Psuedocode for Plot.java

* Global Variables
  + x to hold the x value of the plot
  + y to hold the y value of the plot
  + width to hold the width of the plot
  + depth to get the depth of the plot
* Plot() blank constructor
  + Load a default plot on x 0, y 0, width 1, and depth 1
* Plot(Plot) copy constructor
  + Create a plot copying the passed Plot’s properties
* Plot(int, int, int, int) constructor
  + Set the value of x to the passed value
  + Set the value of y to the passed value
  + Set the value of width to the passed value
  + Set the value of depth to the passed value
* setX(int) method
  + Set x to the value passed
* setY(int) method
  + Set y to the value passed
* setWidth(int) method
  + Set width to the value passed
* setDepth(int) method
  + Set depth to the value passed
* getX() method
  + Returns x
* getY() method
  + Returns y
* getWidth() method
  + Returns width
* getDepth() method
  + Returns depth
* overlaps(Plot) method
  + Get the top right points of the original Plot
  + Get the bottom left points on the original Plot
  + Get the top right points of the passed Plot
  + Get the bottom left points on the passed Plot
  + Check the points of Plot 1 against Plot 2
    - If the first Plot’s x is less than the second’s top x, then the rectangle is left or right of the rectangle
    - Then check the ys of both rectangles to determine if it’s above and below
    - If all pass, then return true because it must be somewhere inside or overlapping
    - If all fail, then return false
* encompasses(Plot) method
  + Check if all the points are within the rectangle,
    - If all points are inside, return true
    - If not all points are inside, return false
* toString() method
  + Return the properties of the Plot

Pseudocode for Property.java

* Global variables
  + city to hold the city
  + owner to hold the owner
  + propertyName to hold the property name
  + rentAmount to holdthe rent amount
  + plot to hold the property’s plot
* Property() blank constructor
  + Set city, owner, propertyName, to blank
  + Set rentAmount to 0
  + Set plot to a default plot
* Property(Property) copy constructor
  + Set all the attributes to the attributes of the Property passed.
* Property(String, String, double, String) constructor
  + Set city, owner, propertyName, rentAmount to what was passed
  + Set plot to a default plot
* Property(String, String, double, String, int, int, int, int)
  + Set city, owner, propertyName, rentAmount to what was passed
  + Create a plot with an x, y, width, and depth to what was passed
* getCity() method
  + Returns city
* getOwner() method
  + Returns owner
* getPropertyName() method
  + Returns propertyName
* getRentAmount() method
  + Returns rentAmount
* setCity(String) method
  + Sets city to what was passed
* setOwner(String) method
  + Sets owner to what was passed
* setPropertyName(String) method
  + Sets propertyName to what was passed
* setRentAmount(double) method
  + Sets rentAmount to what was passed
* getPlot() method
  + Returns the Property’s plot
* setPlot(Plot) method
  + Sets the Property’s plot to what was passed
* toString()
  + Returns the attribute

Pseudocode for ManagementCompany.java

* Global variables:
  + MAX\_PROPERTY to hold the max amount of properties allowed per company
  + mgmFeePer to hold the management fee
  + name to hold the company name
  + properties array to hold all properties
  + taxID to hold the tax id of the company
  + MGMT\_WIDTH to hold the max width
  + MGMT\_DEPTH to hold the max depth
  + plot to hold the company plot
* ManagementCompany() blank constructor
  + Set name and taxID to a blank String
  + Set the management fee to 0
  + Make the plot a default one
  + Set the max amount of properties to 5
* ManagementCompany(String, String, double) constructor
  + Set the name, taxID, and mgmFeePer to what was passed
  + Set the plot to a default plot
  + Set the max amount of properties to 5
* ManagementCompany(String, String, double, int, int, int, int) constructor
  + Set the name, taxID, and mgmFeePer to what was passed
  + Create a plot with the x, y, width, and depth passed through
  + Set the max amount of properties to 5
* MangementCompany(ManagementCompany) copy constructor
  + Set the name, taxID, mgmFeePer, plot, and properties from the company passed through.
* addProperty(Property) method
  + Create a variable to hold a temporary property
  + Create a new property based off of the Property passed
  + If it fails, return -2 because it’s null.
  + Create a variable for a counter
  + Loop through all the properties
    - Check if the current property is null
      * If not null, increase the counter.
      * Check if the property overlaps any other properties in the company, if it does, return -4
  + Check if the counter is greater than 4, if it is, return -1 because the array is full
  + Check if the company plot contains the property plot
    - If it does, add the property to the array
    - Return the counter, which is the index
    - If it’s not in the property plot, return -3
* addProperty(String, String, double, String) method
  + Create a variable to hold a temporary property
  + Create a new property based off of the name, city, rent, and owner passed,, will create a default plot
  + If it fails, return -2 because it’s null.
  + Create a variable for a counter
  + Loop through all the properties
    - Check if the current property is null
      * If not null, increase the counter.
      * Check if the property overlaps any other properties in the company, if it does, return -4
  + Check if the counter is greater than 4, if it is, return -1 because the array is full
  + Check if the company plot contains the property plot
    - If it does, add the property to the array
    - Return the counter, which is the index
    - If it’s not in the property plot, return -3
* addProperty(String, String, double, String, int, int, int, int) method
  + Create a variable to hold a temporary property
  + Create a new property based off of the name, city, rent, owner, with a plot with x, y, width, and depth passed
  + If it fails, return -2 because it’s null.
  + Create a variable for a counter
  + Loop through all the properties
    - Check if the current property is null
      * If not null, increase the counter.
      * Check if the property overlaps any other properties in the company, if it does, return -4
  + Check if the counter is greater than 4, if it is, return -1 because the array is full
  + Check if the company plot contains the property plot
    - If it does, add the property to the array
    - Return the counter, which is the index
    - If it’s not in the property plot, return -3
* displayPropertyAtIndex(int) method
  + Return the property information at the index passed
* getMAX\_PROPERTY() method
  + Returns the property amount limit, or MAX\_PROPERTY
* maxRentProp() method
  + Create a variable to hold the max rent amount
  + Create a property to hold the max property
  + Loop through all the properties
    - Check if the property is null
      * If not null, check if the current property rent amount is greater than the max
        + If it is greater than the max, set the max to the current property rent amount
        + Set the max property to the current property
  + Return the max property information
* maxRentPropertyIndex() method
  + Create a variable to hold the max
  + Create a variable to hold the max index
  + Create a variable to hold a counter
  + Loop through all the properties
    - Check if the current property is null
      * If not, check if the current property rent amount is greater than the max
        + If it is, set the max to the current property’s rent amount
        + Set the max index to counter
      * Increase the counter by 1 before it moves onto the next property so that the index matches with the current property
  + Return the max index
* toString() method
  + Create a string to hold the property’s information
  + loop through all the properties
    - If the property is not null, append its information to the String
  + Return the string
* totalRent() method
  + Create a variable sum to hold the sum
  + Loop through all properties
    - Check if the current property is null
      * If not, add the current property’s rent amount to the sum
  + Return the sum
* getPlot() method
  + Return the company’s plot
* setPlot(Plot) method
  + Set the company’s plot to what was passed
* getName() method
  + Return the name of the company
* setName(String) method
  + Set the name of the company

Test Plan:

Run the unit tests provided and make sure they all run. I will create my own unit test for Management Company so that I can test the important methods.