Property.java

private name

private city

private rental

private owner

private plot

toString:

print out property + name + city + rental + owner + plot

[constructors]

Property(p):

copy constructor creates new object based on p (a property instance)

Property(name, city, owner, rental):

creates new property object, passes parameters to respective fields based on name

Property(name, city, rental, owner, x, y, width, depth):

creates new property object, passes parameters to respective fields based on name

plot gets set to x, y, width, and depth

getName:

print name

setName(n):

set name to whatever user enters

getCity:

return city

setCity(c):

set city to whatever user enters

getRental:

print rental

setRental(r):

set rental to whatever user enters

getOwner:

print owner

setOwner(o):

set owner to whatever user enters

getPlot:

print plot

setPlot(p):

set plot to whatever user enters\

Plot.java

private x

private y

private width

private depth

Plot:

creates a plot object, sets all fields to default values

Plot(p):

copy constructor, creates a new Plot object based on p (a plot object)

Plot(x, y, width, depth):

creates a plot object, passes parameters to respective fields based on name

overlaps(p):  
 if this plot has any part overlapping the parameter p (a plot object):

return true

else

return false

encompasses(p):

if this plot contains (p) inside of it:

return true

else:

return false

setX(x):

set x to whatever user enters

getX:

print x

setY(y):

set y to whatever user enters

getY:

print y

setWidth(width):

set width to whatever user enters

getWidth:

print width

setDepth(depth):

set depth to whatever user enters

getDepth:

print depth

toString:

print depth + x + y + width

ManagementCompany.java

private MAX\_PROPERTY = 5

private mgmFeePer

private name

private properties[MAX\_PROPERTIES] - an array of Property objects

private taxID

private MGMT\_WIDTH = 10

private MGMT\_DEPTH = 10

private plot -a plot object

MangementCompany:

constructor that creates a ManagementCompany object and sets all fields to their default values

ManagementCompany(name, taxID, mgmFeePer):

constructor that creates a ManagementComapny object and sets fields to respective values, all other values are set to their default values

creates a plot object with x,y, width, and depth set to 0

ManagementCompany(name, taxID, mgmFeePer, x, y, width, depth):  
 constructor that creates a ManagementCompany object and sets fields to respective

values also creates a plot object with x,y, width, depth values

ManagementCompany(m):

copy constructor that creates another ManagementCompany object based on m (a ManagementCompany object)

addProperty(property):

a = Property

for value in properties:

if a.overlaps(properties[value]):

return -4

if properties[value] == null

properties[value] = a

return value

if a = null:

return -2

if a.encompasses(ManagementCompany.plot):

return -3

return -1

addProperty(name, city, rent, owner):

a = Property(name, city, rent, owner)

for value in properties:

if a.overlaps(properties[value]):

return -4

if properties[value] == null

properties[value] = a

return value

if a = null:

return -2

if a.encompasses(ManagementCompany.plot):

return -3

return -1

addProperty(name, city, rent, owner, x,y , width,depth):

a = Property(name,city,rent,owner,x,y,width,depth)

for value in properties:

if a.overlaps(properties[value]):

return -4

if properties[value] == null

properties[value] = a

return value

if a = null:

return -2

if a.encompasses(ManagementCompany.plot):

return -3

return -1

totalRent:

total

for value in properties:

total +=properties[value].rent()

return total

maxRentProperty:

max = properties[0].rent()

for value in properties:

if max < properties[value].rent():

max = properties[value].rent()

return max

maxRentPropertyIndex:

max = properties[0].rent()

maxIndex = 0

for value in properties:

if max < properties[value].rent():

max = properties[value].rent()

maxIndex = value

return maxIndex

toString:

print “properties for ” + name + “taxID: ” + taxID

for value in properties:

print properties[value].name

print properties[value].city

print properties[value].owner

print properties[value].rent