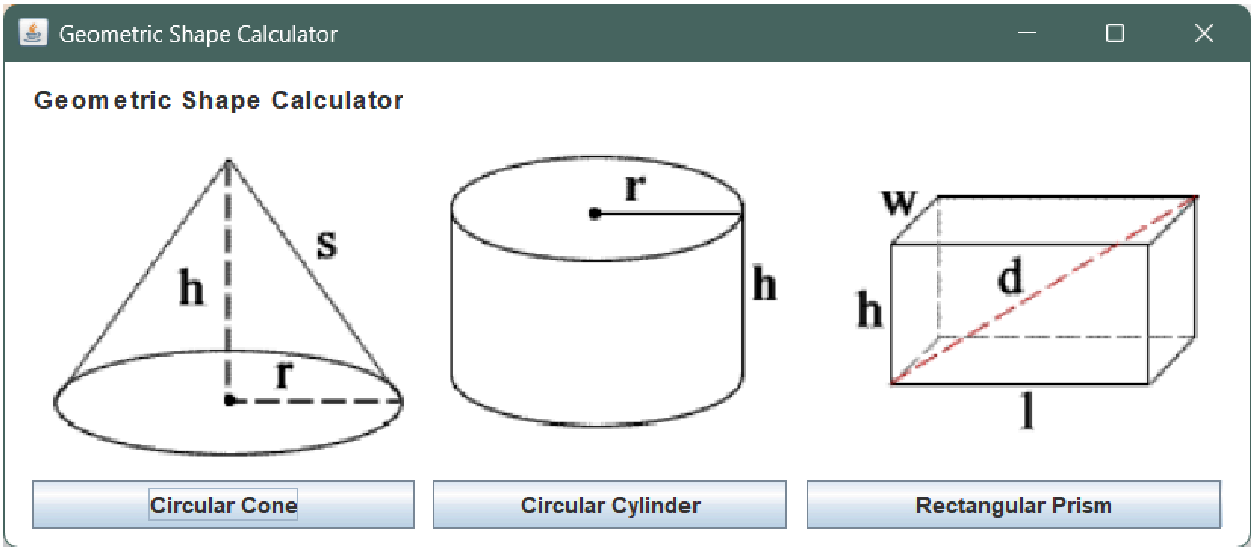
รหัสนิสิต	6621600950	ชื่อ/นามสกุล	นวพร สุขสวัสดิ์

Lab #10

ฝึกการใช้ GUI และ JAR

Question #1

1. จาก Java Project ชื่อว่า *GeometricShapeCalculator* ในเนื้อหาช่วงบรรยาย ให้นิสิตพัฒนาส่วนที่เหลือให้ครบสมบูรณ์
2. สร้าง GUI Form ในรูปทรงส่วนที่เหลือให้ครบถ้วน ได้แก่ Circular Cylinder และ Rectangular Prism โดยในส่วนของ Sub Form ทั้ง 2 นั้นให้ออกแบบได้ตามสะดวก

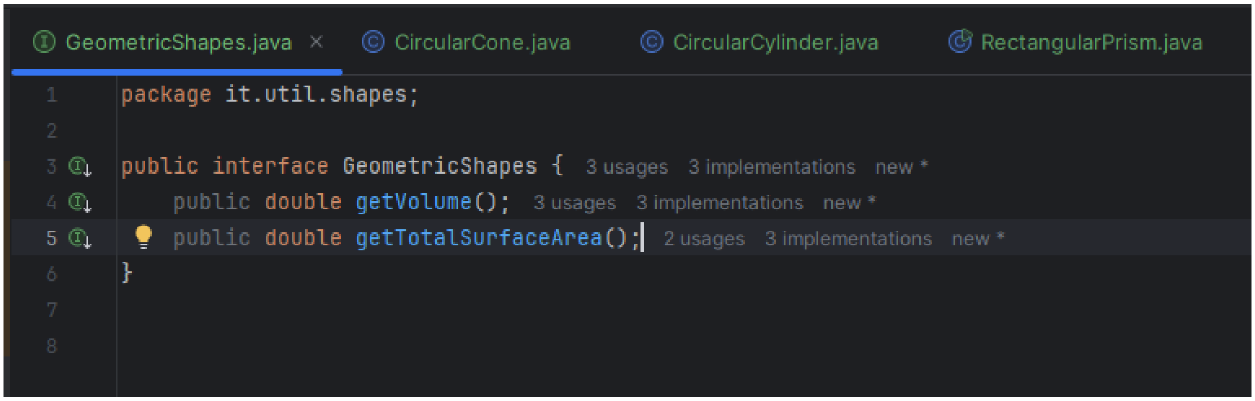


คำตอบ #1

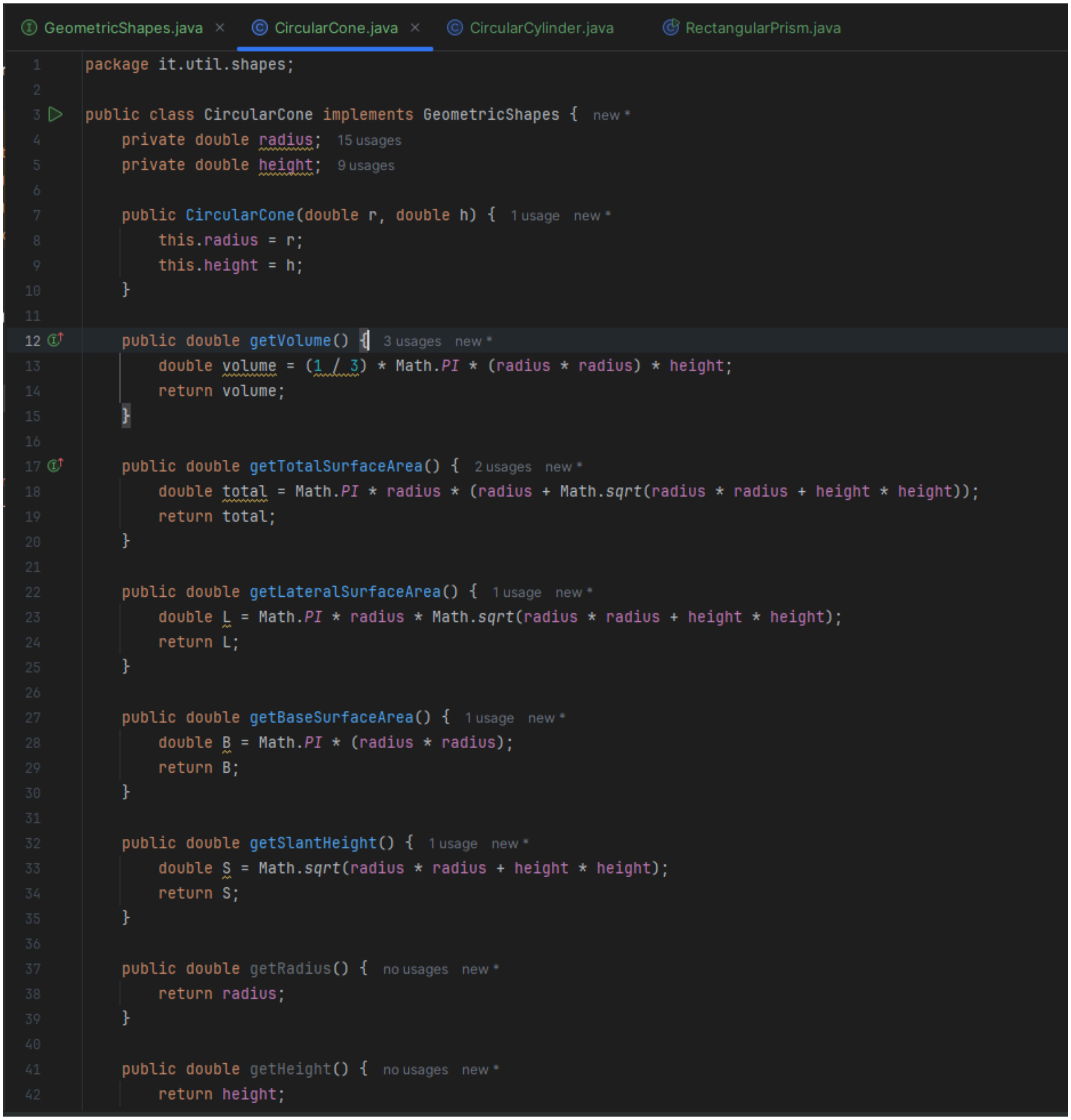
Source Code

// วาง source code ของคลาสต่าง ๆ ตรงจุดนี้

// 1. Code ของ Class ทั้งหมดที่ใช้สร้าง GeometricShapeUtil.jar

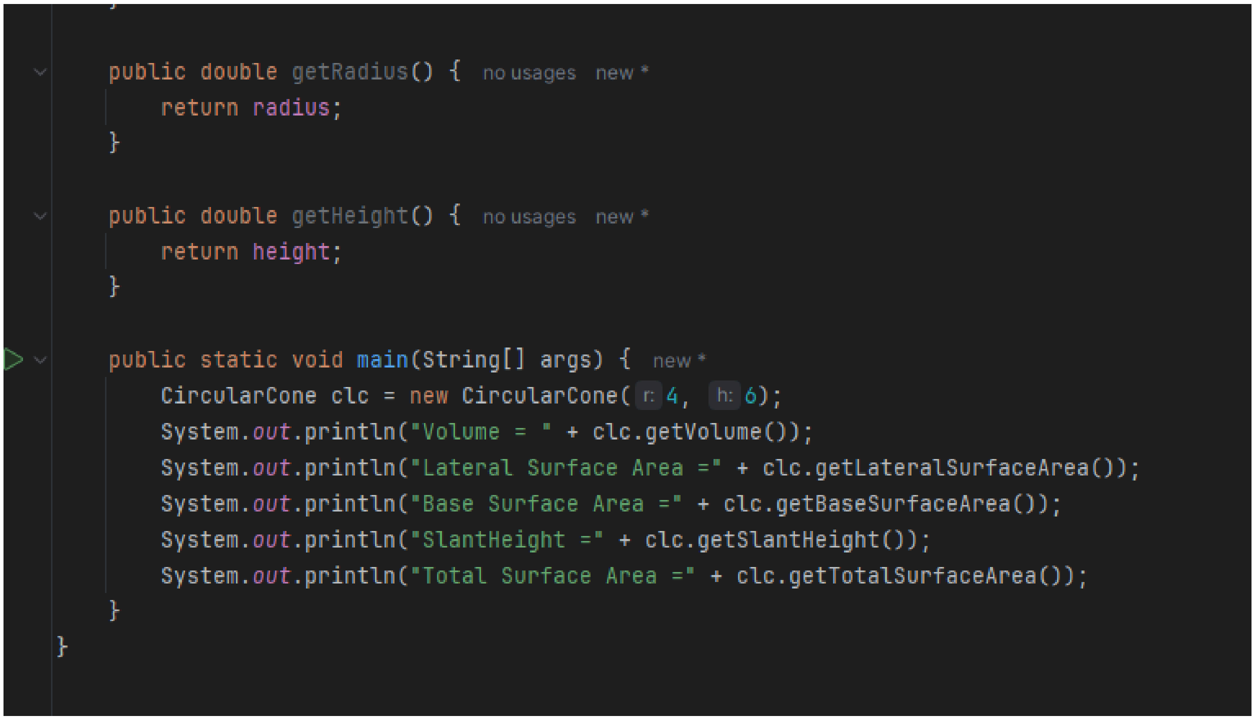

GeometricShapes.java
CircularCone.java
CircularCylinder.java
RectangularPrism.java
3 usages 3 implementations new *
3 usages 3 implementations new *
2 usages 3 implementations new *

```java
package it.util.shapes;

public interface GeometricShapes {
    public double getVolume();
    public double getTotalSurfaceArea();
}
```

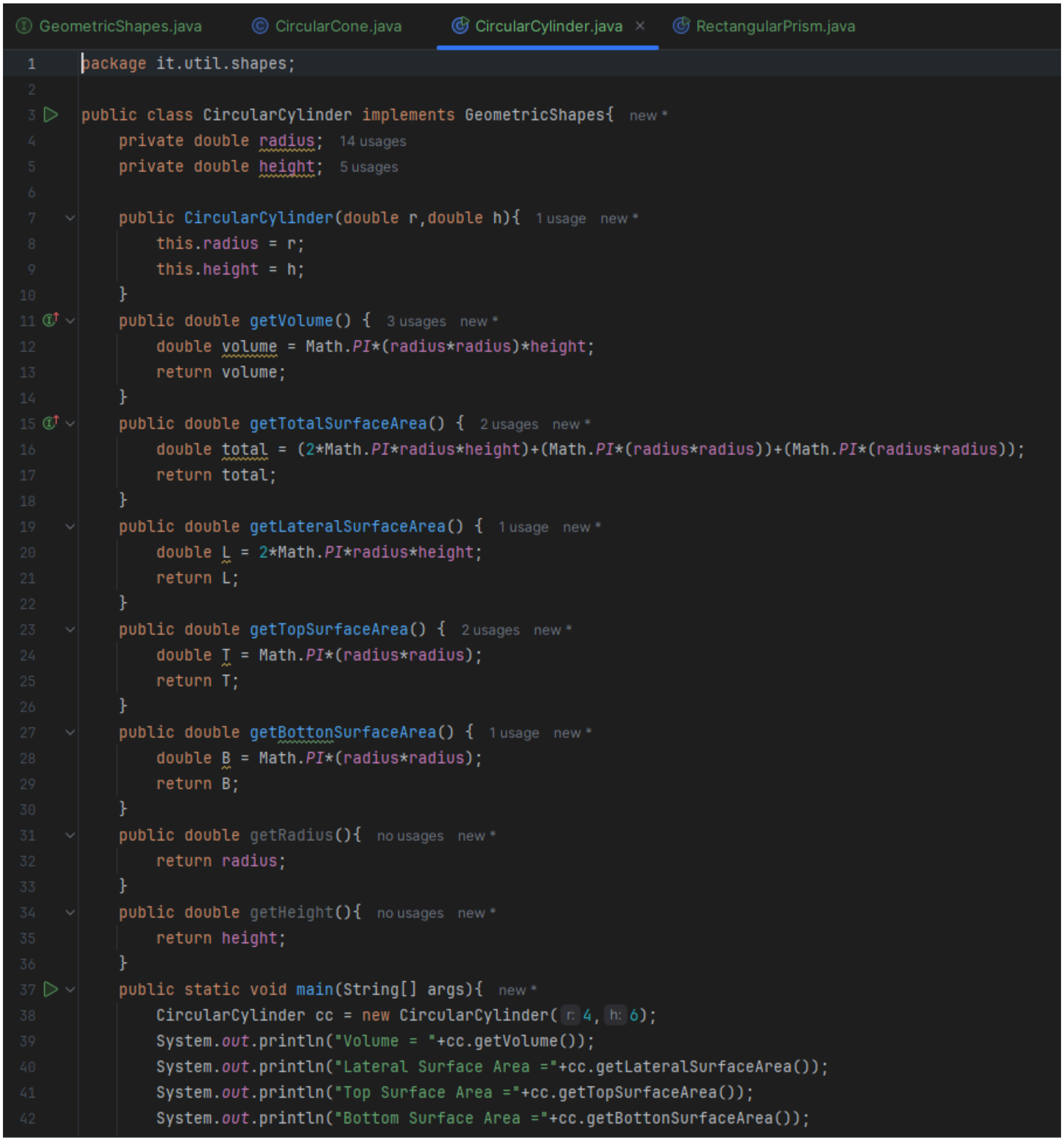


```java
package it.util.shapes;

public class CircularCone implements GeometricShapes {  new *
    private double radius;  15 usages
    private double height;  9 usages

    public CircularCone(double r, double h) {  1 usage  new *
        this.radius = r;
        this.height = h;
    }

    public double getVolume() {  3 usages  new *
        double volume = (1 / 3) * Math.PI * (radius * radius) * height;
        return volume;
    }

    public double getTotalSurfaceArea() {  2 usages  new *
        double total = Math.PI * radius * (radius + Math.sqrt(radius * radius + height * height));
        return total;
    }

    public double getLateralSurfaceArea() {  1 usage  new *
        double L = Math.PI * radius * Math.sqrt(radius * radius + height * height);
        return L;
    }

    public double getBaseSurfaceArea() {  1 usage  new *
        double B = Math.PI * (radius * radius);
        return B;
    }

    public double getSlantHeight() {  1 usage  new *
        double S = Math.sqrt(radius * radius + height * height);
        return S;
    }

    public double getRadius() {  no usages  new *
        return radius;
    }

    public double getHeight() {  no usages  new *
        return height;
```

ต่อ

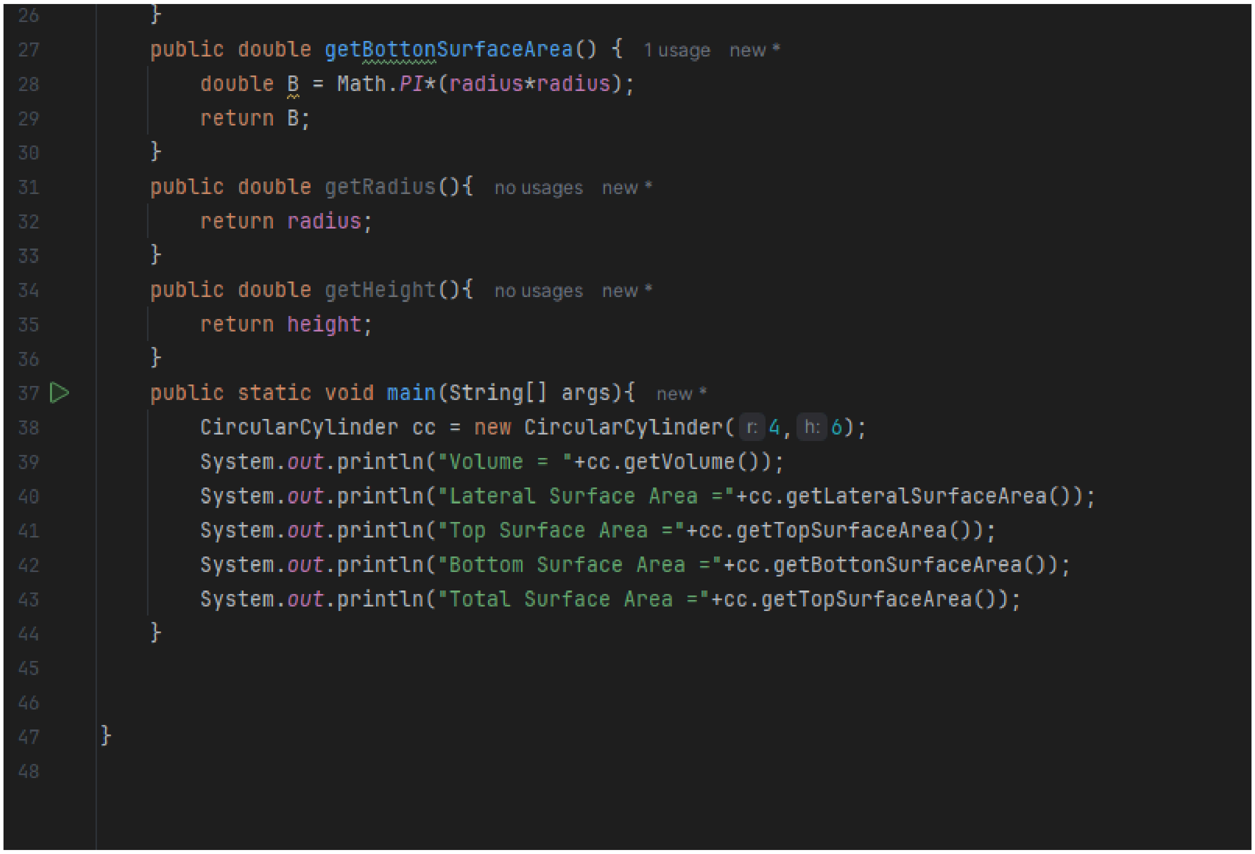


```
    }

    public double getRadius() {
        return radius;
    }

    public double getHeight() {
        return height;
    }

    public static void main(String[] args) {
        CircularCone clc = new CircularCone(4, 6);
        System.out.println("Volume = " + clc.getVolume());
        System.out.println("Lateral Surface Area =" + clc.getLateralSurfaceArea());
        System.out.println("Base Surface Area =" + clc.getBaseSurfaceArea());
        System.out.println("SlantHeight =" + clc.getSlantHeight());
        System.out.println("Total Surface Area =" + clc.getTotalSurfaceArea());
    }
}
```

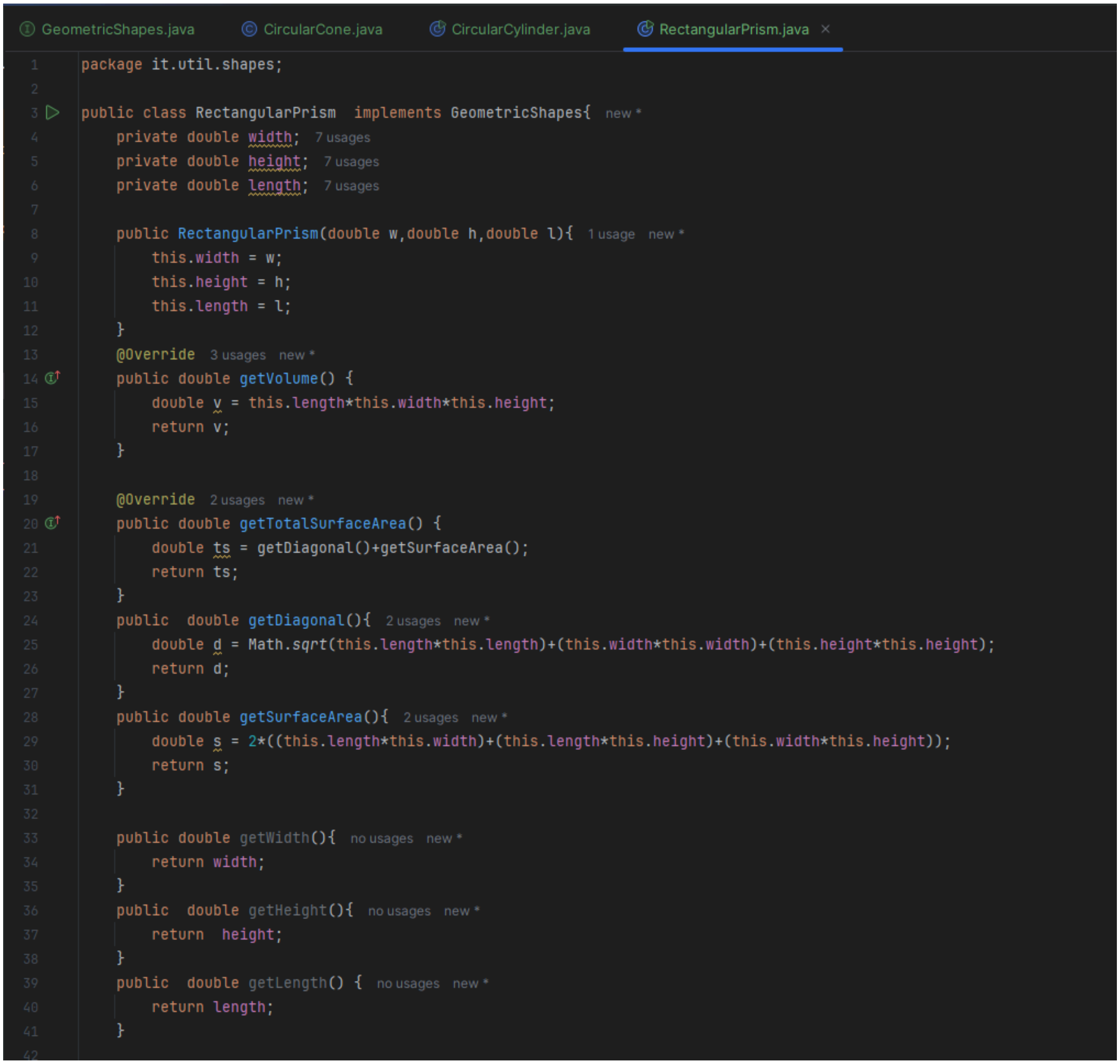


```java
package it.util.shapes;

public class CircularCylinder implements GeometricShapes{
    private double radius;
    private double height;

    public CircularCylinder(double r,double h){
        this.radius = r;
        this.height = h;
    }
    public double getVolume() {
        double volume = Math.PI*(radius*radius)*height;
        return volume;
    }
    public double getTotalSurfaceArea() {
        double total = (2*Math.PI*radius*height)+(Math.PI*(radius*radius))+(Math.PI*(radius*radius));
        return total;
    }
    public double getLateralSurfaceArea() {
        double L = 2*Math.PI*radius*height;
        return L;
    }
    public double getTopSurfaceArea() {
        double T = Math.PI*(radius*radius);
        return T;
    }
    public double getBottonSurfaceArea() {
        double B = Math.PI*(radius*radius);
        return B;
    }
    public double getRadius(){
        return radius;
    }
    public double getHeight(){
        return height;
    }
    public static void main(String[] args){
        CircularCylinder cc = new CircularCylinder(4,6);
        System.out.println("Volume = "+cc.getVolume());
        System.out.println("Lateral Surface Area ="+cc.getLateralSurfaceArea());
        System.out.println("Top Surface Area ="+cc.getTopSurfaceArea());
        System.out.println("Bottom Surface Area ="+cc.getBottonSurfaceArea());
```

ต่อ

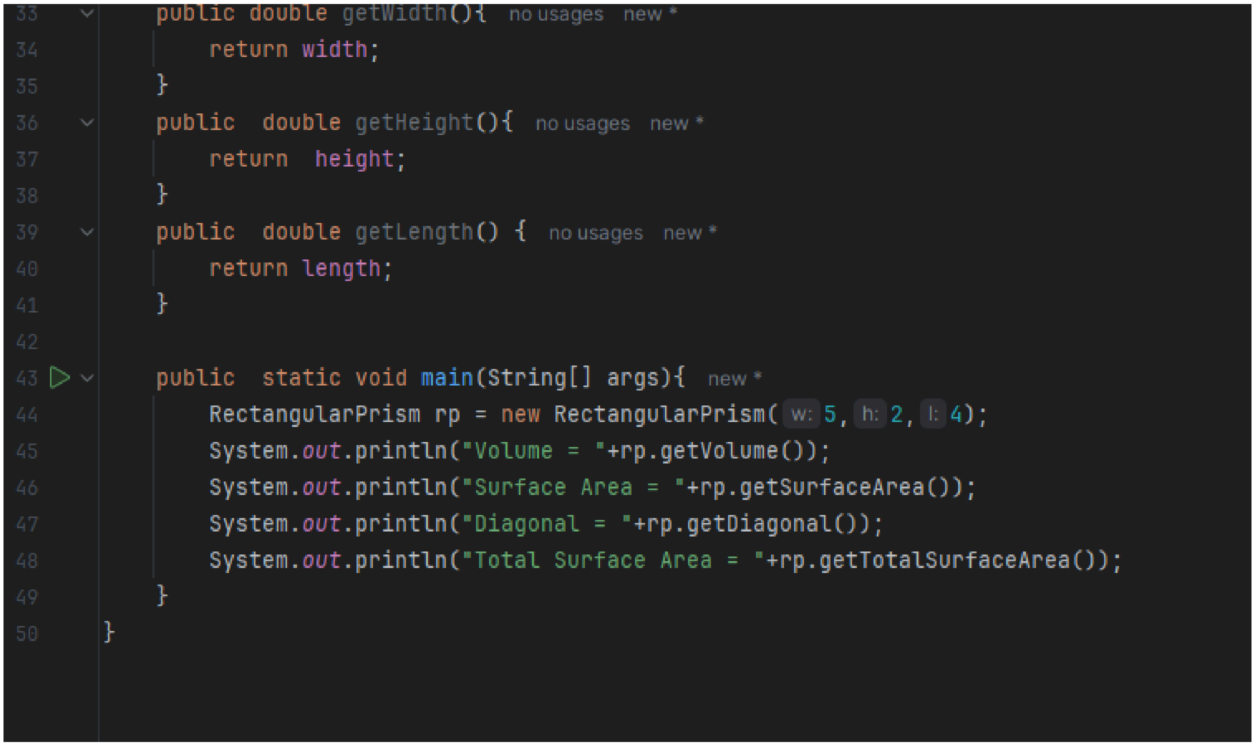


```
    }
    public double getBottonSurfaceArea() {  1 usage   new *
        double B = Math.PI*(radius*radius);
        return B;
    }
    public double getRadius(){  no usages   new *
        return radius;
    }
    public double getHeight(){  no usages   new *
        return height;
    }
    public static void main(String[] args){  new *
        CircularCylinder cc = new CircularCylinder( r: 4, h: 6);
        System.out.println("Volume = "+cc.getVolume());
        System.out.println("Lateral Surface Area ="+cc.getLateralSurfaceArea());
        System.out.println("Top Surface Area ="+cc.getTopSurfaceArea());
        System.out.println("Bottom Surface Area ="+cc.getBottonSurfaceArea());
        System.out.println("Total Surface Area ="+cc.getTopSurfaceArea());
    }


}
```



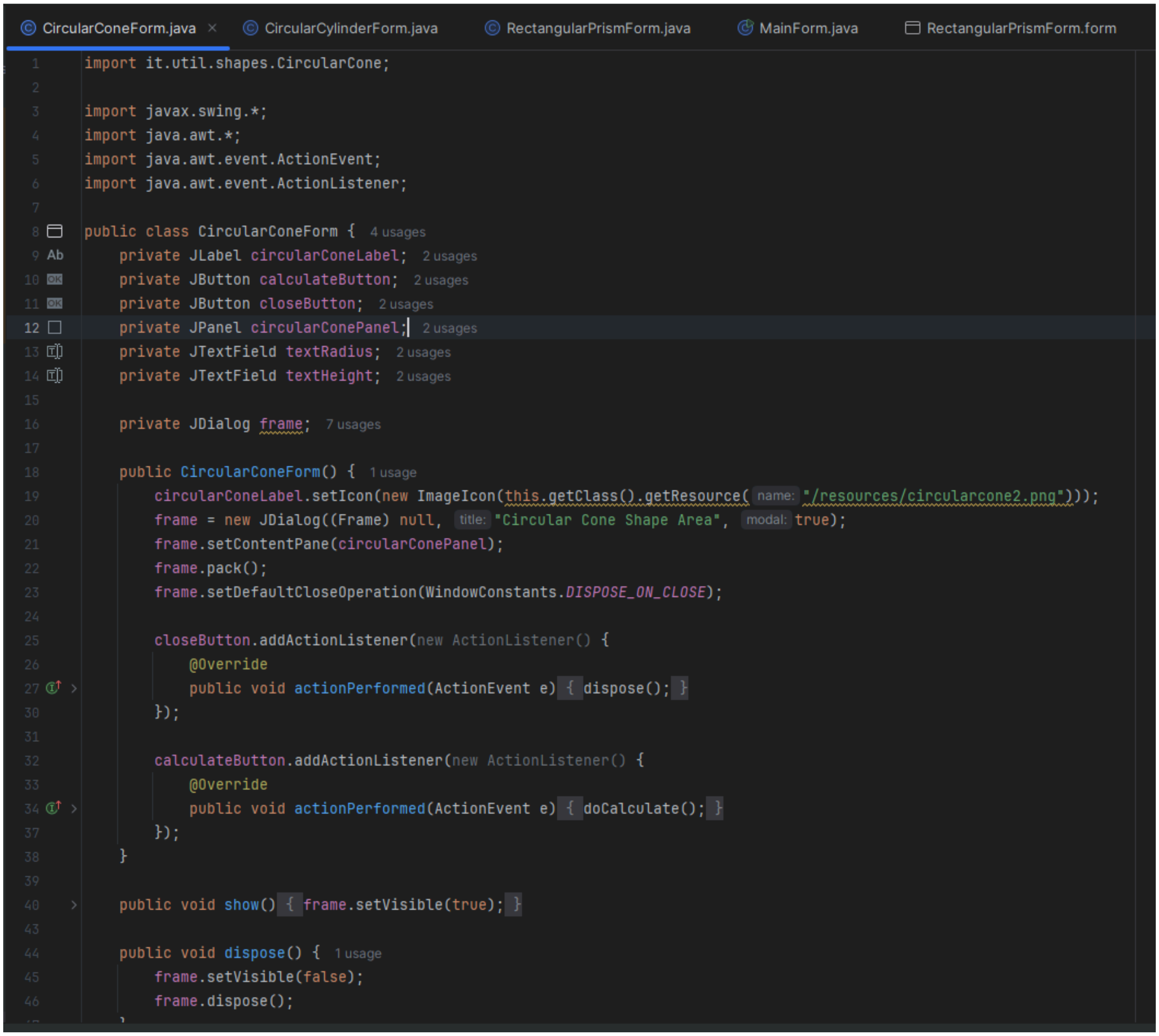
```java
package it.util.shapes;

public class RectangularPrism  implements GeometricShapes{
    private double width;
    private double height;
    private double length;

    public RectangularPrism(double w,double h,double l){
        this.width = w;
        this.height = h;
        this.length = l;
    }
    @Override
    public double getVolume() {
        double v = this.length*this.width*this.height;
        return v;
    }

    @Override
    public double getTotalSurfaceArea() {
        double ts = getDiagonal()+getSurfaceArea();
        return ts;
    }
    public  double getDiagonal(){
        double d = Math.sqrt(this.length*this.length)+(this.width*this.width)+(this.height*this.height);
        return d;
    }
    public double getSurfaceArea(){
        double s = 2*((this.length*this.width)+(this.length*this.height)+(this.width*this.height));
        return s;
    }

    public double getWidth(){
        return width;
    }
    public  double getHeight(){
        return  height;
    }
    public  double getLength() {
        return length;
    }
```

ต่อ



```
    public double getWidth(){  no usages  new *
        return width;
    }
    public  double getHeight(){  no usages  new *
        return  height;
    }
    public  double getLength() {  no usages  new *
        return length;
    }

    public  static void main(String[] args){  new *
        RectangularPrism rp = new RectangularPrism( w: 5, h: 2, l: 4);
        System.out.println("Volume = "+rp.getVolume());
        System.out.println("Surface Area = "+rp.getSurfaceArea());
        System.out.println("Diagonal = "+rp.getDiagonal());
        System.out.println("Total Surface Area = "+rp.getTotalSurfaceArea());
    }
}
```

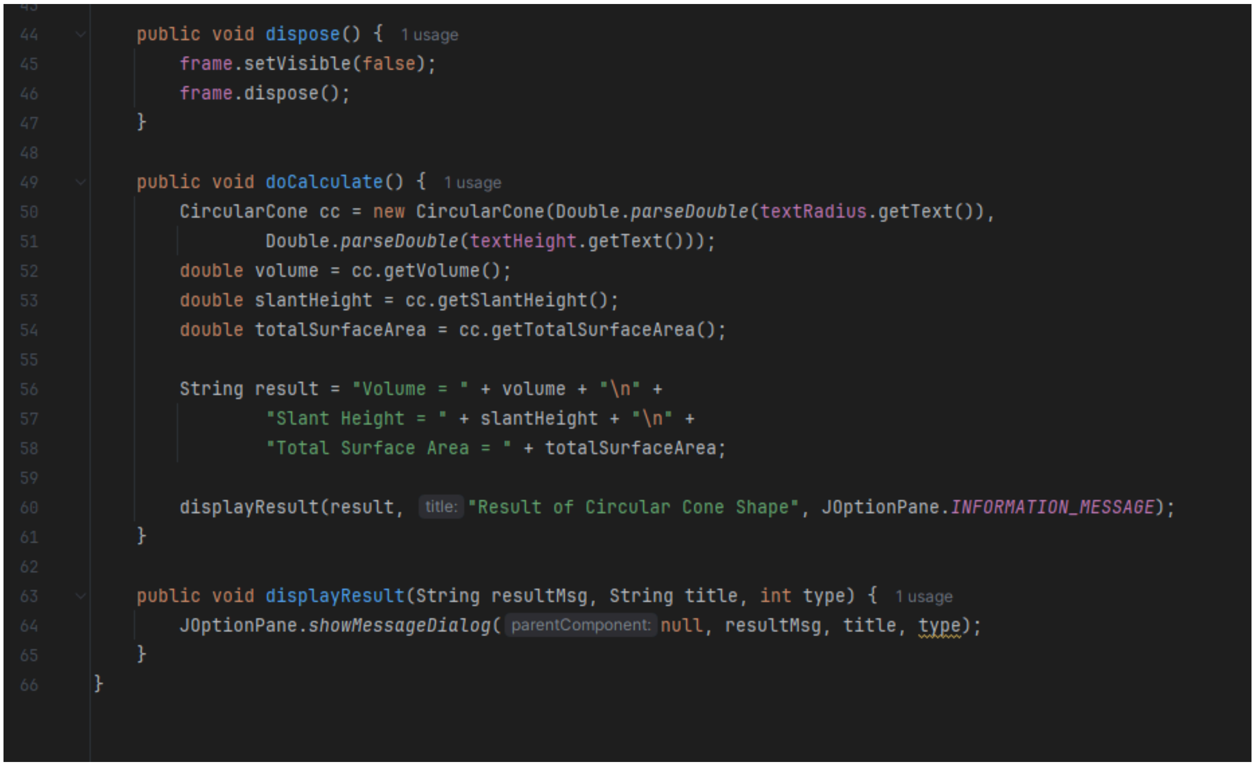
// 2. Code Java ในโปรเจค GeometricShapeCalculator



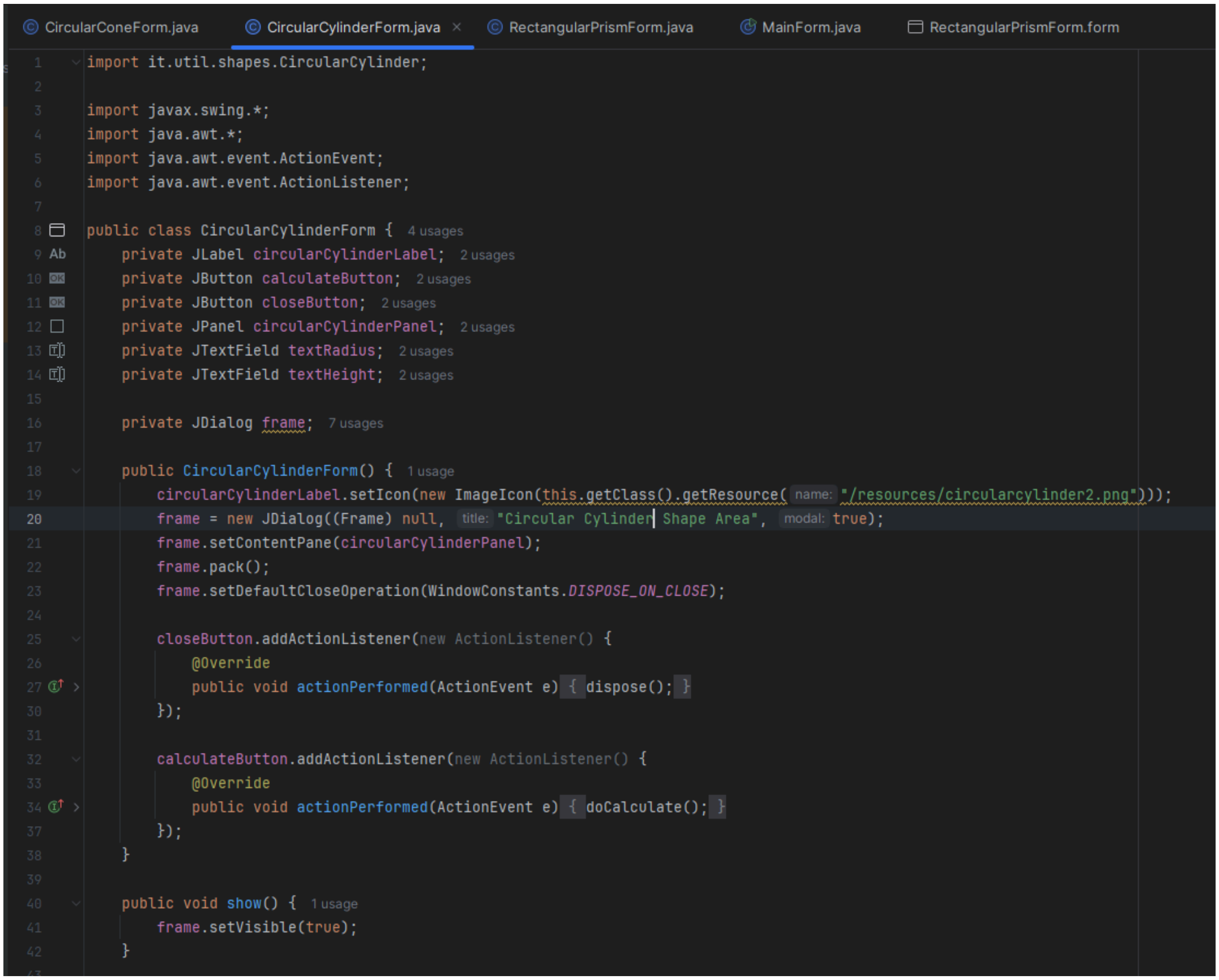
CircularConeForm.java
CircularCylinderForm.java
RectangularPrismForm.java
MainForm.java
RectangularPrismForm.form

```java
import it.util.shapes.CircularCone;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class CircularConeForm {  4 usages
    private JLabel circularConeLabel;  2 usages
    private JButton calculateButton;  2 usages
    private JButton closeButton;  2 usages
    private JPanel circularConePanel;  2 usages
    private JTextField textRadius;  2 usages
    private JTextField textHeight;  2 usages

    private JDialog frame;  7 usages

    public CircularConeForm() {  1 usage
        circularConeLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/circularcone2.png")));
        frame = new JDialog((Frame) null,  title: "Circular Cone Shape Area",  modal: true);
        frame.setContentPane(circularConePanel);
        frame.pack();
        frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);

        closeButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { dispose(); }
        });

        calculateButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { doCalculate(); }
        });
    }

    public void show() { frame.setVisible(true); }

    public void dispose() {  1 usage
        frame.setVisible(false);
        frame.dispose();
    }
```

ต่อ

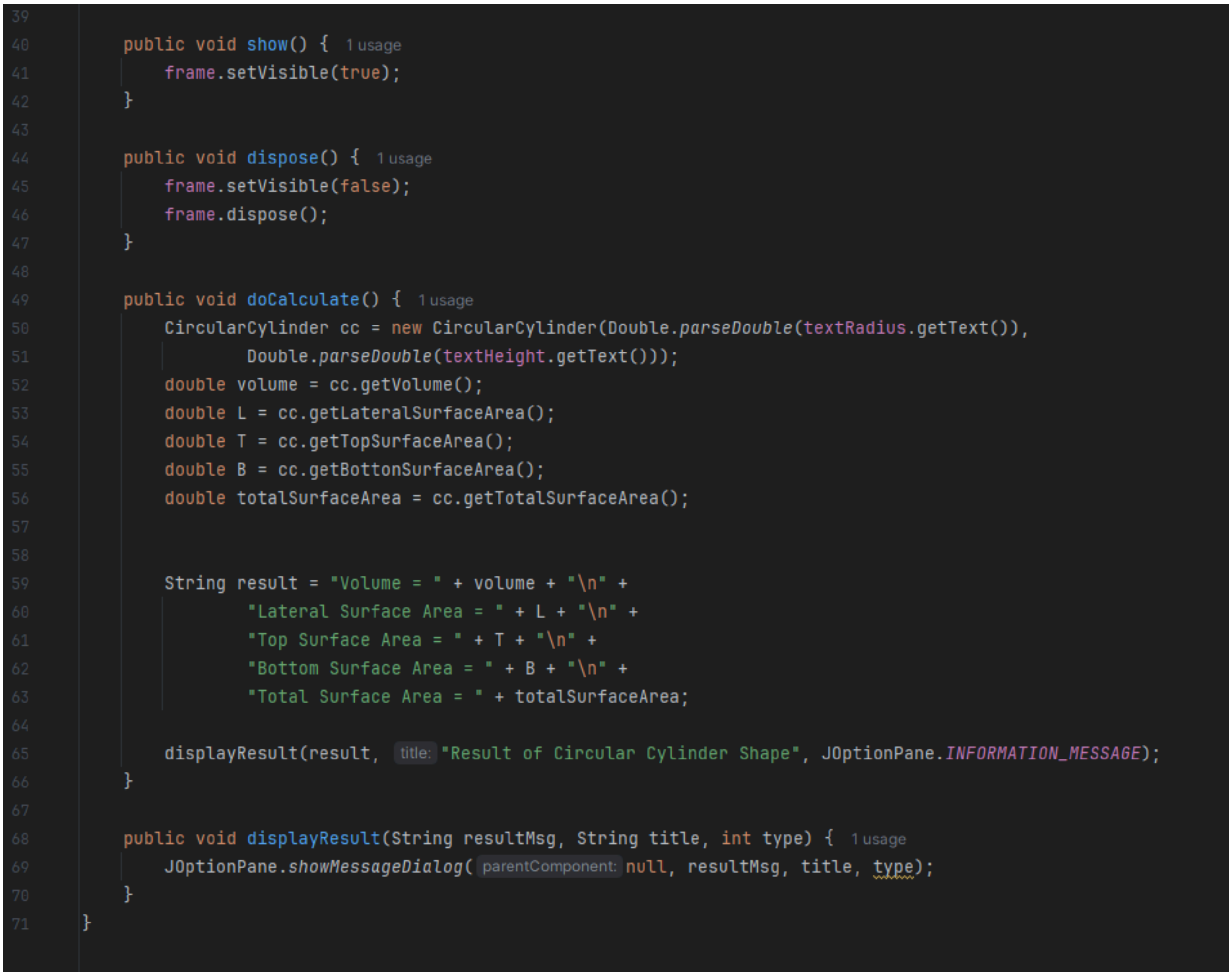


```
    public void dispose() {  1 usage
        frame.setVisible(false);
        frame.dispose();
    }

    public void doCalculate() {  1 usage
        CircularCone cc = new CircularCone(Double.parseDouble(textRadius.getText()),
                Double.parseDouble(textHeight.getText()));
        double volume = cc.getVolume();
        double slantHeight = cc.getSlantHeight();
        double totalSurfaceArea = cc.getTotalSurfaceArea();

        String result = "Volume = " + volume + "\n" +
                "Slant Height = " + slantHeight + "\n" +
                "Total Surface Area = " + totalSurfaceArea;

        displayResult(result,  title: "Result of Circular Cone Shape", JOptionPane.INFORMATION_MESSAGE);
    }

    public void displayResult(String resultMsg, String title, int type) {  1 usage
        JOptionPane.showMessageDialog( parentComponent: null, resultMsg, title, type);
    }
}
```

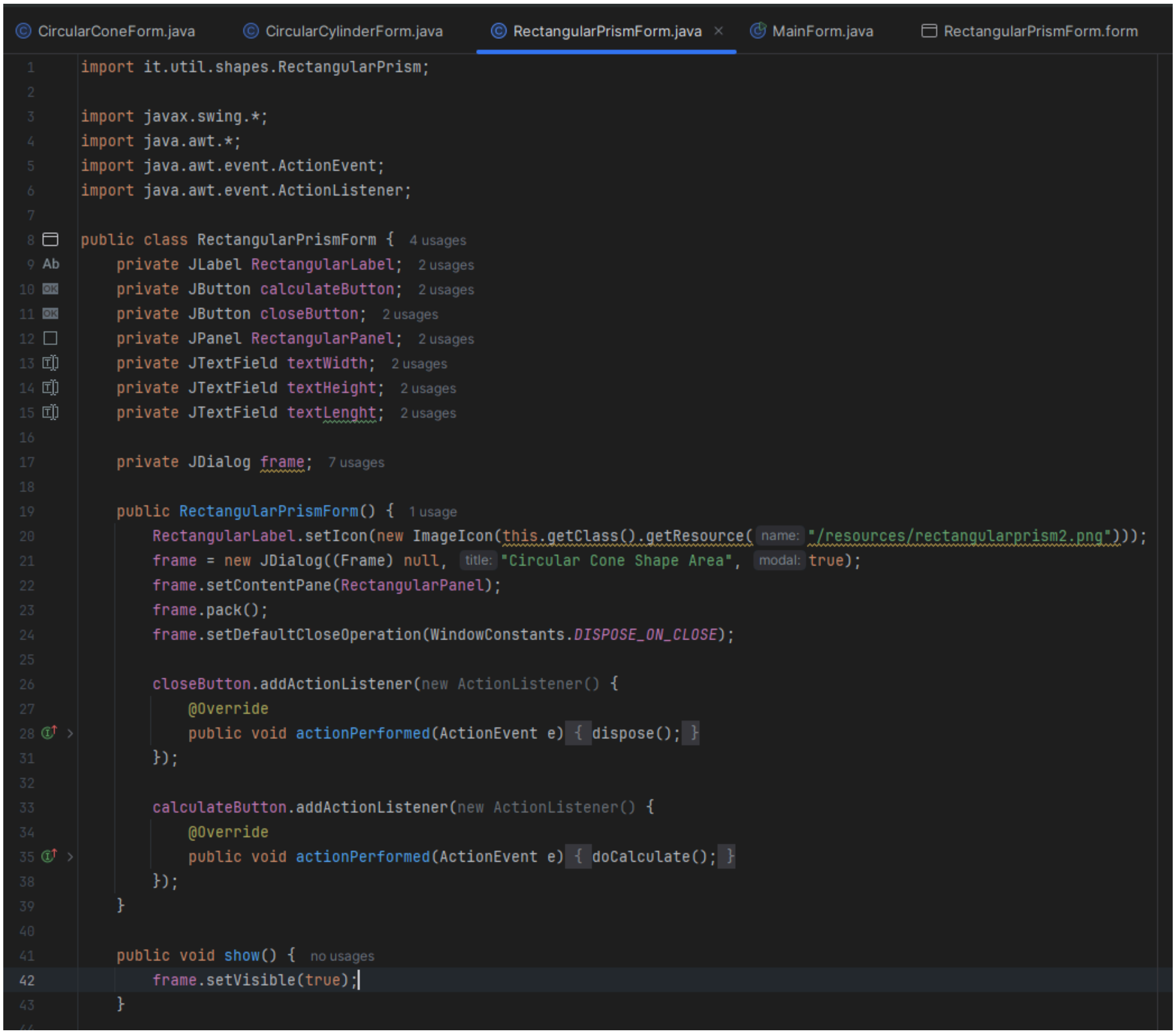

CircularConeForm.java
CircularCylinderForm.java
RectangularPrismForm.java
MainForm.java
RectangularPrismForm.form
4 usages
2 usages
7 usages
1 usage

```java
import it.util.shapes.CircularCylinder;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class CircularCylinderForm {
    private JLabel circularCylinderLabel;
    private JButton calculateButton;
    private JButton closeButton;
    private JPanel circularCylinderPanel;
    private JTextField textRadius;
    private JTextField textHeight;

    private JDialog frame;

    public CircularCylinderForm() {
        circularCylinderLabel.setIcon(new ImageIcon(this.getClass().getResource("/resources/circularcylinder2.png")));
        frame = new JDialog((Frame) null, "Circular Cylinder Shape Area", true);
        frame.setContentPane(circularCylinderPanel);
        frame.pack();
        frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);

        closeButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { dispose(); }
        });

        calculateButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { doCalculate(); }
        });
    }

    public void show() {
        frame.setVisible(true);
    }
```

ต่อ

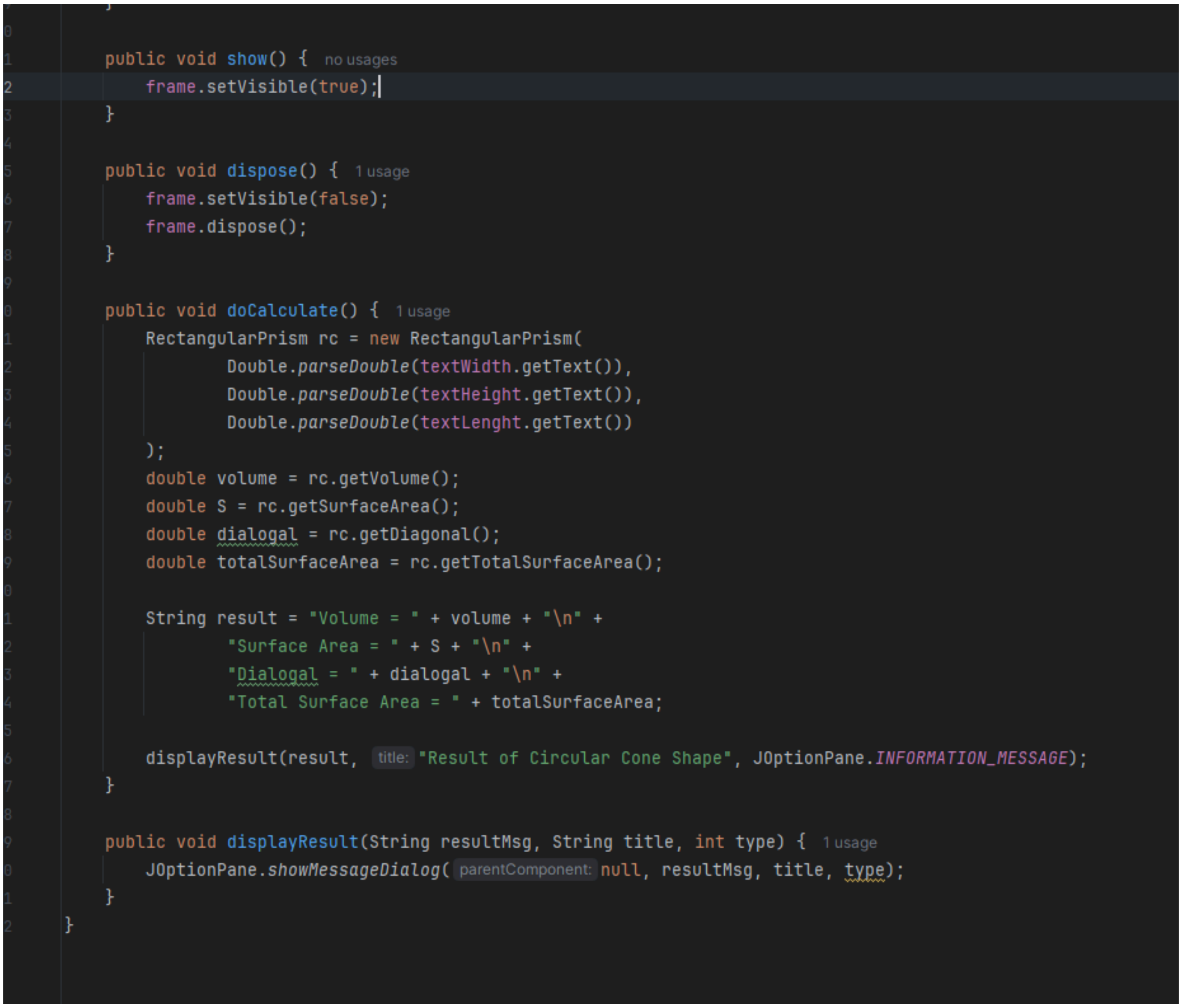


```
    public void show() {  1 usage
        frame.setVisible(true);
    }

    public void dispose() {  1 usage
        frame.setVisible(false);
        frame.dispose();
    }

    public void doCalculate() {  1 usage
        CircularCylinder cc = new CircularCylinder(Double.parseDouble(textRadius.getText()),
                Double.parseDouble(textHeight.getText()));
        double volume = cc.getVolume();
        double L = cc.getLateralSurfaceArea();
        double T = cc.getTopSurfaceArea();
        double B = cc.getBottonSurfaceArea();
        double totalSurfaceArea = cc.getTotalSurfaceArea();


        String result = "Volume = " + volume + "\n" +
                "Lateral Surface Area = " + L + "\n" +
                "Top Surface Area = " + T + "\n" +
                "Bottom Surface Area = " + B + "\n" +
                "Total Surface Area = " + totalSurfaceArea;

        displayResult(result,  title: "Result of Circular Cylinder Shape", JOptionPane.INFORMATION_MESSAGE);
    }

    public void displayResult(String resultMsg, String title, int type) {  1 usage
        JOptionPane.showMessageDialog( parentComponent: null, resultMsg, title, type);
    }
}
```

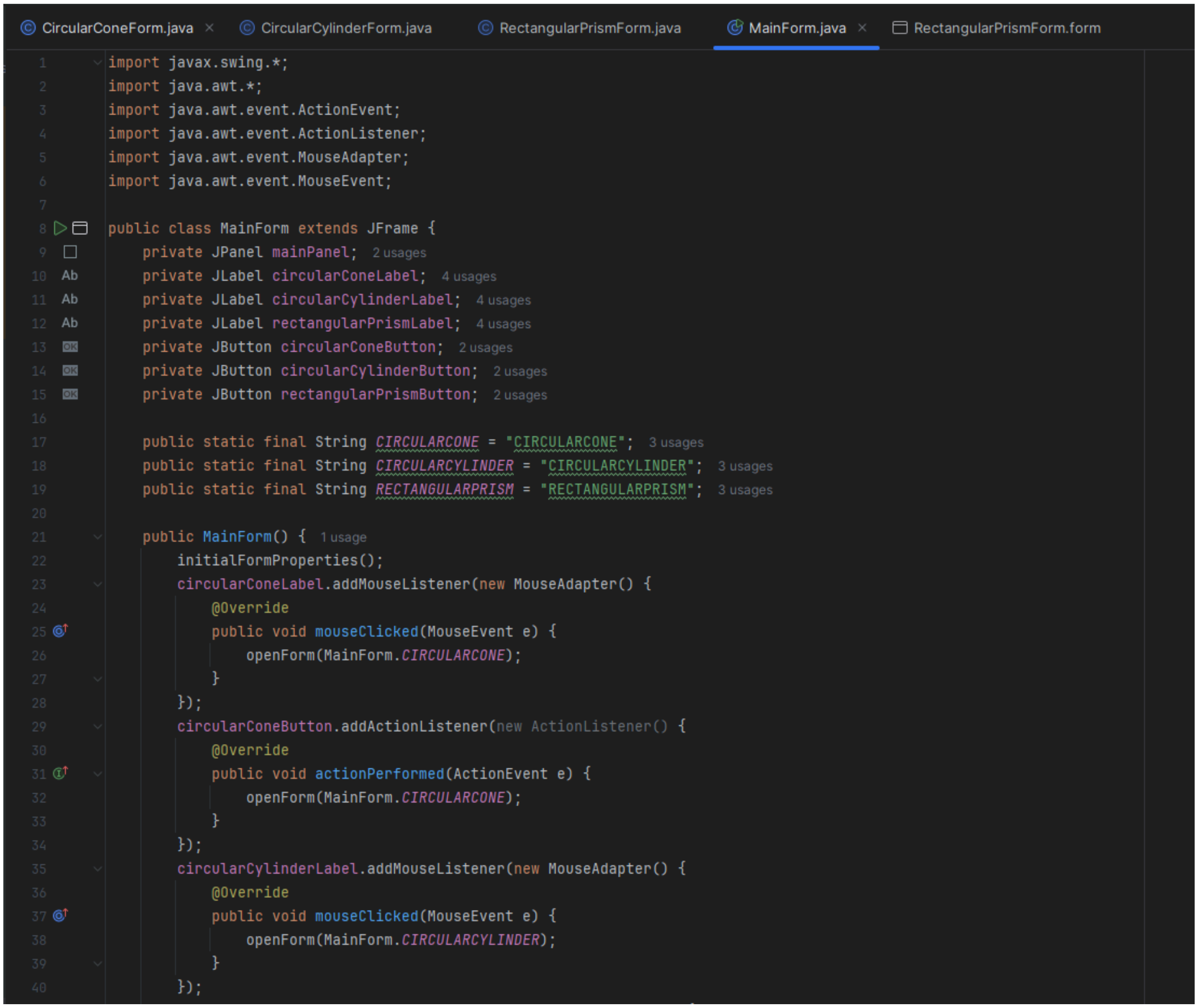

CircularConeForm.java
CircularCylinderForm.java
RectangularPrismForm.java
MainForm.java
RectangularPrismForm.form

```
import it.util.shapes.RectangularPrism;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class RectangularPrismForm {  4 usages
    private JLabel RectangularLabel;  2 usages
    private JButton calculateButton;  2 usages
    private JButton closeButton;  2 usages
    private JPanel RectangularPanel;  2 usages
    private JTextField textWidth;  2 usages
    private JTextField textHeight;  2 usages
    private JTextField textLenght;  2 usages

    private JDialog frame;  7 usages

    public RectangularPrismForm() {  1 usage
        RectangularLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/rectangularprism2.png")));
        frame = new JDialog((Frame) null,  title: "Circular Cone Shape Area",  modal: true);
        frame.setContentPane(RectangularPanel);
        frame.pack();
        frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);

        closeButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { dispose(); }
        });

        calculateButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) { doCalculate(); }
        });
    }

    public void show() {  no usages
        frame.setVisible(true);
    }
```

ต่อ

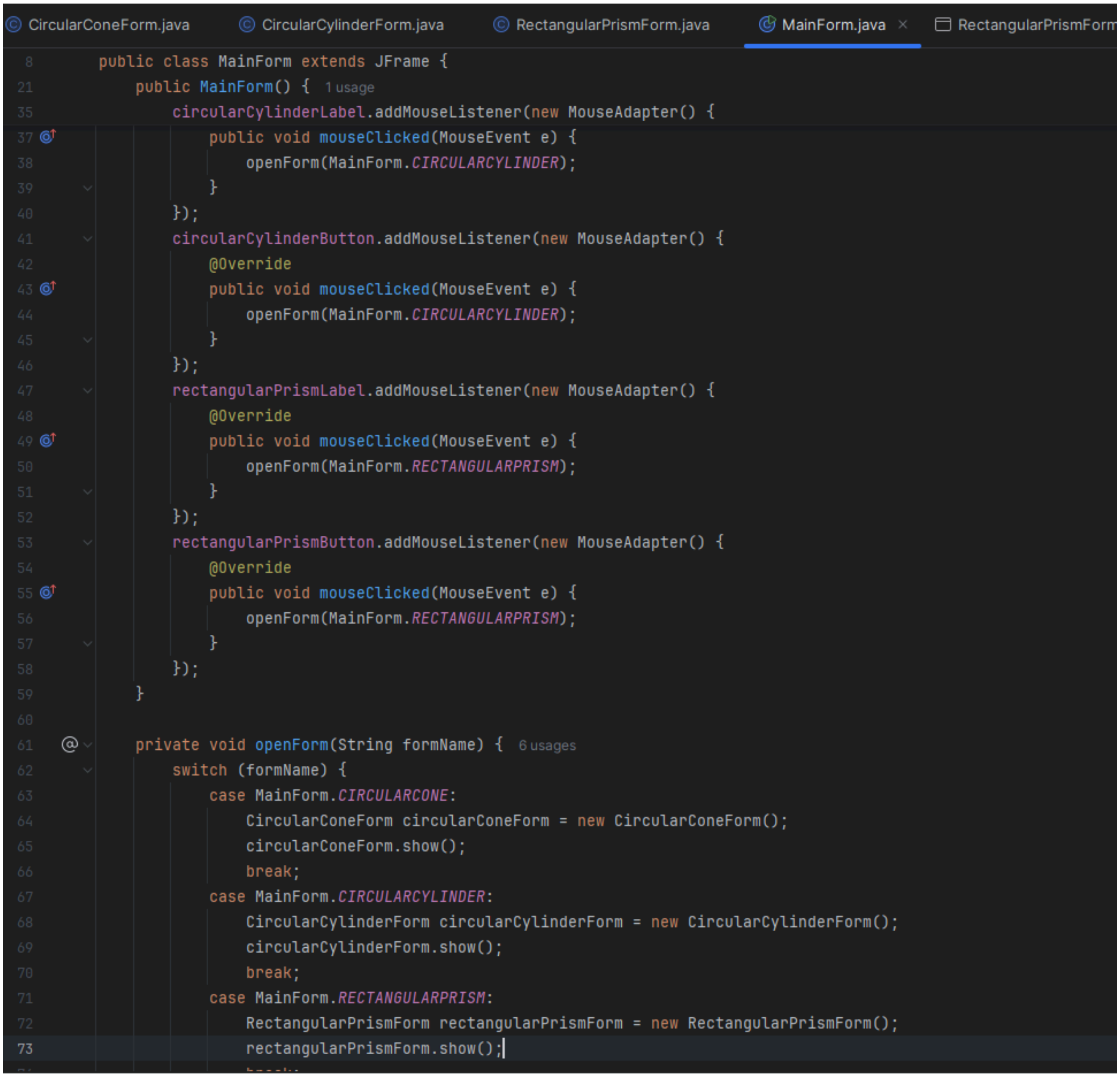


```
    }

    public void show() {  no usages
        frame.setVisible(true);
    }

    public void dispose() {  1 usage
        frame.setVisible(false);
        frame.dispose();
    }

    public void doCalculate() {  1 usage
        RectangularPrism rc = new RectangularPrism(
                Double.parseDouble(textWidth.getText()),
                Double.parseDouble(textHeight.getText()),
                Double.parseDouble(textLenght.getText())
        );
        double volume = rc.getVolume();
        double S = rc.getSurfaceArea();
        double dialogal = rc.getDiagonal();
        double totalSurfaceArea = rc.getTotalSurfaceArea();

        String result = "Volume = " + volume + "\n" +
                "Surface Area = " + S + "\n" +
                "Dialogal = " + dialogal + "\n" +
                "Total Surface Area = " + totalSurfaceArea;

        displayResult(result,  title: "Result of Circular Cone Shape", JOptionPane.INFORMATION_MESSAGE);
    }

    public void displayResult(String resultMsg, String title, int type) {  1 usage
        JOptionPane.showMessageDialog( parentComponent: null, resultMsg, title, type);
    }
}
```

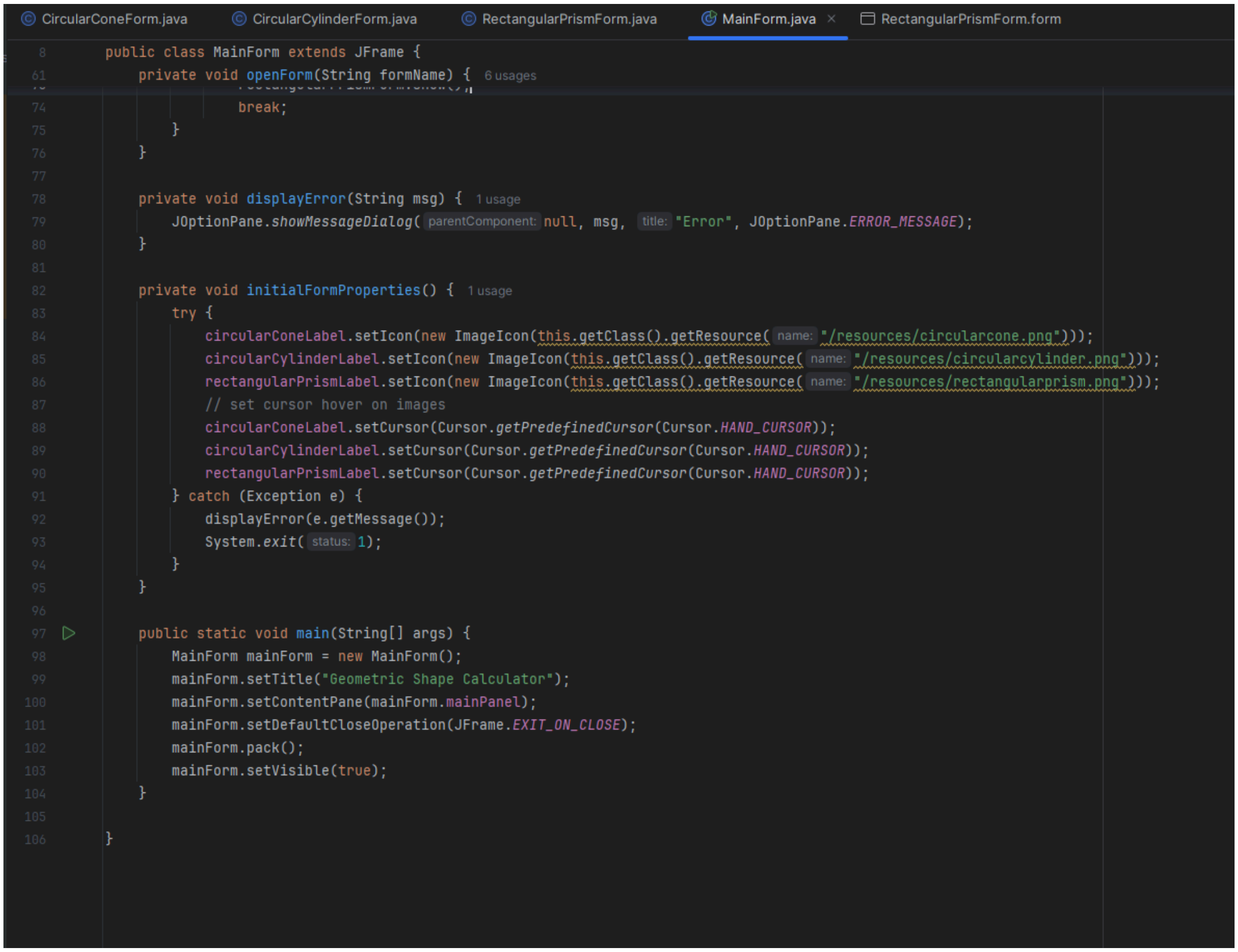

CircularConeForm.java
CircularCylinderForm.java
RectangularPrismForm.java
MainForm.java
RectangularPrismForm.form

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class MainForm extends JFrame {
    private JPanel mainPanel;  2 usages
    private JLabel circularConeLabel;  4 usages
    private JLabel circularCylinderLabel;  4 usages
    private JLabel rectangularPrismLabel;  4 usages
    private JButton circularConeButton;  2 usages
    private JButton circularCylinderButton;  2 usages
    private JButton rectangularPrismButton;  2 usages

    public static final String CIRCULARCONE = "CIRCULARCONE";  3 usages
    public static final String CIRCULARCYLINDER = "CIRCULARCYLINDER";  3 usages
    public static final String RECTANGULARPRISM = "RECTANGULARPRISM";  3 usages

    public MainForm() {  1 usage
        initialFormProperties();
        circularConeLabel.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                openForm(MainForm.CIRCULARCONE);
            }
        });
        circularConeButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                openForm(MainForm.CIRCULARCONE);
            }
        });
        circularCylinderLabel.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                openForm(MainForm.CIRCULARCYLINDER);
            }
        });
```

ต่อ



```java
public class MainForm extends JFrame {
    public MainForm() {
        circularCylinderLabel.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
                openForm(MainForm.CIRCULARCYLINDER);
            }
        });
        circularCylinderButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                openForm(MainForm.CIRCULARCYLINDER);
            }
        });
        rectangularPrismLabel.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                openForm(MainForm.RECTANGULARPRISM);
            }
        });
        rectangularPrismButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                openForm(MainForm.RECTANGULARPRISM);
            }
        });
    }

    private void openForm(String formName) {
        switch (formName) {
            case MainForm.CIRCULARCONE:
                CircularConeForm circularConeForm = new CircularConeForm();
                circularConeForm.show();
                break;
            case MainForm.CIRCULARCYLINDER:
                CircularCylinderForm circularCylinderForm = new CircularCylinderForm();
                circularCylinderForm.show();
                break;
            case MainForm.RECTANGULARPRISM:
                RectangularPrismForm rectangularPrismForm = new RectangularPrismForm();
                rectangularPrismForm.show();
```

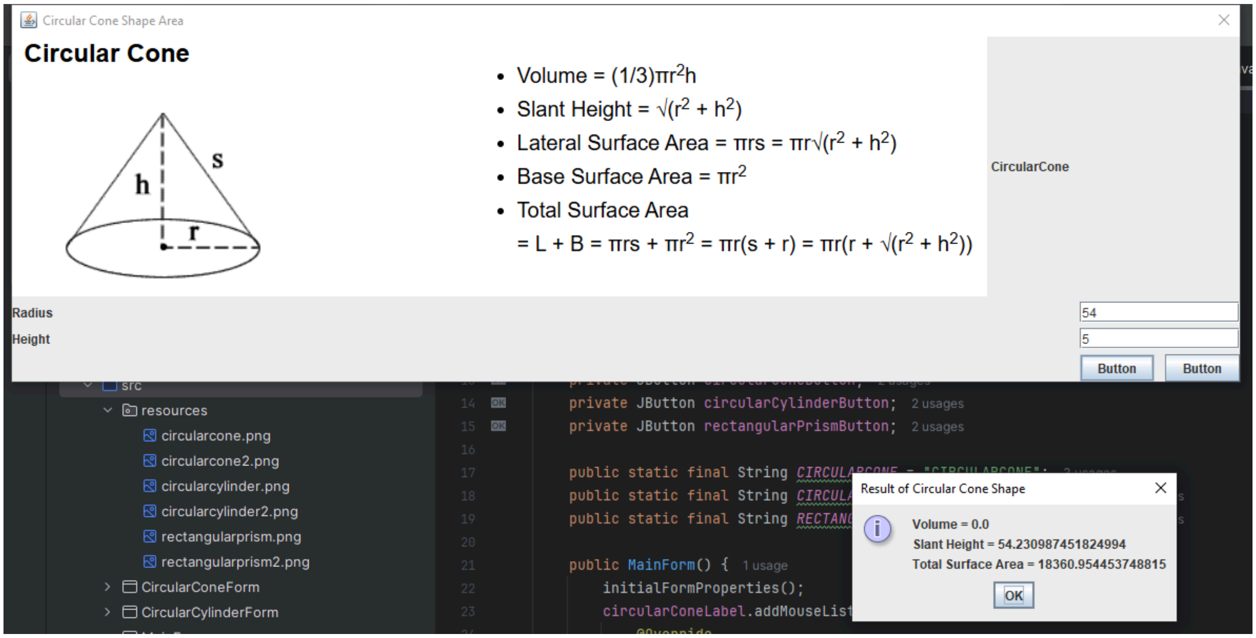

CircularConeForm.java
CircularCylinderForm.java
RectangularPrismForm.java
MainForm.java
RectangularPrismForm.form

```
public class MainForm extends JFrame {
    private void openForm(String formName) {  6 usages
            break;
        }
    }

    private void displayError(String msg) {  1 usage
        JOptionPane.showMessageDialog( parentComponent: null, msg,  title: "Error", JOptionPane.ERROR_MESSAGE);
    }

    private void initialFormProperties() {  1 usage
        try {
            circularConeLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/circularcone.png")));
            circularCylinderLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/circularcylinder.png")));
            rectangularPrismLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/rectangularprism.png")));
            // set cursor hover on images
            circularConeLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
            circularCylinderLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
            rectangularPrismLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        } catch (Exception e) {
            displayError(e.getMessage());
            System.exit( status: 1);
        }
    }

    public static void main(String[] args) {
        MainForm mainForm = new MainForm();
        mainForm.setTitle("Geometric Shape Calculator");
        mainForm.setContentPane(mainForm.mainPanel);
        mainForm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainForm.pack();
        mainForm.setVisible(true);
    }

}
```

<u>ผลการทำงาน</u>

// วางภาพตัวอย่างผลการทำงานของทั้ง 3 รูปทรง ตรงจุดนี้ (ตัวอย่างทำงานปกติ, ตัวอย่างเมื่อเกิดความผิดพลาด)

// ผลที่ได้จากการทดสอบ



Circular Cone Shape Area

Circular Cone

- Volume = $(1/3)\pi r^2 h$
- Slant Height = $\sqrt{(r^2 + h^2)}$
- Lateral Surface Area = $\pi rs = \pi r\sqrt{(r^2 + h^2)}$
- Base Surface Area = πr^2
- Total Surface Area
  = $L + B = \pi rs + \pi r^2 = \pi r(s + r) = \pi r(r + \sqrt{(r^2 + h^2)})$

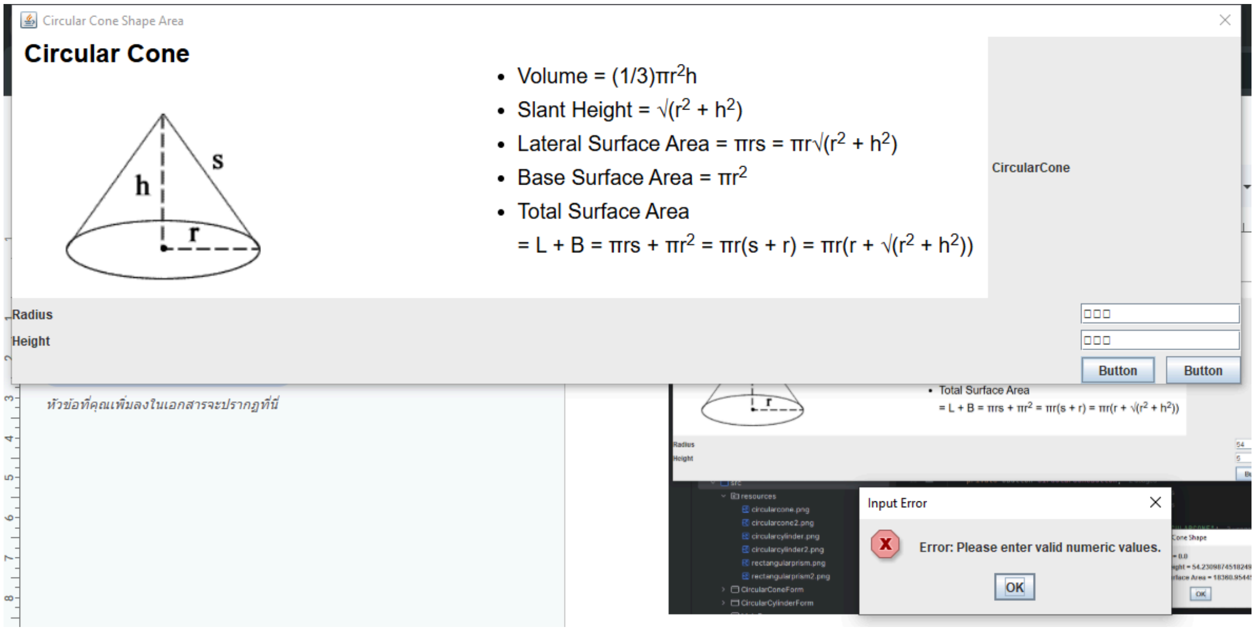
Radius 54

Height 5

```
    private JButton circularCylinderButton;  2 usages
    private JButton rectangularPrismButton;  2 usages

    public static final String CIRCULARCONE = "CIRCULARCONE";
    public static final String CIRCULA
    public static final String RECTANG

    public MainForm() {  1 usage
        initialFormProperties();
        circularConeLabel.addMouseList
```

Result of Circular Cone Shape

Volume = 0.0
Slant Height = 54.230987451824994
Total Surface Area = 18360.954453748815



Circular Cone Shape Area

Circular Cone

- Volume = $(1/3)\pi r^2 h$
- Slant Height = $\sqrt{(r^2 + h^2)}$
- Lateral Surface Area = $\pi rs = \pi r\sqrt{(r^2 + h^2)}$
- Base Surface Area = πr^2
- Total Surface Area
  = $L + B = \pi rs + \pi r^2 = \pi r(s + r) = \pi r(r + \sqrt{(r^2 + h^2)})$

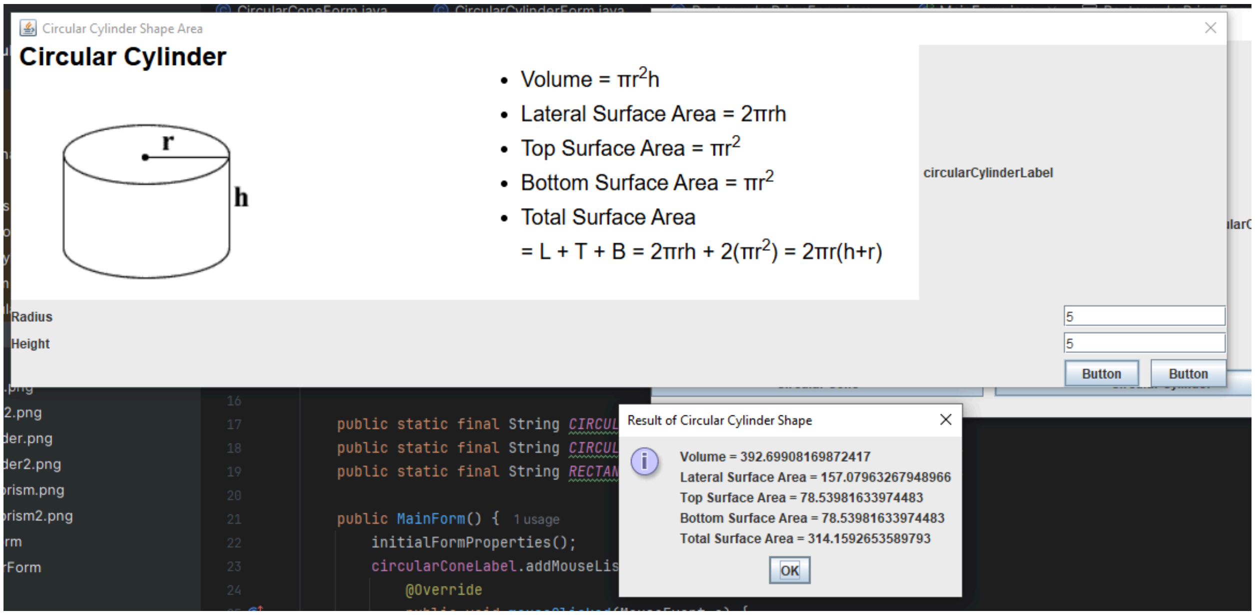
Radius □□□

Height □□□

หัวข้อที่คุณเพิ่มลงในเอกสารจะปรากฏที่นี่

Input Error

Error: Please enter valid numeric values.



Circular Cylinder Shape Area

Circular Cylinder

- Volume = $\pi r^2 h$
- Lateral Surface Area = $2\pi rh$
- Top Surface Area = πr^2
- Bottom Surface Area = πr^2
- Total Surface Area
  = L + T + B = $2\pi rh + 2(\pi r^2) = 2\pi r(h+r)$

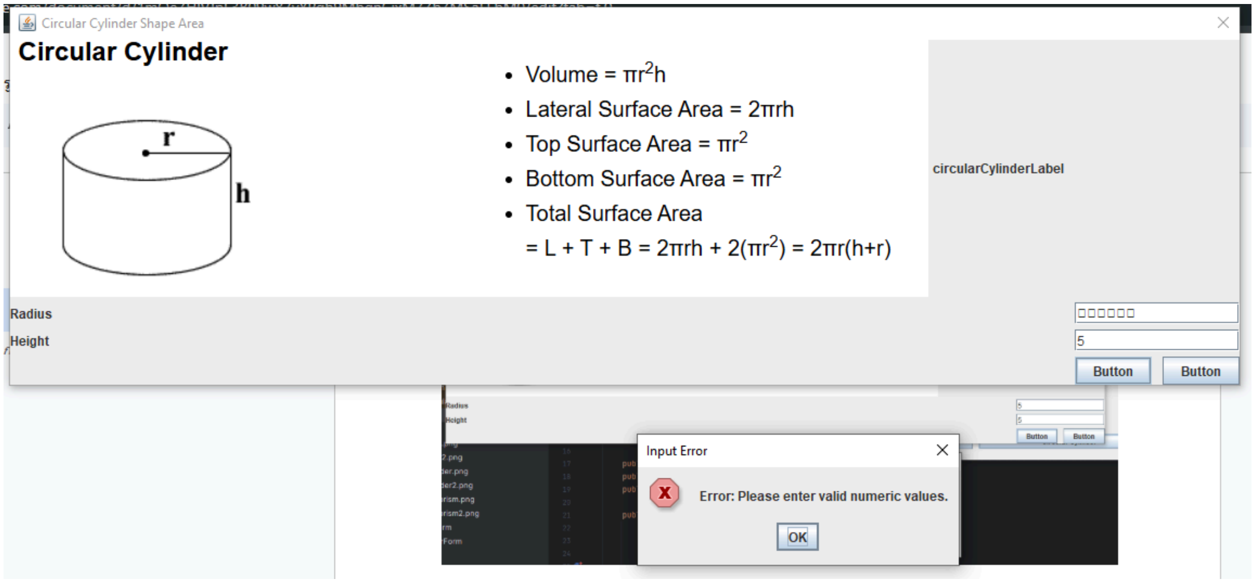
Radius: 5

Height: 5

```
public static final String CIRCUL
public static final String CIRCUL
public static final String RECTAN

public MainForm() {  1 usage
    initialFormProperties();
    circularConeLabel.addMouseLis
        @Override
```

Result of Circular Cylinder Shape

Volume = 392.69908169872417
Lateral Surface Area = 157.07963267948966
Top Surface Area = 78.53981633974483
Bottom Surface Area = 78.53981633974483
Total Surface Area = 314.1592653589793



Circular Cylinder Shape Area

Circular Cylinder

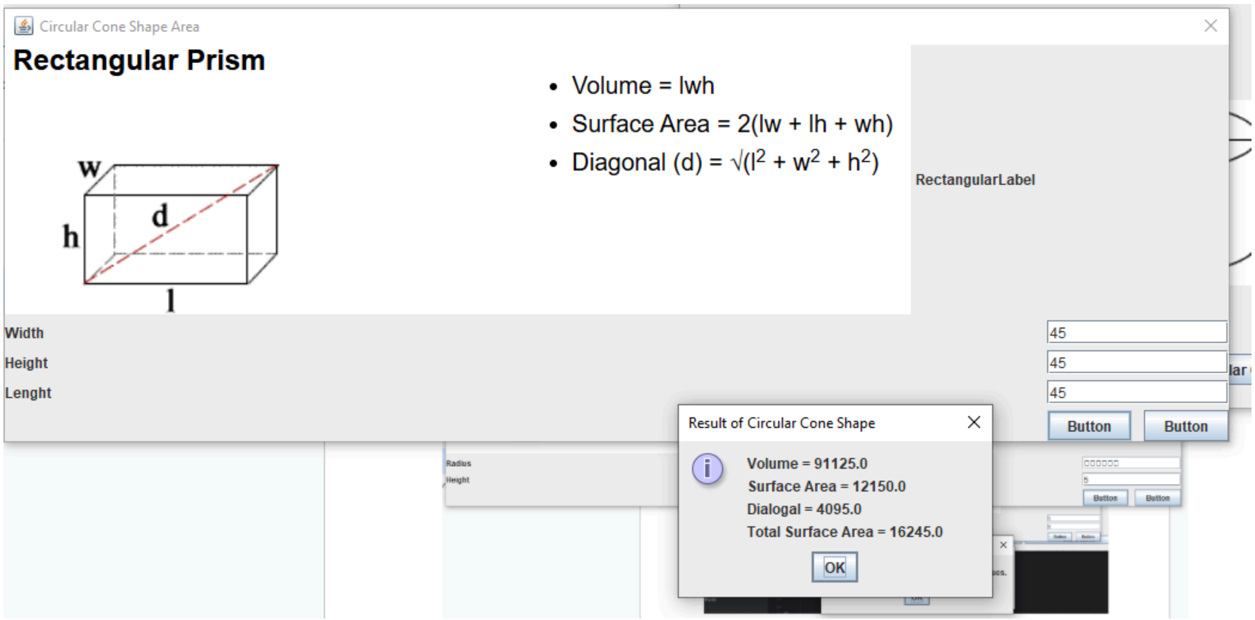
- Volume = $\pi r^2 h$
- Lateral Surface Area = $2\pi rh$
- Top Surface Area = πr^2
- Bottom Surface Area = πr^2
- Total Surface Area
  = L + T + B = $2\pi rh + 2(\pi r^2) = 2\pi r(h+r)$

Radius: □□□□□□

Height: 5

Input Error

Error: Please enter valid numeric values.



Circular Cone Shape Area

Rectangular Prism

- Volume = lwh
- Surface Area = 2(lw + lh + wh)
- Diagonal (d) = √(l^2 + w^2 + h^2)

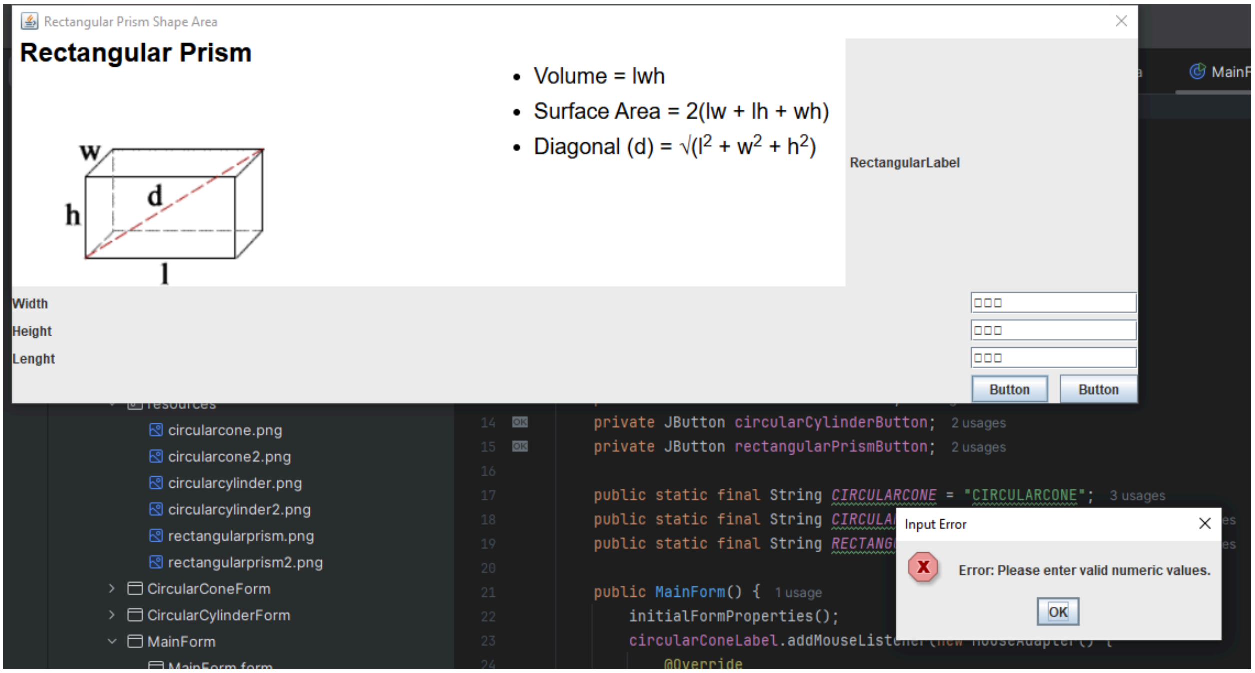
Width 45

Height 45

Lenght 45

Result of Circular Cone Shape

Volume = 91125.0
Surface Area = 12150.0
Dialogal = 4095.0
Total Surface Area = 16245.0



Rectangular Prism Shape Area

Rectangular Prism

- Volume = lwh
- Surface Area = 2(lw + lh + wh)
- Diagonal (d) = √(l^2 + w^2 + h^2)

Width □□□

Height □□□

Lenght □□□

```
private JButton circularCylinderButton;  2 usages
private JButton rectangularPrismButton;  2 usages

public static final String CIRCULARCONE = "CIRCULARCONE";  3 usages
public static final String CIRCULA
public static final String RECTANG

public MainForm() {  1 usage
    initialFormProperties();
    circularConeLabel.addMouseList
        @Override
```

Input Error

Error: Please enter valid numeric values.