Black Jack Semester Project UML Diagram

Paul Schwappach

Card

- suit : char
- rank : char

+ Card()
+ Card(char, char)
+ setSuit(char) : void
+ setRank(char) : void
+ getSuit() : char
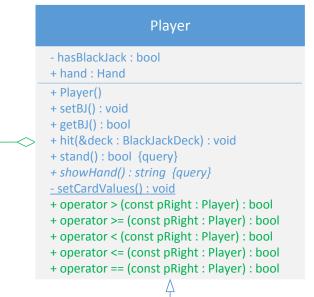+ getRank() : char
+ cardState() : string

BlackJackDeck

- numDecks : int
- deck : vector<Card>

+ BlackJackDeck()
+ BlackJackDeck(dcks : int)
+ deckBuilder(decks : int) : void
+ shuffle() : void
+ resetDeck() : void
+ dealCard() : Card
+ deckState() : string {query}
+ cLastCardDealt : string {query}

Black Jack Semester Project UML Diagram

Paul Schwappach

#Hand

cardValues : map < char : int >
hand : vector<Card>

+ Hand()
+ Hand (num : int)
+ addCard(&deck : BlackJackDeck) : void
+ addCard(&card : Card) : void
+ getValue() : int {query}
+ getSize() : int {query}
+ sameRank() : bool {query}
+ card2IsAce() : bool {query}
+ removeCard() : Card
+ hasBusted() : bool {query}
+ handState() : string {query}
+ dealerHandState() : string {query}
+ operator - (&splitHand : Hand) : void
+ operator + (&deckRight : BlackJackDeck) : void

Player

- hasBlackJack : bool
+ hand : Hand

+ Player()
+ setBJ() : void
+ getBJ() : bool
+ hit(&deck : BlackJackDeck) : void
+ stand() : bool {query}
+ showHand() : string {query}
- setCardValues() : void
+ operator > (const pRight : Player) : bool
+ operator >= (const pRight : Player) : bool
+ operator < (const pRight : Player) : bool
+ operator <= (const pRight : Player) : bool
+ operator == (const pRight : Player) : bool

User

- userName : string
- userMoney : int
- userBet : int
- userBet2 : int
- insurance : int
- hasDoubled : bool
- hasDoubledSplit : bool
- userWantsToQuit : bool
+ splitHand : Hand

+ User(decks : int)
+ User(name : string)
+ setName(name : string) : void
+ setBet(amount : double) : void
+ setBet2(amount : double) : void
+ setMoney(amount : double) : void
+ setInsurance(winnings : double) : void
+ getName() : string {query}
+ getBet() : double {query}
+ getBet2() : double {query}
+ getMoney() : double {query}
+ getInsurance() : bool {query}
+ UserWantsToQuit() : bool {query}
+ getHasDoubled() : bool {query}
+ getHasDoubledSplit() : bool {query}
+ canDoubleDown(handNum : int) : bool : {query}
+ doubleDown(&deck : BlackJackDeck, hasSplit : bool) : void
+ resetHand() : void
+ hasSplit() : bool {query}
+ split(&deck : BlackJackDeck) : void
+ showHand() : string {query}
+ showSplitHand() : string {query}
+ operator << (&strm : ostream, &user : const User) : friend
+ operator >> (&strm : ostream, &user : User) : friend

Dealer

+ finishRound(&deck : BlackJackDeck, value : int) : void
+ offerInsurance() : bool {query}
+ concealHand() : string {query}
+ showHand() : string