NAME : Sarabudheen Amirudeen **26/07/2023**

EMP.ID : 11957

PROBLEM SOLVING FOR CALCULATE AREA AND PERIMETER FOR VARIOUS SHAPES.

W3H :

PROBLEM STATEMENT : DEFINE A APPLICATION TO FIND THE AREA( ) AND PERIMETRER( ) OF VARIOUS SHAPES

WHAT? 1

1) What are the shapes we have to consider?
ANS : Circle,Rectangle,Square and Triangle

2) What are the perimeters we have to consider?
ANS : Length,Breadth,Side and Radius.

3) What are the formulas we have to use to calculate area?
ANS : **Circle** : PI * R ^ 2,**Rectangle** : Length * Breadth
Square : Side * Side,**Triangle** : 1/2 * Breadth *Lenght

4) What are the formulas we have to use to calculate perimeter?
ANS : **Circle** : 2 * PI * Radius,**Rectangle** : 2 * (Lenght + Breadth)
Square : 4 * Side,**Triangle** : Side1 +Side2 + Side3

5)Any Predefined values are required?
ANS : Yes, **PI** = 3.14

HOW? 2

1) Using Single class read the required inputs and calculate area and perimeter, And display the result.

2) Using Single class and overloading calculate area and perimeter, And display the result.

3) Using different class for various shapes and calculate area and perimeter, And display the result.

4) Using different class for various shapes and inherit the common properties from a class called "Shapes" and calculatearea and perimeter, And display the result.

5) Using different class for various shapes and inherit the common properties from a abstract class called "Shapes" and calculate area and perimeter, And display the result.

6) Using different class for various shapes and inherit the common properties from a abstract class called "Shapes" and implemented the interface called "ShapePlan" and calculate area and perimeter, And display the result.

WHY? 3

6) Using different class for various shapes and inherit the common properties from a abstract class called "Shapes" and implemented the interface called "ShapePlan" and calculate area and perimeter, And display the result.

Reasons :

1.We can separate the common properties.
2. We can only declare in interface (Secure).
3. We can declare and also can define in abstract.
4. We can achieve 100% abstraction in interface.
5. Code reusability.
6. We can make a plan by using interface.
7. Code flexibility (Updation).

WHY NOT? 4

4) Using different class for various shapes and inherit the common properties from a class called "Shapes" and calculatearea and perimeter, And display the result.

5) Using different class for various shapes and inherit the common properties from a abstract class called "Shapes" and calculate area and perimeter, And display the result.

Reasons:

1. Compare than this 2 that one is more secure.
2. This 2 is come without interface.
3. We can't achieve 100% abstraction.
4. The 2nd one is partially completed.
5. When we compare to that code flexibility is less efficient in this 2.
6. We can't plan in this 2 solutions because of absence of

ALGORITHM :

Step 01 : Start
Step 02 : Get the Requirements From the client (Shapes and Parameters).
Step 03 : Create an interface for Shape_Plan.
Step 04 : Declare the Methods inside the interface (Area and Perimeter).
Step 06 : Define a PI value inside the Shape_Plan [As Constant].
Step 07 : Create an Abstract class named as Shape and implements with Shape_Plan.
Step 08 : Declare the parameters.
Step 09 : Create classes for various shapes (Circle, Rectangle, Square, and Triangle), extend with shape class (abstract class) and define the Methods .
Step 10 : Apply the correct formula for the shapes.
Step 11 : Call the all shapes area and Perimeter methods by using objects.
Step 12: Display the result
Step 13 : Stop

CODE :

```
package com.sara.day25;
interface ShapePlan {
	void area();
	void perimeter();
	double PI = 3.14;
}
abstract class Shape implements ShapePlan {
	public abstract void area();
	public abstract void perimeter();
}
class Circle extends Shape {
	int radius;
	Circle(int radius) {
		this.radius = radius;
	}
	public void area() {
		double area = PI * (radius * radius);
```

```
            System.out.println("The area of circle is: " + area);
        }
        public void perimeter() {
            double perimeter = 2 * (PI * radius);
            System.out.println("The perimeter of circle is: " + perimeter);
        }
}
class Square extends Shape {
        int side;
        Square(int side) {
            this.side = side;
        }
        public void area() {
            double area = side * side;
            System.out.println("The area of square is: " + area);
        }
        public void perimeter() {
            double perimeter = 4 * side;
            System.out.println("The perimeter of square is: " + perimeter);
        }
}
class Rectangle extends Shape {
        int length;
        int breadth;
        Rectangle(int length, int breadth) {
            this.length = length;
            this.breadth = breadth;
        }
        public void area() {
            double area = length * breadth;
            System.out.println("The area of rectangle is: " + area);
        }
        public void perimeter() {
            double perimeter = 2 * (length * breadth);
            System.out.println("The perimeter of rectangle is: " + perimeter);
```

```
        }
}
class Triangle extends Shape {
        int length;
        int breadth;
        int side1, side2, side3;
        Triangle(int length, int breadth, int side1, int side2, int side3) {
                this.length = length;
                this.breadth = breadth;
                this.side1 = side1;
                this.side2 = side2;
                this.side3 = side3;
        }
        public void area() {
                double area = (length * breadth) / 2;
                System.out.println("The area of triangle is: " + area);
        }
        public void perimeter() {
                double perimeter = side1 + side2 + side3;
                System.out.println("The perimeter of triangle is: " + perimeter);
        }
}
public class Problem_Solving_Ex01 {
        public static void main(String[] args) {
                Circle circle = new Circle(8);
                circle.area();
                circle.perimeter();
                Square square = new Square(12);
                square.area();
                square.perimeter();
                Rectangle rect = new Rectangle(8, 10);
                rect.area();
                rect.perimeter();
                Triangle triangle = new Triangle(6, 8, 12, 14, 16);
                triangle.area();
```

```
            triangle.perimeter();
        }
}
```