

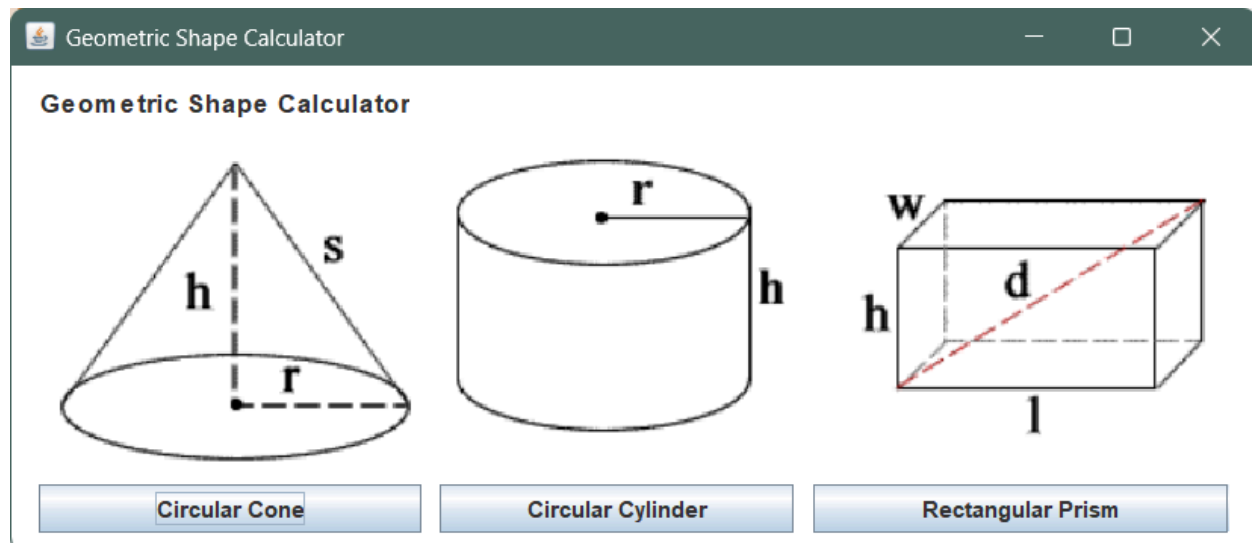
รหัส นิสิต	6621600950	ชื่อ/นามสกุล	นวพร สุขสวัสดิ์
---------------	------------	--------------	-----------------

## 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

```
① GeometricShapes.java × ② CircularCone.java ③ CircularCylinder.java ④ RectangularPrism.java  
1 package it.util.shapes;  
2  
3 public interface GeometricShapes { 3 usages 3 implementations new *  
4     public double getVolume(); 3 usages 3 implementations new *  
5     public double getTotalSurfaceArea(); 2 usages 3 implementations new *  
6 }  
7  
8
```

```

1 package it.util.shapes;
2
3 public class CircularCone implements GeometricShapes { new *
4     private double radius; 15 usages
5     private double height; 9 usages
6
7     public CircularCone(double r, double h) { 1 usage new *
8         this.radius = r;
9         this.height = h;
10    }
11
12    public double getVolume() { 3 usages new *
13        double volume = (1 / 3) * Math.PI * (radius * radius) * height;
14        return volume;
15    }
16
17    public double getTotalSurfaceArea() { 2 usages new *
18        double total = Math.PI * radius * (radius + Math.sqrt(radius * radius + height * height));
19        return total;
20    }
21
22    public double getLateralSurfaceArea() { 1 usage new *
23        double L = Math.PI * radius * Math.sqrt(radius * radius + height * height);
24        return L;
25    }
26
27    public double getBaseSurfaceArea() { 1 usage new *
28        double B = Math.PI * (radius * radius);
29        return B;
30    }
31
32    public double getSlantHeight() { 1 usage new *
33        double S = Math.sqrt(radius * radius + height * height);
34        return S;
35    }
36
37    public double getRadius() { no usages new *
38        return radius;
39    }
40
41    public double getHeight() { no usages new *
42        return height;

```

```

    }

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

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

    public static void main(String[] args) { new *
        CircularCone clc = new CircularCone(r: 4, h: 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());
    }
}

```

```

1 package it.util.shapes;
2
3 public class CircularCylinder implements GeometricShapes{ new *
4     private double radius; 14 usages
5     private double height; 5 usages
6
7     public CircularCylinder(double r,double h){ 1 usage new *
8         this.radius = r;
9         this.height = h;
10    }
11    public double getVolume() { 3 usages new *
12        double volume = Math.PI*(radius*radius)*height;
13        return volume;
14    }
15    public double getTotalSurfaceArea() { 2 usages new *
16        double total = (2*Math.PI*radius*height)+(Math.PI*(radius*radius))+(Math.PI*(radius*radius));
17        return total;
18    }
19    public double getLateralSurfaceArea() { 1 usage new *
20        double L = 2*Math.PI*radius*height;
21        return L;
22    }
23    public double getTopSurfaceArea() { 2 usages new *
24        double T = Math.PI*(radius*radius);
25        return T;
26    }
27    public double getBottomSurfaceArea() { 1 usage new *
28        double B = Math.PI*(radius*radius);
29        return B;
30    }
31    public double getRadius(){ no usages new *
32        return radius;
33    }
34    public double getHeight(){ no usages new *
35        return height;
36    }
37    public static void main(String[] args){ new *
38        CircularCylinder cc = new CircularCylinder(4, 6);
39        System.out.println("Volume = "+cc.getVolume());
40        System.out.println("Lateral Surface Area =" +cc.getLateralSurfaceArea());
41        System.out.println("Top Surface Area =" +cc.getTopSurfaceArea());
42        System.out.println("Bottom Surface Area =" +cc.getBottomSurfaceArea());

```

```

26     }
27     public double getBottomSurfaceArea() { 1 usage new *
28         double B = Math.PI*(radius*radius);
29         return B;
30     }
31     public double getRadius(){ no usages new *
32         return radius;
33     }
34     public double getHeight(){ no usages new *
35         return height;
36     }
37     public static void main(String[] args){ new *
38         CircularCylinder cc = new CircularCylinder(r: 4, h: 6);
39         System.out.println("Volume = "+cc.getVolume());
40         System.out.println("Lateral Surface Area ="+cc.getLateralSurfaceArea());
41         System.out.println("Top Surface Area ="+cc.getTopSurfaceArea());
42         System.out.println("Bottom Surface Area ="+cc.getBottomSurfaceArea());
43         System.out.println("Total Surface Area ="+cc.getTopSurfaceArea());
44     }
45
46
47 }
48

```

```

1 package it.util.shapes;
2
3 public class RectangularPrism implements GeometricShapes{ new *
4     private double width; 7 usages
5     private double height; 7 usages
6     private double length; 7 usages
7
8     public RectangularPrism(double w,double h,double l){ 1 usage new *
9         this.width = w;
10        this.height = h;
11        this.length = l;
12    }
13
14    @Override 3 usages new *
15    public double getVolume() {
16        double v = this.length*this.width*this.height;
17        return v;
18    }
19
20    @Override 2 usages new *
21    public double getTotalSurfaceArea() {
22        double ts = getDiagonal()+getSurfaceArea();
23        return ts;
24    }
25
26    public double getDiagonal(){ 2 usages new *
27        double d = Math.sqrt(this.length*this.length)+(this.width*this.width)+(this.height*this.height);
28        return d;
29    }
30
31    public double getSurfaceArea(){ 2 usages new *
32        double s = 2*((this.length*this.width)+(this.length*this.height)+(this.width*this.height));
33        return s;
34    }
35
36    public double getWidth(){ no usages new *
37        return width;
38    }
39
40    public double getHeight(){ no usages new *
41        return height;
42    }
43
44    public double getLength() { no usages new *
45        return length;
46    }
47
48 }

```

ต่อ

```
33  public double getWidth(){ no usages new *  
34      return width;  
35  }  
36  public double getHeight(){ no usages new *  
37      return height;  
38  }  
39  public double getLength() { no usages new *  
40      return length;  
41  }  
42  
43  public static void main(String[] args){ new *  
44      RectangularPrism rp = new RectangularPrism(w: 5, h: 2, l: 4);  
45      System.out.println("Volume = "+rp.getVolume());  
46      System.out.println("Surface Area = "+rp.getSurfaceArea());  
47      System.out.println("Diagonal = "+rp.getDiagonal());  
48      System.out.println("Total Surface Area = "+rp.getTotalSurfaceArea());  
49  }  
50  }
```

// 2. Code Java ในโปรเจค GeometricShapeCalculator



```

CircularConeForm.java x CircularCylinderForm.java RectangularPrismForm.java MainForm.java RectangularPrismForm.form
1  import it.util.shapes.CircularCone;
2
3  import javax.swing.*;
4  import java.awt.*;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7
8  public class CircularConeForm { 4 usages
9      private JLabel circularConeLabel; 2 usages
10     private JButton calculateButton; 2 usages
11     private JButton closeButton; 2 usages
12     private JPanel circularConePanel; 2 usages
13     private JTextField textRadius; 2 usages
14     private JTextField textHeight; 2 usages
15
16     private JDialog frame; 7 usages
17
18     public CircularConeForm() { 1 usage
19         circularConeLabel.setIcon(new ImageIcon(this.getClass().getResource("/resources/circularcone2.png")));
20         frame = new JDialog((Frame) null, title: "Circular Cone Shape Area", modal: true);
21         frame.setContentPane(circularConePanel);
22         frame.pack();
23         frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
24
25         closeButton.addActionListener(new ActionListener() {
26             @Override
27             public void actionPerformed(ActionEvent e) { dispose(); }
28         });
29
30         calculateButton.addActionListener(new ActionListener() {
31             @Override
32             public void actionPerformed(ActionEvent e) { doCalculate(); }
33         });
34     }
35
36     public void show() { frame.setVisible(true); }
37
38     public void dispose() { 1 usage
39         frame.setVisible(false);
40         frame.dispose();
41     }
42 }

```

```

44 public void dispose() { 1 usage
45     frame.setVisible(false);
46     frame.dispose();
47 }
48
49 public void doCalculate() { 1 usage
50     CircularCone cc = new CircularCone(Double.parseDouble(textRadius.getText()),
51         Double.parseDouble(textHeight.getText()));
52     double volume = cc.getVolume();
53     double slantHeight = cc.getSlantHeight();
54     double totalSurfaceArea = cc.getTotalSurfaceArea();
55
56     String result = "Volume = " + volume + "\n" +
57         "Slant Height = " + slantHeight + "\n" +
58         "Total Surface Area = " + totalSurfaceArea;
59
60     displayResult(result, title: "Result of Circular Cone Shape", JOptionPane.INFORMATION_MESSAGE);
61 }
62
63 public void displayResult(String resultMsg, String title, int type) { 1 usage
64     JOptionPane.showMessageDialog(parentComponent: null, resultMsg, title, type);
65 }
66 }

```

```

1  import it.util.shapes.CircularCylinder;
2
3  import javax.swing.*;
4  import java.awt.*;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7
8  public class CircularCylinderForm { 4 usages
9      private JLabel circularCylinderLabel; 2 usages
10     private JButton calculateButton; 2 usages
11     private JButton closeButton; 2 usages
12     private JPanel circularCylinderPanel; 2 usages
13     private JTextField textRadius; 2 usages
14     private JTextField textHeight; 2 usages
15
16     private JDialog frame; 7 usages
17
18     public CircularCylinderForm() { 1 usage
19         circularCylinderLabel.setIcon(new ImageIcon(this.getClass().getResource(name: "/resources/circularcylinder2.png")));
20         frame = new JDialog((Frame) null, title: "Circular Cylinder Shape Area", modal: true);
21         frame.setContentPane(circularCylinderPanel);
22         frame.pack();
23         frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
24
25         closeButton.addActionListener(new ActionListener() {
26             @Override
27             public void actionPerformed(ActionEvent e) { dispose(); }
28         });
29
30         calculateButton.addActionListener(new ActionListener() {
31             @Override
32             public void actionPerformed(ActionEvent e) { doCalculate(); }
33         });
34     }
35
36     public void show() { 1 usage
37         frame.setVisible(true);
38     }
39 }

```

```

59
60     public void show() { 1 usage
61         frame.setVisible(true);
62     }
63
64     public void dispose() { 1 usage
65         frame.setVisible(false);
66         frame.dispose();
67     }
68
69     public void doCalculate() { 1 usage
70         CircularCylinder cc = new CircularCylinder(Double.parseDouble(textRadius.getText()),
71             Double.parseDouble(textHeight.getText()));
72         double volume = cc.getVolume();
73         double L = cc.getLateralSurfaceArea();
74         double T = cc.getTopSurfaceArea();
75         double B = cc.getBottomSurfaceArea();
76         double totalSurfaceArea = cc.getTotalSurfaceArea();
77
78         String result = "Volume = " + volume + "\n" +
79             "Lateral Surface Area = " + L + "\n" +
80             "Top Surface Area = " + T + "\n" +
81             "Bottom Surface Area = " + B + "\n" +
82             "Total Surface Area = " + totalSurfaceArea;
83
84         displayResult(result, title: "Result of Circular Cylinder Shape", JOptionPane.INFORMATION_MESSAGE);
85     }
86
87     public void displayResult(String resultMsg, String title, int type) { 1 usage
88         JOptionPane.showMessageDialog( parentComponent: null, resultMsg, title, type);
89     }
90 }

```

```

CircularConeForm.java  CircularCylinderForm.java  RectangularPrismForm.java  MainForm.java  RectangularPrismForm.form
1  import it.util.shapes.RectangularPrism;
2
3  import javax.swing.*;
4  import java.awt.*;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7
8  public class RectangularPrismForm { 4 usages
9      private JLabel RectangularLabel; 2 usages
10     private JButton calculateButton; 2 usages
11     private JButton closeButton; 2 usages
12     private JPanel RectangularPanel; 2 usages
13     private JTextField textWidth; 2 usages
14     private JTextField textHeight; 2 usages
15     private JTextField textLength; 2 usages
16
17     private JDialog frame; 7 usages
18
19     public RectangularPrismForm() { 1 usage
20         RectangularLabel.setIcon(new ImageIcon(this.getClass().getResource("/resources/rectangularprism2.png")));
21         frame = new JDialog((Frame) null, title: "Circular Cone Shape Area", modal: true);
22         frame.setContentPane(RectangularPanel);
23         frame.pack();
24         frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
25
26         closeButton.addActionListener(new ActionListener() {
27             @Override
28             public void actionPerformed(ActionEvent e) { dispose(); }
29         });
30
31         calculateButton.addActionListener(new ActionListener() {
32             @Override
33             public void actionPerformed(ActionEvent e) { doCalculate(); }
34         });
35     }
36
37     public void show() { no usages
38         frame.setVisible(true);
39     }
40
41 }

```

```

1 public void show() { no usages
2     frame.setVisible(true);
3 }
4
5 public void dispose() { 1 usage
6     frame.setVisible(false);
7     frame.dispose();
8 }
9
10 public void doCalculate() { 1 usage
11     RectangularPrism rc = new RectangularPrism(
12         Double.parseDouble(textWidth.getText()),
13         Double.parseDouble(textHeight.getText()),
14         Double.parseDouble(textLength.getText())
15     );
16     double volume = rc.getVolume();
17     double S = rc.getSurfaceArea();
18     double dialogal = rc.getDiagonal();
19     double totalSurfaceArea = rc.getTotalSurfaceArea();
20
21     String result = "Volume = " + volume + "\n" +
22         "Surface Area = " + S + "\n" +
23         "Dialogal = " + dialogal + "\n" +
24         "Total Surface Area = " + totalSurfaceArea;
25
26     displayResult(result, title: "Result of Circular Cone Shape", JOptionPane.INFORMATION_MESSAGE);
27 }
28
29 public void displayResult(String resultMsg, String title, int type) { 1 usage
30     JOptionPane.showMessageDialog(parentComponent: null, resultMsg, title, type);
31 }
32 }

```

```

1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5  import java.awt.event.MouseAdapter;
6  import java.awt.event.MouseEvent;
7
8  public class MainForm extends JFrame {
9      private JPanel mainPanel; 2 usages
10     private JLabel circularConeLabel; 4 usages
11     private JLabel circularCylinderLabel; 4 usages
12     private JLabel rectangularPrismLabel; 4 usages
13     private JButton circularConeButton; 2 usages
14     private JButton circularCylinderButton; 2 usages
15     private JButton rectangularPrismButton; 2 usages
16
17     public static final String CIRCULARCONE = "CIRCULARCONE"; 3 usages
18     public static final String CIRCULARCYLINDER = "CIRCULARCYLINDER"; 3 usages
19     public static final String RECTANGULARPRISM = "RECTANGULARPRISM"; 3 usages
20
21     public MainForm() { 1 usage
22         initialFormProperties();
23         circularConeLabel.addMouseListener(new MouseAdapter() {
24             @Override
25             public void mouseClicked(MouseEvent e) {
26                 openForm(MainForm.CIRCULARCONE);
27             }
28         });
29         circularConeButton.addActionListener(new ActionListener() {
30             @Override
31             public void actionPerformed(ActionEvent e) {
32                 openForm(MainForm.CIRCULARCONE);
33             }
34         });
35         circularCylinderLabel.addMouseListener(new MouseAdapter() {
36             @Override
37             public void mouseClicked(MouseEvent e) {
38                 openForm(MainForm.CIRCULARCYLINDER);
39             }
40         });

```

```

8      public class MainForm extends JFrame {
21          public MainForm() { 1 usage
35              circularCylinderLabel.addMouseListener(new MouseAdapter() {
37  @
38                  public void mouseClicked(MouseEvent e) {
39                      openForm(MainForm.CIRCULARCYLINDER);
40                  }
41              });
42              circularCylinderButton.addMouseListener(new MouseAdapter() {
43  @
44                  @Override
45                  public void mouseClicked(MouseEvent e) {
46                      openForm(MainForm.CIRCULARCYLINDER);
47                  }
48              });
49  @
50              rectangularPrismLabel.addMouseListener(new MouseAdapter() {
51                  @Override
52                  public void mouseClicked(MouseEvent e) {
53                      openForm(MainForm.RECTANGULARPRISM);
54                  }
55              });
56              rectangularPrismButton.addMouseListener(new MouseAdapter() {
57                  @Override
58                  public void mouseClicked(MouseEvent e) {
59                      openForm(MainForm.RECTANGULARPRISM);
60                  }
61              });
62          }
63
64  @
65  private void openForm(String formName) { 6 usages
66      switch (formName) {
67          case MainForm.CIRCULARCONE:
68              CircularConeForm circularConeForm = new CircularConeForm();
69              circularConeForm.show();
70              break;
71          case MainForm.CIRCULARCYLINDER:
72              CircularCylinderForm circularCylinderForm = new CircularCylinderForm();
73              circularCylinderForm.show();
74              break;
75          case MainForm.RECTANGULARPRISM:
76              RectangularPrismForm rectangularPrismForm = new RectangularPrismForm();
77              rectangularPrismForm.show();
78          }
79      }
80  }

```



```

8      public class MainForm extends JFrame {
61      private void openForm(String formName) { 6 usages
74          break;
75      }
76  }
77
78  private void displayError(String msg) { 1 usage
79      JOptionPane.showMessageDialog( parentComponent: null, msg, title: "Error", JOptionPane.ERROR_MESSAGE);
80  }
81
82  private void initialFormProperties() { 1 usage
83      try {
84          circularConeLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/circularcone.png")));
85          circularCylinderLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/circularcylinder.png")));
86          rectangularPrismLabel.setIcon(new ImageIcon(this.getClass().getResource( name: "/resources/rectangularprism.png")));
87          // set cursor hover on images
88          circularConeLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
89          circularCylinderLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
90          rectangularPrismLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
91      } catch (Exception e) {
92          displayError(e.getMessage());
93          System.exit( status: 1);
94      }
95  }
96
97  public static void main(String[] args) {
98      MainForm mainForm = new MainForm();
99      mainForm.setTitle("Geometric Shape Calculator");
100     mainForm.setContentPane(mainForm.mainPanel);
101     mainForm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
102     mainForm.pack();
103     mainForm.setVisible(true);
104 }
105
106 }

```

## ผลการทำงาน

// วางภาพตัวอย่างผลการทำงานของทั้ง 3 รูปทรง ตรงจุดนี้ (ตัวอย่างทำงานปกติ, ตัวอย่างเมื่อเกิดความผิดพลาด)

// ผลที่ได้จากการทดสอบ

Circular Cone Shape Area

# Circular Cone

Radius

Height

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

$$= L + B = \pi r s + \pi r^2 = \pi r (s + r) = \pi r (r + \sqrt{r^2 + h^2})$$

CircularCone

Button

Button

src

resources

circularcone.png
circularcone2.png
circularcylinder.png
circularcylinder2.png
rectangularprism.png
rectangularprism2.png

CircularConeForm
CircularCylinderForm

```

private JButton circularConeButton;
private JButton circularCylinderButton;
private JButton rectangularPrismButton;

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

public MainForm() {
    initialFormProperties();
    circularConeLabel.addMouseListener(

```

Result of Circular Cone Shape

Volume = 0.0

Slant Height = 54.230987451824994

Total Surface Area = 18360.954453748815

OK

Circular Cone Shape Area

# Circular Cone

Radius

Height

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

CircularCone

Button

Button

หัวข้อที่คุณเพิ่มลงในเอกสารจะปรากฏที่นี่

Circular Cone Shape Area

Radius

Height

- Total Surface Area  
=  $L + B = \pi r s + \pi r^2 = \pi r (s + r) = \pi r (r + \sqrt{r^2 + h^2})$

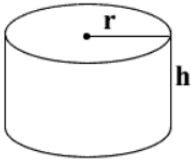
resources

- circularcone.png
- circularcone2.png
- circularcylinder.png
- circularcylinder2.png
- rectangularprism.png
- rectangularprism2.png
- CircularConeForm
- CircularCylinderForm

Input Error

Error: Please enter valid numeric values.

OK



### Circular Cylinder

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

Radius

Height

5

5

Button

Button

16

17

18

19

20

21

22

23

24

```

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

public MainForm() { 1 usage
    initialFormProperties();
    circularCylinderLabel.addMouseLis
    @Override
    public void mouseClicked(MouseEvent e) {

```

Result of Circular Cylinder Shape

Volume = 392.69908169872417

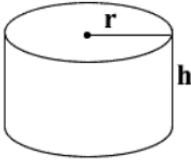
Lateral Surface Area = 157.07963267948966

Top Surface Area = 78.53981633974483

Bottom Surface Area = 78.53981633974483

Total Surface Area = 314.1592653589793

OK



### Circular Cylinder

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

Radius

Height

5

5

Button

Button

16

17

18

19

20

21

22

23

24

```

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

public MainForm() { 1 usage
    initialFormProperties();
    circularCylinderLabel.addMouseLis
    @Override
    public void mouseClicked(MouseEvent e) {

```

Input Error

Error: Please enter valid numeric values.

OK

**Rectangular Prism**

- Volume =  $lwh$
- Surface Area =  $2(lw + lh + wh)$
- Diagonal (d) =  $\sqrt{l^2 + w^2 + h^2}$

Width

Height

Lenght

45

45

45

Button

Button

Result of Circular Cone Shape

Volume = 91125.0

Surface Area = 12150.0

Dialogal = 4095.0

Total Surface Area = 16245.0

OK

**Rectangular Prism**

- Volume =  $lwh$
- Surface Area =  $2(lw + lh + wh)$
- Diagonal (d) =  $\sqrt{l^2 + w^2 + h^2}$

Width

Height

Lenght

Button

Button

resources

circularcone.png

circularcone2.png

circularcylinder.png

circularcylinder2.png

rectangularprism.png

rectangularprism2.png

CircularConeForm

CircularCylinderForm

MainForm

MainForm form

14

15

16

17

18

19

20

21

22

23

24

private JButton circularCylinderButton;

private JButton rectangularPrismButton;

public static final String CIRCULARCONE = "CIRCULARCONE";

public static final String CIRCULA

public static final String RECTANG

public MainForm() {

initialFormProperties();

circularConeLabel.addMouseListener(new MouseAdapter() {

@Override

Input Error

Error: Please enter valid numeric values.

OK