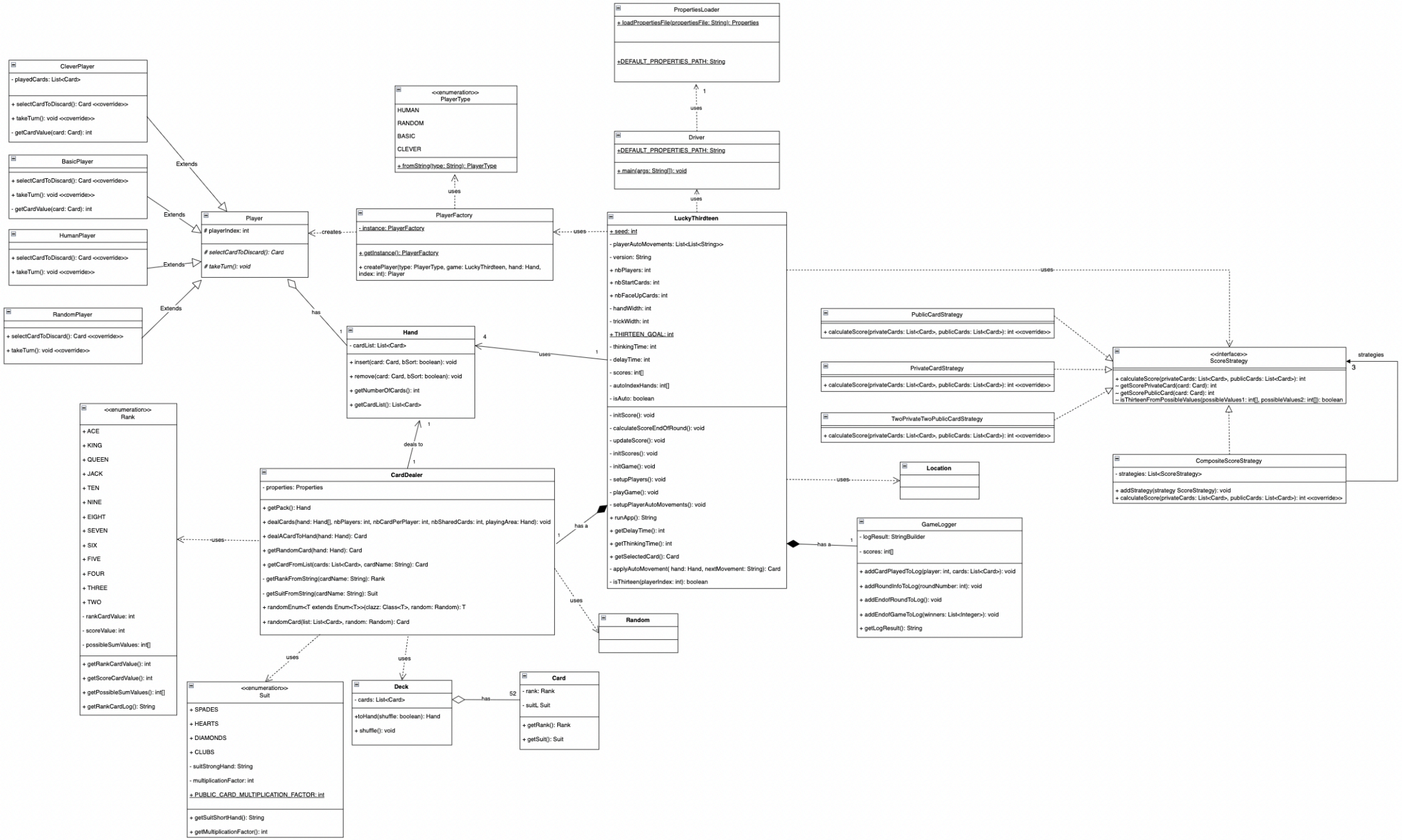
PropertiesLoader
+ loadPropertiesFile(propertiesFile: String): Properties
+DEFAULT_PROPERTIES_PATH: String
uses
1
Driver
+DEFAULT_PROPERTIES_PATH: String
+ main(args: String[]): void
uses
CleverPlayer
- playedCards: List<Card>
+ selectCardToDiscard(): Card <<coverride>>
+ takeTurn(): void <<coverride>>
- getCardValue(card: Card): int
Extends
BasicPlayer
+ selectCardToDiscard(): Card <<coverride>>
+ takeTurn(): void <<coverride>>
- getCardValue(card: Card): int
Extends
HumanPlayer
+ selectCardToDiscard(): Card <<coverride>>
+ takeTurn(): void <<coverride>>
Extends
RandomPlayer
+ selectCardToDiscard(): Card <<coverride>>
+ takeTurn(): void <<coverride>>
Extends
Player
playerIndex: int
selectCardToDiscard(): Card
takeTurn(): void
creates
<<enumeration>>
PlayerType
HUMAN
RANDOM
BASIC
CLEVER
+ fromString(type: String): PlayerType
uses
PlayerFactory
- instance: PlayerFactory
+ getInstance(): PlayerFactory
+ createPlayer(type: PlayerType, game: LuckyThirdteen, hand: Hand, index: int): Player
uses
LuckyThirdteen
+ seed: int
- playerAutoMovements: List<List<String>>
- version: String
+ nbPlayers: int
+ nbStartCards: int
+ nbFaceUpCards: int
- handWidth: int
- trickWidth: int
+ THIRTEEN_GOAL: int
- thinkingTime: int
- delayTime: int
- scores: int[]
- autoIndexHands: int[]
- isAuto: boolean
- initScore(): void
- calculateScoreEndOfRound(): void
- updateScore(): void
- initScores(): void
- initGame(): void
- setupPlayers(): void
- playGame(): void
- setupPlayerAutoMovements(): void
+ runApp(): String
+ getDelayTime(): int
+ getThinkingTime(): int
+ getSelectedCard(): Card
- applyAutoMovement(hand: Hand, nextMovement: String): Card
- isThirteen(playerIndex: int): boolean
has
1
4
uses
1
Hand
- cardList: List<Card>
+ insert(card: Card, bSort: boolean): void
+ remove(card: Card, bSort: boolean): void
+ getNumberOfCards(): int
+ getCardList(): List<Card>
uses
PublicCardStrategy
+ calculateScore(privateCards: List<Card>, publicCards: List<Card>): int <<coverride>>
PrivateCardStrategy
+ calculateScore(privateCards: List<Card>, publicCards: List<Card>): int <<coverride>>
TwoPrivateTwoPublicCardStrategy
+ calculateScore(privateCards: List<Card>, publicCards: List<Card>): int <<coverride>>
<<interface>>
ScoreStrategy
+ calculateScore(privateCards: List<Card>, publicCards: List<Card>): int
~ getScorePrivateCard(card: Card): int
~ getScorePublicCard(card: Card): int
~ isThirteenFromPossibleValues(possibleValues1: int[], possibleValues2: int[]): boolean
strategies
3
CompositeScoreStrategy
- strategies: List<ScoreStrategy>
+ addStrategy(strategy ScoreStrategy): void
+ calculateScore(privateCards: List<Card>, publicCards: List<Card>): int <<coverride>>
uses
Location
uses
GameLogger
- logResult: StringBuilder
- scores: int[]
+ addCardPlayedToLog(player: int, cards: List<Card>): void
+ addRoundInfoToLog(roundNumber: int): void
+ addEndofRoundToLog(): void
+ addEndofGameToLog(winners: List<Integer>): void
+ getLogResult(): String
has a
1
<<enumeration>>
Rank
+ ACE
+ KING
+ QUEEN
+ JACK
+ TEN
+ NINE
+ EIGHT
+ SEVEN
+ SIX
+ FIVE
+ FOUR
+ THREE
+ TWO
- rankCardValue: int
- scoreValue: int
- possibleSumValues: int[]
+ getRankCardValue(): int
+ getScoreCardValue(): int
+ getPossibleSumValues(): int[]
+ getRankCardLog(): String
uses
1
deals to
1
CardDealer
- properties: Properties
+ getPack(): Hand
+ dealCards(hand: Hand[], nbPlayers: int, nbCardPerPlayer: int, nbSharedCards: int, playingArea: Hand): void
+ dealACardToHand(hand: Hand): Card
+ getRandomCard(hand: Hand): Card
+ getCardFromList(cards: List<Card>, cardName: String): Card
- getRankFromString(cardName: String): Rank
- getSuitFromString(cardName: String): Suit
+ randomEnum<T extends Enum<T>>(clazz: Class<T>, random: Random): T
+ randomCard(list: List<Card>, random: Random): Card
has a
1
uses
Random
uses
uses
<<enumeration>>
Suit
+ SPADES
+ HEARTS
+ DIAMONDS
+ CLUBS
- suitStrongHand: String
- multiplicationFactor: int
+ PUBLIC_CARD_MULTIPLICATION_FACTOR: int
+ getSuitShortHand(): String
+ getMultiplicationFactor(): int
Deck
- cards: List<Card>
+toHand(shuffle: boolean): Hand
+ shuffle(): void
has
52
Card
- rank: Rank
- suitL Suit
+ getRank(): Rank
+ getSuit(): Suit