Player

String name : Variable containing the name of the player

String symbol : Variable containing the symbol of the player

Player[] players : Variable containing a reference to an array which contains the player objects of the players.

Location[] propertyOwned : Variable containing a reference to an array which contains the location objects of the location which players have bought

Int playerMoney = 1500 : Variable containing the amount of money a player has. Initial value of 1500 as that is the starting amount of money.

Int numOfPropertyOwned : Variable containing the number of properties owned by the player.

Int position = 0 : Variable which keeps track of the position of the player throughout the game

Int jailCounter = 0 : To keep track of the number of times a player has rolled during their time in jail

boolean isInDebt = false : To keep track if the player is in debt – (Has no money but has properties to mortgage.)

Boolean isBankrupt = false : To keep track if the player is currently bankrupt – (No money and properties)

Boolean isInJail = false : To keep track if the player is in jail Boolean getOutOfJailFree = false : Keeps track if the player has a get out of jail free card.

Constructor **public** Player()
Default constructor
public Player (**int** numOfPlayers)
Overloaded constructor to define the array for the number of players numOfPlayers : number of players playing
public Player(String name)
Overloaded constructor method to create instances of players and also store their names in the player variables. Name : player's name
public boolean SetPos(**int** newPos)
Setter Method to set the player position newPos : the new position of the player - Return true when the player is not in jail has successfully moved Return false if player is in jail and cannot move
public void SetSymbol(String symbol)

Setter Method to set the symbols that represents the player

Symbol : player's symbol

- Void return because it sets the variable and has no return here

public void SetMoney(**int** amount)

Setter method to set the money

Amount : player's money

- Void return because it sets the variable and has no return here

public void SetJailCounter(**int** counter)

Setter method to set the jail counter

counter : The number of times a player has rolled in jail, if 3 times execute

rules section where they are forced out of jail for 50 euros

- Void return because it sets the variable and has no return here

public void SetIsInDebt(**boolean** isInDebt)

Setter method to set is the player in debt

isInDebt : sets a false

- Void return because it sets the variable and has no return here

public Player[] GetPlayerArray()

Method to get the array for players

- Return the array of players to the main when it is called so that the instance is saved in a variable there.
- Player array return type is used because it is getting the player array containing all the players in this player class

public Location[] GetPropertyOwned()

Getter method to get owned property

- Return the property owned by the player
- Location array return type is used because it is getting the property owned array in this player class.

public String GetName()

Getter method to get the player's name

- Return player's name
- String return type is used because the variable is a String type

public String GetSymbol()

Getter Method to get the player symbol

- Return player symbols
- String return type is used because the variable is a String type

public int GetMoney()

Getter method to get the money

- Return player's money

- Int return type is used because the variable is an int type

public int GetPos()

Getter method to get the position of the player

- Returns player's position
- Int return type is used because the variable is an int type

public int GetNumOfPropertyOwned()

Getter method to get the number of property owned

- Returns number of property owned
- Int return type is used because the variable is an int type

public boolean GetIsInJail()

Getter method to check if the player is in jail

- Return a false to set a player is not in jail
- Boolean return type is used because the variable is a Boolean type

public boolean GetIsBankrupt()

Getter method to get if player is bankrupt

- Returns variable is bankrupt
- Boolean return type is used because the variable is a Boolean type

public boolean GetIsInDebt()

Getter method to get the variable is in debt, to check is they are currently in debt. If they are, will not allow them to use options 1-3 in game menu.

- Returns variable is in debt
- Boolean return type is used because the variable is a Boolean type

public boolean GetGetOutOfJailFree()

Getter method to get if the player has a get out of jail free card

- Returns variable get out of jail
- Boolean return type is used because the variable is a Boolean type

public int GetJailCounter()

Getter method to get jail counter

- Returns variable jail counter
- Int return type is used because the variable is an int type

public void AddLocationToOwned(Location location)

Method to add bought property to the array property owned

- Uses helper method increase number of property owned, and adds the location parameter into the player location owned array

Location : A location type value which will be added to the property owned array

- Void return type is used here because the actions are done directly in

this method itself and does not require to return anything to any other method.

public void IncreaseNumOfPropertyOwned()

Helper method to increase the count of properties owned by a player

- Adds one to the number of properties owned
- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other method.

public void DecreaseNumOfPropertyOwned()

Method to decrease the count of properties owned by a player

- Minus one to the number of properties owned
- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other method.

public void GoingToJail()

Method to send player to jail

- Sets the variable is in jail to true.
- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other

method

public void ReleaseFromJail()

Method to release player from jail

- Sets the variable is in jail to false.
- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other method.

public void GetOutOfJailFreeCard()

Method to set GetOutOfJailFree to false

- Sets it to false when player sells it, or uses it to get out of jail
- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other method.

public void UseGetoutOfJailFree()

Method to set GetOutOfJailFree to true

- If a player gets it from a card, set it to true, or if a player sells it to them
- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other method.

public void GoneBankrupt()

Method to tell players they have gone bankrupt

- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other method.

public void RollDice(Player[] playerArray, **int** playerCounter, Cards[] commCard, Cards[] chanceCard)

Method for rolling the dice. Takes in community card array, chance card array, current player counter, and array containing players.

- Takes in Community and Chance card array in order to call and transfer these parameters into “check rules” method in rules class after the position of a player is changed.
- Takes in array of players and player counter in order to set player position directly.

playerArray : player array

playerCounter : int that increase to set which player turn it is

commCard : card array

chanceCard : card array

- Void return type is used here because the actions are done directly in this method itself and does not require to return anything to any other

method.