|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  **lab - Mid** | **Subject:**  **object oriented programming**  **submitted by:**  **Daoud Hussain**  (Sp21-bcs-102)  **Class:**  **bcs-3b**  **submitted to:**  **mam saneeha amir**  **date of submission:**  **May 16 , 2022** |

Class Geometric Object

public abstract class GeometricObject {

protected boolean isFilled;

protected String color;

protected double thickness;

//Full-Argument Constructor

public GeometricObject(boolean givenIsFilled, String givenColor, double givenThickness){

if((givenIsFilled == true || givenIsFilled == false)){

isFilled = givenIsFilled;

}

if(givenColor != ""){

color = givenColor;

}

if(givenThickness > 0){

thickness = givenThickness;

}

}

public boolean getIsFilled(){

return isFilled;

}

public abstract double calculateArea();

}

Class Circular

public class Circle extends GeometricObject {

protected int radius;

//Full-Argument Constructor

public Circle(boolean givenIsFilled, String givenColor, double givenThickness, int givenRadius){

super(givenIsFilled, givenColor, givenThickness);

if(givenRadius > 0){

radius = givenRadius;

}

}

public double calculateArea(){

return 3.1416\*3.1416\*radius;

}

}

Class Rectangle

public class Rectangle extends GeometricObject {

protected int length;

protected int width;

//Full-Argument Constructor

public Rectangle(boolean givenIsFilled, String givenColor, double givenThickness, int givenLength, int givenWidth){

super(givenIsFilled, givenColor, givenThickness);

if(givenLength > 0){

length = givenLength;

}

if(givenWidth > 0){

width = givenWidth;

}

}

public double calculateArea(){

return length\*width;

}

}

Class Drawing

public class Drawing {

GeometricObject[] go = new GeometricObject[5];

//Method to add a Drawing

public void add(GeometricObject givenGeometricObj){

for(int i = 0; i < go.length; i++){

if(go[i] == null){

go[i] = givenGeometricObj;

}

}

}

public double calculateArea(){

double area = 0;

for(int i = 0; i < go.length; i++){

area += go[i].calculateArea();

}

return area;

}

public int countFilled(){

int count = 0;

for(int i = 0; i < go.length; i++){

if(go[i].getIsFilled()){

count++;

}

}

return count;

}

}

Class Runner

public class Runner{

public static void main(String[] args) {

Circle c1 = new Circle(true, "Blue", 2.5, 5);

Circle c2 = new Circle(false, "Red", 1.5, 10);

Rectangle r1 = new Rectangle(true, "Blue", 2.5, 4, 5);

Rectangle r2 = new Rectangle(false, "Red", 3.6, 6, 5);

Rectangle r3 = new Rectangle(true, "Green", 4.5, 1, 5);

Drawing d = new Drawing();

d.add(c1);

d.add(c2);

d.add(r1);

d.add(r2);

d.add(r3);

System.out.println("Total Area of Drawing: " + d.calculateArea());

System.out.println("Number of Filled Figures in the drawing is: " + d.countFilled());

}

}

-----------------