

```
package a_elevens;

import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.List;
import java.util.Scanner;
/// main program that runs the whole game
public class MainProgram {
    public static void main(String[] args) {

        List<Card> faceUp;
        int cardNumber1, cardNumber2, cardNumber3;
        Card card1, card2, card3, newCard;
        boolean valid;
        boolean face_valid = false;
        boolean win;
        Deck deck, deck_copy;
        ArrayList< ArrayList<Card> > history = new ArrayList<>();

        while(true){

            System.out.println("welcome to the games of ELEVEN");
            System.out.println("*****lets start the game***** ");
            System.out.println("choose number against the options");
            System.out.println("1.demo");
            System.out.println("2.play game");
            System.out.println("3.exit");
            System.out.println("Type your Answer: ");
            String prompt_demo = get_string_input();
            if(prompt_demo.equals("1")){
                play_demo();
            }
            if(prompt_demo.equals("3")){
                System.exit(0);
            }

            System.out.println("*****");
            System.out.println("*****Starting Game *****");
            System.out.println("*****");

            //entring the Deck
            deck = new Deck();
            deck_copy = new Deck(deck);

            //inseting the Board
            faceUp = new ArrayList<>();
```

```

        for(int i = 0; i < 9; i++) {
            faceUp.add(deck.get_one_card());
        }
        win = false;

        while(!is_end_reached(faceUp)){

/////////////////////////////////////////////////////////////////
//Print Cards output
System.out.println("*****GAMES OF
11*****");
        for (int i = 0; i < faceUp.size(); i++) {

            System.out.print(String.valueOf(i) + ": ");
            System.out.println(faceUp.get(i).toString());

        }
        System.out.println("number of cards in desk are : " +
deck.get_size());

System.out.println("*****");
/////////////////////////////////////////////////////////////////

//Choose Cards and valid invalid check
System.out.println("Do you want hint type 'yes' or 'no':");
String prommpt_yes_string = get_string_input();
System.out.println(prommpt_yes_string);
if(prommpt_yes_string.equals("yes")){
    hint(faceUp);
}

System.out.println("Choose the number in front of the card to pick
Card 1 :");
cardNumber1 = get_valid_input();
card1 = faceUp.get(cardNumber1);

System.out.println("Choose the number in front of the card to pick
Card 2 :");
cardNumber2 = get_valid_input();
card2 = faceUp.get(cardNumber2);

face_valid = false;
card3 = faceUp.get(cardNumber2); // card 3 in case card 1 and 2 are
less than 11 and can score 11

System.out.println(card1.toString()+" and " + card2.toString());

```

```

        if(card1.isFaceCard() & card2.isFaceCard()){

            System.out.println("Choose the number in front of the card to
pick Card 3 :");
            cardNumber3 = get_valid_input();
            card3 = faceUp.get(cardNumber3);

            if(card3.isFaceCard()){
                valid = true;
                face_valid = true;

                ArrayList<Card> a = new ArrayList<>();
                a.add(card1);
                a.add(card2);
                a.add(card3);
                history.add(a);
            }
            else{
                System.out.println("Invalid Selection,Please Try Again");// if we choose wrong card
                valid = false;
            }

        }
        else if(card1.isFaceCard() | card2.isFaceCard()){
            System.out.println("Invalid Selection,Please Try Again");
            valid = false;
        }
        else{
            if( card1.sum(card2) == 11){
                valid = true;
                ArrayList<Card> a = new ArrayList<>();
                a.add(card1);
                a.add(card2);
                history.add(a);
            }
            else{
                System.out.println("Invalid Selection");// after 3 turns
                valid = false;
            }
        }
    }

    if(valid){

        newCard = deck.get_one_card();
        if( newCard != null){
            faceUp.remove(faceUp.indexOf(card1));
            faceUp.add(newCard);
        }
        else{
    
```

```

        faceUp.remove(faceUp.indexOf(card1));
    }

    newCard = deck.get_one_card();
    if(newCard != null){
        faceUp.remove(faceUp.indexOf(card2));
        faceUp.add(newCard);
    }
    else{
        faceUp.remove(faceUp.indexOf(card2));
    }

    if(face_valid){
        newCard = deck.get_one_card();
        if(deck.get_one_card() != null){
            faceUp.remove(faceUp.indexOf(card3));
            faceUp.add(newCard);
        }
        else{
            faceUp.remove(faceUp.indexOf(card3));
        }
        face_valid = false;
    }

    if(faceUp.isEmpty()){
        win = true;
        break;
    }
}

System.out.println(faceUp.size());
System.out.println(faceUp);

}

if(win){
    System.out.println("*****");
    System.out.println("*****HOORAYYY!!*****");
    System.out.println("*****");
}
else{

System.out.println("*****");
    System.out.println("*****OOPS TRY
AGAIN*****");
}

System.out.println("*****");
}
```

```

        System.out.println("Want to replay please type yes or no ");
        String prompt_for_replay = get_string_input();
        if(prompt_for_replay.equals("yes")){
            replay(deck_copy, history);
        }
    }

//*****DEMO STARTS HERE
*****
public static void play_demo(){

    List<Card> faceUp;
    int cardNumber1, cardNumber2, cardNumber3;
    Card card1, card2, card3, newCard;
    boolean valid;
    boolean face_valid = false;
    boolean win;
    Deck deck, deck_copy;
    ArrayList< ArrayList<Card> > history = new ArrayList<>();

    System.out.println("*****");
    System.out.println("*****STARTING DEMO BY SAVI*****");
    System.out.println("*****");

    //setting up the Deck
    deck = new Deck();
    deck_copy = new Deck(deck);

    // setting up Board
    faceUp = new ArrayList<>();
    for(int i = 0; i < 9; i++) {
        faceUp.add(deck.get_one_card());
    }
    win = false;

    while(!is_end_reached(faceUp)){
        //Print the Output
        System.out.println("*****9 CARDS SHOW *****");
        for (int i = 0; i < faceUp.size(); i++) {

            System.out.print(String.valueOf(i) + ": ");
            System.out.println(faceUp.get(i).toString());

        }
        System.out.println("number of cards left in the deck:
"+deck.get_size());
        System.out.println("*****");
    }
}

```

```
//////////  
  
//////////  
    //Choose Cards and valid invalid check  
    System.out.println("Press Enter please");  
    String prommpt_yes_string = get_string_input();  
  
    List<Integer> next_move = get_next_move(faceUp);  
    cardNumber1 = next_move.get(0);  
    cardNumber2 = next_move.get(1);  
    cardNumber3 = next_move.get(1);  
    if(next_move.size() == 3){  
        cardNumber3 = next_move.get(2);  
    }  
  
    card1 = faceUp.get(cardNumber1);  
    card2 = faceUp.get(cardNumber2);  
    face_valid = false;  
    card3 = faceUp.get(cardNumber3);  
  
    if(card1.isFaceCard() & card2.isFaceCard()){  
  
        if(card3.isFaceCard()){  
            valid = true;  
            face_valid = true;  
  
            ArrayList<Card> a = new ArrayList<>();  
            a.add(card1);  
            a.add(card2);  
            a.add(card3);  
            history.add(a);  
        }  
        else{  
            System.out.println("Invalid Selection, Try Again");  
            valid = false;  
        }  
  
    }  
    else if(card1.isFaceCard() | card2.isFaceCard()){  
        System.out.println("Invalid Selection, Try Again");  
        valid = false;  
    }  
    else{  
        if( card1.sum(card2) == 11){  
            valid = true;  
            ArrayList<Card> a = new ArrayList<>();  
            a.add(card1);  
        }  
    }  
}
```

```

        a.add(card2);
        history.add(a);
    }
    else{
        System.out.println("Invalid Selection");
        valid = false;
    }
}

if(valid){

    newCard = deck.get_one_card();
    if( newCard != null){
        faceUp.remove(faceUp.indexOf(card1));
        faceUp.add(newCard);
        System.out.print("Removing : ");
        System.out.println(card1);
    }
    else{
        faceUp.remove(faceUp.indexOf(card1));
        System.out.print("Removing : ");
        System.out.println(card1);
    }

    newCard = deck.get_one_card();
    if(newCard != null){
        faceUp.remove(faceUp.indexOf(card2));
        faceUp.add(newCard);
        System.out.print("Removing : ");
        System.out.println(card2);
    }
    else{
        faceUp.remove(faceUp.indexOf(card2));
        System.out.print("Removing : ");
        System.out.println(card2);
    }
}

if(face_valid){
    newCard = deck.get_one_card();
    if(newCard != null){
        faceUp.remove(faceUp.indexOf(card3));
        faceUp.add(newCard);
        System.out.print("Removing : ");
        System.out.println(card3);
    }
    else{
        faceUp.remove(faceUp.indexOf(card3));
        System.out.print("Removing : ");
        System.out.println(card3);
    }
}

```

```

        }
        face_valid = false;
    }

    if(faceUp.isEmpty()){
        win = true;
        break;
    }
}

System.out.println(faceUp.size());
System.out.println(faceUp);

}

if(win){
    System.out.println("*****");
    System.out.println("*****WINNER!!!!*****");
    System.out.println("*****");
}
else{
    System.out.println("*****");
    System.out.println("*****TRY AGAIN~~~~~*****");
    System.out.println("*****");
}

System.out.println("Want to replay the demo say yes or no ");
String prompt_for_replay = get_string_input();
if(prompt_for_replay.equals("yes")){
    replay(deck_copy, history);
}

}

public static void replay(Deck deck, ArrayList< ArrayList<Card> > history){

    List<Card> faceUp;
    boolean win, valid;
    Card card1, card2, card3, newCard;
    boolean face_valid = false;

    //Initializing FaceUp Board
    faceUp = new ArrayList<>();
    for(int i = 0; i < 9; i++) {
        faceUp.add(deck.get_one_card());
    }
    win = false;

    while(!is_end_reached(faceUp)){

```

```

//////////////////////////////Print Cards Names Output
get_string_input();
System.out.println("*****9 CARD GAME*****");
for (int i = 0; i < faceUp.size(); i++) {

    System.out.print(String.valueOf(i) + ": ");
    System.out.println(faceUp.get(i).toString());


}
System.out.println(deck.get_size());
System.out.println("*****");

//////////////////THIS WILL CHECK IF CARDS SUM ARE VALID OR NOT
/////////////////
ArrayList<Card> move = history.get(0);
card1 = move.get(0);
card2 = move.get(1);
card3 = move.get(1);

if(card1.isFaceCard() & card2.isFaceCard()){
    card3 = move.get(2);
    valid = true;
    face_valid = true;
}
else{
    valid = true;
}
if(valid){

    newCard = deck.get_one_card();
    if( newCard != null){
        faceUp.remove(faceUp.indexOf(card1));
        faceUp.add(newCard);
    }
    else{
        faceUp.remove(faceUp.indexOf(card1));
    }

    newCard = deck.get_one_card();
    if(newCard != null){
        faceUp.remove(faceUp.indexOf(card2));
        faceUp.add(newCard);
    }
    else{
        faceUp.remove(faceUp.indexOf(card2));
    }
}

```

```

        if(face_valid){
            newCard = deck.get_one_card();
            if(deck.get_one_card() != null){
                faceUp.remove(faceUp.indexOf(card3));
                faceUp.add(newCard);
            }
            else{
                faceUp.remove(faceUp.indexOf(card3));
            }
            face_valid = false;
        }

        if(faceUp.isEmpty()){
            win = true;
            break;
        }
    }

    System.out.println("Total Number Of Cards Are: "+faceUp.size());
    System.out.println(faceUp);

}

public static void hint(List<Card> list_of_cards){

    if(!list_of_cards.isEmpty()){

        List<Card> faceCards = new ArrayList<>();
        int length = list_of_cards.size();
        boolean possible_move = false;

        if(length == 0){
            System.out.println("No possible Move Left");
        }

        for (int i = 0; i < length; i++) {
            if(list_of_cards.get(i).isFaceCard()){
                faceCards.add(list_of_cards.get(i));
            }
        }

        if(faceCards.size() > 2){
            System.out.print("Possible Move is any 3 of These: ");
            System.out.print(faceCards);
        }
    }
}

```

```

        possible_move = true;
    }

    for (int i = 0; i < length - 1; i++) {
        for(int j = i + 1; j < length; j++){
            if(list_of_cards.get(i).sum( list_of_cards.get(j)) == 11){
                System.out.print("Possible move is : ");
                System.out.println(list_of_cards.get(i).toString() + " +
+ list_of_cards.get(j).toString());
                possible_move = true;
            }
        }
    }

    if(!possible_move){
        System.out.println("No possible Move Left");
    }
}

else{
    System.out.println("You have won the game");
}
}

public static List<Integer> get_next_move(List<Card> list_of_cards){

    if(!list_of_cards.isEmpty()){

        List<Card> faceCards = new ArrayList<>();
        int length = list_of_cards.size();
        boolean possible_move = false;
        List<Integer> move_numbers = new ArrayList<>();

        if(length == 0){
            System.out.println("No possible Move Left");
        }

        for (int i = 0; i < length; i++) {
            if(list_of_cards.get(i).isFaceCard()){
                faceCards.add(list_of_cards.get(i));
                move_numbers.add(i);
            }
        }
    }
}

```

```

        }

        if(faceCards.size() > 2){
            return move_numbers;
        }

        move_numbers = new ArrayList<>();
        for (int i = 0; i < length - 1; i++) {

            for(int j = i + 1; j < length; j++){

                if(list_of_cards.get(i).sum( list_of_cards.get(j)) == 11){
                    move_numbers.add(i);
                    move_numbers.add(j);
                    possible_move = true;
                    return move_numbers;
                }
            }
        }

        if(!possible_move){
            System.out.println("No possible Move");
            return null;
        }

    }

    else{
        System.out.println("You have won the game ");
        return null;
    }

    return null;
}

public static boolean is_end_reached(List<Card> list_of_cards){

    if(!list_of_cards.isEmpty()){

        int face_card_counts = 0;
        int length = list_of_cards.size();

        if(length == 0){
            return true;
        }

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

```

```

        if(list_of_cards.get(i).isFaceCard()){
            face_card_counts = face_card_counts + 1;
        }
    }

    if(face_card_counts > 2){
        return false;
    }

    for (int i = 0; i < length - 1; i++) {

        for(int j = i + 1; j < length; j++){

            if(list_of_cards.get(i).sum( list_of_cards.get(j)) == 11){

                return false;
            }
        }
    }

    return true;
}
else{
    return true;
}
}

public static int get_valid_input(){

    int int_to_return = -1;

    while(int_to_return == -1){

        Scanner input_reader = new Scanner(System.in);
        if(input_reader.hasNextInt()){

            try{
                int_to_return = input_reader.nextInt();
            }
            catch(InputMismatchException exception){

                System.out.println("You Entered a non Integer");
            }
        }
        else{
            System.out.println("You Entered a non Integer");
        }
    }

    if((int_to_return > 9) | (int_to_return < 0)){

```

```
        System.out.println("Please enter value between 0 - 9");
        int_to_return = -1;
    }

    return int_to_return;
}

public static String get_string_input(){

    String string_to_return = " ";

    Scanner input_reader = new Scanner(System.in);
    if(input_reader.hasNextLine()){
        string_to_return = input_reader.nextLine();
    }

    return string_to_return;
}
}
```