| **Problem Statement : Define a Application to find the area() and Perimeter() of**  **various shapes [Ex : Circle,Triangle,Rectangle and Square]** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|
| **What ? 1** | | | |  | **How ? 2** | | | |
| 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 formula we have to use to  calculate area and perimeter?  **Ans** :(Area) **Circle** : PI\*R^2, **Rectangle** : Length\*Breadth,  **Square** : Side \* Side ,**Triangle** : 1/2\*Breadth\*Height  : (Perimeter)**circle** : 2\*PI\*R, **Rectangle** : 2\*(l+b),  **Square** : 2\*side, **Triangle** : A+B+C.  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 results.  2. Using single class and method overloading calculate area ,  perimeter and display the result.  3. Using different classes for various shapes and calculating area ,  perimeter and display the results.  4.Using different classes for various shapes and inherit the common  properties from a class called "Shape" and calculate area ,  parameter and display the result.  5. Using different classes for various shapes and inherit the  commonproperties from a class called "Shape" ,and  calculate the area ,perimeter and display the result.  6. Using different classes for various shapes and inherits the  common properties from a abstract class called "Shape" and  implements an interface called "ShapePlan" and calculate area  and perimeter.and display the result | | | |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|  | | | | | | | | |
| 6. Using different classes for various shapes and inherits the  common properties from a abstract class called "Shape" and  implements an interface called "ShapePlan" and calculate area  and perimeter.and display the result  Reason:  1)Code reusability by using inheritance  2)Using interface we can achieve full abstraction  3)Using Interface we can obtain multiple inheritance  4)We can declare and also define in abstract.  5)We can declare in the interface (secure). | | | |  | 1. Compared to this 2 that is more secure.  2. This 2 comes without an 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. | | | |
|
|
|
|
|
|
|
|
|
| **Why ? 3** | | | |  | **Why Not ? 4** | | | |

**Algorithm :**

Step1 : Start.

Step2 : Get the inputs from the clients(Shape and Parameters).

Step3 : Create an Interface class named Shape\_Plan.

Step4 : Define a method perimeter and area and a variable PI = 3.14 in the interface.

Step5 : Make an Abstract class named Shapes and implement Shape\_Plan.

Step6 : Declare variables(Length, Breadth, Side, Radius).

Step7 : Inherit the abstract class and override the methods.

Step8 : Give the correct formulas inside the methods.

Step9 : Stop.

**Program :**

**package** com.matthew.day15;

**interface** ShapePlan{

**void** area();

**void** perimeter();

**float** ***pi*** = 3.14f;

}

**abstract** **class** Shape **implements** ShapePlan{

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

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

}

**class** Circle **extends** Shape{

**float** radius;

Circle(**float** radius){

**this**.radius = radius;

}

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

System.***out***.println("Area : "+(***pi***\*radius\*radius));

}

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

System.***out***.println("Perimeter : "+(2\****pi***\*radius));

}

}

**class** Square **extends** Shape{

**float** side;

Square(**float** side){

**this**.side = side;

}

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

System.***out***.println("Area : "+(side\*side));

}

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

System.***out***.println("Perimeter : "+(2\*side));

}

}

**class** Rectangle **extends** Shape{

**float** length,breadth;

Rectangle(**float** length,**float** breadth){

**this**.length = length;

**this**.breadth = breadth;

}

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

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

}

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

System.***out***.println("perimeter : "+(2\*(length+breadth)));

}

}

**class** Triangle **extends** Shape{

**float** base,height,length;

Triangle(**float** base,**float** height,**float** length){

**this**.base = base;

**this**.height = height;

**this**.length = length;

}

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

System.***out***.println("Area : "+(0.5\*(base\*height)));

}

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

System.***out***.println("perimeter : "+(length+base+height));

}

}

**public** **class** formula {

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

Circle circle = **new** Circle(25);

Square square = **new** Square(20);

Rectangle rectangle = **new** Rectangle(10.34f,20.23f);

Triangle triangle = **new** Triangle(12,13,14);

System.***out***.println("Circle");

circle.area();

circle.perimeter();

System.***out***.println("Square");

square.area();

square.perimeter();

System.***out***.println("Rectangle");

rectangle.area();

rectangle.perimeter();

System.***out***.println("Triangle");

triangle.area();

triangle.perimeter();

}

}