**15.1**

public abstract class AbstractDrawFunction extends JPanel {

private Polygon p = new Polygon();

protected AbstractDrawFunction () {

drawFunction();

}

abstract double f(double x);

public void drawFunction() {

for (int x = -100; x <= 100; x++) {

p.addPoint(x + 200, 200 - (int)f(x));

}

}

protected void paintComponent(Graphics g) {

}

}

**15.8**

|  |
| --- |
| import ToolKit.Circle2D; |
|  | import ToolKit.GeometricObject; |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | public class circle { |
|  |  |
|  | public static void main(String[] args) { |
|  | ComparableCircle c1 = new ComparableCircle(0,0,5); |
|  | ComparableCircle c2 = new ComparableCircle(0,0,10); |
|  | ComparableCircle c3 = (ComparableCircle) GeometricObject.max(c1, c2); |
|  | System.out.println(c1); |
|  | System.out.println(c2); |
|  |  |
|  | System.out.println("Max circle = " + c3.getRadius()); |
|  | System.out.println(c3); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | class ComparableCircle extends Circle2D { // GeometricObject Implements compareTo |
|  |  |
|  | ComparableCircle() { |
|  | } |
|  |  |
|  | ComparableCircle(double x, double y, double radius) { |
|  | super(x, y, radius); |
|  | } |
|  |  |
|  | } |

**15.11**

package hw5;

import java.lang.\*;

import java.util.Scanner;

interface Comparable<Circle>{

public boolean equal(Circle c);

}

class GeometricObject{

double radius;

GeometricObject(double radius){

this.radius=radius;

}

public double getArea(){

double area = 3.1416\*radius\*radius;

return area;

}

}

class Circle extends GeometricObject implements Comparable<Circle>{

public Circle(double radius) {

super(radius);

}

@Override

public boolean equal(Circle c) {

if(getArea()==c.getArea())

return true;

else

return false;

}

}

public class E15\_11 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.println("Enter 1st radius");

double radius1 = input.nextDouble();

System.out.println("Enter 2nd radius");

double radius2 = input.nextDouble();

Circle c1= new Circle(radius1);

Circle c2= new Circle(radius2);

if(c1.equal(c2)==true)

System.out.println("both the circles are equal");

else

System.out.println("both the circles are not equal");

}

}

**15.22**

|  |
| --- |
|  |
|  | public static void main(String[] args) { |
|  |  |
|  | Rational r = new Rational(BigInteger.ONE, new BigInteger("2")); |
|  |
|  |
|  | for (BigInteger i = r.getNumerator().add(BigInteger.ONE); |
|  | i.compareTo(new BigInteger("100")) <= 0; |
|  | i = i.add(BigInteger.ONE)) { |
|  | r = r.add(new Rational(i, i.add(BigInteger.ONE))); |
|  | } |
|  |  |
|  | // Display results |
|  | System.out.println(r); |
|  | } |
|  | } |