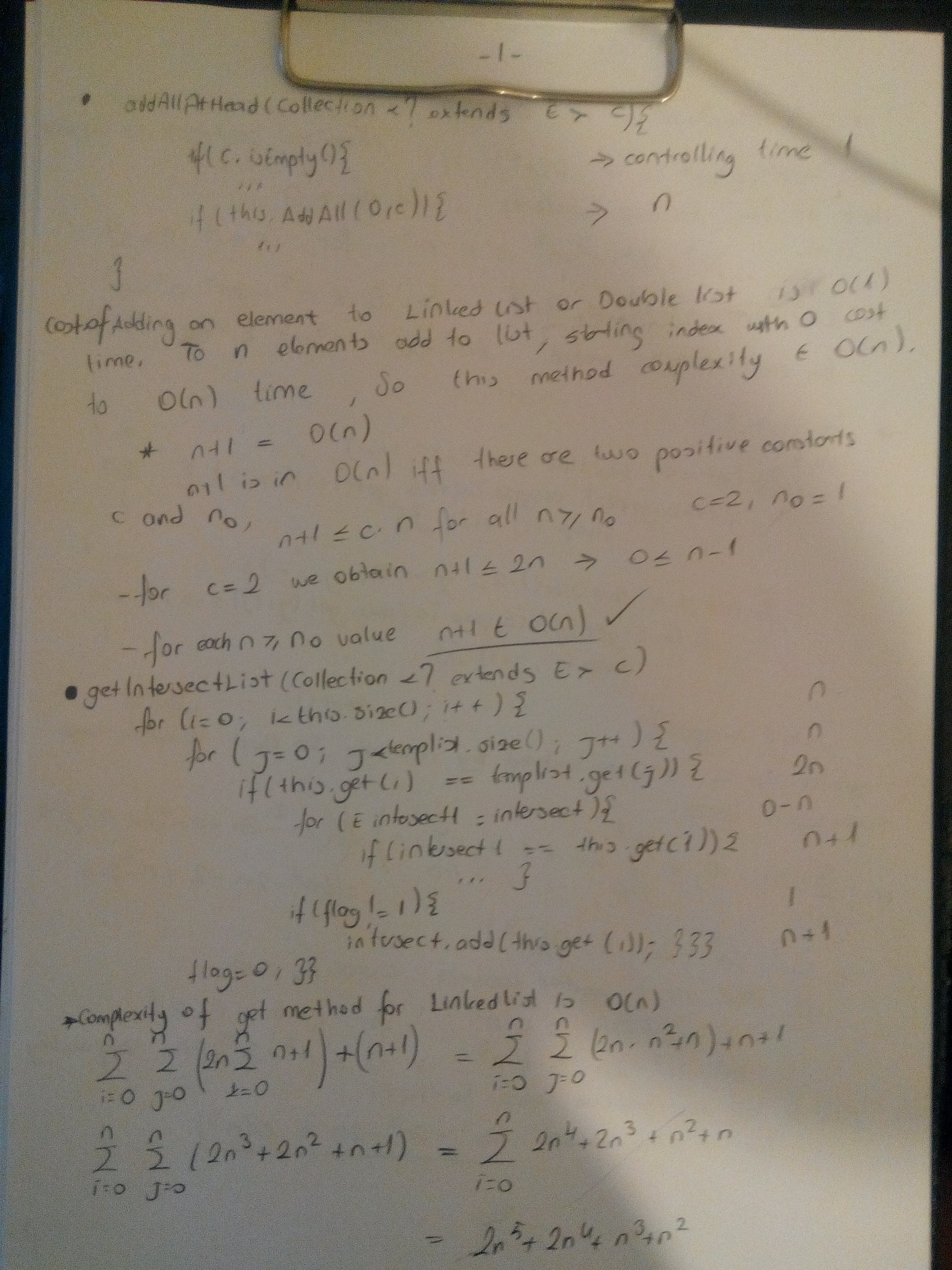
For each method there are 3 Junit Test. Each test types are String, Double and Integer.

* getIntersect: Three SpecList object are created then insert some elements to lists that types String, Integer, Double in the tests of getIntersectList. First two objects get intersect. İntersect elements are added to third object by hand. Finally, result and intersectList compared with assertEquals
* addAllAtHead: this method return boolean result if all the elements are added to first SpecList Object, method return true. This statement control return value and true using with assertEquals for each primitive type that assumed.
* SortList:



Run Time Analysis And Provement

* addAllAtHead and getIntersectList methods



* SortedList method

