Card Game Coding Exercise

Outline

For this task, you are required to write an application (with C# back-end and user interface of your choice, e.g. a command line, Windows Forms or web interface), which takes a list of playing cards and returns a score for the cards given.

The code that you write must be written in the Test Driven Development (TDD) style, with 100% unit test coverage.

This exercise is an 'open-book' exercise. You are encouraged to use online resources (such as Google or Stack Overflow) and ask the interviewers about any requirements you are not sure about; this will be seen as positive behaviour.

You **must not "**copy-and-paste" a solution from another source; the code you submit must be entirely your own work.

The Card Game

A card game is played and at the end of the game, a score is calculated based on the cards the winner has left at the end of the game. Each card (in a standard deck of cards) has a value, based on its value and suit, according to the rules below:

- Number cards are worth their face value; e.g. 4 equals 4 points.
- A Jack is worth 11 points, a Queen 12 points, a King 13 points and an Ace 14 points.
- The suit of the card determines what to multiply the card's value by:
    - Club: Multiply by 1
    - Diamond: Multiply by 2
    - Heart: Multiply by 3
    - Spade: Multiply by 4

For example, the Ace of Spades point value would be 56 (14 x 4).

The point values for each card are added up to give a final score. If a Joker appears anywhere in the list of cards, the score for that hand is doubled.

Each card can only appear once in the list, however a Joker may appear twice (as a deck contains two Jokers), however a Joker cannot appear three or more times.

Requirements

- The application's back-end must be written in C#.
- The code to convert the list of cards to their score must be performed in the back-end C# code.
- The application must contain a user interface, which allows the user to enter their list of cards.
- The list of cards must be given as a comma separated list.
- Each card must use a two-character representation as follows:
    - The first character represents the card's value:
        - 2-9 for the card values 2 through 9

        - T for 10
        - J for a Jack
        - Q for a Queen
        - K for a King
        - A for an Ace
    - The second character represents the card’s suit:
        - C for Clubs
        - D for Diamonds
        - H for Hearts
        - S for Spades
    - *E.g. “AS” would be the Ace of Spades, “4C” would be the 4 of Clubs etc.*
    - A Joker is represented by two-character code “JK”.
- The score must be displayed on the user interface.
- The application must convert any valid list of cards to their score.
- The application must display an error message if the user enters an invalid list of cards.
- An error message must be displayed if the user duplicates a card in their list. *Note however that a Joker can appear twice, so an error message must be displayed if three or more Jokers appear in the list.*
- All code must be written using TDD.
- There must be 100% unit test coverage.

Scenarios

Scenario: Convert a List of Cards to their Score

Given I have started the Card Game application
When I enter a ‘<listOfCards>’
Then the correct ‘<score>’ should be displayed on the user interface

Examples:

listOfCards	**score**
2C	2
2D	4
2H	6
2S	8
TC	10
JC	11
QC	12
KC	13
AC	14
3C,4C	7
TC,TD,TH,TS	100

Scenario: Jokers

Given I have started the Card Game application
When I enter a ‘<listOfCards>’ containing one or two Jokers
Then the ‘<score>’ for the hand should be doubled and displayed on the user interface

Examples:

listOfCards	**score**
JK	0

JK,JK	0
2C,JK	4
JK,2C,JK	8
TC,TD,JK,TH,TS	200
TC,TD,JK,TH,TS,JK	400

Scenario: Invalid Lists

Given I have started the Card Game application

When I enter a '<listOfCards>' that's invalid

Then an '<errorMessage>' should be displayed on the user interface

Examples:

listOfCards	**errorMessage**
1S	Card not recognised
2B	Card not recognised
2S,1S	Card not recognised
3H,3H	Cards cannot be duplicated
4D,5D,4D	Cards cannot be duplicated
JK,JK,JK	A hand cannot contain more than two Jokers
2S|3D	Invalid input string