Tests.pdf

Test-Cases

**Only two test cases are necessary, as the program doesn't end or close without the user's consent. One test case includes reading another document with 100 orders following the initial to ensure that the AVLTree can't surpass 100 orders (as intended). The other test case ensures that the rest of the exceptions and limitations are properly checked and work as intended.

First Test Case – Surpassing the AVLTree's order limit

- Create an additional .txt file containing 100 orders with the format: "orderNumber,orderName". This test case requires the code to be changed (for the sake of it being easier), adding an additional addOrdersFromFile() method after reading the first. Another way of testing the order limit would be to add 80 orders manually through the commands given upon running the program, seeing that the last few can't be added because of the order limit. I tested it, and the order limit works, but I understand if this test case can't be properly replicated.

Second Test Case – Testing (almost) every other limitation and exception

**Some limitations and exceptions are annoying to test (i.e. finding the oldest book order of an empty AVL tree will inform the user that there are no orders), but they've already been tested for. With the complete program though, it's painstaking to test them without editing the code (you would need to remove every single order, for example). If you'd like to test these, a full list of limitations and exceptions can be found below.

- Run the program
- Enter "menu" to ensure that the menu is properly reprinted.
- Enter "7" to ensure that non-existent commands are properly handled by the program.
- Enter "1" to add a book order.
    - Type "cancel" to ensure that cancelling an operation works.
- Enter "1" again to add a book order.
    - Type a non-integer value to ensure that the program handles it properly.
    - Type a negative integer value to ensure that the program prevents it.
    - Type an integer to progress to the next step of adding a book order.

    - Press enter without typing anything for the name to ensure that the program prevents it.
    - Type a name for the order, the program will add it to the AVLTree.
- Enter “1” again.
    - Attempt to add an order with an order number that already exists to ensure that the program handles it properly.
- Enter “1” again.
    - Attempt to add an order with an order name of over 150 characters to ensure that the program properly prevents it.
- Enter “2” to remove a book order.
    - Type an integer for an order ID that doesn’t exist to make sure the program handles it properly.
- Enter “2” again to remove a book order.
    - Type an order ID that exists in the AVLTree, the program will remove it.
    - Non-integer inputs will re-prompt the user to enter an integer. Feel free to test this again for each operation that requires it.
- Enter “3” to print the full list of orders.
- Enter “4” to search for an order’s book name using its order number.
    - Type an order number that doesn’t exist to make sure the program handles it properly.
- Enter “4” again to search.
    - Type an order number that exists, the program will give the order’s book name.
- Enter “5” to get the oldest book order.
- Enter “6” to get the latest book order.
- Type “exit” to exit the program.

Limitations & Exceptions

- The AVLTree can store a maximum of 100 orders at once.
- Order names can’t surpass 150 characters.
- Order numbers can’t be negative
- The user can’t add a book order with an already-existing order number.
- When the AVLTree is empty, lets the user know when they are trying to print the orders, find the latest order, or find the oldest order.
- Whenever the file being read doesn’t exist, catches the exception and lets the user know.