National College of Ireland

Higher Diploma in Science in Computing

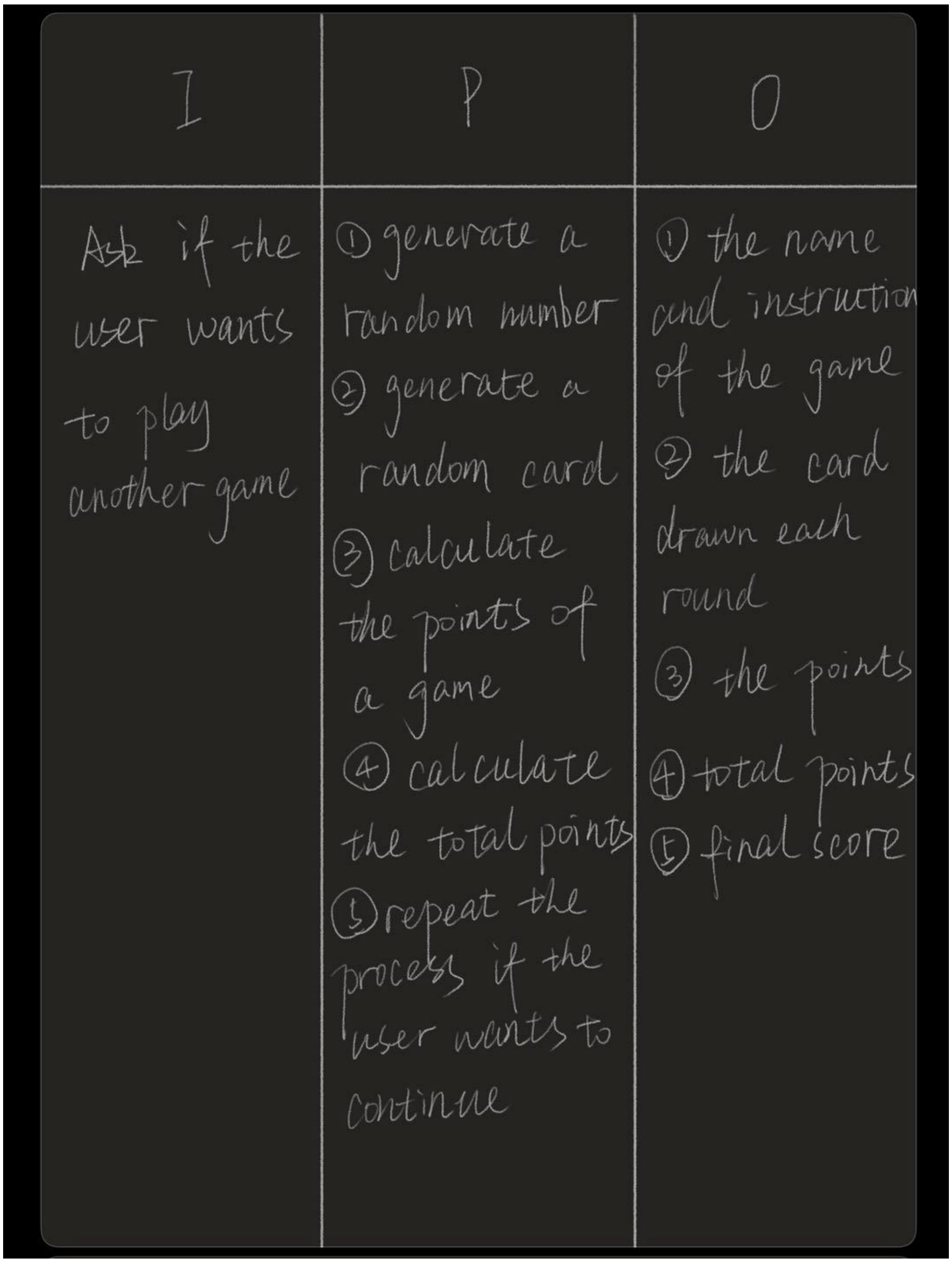
Software Development

Lecturer: Enda Stafford

Game of Chance: Card Game

Shan Liang 22187804

1. IPO



I	P	O
Ask if the user wants to play another game	① generate a random number ② generate a random card ③ calculate the points of a game ④ calculate the total points ⑤ repeat the process if the user wants to continue	① the name and instruction of the game ② the card drawn each round ③ the points ④ total points ⑤ final score

Description:

- Input:

The user is prompted to enter whether they want to play another game, and their response is stored in the moreGames variable.

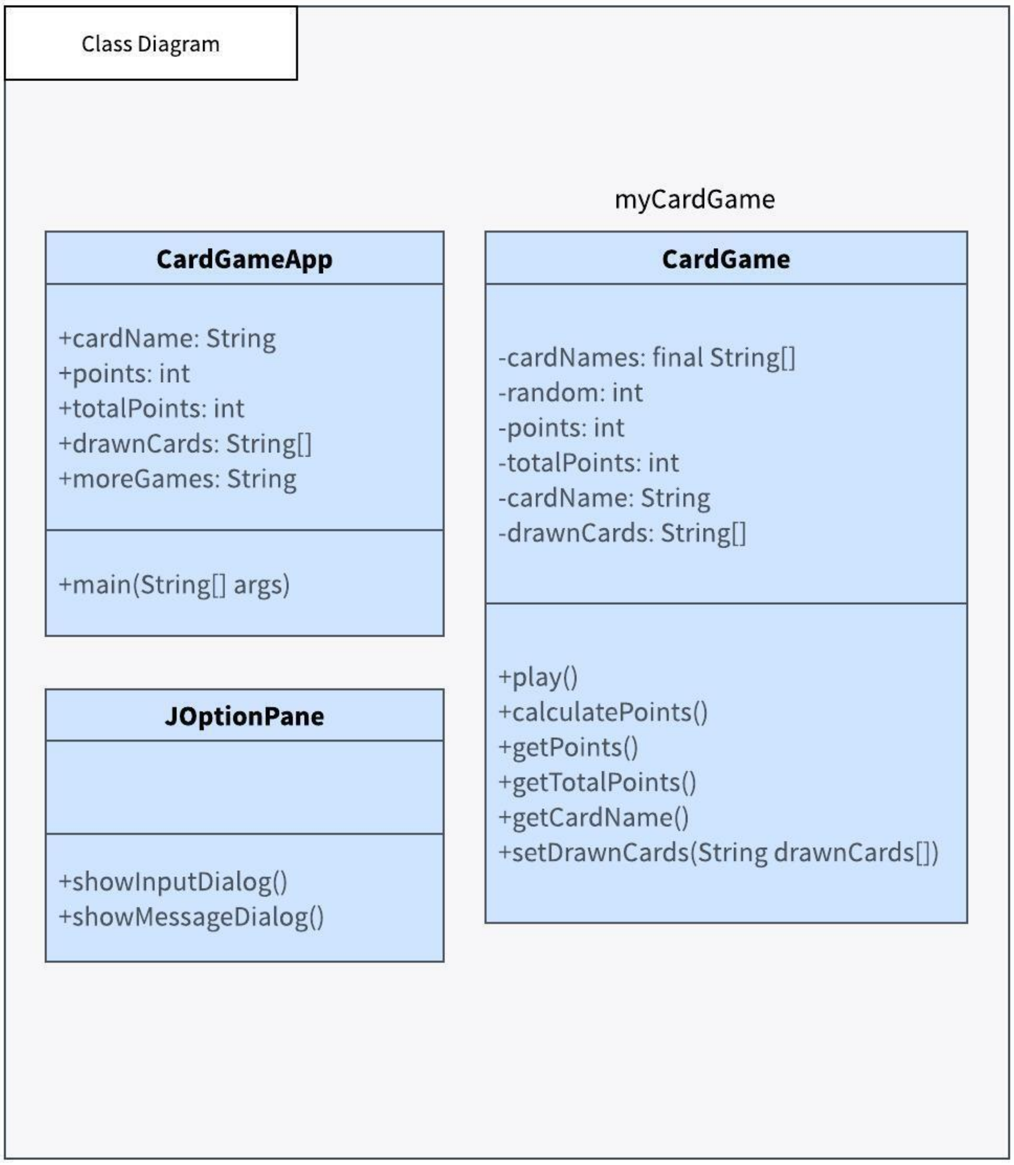
- Processing:

The processing mainly happens in the CardGame class. The play method generates a random card, and the calculatePoints method determines the points earned based on the three cards drawn, and the total point the user got. And there is a loop in the CardGameApp class that allows the game to continue when the user types yes.

- Output:

The program displays the game name and the instruction of the game, the card drawn in each round, the points earned in the game and the total points earned across all games played. Finally, the program displays the user's final score after they have finished playing.

2. Class diagram



3. Decisions I take in designing and implementing the application

- The application uses meaningful variable and method names to enhance code readability and maintainability.
- The application uses an object-oriented design with two classes: CardGame and CardGameApp. The CardGame class represents a single game of drawing cards, while the CardGameApp class manages multiple games and the overall flow of the application.
- The application uses arrays to represent the drawn cards and their names. This decision was made to simplify the implementation of the game logic.
- Implement validation to ensure that the application works as expected.

4. Source code explanation for the key functionalities and the approach taken to implement those functionalities

- Playing the game:

The key functionality of the Card Game app is playing the game itself. The approach taken to implement this functionality involves generating a random card from a deck of 13 cards in each round, and storing the name of the card that was drawn. This is accomplished in the play() method of the CardGame class, which generates a random integer between 0 and 12, and uses it to select the corresponding card name from an array of 13 card names. This card name is then stored in the cardName field of the CardGame object.

- Calculating points:

The points earned by the player in each round of the game are calculated based on the card that was drawn. The approach taken to implement this functionality involves checking the name of each card that was drawn, and assigning points according to a specific set of rules. This is accomplished in the calculatePoints() method of the CardGame class, which checks the names of the cards that were drawn and assigns points.

- Displaying game information:

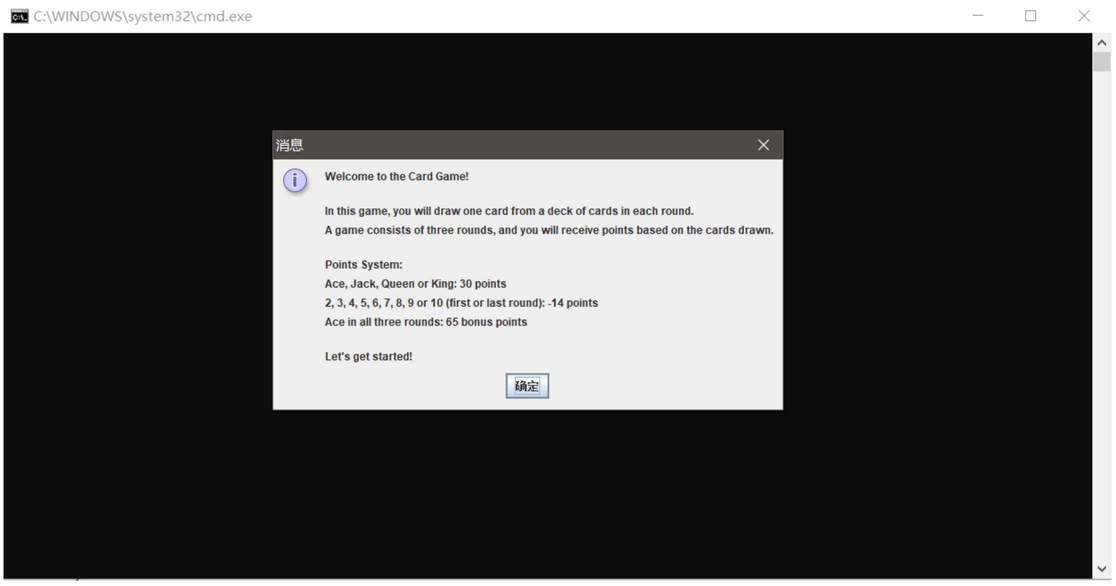
The app displays game information to the player using the JOptionPane class. This includes displaying the game name and instructions at the beginning of the game, displaying the card that was drawn in each round, displaying the points earned in each game, displaying the total points earned and asking the player if they want to play again. The approach taken to implement this functionality involves using showInputDialog and showMessageDialog methods provided by the JOptionPane class to display messages and receive input from the user

- Storing and retrieving game data:

The Card Game app stores and retrieves data related to each game, such as the card names that were drawn and the points earned in each game. The approach taken to implement this functionality involves storing this data in instance variables of the CardGame object, and using getter and setter methods to access the data.

5. Screenshots of the application's screens/output

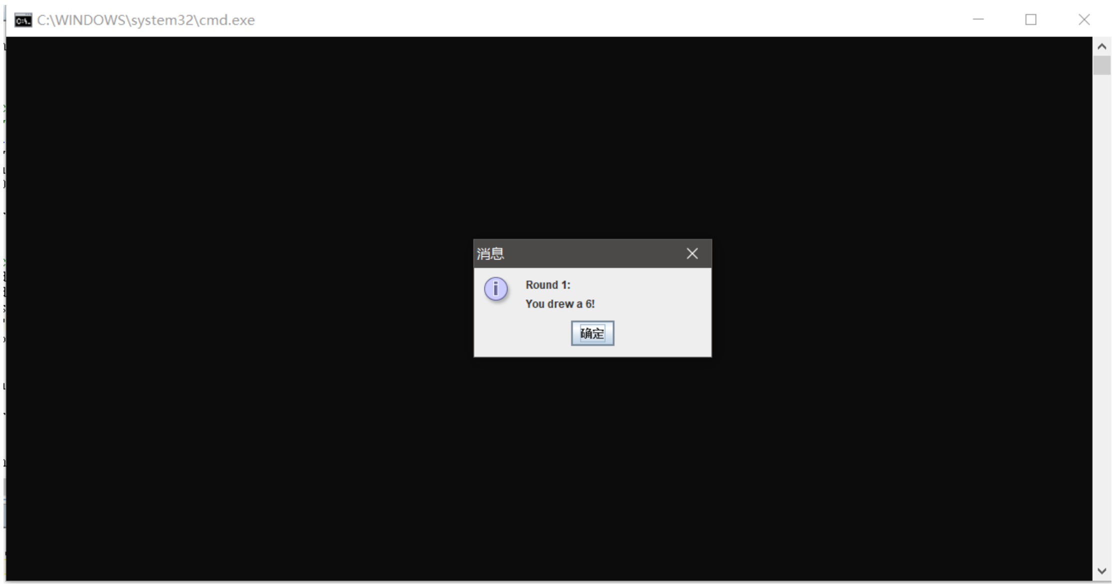
- The start of the game and the instructions:



```
Welcome to the Card Game!

In this game, you will draw one card from a deck of cards in each round.
A game consists of three rounds, and you will receive points based on the cards drawn.

Points System:
Ace, Jack, Queen or King: 30 points
2, 3, 4, 5, 6, 7, 8, 9 or 10 (first or last round): -14 points
Ace in all three rounds: 65 bonus points

Let's get started!
```

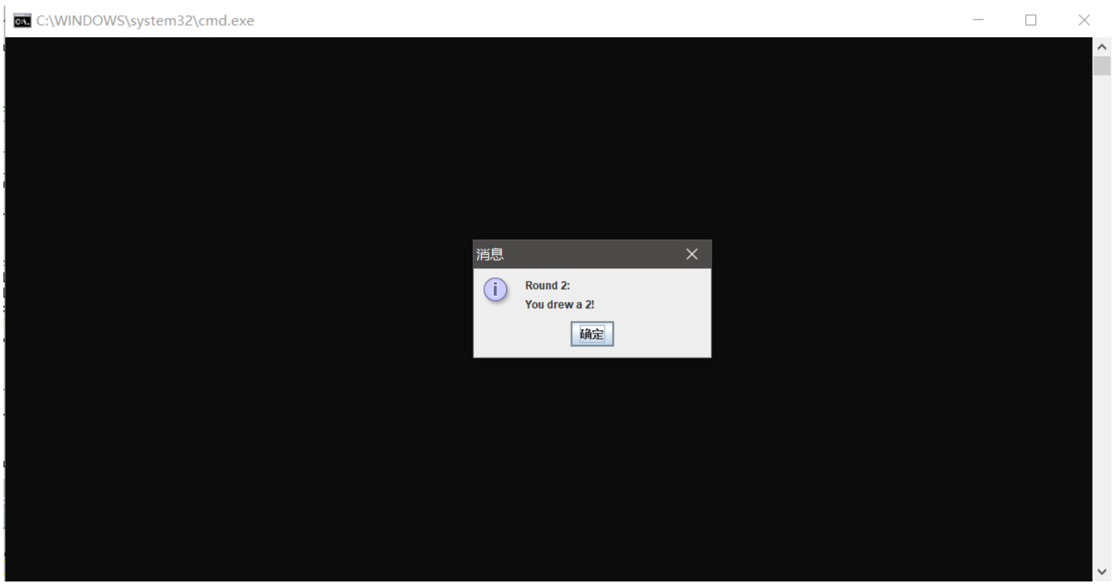
- The output of the first game:

First round:



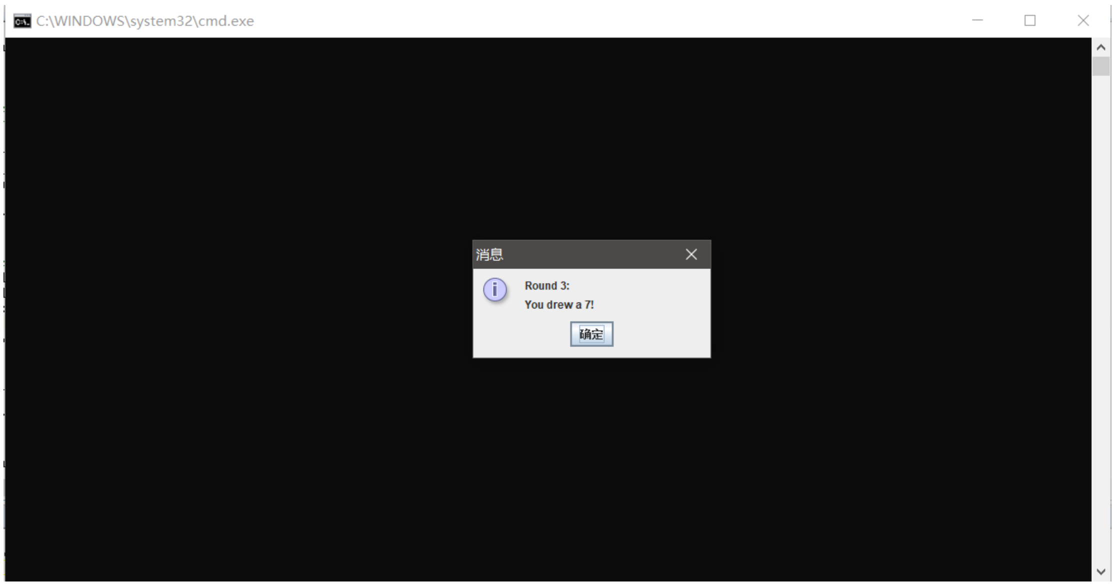
```
Round 1:
You drew a 6!
```

Second round:



```
Round 2:
You drew a 2!
```

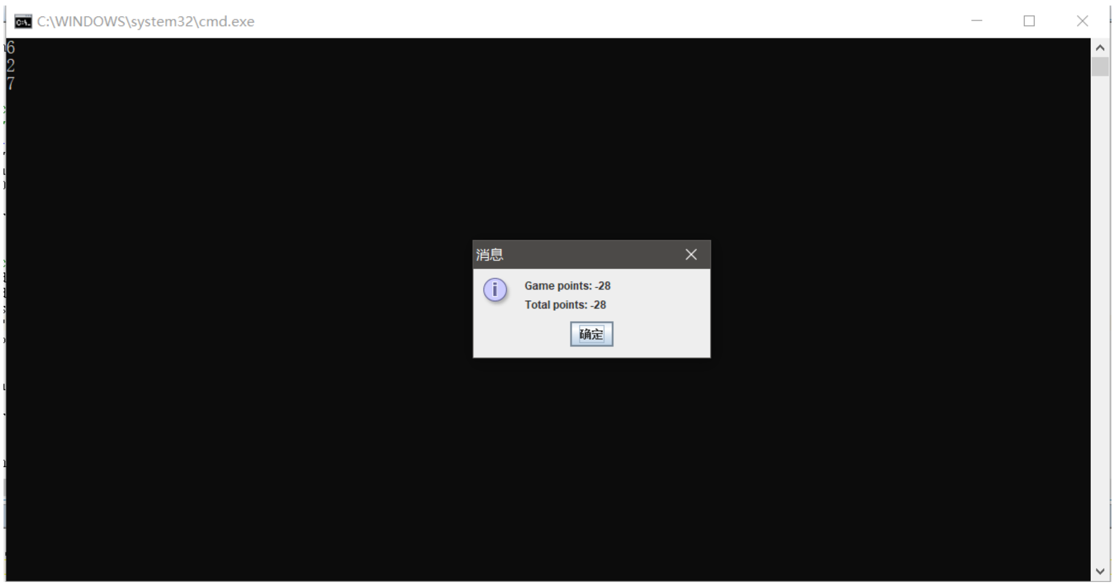
Third round:


C:\WINDOWS\system32\cmd.exe

消息

```
Round 3:
You drew a 7!
```

The result of the game:


C:\WINDOWS\system32\cmd.exe

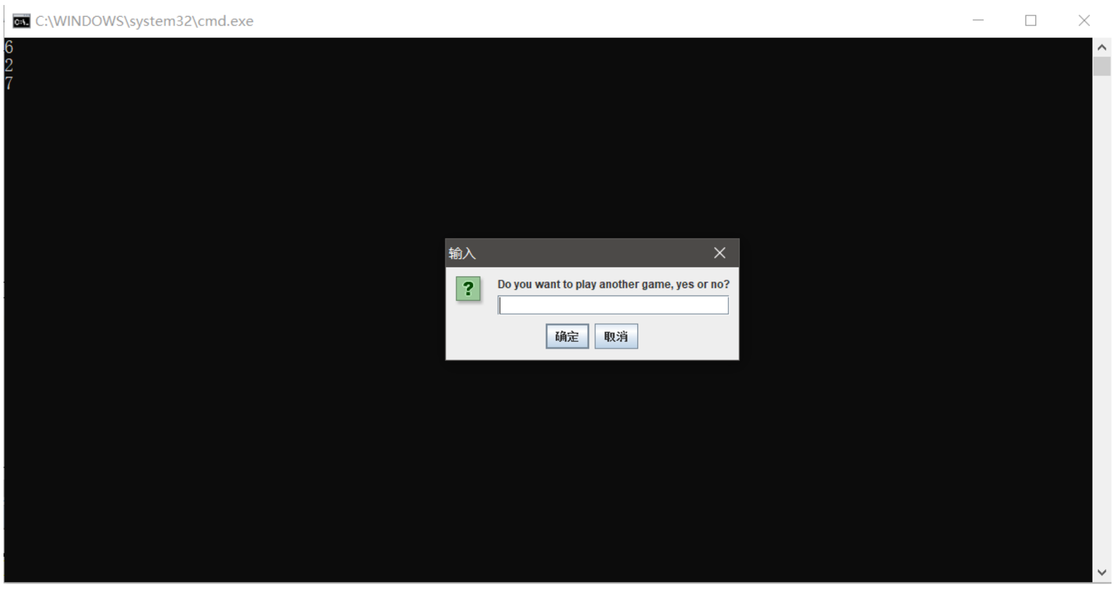
```
6
2
7
```

消息

```
Game points: -28
Total points: -28
```

The show up of the numbers at the left top corner is just for testing and recording, it is not a part of the design of the game.

- Ask the user if another game wanted:



```
C:\WINDOWS\system32\cmd.exe
6
2
7
```

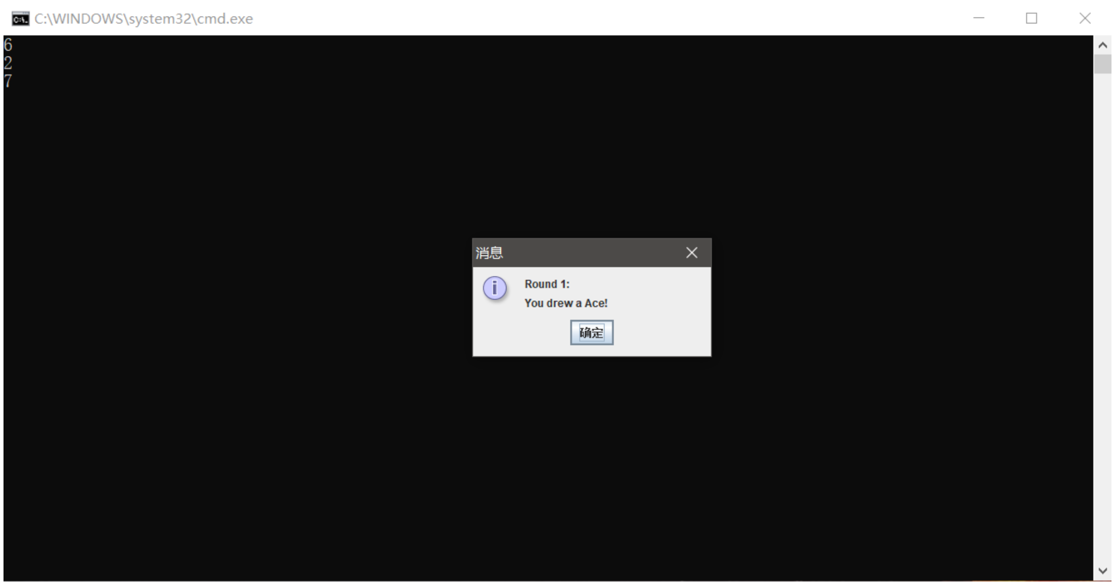
输入

Do you want to play another game, yes or no?

Type “yes”, the second game starts.

- The output of the second game:

First round:

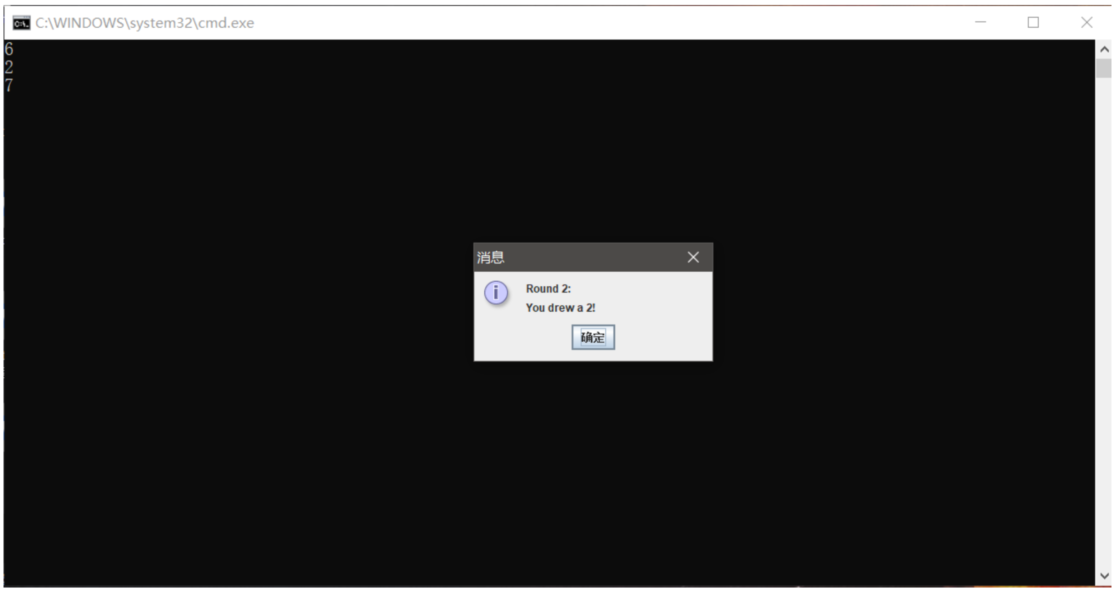


```
C:\WINDOWS\system32\cmd.exe
6
2
7
```

消息

Round 1:
You drew a Ace!

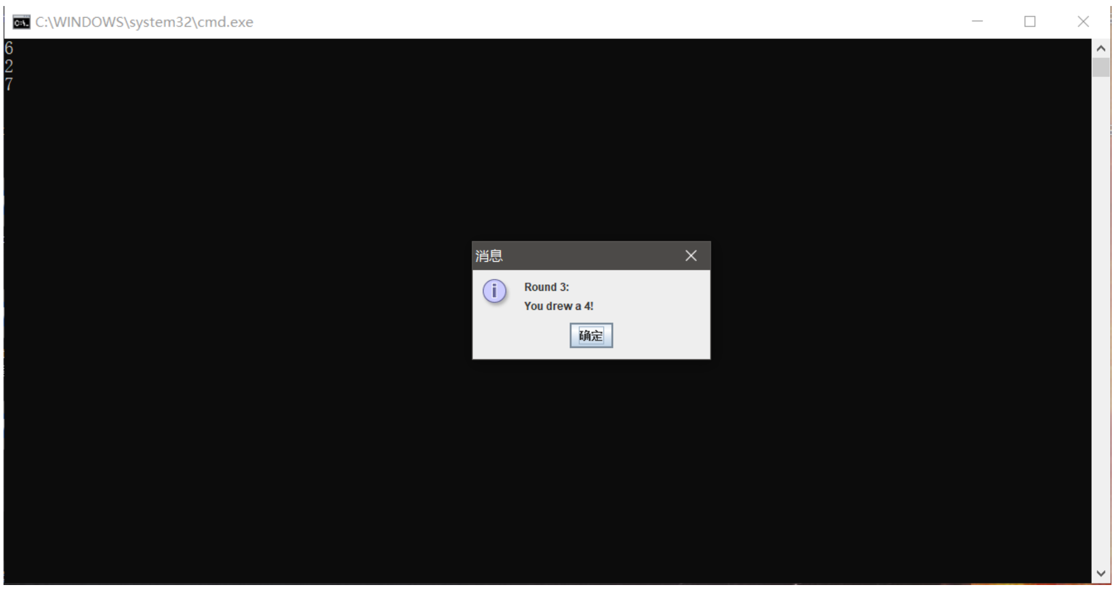
Second round:



```
6
2
7
```

Round 2:
You drew a 2!

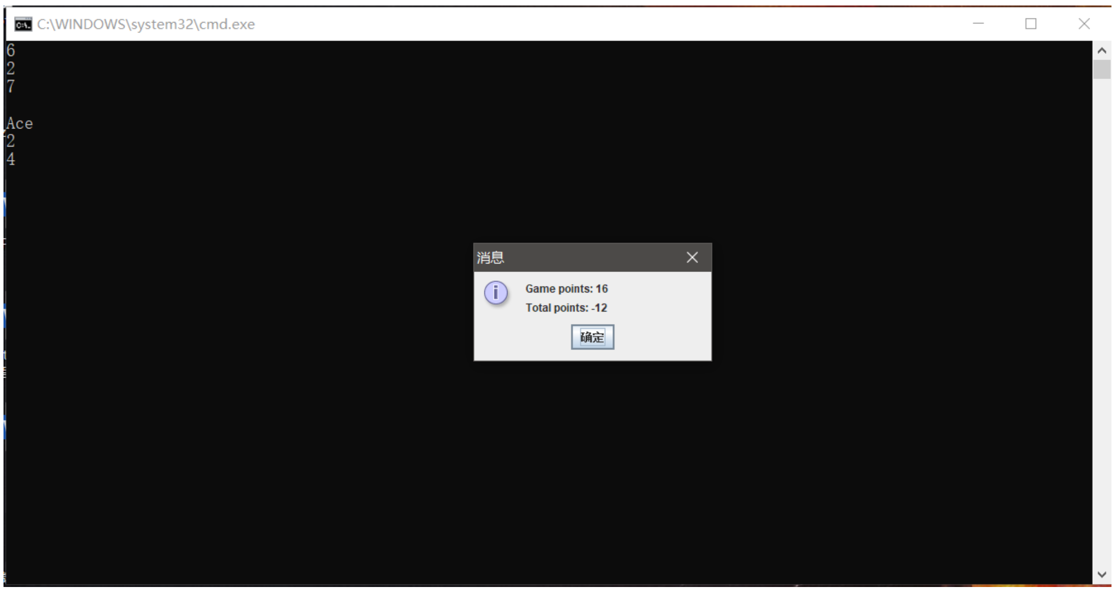
Third round:



```
6
2
7
```

Round 3:
You drew a 4!

The result:

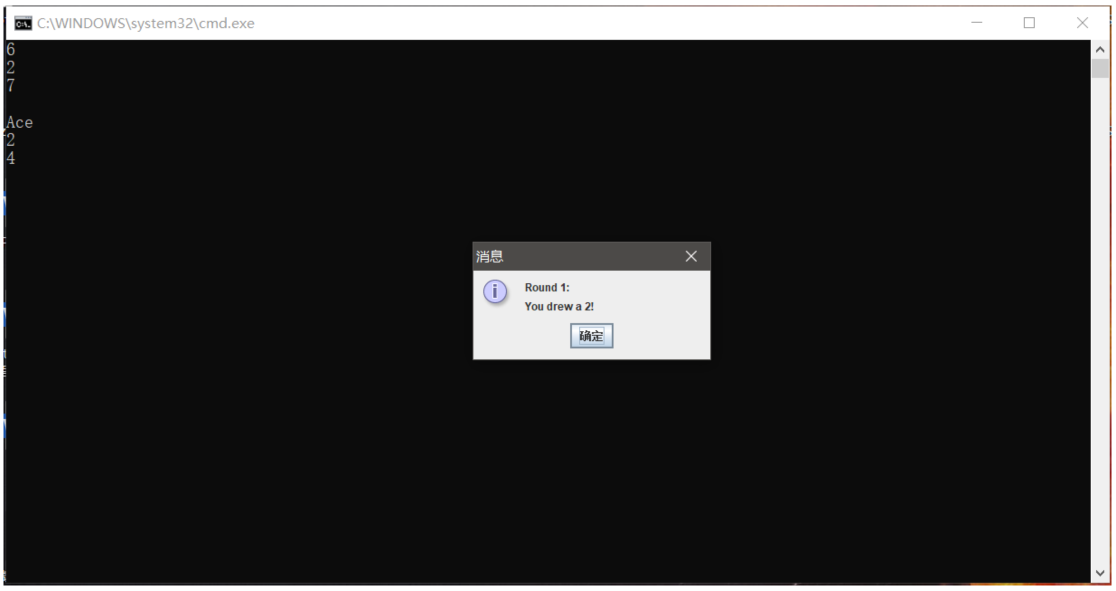


```
C:\WINDOWS\system32\cmd.exe
6
2
7

Ace
2
4
```

```
消息
Game points: 16
Total points: -12
确定
```

● The output of the third game:

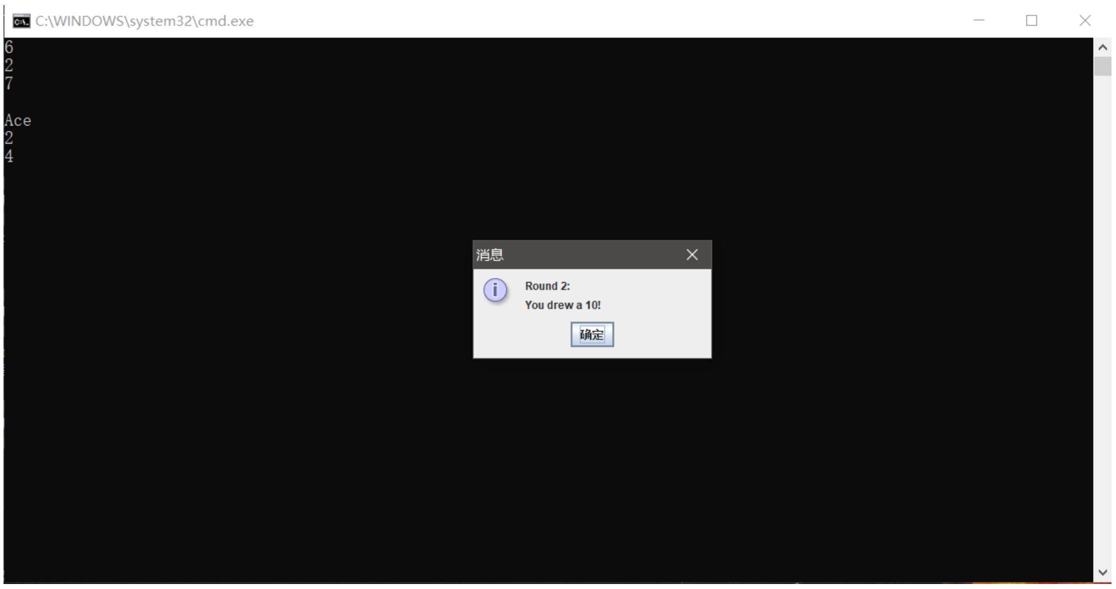
First round:



```
C:\WINDOWS\system32\cmd.exe
6
2
7

Ace
2
4
```

```
消息
Round 1:
You drew a 2!
确定
```

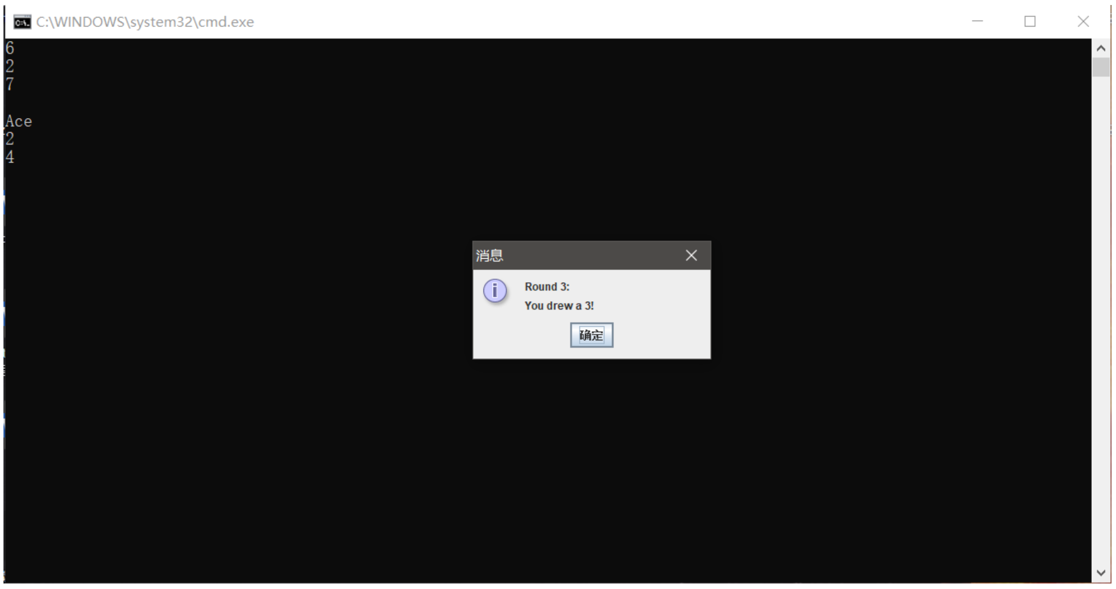
Second round:



```
C:\WINDOWS\system32\cmd.exe
6
2
7

Ace
2
4
```

```
消息
Round 2:
You drew a 10!
确定
```

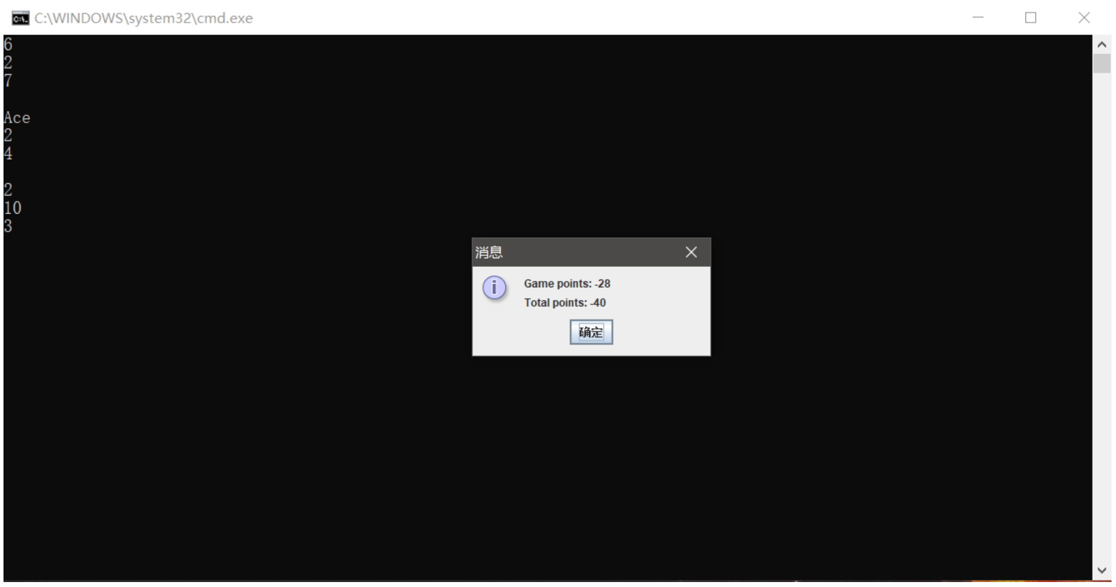
Third round:



```
C:\WINDOWS\system32\cmd.exe
6
2
7

Ace
2
4
```

```
消息
Round 3:
You drew a 3!
确定
```

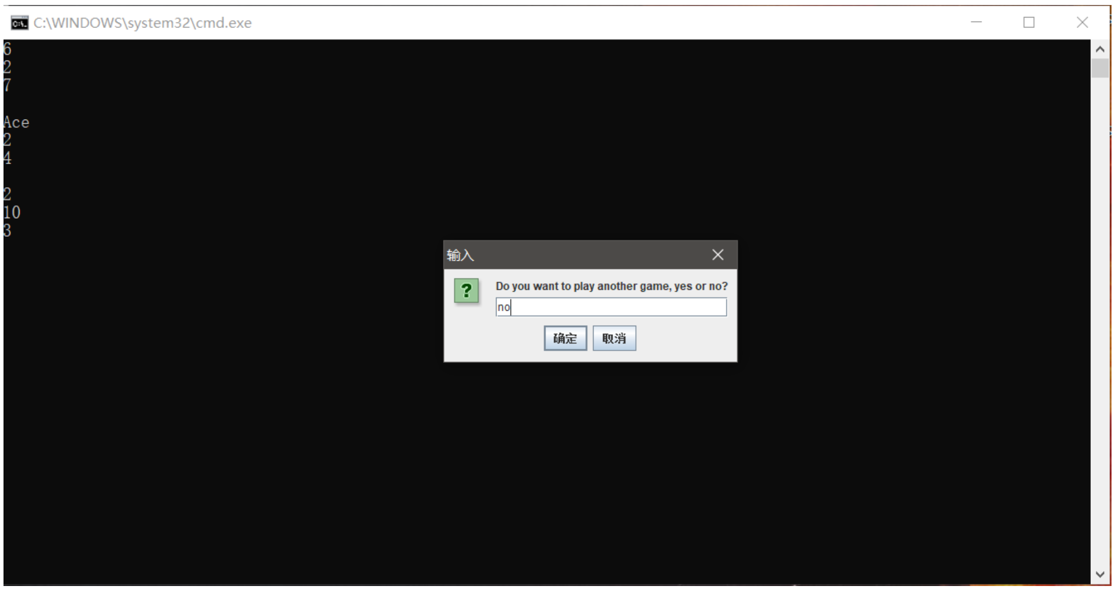
The result:



```
C:\WINDOWS\system32\cmd.exe
6
2
7

Ace
2
4

2
10
3
```

```
消息
Game points: -28
Total points: -40
```

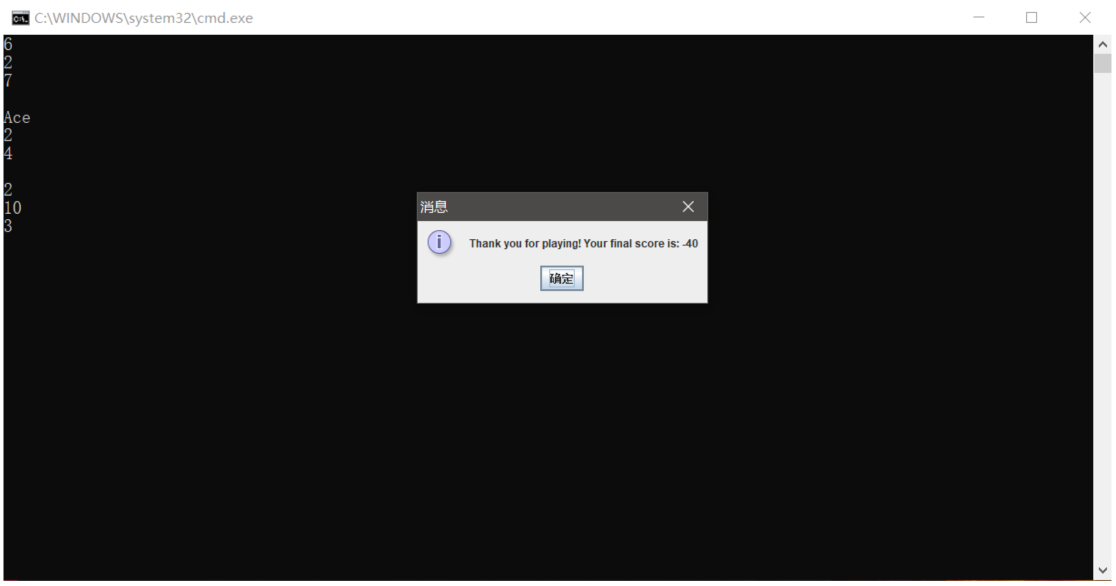
- The game ends if the input is not “yes”:



```
C:\WINDOWS\system32\cmd.exe
6
2
7

Ace
2
4

2
10
3
```

```
输入
Do you want to play another game, yes or no?
no
```

- The output at the end of all games:


C:\WINDOWS\system32\cmd.exe

```
6
2
7

Ace
2
4

2
10
3
```

消息

Thank you for playing! Your final score is: -40

Appendix:

- CardGameApp.java

```
/*
*CardGameApp.java
*@author Shan Liang
*03/2023
*/

import javax.swing.JOptionPane;

public class CardGameApp {
	public static void main(String[] args) {

		//declare variables
		String cardName;
		int points = 0, totalPoints = 0;
		String[] drawnCards = new String[3];
		String moreGames = "no";


		//display game name and instructions
		JOptionPane.showMessageDialog(null, "Welcome to the Card Game!\n\n"
								+ "In this game, you will draw one card from
a deck of cards in each round.\n"
								+ "A game consists of three rounds, and you
will receive points based on the cards drawn.\n\n"
								+ "Points System:\n"
								+ "Ace, Jack, Queen or King: 30 points\n"
								+ "2, 3, 4, 5, 6, 7, 8, 9 or 10 (first or last
round): -14 points\n"
                                                            + "Ace in all three rounds: 65 bonus
points\n\n"
                                                            + "Let's get started!");



          //declare and create new objects
          CardGame myCardGame = new CardGame();



          do{
               //process and output
               //play a 3-round game and display the result every round
               for (int i = 0; i < 3; i++){
                    myCardGame.play();
                    cardName = myCardGame.getCardName();
                    JOptionPane.showMessageDialog(null, "Round " + (i + 1) + ":\n"
                                                                      + "You drew a " + cardName +
"!");
                    drawnCards[i] = cardName;
               }

               //calculate the points
               myCardGame.setDrawnCards(drawnCards);
               myCardGame.calculatePoints();
               points = myCardGame.getPoints();
               totalPoints = myCardGame.getTotalPoints();
               JOptionPane.showMessageDialog(null, "Game points: " + points
                                                                 + "\nTotal points: " + totalPoints);
                    moreGames = JOptionPane.showInputDialog(null, "Do you want to play another
game, yes or no?");

            }while(moreGames.equalsIgnoreCase("yes"));


            JOptionPane.showMessageDialog(null, "Thank you for playing! Your final score is: " +
totalPoints);


        }//main
}//class
```

- CardGame.java

```
/*
*CardGame.java
*@author Shan Liang
*03/2023
*/

public class CardGame {

        //declare data members
        private final String[] cardNames = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack",
"Queen", "King"};
        private int random, points, totalPoints;
        private String cardName;
        private String[] drawnCards = new String[3];
```

```
	//constructor
	public CardGame(){
		points = 0;
		totalPoints = 0;
		random = 0;
		cardName = " ";
	}

	//compute methods
	//generate an integer randomly from 0 to 12, and then get a random card name
	public void play() {
		random = (int) (Math.random() * (cardNames.length - 1));
		cardName = cardNames[random];

	}

	//calculate the points of 1 game and the total points
	public void calculatePoints(){

		//just for the record, uncomment to make it easier to review
		//System.out.println(drawnCards[0]);
		//System.out.println(drawnCards[1]);
		//System.out.println(drawnCards[2]);
		//System.out.println(" ");

		points = 0;
		if (drawnCards[0] == "Ace" || drawnCards[0] == "Jack" || drawnCards[0] == "Queen" ||
drawnCards[0] == "King"){
			points = points + 30;
		}
```

```
        else {
            points = points - 14;
        }
        if (drawnCards[1] == "Ace" || drawnCards[1] == "Jack" || drawnCards[1] == "Queen" ||
drawnCards[1] == "King"){
            points = points + 30;
        }
        if (drawnCards[2] == "Ace" || drawnCards[2] == "Jack" || drawnCards[2] == "Queen" ||
drawnCards[2] == "King"){
            points = points + 30;
        }
        else {
            points = points - 14;
        }
        if (drawnCards[0] == "Ace" && drawnCards[1] == "Ace" && drawnCards[2] ==
"Ace"){
            points = points + 65;
        }
        totalPoints = totalPoints + points;
    }

    //setters and getters

    public int getPoints() {
        return points;
    }

    public int getTotalPoints() {
        return totalPoints;
    }
```

```
        public String getCardName(){
                return cardName;
        }

        public void setDrawnCards(String drawnCards[]){
                this.drawnCards = drawnCards;
        }
}
```