**Euchre Project Part 2**

**Due, Wednesday, June 13, 2018 for maximum 100%**

**Thursday, June 14, 2018 for maximum 90%**

**Friday, June 15, 2018 for maximum 80%**

**Saturday, June 16, 2018 for maximum 70%**

**Assignment Scope**

1. Add methods to classes to accomplish specific tasks
2. Use class HashSet
3. Convert a HashSet to an ArrayList
4. Convert an ArrayList to a HashSet
5. Iterate through a HashSet
6. Generate objects of specific classes
7. Compile and run a project
8. Compress a project and submit to Webcourses
9. Decompress compressed project and verify it is a Netbeans project

**References**

1. Netbeans.docx
2. Setting up a project in Netbeans.docx
3. Netbeans right click menu help.docx

**Deliverables**

To complete this assignment you must submit your **compressed Netbeans project** to Webcourses.

**Tasks**

|  |  |
| --- | --- |
| Activity | |
| Euchre project |  |
| Euchre class |  |
| constants |  |
| Constants class |  |
| core package |  |
| AiPlayer class |  |
| Card class | Update class to include:   1. Method hashCode, it shall    1. Return type int    2. Receive no parameters    3. Include local variable hashCode of data type int initialized to the value of zero    4. Concatenate member variable face and its hashcode    5. Concatenate member variable value and its hashcode    6. Concatenate member variable color and its hashcode    7. Return the local variable hashCode 2. Method equals, it shall    1. Return type boolean    2. Receive one parameter of data type Object    3. Check if the parameter is an instanceof class Card       1. If true          1. Explicitly convert the parameter to an instance of class Card          2. Return the result of comparing if the face, suit, and color match an existing object          3. Code example:   return (card.face.equals(this.face) &&  card.color.equals(this.color) &&  card.suit.equals(this.suit));   * + 1. Else        1. Return false |
| Deck class | Update class to   1. Write a custom constructor that    1. Receives no parameters    2. Calls method generateDeck()    3. Calls method displayDeck()    4. Calls method shuffleDeck()    5. Calls method displayDeck() 2. Write method generateDeck so that    1. Return type is void    2. Receives no parameters    3. Instantiates the member variable of type Set calling the constructor for class HashSet    4. Loops through the values of enumeration Face       1. Loops through the values of enumeration Suit          1. Instantiates an instance of class Card          2. Sets the face value of the card          3. Sets the suit of the card          4. Determines the color of the card based on the suit and sets the color of the card          5. Verifies the instance of Card created is not contained in the HashSet of cards             1. If it does not exist, add the instance of class Card to the HashSet 3. Write method displayDeck so that    1. Return type is void    2. Receives no parameters    3. Iterates through the HashSet collection outputting to the console the face value, suit, and color of each card 4. Write method shuffleDeck so that    1. Instantiates an instance of class ArrayList, explicitly for data type of class Card passing the member variable of interface Set as an argument    2. Call static method Collections.shuffle passing the ArrayList from above as an argument    3. Reinstantiate the member variable of interface Set by calling the constructor for class HashSet passing the ArrayList above as an argument |
| Game class | Update class to   1. Update the custom constructor to call method generateDeck() 2. Add method generateDeck so that    * + 1. Return type is void        2. Receives no parameters        3. Instantiates the member variable of class Deck |
| HumanPlayer class |  |
| IPlayer interface |  |
| Player class |  |
| Team class |  |
| userinterface package |  |
| Euchre application |  |
| Test Case 1 | Test Case 1 passes |
| Test Case 2 | Test Case 2 passes |
| Test Case 3 | Test Case 3 passes |
| Test Case 4 | Test Case 4 passes |
| Test Case 5 | Test Case 5 passes |
|  | Source compiles with no errors |
|  | Source runs with no errors |
|  | Source includes comments |

**Perform the following test cases**

|  |  |  |
| --- | --- | --- |
| Test Cases | | |
|  | **Action** | **Expected outcome** |
| Test Case 1 | **Project view** | Completed project view should look like figure 1 |
| Test case 2 | **Regression testing; Run application** | The console window should look like figure 2 |
| Test case 3 | **Regression testing; Run application** | The JOptionPane.showMessageDialog() method call should look like figure 3 |
| Test case 4 | **Run application** | The console window should look similar to figure 4 for the initial display of the deck of cards |
| Test case 5 | **Run application** | The output in the console window should display the deck of cards after being shuffled and should NOT be in the same order as the first output, similar to figure 5 |

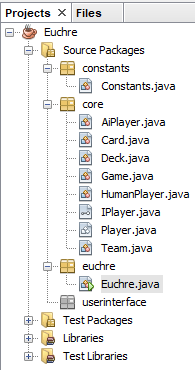


Figure 1 Project View

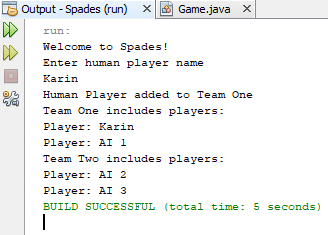


Figure 2 Output in console window

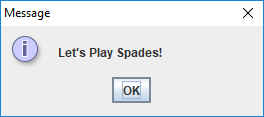


Figure 3 Display from JOptionPane.showMessageDialog() method



Figure 4 Initial display of deck of cards



Figure 5 Display of deck of cards after shuffle