Osama Abdo modhish

Calculat shapes Algorithm

1. Define a `Shape` class as the base class for different shapes. It contains two empty methods, `calculate_area` and `calculate_perimeter`, which will be implemented by subclasses.

2. Define a `Circle` class that inherits from `Shape`. It takes the `radius` as a parameter in its constructor and stores it as an instance variable.

3. Implement the `calculate_area` method in the `Circle` class. It calculates the area of the circle using the formula `pi * radius^2` and returns the result.

4. Implement the `calculate_perimeter` method in the `Circle` class. It calculates the perimeter of the circle using the formula `2 * pi * radius` and returns the result.

5. Define a `Triangle` class that inherits from `Shape`. It takes the lengths of the three sides (`side1`, `side2`, and `side3`) as parameters in its constructor and stores them as instance variables.

6. Implement the `calculate_area` method in the `Triangle` class. It calculates the area of the triangle using Heron's formula: `sqrt(s * (s - side1) * (s - side2) * (s - side3))`, where `s` is the semiperimeter of the triangle (`(side1 + side2 + side3) / 2`), and return the result.

7. Implement the `calculate_perimeter` method in the `Triangle` class. It calculates the perimeter of the triangle by summing the lengths of all three sides (`side1 + side2 + side3`) and returns the result.

8. Define a `Rectangle` class that inherits from `Shape`. It takes the `length` and `width` as parameters in its constructor and stores them as instance variables.

9. Implement the `calculate_area` method in the `Rectangle` class. It calculates the area of the rectangle using the formula `length * width` and returns the result.

10. Implement the `calculate_perimeter` method in the `Rectangle` class. It calculates the perimeter of the rectangle using the formula `2 * (length + width)` and returns the result.

11. Prompt the user to choose the type of shape they want to calculate the area and perimeter for (circle, triangle, or rectangle).

12. Based on the user's input, collect the required inputs (e.g., radius, side lengths) from the user.

13. Create an instance of the corresponding shape class based on the user's choice and provided inputs.

14. Calculate and print the area and perimeter of the chosen shape by calling the respective methods on the shape object.

15. If the user's input doesn't match any of the available shape options, print an error message indicating an incorrect shape type.