Monopoly - dice : Dice Anh Viet Vo | November 15, 2020 - state : State - lost : boolean Monopoly() initialize(Input input) : void + main(String[] args) - run() : void - turn() : void State - handleBlock(Player player, Block block): void buyBlock(Player player, Block block) : void + players : Queue<Player> rentBlock(Player player, Block block): void + board : Board drawCard(Player player, ChanceBlock bl) : void + current : Player - payTax(Player player, TaxBlock bl) : void - organizeFestival(Player player) : void - busGo(Player player) : void Dice additionMoney(Player player, int cost) : int availableAssets(Player player) : Queue<Block> rand : Random sellVal(Queue<Block> props) : int Player val : int - lose(Player player) : void - is\_double : boolean blockSelect(Player player) : Block properties : Queue<Block> purchase(Player player, Block block) : void - playerName : String + Dice() printState() : void money: int + roll() : void position : int + getVal() : int - inJail : boolean + getDouble() : boolean Input Card + setDouble() : void + Player(String playerName) - scanner : Scanner - action : CardAction + addProperty(Block block) : void - value : int + move(int numSpace) : void + Input() : void - travelTo : int + moveTo(int pos) : void + inputString() : String - text : String + position(): int + inputBool() : boolean + enum CardAction = + properties() : Queue<Block> + inputInt() : int {BANK MONEY, MOVE TO} + name() : String + inputPlayer() : Player + getMoney() : int + Card(int a) + excMoney(int money) : void - chance(int a) : void + toJail(): void ChanceBlock + value() : int + sellProps(Block block) : void + travelTo(): int + leaveJail(): void - DECK\_SIZE : int + text() : String + inJail(): boolean deck : Deck + action() : CardAction + getAssets() : int name : String - {nameOfCard}() : void + getHouseVal(PropertyBlock pos : int **JailBlock BusBlock** prop) : int + inputBool(): bool - name : String - name : String + ChanceBlock(int pos) + inputInt(): int - pos : int - pos : int chance(): void Deck - draw() : Card + cards() : Iterable<Card> + BusBlock(int pos) + JailBlock(int pos) - deck : ArrayList<Card> + position() : int + position() : int + position(): int - SIZE : int + name(): String + name(): String + name(): String - current : int + isOwnable() : boolean + isOwnable(): boolean + isOwnable() : boolean + isOwned(): boolean + isOwned() : boolean + isOwned() : boolean + cost(): int + Deck() + cost(): int + cost() : int + purchase(Player player) : void + initialize(Card[] cards) + purchase(Player player) : void + purchase(Player player) : void + rent(): int + drawCard() : Card + rent() : int + rent() : int + setFestival(boolean stt) : void + setFestival(boolean stt) : void + cards() : Iterable<Card> + setFestival(boolean stt) : void + getFestival() : boolean **Board** + getFestival() : boolean + getFestival() : boolean + owner() : Player + owner() : Player + owner() : Player - N : int + toString() : String + toString() : String + toString() : String - board : Block[] - chance : Deck Board(Deck chance) size() : int block(int pos) : Block PropertyBlock + getBoard() : Block[] + makeBlock(int pos) : Block - rent : int + makeTravel() : void oneHouseRent: int **TravelBlock** + {nameOfBlock}(): - twoHouseRent : int **TaxBlock FestivalBlock GoBlock** PropertyBlock/ TravelBlock/ COST : int - threeHouseRent: int BusBlock/ JailBlock/ - others : TravelBlock[] - fourHouseRent : int name : String ChanceBlock/ FestivalBlock/ name : String name : String numOwned : int - hotelRent : int GoBlock/ TaxBlock pos : int pos : int pos : int owner : Player - buyPrice : int - buildPrice : int festivalStatus : boolean + TaxBlock(int pos) + FestivalBlock(int pos) + GoBlock(int pos) - buildings : int owned : boolean + position() : int + tax(int value) : int + position() : int name : String - owned : boolean + position() : int + name(): String + name(): String - festivalStatus : boolean - pos : int + isOwnable() : boolean + name(): String + isOwnable() : boolean - name : String + isOwnable() : boolean + isOwned(): boolean + isOwned() : boolean - pos : int + TravelBlock(String name, int + isOwned(): boolean + cost(): int + cost(): int - owner : Player + purchase(Player player) : void + purchase(Player player) : void + cost(): int + createGroup(TravelBlock a, + purchase(Player player) : void + rent(): int + rent(): int TravelBlock b, TravelBlock c, + PropertyBlock(String name, int pos, int rent, + setFestival(boolean stt) : void + setFestival(boolean stt) : void + rent() : int int oneH, int twoH, int threeH, int fourH, int TravelBlock d): void + setFestival(boolean stt): void + getFestival() : boolean + getFestival() : boolean + updateOwners() : void hotel, int buy, int build) + getFestival(): boolean + owner() : Player + owner() : Player + position() : int + build(int a) : void + owner() : Player + toString() : String + toString(): String + name(): String + numHouses(): int + toString(): String + isOwnable() : boolean + houseCost(): int + isOwned() : boolean + position(): int + cost() : int + name(): String + purchase(Player player) : void + isOwnable() : boolean + rent() : int + isOwned() : boolean + setFestival(boolean stt) : void + cost(): int + getFestival() : boolean + purchase(Player player) : void + owner() : Player + rent() : int + toString() : String + setFestival(boolean stt) : void + getFestival() : boolean + owner() : Player + toString() : String <<interface>> Block position(): int name(): String isOwnable(): boolean isOwned(): boolean cost(): int purchase(Player player) : void rent(): int setFestival(boolean stt) : void getFestival(): boolean owner(): Player toString(): String