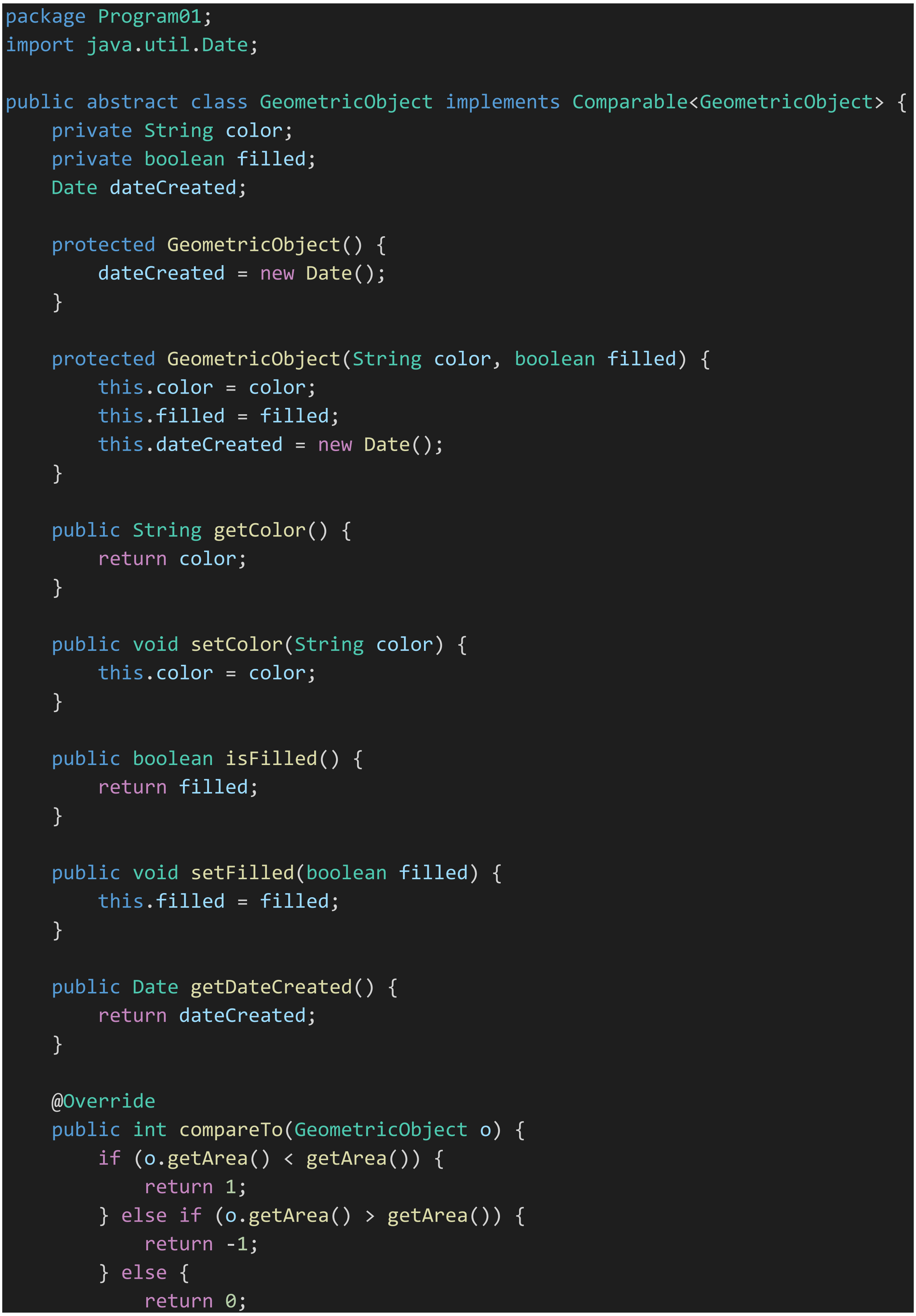
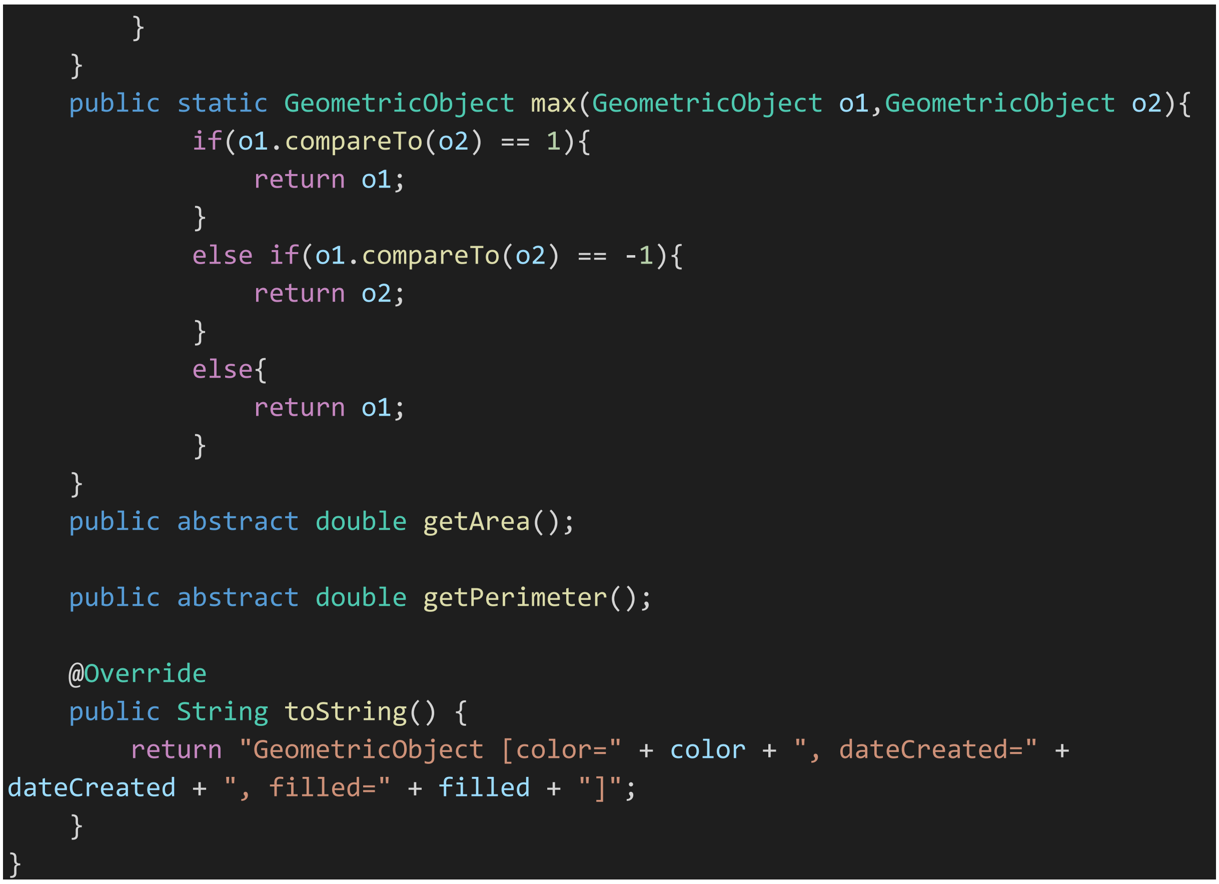
Class : Circle

```
package Program01;

public class Circle extends GeometricObject{
    private double radius;

    Circle(){

    }

    public Circle(double radius) {
        this.radius = radius;
    }

    public Circle(String color, boolean filled, double radius) {
        super(color, filled);
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }
    @Override
    public double getArea(){
        return Math.PI*(radius*radius);
    }
    @Override
    public double getPerimeter() {
        return 2*Math.PI*radius;
    }
    public double getDiameter(){
        return 2*radius;
    }

    @Override
    public String toString() {
        return super.toString()+"Circle [radius=" + radius + "]";
    }


}
```

Class: GeometricObject

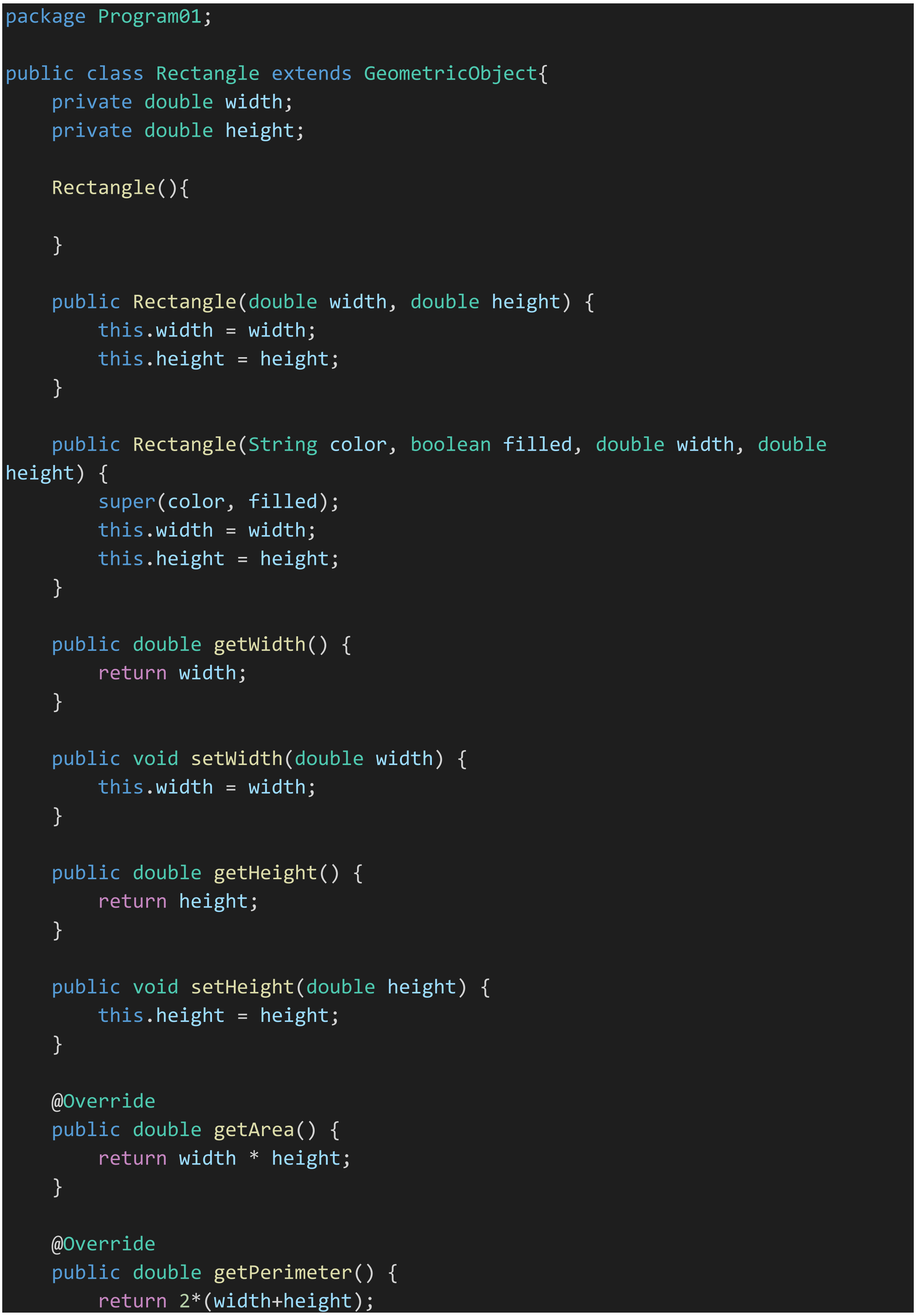


```java
package Program01;
import java.util.Date;

public abstract class GeometricObject implements Comparable<GeometricObject> {
    private String color;
    private boolean filled;
    Date dateCreated;

    protected GeometricObject() {
        dateCreated = new Date();
    }

    protected GeometricObject(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
        this.dateCreated = new Date();
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public boolean isFilled() {
        return filled;
    }

    public void setFilled(boolean filled) {
        this.filled = filled;
    }

    public Date getDateCreated() {
        return dateCreated;
    }

    @Override
    public int compareTo(GeometricObject o) {
        if (o.getArea() < getArea()) {
            return 1;
        } else if (o.getArea() > getArea()) {
            return -1;
        } else {
            return 0;
```



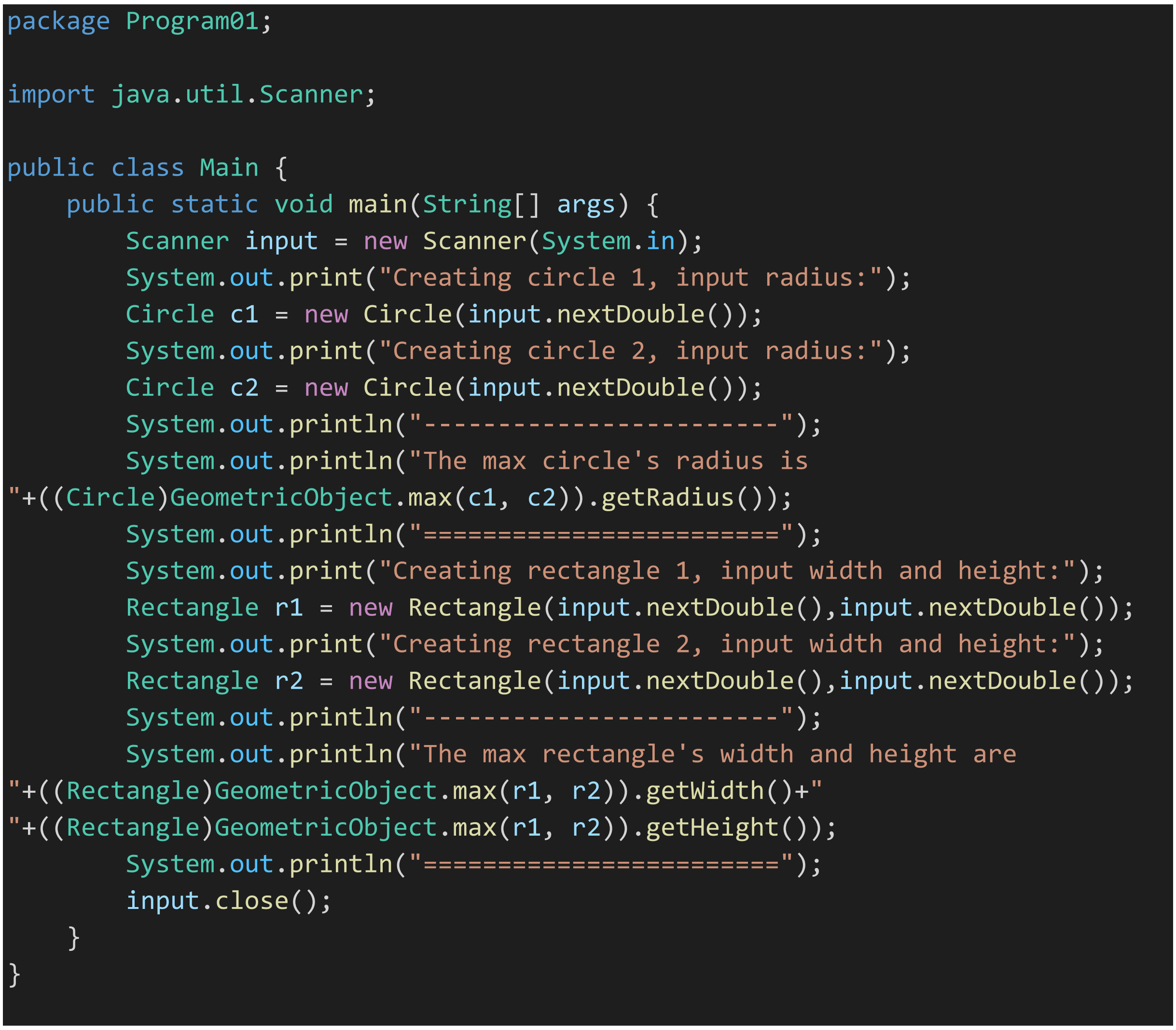
```
        }
    }
    public static GeometricObject max(GeometricObject o1,GeometricObject o2){
            if(o1.compareTo(o2) == 1){
                return o1;
            }
            else if(o1.compareTo(o2) == -1){
                return o2;
            }
            else{
                return o1;
            }
    }
    public abstract double getArea();

    public abstract double getPerimeter();

    @Override
    public String toString() {
        return "GeometricObject [color=" + color + ", dateCreated=" +
dateCreated + ", filled=" + filled + "]";
    }
}
```

Class: Rectangle



```
package Program01;

public class Rectangle extends GeometricObject{
    private double width;
    private double height;

    Rectangle(){

    }

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    public Rectangle(String color, boolean filled, double width, double
height) {
        super(color, filled);
        this.width = width;
        this.height = height;
    }

    public double getWidth() {
        return width;
    }

    public void setWidth(double width) {
        this.width = width;
    }

    public double getHeight() {
        return height;
    }

    public void setHeight(double height) {
        this.height = height;
    }

    @Override
    public double getArea() {
        return width * height;
    }

    @Override
    public double getPerimeter() {
        return 2*(width+height);
    }

    @Override
    public String toString() {
        return super.toString() + "Rectangle [height=" + height + ", width=" +
width + "]";
    }

}
```

Class: Main



```java
package Program01;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Creating circle 1, input radius:");
        Circle c1 = new Circle(input.nextDouble());
        System.out.print("Creating circle 2, input radius:");
        Circle c2 = new Circle(input.nextDouble());
        System.out.println("------------------------");
        System.out.println("The max circle's radius is 
"+((Circle)GeometricObject.max(c1, c2)).getRadius());
        System.out.println("========================");
        System.out.print("Creating rectangle 1, input width and height:");
        Rectangle r1 = new Rectangle(input.nextDouble(),input.nextDouble());
        System.out.print("Creating rectangle 2, input width and height:");
        Rectangle r2 = new Rectangle(input.nextDouble(),input.nextDouble());
        System.out.println("------------------------");
        System.out.println("The max rectangle's width and height are 
"+((Rectangle)GeometricObject.max(r1, r2)).getWidth()+" 
"+((Rectangle)GeometricObject.max(r1, r2)).getHeight());
        System.out.println("========================");
        input.close();
    }
}
```