Taoheed King

CMSC 203 SUMMER

Assignment 4 Documentation

# **Documentation**

## **PSUEDO CODE FOR ManagementCompany**

addProperty #2-Adds the property object to the "properties" array.

addProperty#3-Creates a property object with a default plot and adds it to the "properties" array.

addProperty #4- Creates a property object and adds it to the "properties" array.

displayPropertyAtIndex- Displays the information of the property at index i

getMAX\_PROPERTY: Return the MAX\_PROPERTY constant that represent the size of the "properties" array.

maxPropertyRent: This method finds the maximum rent amount.

maxPropertyRentIndex: This method finds the index of the property with the maximum rent amount.

toString: Displays the information of all the properties in the "properties" array.

totalRent: This method accesses each "Property" object within the array "properties" and sums up the property rent and returns the total amount.

## **PSUEDO CODE FOR Property**

Property #1: No-arg Constructor, creates a new object with default values of empty strings, 0 for rent amount, and default Plot

Property #2: Copy Constructor, creates a new object using the information of the object passed to it.

Property #3: Parameterized Constructor, no Plot information provided

Property #4: Constructor, Parameterized constructor

Get City: return city

Get Owner: return owner

Get Property Name: return property name

Get Rent Amount: return rent amount

Set Plot: set the Plot values and return the Plot instance

Set City: set the city

Set Owner: set the owner name

Set Property Name: set the property name

Set Rent Amount: set the rent amount

To String: Prints out the name, city, owner and rent amount for a property. </Text>

## **PSUEDO CODE for Plot**

Plot #1: No-arg Constructor, creates a default Plot with args x=0, y=0, width=1, depth=1

Plot #2: Copy Constructor, creates a new object using the information of the object passed to it.

Plot #3: Parameterized Constructor

Overlaps: determines if this plot overlaps the parameter

Encompasses: determines if this plot encompasses the parameter

Set X: set x value

Get X: return x value

Set Y: set y value

Get Y: return y value

Set Width: set width

Get Width: return width

Set Depth: set depth

Get Depth: return depth

# **UML**

![A screenshot of a cell phone

Description automatically generated]()

***PropertyMgmGui.java with Management info added***

![A screenshot of a social media post

Description automatically generated]()

***PropertyMgmGui.java with property added***

![A screenshot of a social media post

Description automatically generated]()

![A screenshot of a cell phone

Description automatically generated]()

![A screenshot of a social media post

Description automatically generated]()

![A screenshot of a social media post

Description automatically generated]()

![A screenshot of a social media post

Description automatically generated]()

***PropertyMgmGui.java unsuccessful:overlap***

![A screenshot of a cell phone

Description automatically generated]()

![A screenshot of a cell phone

Description automatically generated]()

***PropertyMgmGui.java unsuccessful: Mgmt Co Plot does not encompass Property Plot***

![A screenshot of a social media post

Description automatically generated]()

***Add property information - unsuccessful: too many properties***

![A screenshot of a cell phone

Description automatically generated]()

***Result of “List of Properties” button***

![A screenshot of a cell phone

Description automatically generated]()

![A picture containing screenshot

Description automatically generated]()

***Result of “Max Rent” button***

***![A screenshot of a cell phone

Description automatically generated]()***

***Result of “Total Rent” button***

![A screenshot of a cell phone

Description automatically generated]()

Assignment 4 Check List

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment#\_Moss.zip | **<Yes or No>** | **Yes** |
|  | * FirstInitialLastName\_Assignment#.docx/.pdf | **<Yes or No>** | **Yes** |
|  | * Source java files | **<Yes or No>** | **Yes** |
|  | **Program compiles** | **<Yes or No>** | **Yes** |
|  | **Program runs with desired outputs related to a Test Plan** | **<Yes or No>** | **Yes** |
|  | **Documentation file:** |  | **Yes** |
|  | * Comprehensive Test Plan | **<Yes or No>** | **N/A** |
|  | * Screenshots for each Test case listed in the Test Plan | **<Yes or No>** | **Yes** |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) | **<Yes or No or N/A>** | **Yes** |
|  | * UML Diagram | **<Yes or No >** | **Yes** |
|  | * Algorithms/Pseudocode | **<Yes or No >** | **Yes** |
|  | * Flowchart (if required) | **<Yes or No or N/A>** | **N/A** |
|  | * Lessons Learned | **<Yes or No>** | **Yes** |
|  | * Checklist is completed and included in the Documentation | **<Yes or No>** | Yes |

# **Learning Experience:**

During this assignment I learned how to make overlaps and encompass work in the program. The to string was also something which I got practice on in this assignment. Something which I found easy was the getters and setters and writing the constructors. Something which was difficult came in the MaxPropertyRentIndex method of Management Company and the toString method. In the MaxPropertyRentIndex method, I couldn’t quite get the method to loop correctly initially to get the right indexValue. For the toString method, I could not quite get it formatted correctly. In the end I solved both problems with a tutor.

# **Github Submission:**

![A screenshot of a cell phone

Description automatically generated]()![A screenshot of a social media post

Description automatically generated]()