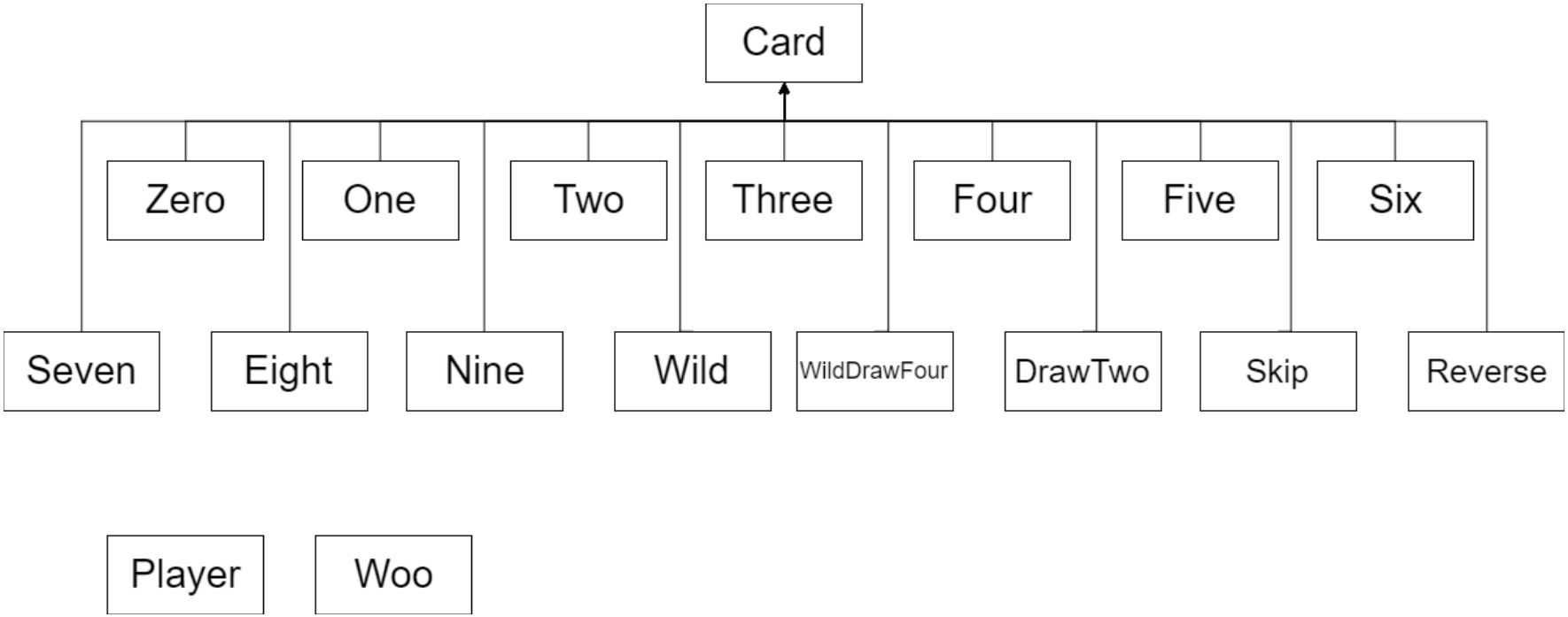
Card
Zero
One
Two
Three
Four
Five
Six
Seven
Eight
Nine
Wild
WildDrawFour
DrawTwo
Skip
Reverse
Player
Woo

YaLikeJazz APCS1 pd 08
Colin Hosking, George Liang, Xinyi Huang 2018-01-16

UML Diagram

Player

- String _name
- int _number
+ ArrayList<Card> _hand

+ Player(String name, int num)
+ ArrayList<Card> getHand()
+ String getName() //returns player name
+ boolean isWinner() //returns true when _hand is empty
+ void sortHand() //sorts hand by color
+ boolean handFull() //returns true if a player has 7 or more cards, returns false otherwise

Card (abstract class)

int _type //Zero - Nine have 0 - 9, Wild - 10, WildDrawFour - 11, DrawTwo - 12, Skip -13, Reverse - 14
int _suite //Red - 0 Blue - 1 Green - 2 Yellow - 3 Special - 4
+ String ANSI_RESET //ANSI constant for Resetting String color
+ String ANSI_RED //ANSI constant for Red String color
+ String ANSI_GREEN //ANSI constant for Green String color
+ String ANSI_YELLOW //ANSI constant for Yellow String color
+ String ANSI_BLUE //ANSI constant for Blue String color
+ String ANSI_BLACK //ANSI constant for Black String color

+ int getType() //Returns card type
+ int getSuite() //Returns card suite
+ void setSuite(int) //Sets card suite
+ void action() //Abstract method that is called to execute a card's action. For example, a Wild card will change suite. Numbered cards do not have any action.
+ String toString() //displays the card in colored text in the terminal

==========

Zero (extends Card)

+ Zero(int) //Sets _suite to suite and sets _type to 0

One (extends Card)
+ One(int) //Sets _suite to int suite and sets _type to 1

Two (extends Card)
+ Two(int) //Sets _suite to int suite and sets _type to 2

Three (extends Card)
+ Three(int) //Sets _suite to int suite and sets _type to 3

Four (extends Card)
+ Four(int) //Sets _suite to int suite and sets _type to 4

Five (extends Card)
+ Five(int) //Sets _suite to int suite and sets _type to 5

Six (extends Card)
+ Six(int) //Sets _suite to int suite and sets _type to 6

Seven (extends Card)
+ Seven(int) //Sets _suite to int suite and sets _type to 7

Eight (extends Card)
+ Eight(int) //Sets _suite to int suite and sets _type to 8

Nine (extends Card)
+ Nine(int) //Sets _suite to int suite and sets _type to 9

Wild (extends Card)
+ Wild() //Sets _suite to 4 and sets _type to 10

WildDrawFour (extends Card)
+ WildDrawFour() //Sets _suite to 4 and sets _type to 11

DrawTwo (extends Card)
+ DrawTwo(int) //Sets _suite to int suite and sets _type to 12

Skip (extends Card)
+ Skip(int) //Sets _suite to int suite and sets _type to 13

Reverse (extends Card)
+ Reverse(int) //Sets _suite to int suite and sets _type to 14
==========

Woo
- ArrayList<Card> _deck // Container of cards from which players draw from
- ArrayList<Card> _discardPile // Container of cards already played
- ArrayList<Card> _topCard // Container for the faceup card that determines the suite and type of the next card to be played
- ArrayList<Player> _turnOrder //Array of players that determines the order of play
- Player _currentPlayer //the player that has the current turn
- int numPlayers //the number of players that are playing Juno
- int cardtoPlay //the index of the card that _currentPlayer will play
---
+ Woo() //Instantiates _deck, _turnOrder, _topCard, and _discardPile
+ boolean isInt(String s) //checks if a String can be converted into int. Helper function for checking the input from the terminal is an int.
+ void setup() //Instantiates each Card, Players, and shuffles the deck with shuffle()
+ void playerDraw(Player) //Removes the top card from deck and gives it to the player
+ boolean anyWinner() //Determines if the game has any winners by making each player call isWinner(). Returns true if there is a winner, false otherwise
+ void shuffle() //randomizes order of deck
+ void distribute() //each player gets 7 cards from _deck at the beginning and they sort their hand with sortHand()
+ 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.
+ void printUserDisplay() //Prints empty lines to hide previous Player's hand and displays the current Player's turn, topcard, and the current Player's hand
+ void main() //Instantiates a new game. Game will be setup and cards will be distributed to all Players. While there are no winners, Players will continue to play the game by passing each other the computer each turn. Once there is a winner, the winning Player will be displayed.