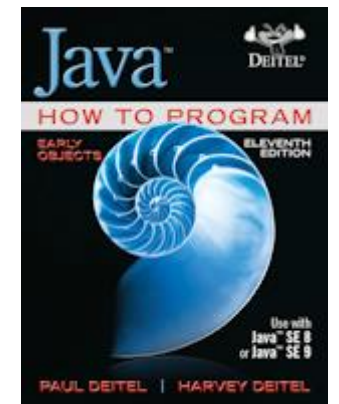


# Guia de Laboratório

ECOP15 - Lab 05

Prof. André Bernardi

[andrebernardi@unifei.edu.br](mailto:andrebernardi@unifei.edu.br)





# 5º Laboratório ECOP15

## 04 e 05 de outubro de 2021

---

- **Utilizar como referência os exemplos:**
  - **RadioButtonFrame.java:** Como criar JButtons e ButtonGroup, em java;
  - **RadioButtonTest.java:** JFrame com JButtons;

# 5º Laboratório ECOP15

04 e 05 de setembro de 2021

- **ShapesJPanel.java**: Utilizar o método fill para preencher formas;
- **Shapes.java**: JFrame com o método fill;
- **Shapes2JPanel.java**: Criando formas com GeneralPath;
- **Shapes2.java**: JFrame com GeneralPath;



# 1ª Tarefa: (1A)

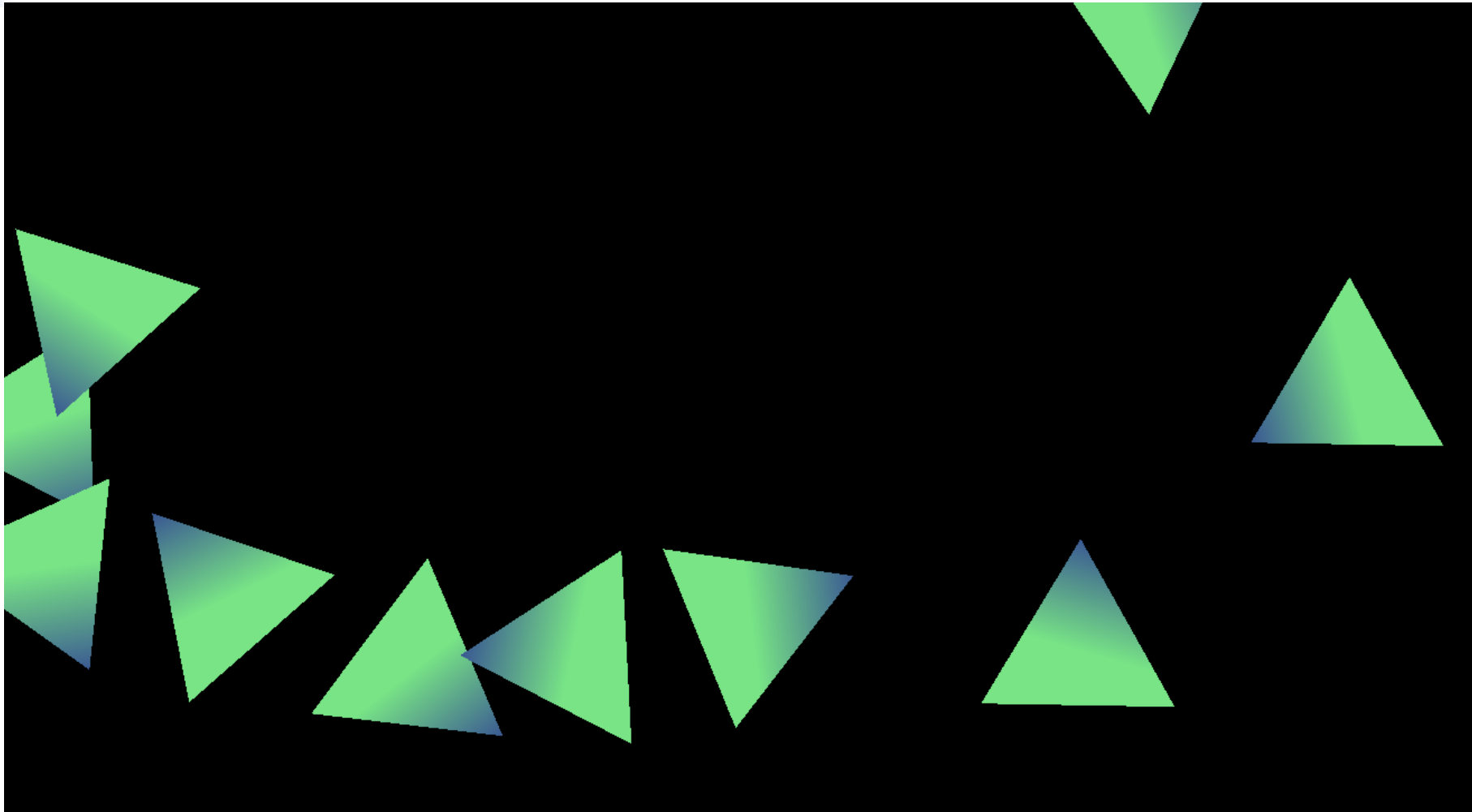
---

- Escreva um programa que mostre em posições aleatórias do fundo da janela triângulos. Utilize a classe *GeneralPath* e o método *fill* da classe *Graphics2D* para desenhar os triângulos.
- Torne este programa um pseudo-protetor de tela usando as funções para desenho de uma janela em full-screen e registrando os eventos do mouse e teclado para encerrar o aplicativo.



# 1ª Tarefa:

---



```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import java.util.Random;
import javax.swing.*;

public class Protetor extends JPanel { // painel para desenhar triângulos

    GeneralPath poligono;
    int raio = 100;

    public Protetor() {
        poligono = new GeneralPath();
        int x, y, forma = 3;

        // construir o triângulos equilátero
        poligono.moveTo(0, -raio);

        for (int i = 1; i < forma; i++) {

            x = (int) (raio * Math.cos(2 * Math.PI / forma * i));
            y = (int) (raio * Math.sin(2 * Math.PI / forma * i));
            poligono.lineTo(x, y);
        }
        poligono.closePath();
    }
}
```

```

    Timer t = new Timer(500, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            repaint();
        }
    });

    t.start();

    // registrar o evento do mouse para fechar o programa
    addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent e) {
            System.exit(0);
        }
    });
    setBackground(Color.BLACK);
}

```

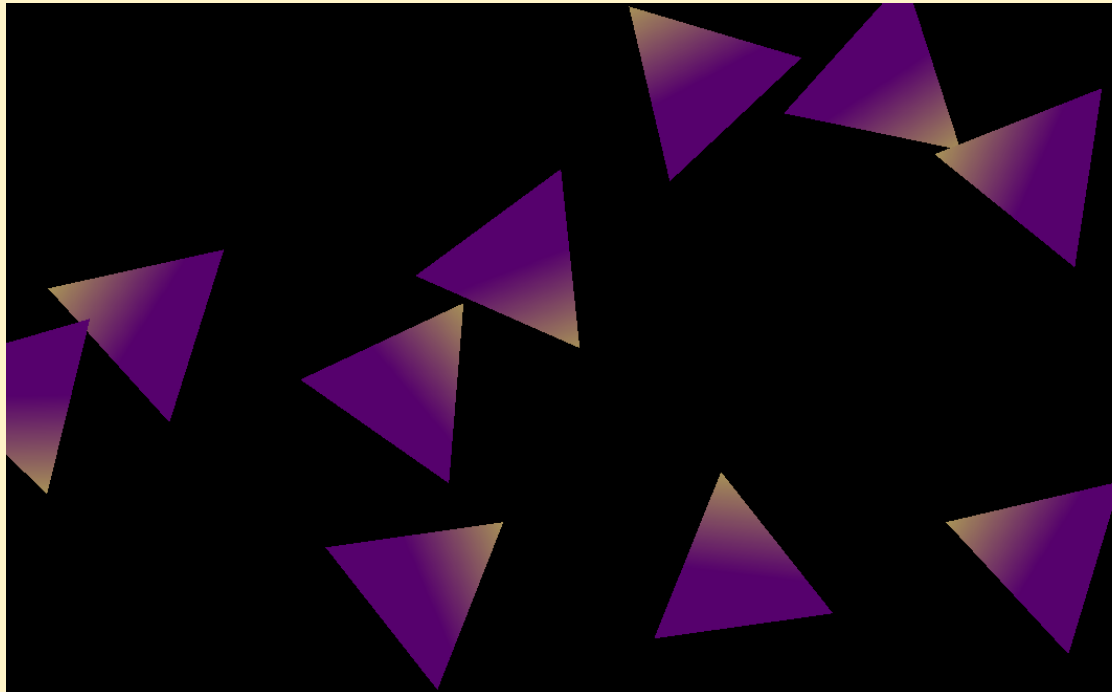
```

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    Random rand = new Random();
    Point p = new Point();
    Graphics2D g2d = (Graphics2D) g;
    Color color1 = new Color(rand.nextInt(256),
                             rand.nextInt(256), rand.nextInt(256));
    Color color2 = new Color(rand.nextInt(256),
                             rand.nextInt(256), rand.nextInt(256));
}

```

```
// desenha 10 triângulos em posições aleatórias
for(int k = 0; k < 10 ; k++){
    p.x = rand.nextInt(getWidth() - raio);
    p.y = rand.nextInt(getHeight() - raio);

    g2d.translate(p.x , p.y );
    g2d.rotate(p.x*Math.PI/180);
    g2d.setPaint(new GradientPaint(0, 0, color1, raio, raio, color2));
    g2d.fill(poligono);
    g2d.rotate(-1*(Math.toRadians(p.x)));
    g2d.translate(-p.x, -p.y);
}
}
```





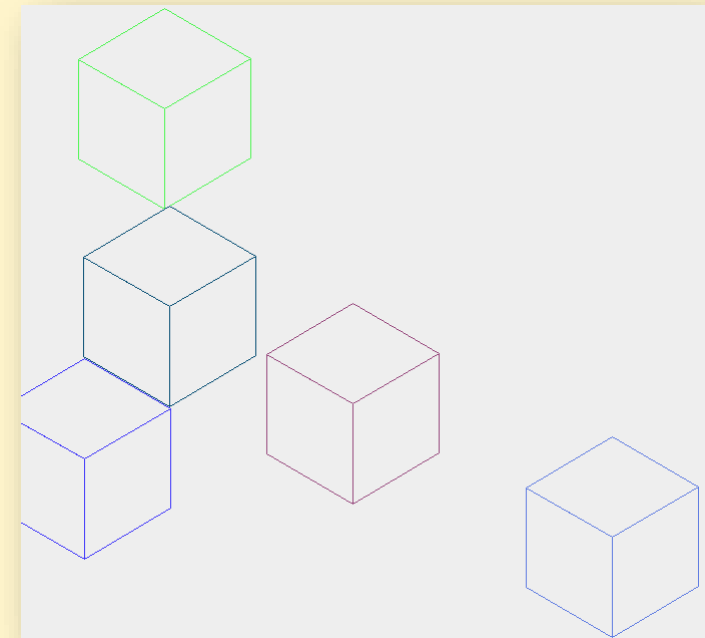
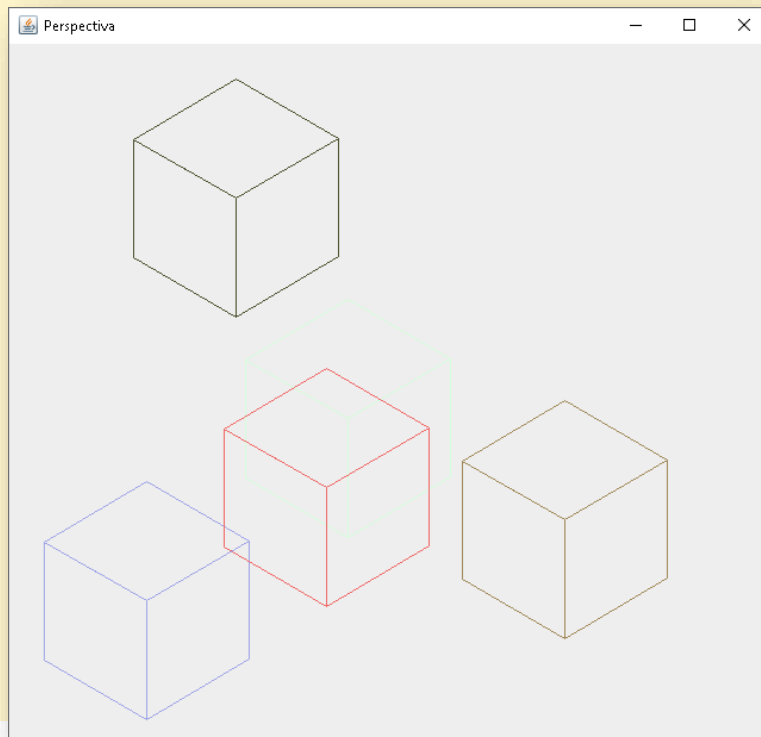
```
// Aplicativo de teste que exibe triângulos em tela cheia.
import java.awt.Dimension;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ProtetorDeTela {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Protetor de tela");
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension D = tk.getScreenSize();
        frame.setSize(D);
        frame.add(new Protetor());
        frame.setUndecorated(true);
        frame.setVisible(true);

        // Registrar evento do teclado para fechar o programa.
        frame.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent e) {
                System.exit(0);
            }
        });
    }
} // fim da classe ProtetorDeTela
```

## 2ª Tarefa: (1B)

- Escreva um programa que utilize a classe *GeneralPath* e o método *draw* da classe *Graphics2D* para desenhar um cubo em perspectiva isométrica.



```
import java.awt.*;
import java.awt.geom.*;
import javax.swing.*;
import java.util.*;

public class Perspectiva extends JPanel {
    GeneralPath poligono;
    int raio = 100;

    public Perspectiva() {
        poligono = new GeneralPath();
        int x, y, forma = 6;
        // montar o hexágono
        poligono.moveTo(0, -raio);
        for (int i = 1; i < forma; i++) {
            x = (int) (raio * Math.cos(-Math.PI / 2 + 2 * Math.PI / forma * i));
            y = (int) (raio * Math.sin(-Math.PI / 2 + 2 * Math.PI / forma * i));
            poligono.lineTo(x, y);
            if (i % 2 == 1) { //interligar o centro ao vértice par
                poligono.lineTo(0, 0);
                poligono.lineTo(x, y);
            }
        }
        poligono.closePath();
    }
}
```

```

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    Random rand = new Random();
    Point p = new Point();
    Graphics2D g2d = (Graphics2D) g;

    Color color1 = new Color(rand.nextInt(256), rand.nextInt(256), rand.nextInt(256));
    Color color2 = new Color(rand.nextInt(256), rand.nextInt(256), rand.nextInt(256));

    for(int k = 0; k < 5 ; k++){
        p.x = rand.nextInt(getWidth() - raio);
        p.y = rand.nextInt(getHeight() - raio);

        g2d.translate(p.x , p.y );
        //g2d.rotate(p.x*Math.PI/180);
        g2d.setColor(new Color(rand.nextInt( 256 ),
                                rand.nextInt( 256 ), rand.nextInt( 256 )));
        //g2d.setPaint(new GradientPaint(0, 0, color1, raio, raio, color2));
        g2d.draw(poligono);
        //g2d.rotate(-1*(Math.toRadians(p.x)));
        g2d.translate(-p.x, -p.y);
    }
}
} // end class

```



# Janela para o programa

```
import javax.swing.*;

public class MainPerspectiva {

    public static void main(String[] args) {

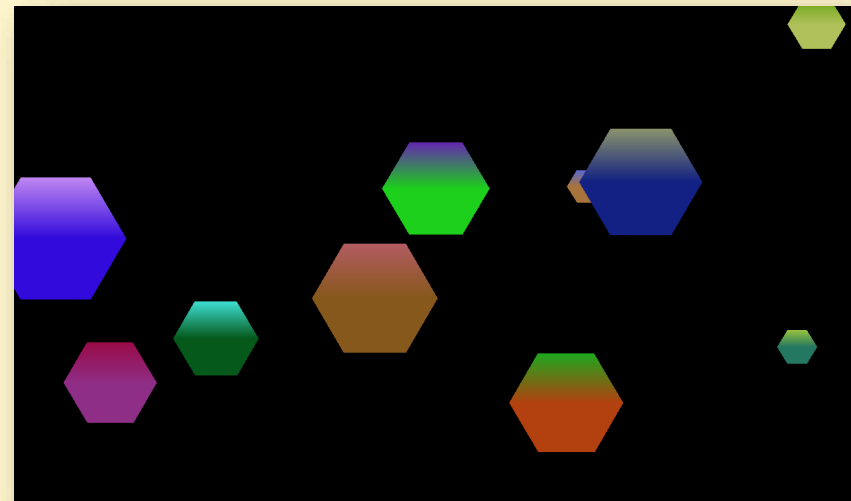
        JFrame frame = new JFrame("Perspectiva");
        frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );

        frame.add( new Perspectiva() );

        frame.setSize(1024,768);
        frame.setVisible(true);
    }
}
```

## 3ª Tarefa: (1C)

Escreva um programa que mostre em posições aleatórias do fundo da janela Hexágonos regulares, com tamanhos aleatórios. Utilize a classe GeneralPath e o método fill da classe Graphics2D para desenhar os Hexágonos regulares, preenchidos com um gradiente de cores aleatórias. Deixe esse programa funcionar em modo Tela-Cheia.



```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import java.util.Random;
import javax.swing.*;

public class Hexagono extends JPanel {

    public Hexagono() {
        // timer para atualizar a janela
        Timer t = new Timer(250, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                repaint();
            }
        });
        t.start();

        // registrar eventodouse para fechar janela
        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                System.exit(0);
            }
        });
        setBackground(Color.BLACK);
    }
}
```

```

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    Random rand = new Random();
    Point p = new Point();
    int raio, forma = 6;
    Graphics2D g2d = (Graphics2D) g;
    for(int k = 0; k < 10 ; k++){
        raio = 30 + rand.nextInt(80);
        //forma = 3 + rand.nextInt(3); // sortear forma a ser desenhada
        GeneralPath poligono = montaForma(forma, raio); // montar o hexagono

        // sortear posição que vai aparecer na tela
        p.x = rand.nextInt(getWidth() - raio );
        p.y = rand.nextInt(getHeight() - raio );

        Color color1 = new Color(rand.nextInt( 256 ),rand.nextInt( 256 ),rand.nextInt( 256 ));
        Color color2 = new Color(rand.nextInt( 256 ),rand.nextInt( 256 ),rand.nextInt( 256 ));
        g2d.translate(p.x, p.y);
        g2d.rotate(-Math.PI/2);
        g2d.setPaint(new GradientPaint(0,raio/2,color1, raio, raio/2, color2));
        g2d.fill(poligono);
        g2d.rotate(Math.PI/2);
        g2d.translate(-p.x, -p.y);
    }
}

```



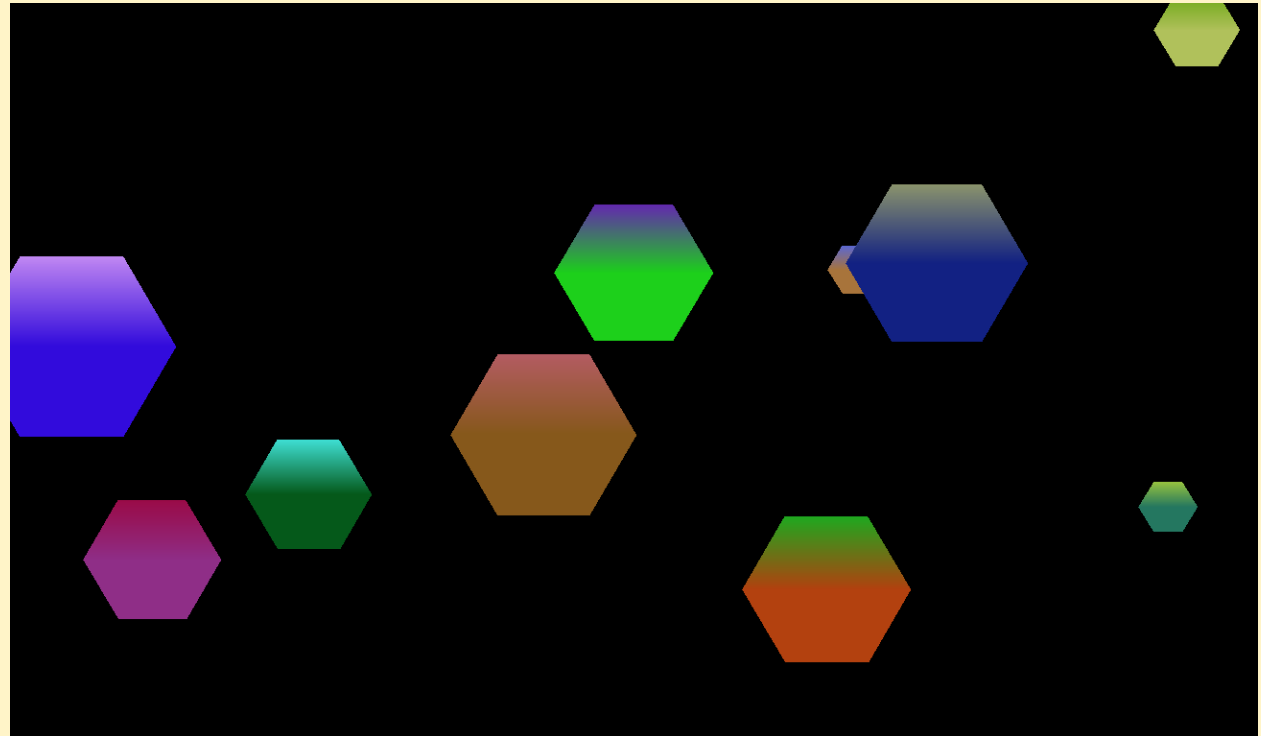
```

public GeneralPath montaForma(int forma, int raio)
{
    int x, y;
    GeneralPath poligono = new GeneralPath();
    poligono.moveTo(0, -raio);
    for (int i = 1; i < forma; i++) {
        x = (int) (raio * Math.cos(-Math.PI / 2 + 2 * Math.PI / forma * i));
        y = (int) (raio * Math.sin(-Math.PI / 2 + 2 * Math.PI / forma * i));
        poligono.lineTo(x, y);
        // se quiser desenhar cubo isométrico
        if (i % 2 == 1) {
            poligono.lineTo(0, 0);
            poligono.lineTo(x, y);
        }
    }
    poligono.closePath();

    return poligono;
}

} // final da classe

```



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

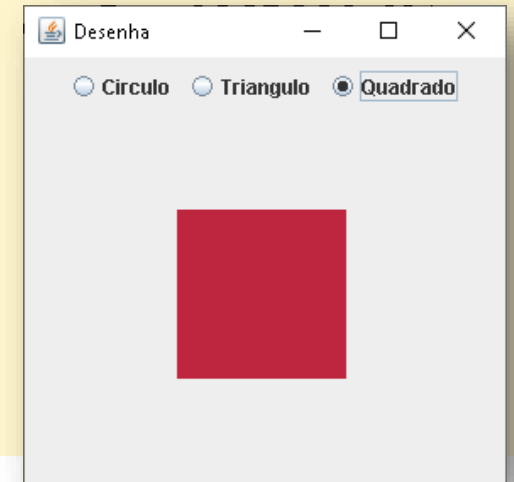
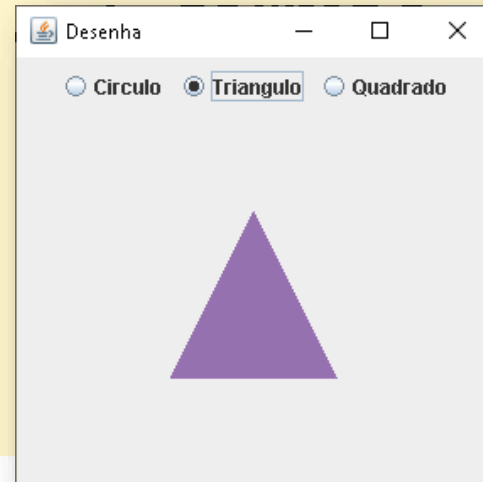
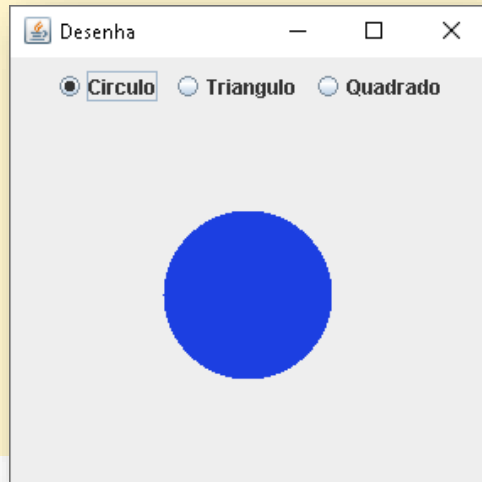
public class MainHexagono {

    public static void main(String[] args) {

        JFrame frame = new JFrame("Hexagonos");
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension D = tk.getScreenSize();
        frame.setSize(D);
        frame.add(new Hexagono());
        frame.setUndecorated(true);
        frame.setVisible(true);
        // registrar evento do teclado para sair do programa
        frame.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent e) {
                System.exit(0);
            }
        });
    }
} // final da classe
```

## 4ª Tarefa: (1D)

Criar uma subclasse de *JPanel* que permita que o usuário selecione um tipo de forma em um *JRadioButton* e ao clicar no espaço ao lado a classe desenhe esta forma. Permita a seleção de pelo menos 3 formas diferentes.



```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.util.*;

public class Formas extends JPanel {
    private int forma = 1;
    private JRadioButton bola;
    private JRadioButton tri;
    private JRadioButton quad;
    private ButtonGroup radioGroup;

    public Formas() {
        bola = new JRadioButton("Circulo", true);
        tri = new JRadioButton("Triangulo", false);
        quad = new JRadioButton("Quadrado", false);
        add(bola);
        add(tri);
        add(quad);

        radioGroup = new ButtonGroup();
        radioGroup.add(bola);
        radioGroup.add(tri);
        radioGroup.add(quad);
    }
}
```

```
// register events for JRadioButtons
bola.addItemListener( new TrataEvento(1) );
tri.addItemListener( new TrataEvento(2) );
quad.addItemListener( new TrataEvento(3) );
} // end Formas constructor

// private inner class to handle radio button events
private class TrataEvento implements ItemListener {
    // Numero da forma a ser desenhada
    private int var;

    public TrataEvento(int i) {
        var = i;
    }

    @Override
    public void itemStateChanged(ItemEvent ie) {
        forma = var;
        repaint();
    }
} // end private inner class TrataEvento
```

```
@Override
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    Random rand = new Random();

    g.setColor(new Color(rand.nextInt( 256 ),
                          rand.nextInt( 256 ),rand.nextInt( 256 )));

    if (forma == 1) {
        g.fillOval(90, 90, 100, 100);
    }

    if (forma == 2) {

        int[] x = {90, 190, 140};
        int[] y = {190, 190, 90};
        g.fillPolygon(x, y, 3);

    }
    if (forma == 3) {
        g.fillRect(90, 90, 100, 100);
    }
}
}
```



# Janela para o programa

```
import javax.swing.JFrame;

public class MainForma {

    public static void main(String[] args) {
        JFrame frame = new JFrame("Desenha");
        frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        frame.add(new Formas());
        frame.setSize( 300, 300 ); // set frame size
        frame.setVisible( true ); // display frame
    }
}
```



## 5ª Tarefa:

---

Escreva um método recursivo para calcular a potência de um determinado número. Denomine o método como potencia ( base , expoente ) e assumir que o expoente é um número maior ou igual a 1.

Lembrar que :

$$\text{base}^{\text{expoente}} = \text{base} . \text{base}^{\text{expoente}-1}$$

e que

$$\text{base}^1 = \text{base}$$

Utilize esta função em um aplicativo Java com tratamento de evento de ação.