PROG1442
D. Stovell

Unit Testing - Exercise 2

Lab Summary

Willie is a card game played where each player only gets 2 cards. Card Suit does not matter. The value of the players hand (his Willie) is the total of the face value of the two cards except… face cards are all worth 10 and Aces are always worth one. A player can pass on the second card and 'stand' on a one card Willie hand. This is achieved with the class file given by passing a Null for the second card to the constructor.

Part A

Begin by creating a class library project (with your name in the title) in a folder created in the root of your hard drive. Use the version of the code for the WillieHand class from Blackboard, posted with this assignment. Note that is has been created as a partial class. Although using an enumerator can help make sure that only valid card values are passed to the WillieHand constructor, we have opted to go a different route for a chance to practice exception handling in our Unit Tests.

Complete the following steps:

1. Add a Unit Test project to the solution with a TestClass for WillieHand.
2. Modify the constructor in WillieHand to throw an `ArgumentOutOfRangeException` if an invalid value is passed in for either the first or second card.
3. *Boundary Testing* is used to look for a change in behaviour when values are above or below a 'boundary' value. Create the following Use Cases to perform 'boundary testing' for Card Two.

Use Case	Card One	Card Two	Willie Value/Exception
A1	7	-1	ExpectedException
A2	7	Null	7
A3	7	13	17
A4	7	14	ExpectedException

4. Make up your own Use Cases for boundary testing Card One.

Part B

You just found out that if two players are tied in the value of their 'Willie' then they need to compare their values for 'Wonka.' The Wonka of a Willie hand is the actual face value of the highest card in the hand. You will extend the Willie class to add a new Wonka property and add unit tests as shown below.

Complete the following steps:

5. Add a second class file to the library project and modify the class in the new file so you can extend the functionality of the WillieHand class in the new class file. (i.e. `public partial class WillieHand)`
6. Add a new read only property in the new class file called 'Wonka' that returns the actual face value of the highest card in the hand.

Create tests for both the Willie and Wonka values for each case (eg. B1WillieTest, B1WonkaTest etc.)

Use Case	Card One	Card Two	Willie Value	Wonka Value
B1	7	Jack	17	11
B2	Queen	Ace	11	12
B3	King	Jack	20	13
B4	7	7	14	7

Part C

In the class library project, add a new class called WillieWinner in a new class file. The WillieWinner class must be able to compare two WillieHand objects and determine which is the winner. You can either use a method or properties in the design of the class. The 'Winner' value must return the following:

- 1 if the first hand beats the second hand.
- -1 if the second hand beats the first one.
- 0 if they are tied

When comparing the hands, you compare the Willie values first and the highest Willie value wins. If the Willie values of the two hands are the same, then you compare the Wonka values of the hands and the highest Wonka value wins.

Add a new TestClass to the test project to hold Unit Tests for the WillieWinner class. In the new TestClass, add Unit Tests as per the following Use Cases:

Use Case	Hand One	Hand Two	Winner Value
C1	(Ace, King)	(Jack, 7)	-1
C2	(Ace, King)	(Six,Four)	1
C3	(Ace, King)	(Queen, Ace)	1
C4	(Six,Five)	(Ace,King)	-1
C4	(Jack, 10)	(10, Jack)	0

Feel free to make up additional Use Cases as you see fit.