**Problem Statement: Create an application that calculates the area() and perimeter() of various shapes (for example, circle, triangle, rectangle, and square).**

|  | **What ? 01** | **How ? 02** |
| --- | --- | --- |
|  | 1). What are the shapes we have to consider ?  **Ans:** Circle,Rectangle, Square and Triangle  2). What are the parameters we have to consider ?  **Ans:** Length, Breadth, Side, Radius  3). What are the formulas we have to use to calculate area and perimeter ?  **Ans: Circle:** PI\*Radius^2, **Rectangle:** Length\*Breadth  **Square:** Side\*Side, **Triangle:** (Breadth\* Length) / 2  **Circle Perimeter:** 2\*PI\*Radius, **Rectangle Perimeter:** 2\*(Breadth+Length)  **Square Perimeter:** 4\*Side , **Triangle Perimeter:** Side1+Side2+Side3  4). Any predefined values are required ?  **Ans:** PI=**3.14** | 1). Using Single Class read all the required inputs and calculate area  and perimeter.And display the result.  2). Using Single class and method overloading calculate area and  perimeter and display the result.  3). Using different classes for various shapes and calculate area and  perimeter and display the result.  4). Using different classes for various shapes and inherit the common  properties from a class called "Shape" and calculate area and perimeter  and display the result.  5).Using different classes for various shapes and inherit the common  properties from a abstract class called "Shape" and calculate area and  perimeter and display the result.  6). Using different classes for various shapes and Inherit the common  properties from a abstract class called "Shape" and implements the  interface called "ShapePlan" and calculate area and perimeter and  display the result. |
|  |
|  |
|  |
|  |
|  |
|  | **Why ? 03** | **Why Not ? 04** |
|  | 6. Using different classes for various shapes and Inherit the common  properties from a abstract class called "Shape" and implements the interface  called "ShapePlan" and calculate area and perimeter and display the result.  **Ans:** 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). | 4. Using different classes for various shapes and inherit the common  properties from a class called "Shape" and calculate area and perimeter  and display the result.  5. Using different classes for various shapes and inherit the common  properties from a abstract class called "Shape" and calculate area and  perimeter and display the result.  **Ans:** 1. Compare than this 2 that 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  interface. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**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.mahesh.PS\_Day25;

**interface** Shape\_Plan {

**double** ***PI*** = 3.14;

**void** area();

**void** perimeter();

}

**abstract** **class** Shape **implements** Shape\_Plan {

**public** **abstract** **void** area();

**public** **abstract** **void** perimeter();

}

**class** Circle\_Shape **extends** Shape {

**double** radius;

Circle\_Shape(**double** radius) {

**this**.radius = radius;

}

@Override

**public** **void** area() {

System.***out***.println("Area of Circle : " + ***PI*** \* radius \* radius);

}

@Override

**public** **void** perimeter() {

System.***out***.println("Perimeter of Circle : " + 2 \* ***PI*** \* radius + "\n");

}

}

**class** Rectangle\_Shape **extends** Shape {

**double** length, breadth;

Rectangle\_Shape(**double** length, **double** breadth) {

**this**.length = length;

**this**.breadth = breadth;

}

@Override

**public** **void** area() {

System.***out***.println("Area of Rectangle : " + length \* breadth);

}

@Override

**public** **void** perimeter() {

System.***out***.println("Perimeter of Rectangle : " + 2 \* (breadth + length) + "\n");

}

}

**class** Square\_Shape **extends** Shape {

**double** S\_side;

Square\_Shape(**double** S\_side) {

**this**.S\_side = S\_side;

}

@Override

**public** **void** area() {

System.***out***.println("Area of Square : " + S\_side \* S\_side);

}

@Override

**public** **void** perimeter() {

System.***out***.println("Perimeter of Square : " + 4 \* (S\_side) + "\n");

}

}

**class** Triangle\_Shape **extends** Shape {

**double** T1\_Side, T2\_Side, T3\_Side, length, breadth;

Triangle\_Shape(**double** T1\_Side, **double** T2\_Side, **double** T3\_Side, **double** length, **double** breadth) {

**this**.T1\_Side = T1\_Side;

**this**.T2\_Side = T2\_Side;

**this**.T3\_Side = T3\_Side;

**this**.length = length;

**this**.breadth = breadth;

}

**public** **void** area() {

System.***out***.println("Area of Triangle : " + (breadth \* length) / 2);

}

**public** **void** perimeter() {

System.***out***.println("Perimeter of Square : " + T1\_Side + T2\_Side + T3\_Side);

}

}

**public** **class** Area\_of\_Shapes {

**public** **static** **void** main(String[] args) {

Circle\_Shape Ci = **new** Circle\_Shape(4.2);

Ci.area();

Ci.perimeter();

Rectangle\_Shape Re = **new** Rectangle\_Shape(3, 5);

Re.area();

Re.perimeter();

Square\_Shape Sq = **new** Square\_Shape(8);

Sq.area();

Sq.perimeter();

Triangle\_Shape Tr = **new** Triangle\_Shape(2, 4, 5, 6, 5);

Tr.area();

Tr.perimeter();

}

}

**o/p:**

Area of Circle : 55.3896

Perimeter of Circle : 26.376

Area of Rectangle : 15.0

Perimeter of Rectangle : 16.0

Area of Square : 64.0

Perimeter of Square : 32.0

Area of Triangle : 15.0

Perimeter of Square : 2.04.05.0