Name. JATIN LAMBA

PRN. 21070126042

Batch. AIML – A2

Assignment – 4

Write a menu-driven Java Program for the following:

There are 52 cards in a deck, each of which belongs to one of four suits and one of 13 ranks.

Represent a deck of cards as an array of

Objects (\*you may use the Vector class)

1. Use integers to encode the ranks and suits.
2. Have suitable default & parameterized constructors.
3. all data members to have private access.
4. The class ‘Card’ to have the following methods:

createDeck(), printCard(), printDeck (),sameCard(),compareCard(), sortCard(), findCard() which searches through an array or vector of Cards to see whether it contains a certain card, dealCards() function: to print 5 random cards from the existing deck.

Code.

Card.

package mypackage;

public class Card {

private int suit, rank; private String[] suits = {"Clubs", "Diamonds", "Hearts", "Spades"}; private String[] ranks = {"2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace"};

public int getSuit() { return suit;

}

public void setSuit(int suit) { this.suit = suit;

}

public int getRank() { return rank;

}

public void setRank(int rank) { this.rank = rank;

}

// Constructor public Card() { this.suit = 0; this.rank = 0;

}

// Constructor public Card(int suit, int rank){ this.suit = suit; this.rank = rank;

}

// Print this card public void printCard() {

System.out.println(ranks[rank] + " of " + suits[suit]);

}

// Compare with another Card public int compareCard(Card toCompare) {

/\*

* Return 1 if this Card is greater than toCompare
* Return 0 if cards are equal
* Return -1 if toCompare is greater

\*/

if(this.suit > toCompare.suit) return 1; if(this.suit < toCompare.suit) return -1;

// Suits are equal if(this.rank > toCompare.rank) return 1; if(this.rank < toCompare.rank) return -1;

// Cards are the same return 0;

}

// return a new card just like this one public Card sameCard() {

Card toReturn = new Card(suit, rank); return toReturn;

} }

Deck.

package mypackage;

import java.util.ArrayList;

public class Deck {

private ArrayList<Card> deck = new ArrayList<Card>();

public void createDeck() { for(int i=0; i<4; i++) { for(int j=0; j<13; j++) { deck.add(new Card(i,j));

}

}

}

// Print the current deck

public void printDeck() { for(Card c:deck) { c.printCard();

}

}

// Deal 5 cards and print and remove them from deck public void dealCards() { int randomIndex; Card c; for(int i=0; i<5; i++) { randomIndex = (int)(Math.random() \* deck.size()); c = deck.get(randomIndex); c.printCard(); deck.remove(randomIndex);

}

}

// Returns 1 if the card is found in deck, else 0.

public int findCard(Card toFind) { if(deck.contains(toFind)) return 1; return 0;

} }

CardsClass.

package mypackage;

public class CardsClass {

public static void main(String[] args) {

Deck d = new Deck(); d.createDeck();

d.dealCards();

d.printDeck();

Card toFind = new Card(2, 4);

int found = d.findCard(toFind);

if(found == 1)

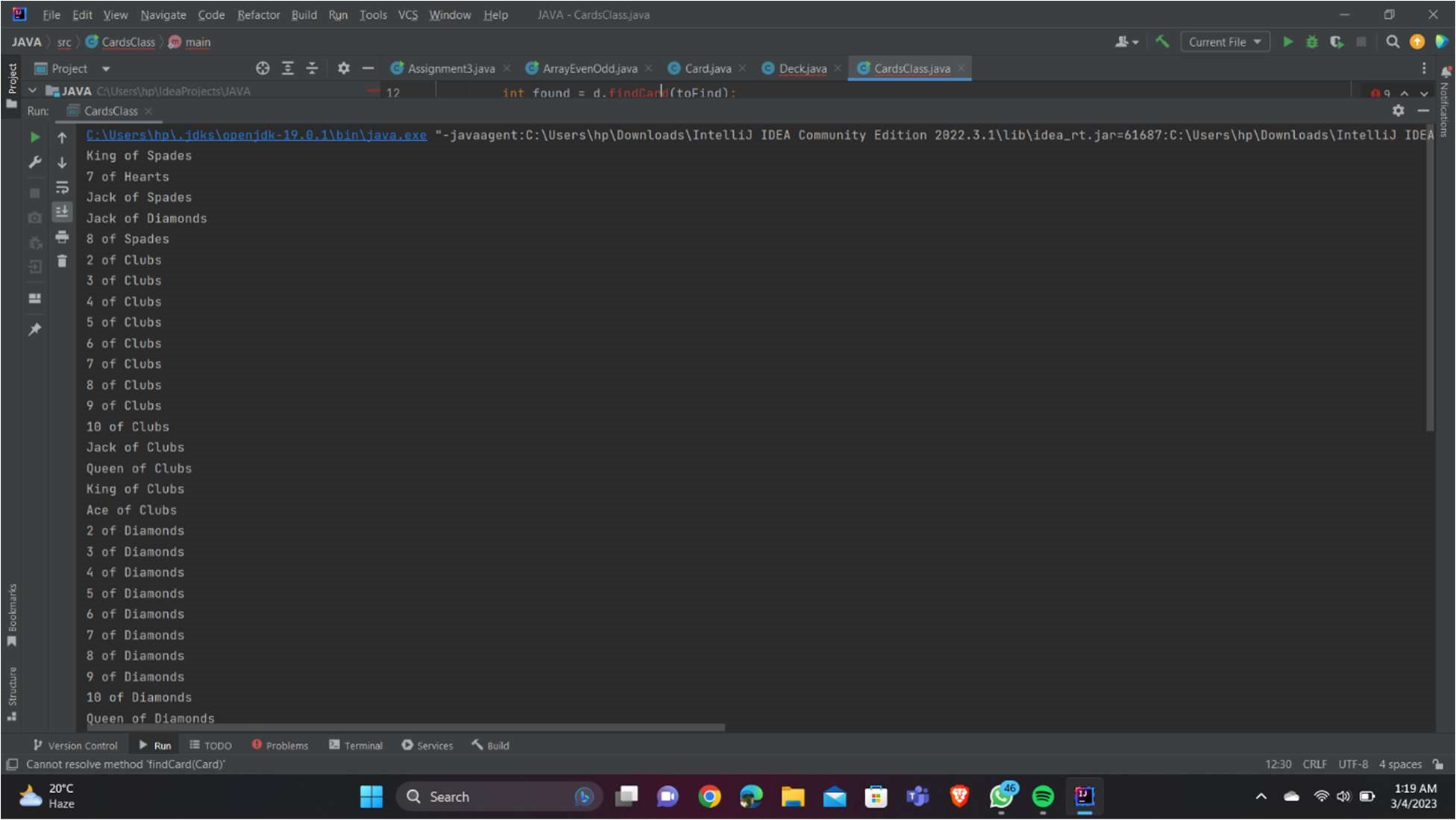
System.out.println("Card found.");

else

System.out.println("Card not found.");

} }

Output.



Github Link.

https://github.com/jatinlamba2025/java-sem-4/upload/main/java%20assignment%204