

## Table

```
-DeckOfCards myDeck;  
-map<string, int> players;  
-map<string, int> activePlayers;  
-list<Card> player1Cards;  
-list<Card> player2Cards;  
-list<Card> player3Cards;  
-list<Card> dealerCards;  
-int highBet;  
-int pot;  
-int turn;
```

```
Table(int num_Players)  
void setTurn(int t)  
void printPlayers()  
void placeBets()  
void startTurn()  
int findWinner(pair<int, int> p1, pair<int, int> p2, pair<int, int> p3)  
void declareWinner(int player)  
void addEarnings(string winner)  
void raise(string raiser)
```