Vincent Curran Project 1 Part 2

Problem 1 : Cards Game

Code:

```
public class Card {
   // cards have a suit and rank
   int suit_;
   int rank_;

   // if no value given initialize a dummy card that results in error if read
   public Card() {
       suit_ = -1;
       rank_ = -1;
   }

   // initialize Card with suit and rank
   public Card(int suit, int rank) {
       suit_ = suit;
       rank_ = rank;
   }

   String ConvertSuit() {
       // convert suit int to an actual suit, error if theres no match
       switch (suit_) {
           case 0:
               return "Clubs";
           case 1:
               return "Diamonds";
           case 2:
               return "Hearts";
           case 3:
               return "Spades";
           case 98:
           case 99:
               return "Joker";
           default:
               return "Suit Error : " + suit_;
       }
   }

   String ConvertRank() {
       switch (rank_) {
```

```
            // for regular nums convert int to string
            case 2:
            case 3:
            case 4:
            case 5:
            case 6:
            case 7:
            case 8:
            case 9:
            case 10:
                return String.valueOf(rank_);
            // cases for face cards
            case 11:
                return "Jack";
            case 12:
                return "Queen";
            case 13:
                return "King";
            case 14:
                return "Ace";
            case 98:
            case 99:
                return "Joker";
            // if no cases match report error
            default:
                return ("Rank Error : " + String.valueOf(rank_));
        }
    }

    void Display() {
        // display the cards properties
        System.out.println(ConvertRank() + " of " + ConvertSuit());
    }
}
```

```
import java.util.Random;

public class Deck {
    // a deck is a collection of cards
    private Card[] cards_;
    // current number of cards in deck
```

```
private int num_cards_;

public Deck() {
    // initialize deck with 54 cards every time
    cards_ = new Card[54];
    Reset();
}

void ShowDeck() {
    // print the whole deck
    for (Card card  : cards_) {
        card.Display();
    }
}

int ChooseCard() {
    // return a random index in rage of cards available
    Random rand = new Random();
    int rando = rand.nextInt(num_cards_);
    return rando;
}

void DrawCards(int n) {
    System.out.println("Drawing " + String.valueOf(n) + " cards");
    if (n < 0) {
        // print error for negative input
        System.out.println("Error: Input must be 0 or greater");
    } else if (n > 54) {
        // print error for input too high
        System.out.println("Error: Input must be 54 or less");
    } else if (n == 0) {
        // print blank line if they want 0 cards
        System.out.println("");
    } else {
        // for valid input draw however many cards user wants
        for (int i = 0; i < n; i ++) {
            int c_i = ChooseCard();
            cards_[c_i].Display();
            RemoveCard(c_i);
        }
    }
    // reset the deck after every draw of cards
```

```
        Reset();
    }

    void RemoveCard(int index) {
        // move every card after the one to remove to the left
        for (int i = index; i < num_cards_ - 1; i++) {
            cards_[i] = cards_[i + 1];
        }
        num_cards_--;
    }

    void Reset() {
        // reset deck to initial state
        num_cards_ = 54;
        int counter = 0;
        for (int i = 0; i < 4; i++) {
            for (int j = 2; j < 15; j++) {
                cards_[counter] = new Card(i, j);
                counter++;
            }
        }

        cards_[52] = new Card(99, 99);
        cards_[53] = new Card(99,99);
    }
}
```

```
import java.util.Scanner;

public class App {
    static void Run() {
        // welcome user and initialize scanner and deck
        System.out.println("Welcome to the card drawing simulation");
        Scanner scanner = new Scanner(System.in);
        Deck deck = new Deck();
        boolean cont = true;
        while (cont) {
            // get user input
            System.out.println("How many cards would you like to draw?");
            int input;
            // check user inout is valid
```

```
            try {
                // draw card for valid input
                input = Integer.parseInt(scanner.nextLine());
                deck.DrawCards(input);
            } catch (Exception e) {
                // print error if input is invalid
                System.out.println("Error: Input must be an integer");
            }
            String str_input;
            do {
                // see if user wants to draw more cards
                System.out.println("Would you like to draw another set of cards?");
                str_input = scanner.nextLine();
                if ((str_input.charAt(0) == 'n') || (str_input.charAt(0) == 'N') ) {
                    // if the user doesn't want to draw another card end the program
                    cont = false;
                }
            } while (str_input.charAt(0) != 'n' && str_input.charAt(0) != 'N' &&
str_input.charAt(0) != 'y' && str_input.charAt(0) != 'Y');
        }
        // say bye and close the scanner
        System.out.println("Goodbye");
        scanner.close();
    }

    public static void main(String[] args) throws Exception {
        Run();
    }
}
```

Output:

```
Welcome to the card drawing simulation
How many cards would you like to draw?
-3
Drawing -3 cards
Error: Input must be 0 or greater
Would you like to draw another set of cards?
yes
How many cards would you like to draw?
1
Drawing 1 cards
10 of Hearts
Would you like to draw another set of cards?
YES
How many cards would you like to draw?
5
Drawing 5 cards
Joker of Joker
9 of Clubs
3 of Clubs
7 of Spades
Queen of Spades
Would you like to draw another set of cards?
y
How many cards would you like to draw?
10
Drawing 10 cards
10 of Spades
King of Clubs
3 of Hearts
4 of Hearts
10 of Diamonds
8 of Clubs
6 of Spades
5 of Spades
9 of Spades
2 of Spades
Would you like to draw another set of cards?
ye
How many cards would you like to draw?
20
Drawing 20 cards
10 of Clubs
Jack of Clubs
4 of Spades
3 of Clubs
5 of Diamonds
10 of Diamonds
8 of Clubs
3 of Hearts
Queen of Clubs
3 of Spades
Ace of Hearts
King of Spades
6 of Diamonds
2 of Hearts
6 of Hearts
9 of Hearts
King of Clubs
Queen of Diamonds
Queen of Hearts
4 of Diamonds
Would you like to draw another set of cards?
yep
```

```
Would you like to draw another set of cards?
yep
How many cards would you like to draw?
54
Drawing 54 cards
Ace of Clubs
6 of Spades
King of Spades
9 of Diamonds
Ace of Diamonds
7 of Hearts
Queen of Spades
3 of Spades
Queen of Diamonds
2 of Clubs
9 of Hearts
8 of Clubs
10 of Hearts
8 of Hearts
6 of Hearts
10 of Diamonds
6 of Clubs
Joker of Joker
Joker of Joker
Jack of Spades
King of Clubs
3 of Clubs
9 of Clubs
4 of Diamonds
Queen of Hearts
4 of Spades
Queen of Clubs
Jack of Clubs
5 of Spades
8 of Spades
Jack of Diamonds
10 of Spades
Ace of Spades
7 of Diamonds
4 of Hearts
7 of Spades
3 of Hearts
King of Diamonds
Ace of Hearts
7 of Clubs
2 of Hearts
2 of Spades
2 of Diamonds
4 of Clubs
3 of Diamonds
6 of Diamonds
King of Hearts
5 of Clubs
10 of Clubs
8 of Diamonds
5 of Diamonds
5 of Hearts
9 of Spades
Jack of Hearts
Would you like to draw another set of cards?
yes, please
```

```
Would you like to draw another set of cards?
yes, please
How many cards would you like to draw?
55
Drawing 55 cards
Error: Input must be 54 or less
Would you like to draw another set of cards?
Y
How many cards would you like to draw?
0
Drawing 0 cards

Would you like to draw another set of cards?
no
Goodbye
(base) vincentcurran@Vincents-MacBook-Pro-2 src %
```

Fun fact : the odds of getting the 2 Jokers in a row like I did in the draw of 54 is ~0.0007

Problem 2 - Matching Game

```java
import java.util.Scanner;
import java.util.Random;

public class Matching {
   // use same 12 words every time
   static String[] words_ = new String[] {
       "CLASS",
       "FLAGS",
       "SHADE",
       "DRIVE",
       "ARRAY",
       "ABOVE",
       "ERROR",
       "MAJOR",
       "LEARN",
       "ENTRY",
       "SKILL",
       "PITCH"
   };
```

```
static void Run() {
       // initialize rand and scanner
       Random rand = new Random();
       Scanner scanner = new Scanner(System.in);
       // pick the index of the word to be guessed
       int index = rand.nextInt(12);
       boolean cont = true;
       while (cont) {

           // ask user for a guess
           System.out.println("Guess a 5-letter word: ");
           String guess = scanner.nextLine();
           if (guess.length() == 0) {
               // if empty input then exit program
               System.exit(0);
           } else if (guess.length() != 5) {
               // if guess length is too big or small report error
               System.out.println("Please input a 5 letter word next time.");
           } else {
               int matches = 0;
               // if we have valid 5 letter word convert to uppercase for comparison
               guess = guess.toUpperCase();
               // check each letter of the guess against the real word
               // give information wether each letter is correct, in the wrong
position, or not in the word
               for (int i = 0; i < 5; i++) {
                   if (guess.charAt(i) == words_[index].charAt(i)) {
                       System.out.println(guess.charAt(i) + " is a match in the
correct location.");
                       matches++;
                   } else if (words_[index].contains(String.valueOf(guess.charAt(i))))
{
                       System.out.println(guess.charAt(i) + " is in the word, but at a
different location.");
                   } else {
                       System.out.println(guess.charAt(i) + " is not in the word");
                   }
               }
               // if all 5 letters match then tell user they win and end program
               if (matches == 5) {
```

```
                    System.out.println("You've guessed the correct word!
Congratulations!");
                    cont = false;
                }
            }
        }
        scanner.close();
    }
}
```

```
public class App {
    public static void main(String[] args) throws Exception {
        Matching.Run();
    }
}
```

Output:

```
(base) vincentcurran@Vincents-MacBook-Pro-2 src % cd '
Guess a 5-letter word:
CLASS
C is not in the word
L is not in the word
A is not in the word
S is not in the word
S is not in the word
Guess a 5-letter word:
DRIVE
D is a match in the correct location.
R is a match in the correct location.
I is a match in the correct location.
V is a match in the correct location.
E is a match in the correct location.
You've guessed the correct word! Congratulations!
(base) vincentcurran@Vincents-MacBook-Pro-2 src % cd '
Guess a 5-letter word:
CLASS
C is not in the word
L is not in the word
A is not in the word
S is not in the word
S is not in the word
Guess a 5-letter word:
DRIVE
D is not in the word
R is a match in the correct location.
I is not in the word
V is not in the word
E is in the word, but at a different location.
Guess a 5-letter word:
entry
E is a match in the correct location.
N is not in the word
T is not in the word
R is in the word, but at a different location.
Y is not in the word
Guess a 5-letter word:
enterrrrr
Please input a 5 letter word next time.
Guess a 5-letter word:
fe
Please input a 5 letter word next time.
Guess a 5-letter word:
error
E is a match in the correct location.
R is a match in the correct location.
R is a match in the correct location.
O is a match in the correct location.
R is a match in the correct location.
You've guessed the correct word! Congratulations!
(base) vincentcurran@Vincents-MacBook-Pro-2 src % cd '
Guess a 5-letter word:

(base) vincentcurran@Vincents-MacBook-Pro-2 src %
```