

CS2ME3 Assignment 1 report

Mingnan Su
macID: sum1

January 31, 2017

Contents

1	My Codes	2
1.1	CircleADT.py	2
1.2	Statistics.py	4
1.3	testCircles.py	5
1.4	Makefile	6
2	Partner Codes	7
2.1	CircleADT.py	7
2.2	Statistics.py	9
3	The result of testing my files	11
4	The result of testing partner's files	12
5	Discussion of test results	13
5.1	What you learned doing the exercise	13
5.2	List any problem I found with:	13
6	how I handled the value of π in my program	14
6.1	Why I made this choice	14
6.2	Is π explicitly expanded in my formula	14

1 My Codes

1.1 CircleADT.py

```
#name: Mingnan Su
#macID: sum1

import math

class CircleT:
    x=0
    y=0
    r=0

    #constructor
    def __init__(self,x,y,r):
        self.x=x
        self.y=y
        self.r=r

    def xcoord(self):
        return self.x

    def ycoord(self):
        return self.y

    def radius(self):
        return self.r

    #return the area of the circle
    def area(self):
        return math.pi*self.r*self.r

    #return the circumference of the circle
    def circumference(self):
        return math.pi*self.r*2

    #a rectangle with left top point at (x0,y0)
    #width=w,height=h
    #determine if the circle is inside the box
    def insideBox(self,x0,y0,w,h):
        r=self.r
        x=self.x
```

```

y=self.y
#limit of x0
posx=x-r
#limit of y0
posy=y+r
#limit of width
limtw=abs(posx-x0)+2*r
#limit of height
limth=abs(posy-y0)+2*r

#check if the box statisfy all the limits
if w>=limtw:
    if h>=limth:
        if x0<=posx:
            if y0>=posy:
                return True
    return False

#determine if another circle is intersected with this one
def intersect(self,c):
    #calculate the sum of two radius
    rds=self.r+c.r

    #calculate the distance between center
    xsq=(self.x-c.x)**2
    ysq=(self.y-c.y)**2
    dist=math.sqrt(xsq+ysq)

    #if they intersected , ditstance between center <= rds
    if dist<=rds:
        return True
    else:
        return False

#scale the radius by k
def scale(self , k):
    self.r=self.r*k

#move the center of circle to (dx,dy)
def translate(self,dx,dy):
    self.x=self.x+dx
    self.y=self.y+dy

```

1.2 Statistics.py

```
#name:Mingnan Su
#macID: sum1

import numpy

def average(lst):
    radius=[]
    for c in lst:
        radius.append(c.r)
    return numpy.average(radius)

def stdDev(lst):
    radius=[]
    for c in lst:
        radius.append(c.r)
    return numpy.std(radius)

def rank(lst):
    radius=[]
    for c in lst:
        radius.append(c.r)
    tuples=list(enumerate(sorted(radius),1))

    #result is stored in ranked
    ranked=[]
    for rd in radius:
        for tp in tuples:
            if tp[1]==rd:
                ranked.append(tp[0])
                break
    return ranked
```

1.3 testCircles.py

```
#name: Mingnan Su  
#macID: sum1
```

```
import CircleADT  
import Statistics
```

```
c1=CircleADT.CircleT(1,4,3)  
c2=CircleADT.CircleT(-3,2,1)  
c3=CircleADT.CircleT(-1,5,2)  
c4=CircleADT.CircleT(4,-2,4)
```

```
#test CircleADT  
print("Testing CircleADT ...")  
print("The area of circle 1 should be around 28.27, ")  
print("    the output is "+str(c1.area()))  
print("The circumference of circle 2 should be around 6.28, ")  
print("    the output is "+str(c2.circumference()))  
print("We have a box, circle 2 should inside the box.")  
print("    the output is "+str(c2.insideBox(-5,3,5,2)))  
print("We have a box, circle 1 should outside the box.")  
print("    the output is "+str(c1.insideBox(2,-3,100,100)))  
print("circle 1 does not intersect circle 2.")  
print("    the output is "+str(c1.intersect(c2)))  
print("circle 1 intersect circle 3.")  
print("    the output is "+str(c1.intersect(c3)))  
print("Scale the radius of circle 1 by k=2.Now radius is 6.")  
c1.scale(2)  
print("    the output is "+str(c1.radius()))  
print("Translate the center of circle 1 by (-1,-1).Now center should be (0,3)  
c1.translate(-1,-1)  
print("    the output is "+str(c1.xcoord())+", "+str(c1.ycoord()))  
  
#test Statisticics  
c1.scale(0.5)  
c1.translate(1,1)  
print("\nTesting Statistics ...")  
clist=[c1,c2,c3,c4]  
print("The average of circles should be 2.5")  
print("    the output is "+str(Statistics.average(clist)))  
print("The standard deviation shoould be around 1.118")  
print("    the output is "+str(Statistics.stdDev(clist)))
```

```
print("The ranked list should be [3,1,2,4]")
print("    the output is "+str(Statistics.rank(clist)))
```

1.4 Makefile

```
test: src/testCircles.py
    python src/testCircles.py

doc: src/CircleADT.py src/Statistics.py src/testCircles.py
    doxygen -g circledoc
    doxygen circledoc
```

[Return to content page.](#)

2 Partner Codes

2.1 CircleADT.py

```
import math
```

```
class CircleT(object):
    ## @brief Creates a CircleT object and sets it's x/y-coordinates and radi
    # @param x The x-coordinate of the center of the circle
    # @param y The y-coordinate of the center of the circle
    # @param r The radius of the circle
    def __init__(self, x, y, r):
        self.x = x
        self.y = y
        self.r = r

    ## @brief Returns the x-coordinate of the given circle object
    # @param self The CircleT object itself
    # @return The x-coordinate of the circle
    def xcoord(self):
        return self.x

    ## @brief Returns the y-coordinate of the given circle object
    # @param self The CircleT object itself
    # @return The y-coordinate of the circle
    def ycoord(self):
        return self.y

    ## @brief Returns the radius of the given circle object
    # @param self The CircleT object itself
    # @return The radius of the circle
    def radius(self):
        return self.r

    ## @brief Returns the area of the given circle object
    # @param self The CircleT object itself
    # @return The area of the circle
    def area(self):
        return math.pi*self.r*self.r
```

```

## @brief Returns the circumference of the given circle object
# @param self The CircleT object itself
# @return The circumference of the circle
def circumference(self):
    return 2*math.pi*self.r

## @brief Determines if the circle is inside a box
# @param self The CircleT object itself
# @param x0 The x-coordinate of the top left corner of the box
# @param y0 The y-coordinate of the top left corner of the box
# @param w The width of the box
# @param h The height of the box
# @return True if the circle is inside the box, and False if otherwise
def insideBox(self, x0, y0, w, h):
    if (self.x > x0 and self.y > y0):
        if (self.x-self.r > x0 and self.x+self.r < x0+h and self.y-self.r < y0+h):
            return True
    else:
        return False

## @brief Determines if the circle intersects with another circle, including the case where they are tangent
# @param self The CircleT object itself
# @param c Another CircleT object
# @return True if the circles intersect, and False if otherwise
def intersect(self, c):
    totalRadius = self.r + c.r
    distance = math.sqrt(math.pow(self.x-c.x,2) + math.pow(self.y-c.y,2))
    if (distance > totalRadius):
        return False
    else:
        return True

## @brief Manipulates the size of the radius
# @param self The CircleT object itself
# @param c A number that the radius will be scaled by
def scale(self, k):
    self.r = self.r*k

## @brief Manipulates the location of the circle
# @param self The CircleT object itself
# @param dx The horizontal shift of the circle
# @param dy The vertical translation of the circle

```



```

def translate(self, dx, dy):
    self.x = self.x + dx
    self.y = self.y + dy

```

2.2 Statistics.py

```

import numpy as np
### @package Statistics
# Calculate blablabla by implementing circleT.

## @brief Calculates the average radius of a list of CircleT objects
# @param circleList A list containing CircleT objects
# @return The average radius of all the CircleT objects in the list
def average(circleList):
    R = np.arange(0)
    for x in circleList:
        R = np.append(R, [x.radius()])
    return np.average(R)

## @brief Calculates the standard deviation of all the radii in a list of CircleT objects
# @param circleList A list containing CircleT objects
# @return The standard deviation of all the radii
def stdDev(circleList):
    R = np.arange(0)
    for x in circleList:
        R = np.append(R, [x.radius()])
    return np.std(R)

## @brief Calculates the descending ranks of CircleT objects based on radius.
# @param circleList A list containing CircleT objects
# @return The descending ranks of CircleT objects based on radius
def rank(circleList):
    R = []

```

```

for x in circleList:
    R.append(x.radius())

Rank = range(1, len(R)+1)
for x in range(len(R)):
    for i in range(len(R)):
        if (R[x] > R[i] and Rank[x] > Rank[i]):
            temp = Rank[x]
            Rank[x] = Rank[i]
            Rank[i] = temp
        if (R[x] < R[i] and Rank[x] < Rank[i]):
            temp = Rank[x]
            Rank[x] = Rank[i]
            Rank[i] = temp
return Rank

```

[Return to content page.](#)

3 The result of testing my files

Testing CircleADT ...

The area of circle 1 should be around 28.27,
the output is 28.2743338823

The circumference of circle 2 should be around 6.28,
the output is 6.28318530718

We have a box, circle 2 should inside the box.
the output is True

We have a box, circle 1 should outside the box.
the output is False

circle 1 does not intersect circle 2.
the output is False

circle 1 intersect circle 3.
the output is True

Scale the radius of circle 1 by k=2.Now radius is 6.
the output is 6

Translate the center of circle 1 by $(-1, -1)$.Now center should be $(0, 3)$
the output is 0,3

Testing Statistics ...

The average of circles should be 2.5
the output is 2.5

The standard deviation shoould be around 1.118
the output is 1.11803398875

The ranked list should be $[3, 1, 2, 4]$
the output is $[3, 1, 2, 4]$

[Return to content page.](#)

4 The result of testing partner's files

Testing CircleADT ...

The area of circle 1 should be around 28.27,
the output is 28.2743338823

The circumference of circle 2 should be around 6.28,
the output is 6.28318530718

We have a box, circle 2 should inside the box.
the output is False

We have a box, circle 1 should outside the box.
the output is False

circle 1 does not intersect circle 2.
the output is False

circle 1 intersect circle 3.
the output is True

Scale the radius of circle 1 by k=2.Now radius is 6.
the output is 6

Translate the center of circle 1 by $(-1, -1)$.Now center should be $(0, 3)$
the output is 0,3

Testing Statistics ...

The average of circles should be 2.5
the output is 2.5

The standard deviation shoould be around 1.118
the output is 1.11803398875

The ranked list should be $[3, 1, 2, 4]$
the output is $[2, 4, 3, 1]$

[Return to content page.](#)

5 Discussion of test results

Test file works everying OK on my files, partially OK on my partner's file.
Functions that not worked out right on my partner's files are:

1. CircleADT.CircleT.insideBox
2. Statistics.rank

5.1 What you learned doing the exercise

I learned many usefull tools by completing this assingment: latex, doxygen, git, etc.

5.2 List any problem I found with:

- My program
I didn't use the comment style which could be detected by doxygen.
- My partner's module
 1. CircleADT.CircleT.insideBox
In my partner's module, in CircleT class, under the function insideBox, he defines:

```
if (self.x>x0 and self.y>y0)
```

Problem occurs at the second part. The y-coordinate of the center of the circle should always place under y0. In another words, self.y is less than y0
 2. Statistics.rank
In my partner's module, he implement bubble sort to solve the problem. But inside the loops, he writes:

```
for x in range (len(R)):  
    for i in range (len(R)):  
        if (R[x] > R[i] and Rank[x] > Rank[i])
```

He should only compare the elements in R, and swap the elements in Rank, instead of compare two lists.
- the specification of the modules
We might can create another ADT called RectT, which defines a rectangular. Then function insideBox becomes

```
def insideBox (rectT)
```

[Return to content page.](#)

6 how I handled the value of π in my program

I import a python package math and use math.pi to generate a π value.

6.1 Why I made this choice

In this way, I think it's more accurate than I manually type in the value (3.14159)

6.2 Is π explicitly expanded in my formula

When I use a pi value, I just simply replace it with "math.pi".