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>* tKey: map<int, string>* srt(list<pair<string, int>>*&, int): void + Passed : exception class + Game(): + Game(int) : Player + Game(Game) : + ~Game() hand : AVLTree* + start(): void played : list<pair<string, int>> + trick(list<Player*>::iterator) : void title: int + playcards(Player*) : void name: string + prntRanks(): void emptyHand : bool + dealCards(int) : void + cardTrade(): void + resetTitles(): void + Player(): + badRank(stirng) : bool + Player(string) : + Player(Player): + ~Player(): + setName(string) : void + getName(): string + dupRmve(): void + dspHand() : void + addHand(pair<string, int>) : void + clearHand(): void **AVLTree** + empt(): bool + setTitle(int) : void + root : BNTnode* + getTitle(): int + cKey : map<int, int> + playCards(string, int) : list<pair<string, int>> + getBest(int) : list<pair<string, int>> + JSHash(string) : unsigned int + getWorst(int) : list<pair<string, int>> + height(BNTnode*): int + reject() : void + diff(BNTnode*) : int + rr rotation(BNTnode*) : BNTnode* + II rotation(BNTnode*) : BNTnode* + Ir_rotation(BNTnode*) : BNTnode* + rl_rotation(BNTnode*) : BNTnode* + balance(BNTnode*) : BNTnode* + insert(BNTnode*&, pair<string, int>): BNTnode* + display(BNTnode*, int): void + inorder(BNTnode*) : void + preorder(BNTnode*) : void + postorder(BNTnode*) : void + dsp(): void + dspSort(): void + AVLTree(): + ~AVLTree(): + clear(): void + destruct(BNTnode*): void + treeDel(BNTnode*, string) : BNTnode* + confirmNode(BNTnode*, string) : bool + delCall(string) : bool + getMax(): pair<string, int> + getMin(): pair<string, int> + maxFind(BNTnode*) : pair<string, int> + minFind(BNTnode*) : pair<string, int> + inCall(pair<string, int>) : void