- idCount : int <<enumeration>> - id : int Blackjack Class Diagram - type : MessageType text : String - from : String - username : String - password : String - date : Date - card : Card + Message(type: MessageType, username: String, password : String, text : String, from : String, card : Card) + getID : int Client + getUsername : String + getPassword : String - Socket: socket + getType() : MessageType Server - inStream: ObjectinputStream + getText() : String - outStram: ObjectOutputStream <<enumeration>> - loggedInPlayers : - card : Card + getDate() : Date - loggedIn : Boolean Status List<Player> - players : List<Player> + getFrom() : String - Player: player - loggedInDealers : - dealer : Dealer ClientHandler + getCard() : Card - Game: game Accepted List<Dealer> - shoe : Shoe Denied + clientSocket : Socket - tables : List<Table> - currentPlayer : int + connectToServer(ipAddress : - timer : Timer <<enumeration>> String, port : int) : Boolean + ClientHandler(socket : Socket) + main(args : String[]) : void turnTimeoutTask : TimerTask Suit + sendLoginMessage(username : + run() : void maxPlayers : int <<enumeration>> String, password : String) : String - getLoggedInPlayer(username : Club + getTableCountMessage() : void String) : Player Spades - startTurnTimer() : void + disconnect() : void - getLoggedInDealer(username : Error Diamond + nextPlayerTurn(): void String) : Dealer Server + dealerTurn(): void Heart Client + roundEnd() : void + hit() : void + stay() : void Player Dealer - tables: ArrayList<tables> - maxTables: int - id : int - decks : ArrayList<Deck> - id : int - idCount : int · numDecks : int - username : String numCardsInShoe : int - password : String - username : String + addTable(): void - hand : Hand - password : String + removeTable(): void Shoe(numDecks: int) - handValue : int - balance : int + getAvaliableTable(): ArrayList<Tables> + getNumDecks(): int - hand : Hand + isTableAvaliable(): bool + getCard() : Card + Dealer(username : String, password : - currBet : int + reset() : void - stayed : boolean + getDealer() : Dealer + Player(username : String, password : String, balance : int) + getID(): int + getID(): int + getUsername() : String + getUsername() : String + getPassword() : String + getPassword() : String + getHand() : hand + getBalance() : String + getHandValue(): int players : ArrayList<Player> + getHand() : Hand + addCardToHand(newCard : Card) : void dealer : Dealer + getHandValue() : int + startGame(): void maxPlayers: int + startRound(): void + getCurrBet() : int maxDealers: int + dealStartingCards(): void + setBalance(amount : int) : void - game : Game + setHand(card1 : Card, card2 : Card) : void + setCurrBet(amount : int) + Table(dealer : Dealer) + withdrawalMoney(amount : int) : void + getPlayers() : ArrayList<Player> + depositMoney(amount : int) : void + getNumPlayers : int + addCardToHand(newCard : Card) : void + getDealer() : Dealer + hasStayed() : boolean + getMaxPlayers(): int + stay() : void + getMaxDealers() : int + clearPlayerHand(): void + getGame() : Game + isEmpty() : Boolean Hand Deck Card value : int # cards : List<Card> - value : int - cards : List<Card> suit: String - handTotal : int + Deck() - Busted : boolean + getNumCards() : int + Card() - hasBlackJack : boolean + isEmpty() : boolean + getValue() : int aceasEleven : int # fillDeack() : void + getSuit() : Suit + getSymbol(): Symbol + Hand() + Hand(firstCard : Card, secondCard : Card) + getHandTotal() : int + getCards() : List<Card> + addCard(newCard : Card) : void + isBusted() : Boolean + isBlackJack() : boolean + clearHand() : void

Message

- serialVersionUID : long