

PackageName

events

SystemEvent

- +getEventTime()
- +getEventInitiator()

gameFlowEvents

GameEvent

- +getEventAction()

MoveEvent

- +getAmountToRaise()

chatEvents

MessaveEvent

- +getContent()

WhisperEvent

- +getParticipantToSendTo()
- +getContent()

system

GameCenter

- +registerUser(UserDetails): void
- +login(Username, Password): void
- +logout(Username): void
- +editProfile(Username, UserDetails): void
- +createGame(Username, GameSettings): void
- +joinGame(GameName, Username, Specmode): void
- +findAvailableGames(User): List<Game>

game

Round

- +playTurnOfPlayer(MoveEvent)

Game

- +startNewRound(): void
- +joinGameAsPlayer(User): void
- +joinGameAsPlayer(User): void
- +removeParticipant(Participant): void

GameSettings

- gameType: GamePolicy
- gameTypeLimit: int
- minBet: int
- buyInPolicy: int
- chipPolicy: int
- playerRange: Pair<int, int>
- acceptsSpectating: boolean
- leagueCriteria: int

hand

HandCalculator

- getPossibleHands(List<Card>): Card[][]
- +getHand(List<Card>): Hand

Hand

- hand: List<Card>
- category: Category
- +computeTo(Hand): int

card

Dealer

- +deal(Collection<Player>): void
- +open(int): List<Card>

Deck

- prepare: void
- shuffle: void
- +get(): List<Card>

Card

- suit: Suit
- rank: Rank

participants

<<abstract>>
Participant

Player

- +payChip(int): void
- +chooseAction(List<GameActions>): GameAction
- +chooseAmountToRaise(int): int

Spectator

users

User

- +deposit(int): void
- +withdraw(int): int

Wallet

- +getBalance(): int
- +serBalance(int): void

LeaguesManager

- +addNewUserToLeague(User): void
- +updateUserLeague(User): void