Rules

String euro : A variable which contains the euro symbol from the Displayer class.

Constructor

public Rules()

Default constructor

public void CheckingRules(Player[] playerArray, **int** playerCounter, Cards[] commCards, Cards[] chanceCards, **int** diceSum)

Method which executed all other “Check” helper methods in this class once this method is called. Takes in parameters playerArray, playerCounter, CommCards, ChanceCards, and diceSum which will be passed onto other checking methods through this method.

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

commCard : card array

chanceCard : card array

diceSum : takes in the dice sum

- Void return type because there are multiple types of methods in which to return true or false does not make any sense as there are too many scenarios. A better option than Boolean would be int but then again, it would be far quicker to execute the actions in the helper method themselves.

public void CheckPassGo(Player[] playerArray, **int** playerCounter, **int** diceSum)

Helper method that checks if a player passes go

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

diceSum : takes in the dice sum

- Will check if player passes go through calculations and if they pass go, the current player will receive 200.
- Void type return because the actions are executed in this method.

public void CheckIsInDebt(Player[] playerArray, **int** playerCounter)

Helper method that checks if a is in debt

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

- Will check if players money is less than 0, and they still have properties. If they do, set in debt variable in player class to true.
- Again void return because actions are executed here.

public void GoToJail(Player[] playerArray, **int** playerCounter)

Helper method that sets the player in jail status

- Position is in jail
- Player variable is in jail is now true.

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

- Again void return because actions are executed here.

public void CheckGoToJail(Player[] playerArray, **int** playerCounter)

Helper method which checks if the player position is on jail after a roll

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

- Again void return because actions are executed here.

public void CheckJailCounter(Player[] playerArray, **int** playerCounter)

Helper method which checks if the player has passed three turns in jail

- If passed 3 turns in jail, force them out of jail and have to pay 50 euro fine

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

- Again void return because actions are executed here.

public void CheckBankrupt(Player[] playerArray, **int** playerCounter)

Helper method which checks if the current player is bankrupt

- Accesses current players money and checks if it is less than 0, and also they have no property remaining.

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

- Again void return because actions are executed here.

public void CheckGettingCommOrChance(Player[] playerArray, **int** playerCounter, Cards[] commCards, Cards[] chanceCards)

Helper method that checks if player is on properties that will give out chance and community cards

- If they are, assign them a card from the card array with the randomly generated integer from random objects card.

playerArray : The array of players

playerCounter: Integer which keeps track of which player did the action

commCard : The array for community cards

commCard : The array for chance cards

- Again void return because actions are executed here.

public void CheckTax(Player[] playerArray, **int** playerCounter)

Method that will check the properties that will tax a player

- If they are, method will deduct amount of money from player based on the type of tax – 200 for normal, 400 for super tax

playerArray : player array

playerCounter: Integer which keeps track of which player did the action

commCard : card array

chanceCard : card array

- Again void return because actions are executed here.

public void CheckRent(Player[] playerArray, **int** playerCounter)

Method that will check the location array of every player to see if the current player will be charged rent.

- Searches through all the locations owned by the other players, if it matches the location of current player, charge rent based on the amount of upgrades the location has.

playerArray : player array

playerCounter: Integer which keeps track of which player did the action

- Again void return because actions are executed here.

public boolean CheckWinner(Player[] playerArray, **int** playerCounter)

Method that will check the winner

- If all other players except for the current player is bankrupt, will declare winner

playerArray : player array

playerCounter: Integer which keeps track of which player did the action

- Boolean method here to return true as it is executed in the main and also in the checking method.
- If it is true, will break main loop and declare winner of game.