**Bridge Dealing Program**

**Classes & Description:**

**Cards –** Array holding the types of cards a player has in their hand.

**PlayerNum –** A value between 1-4 indicating which player is which.

**HandHcp** – Cotains the calculated value of the players hand – calculated after being dealt cards.

**CalculateHcp()** – Method to calculate the current HCP value of the hand.

**listCards()** – output the textual names of the cards in the players hand.

**Suit** – Simple variable that’ll have a single character to dictate the suit of the card.

**faceNumber** – Pretty self-explanatory.

**hcpVal** – Value of the card from 1 to 4.

Deck contains 52 instances of card, to replicate an actual playing card deck.

createDeck() is the equivalent of a shuffle, it will create 52 unique instances of card inside the deck array. Read in from file probably easiest.

Shuffle mixes up the order of the deck array somewhat randomly.

Manager has access to the deck class, and if the full game was built fully, it would also hold gameplay-related functions like resetting the table, etc.

|  |
| --- |
| **Hand** |
| **Card[]** cards;  **Int** playerNum;(1-4)  **Int** handHcp; |
| **calculateHcp();**  **listCards();** |

|  |
| --- |
| **Card** |
| **Char** suit; (‘DHCS’)  **Int** faceNumber;  **Int** hcpVal; |
|  |

|  |
| --- |
| **Deck** |
| **Card[]** deck; |
| **createDeck();**  **shuffle();** |

|  |
| --- |
| **Manager** |
| **Hand[]** players; |
| **dealDeck();** |

**Pseudo-Code (In rough order of execution):**

An instance of deck runs createDeck.

createDeck(){

for **52 times**{

deck[i] = **new card by reading line X of text file to get suit/value/hcp.**

}

}

Deck array is then shuffled with a basic swap algorithm

Shuffle{

Card a **= random number 1~52**

Card b **= random number 1~52 excluding a.**

**Swap algorithm to move a & b’s place.**

}

Initialize each players hand, and setup a currently empty arraylist of type Card. Then feed those hands into the array in the manager class. Each instance has a unique playerNum from 0-3

Manager then uses dealDeck() to divvy the cards to each player.

dealDeck(){

handsize = 13;

playerCount = 4;

for **each time playerCount is less than 4{**

for **each time hand.size is less than handsize variable{**

playerNum.hand[playerCount] = deck[i]

**remove** **deck[i] from deck arraylist**

**}**

}

}

After all cards are dealt, calculateHcp() is run.

handHcp = **the total value of all hcpVal attributes in the hand array.**

Then all cards in the players hand (and total hcp) are textually output to the form with a for loop similar to the dealDeck() method but with write’s instead of adding/removing deck cards.