SWE 4301 - Object Oriented Concepts II

LAB 3: Refactoring and Testing

=====
Tasks
=====

I have added 2 utility classes (Inventory and Item) and 1 test class (IneventoryTest). The first thing you have to do is create a java project with JUnit 5. In the last class before EID, I demonstrated how to create a java project with JUnit 5 in Intellij. If you use any other IDE, you can find resources to set up your JUnit 5 environment online.
Then add the two utility classes to your project and add another test class. Inside the test class, add the test method foo (try to give it a better name) and do everything you need to do to pass this test. This ensures that your testing environment is okay.
After that, go through the inventory class and requirements specification. Try to understand what these two classes do. For this lab, you have to understand the updateQuality method from the Inventory class only.
Now, write very extensive sets of tests for checking the successful update of Item name, quality value etc.
The final task is to refactor the updateQuality method. Refactor this method iteratively.

Submission:
In google classroom you have to submit 3 things -

1. Compressed file of your project.
2. A PDF document - Report.
3. Video Demonstration, showing that your code passes all the tests.

In the PDF Document, provide justification for all the test methods you have created. Also, the justification for refactoring. After every iteration of refactoring, explain what are the things you have changed and the code snippet where you have changed just like I demonstrated in the last class.

Naming Convention of files: 'LAB3_FullStudentID.extension(.pdf, .zip, .mp4)'

Submission Deadline: **May 23, 2021**

***There will be no viva for this lab.
***There is a thing called 'Plagiarism Checker'. It will check the similarities between all your submissions. If I find similarities between any two scripts I will discard both. Before sharing your code or report with others, please think again.
***Please spend some time learning these things and then, complete the task without taking help from others. That is the reason behind giving 5 days for a single task.

==============================
Inventory Requirements Specification
==============================

We are a small inn with a prime location in a prominent city run by a friendly innkeeper named Allison. We also buy and sell only the finest goods. Unfortunately, our goods are constantly degrading in quality as they approach their sell by date. We have a system in place that updates our inventory for us. It was developed by a no-nonsense type named Leeroy, who has moved on to new adventures. Your task is to add the new feature to our system so that we can begin selling a new category of items. First an introduction to our system:

- All items have a SellIn value which denotes the number of days we have to sell the item
- All items have a Quality value which denotes how valuable the item is
- At the end of each day our system lowers both values for every item

Pretty simple, right? Well this is where it gets interesting:

- Once the sell by date has passed, Quality degrades twice as fast
- The Quality of an item is never negative
- "Aged Brie" actually increases in Quality the older it gets
- The Quality of an item is never more than 50
- "Sulfuras", being a legendary item, never has to be sold or decreases in Quality
- "Backstage passes", like aged brie, increases in Quality as its SellIn value approaches; Quality increases by 2 when there are 10 days or less and by 3 when there are 5 days or less but Quality drops to 0 after the concert

We have recently signed a supplier of conjured items. This requires an update to our system:

- "Conjured" items degrade in Quality twice as fast as normal items

Feel free to make any changes to the UpdateQuality method and add any new code as long as everything still works correctly. However, do not alter the Item class or Items property as those belong to the goblin in the corner who will insta-rage and one-shot you as he doesn't believe in shared code ownership (you can make the UpdateQuality method and Items property static if you like, we'll cover for you).

Just for clarification, an item can never have its Quality increase above 50, however "Sulfuras" is a legendary item and as such its Quality is 80 and it never alters.