REPORT OF PROGRAMMING ASSIGNMENT 3

Problem Definition:

In this assignment we are expected to simulate real Monopoly game with object oriented programming. In this assignment we try to play the game in the way "how real Monopoly game plays" and determine the winner.

Solution Approach:

In my Main class, 1 read all the files (list.json, property.json,.) and send them to monopoly class. In my monopoly class 1 create all players, banker and create my gameboard in Gameboard class. In my gameboard class there is an array of 40 square's (square is super class of property, cards, go, jail,.. Etc.). In my gameboard class, commands are determined and make moves.

My UML Class Diagram:

square
- id : int
- square(int)
- getId() : int
- setId(int) : void

property
- name : String
- cost : int
- status : String
- whoOwns : int
- property(int, String, in...)
- getId() : int
- setId(int) : void
- getName() : String
- setName(String) : void
- getCost() : int
- setCost(int) : void
- getStatus() : String
- setStatus(String) : void
- getWhoOwns() : int
- setWhoOwns(int) : void

freeParking
- freeParking(int)
- getId() : int
- setId(int) : void

jail
- jail(int)
- getId() : int
- setId(int) : void

cards
- content : String
- cards(int)
- getContent() : String
- setContent(String) : void
- getId() : int
- setId(int) : void

goToJail
- goToJail(int)
- getId() : int
- setId(int) : void

go
- go(int)
- getId() : int
- setId(int) : void

tax
- tax(int)
- getId() : int
- setId(int) : void

communityChestCards
- content : String
- communityChestCards(int)
- communityChestCards(int, String)
- getContent() : String
- setContent(String) : void

chanceCards
- content : String
- chanceCards(int)
- chanceCards(int, String)
- getContent() : String
- setContent(String) : void

land
- land(int, String, in...)

railroad
- railroad(int, String, in...)

community
- community(int, String, in...)

person
- playerNo : int
- money : int
- beforeBankrupt : int
- person(int, int)
- increaseMoney(int) : void
- decreaseMoney(int) : boolean
- bankrupt() : boolean
- getPlayerNo() : int
- setPlayerNo(int) : void
- getMoney() : int
- setMoney(int) : void
- getBeforeBankrupt() : int
- setBeforeBankrupt(int) : void

gameboard
- squaresOfBoard : square[]
- chanceCards : chanceCards[]
- communityChestCards : communityChestCards[]
- whereCommunityChest : int
- whereChance : int
- addProperty(String[], int) : void
- addChanceCards(String) : void
- addCommunityChestCards(String) : void
- squares() : void
- play(String, int, player, player, banker, boolean) : boolean
- nextMove(int, player, player, banker, int, boolean, boolean) : boolean
- check(int, player, player, banker, int, int, boolean) : boolean
- drawChanceCard(int, player, player, banker, int) : boolean
- drawComChestCard(int, player, player, banker, int) : boolean
- getWhereChance() : int
- setWhereChance(int) : void
- getWhereCommunityChest() : int
- setWhereCommunityChest(int) : void

monopoly
- myGameBoard : gameboard
- Banker : banker
- Player1 : player
- Player2 : player
- addProperty(Iterator, int) : void
- addChanceCard(String) : void
- addCommunityCard(String) : void
- squares() : void
- command(String) : boolean
- bankrupt(String) : void
- show() : void

Main
- main(String[]) : void

player
- properties : String[]
- whereNow : int
- numberofRailroads : int
- jailCount : int
- player(int, int)
- addProperty(String) : void
- getProperties() : String[]
- setProperties(String[]) : void
- getWhereNow() : int
- setWhereNow(int) : void
- getNumberofRailroads() : int
- setNumberofRailroads(int) : void
- getJailCount() : int
- setJailCount(int) : void

banker
- banker(int, int)