Player RouletteGame - currentValue : double tableWinnings:int - percentOfWinnings : double tableBets : int - betSpots : int[] players : Player[] - timesBets : int round:int - tokenValue : int - status : String #RouletteGame() - name : String + toString() : String + equals() : boolean + Player(name : String) + setPlayer(player : Player, index : int) : void + getStatus() : String + removePlayer(index : int) : Player + setStatus(status : String) + getPlayers() : Player[] + addCurrentValue() : int + getLayout() : String[] + getName() : String + spin() : int + setName(name : String) + getRound() : int + setToken(number : int, spot : int) : void + addBets(totalRoundBets : int) : void + getBets() : int[] + resetBets() : void + getTokenValue() : int + getTableWinnings() : int + setTokenValue(value : int) + setTableWinnings(tableWinnings : int) : void + getTimesBet() : int + getTableBets():int + getPercentOfWinnings() : double + setTableBets(tableBets : int) : void + setPercentOfWinnings(percent : double) : void + stringToStraightNumber(straight : String) : String + getCurrentValue() : double + setCurrentValue(value : double) : void + equals(player : Player) : boolean Extends + toString() : String AmericanRoulette EuropeanRoulette + AmericanRoulette() + EuropeanRoulette() Extends Extends + stringToStraightNumbers(straight :String) : String + stringToStraightNumbers(straight :String) : String + getLayout() : String[] + getLayout() : String[] NormalPlayer VIP currentFakeVal : int + NormalPlayer(name : String) + equals(normalPlayer : NormalPlaye + VIP(name : String) + getTokenWorth():int addFakeVal(): void + addCurrentValue() : void + equals(vip : VIP) : boolean Dealer + getCurrentFakeVal() : double + setCurrentFakeVal(val : double) : void game:RouletteGame + toString() : String winningsAnnouncement : String + turnWheel(): void

+ inputBets() : void + toString() : String + equals() : boolean + getGame() : RouletteGame + setGame(game : RouletteGame) : void - calculateWinnings(winningNubmer : int) : void

- announceWinnings(): void