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.

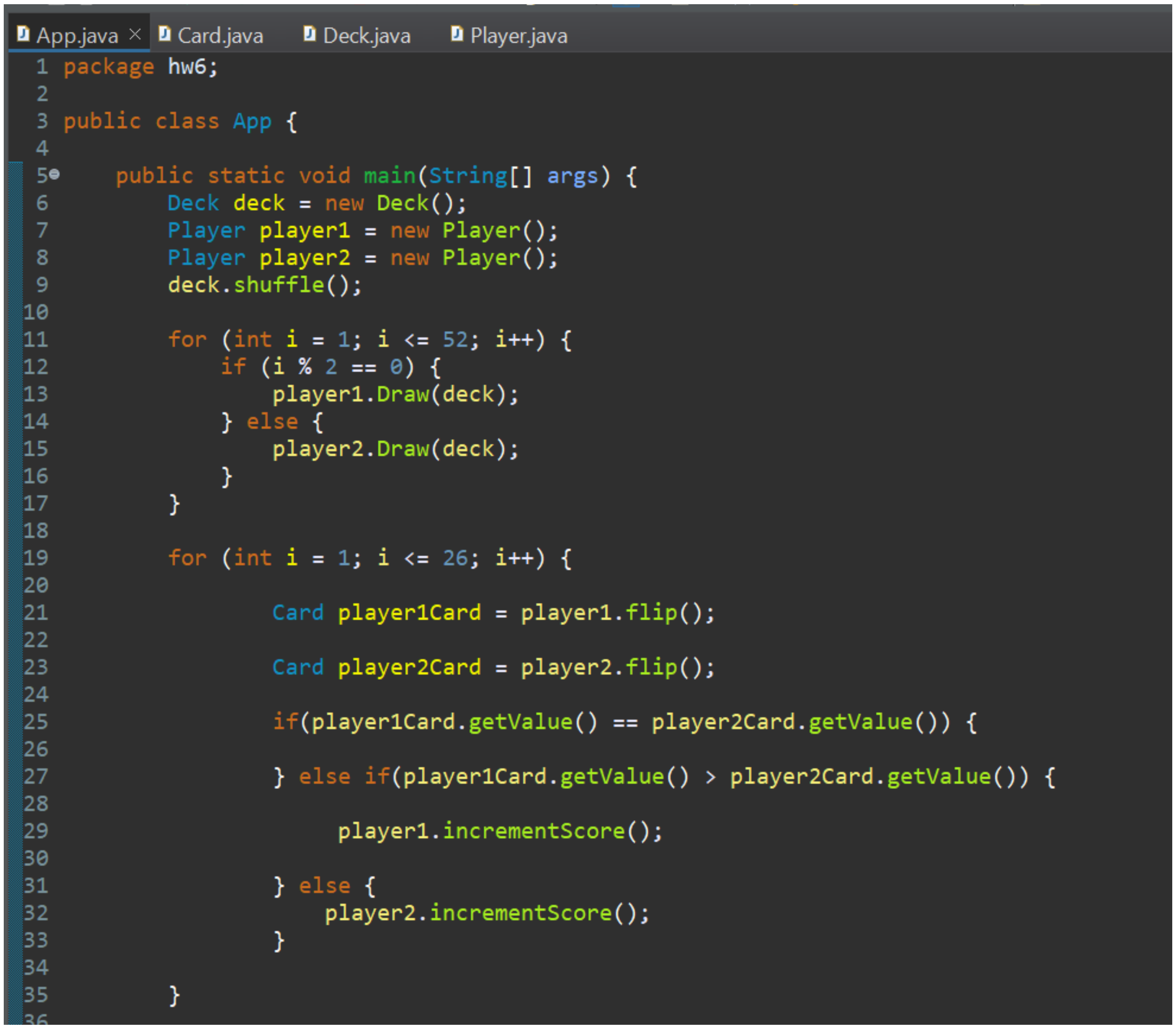
Coding Steps:

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

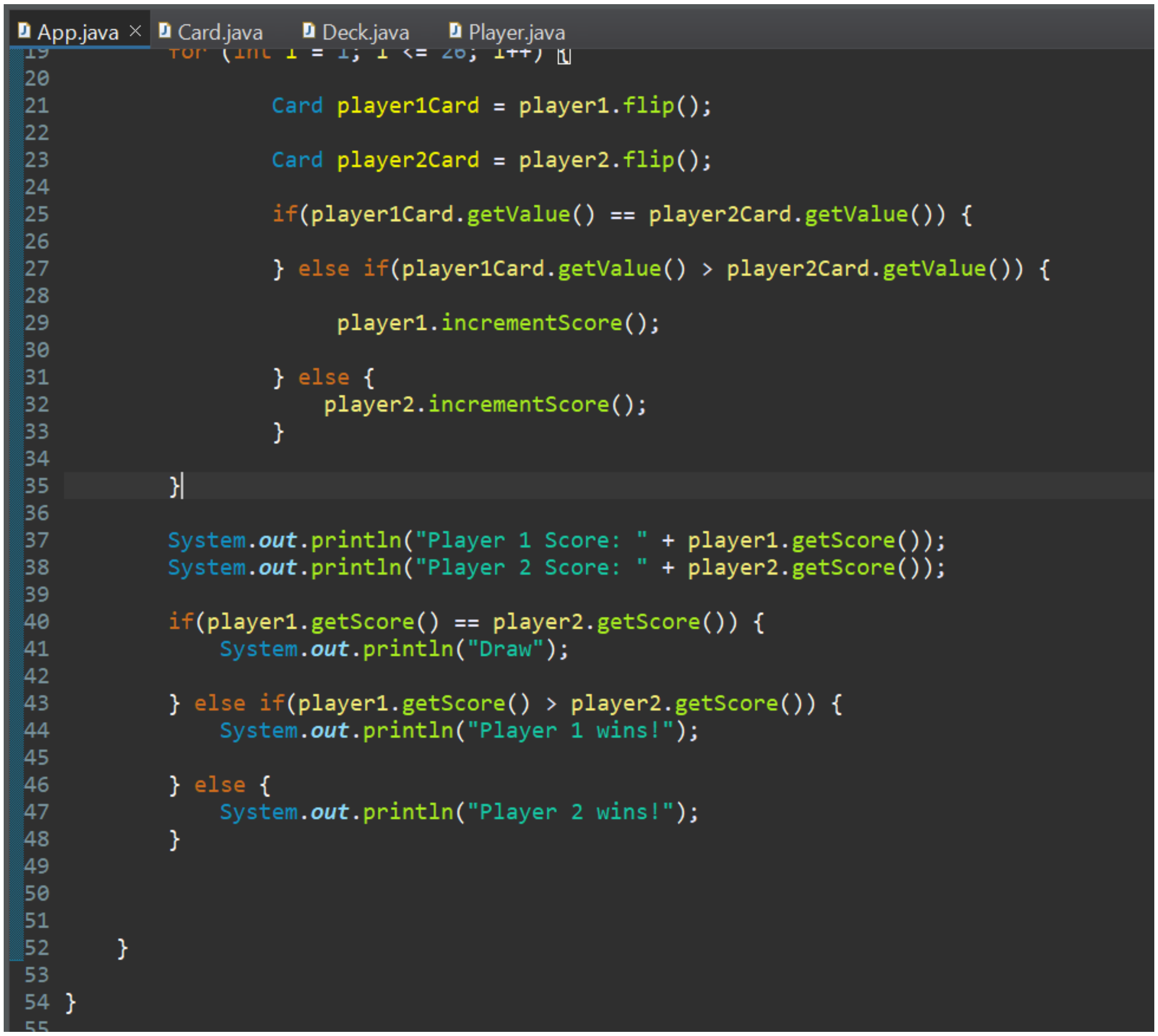
1. Create the following classes.
    a. Card
        i. Fields
            1. **value** (contains a value from 2-14 representing cards 2-Ace)
            2. **name** (e.g. Ace of Diamonds, or Two of Hearts)
        ii. Methods
            1. Getters and Setters
            2. **describe** (prints out 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.

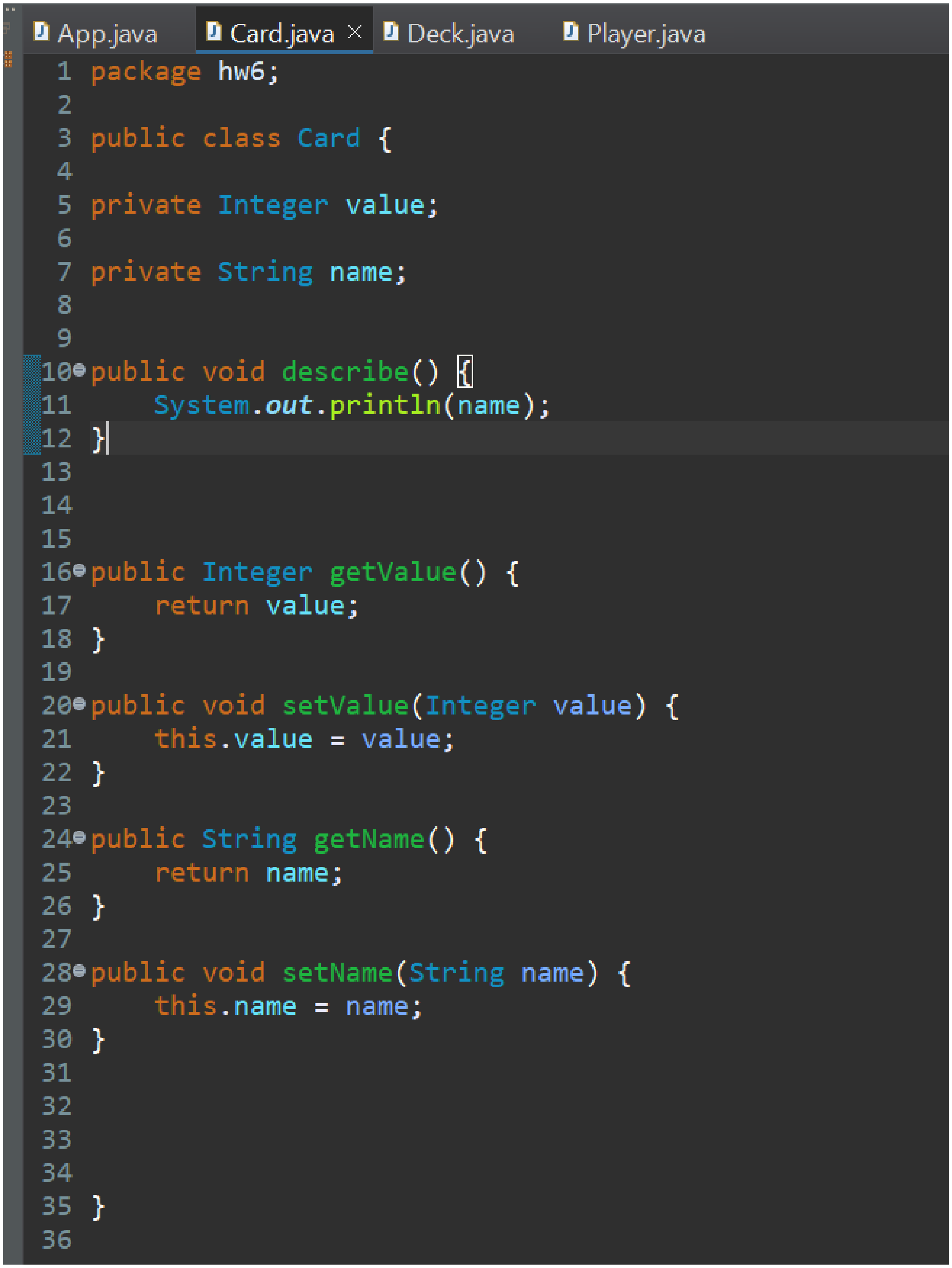
Screenshots of Code:



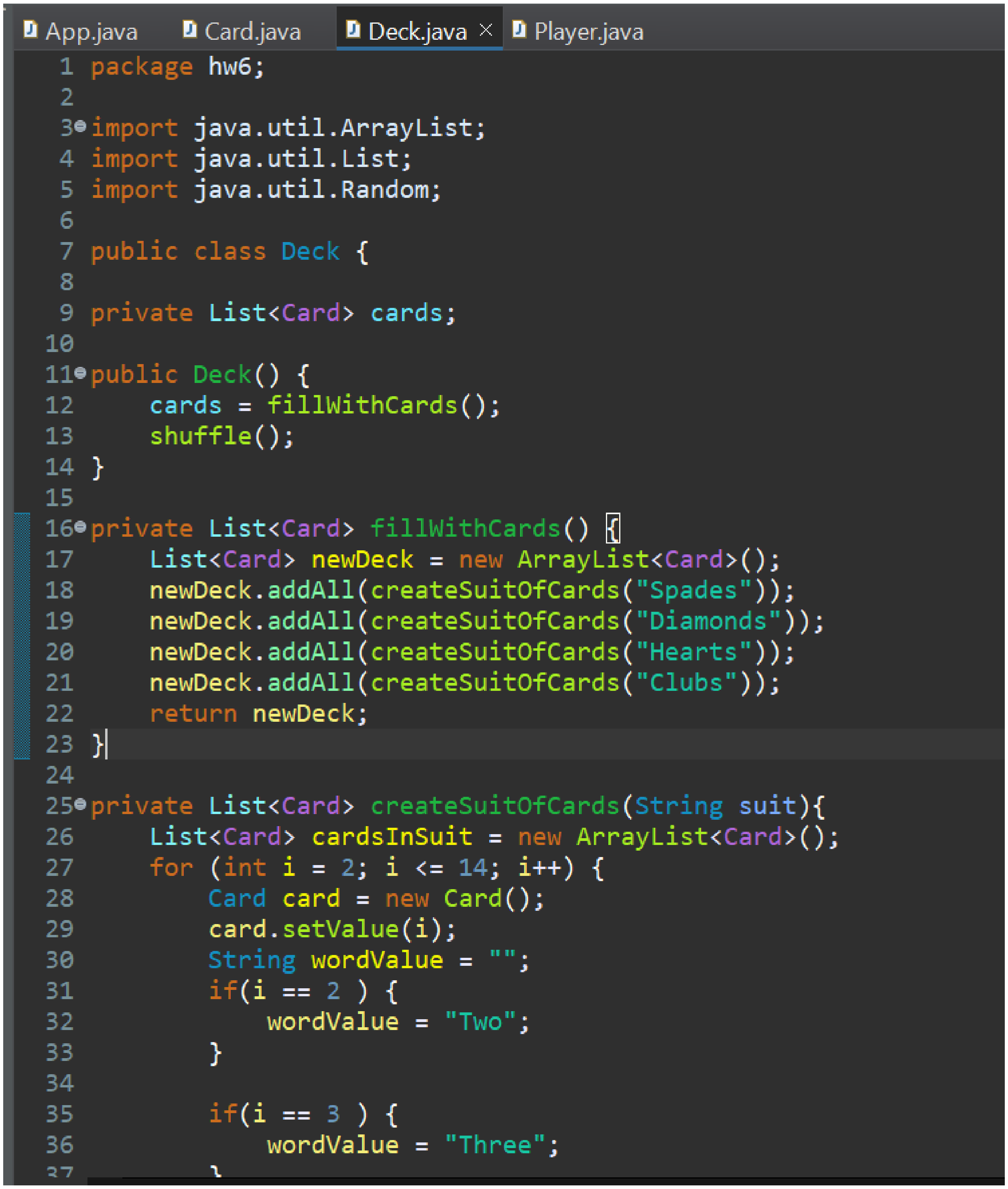
```java
package hw6;

public class App {

    public static void main(String[] args) {
        Deck deck = new Deck();
        Player player1 = new Player();
        Player player2 = new Player();
        deck.shuffle();

        for (int i = 1; i <= 52; i++) {
            if (i % 2 == 0) {
                player1.Draw(deck);
            } else {
                player2.Draw(deck);
            }
        }

        for (int i = 1; i <= 26; i++) {

                Card player1Card = player1.flip();

                Card player2Card = player2.flip();

                if(player1Card.getValue() == player2Card.getValue()) {

                } else if(player1Card.getValue() > player2Card.getValue()) {

                     player1.incrementScore();

                } else {
                    player2.incrementScore();
                }

        }
```



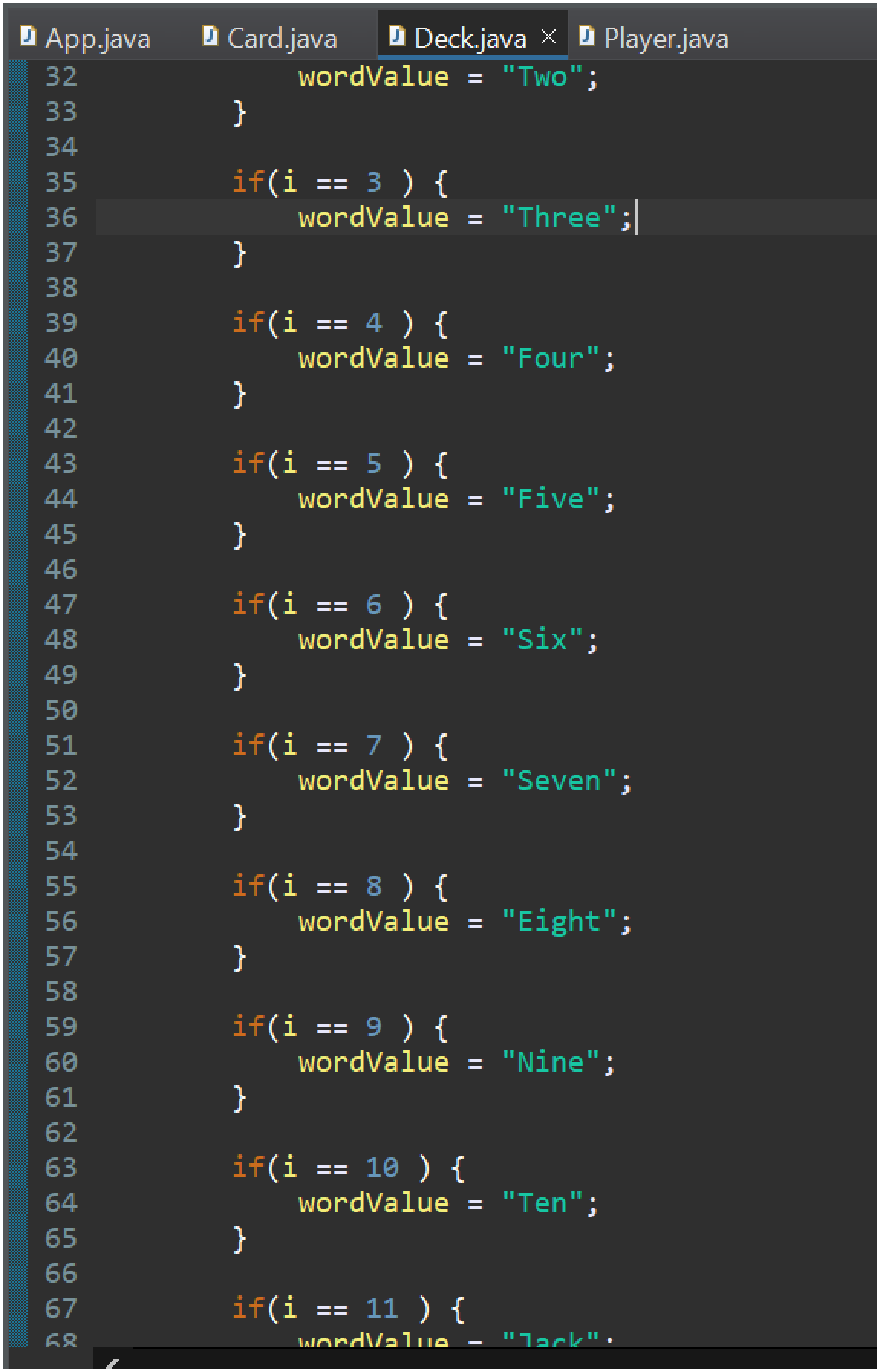
```java
        for (int i = 1; i <= 26; i++) {

                Card player1Card = player1.flip();

                Card player2Card = player2.flip();

                if(player1Card.getValue() == player2Card.getValue()) {

                } else if(player1Card.getValue() > player2Card.getValue()) {

                     player1.incrementScore();

                } else {
                    player2.incrementScore();
                }

        }

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

        if(player1.getScore() == player2.getScore()) {
            System.out.println("Draw");

        } else if(player1.getScore() > player2.getScore()) {
            System.out.println("Player 1 wins!");

        } else {
            System.out.println("Player 2 wins!");
        }



    }

}
```



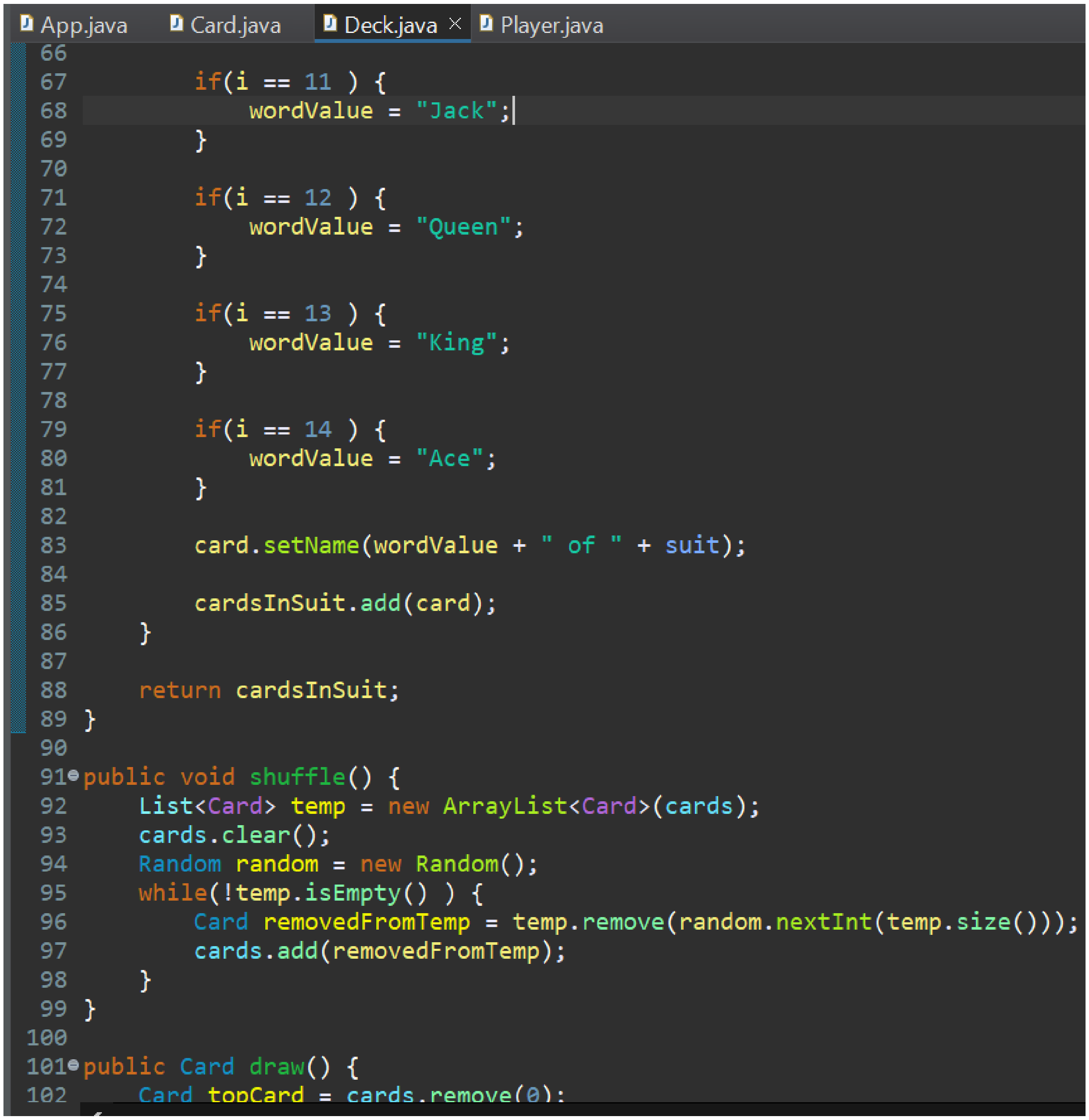
```java
package hw6;

public class Card {

private Integer value;

private String name;


public void describe() {
	System.out.println(name);
}



public Integer getValue() {
	return value;
}

public void setValue(Integer value) {
	this.value = value;
}

public String getName() {
	return name;
}

public void setName(String name) {
	this.name = name;
}




}

```


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

```java
package hw6;

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

public class Deck {

private List<Card> cards;

public Deck() {
    cards = fillWithCards();
    shuffle();
}

private List<Card> fillWithCards() {
    List<Card> newDeck = new ArrayList<Card>();
    newDeck.addAll(createSuitOfCards("Spades"));
    newDeck.addAll(createSuitOfCards("Diamonds"));
    newDeck.addAll(createSuitOfCards("Hearts"));
    newDeck.addAll(createSuitOfCards("Clubs"));
    return newDeck;
}

private List<Card> createSuitOfCards(String suit){
    List<Card> cardsInSuit = new ArrayList<Card>();
    for (int i = 2; i <= 14; i++) {
        Card card = new Card();
        card.setValue(i);
        String wordValue = "";
        if(i == 2 ) {
            wordValue = "Two";
        }

        if(i == 3 ) {
            wordValue = "Three";
```

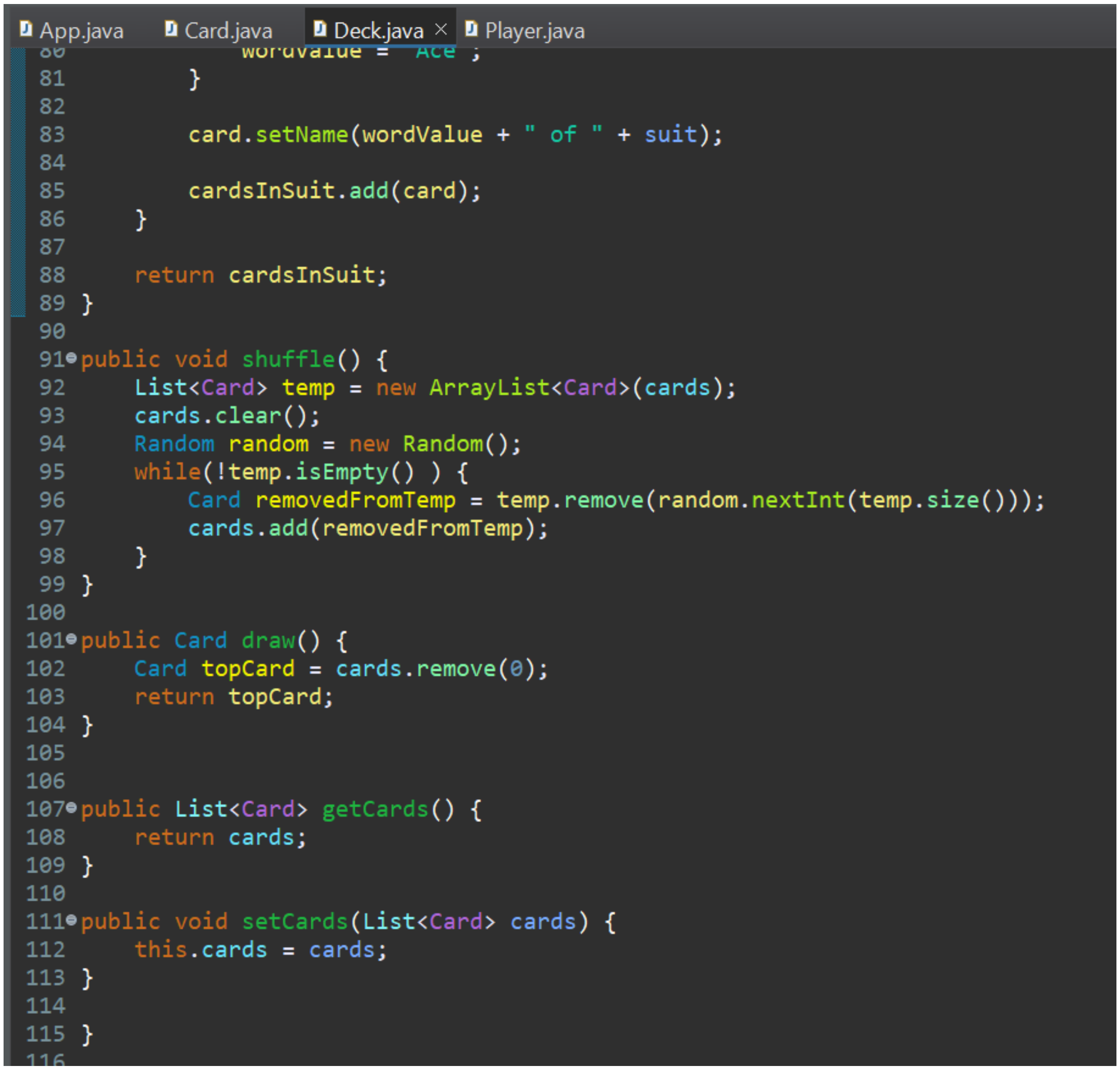


```java
                wordValue = "Two";
            }

            if(i == 3 ) {
                wordValue = "Three";
            }

            if(i == 4 ) {
                wordValue = "Four";
            }

            if(i == 5 ) {
                wordValue = "Five";
            }

            if(i == 6 ) {
                wordValue = "Six";
            }

            if(i == 7 ) {
                wordValue = "Seven";
            }

            if(i == 8 ) {
                wordValue = "Eight";
            }

            if(i == 9 ) {
                wordValue = "Nine";
            }

            if(i == 10 ) {
                wordValue = "Ten";
            }

            if(i == 11 ) {
                wordValue = "Jack";
```

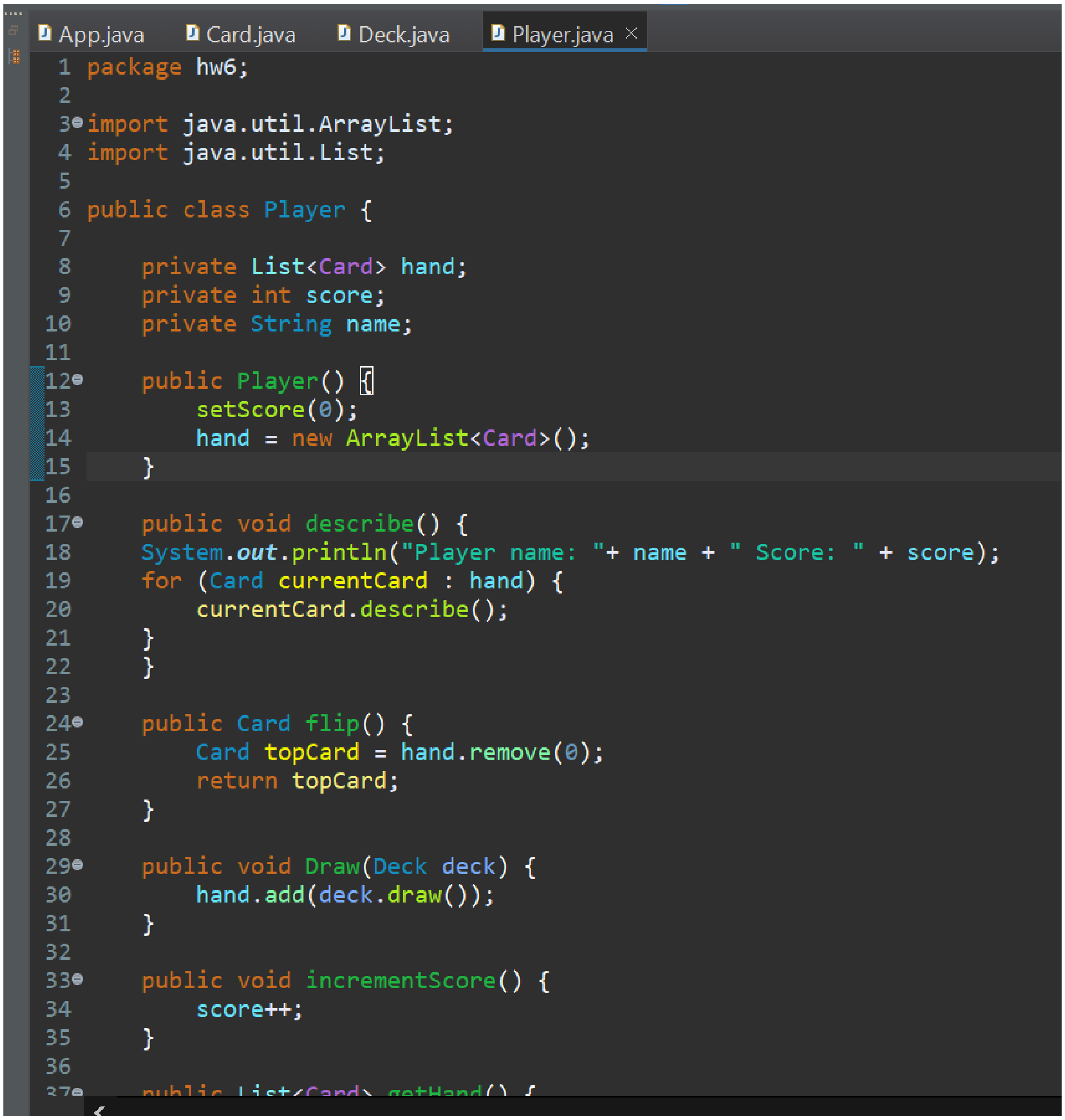


App.java   Card.java   Deck.java   Player.java

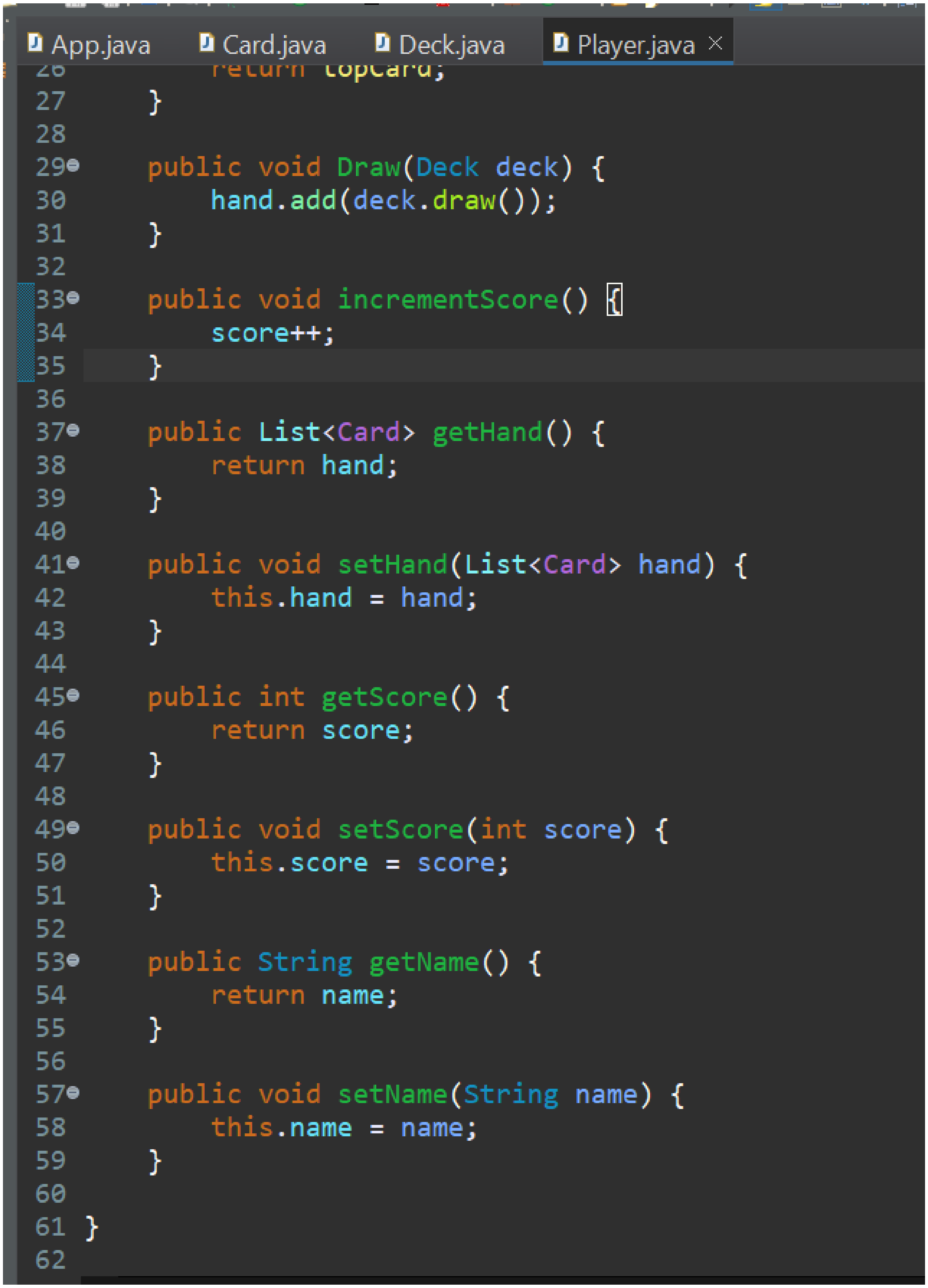
```java

        if(i == 11 ) {
            wordValue = "Jack";
        }

        if(i == 12 ) {
            wordValue = "Queen";
        }

        if(i == 13 ) {
            wordValue = "King";
        }

        if(i == 14 ) {
            wordValue = "Ace";
        }

        card.setName(wordValue + " of " + suit);

        cardsInSuit.add(card);
    }

    return cardsInSuit;
}

public void shuffle() {
    List<Card> temp = new ArrayList<Card>(cards);
    cards.clear();
    Random random = new Random();
    while(!temp.isEmpty() ) {
        Card removedFromTemp = temp.remove(random.nextInt(temp.size()));
        cards.add(removedFromTemp);
    }
}

public Card draw() {
    Card topCard = cards.remove(0);
```



```java
            wordValue = "Ace";
        }

        card.setName(wordValue + " of " + suit);

        cardsInSuit.add(card);
    }

    return cardsInSuit;
}

public void shuffle() {
    List<Card> temp = new ArrayList<Card>(cards);
    cards.clear();
    Random random = new Random();
    while(!temp.isEmpty() ) {
        Card removedFromTemp = temp.remove(random.nextInt(temp.size()));
        cards.add(removedFromTemp);
    }
}

public Card draw() {
    Card topCard = cards.remove(0);
    return topCard;
}


public List<Card> getCards() {
    return cards;
}

public void setCards(List<Card> cards) {
    this.cards = cards;
}

}
```

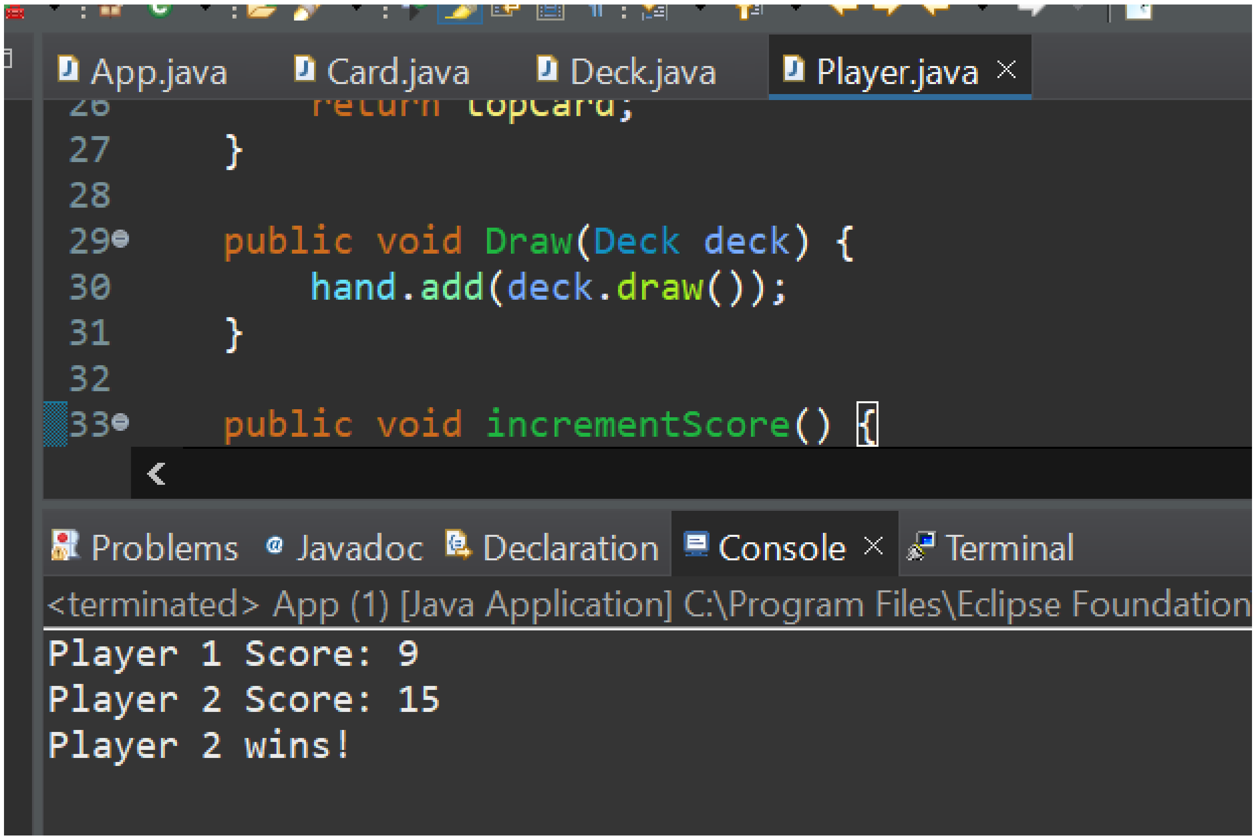


```java
package hw6;

import java.util.ArrayList;
import java.util.List;

public class Player {

    private List<Card> hand;
    private int score;
    private String name;

    public Player() {
        setScore(0);
        hand = new ArrayList<Card>();
    }

    public void describe() {
    System.out.println("Player name: "+ name + " Score: " + score);
    for (Card currentCard : hand) {
        currentCard.describe();
    }
    }

    public Card flip() {
        Card topCard = hand.remove(0);
        return topCard;
    }

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

    public void incrementScore() {
        score++;
    }

    public List<Card> getHand() {
```



```java
        return topCard;
    }

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

    public void incrementScore() {
        score++;
    }

    public List<Card> getHand() {
        return hand;
    }

    public void setHand(List<Card> hand) {
        this.hand = hand;
    }

    public int getScore() {
        return score;
    }

    public void setScore(int score) {
        this.score = score;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

}

```

Screenshots of Running Application:



```
        return topCard;
    }

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

    public void incrementScore() {
```

```
<terminated> App (1) [Java Application] C:\Program Files\Eclipse Foundation
Player 1 Score: 9
Player 2 Score: 15
Player 2 wins!
```

URL to GitHub Repository:

https://github.com/tyoung3614/hw6