Deck of Cards

Main Classes

Card: This class represents the playing card. Since every card has a suit and value, the constructor for this class requires both suit and the card value to instantiate the class. This class also implements the IComparable interface since we need to compare cards to sort the hand.

Deck: This class represents the deck of playing cards. So the constructor of this class creates a list and adds all the possible suits and card values of cards to the list (which represent the deck).

Hand: This class contains a list of cards in a hand held by a player.

Player: This class represents the player with name property and hand object.

Additional Class

Constants: This method contains the constant for the number of players and number of cards in a hand. It also contains the Enums for the **CardValue** and the **Suit** of cards.

Interfaces

IDeck: This interface exposes the following methods for the deck.

- **Shuffle()**: This method shuffles the deck
- **DealCard()**: This method deals the card on the top of deck to a player
- **ResetCards()**: This method basically resets the game and sets the original deck back.

IPlayer: This interface exposes the following methods relevant for the player

- **ReceiveCard()**: This method adds the card to the hand once it is dealt by the dealer.
- **ShowHand()**: This method is used to display all the cards currently held by the player

IHand: This interface exposes the following methods for a hand

- **AddCard()**: This method adds the card received by the player to the list of cards in the hand.
- **GetCardsInHand()**: This method returns the number of cards held by the player.
- **ShowHand()**: This method returns all the cards in the hand

Exceptions

EmptyDeckException: This method represents the exception when the dealer tries to deal the card out of an empty deck

Unit Test Scenarios:

Scenario#1: When the dealer tries to deal card out of an empty deck we should expect the EmptyDeckException.

Scenario#2: When we try to show a hand when no cards are dealt to the player we should get an error message

Scenario#3: When we create the deck and we deal all the cards, no two cards are the same

Scenario#4: When we deal certain cards for a player, the number of cards in the hand is equal to the number of cards dealt

Scenario#5: When we write the output to the file, we can reopen the file and read whether the output is correct.

Main Program Methods:

GetPlayers(): This method instantiates the list of players

- Input: number of hands on the game
- Output: array of hands

DealCards(): This method instantiates the list of players

- Input: deck, array of players, number of cards to be given to each player
- Output: none

GetAllPlayerHands(): This method creates the output in a string builder and returns a string.

- Input: array of players
- Output: string containing all the hands

WriteAllPlayerHands(): This method writes the output of all the player hands in a game to a file. The filename is part of the appSettings in the App.config file.

```
<add key="outputFileName" value="output.txt"/>
```

However, if the config is missing, the code just writes it to a file output.txt which can be found in the bin folder.

- Input: array of players and the output file path
- Output: none

WriteAllPlayerHands(): This method writes the output of all the player hands in a game to the Console. This is an overloaded method for previous method.

- Input: Array of players
- Output: None