

Test Plan Report

Code: 2536

Test Plan: 4930

Following Test Plan

General Notes

There is a lack of consistency between the file names in the test plan and the actual files given. For example, test 3 refers to using `test3.txt`, yet `test03.txt` was provided. While this is not a major issue, in a larger project it could lead to a lot of confusion. Computer Science is a field of precision.

What is the `sortTable` class? It is mentioned several times in your test plan but was not provided. Did you mean `TableSorter` class? Similarly, in section 2.2 you describe the `sortTable()` method, however, according to the interface given this method should be called `sortable()`.

Test No. 1

The code provided could not be run and required several corrections before it would run error free. Some of the errors included a lack of quotes around the filename (a string), missing parentheses, and a missing try-catch block. Below is a comparison of the original code vs code without errors.

```
// Code given
Public class test01 {
    Public static void main (String[] args){
Table t = getTable(test1.txt);
System.out.println(tableSorter.isSorted(t);//Result should be false
    }
}

// Code required to run test plan
public class test01 {
    public static void main (String[] args){
        try{
            Table t = Table.GetTable("test1.txt");
            System.out.println(TableSorter.isSorted(t));//Result should be false
        }catch(Exception e){
            System.out.println("Exception: " + e.toString());
        }
    }
}
```

Test No. 2

Similar to the first test case, significant modifications were required to the code given, including basic syntax errors and incorrect method calls. Additionally, the text file for test 2 does not contain a square table and thus, throws an error. As a result, this test could not be performed. Furthermore, the title of the test is `isSorted` test positive but yet the expected results of step 3 are the method should return false. These two statements are directly conflicting with each other.

Test No. 3

Again, several modifications were required to the provided code to make it run error free. Additionally, there was not supposed to be a `sortTable` method. According to the interface given, it is supposed to be called `sortable()`.

```
// Code given
Public class test03 {
    Public static void main (String[] args){
Table t = getTable(test03.txt);
tableSorter.sortTable(t);
System.out.println(tableSorter.isSorted(t);
    }
}

// Code required to run test plan
public class test03 {
    public static void main (String[] args){
        try{
            Table t = Table.GetTable("test03.txt");
            TableSorter.sortable(t);
            System.out.println(TableSorter.isSorted(t));
        }catch(Exception e){
            System.out.println("Exception: " + e.toString());
        }
    }
}
```

Test No. 4

Ran as expected after accommodating for the previously mentioned coding corrections.

Test No. 5

Ran as expected after accommodating for the previously mentioned coding corrections.

Test No. 6

While this test does execute as expected, this is not a test of the `TableSorter` class. It is a test of the `Table` class and should not be a part of this test plan.

Test No. 7

While this test does execute as expected, this is not a test of the `TableSorter` class. It is a test of the `Table` class and should not be a part of this test plan.

Test No. 8

While this test does execute as expected, this is not a test of the `TableSorter` class. It is a test of the `Table` class and should not be a part of this test plan.

Overall Test Plan Effectiveness

Overall, I do not think it provides effective coverage of the `TableSorter` class. The `isSorted()` method is only tested one time (with a negative result). Based on this test alone, you cannot determine if the `isSorted()` method returns the proper result or simply returns `False` under all conditions. Similarly, the `sortable()` method is only tested three times. This is not enough to cover all edge cases, partitions, or other unique situations that may arise. Of the tests that were carried out, all of them required significant modification before they could be run. In a true black box scenario, all of these tests would be impossible to run because of the incorrect method names (ex. `sortTable()` vs `sortable()` and `getTable()` vs `GetTable()`). Lastly, the last three tests do not fall within the scope of this test plan. Although they performed as expected, they are testing the `Table` class, not the `TableSorter` class, and thus, should be included in the `Table` class test plan. Test plans should be focused in their scope and not simply test every part of a program.