Java Lab Assignment 4

BY

Vaibhav Jaiswal

PRN=17070122071

CS-4

SY CSE

Batch of 2017-2021

CODE :

import java.util.\*;

import java.io.\* ;

import java.lang.\* ;

class Card implements Comparator<Card>{

int rank ;

int suit ;

Card(){

rank = -1 ;

suit = -1 ;

}

Card(int rank , int suit){

this.rank = rank ;

this.suit = suit ;

}

public String toString(){

String suits[] = {"Diamonds" , "Club" , "Heart" , "Spades"} ;

String ranks[] = {"Ace" , "2" , "3" , "4" , "5" , "6" , "7" , "8" , "9" ,"10" , "Jack" , "Queen" , "King"} ;

return (ranks[this.rank-1] + " of " + suits[this.suit-1]) ;

}

/\*Print Card in a Rectange Form\*/

public static void printCard(Card a ){

char ch = 'D' ;

switch(a.suit){

case 1:

ch = 'C' ;

break ;

case 2:

ch = 'H' ;

break ;

case 3:

ch = 'S' ;

break ;

}

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_") ;

for(int i = 0 ; i < 12 ; i ++){

if (i == 0){

if(a.rank <10 )

System.out.println("| " + a.rank+'|') ;

else

System.out.println("| " + a.rank+'|') ;

}

else if (i == 5){

System.out.println("| "+ch+" |") ;

}

else if (i == 11){

if(a.rank<10)

System.out.println("|" + a.rank + " |") ;

else

System.out.println("|" + a.rank + " |") ;

}

else

System.out.println("| |") ;

}

System.out.println("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_|") ;

}

public static boolean sameCard(Card a , Card b){

if(a.rank == b.rank && a.suit == b.suit ){

return true ;

}

else return false ;

}

public int compare(Card a , Card b){

if (a.suit > b.suit){

return 1 ;

}

else if (a.suit < b.suit){

return -1 ;

}

else if (a.suit == b.suit ){

if(a.rank > b.rank){

return 1 ;

}

else if(a.rank < b.rank){

return -1 ;

}

else return 0 ;

}

return 0 ;

}

public static void compareCard(Card a , Card b){

Card tmp = new Card() ;

int result = tmp.compare(a,b) ;

if(result == 1 )

System.out.println("Card : "+a.toString()+" is greater card than " + b.toString() + " \n");

else if(result == -1 )

System.out.println("Card : "+a.toString()+" is Less card than " + b.toString() + " \n");

else

System.out.println("Card : "+a.toString()+" is equal to " + b.toString() + " \n");

}

@Override

protected void finalize() throws Throwable

{

System.out.println("Garbage collector called");

System.out.println("Object garbage collected : " + this + "\n");

}

} ;

class Deck{

ArrayList<Card> cards = new ArrayList<Card>() ;

Deck(){

for(int suit = 1 ; suit <= 4 ; suit++){

for(int rank = 1 ; rank <= 13 ; rank++){

this.cards.add(new Card(rank , suit)) ;

}

}

}

public static void printDeck(Deck deck){

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

for (Card tmp : deck.cards)

System.out.println(tmp) ;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

public static void shuffleDeck(Deck deck){

Collections.shuffle(deck.cards);

}

public static void sortCard(Deck deck ){

Collections.sort(deck.cards , new Card()) ;

System.out.println("Deck has been Sorted ") ;

}

public Card getCard(int pos){

return this.cards.get(pos) ;

}

public void dealCards(){

Random ran = new Random() ;

System.out.println("Cards Dealt : \n") ;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

for (int i = 0 ; i < 5 ; i++ ){

Card.printCard(this.cards.get(ran.nextInt(51)));

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

public void findCard(Card a){

boolean found = false ;

int indx = 0 ;

for( Card tmp : this.cards){

if(Card.sameCard(tmp , a)){

found = true ;

break ;

}

indx++ ;

}

if(found)

System.out.println("Card : "+ a.toString()+ " is present in deck at : "+ (indx+1) + " Position \n") ;

else

System.out.println("Card : "+ a.toString() + " is not present in Deck \n") ;

}

@Override

protected void finalize() throws Throwable

{

System.out.println("Garbage collector called");

System.out.println("Object garbage collected : " + this + "\n");

}

}

class Driver{

public static void main(String[] args) {

Deck deck = new Deck() ;

int choice ;

boolean exit = false ;

Scanner ip = new Scanner(System.in) ;

do{

System.out.println("Operations : ");

System.out.println("1. Shuffle Deck ");

System.out.println("2. Sort Deck : ");

System.out.println("3. Check if 2 Cards is Same or Not : ");

System.out.println("4. Deals 5 Random Cards from Deck : ");

System.out.println("5. Compare 2 Cards : ");

System.out.println("6. Find a card in deck : ");

System.out.println("Your choice : ");

choice = ip.nextInt() ;

switch(choice){

case 1:

/\*Shuffle Deck Function \*/

System.out.println("Deck Shuffled");

Deck.shuffleDeck(deck);

Deck.printDeck(deck) ;

break ;

case 2:

/\*Sort Deck Function \*/

System.out.println("Deck Sorted");

Deck.sortCard(deck);

Deck.printDeck(deck) ;

break ;

case 3:

{//Same Card Function

int card1 , card2 ;

System.out.println("Enter indx from 1-52 of first card to compare : ");

card1 = ip.nextInt() ;

System.out.println("Enter indx from 1-52 of Second card to compare : ");

card2 = ip.nextInt() ;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.out.println("Are Cards Same: ");

System.out.println(Card.sameCard(deck.getCard(card1-1) , deck.getCard(card2-1)));

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

break ;

}

case 4:

{

// Deal Card Function

System.out.println("Deal Cards Function : ");

deck.dealCards() ;

break ;

}

case 5:

{

//Compare Card Function

int card1 , card2 ;

System.out.println("Enter indx from 1-52 of first card to compare : ");

card1 = ip.nextInt() ;

System.out.println("Enter indx from 1-52 of Second card to compare : ");

card2 = ip.nextInt() ;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.out.println("Compare Cards Function \n");

Card.compareCard(deck.getCard(card1-1) ,deck.getCard(card2-1)) ;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

break ;

}

case 6:

{

int suit , rank ;

System.out.println("Enter Suit of card to find : \n1.Diamond \n2.Club\n3.Heart\n4.Spades\nYour Input :") ;

suit = ip.nextInt() ;

System.out.println("Enter Rank of card to find : from 1-10 , 11 for Jack , 12 for Queen , 13 for King\nYour Input : ");

rank = ip.nextInt() ;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.out.println("Find Card Function \n");

deck.findCard(new Card(rank , suit)) ;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

break ;

}

case 7:

Card d = new Card(1,2) ;

d = null ;

System.gc() ;

Card d1 = new Card(4,2) ;

d1 = null ;

System.gc() ;

exit = true ;

break ;

}

}while(!exit) ;

deck = null ;

System.gc() ;

ip.close() ;

}

}

PTO

OUTPUT:



