The code is :

```
import java.lang.Math;
class Square {
        private int s;

    ////////////////////////////////////////////
    // Helping functions
    ////////////////////////////////////////////
    private int square(int i) {
         return i*i;
    }
    ////////////////////////////////////////////
    // Manager functions
    ////////////////////////////////////////////
    public Square(int side) { s=side; }

    ////////////////////////////////////////////
    // Access function
    ////////////////////////////////////////////
    // get
    public int getSide() { return s; }

    // set
    public void setSide(int side) { s = side; }

    public boolean isLarge(int s) { return s > 10;}

    ////////////////////////////////////////////
    // Implementor functions
    ////////////////////////////////////////////

    public void enLarge(int ds){
           s += ds;
    }
    public int area(int s){
           return    square(s);
    }
    public int circumference(int s) {
         return s*4;
    }
```

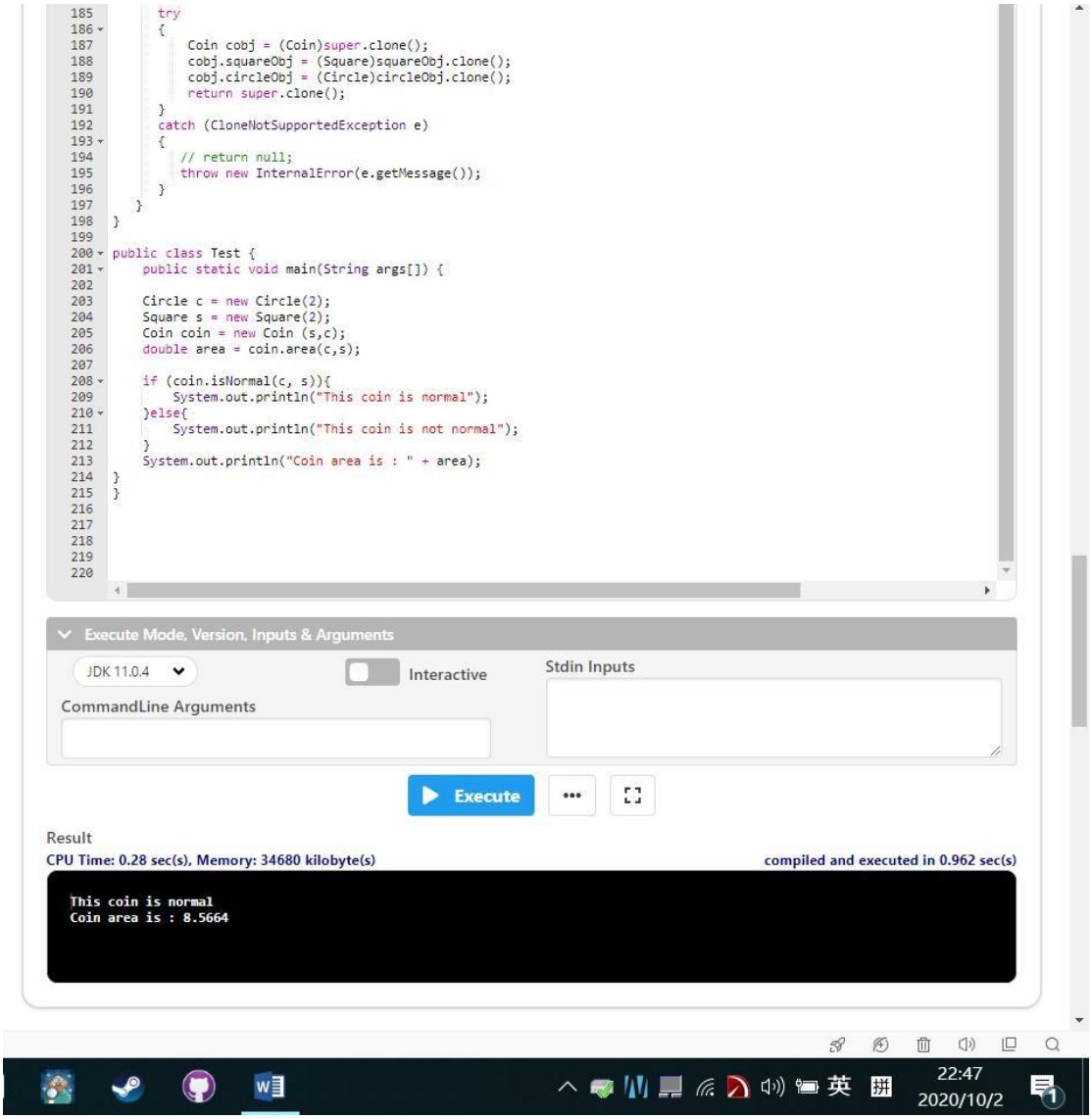
```
    public String toString(){
        return ("Square side is : " + s );
    }
    public boolean equals(Object obj){
        Square t;
        if(!(obj instanceof Square)){return false;}
        t = (Square) obj;
        return(s==t.s);
    }
    public Object clone() {
        try
        { return super.clone();
        }
        catch (CloneNotSupportedException e)
        {
            // return null;
            throw new InternalError(e.getMessage());
        }
     }
}


class Circle {

    private double r;

    ////////////////////////////////////////////////
    // Helping functions
    ////////////////////////////////////////////////
    private double pi() {
        return 3.1416;
    }
    ////////////////////////////////////////////////
    // Manager functions
    ////////////////////////////////////////////////
    public Circle(double radius) { r = radius; }

    ////////////////////////////////////////////////
    // Access function
    ////////////////////////////////////////////////
    // get
    public double getRadius() { return r; }
```

```
    // set
    public void setRadius(double radius) { r = radius; }

    public boolean isLarge(double radius) { return radius > 10;}

    public boolean isAPoint(double radius){ return radius == 0;}

    ////////////////////////////////////////////
    // Implementor functions
    ////////////////////////////////////////////

    public void enLarge(double dr){
          r += dr;
    }
    public double area(double r){
          return    pi() * r * r;
    }
    public double circumference(double r) {
         return 2 * r * pi();
    }

    public String toString(){
          return ("Circle radius is : " + r );
    }
    public boolean equals(Object obj){
          Circle t;
          if(!(obj instanceof Circle)){return false;}
          t = (Circle) obj;
          return(r==t.r);
    }
    public Object clone() {
          try
          { return super.clone();
          }
          catch (CloneNotSupportedException e)
          {
               // return null;
               throw new InternalError(e.getMessage());
          }
      }
}

class Coin {
```

```
  private Circle circleObj;
  private Square squareObj;

  ////////////////////////////////////////////
  // Helping functions
  ////////////////////////////////////////////
  private double calcCircleArea(Circle obj) {
       return obj.area(obj.getRadius());
  }

  private int calcSquareArea(Square obj) {
     return obj.area(obj.getSide());
 }
  ////////////////////////////////////////////
  // Manager functions
  ////////////////////////////////////////////
  public Coin(int s1,double r1) {
     circleObj = new Circle(r1);
     squareObj = new Square(s1);
  }
  public Coin(Square squareObj1, Circle circleObj1){
     circleObj=new Circle (circleObj1.getRadius());
     squareObj=new Square(squareObj1.getSide());
  }

  ////////////////////////////////////////////
  // Access function
  ////////////////////////////////////////////
  // get
  public int getSide(Square squareObj){ return squareObj.getSide();}
  public double getRadius(Circle circleObj) { return circleObj.getRadius(); }

  // set
  public void setSide(Square squareObj, int s){squareObj.setSide(s);}
  public void setRadius(Circle circleObj, double r) { circleObj.setRadius(r); }

  public boolean isNormal(Circle circleObj, Square squareObj) { return 2 * circleObj.getRadius() >
Math.sqrt(squareObj.getSide()*squareObj.getSide()*2);}


  ////////////////////////////////////////////
  // Implementor functions
  ////////////////////////////////////////////
```

```
    public double area(Circle circleObj, Square squareObj){
          return calcCircleArea(circleObj) - calcSquareArea(squareObj) ;
    }

     public String toString(){
           return ("Coin is : " );
     }
     public boolean equals(Object obj){
           Coin t;
           if(!(obj instanceof Coin)){return false;}
           t = (Coin) obj;
        return(circleObj==t.circleObj && squareObj==t.squareObj);
     }
       public Object clone() {
          try
          {
                Coin cobj = (Coin)super.clone();
                cobj.squareObj = (Square)squareObj.clone();
                cobj.circleObj = (Circle)circleObj.clone();
                return super.clone();
          }
          catch (CloneNotSupportedException e)
          {
               // return null;
               throw new InternalError(e.getMessage());
          }
     }
}

public class Test {
       public static void main(String args[]) {

       Circle c = new Circle(2);
       Square s = new Square(2);
       Coin coin = new Coin (s,c);
       double area = coin.area(c,s);

       if (coin.isNormal(c, s)){
             System.out.println("This coin is normal");
       }else{
             System.out.println("This coin is not normal");
       }
       System.out.println("Coin area is : " + area);
}
```

}


try
{
Coin cobj = (Coin)super.clone();
cobj.squareObj = (Square)squareObj.clone();
cobj.circleObj = (Circle)circleObj.clone();
return super.clone();
}
catch (CloneNotSupportedException e)
{
// return null;
throw new InternalError(e.getMessage());
}
}
}
public class Test {
public static void main(String args[]) {
Circle c = new Circle(2);
Square s = new Square(2);
Coin coin = new Coin (s,c);
double area = coin.area(c,s);
if (coin.isNormal(c, s)){
System.out.println("This coin is normal");
}else{
System.out.println("This coin is not normal");
}
System.out.println("Coin area is : " + area);
}
}
Execute Mode, Version, Inputs & Arguments
JDK 11.0.4
Interactive
Stdin Inputs
CommandLine Arguments
Execute
Result
CPU Time: 0.28 sec(s), Memory: 34680 kilobyte(s)
compiled and executed in 0.962 sec(s)
This coin is normal
Coin area is : 8.5664
22:47
2020/10/2