<<author name redacted>>

TableSorter Test plan

Version 1.1

02/11/2020

© 2020 <Enter team name here>

|  |  |
| --- | --- |
| 1. | Test Items and Features |

This document will focus on unit testing TableSorter class with a black box approach. Testing Approach

Table 1: Test Plan

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | TEST SUITE Unit SortTable | | | | | | |
| Description of Test Suite | | | Unit Testing Suite to be implemented in Junit for testing TableSorter class. | | | | | | |
| Test Case Identifier | | | Objective | | | Criticality | | | |
| U01 | | | Unit test SortTable method on TableSorter class by passing a null value. | | | High | | | |
| U02 | | | Unit test SortTable method on TableSorter class by passing a table of size 1 (1x1). | | | Medium | | | |
| U03 | | | Unit test SortTable method on TableSorter class by passing a table of size 4 (2x2). | | | High | | | |
| U04 | | | Unit test isSorted method on TableSorter class by passing a null value. | | | High | | | |
| U05 | | | Unit test isSorted method on TableSorter class by passing a table of size 1 (1x1). | | | Medium | | | |
| U06 | | | Unit test isSorted method on TableSorter class by passing a sorted table of size 4 (2x2). | | | High | | | |
| U07 | | | Unit test isSorted method on TableSorter class by passing an unsorted row table of size 4 (2x2). | | | High | | | |
| U08 | | | Unit test isSorted method on TableSorter class by passing an unsorted column table of size 4 (2x2). | | | High | | | |
| Test Plan | | | | <Enter team name here> | 02/11/2020 | | Page 1 |
| 2. | Unit Test Plan Details | | | | | | |

# 2.1. Test U01

Objective: Unit test SortTable method on TableSorter class by passing a null value. Notes: A test class needs to be created on Junit to run this test.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test No.: U01 | | | | | Current Status: Pending | | | | | |
| Test title: Pass empty array | | | | | | | | | | |
| Testing approach: In Junit, create a test case class that creates a null object and passes this object as a parameter to the TableSorter.SortTable(Table t) method. | | | | | | | | | | |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | | PURPOSE    Create an object reference without initializing it so its value is null. | | | EXEPCTED RESULTS    Program continues to next statement. | | COMMENTS | | |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | | PURPOSE    Initialize  TableSorter object, so we can test the SortTable() method. | | | EXEPCTED RESULTS    Program continues to next statement. | | COMMENTS | | |
| STEP    3 | OPERATOR ACTION    Call SortTable method by passing object t as a parameter. | | PURPOSE    Ensure TableSorter class will show an error when receiving a null parameter. | | | EXEPCTED RESULTS    SortTable() Method generates a  NullPointerException.  Error. | | COMMENTS | | |
| STEP    4 | OPERATOR ACTION    Handle  NullPointerException on a TryCatch statement. | | PURPOSE    Prevent the program from crashing and continue processing. | | | EXEPCTED RESULTS    Error is logged, but program continues processing. | | COMMENTS | | |
| Concluding Remarks: | | | | | | | | | | |
| Test Plan | | <Enter team name here> | | | | | 02/11/2020 | | Page 2 |
| Testing Team:  Testing and integration class. | | | | Date Completed: | | | | | | |

# 2.2. Test U02

Objective: Unit test SortTable method on TableSorter class by passing a table of size 1. Notes: A test class needs to be created on Junit to run this test.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test No.: U02 | | | | Current Status: Pending | | | | | |
| Test title: Pass a Table with only position [0][0] | | | | | | | | | |
| Testing approach: In Junit, create a test case class that creates a single position Table object t and passes this object as a parameter to the TableSorter.SortTable(Table t) method. | | | | | | | | | |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | | PURPOSE    Create an object reference and initialize it with the default constructor. | | EXEPCTED RESULTS    Program continues to next statement. | | COMMENTS | | |
| STEP    1 | OPERATOR ACTION    Call t.GetTable(String s) method to load a file with only one valid int value inside it. | | PURPOSE    Load a Table with a size of 1. | | EXEPCTED RESULTS    Program continues to next statement. | | COMMENTS | | |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | | PURPOSE    Initialize  TableSorter object, so we can test the SortTable() method. | | EXEPCTED RESULTS    Program continues to next statement. | | COMMENTS | | |
| STEP    3 | OPERATOR ACTION    Call SortTable method by passing object t as a parameter. | | PURPOSE    Ensure TableSorter class will not crash when receiving a Table of size 1. | | EXEPCTED RESULTS    Method completes successfully. | | COMMENTS | | |
| Test Plan | | <Enter team name here> | | | | 02/11/2020 | | Page 3 |

# 2.3. Test U03

Objective: Unit test SortTable method on TableSorter class by passing a table of size 4. Notes: A test class needs to be created on Junit to run this test.

|  |  |
| --- | --- |
| Concluding Remarks: |  |
| Testing Team:  Testing and integration class. | Date Completed: |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: U03 | | | Current Status: Pending | | |
| Test title: Pass a Table with a size of 4 | | | | | |
| Testing approach: In Junit, create a test case class that creates 2x2 Table object t and passes this object as a parameter to the TableSorter.SortTable(Table t) method. | | | | | |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | PURPOSE    Create an object reference and initialize it with the default constructor. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    1 | OPERATOR ACTION    Call t.GetTable(String s) method to load a file with a 2x2 matrix of valid int values. | PURPOSE    Load an object of Table with a size of 2x2. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | PURPOSE    Initialize  TableSorter object, so we can test the SortTable() method. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |

4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STEP    3 | OPERATOR ACTION    Call SortTable method by passing object t as a parameter. | PURPOSE    Ensure TableSorter class will not crash when receiving a  Table of size 2x2. | | EXEPCTED RESULTS    Method completes successfully. | COMMENTS |
| STEP    4 | OPERATOR ACTION    Call IsSorted method by passing object t as a parameter. | PURPOSE    Ensure Table object t is now sorted left to right on rows and top to  bottom on columns. | | EXEPCTED RESULTS    Method returns “True” value. | COMMENTS |
| Concluding Remarks: | |  | | |  |
| Testing Team:  Testing and integration class. | |  | Date Completed: | |  |

# 2.4. Test U04

Objective: Unit test isSorted method on TableSorter class by passing a null value. Notes: A test class needs to be created on Junit to run this test.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test No.: U04 | | | | | Current Status: Pending | | | | | |
| Test title: Pass empty array | | | | | | | | | | |
| Testing approach: In Junit, create a test case class that creates a null object and passes this object as a parameter to the TableSorter.isSorted(Table t) method. | | | | | | | | | | |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | | PURPOSE    Create an object reference without initializing it so its value is null. | | | EXEPCTED RESULTS    Program continues to next statement. | | COMMENTS | | |
| Test Plan | | <Enter team name here> | | | | | 02/11/2020 | | Page 2 |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | | PURPOSE    Initialize  TableSorter object, so we can test the isSorted() method. | | | EXEPCTED RESULTS    Program continues to next statement. | | COMMENTS | | |
| STEP    3 | OPERATOR ACTION    Call isSorted method by passing object t as a parameter. | | PURPOSE    Ensure TableSorter class will show an error when receiving a null parameter. | | | EXEPCTED RESULTS    isSorted() Method generates a  NullPointerException.  Error. | | COMMENTS | | |
| STEP    4 | OPERATOR ACTION    Handle  NullPointerException on a TryCatch statement. | | PURPOSE    Prevent the program from crashing and continue processing. | | | EXEPCTED RESULTS    Error is logged, but program continues processing. | | COMMENTS | | |
| Concluding Remarks: | | |  | | | | |  | | |
| Testing Team:  Testing and integration class. | | |  | Date Completed: | | | |  | | |

# 2.5. Test U05

Objective: Unit test isSorted method on TableSorter class by passing a table of size 1. Notes: A test class needs to be created on Junit to run this test.

|  |  |
| --- | --- |
| Test No.: U05 | Current Status: Pending |
| Test title: Pass a Table with only position [0][0] | |
| Testing approach: In Junit, create a test case class that creates a single position Table object t and passes this object as a parameter to the TableSorter.isSorted(Table t) method. | |

3

Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | PURPOSE    Create an object reference and initialize it with the default constructor. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    1 | OPERATOR ACTION    Call t.GetTable(String s) method to load a file with only one valid int value inside it. | PURPOSE    Load a Table with a size of 1. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | PURPOSE    Initialize  TableSorter object, so we can test the isSorted() method. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    3 | OPERATOR ACTION    Call isSorted method by passing object t as a parameter. | PURPOSE    Ensure TableSorter class will not crash when receiving a Table of size 1. | | EXEPCTED RESULTS    Method completes successfully and returns a value of “True”. | COMMENTS |
| Concluding Remarks: | |  | | |  |
| Testing Team:  Testing and integration class. | |  | Date Completed: | |  |

# 2.6. Test U06

Objective: Unit test isSorted method on TableSorter class by passing a sorted table of size 4. Notes: A test class needs to be created on Junit to run this test.

|  |  |
| --- | --- |
| Test No.: U06 | Current Status: Pending |
| Test title: Pass a sorted Table with a size of 4 |  |

4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Testing approach: In Junit, create a test case class that creates 2x2 sorted Table object t and passes this object as a parameter to the TableSorter.isSorted(Table t) method. Matrix values:  [1 2]  [3 4] | | | | | |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | PURPOSE    Create an object reference and initialize it with the default constructor. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    1 | OPERATOR ACTION    Call t.GetTable(String s) method to load a file with a sorted 2x2 matrix of valid int values. Matrix values: [1 2]  [3 4] | PURPOSE    Load an object of Table with a size of 2x2. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | PURPOSE    Initialize  TableSorter object, so we can test the isSorted() method. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    3 | OPERATOR ACTION    Call isSorted method by passing object t as a parameter. | PURPOSE    Ensure TableSorter class will not crash when receiving a  Table of size 2x2. | | EXEPCTED RESULTS    Method completes successfully and returns a “True” value. | COMMENTS |
| Concluding Remarks: | | | | | |
| Testing Team:  Testing and integration class. | | | Date Completed: | | |

# 2.7. Test U07

Objective: Unit test isSorted method on TableSorter class by passing an unsorted table of size 4. Notes: A test class needs to be created on Junit to run this test.

4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: U07 | | | Current Status: Pending | | |
| Test title: Pass an unsorted Table with a size of 4 | | | | | |
| Testing approach: In Junit, create a test case class that creates 2x2 unsorted Table object t and passes this object as a parameter to the TableSorter.isSorted(Table t) method. Matrix values:  [2 3]  [1 4] | | | | | |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | PURPOSE    Create an object reference and initialize it with the default constructor. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    1 | OPERATOR ACTION    Call t.GetTable(String s) method to load a file with a sorted 2x2 matrix of valid int values. Matrix values: [2 3]  [1 4] | PURPOSE    Load an object of Table with a size of 2x2. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | PURPOSE    Initialize  TableSorter object, so we can test the isSorted() method. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    3 | OPERATOR ACTION    Call isSorted method by passing object t as a parameter. | PURPOSE    Ensure TableSorter class will not crash when receiving a  Table of size 2x2. | | EXEPCTED RESULTS    Method completes successfully and returns a “False” value. | COMMENTS |
| Concluding Remarks: | | | | | |
| Testing Team:  Testing and integration class. | | | Date Completed: | | |

# 2.8. Test U08

Objective: Unit test isSorted method on TableSorter class by passing an unsorted column table of size 4.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Plan |  | 02/11/2020 | Page 4 |

Notes: A test class needs to be created on Junit to run this test.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Plan |  | 02/11/2020 | Page 4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: U08 | | | Current Status: Pending | | |
| Test title: Pass an unsorted Table with a size of 4 | | | | | |
| Testing approach: In Junit, create a test case class that creates 2x2 unsorted Table object t and passes this object as a parameter to the TableSorter.isSorted(Table t) method. Matrix values:  [-3 4]  [-1 2] | | | | | |
| STEP    1 | OPERATOR ACTION    Write code to declare an object t of type Table. | PURPOSE    Create an object reference and initialize it with the default constructor. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    1 | OPERATOR ACTION    Call t.GetTable(String s) method to load a file with a sorted 2x2 matrix of valid int values. Matrix values: [-3 4]  [-1 2] | PURPOSE    Load an object of Table with a size of 2x2. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    2 | OPERATOR ACTION    Create a new object of type TableSorter and initialize the object with the default constructor. | PURPOSE    Initialize  TableSorter object, so we can test the isSorted() method. | | EXEPCTED RESULTS    Program continues to next statement. | COMMENTS |
| STEP    3 | OPERATOR ACTION    Call isSorted method by passing object t as a parameter. | PURPOSE    Ensure TableSorter class will not crash when receiving a  Table of size 2x2. | | EXEPCTED RESULTS    Method completes successfully and returns a “False” value. | COMMENTS |
| Concluding Remarks: | | | | | |
| Testing Team:  Testing and integration class. | | | Date Completed: | | |