Glarmester Test

- How are you going make your integration test (which approach, which method)?

We have used Manual Testing approach, since we presume the project is in its final version and no changes in the code are going to occur. Manual tests are taking up on human resources, but the test cases are run only once, hence the cost is not so big. The tests are conducted in human observation; therefore, we have come up and defined some cases on the go as well.

In manual UI testing, where we manually input data to see if we manage to get any errors. We will use numbers, negative numbers, symbols, letters (both big and small) and special characters. It is less exhaustive and time-consuming to use Black box testing in this particular case. Apart from the Black-box testing we should also make a performance test, which gives us an acceptable execution time.

1.Here we test manually the input for “height” and “width” with strings, symbols, negative numbers and 0/null.

```
compile:
run:
Enter height of window (in cm.): A
Enter height of window (in cm.): a
Enter height of window (in cm.): !
Enter height of window (in cm.): ?
Enter height of window (in cm.): #
Enter height of window (in cm.): -1
Enter height of window (in cm.): 0
Enter height of window (in cm.): 3
Enter width of window (in cm.): B
Enter width of window (in cm.): b
Enter width of window (in cm.): !
Enter width of window (in cm.): ?
Enter width of window (in cm.): #
Enter width of window (in cm.): -10
Enter width of window (in cm.): 0
```

2. When a case is accepted and runs through, the system gets to the next input question. We can see that we experience an issue with passing through “0” in the “width” parameter, which would not be a viable value.

```
compile:
run:
Enter height of window (in cm.): 0
Enter height of window (in cm.): 500
Enter width of window (in cm.): 0
Select frametype:
1) Simple
2) Ornate
3) Lavish


Select frametype:
1) Simple
2) Ornate
3) Lavish
A
Select frametype:
1) Simple
2) Ornate
3) Lavish
!
Select frametype:
1) Simple
2) Ornate
3) Lavish
0
java.lang.NullPointerException
```

Apart from that, all goes as supposed to in input question number 3.

```
Select frametype:
1) Simple
2) Ornate
3) Lavish
4
Select frametype:
1) Simple
2) Ornate
3) Lavish
5000
```

3. Another test case that we came up is if we could take any value, even if it is unreal per se. For example, having a window with height of 1 cm and 500000 cms in width.

```
compile:
run:
Enter height of window (in cm.): 1
Enter width of window (in cm.): 500000
Select frametype:
1) Simple
2) Ornate
3) Lavish
```