

Location

Int COLOUR LOCATION = 0: A public final variable which is used in the importing of location from the text file

Int BLANK LOCATION = 1: A public final variable which is used in the importing of location from the text file

Int SPECIAL LOCATION = 2: A public final variable which is used in the importing of location from the text file

Location [] monopolyNorth: A variable named monopolyNorth which holds the reference to the array which contains all the location objects for the North side of the board

Location [] monopolySouth: A variable named monopolySouth which holds the reference to the array which contains all the location objects for the North side of the board

Location [] monopolyEast: A variable named monopolyEast which holds the reference to the array which contains all the location objects for the North side of the board

Location [] monopolyWest: A variable named monopolyWest which holds the reference to the array which contains all the location objects for the North side of the board

String name: Variable containing the name of the location imported from text file

Int price : Variable containing the price of the location which is imported from the text file

String colour : Variable containing the colour of the location which is imported from the text file

Int position : Variable containing the position of the location which is imported from the text file

Consturctor

public Location()

Default constructor

public Location(String name, **int** price, String color, **int** position)

Overloaded constructor for the Colour Location sub class instance.

Name : location name

Price : location price

Colour : location colour

Position : the position of the location

public Location (String name, **int** position)

Overloaded constructor for the Blank Location sub class instance.

Name : location name

Position : the position of the location

public Location(String name, **int** price, **int** position)

Overloaded constructor for the Special Location sub class instance.

Name : location name

Price : location price

Position : the position of the location

public String GetName()

Getter method to get the name of the location

- Return location's name

public int GetPrice()

Getter method to get the price of the location

- Return location's price

public String GetColor()

Getter method to get the colour of a property

- Return a property colours

public Location[] GetMonopolyNorth()

Getter method to get the North side of the board

- Return the location array at North direction of the board

public Location[] GetMonopolySouth()

Getter method to get the South side of the board

- Return the location array at South direction of the board

public Location[] GetMonopolyEast()

Getter method to get the East side of the board

- Return the location array at East direction of the board

public Location[] GetMonopolyWest()

Getter method to get the West side of the board

- Return the location array at West direction of the board

public int GetPosition()

Getter method to get the position

- Returns the position of the location

public void AddLocation(String filename)

Method which takes in the name of the file to add in locations from text file, creates instance of fileaccessor and calls the import location from file method where it places in the arrays as parameters.

Filename : text file name

- Return type is void here because the method will call another method to import location. And since it is a initializing type of method, it does not have to return anything as it imports the locations.

public String Direction(**int** position)

Method to print out its direction

Position : the position of the location

- Return North when position less and equal 10
- Return East when position more then 10 and less then equal to 19
- Return South when position is more then 19 and less then equal to 30
- Return West for the rest of the position

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

Method to buy location

playerArray : player array

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

- Returns a true when a location is successfully bought
- Returns a false when the location is not successfully bought
- Boolean is chosen as the return type here because in the main, if it is true, it ends the method and print out successfully bought. Otherwise the rest of the error handling printing is done within this method

public boolean UpgradingLocation(Player[] playerArray,**int** playerCounter,
int locationToBeUpgraded , String stringColourOfLocationToUpgrade)

Method to upgrade location

playerArray : player array

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

locationToBeUpgraded : the location that will be upgrade

stringColourOfLocationToUpgrade : colour of the location that will be upgraded

- Returns a true when the location is successfully upgraded
Returns a false when the location is not successfully upgraded
- Boolean chosen as the return type because it is either did not upgrade or upgrade, therefore only a true or false value is needed, and thus Boolean is chosen

```
public boolean DowngradingLocation(Player[] playerArray,int playerCounter,  
int locationToBeDowngraded , String stringColourOfLocationToUpgrade)
```

Method to downgrade a location

playerArray : player array

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

locationToBeDowngraded : the location that will be upgrade

stringColourOf LocationToUpgrade : colour of the location that will be upgraded

- Return a true when the location is successfully downgraded
Return a false when the location is not successfully downgraded
- Boolean chosen as the return type because it is either did not upgrade

or upgrade, therefore only a true or false value is needed, and thus

Boolean is chosen

```
public boolean MortgageLocation(Player[] playerArray,int playerCounter, int
locationToBeMortgaged)
```

Method to sell a location

playerArray : player array

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

locationToBeMortgaged : the location that will be sold

- Return a true when the location is successfully sold
Return a false when the location is not successfully sold
- Boolean chosen as the return type because it is either did not upgrade
or upgrade, therefore only a true or false value is needed, and thus
Boolean is chosen