

Deck
-name:string -suit:string -value:int -ary:array<int> -q:queue<int>
+Deck(): +SetVal(string):void +getSui():string +getName():string +getVal():int> +fillDeck(int):Deck +fillQue(int):queue<int> +shuffle():void +getQue():queue<int>

Node
-hash2:int -name:string -bet:float -next*:Node -data:float -left*:Node -right*:Node -count:int

Hash Table
-hash:unsigned int -ary*:Node -size:int -hshSze:int -nam:string -dat*:string -bets*:float -betC:float -last*:Node
+setDat(string*,int,float*):void +create(int):void +HashTable() +ELFHash(string):void +insert():void +srch(string):void +end():void

Hand
-uHand:Deck* -dHand:Deck*
+deal(Deck*,queue<int>):void +getUhand():Deck* +getDhand():Deck*

Menu
-choose:char -bet:float -in:ifstream -out:ofstream -data:fstream -file:fstream -lst:list<float> -lst2:list<float> -mp:multimap<float,string> -mp2:multimap<string,float> -s:multiset<float> -stak:stack<int> -name:string -choice:char
+menu():void +add():void +check():void +betMon():void +getBet():float +getChse():char +warn():void +&operator(ostream&,Menu&):ostream +createList():void +leadBrd:void +askName():void +getName():string +betHist():void +createMap():void

Player
-users:Deck* -dealers:Deck* -binCards:Deck* -cards:Deck* -count,sum,dSum,games,offSet:int -result,names:string -choice:char -bet:float -dbl:bool -dataFile,totGame,binFile,record:fstream -in:ifstream -out:ofstream
+Player(Player &) +Player(Deck*, Deck*, int) +Player(Deck*, Deck*, int, string) +operator++():void +operator++(int):void +setUsers(Deck*):void +setDealers(Deck*):void +getUsum():int +getDsum():int +getChoice():char +usersH():Deck* +dealersH():Deck* +play():void +isAce():void +display():void +dCards():void +winLoss():void +inFile(int&):void +wrtRecord():void +dblDown():void +getRes():string +getBet():float +getCount():int +monHist(string,float,string):void

Tree
-root*:Node
+height(Node*):int +diff(Node*):int +rr(Node*):Node* +ll(Node*):Node* +balance(Node*):Node* +insert(Node*, int):Node* +inorder(Node*):void