**Object-Oriented Programming**

**webshapes**

In this assignment, you will create a tool that will

* download an xml file describing shapes from a website
* parse the xml file to determine the shapes contained in it
* draw the shapes using Turtle Graphics

The xml file is located at http://cs.lewisu.edu/~klumpra/2016Summer/shapes.xml.

Here are its contents:

<shapes>

<line>

<x1>-200</x1>

<y1>100</y1>

<x2>-125</x2>

<y2>75</y2>

</line>

<line>

<x1>50</x1>

<y1>-75</y1>

<x2>75</x2>

<y2>25</y2>

</line>

<circle>

<radius>25</radius>

<x>15</x>

<y>100</y>

</circle>

<rectangle>

<length>40</length>

<width>70</width>

<x>-20</x>

<y>120</y>

</rectangle>

<circle>

<radius>50</radius>

<x>-50</x>

<y>-50</y>

</circle>

<regpoly>

<x>200</x>

<y>-100</y>

<length>25</length>

<sides>5</sides>

</regpoly>

<face>

<length>120</length>

<width>40</width>

<x>-50</x>

<y>50</y>

</face>

<house>

<length>70</length>

<width>120</width>

<x>-130</x>

<y>-120</y>

<roof>50</roof>

</house>

<banner>

<text>Think Spring</text>

<x>100</x>

<y>150</y>

</banner>

</shapes>

There are seven types of drawings defined in this file: circles, rectangles, regular polygons, lines, banner, faces, and houses. Regular polygons are shapes for which all sides are of equal length and all angles are of equal size. A banner is a string of text that is to be drawn at a particular location on the screen. A house is basically a rectangle with a pointy roof.

Your program will ask the user for the url of this file online as well as the name of the file to which to save it on your local machine. Once you specify these two pieces of information, your program will download the file from the remote site and save it locally. It will then go through this file to determine what shapes it needs to draw. Once it has determined that, it will draw the shapes using Turtle Graphics.

You are to create a single source file called webshapes\_yourname.py. In that file, you must include the following classes:

* Turtle\_Draw\_Shape\_Controller, which is able to draw the seven drawing components.
* Web\_Retriever, which features a function called download that takes in a url and a local file name and will download the resource at the url and save it locally to that local file name
* Shape, the superclass for all shapes, which we developed in class
* Circle, a subclass of Shape, which we developed in class
* Rectangle, a subclass of Shape, which we developed in class.
* Face, a subclass of Rectangle, which we developed in class.
* Regular\_Polygon, a subclass of Shape, which must have members length and sides implemented as Python properties.
* Line, a subclass of Shape, which must have additional members x2 and y2 implemented as properties,
* Banner, a subclass of Shape designed to hold text, which must have an additional member called text implemented as a property. The inherited properties x and y define where the starting corner of the text will be located.
* House, a subclass of Rectangle, which must have an additional member called roof\_height implemented as a property.
* Shape\_XML\_Parser, which has a function called parse that returns a list of Shape objects that were read from the file whose name is passed to the function.

The way these classes fit together is as follows:

* You use Web\_Retriever to retrieve the xml file from the web and save it locally.
* You use Shape\_XML\_Parser to read the file and return a list of shapes to be drawn.
* You use Turtle\_Draw\_Shape\_Controller to draw the shapes using Turtle Graphics.
* You use Shape\_IO\_Controller to write the shapes to the screen.

The Shape\_XML\_Parser class requires some effort to write. Here is the start of it, which comes directly from the textbook and shows how to parse a Circle from the xml file:

class Shape\_XML\_Parser:

def parse(self,fname):

file\_var = open(fname,"r")

text = ""

for line in file\_var:

text = text + line.strip()

shapes\_list = []

pos = text.find("<circle>")

while pos >= 0:

endpos = text.find("</circle>",pos+1)

tagbeg = text.find("<x>",pos+1)

tagend = text.find("</x>",tagbeg+1)

x = float(text[tagbeg+3:tagend])

tagbeg = text.find("<y>",pos+1)

tagend = text.find("</y>",tagbeg+1)

y = float(text[tagbeg+3:tagend])

tagbeg = text.find("<radius>",pos+1)

tagend = text.find("</radius>",tagbeg+1)

radius = float(text[tagbeg+8:tagend])

cir = shapes.Circle(radius,x,y)

shapes\_list.append(cir)

pos = text.find("<circle>",endpos+9)

Here’s how your program should operate:

Enter url of shapes file: http://cs.lewisu.edu/~klumpra/2016Fall/shapes.xml

Enter name of file to save locally: junk.xml

These are the shapes that were read:

circle 15 100 25

circle -50 -50 50

rect -20 120 70 40

regpoly 200 -100 5 25

line -200 100 -125 75

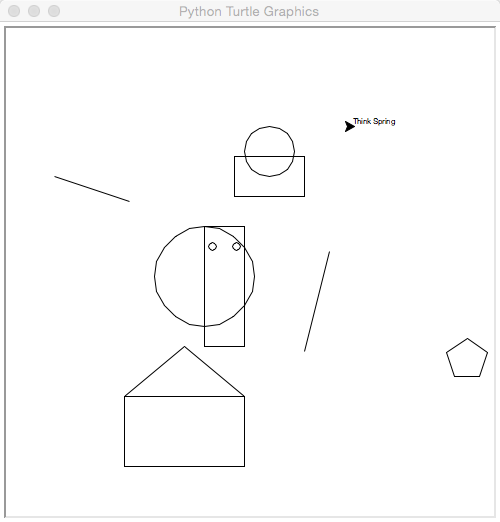
line 50 -75 75 25

house -130 -120 120 70 50

face -50 50 40 120

banner 100 150 Think Spring

In addition to printing this output, it will draw the following arrangement of shapes:



Here is how I will grade your work:

It fails to compile or crashes. -15 points

You define the Regular\_Polygon class correctly, 3 points

including its various member functions and properties

You define the Web\_Retriever class and its 3 points

download function correctly, and it successfully

downloads the XML file

You define the Shape\_XML\_Parser class so 7 points

that it returns a list of Circle, Rectangle,

Regular\_Polygon, Line, Face, Banner, and House

objects correctly

You expand the Turtle\_Shape\_Controller class’ 7 points

draw function to draw Regular\_Polygon, Line,

Banner, and House objects, and you use the class

to actually draw them

Your program lists the text representations 3 points

of the various shapes using Shape\_IO\_Controller

Your program is well-commented 1 point

Your program is submitted to Blackboard with 1 point

the correct name

So, this project is worth 25 points.

Please let me know if you have questions.