

UML Diagram

Player

```

- String  _name
- int     _number
- Card[]  _hand
-----
+ Card[]  getHand()
+ String  getName()
+ int     getNumber()
+ void    drawCard() //removes card from deck, adds the same card to _hand
+ boolean isWinner() //returns true when _hand is empty
+ void    sortHand() //sorts hand according to suite/type
+ Card    playCard() //First checks if the card is playable by calling the card's isPlayable()
function. Removes card from _hand and it becomes woo's _topCard. The current _topCard gets
copied into the _discardPile.

```

Card (abstract class)

```

- int  _type //0-9 have their number, Wild-10, WildDraw4-11, Draw2-12, Skip-13, Reverse-14
- int  _suite //Red-0 Blue-1 Green-2 Yellow-3
-----
+ int   getType()
+ int   getSuite()
+ boolean isPlayable() //Determines whether the card is playable according to the _topCard
+ void   action() //When each card gets played, it's action gets executed. For example,
the draw2 card makes the next player draw 2 cards, while the numbered cards have an empty
action.
+ String toString() //displays the card type in colored text in the terminal

```

0 (extends Card)**1** (extends Card)**2** (extends Card)**3** (extends Card)**4** (extends Card)**5** (extends Card)

6 (extends Card)

7 (extends Card)

8 (extends Card)

9 (extends Card)

Wild (extends Card)

+ int _nextSuite //the user-selected suite the next card must follow

WildDraw4 (extends Card)

+ int _nextSuite //the user-selected suite the next card must follow

+ void draw4(player) //calls the player's drawCard() function 4 times

Draw2 (extends Card)

+ void draw2(player) //calls the player's drawCard() function 2 times

Skip (extends Card)

Reverse (extends Card)

=====

Woo

- Card[] _deck //the central deck in which all cards start

- Card[] _discardPile //the pile of cards that have been played

- Card _topCard //the faceup card that determines the suite and type of the next card to be played

- Player[] _turnOrder //Array of players that determines the order of play

- Player _currentPlayer //the player that has the current turn

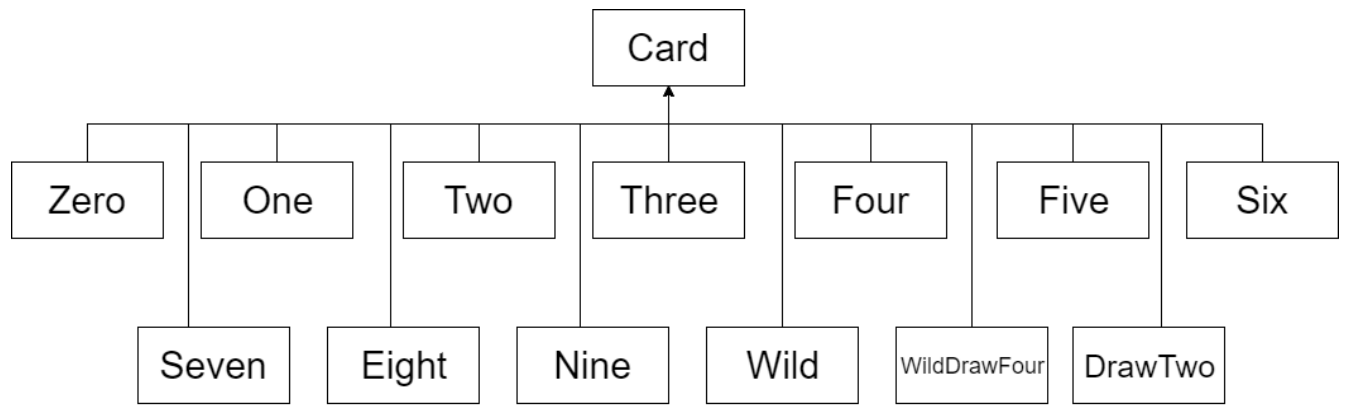
+ boolean anyWinner() //prompts each player to execute isWinner()

+ void shuffle() //randomizes order of array

+ void distribute() //each player gets 7 cards from _deck at the beginning

+ void playCard(Player, int) //_currentPlayer and the user-selected int are parameters. The int represents the index of the card to be played in the player's _hand. This function then calls the player's own playCard() function.

+ void main()



Player Woo