## Deck cards: vector<pair<string, int>>\* - sDeck : stack<pair<string, int>>\* Game + Deck(): + Deck(Deck): players : list<Player\*>\* + ~Deck(): titles : queue<int> + draw(): pair<string, int> rounds : int + getDeck(): stack<pair<string, int> passed : int + insert(pair<string, int>) : void + resetDeck(): void pCount: int pActive : int tCC : int drawPile : Deck discard : Deck key : map<string, int>\* Player tKey: map<int, string>\* + Passed : exception class hand : list<pair<string, int>>\* played: list<pair<string, int>> + Game(): + Game(int): title : int name : string + Game(Game): + ~Game() emptyHand : bool + start() : void + trick(list<Player\*>::iterator) : void + Player(): + playcards(Player\*) : void + Player(string) : + prntRanks(): void + Player(Player) : + dealCards(int) : void + ~Player() : + cardTrade() : void + setName(string) : void + resetTitles(): void + getName() : string + badRank(stirng) : bool + dupRmve(): void + dspHand() : void + addHand(pair<string, int>) : void + clearHand(): void + empt(): bool + setTitle(int) : void + getTitle(): int

+ playCards(string, int) : list<pair<string, int>>

+ getBest(int) : list<pair<string, int>> + getWorst(int) : list<pair<string, int>>

+ reject(): void