Nicks Neon Objects

CIS 261

Prof. Collard

2/3/2020

Rodney Breslin

Problem Description

Create a program that will have a super class and subclasses. Then create a test program to use the various methods within those objects to create a program that will take the perimeter of multiple shapes and add them.

Algorithm

Step one: Create the Super Class Geometric object

Step Two: Create subclasses Triangle, Circle, Rectangle, and Square using extends Geometric Object

Step Three: Make sure circle class has a method for circumference and diameter

Step Four: For all other methods, make sure each one has a getPerimeter method. For the rectangle the perimeter method will add the base times height and multiply it by two. For the square it will do the same calculation but for height+height times 2. Triangle will add three sides for the perimeter

Step Five: create the test program by importing java arraylists and scanner.

Step Six: Declare a Boolean named continueProgram and set it to true

Step seven: Declare 4 new arraylists that use the Circle, Square, Rectangle, and Triangle objects. Name them circ, sq, re, and tri

Step eight: Create a while loop that loops when continueProgram is true

Step nine: Declare a new scanner sc and make it equal to the String decision. This will be the users string input

Step Ten: Create a new scanner called input. This will be used for double values

Step Eleven: Declare doubles rad, height, width, side1, side2, and side3. These will be used later for creating new objects

Step Twelve: create 5 if statements that will allow the user to choose which shape they want.

Step Thirteen: If User presses C, create a new circle object with given diameter and add it to circ ArrayLIst. If user presses S create a new square object and add it to sq ArrayList. If user types R create a new Rectangle object and put it in sq ArrayList. If the user types T create a new triangle object and put it in arraylist tri.

If the user presses Q end the while loop by setting Boolean continueProgram to false.

Step Fourteen: After the while loop, print out the list of Shapes.

Step Fifteen: Create a new double sum which will total all the perimeters of the objects.

Step Sixteen: Create new if statements that check if each object array is not equal to zero. While not equal to zero, there is a for loop that will print the contents of each arrayList.

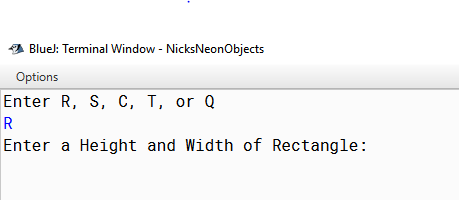
Step Seventeen: Display to the user the total

Test Set Design/Sample Output

Test Set #1 ‘R’

Expected output: Should display prompt for Rectangle

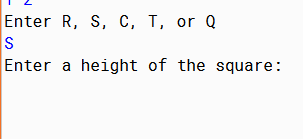
Actual Output:



Test Set #2 ‘S’

Expected output: Should display prompt for square

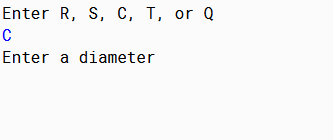
Actual Output:



Test Set #3 ‘C’

Expected output: Should display prompt for circle

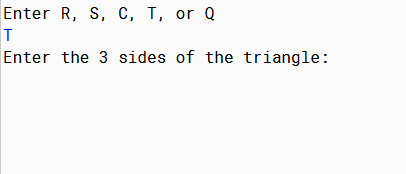
Actual Output:



Test Set #4 ‘T’

Expected output: Should display prompt for triangle

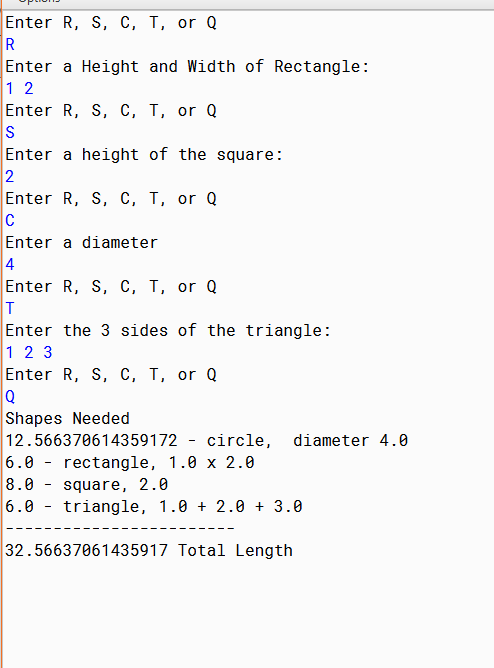
Actual Output:



Test Set #5 ‘Q’

Expected output: Should quit the main loop and calculate total of all shapes

Actual Output:



Test Set #6 ‘R 6.5 12

C 12’

Expected output: Should display a rectangle and circle in the shapes needed list, then print the total length as 74

Actual Output:

