Testing Documentation

Assignment 2 - Group 10

Q1. The program will take an input in the form: [1 [2 3] [4 5] [6 4]] and store it using a ListComponent. The program should create a ListComposite and add a child of type Item for 1. Then it will add another child of type ListComposite for [2 3], in which it will add two Items 2 and 3, and so on.

Case 1. To check if program works or not	
Input	[1 [2 3] [4 5] [6 4]]
Output	<terminated> Question1 [Java Application] C:\Program Files\Java\jdk-11.0.5\bin\javaw.exe (Mar. 9, 2021, 1:28: Welcome to the Composite pattern and Builder pattern implementation Only input in the form of [int int] is accepted The value of our ListComponent object obtained from ListBuilder.getlist(): [1 [2 3] [4 5] [6 4]] Testing get child for index 1 on the ListComponent [2 3] Testing remove child for index 0 on the ListComponent New list after removal of index 0: [[2 3] [4 5] [6 4]]

Case 2. Checking with different inputs	
Input	[1 2 [3 4] 7]
Output	Welcome to the Composite pattern and Builder pattern implementation Only input in the form of [int int] is accepted The value of our ListComponent object obtained from ListBuilder.getlist(): [1 2 [3 4] 7] Testing get child for index 1 on the ListComponent 2 Testing remove child for index 0 on the ListComponent New list after removal of index 0: [2 [3 4] 7]

Case 3. Checking with different inputs	
Input	[[10 80] [55 47]]
Output	Welcome to the Composite pattern and Builder pattern implementation Only input in the form of [int int] is accepted The value of our ListComponent object obtained from ListBuilder.getlist(): [[10 80] [55 47]] Testing get child for index 1 on the ListComponent [55 47] Testing remove child for index 0 on the ListComponent New list after removal of index 0: [55 47]

Case 4. Error checking (Invalid Input)	
Input	[[10 80] [55 47] [ ]
Output	Welcome to the Composite pattern and Builder pattern implementation Only input in the form of [int int] is accepted The value of our ListComponent object obtained from ListBuilder.getlist(): Error! Invalid Input. Execution Stopped

Case 5. Error Checking (Invalid Input)	
Input	[[10 80] [55 47] aa]
Output	Welcome to the Composite pattern and Builder pattern implementation Only input in the form of [int int] is accepted The value of our ListComponent object obtained from ListBuilder.getlist(): Error! Invalid Input. Execution Stopped

Q2. Write proxies that can be set up at construction time to protect its real subject by allowing to read only, write only, or read/write (3 options) the content of the real subject. Write a main program to test your proxy objects with different protection options.

<table>
<tr><td colspan="2">Case : Check program runs</td></tr>
<tr><td>Input</td><td>None required. A string is created. First we try write only. When we try to get content, it will not work due to protection level of write only. Then read only is tried, where we are unable to write/edit because of protection level read only. Then we do read/write where everything will work : The “trying to write” is just followed by write function being called thus no output is print as such.</td></tr>
<tr><td>Output</td><td>String = “Hello check, write only”
<pre>
WRITE ONLY

Trying to read :
Error : Write only, cannot access read due to protection level

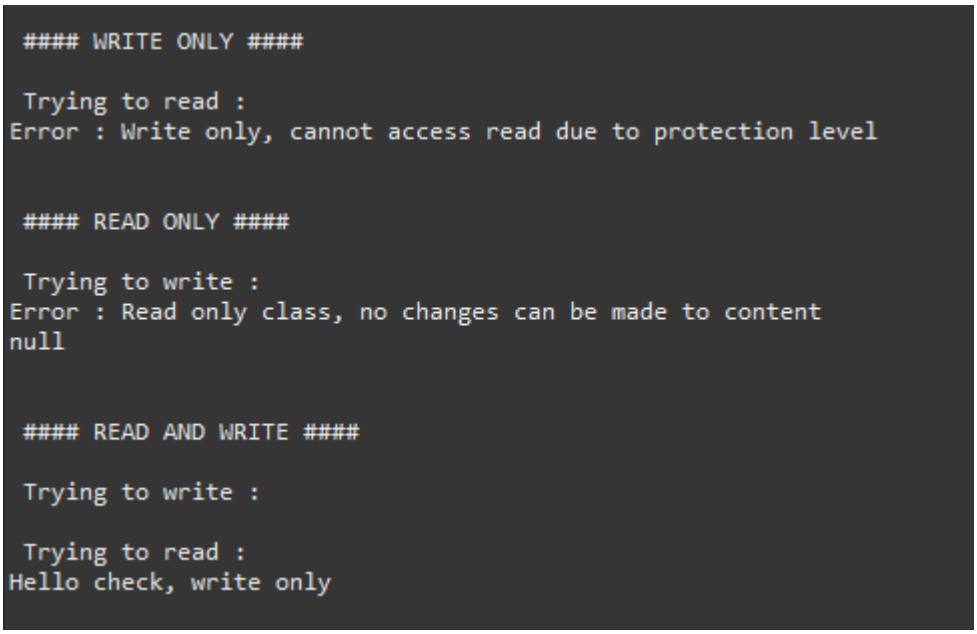
READ ONLY

Trying to write :
Error : Read only class, no changes can be made to content
Hello check, write only

READ AND WRITE

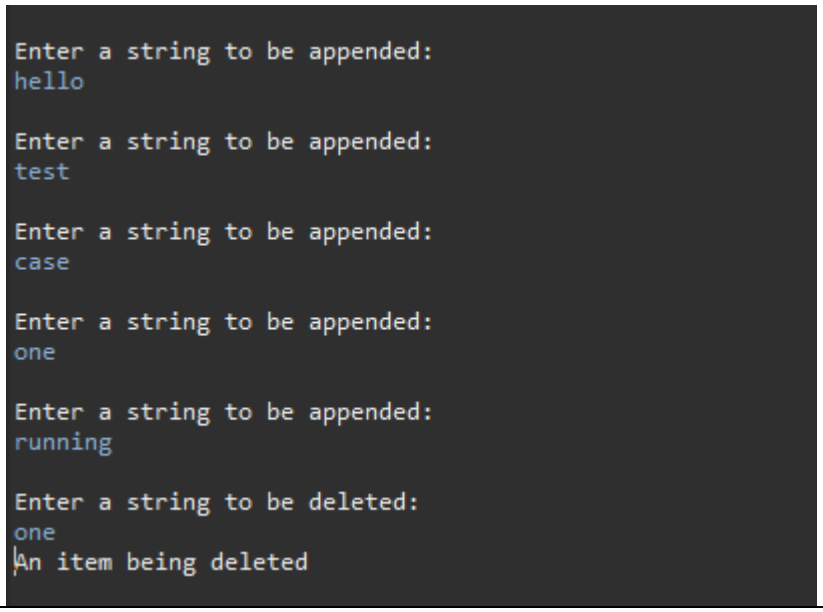
Trying to write :

Trying to read :
Hello check, write only
</pre></td></tr>
</table>

Case : Check program runs	
Input	This case is what we had previously in our code. There was no string parameter in the Text constructor or any constructor – but then since read only was read only, there was no content set to the read only object string content. So, it printed null. This screenshot depicts that.
Output	String = “Hello check, write only” 

Q3 . Use the observer pattern to define an observer

We ask the user for five string inputs – these are appended to the ArrayListSubject arraylist as objects. Then we ask the user to enter a string to be deleted from the array list. The first case is when the string entered to be deleted exists. The second one is when string entered to be deleted does not exist in the arraylist.

Case : An item that exists in the array list is deleted	
Input	Hello, test, case, one, running, (for deleted -) one
Output	

```
Enter a string to be appended:
hello

Enter a string to be appended:
test

Enter a string to be appended:
case

Enter a string to be appended:
one

Enter a string to be appended:
running

Enter a string to be deleted:
one
An item being deleted
```

Case : An item that does not exist in the array list is tried to be deleted	
Input	Hello, test, case, one, running, (for deleted -) professor
Output	

```
Enter a string to be appended:
hello

Enter a string to be appended:
test

Enter a string to be appended:
case

Enter a string to be appended:
one

Enter a string to be appended:
running

Enter a string to be deleted:
professor

Item was not found in array
```