Intro to Java Week 6 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

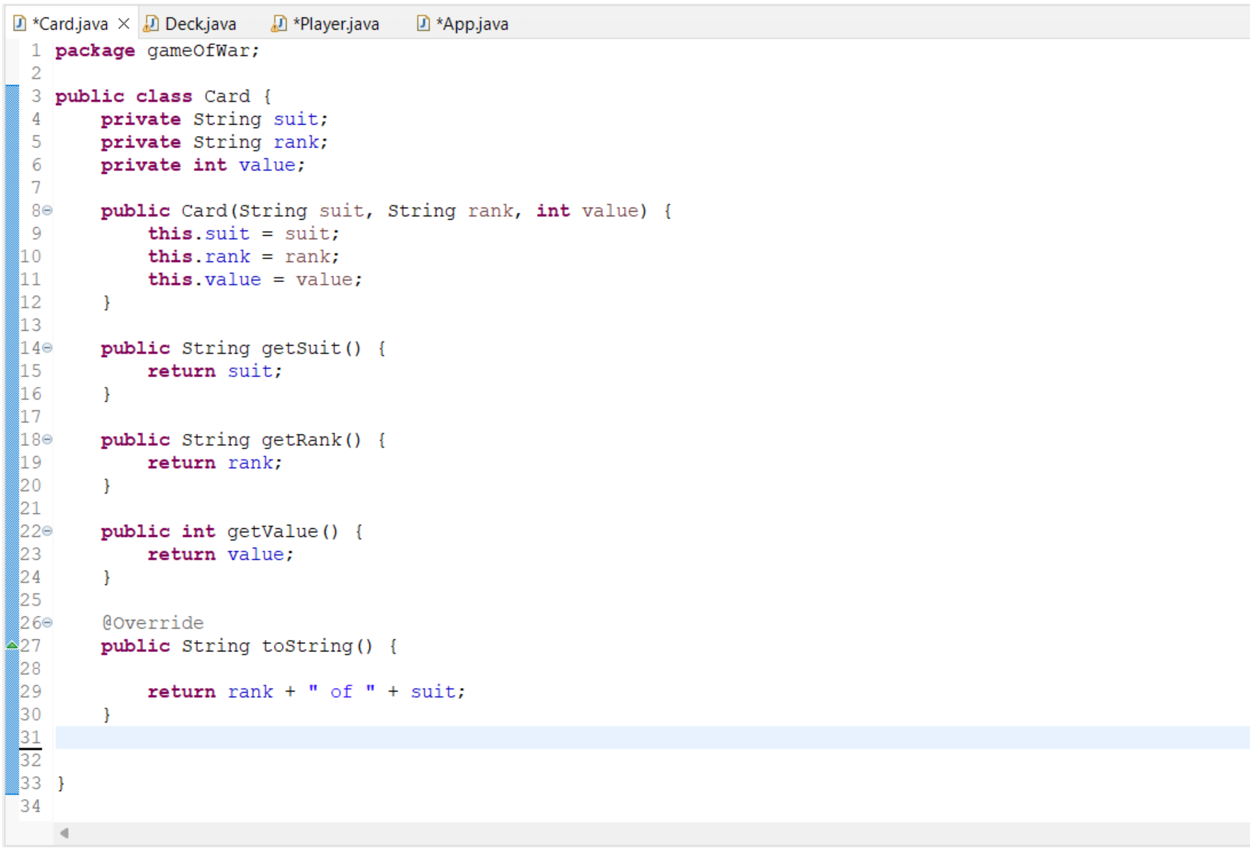
For the final project you will be creating an automated version of the classic card game *WAR.*

1. Create the following classes.
    a. Card
        i. Fields
            1. **value** (contains a value from 2-14 representing cards 2-Ace)
            2. **name** (e.g. Ace of Diamonds, or Two of Hearts)
        ii. Methods
            1. Getters and Setters
            2. **describe** (prints out information about a card)
    b. Deck
        i. Fields
            1. **cards** (List of Card)
        ii. Methods
            1. **shuffle** (randomizes the order of the cards)
            2. **draw** (removes and returns the top card of the Cards field)

            3. In the constructor, when a new Deck is instantiated, the Cards field should be populated with the standard 52 cards.
    c. Player
        i. Fields
            1. **hand** (List of Card)
            2. **score** (set to 0 in the constructor)
            3. **name**
        ii. Methods
            1. **describe** (prints out information about the player and calls the describe method for each card in the Hand List)
            2. **flip** (removes and returns the top card of the Hand)
            3. **draw** (takes a Deck as an argument and calls the draw method on the deck, adding the returned Card to the hand field)
            4. **incrementScore** (adds 1 to the Player's score field)

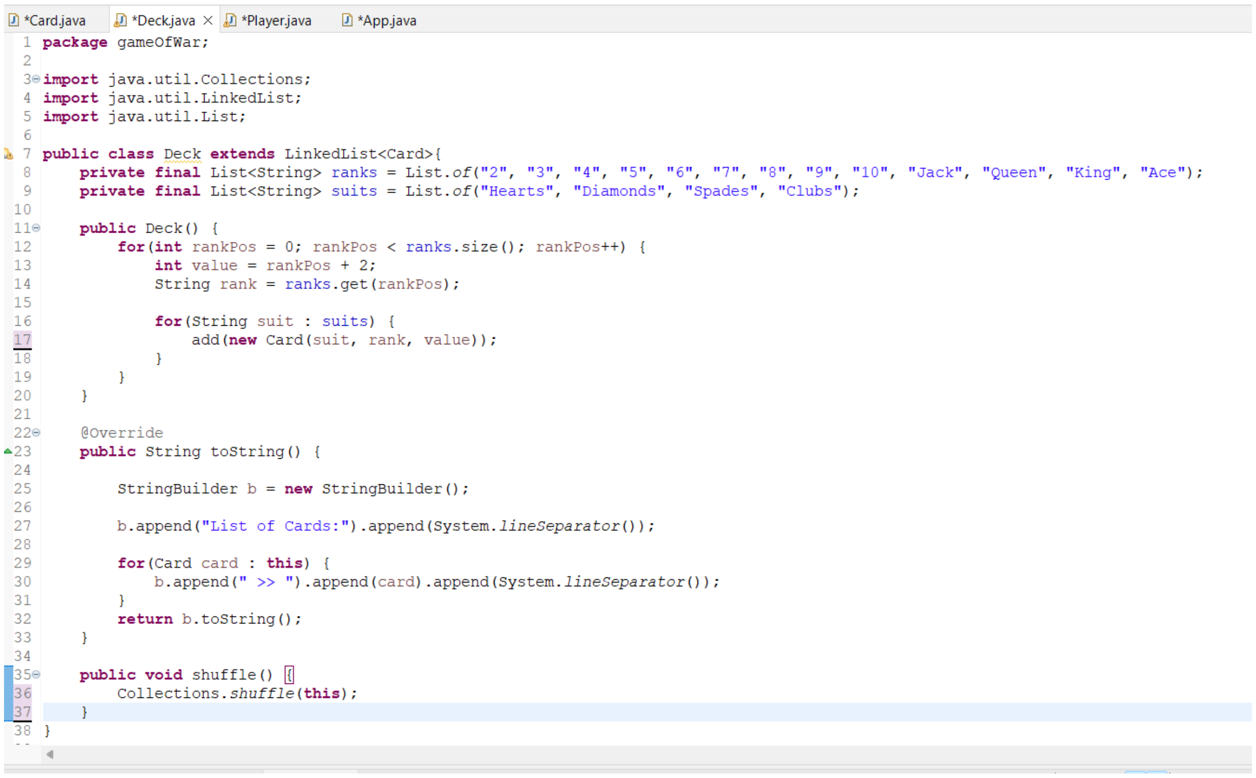
2. Create a class called App with a main method.
3. Instantiate a Deck and two Players, call the shuffle method on the deck.
4. Using a traditional for loop, iterate 52 times calling the Draw method on the other player each iteration using the Deck you instantiated.
5. Using a traditional for loop, iterate 26 times and call the flip method for each player.
    a. Compare the value of each card returned by the two player's flip methods. Call the incrementScore method on the player whose card has the higher value.
6. After the loop, compare the final score from each player.
7. Print the final score of each player and either "Player 1", "Player 2", or "Draw" depending on which score is higher or if they are both the same.
8. **Screenshots of Code:**

Card.java -



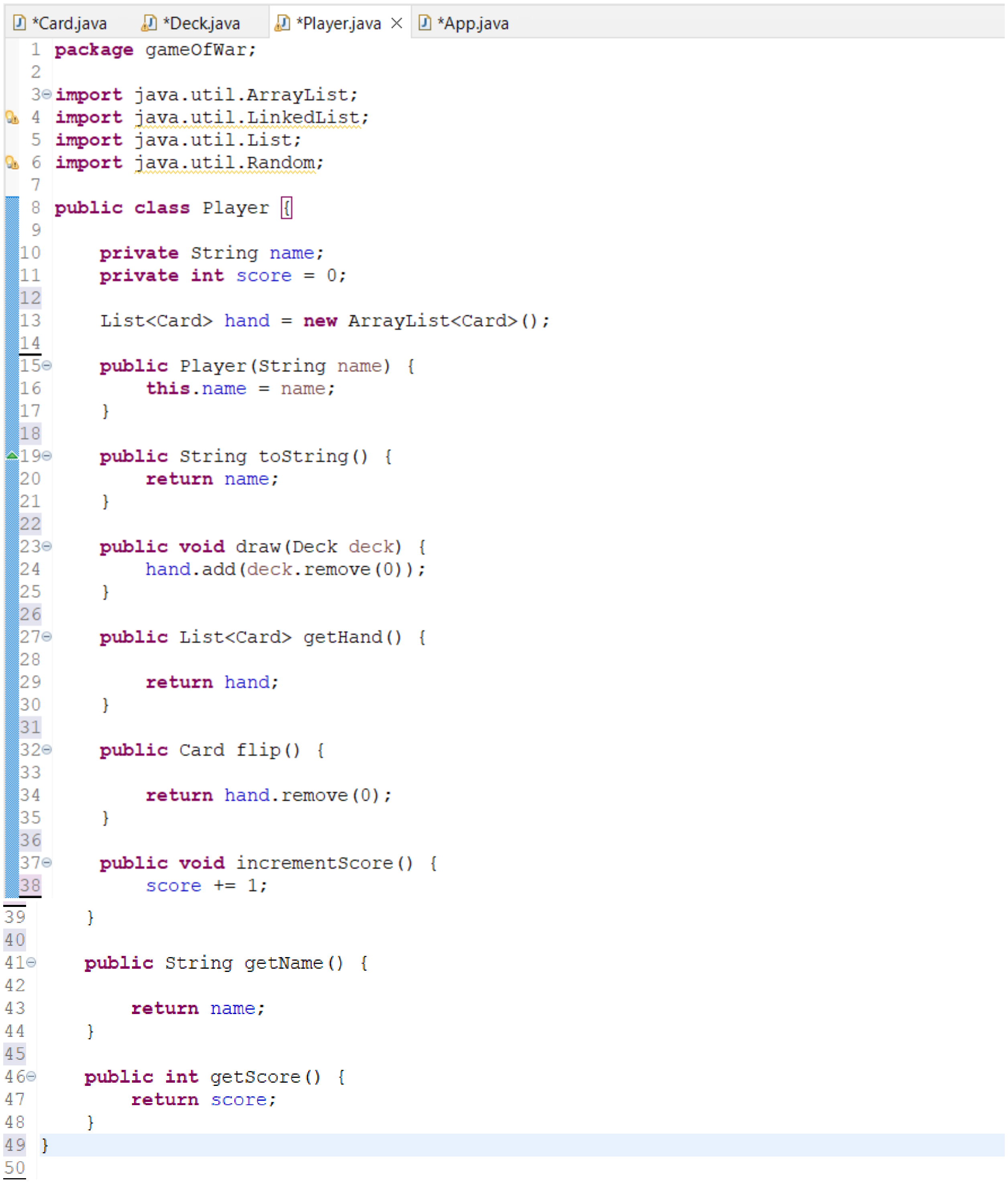
```java
package gameOfWar;

public class Card {
    private String suit;
    private String rank;
    private int value;

    public Card(String suit, String rank, int value) {
        this.suit = suit;
        this.rank = rank;
        this.value = value;
    }

    public String getSuit() {
        return suit;
    }

    public String getRank() {
        return rank;
    }

    public int getValue() {
        return value;
    }

    @Override
    public String toString() {

        return rank + " of " + suit;
    }


}
```

Deck.java -



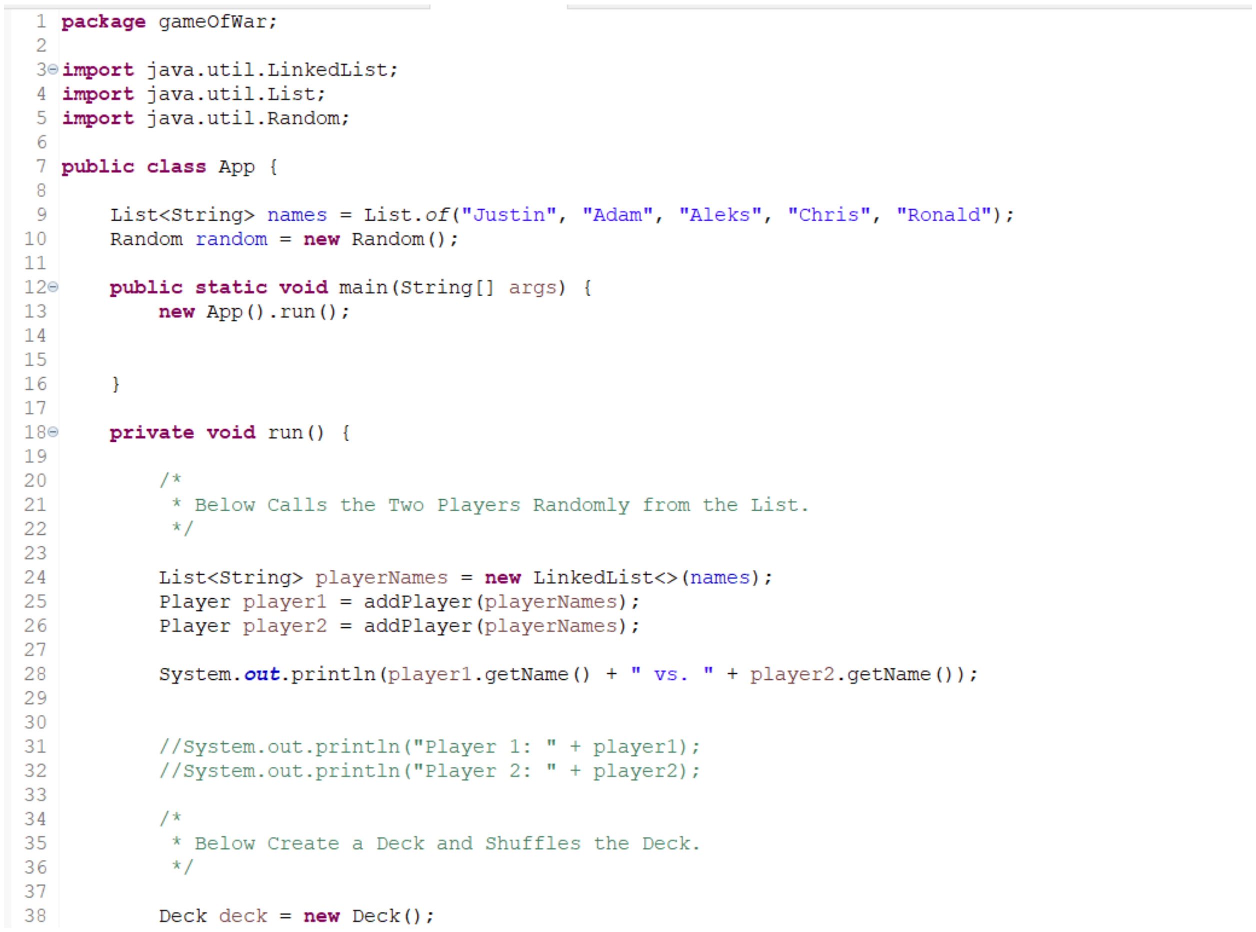
```java
package gameOfWar;

import java.util.Collections;
import java.util.LinkedList;
import java.util.List;

public class Deck extends LinkedList<Card>{
    private final List<String> ranks = List.of("2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace");
    private final List<String> suits = List.of("Hearts", "Diamonds", "Spades", "Clubs");

    public Deck() {
        for(int rankPos = 0; rankPos < ranks.size(); rankPos++) {
            int value = rankPos + 2;
            String rank = ranks.get(rankPos);

            for(String suit : suits) {
                add(new Card(suit, rank, value));
            }
        }
    }

    @Override
    public String toString() {

        StringBuilder b = new StringBuilder();

        b.append("List of Cards:").append(System.lineSeparator());

        for(Card card : this) {
            b.append(" >> ").append(card).append(System.lineSeparator());
        }
        return b.toString();
    }

    public void shuffle() {
        Collections.shuffle(this);
    }
}
```

Player.java –

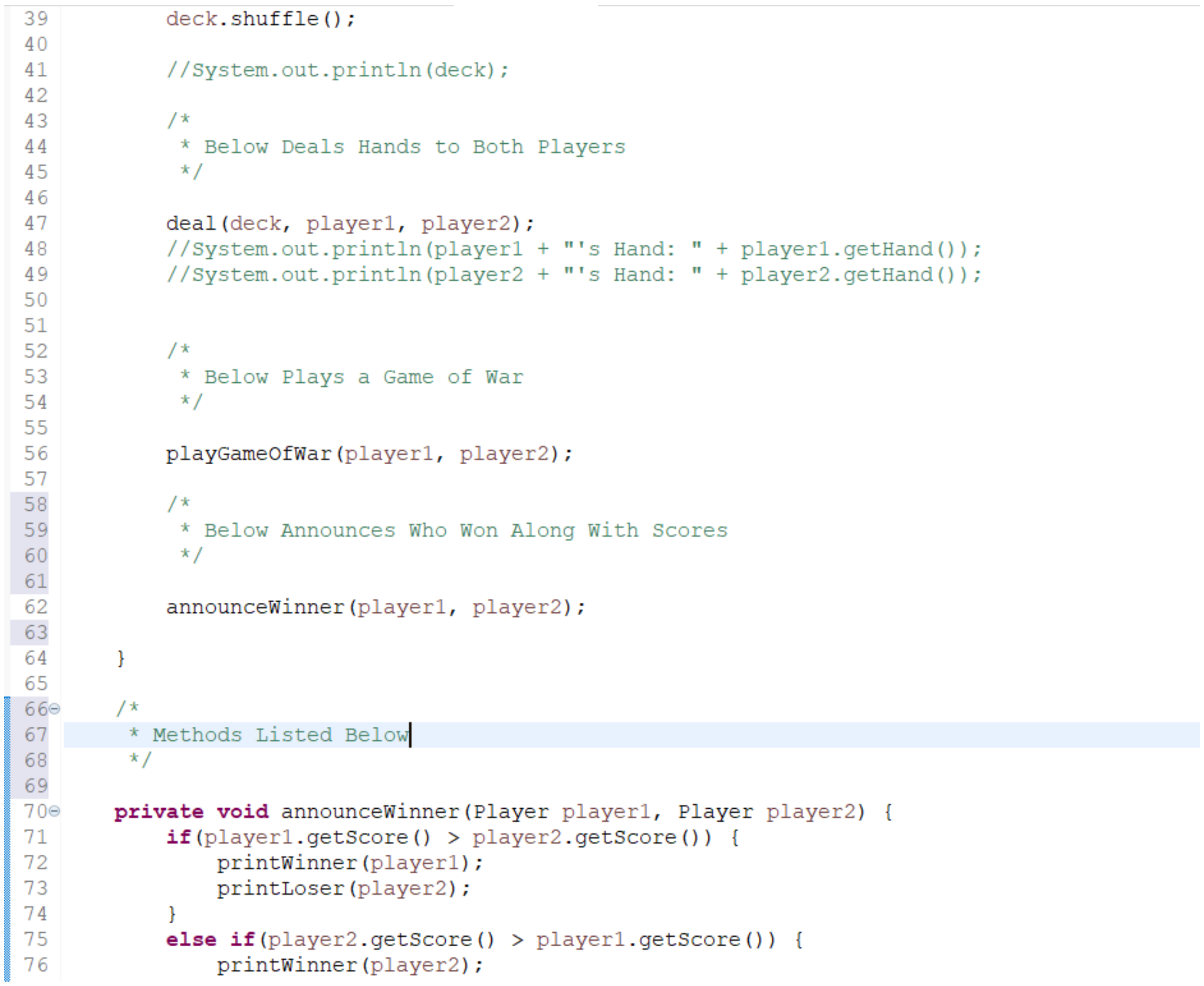

*Card.java *Deck.java *Player.java *App.java

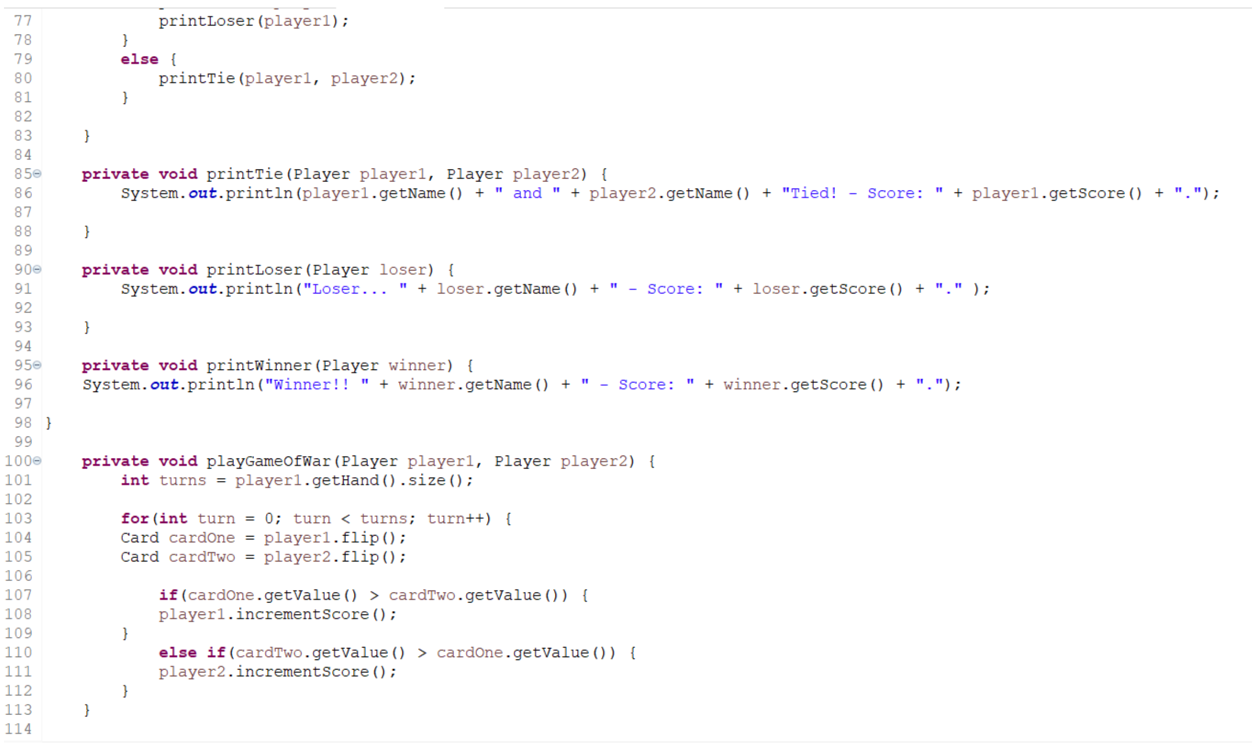
```java
package gameOfWar;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.Random;

public class Player {

    private String name;
    private int score = 0;

    List<Card> hand = new ArrayList<Card>();

    public Player(String name) {
        this.name = name;
    }

    public String toString() {
        return name;
    }

    public void draw(Deck deck) {
        hand.add(deck.remove(0));
    }

    public List<Card> getHand() {

        return hand;
    }

    public Card flip() {

        return hand.remove(0);
    }

    public void incrementScore() {
        score += 1;
    }

    public String getName() {

        return name;
    }

    public int getScore() {
        return score;
    }
}
```

App.java –

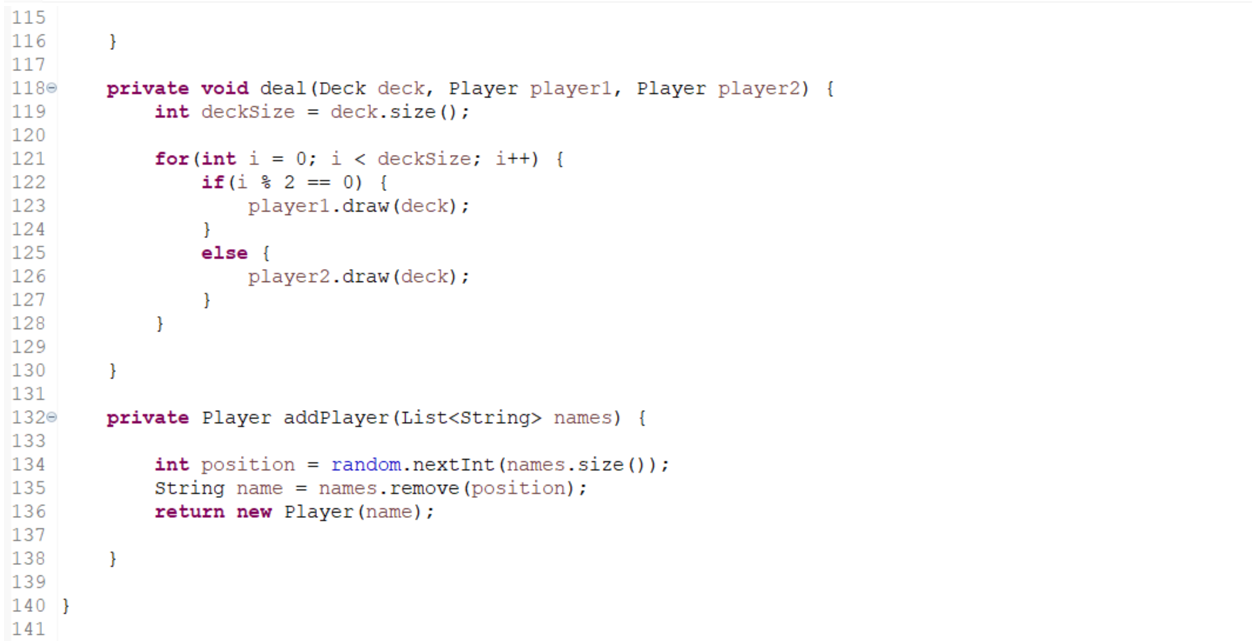


```java
package gameOfWar;

import java.util.LinkedList;
import java.util.List;
import java.util.Random;

public class App {

    List<String> names = List.of("Justin", "Adam", "Aleks", "Chris", "Ronald");
    Random random = new Random();

    public static void main(String[] args) {
        new App().run();


    }

    private void run() {

        /*
         * Below Calls the Two Players Randomly from the List.
         */

        List<String> playerNames = new LinkedList<>(names);
        Player player1 = addPlayer(playerNames);
        Player player2 = addPlayer(playerNames);

        System.out.println(player1.getName() + " vs. " + player2.getName());


        //System.out.println("Player 1: " + player1);
        //System.out.println("Player 2: " + player2);

        /*
         * Below Create a Deck and Shuffles the Deck.
         */

        Deck deck = new Deck();
```


```java
        deck.shuffle();

        //System.out.println(deck);

        /*
         * Below Deals Hands to Both Players
         */

        deal(deck, player1, player2);
        //System.out.println(player1 + "'s Hand: " + player1.getHand());
        //System.out.println(player2 + "'s Hand: " + player2.getHand());


        /*
         * Below Plays a Game of War
         */

        playGameOfWar(player1, player2);

        /*
         * Below Announces Who Won Along With Scores
         */

        announceWinner(player1, player2);

    }

    /*
     * Methods Listed Below
     */

    private void announceWinner(Player player1, Player player2) {
        if(player1.getScore() > player2.getScore()) {
            printWinner(player1);
            printLoser(player2);
        }
        else if(player2.getScore() > player1.getScore()) {
            printWinner(player2);
```



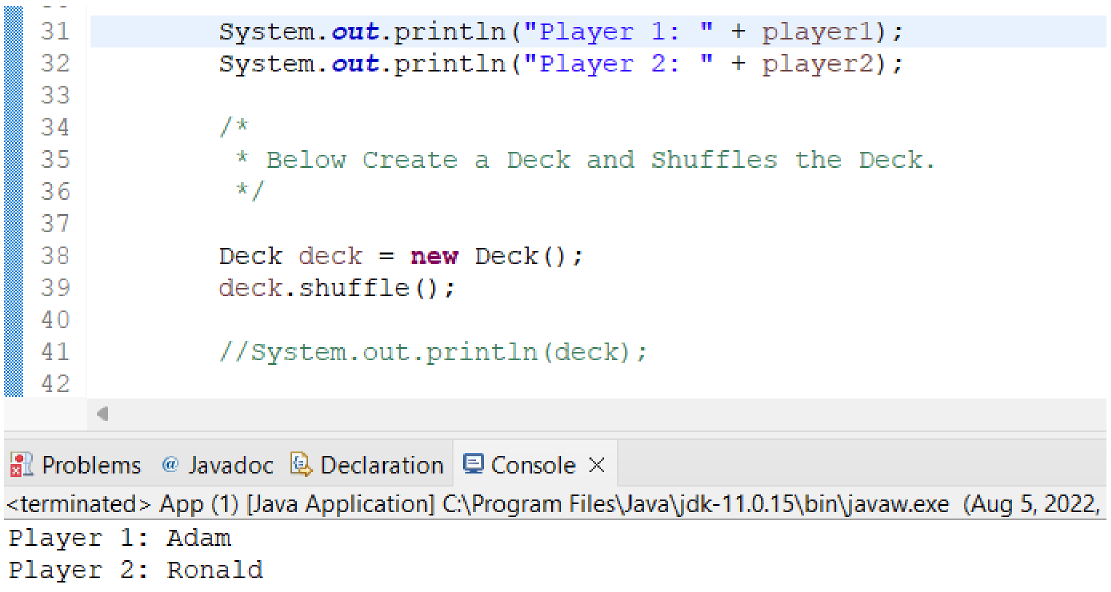
```java
            printLoser(player1);
        }
        else {
            printTie(player1, player2);
        }

    }

    private void printTie(Player player1, Player player2) {
        System.out.println(player1.getName() + " and " + player2.getName() + "Tied! - Score: " + player1.getScore() + ".");

    }

    private void printLoser(Player loser) {
        System.out.println("Loser... " + loser.getName() + " - Score: " + loser.getScore() + "." );

    }

    private void printWinner(Player winner) {
    System.out.println("Winner!! " + winner.getName() + " - Score: " + winner.getScore() + ".");

}

    private void playGameOfWar(Player player1, Player player2) {
        int turns = player1.getHand().size();

        for(int turn = 0; turn < turns; turn++) {
        Card cardOne = player1.flip();
        Card cardTwo = player2.flip();

            if(cardOne.getValue() > cardTwo.getValue()) {
            player1.incrementScore();
        }
            else if(cardTwo.getValue() > cardOne.getValue()) {
            player2.incrementScore();
        }
    }

```



```java

    }

    private void deal(Deck deck, Player player1, Player player2) {
        int deckSize = deck.size();

        for(int i = 0; i < deckSize; i++) {
            if(i % 2 == 0) {
                player1.draw(deck);
            }
            else {
                player2.draw(deck);
            }
        }

    }

    private Player addPlayer(List<String> names) {

        int position = random.nextInt(names.size());
        String name = names.remove(position);
        return new Player(name);

    }

}

```

Screenshots of Running Application:

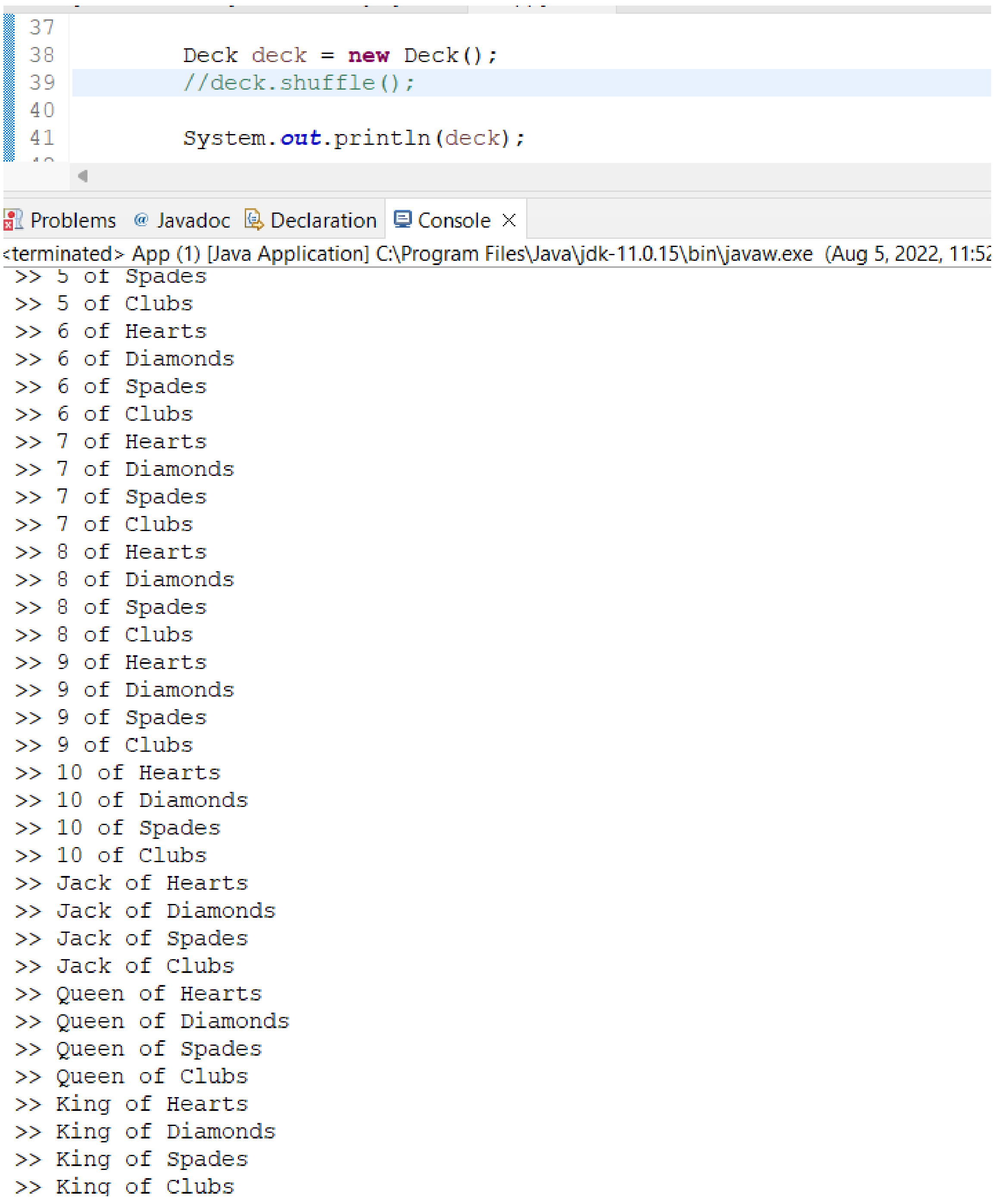
Listing Players with Names:



```java
        System.out.println("Player 1: " + player1);
        System.out.println("Player 2: " + player2);

        /*
         * Below Create a Deck and Shuffles the Deck.
         */

        Deck deck = new Deck();
        deck.shuffle();

        //System.out.println(deck);

```

```
<terminated> App (1) [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe  (Aug 5, 2022,
Player 1: Adam
Player 2: Ronald
```

Listing Ordered Cards in Deck:

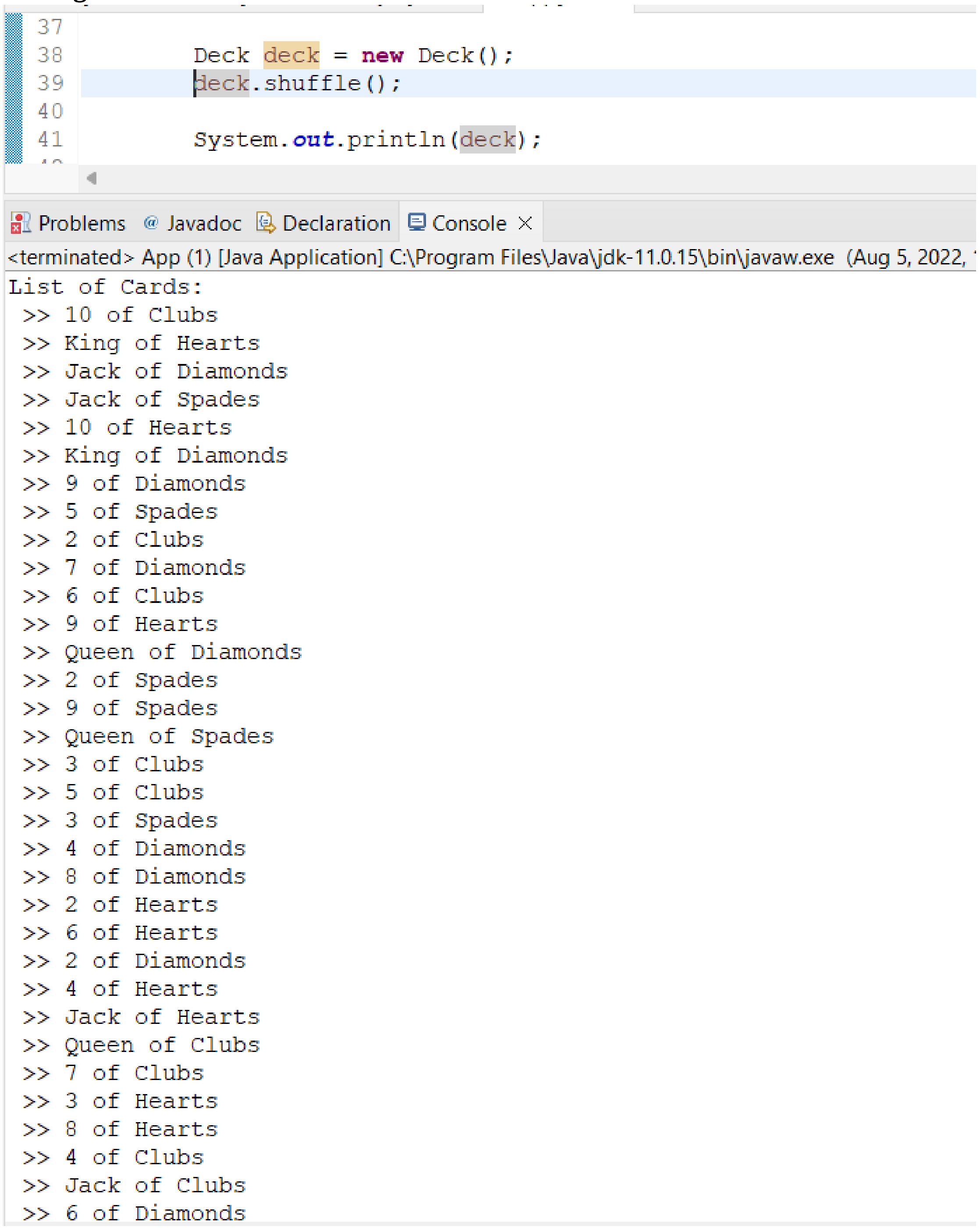


```
        Deck deck = new Deck();
        //deck.shuffle();

        System.out.println(deck);
```

Problems Javadoc Declaration Console

```
<terminated> App (1) [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe  (Aug 5, 2022, 11:52
>> 5 of Spades
>> 5 of Clubs
>> 6 of Hearts
>> 6 of Diamonds
>> 6 of Spades
>> 6 of Clubs
>> 7 of Hearts
>> 7 of Diamonds
>> 7 of Spades
>> 7 of Clubs
>> 8 of Hearts
>> 8 of Diamonds
>> 8 of Spades
>> 8 of Clubs
>> 9 of Hearts
>> 9 of Diamonds
>> 9 of Spades
>> 9 of Clubs
>> 10 of Hearts
>> 10 of Diamonds
>> 10 of Spades
>> 10 of Clubs
>> Jack of Hearts
>> Jack of Diamonds
>> Jack of Spades
>> Jack of Clubs
>> Queen of Hearts
>> Queen of Diamonds
>> Queen of Spades
>> Queen of Clubs
>> King of Hearts
>> King of Diamonds
>> King of Spades
>> King of Clubs
```

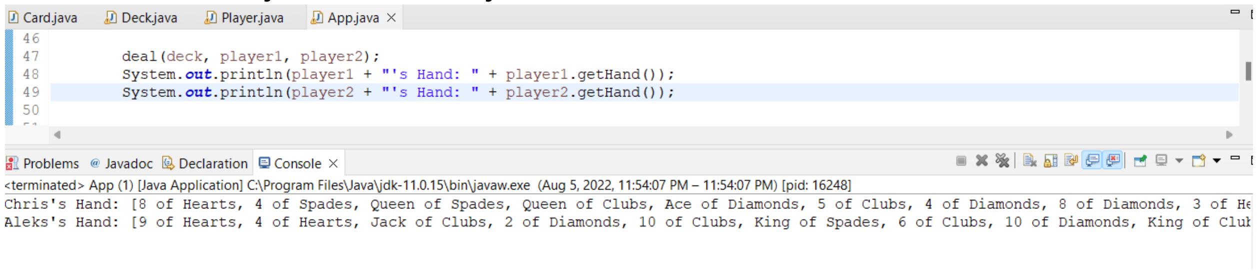
Listing Shuffled Deck:



```
        Deck deck = new Deck();
        deck.shuffle();

        System.out.println(deck);
```

```
<terminated> App (1) [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe  (Aug 5, 2022,
List of Cards:
 >> 10 of Clubs
 >> King of Hearts
 >> Jack of Diamonds
 >> Jack of Spades
 >> 10 of Hearts
 >> King of Diamonds
 >> 9 of Diamonds
 >> 5 of Spades
 >> 2 of Clubs
 >> 7 of Diamonds
 >> 6 of Clubs
 >> 9 of Hearts
 >> Queen of Diamonds
 >> 2 of Spades
 >> 9 of Spades
 >> Queen of Spades
 >> 3 of Clubs
 >> 5 of Clubs
 >> 3 of Spades
 >> 4 of Diamonds
 >> 8 of Diamonds
 >> 2 of Hearts
 >> 6 of Hearts
 >> 2 of Diamonds
 >> 4 of Hearts
 >> Jack of Hearts
 >> Queen of Clubs
 >> 7 of Clubs
 >> 3 of Hearts
 >> 8 of Hearts
 >> 4 of Clubs
 >> Jack of Clubs
 >> 6 of Diamonds
```

Deals Hands to Player One and Player Two:



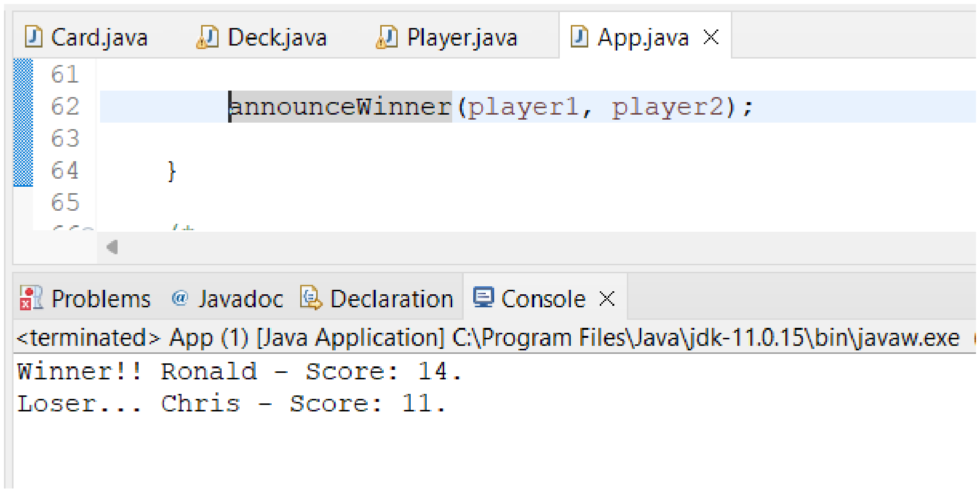
Card.java Deck.java Player.java App.java

```
        deal(deck, player1, player2);
        System.out.println(player1 + "'s Hand: " + player1.getHand());
        System.out.println(player2 + "'s Hand: " + player2.getHand());
```

Problems Javadoc Declaration Console

```
<terminated> App (1) [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe  (Aug 5, 2022, 11:54:07 PM – 11:54:07 PM) [pid: 16248]
Chris's Hand: [8 of Hearts, 4 of Spades, Queen of Spades, Queen of Clubs, Ace of Diamonds, 5 of Clubs, 4 of Diamonds, 8 of Diamonds, 3 of He
Aleks's Hand: [9 of Hearts, 4 of Hearts, Jack of Clubs, 2 of Diamonds, 10 of Clubs, King of Spades, 6 of Clubs, 10 of Diamonds, King of Clut
```

Announces Winner and Scores:



Card.java Deck.java Player.java App.java

```
        announceWinner(player1, player2);

    }
```

Problems Javadoc Declaration Console

```
<terminated> App (1) [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe
Winner!! Ronald - Score: 14.
Loser... Chris - Score: 11.
```

URL to GitHub Repository:

https://github.com/JustinPayne96/GameOfWar