Name of the Course : Java SE8 Developer Bootcamp

Level : Easy

Tool Stack : Java 8

Problem Statement : Provide a code solution to print area of a given shapes [Rectangle , Square , triangle] .

Description :

Create classes Shape class attribute area(double) with methods printArea() .

|  |  |
| --- | --- |
| area | double( round to 2 decimal places) |
| abstract void getInput() |  |
| printArea() | method to print the area. |
|  |  |

Create class Square , Rectangle, Triangle extending Shape and overriding printArea().

Create a class ShapeMain class with attribute shape of type Shape.

get input from user for what type of object to create and get the input and display the area.

Assign the shape object to different child object and print the area.

Sample Input :

Enter the object to create :

1.Square

2.Rectangle

3.Triangle

1

Provide the side of square :

5

Area is :25.0

Code:

**import** java.text.DecimalFormat;

**public** **abstract** **class** Shape {

**double** area;

**abstract** **double** getArea();

}

import java.text.DecimalFormat;

import java.util.Scanner;

public class ShapeMain {

static Shape s;

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Shape to create :");

System.out.println("1.Square");

System.out.println("2.Rectangle");

System.out.println("3.Triangle");

int type = Integer.parseInt(sc.nextLine());

switch (type) {

case 1:System.out.println("Provide side of square");

int side= Integer.parseInt(sc.nextLine());

s= new Square (side);

System.out.println("Area is " +s.getArea());

break;

case 2:System.out.println("Provide length of rectangle");

int length= Integer.parseInt(sc.nextLine());

System.out.println("Provide breadth of rectangle");

int breadth= Integer.parseInt(sc.nextLine());

s= new Rectangle (length,breadth);

System.out.println("Area is " +s.getArea());

break;

case 3 : System.out.println("Provide base of Triangle");

int base= Integer.parseInt(sc.nextLine());

System.out.println("Provide height of Triangle");

int height= Integer.parseInt(sc.nextLine());

s= new Triangle(base,height);

System.out.println("Area is " +s.getArea());

break;

default:

break;

}

}

}

import java.text.DecimalFormat;

public class Square extends Shape{

int side;

public Square(int side) {

super();

this.side = side;

}

@Override

public double getArea() {

DecimalFormat df= new DecimalFormat("0.00");

return Double.parseDouble(df.format((double)(side\* side)));

}

}

public class Rectangle extends Shape{

int length;

int breadth;

public Rectangle(int length, int breadth) {

super();

this.length = length;

this.breadth = breadth;

}

@Override

public double getArea() {

return (double)(length\* breadth);

}

}

import java.text.DecimalFormat;

public class Triangle extends Shape{

int base;

int height;

public Triangle(int base, int height) {

super();

this.base = base;

this.height = height;

}

@Override

public double getArea() {

DecimalFormat df= new DecimalFormat("0.00");

return Double.parseDouble(df.format((0.5d)\*(base\* height)));

}

}

Junit Code:

import static org.junit.Assert.assertEquals;

import org.junit.Test;

import handson.Rectangle;

import handson.Square;

import handson.Triangle;

public class TestShape {

@SuppressWarnings("deprecation")

@Test

public void testFibonacci() {

assertEquals(25,new Square(5).getArea(),0);

assertEquals(30 ,new Rectangle(6,5).getArea(),0);

assertEquals(40.0, new Triangle(10,8).getArea(),0);

assertEquals(952.0, new Triangle(56,34).getArea(),0);

}

}

Test Data 1:

Enter the Shape to create :

1.Square

2.Rectangle

3.Triangle

1

Provide the side of square

5

Area is :25

Test Data 2:

Enter the Shape to create :

1.Square

2.Rectangle

3.Triangle

3

Provide the base of triangle

10

Provide the height of triangle

34

Area is :170