```
import java.awt.Frame;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.nio.FloatBuffer;

import java.nio.IntBuffer;


import javax.media.opengl.*;

import javax.media.opengl.awt.GLCanvas;


import com.jogamp.common.nio.Buffers;

import com.jogamp.opengl.util.*;


public class SimpleScene implements GLEventListener {


   private double theta = 0;

   private double s = 0;

   private double c = 0;

   float co = 0.0f;


   public static void main(String[] args) {

      GLProfile glp = GLProfile.get(GLProfile.GL2 );

      GLCapabilities caps = new GLCapabilities(glp);


      GLCanvas canvas = new GLCanvas(caps);
```

```
    Frame frame = new Frame("Window with GLCanvas");
    frame.setSize(400, 400);
    frame.add(canvas);
    frame.setVisible(true);

    frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });

    canvas.addGLEventListener(new SimpleScene());

    Animator animator = new Animator(canvas);
    animator.start();
}

@Override
public void display(GLAutoDrawable drawable) {
    update();
    render(drawable);
}

@Override
public void dispose(GLAutoDrawable drawable) {
```

```
    }

    @Override
    public void init(GLAutoDrawable drawable) {
    }

    @Override
    public void reshape(GLAutoDrawable drawable, int x, int y, int w, int h) {
    }

    private void update() {
        theta += 0.01;
        s = Math.sin(theta);
        c = Math.cos(theta);
    }

    private void render(GLAutoDrawable drawable) {
        GL2 gl = drawable.getGL().getGL2();

        gl.glClear(GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT |
GL2.GL_STENCIL_BUFFER_BIT);

        gl.glMatrixMode(GL2.GL_PROJECTION);
        gl.glLoadIdentity();

        gl.glMatrixMode(GL2.GL_MODELVIEW);
```

```
    gl.glLoadIdentity();


  //      drawTriangleWithBeginEnd(gl);

  //   drawTriangleWithVertexArray(gl);

  //   drawTriangleWithIndices(gl);

  //   drawTriangleWithVertexBufferObject(gl);

  //  drawTriangleWithIndicesAndVertexBufferObject(gl);


}

//Using deprecated methods in OpenGL >3.0

//Don't use in new programs

void drawTriangleWithBeginEnd(GL2 gl)

{

          gl.glBegin(GL.GL_TRIANGLES);

    gl.glColor3f(1, 0, 0);

    gl.glVertex3f(-0.5f,-0.5f,0.0f);

    gl.glColor3f(0, 1, 0);

    gl.glVertex3f(0.5f,-0.5f,0.0f);

    gl.glColor3f(0, 0, 1);

    gl.glVertex3f(0.5f,0.5f,0.0f);

    gl.glEnd();

}


//Using Vertex arrays

void drawTriangleWithVertexArray(GL2 gl)
```

```
  {
    FloatBuffer cBuffer = Buffers.newDirectFloatBuffer(9);
    cBuffer.put(1).put(0).put(0);
    cBuffer.put(0).put(1).put(0);
    cBuffer.put(0).put(0).put(1);
    cBuffer.flip();

    FloatBuffer vBuffer = Buffers.newDirectFloatBuffer(9);
    vBuffer.put(-0.5f).put(-0.5f).put(0.0f);
    vBuffer.put(+0.5f).put(-0.5f).put(0.0f);
    vBuffer.put(+0.5f).put(+0.5f).put(0.0f);
    vBuffer.flip();

    gl.glEnableClientState(GL2.GL_VERTEX_ARRAY);
    gl.glEnableClientState(GL2.GL_COLOR_ARRAY);

    gl.glColorPointer(3, GL2.GL_FLOAT, 12, cBuffer);
    gl.glVertexPointer(3, GL2.GL_FLOAT, 12, vBuffer);
    gl.glDrawArrays(GL2.GL_TRIANGLES, 0, 3);

    gl.glDisableClientState(GL2.GL_COLOR_ARRAY);
    gl.glDisableClientState(GL2.GL_VERTEX_ARRAY);
  }

//Using DrawElements
```

```
void drawTriangleWithIndices(GL2 gl)
{
  FloatBuffer cBuffer = Buffers.newDirectFloatBuffer(9);
  cBuffer.put(1).put(0).put(0);
  cBuffer.put(0).put(1).put(0);
  cBuffer.put(0).put(0).put(1);
  cBuffer.flip();

  FloatBuffer vBuffer = Buffers.newDirectFloatBuffer(9);
  vBuffer.put(-0.5f).put(-0.5f).put(0.0f);
  vBuffer.put(+0.5f).put(-0.5f).put(0.0f);
  vBuffer.put(+0.5f).put(+0.5f).put(0.0f);
  vBuffer.flip();

  IntBuffer iBuffer = Buffers.newDirectIntBuffer(3);
  iBuffer.put(0);
  iBuffer.put(1);
  iBuffer.put(2);
  iBuffer.flip();

  gl.glEnableClientState(GL2.GL_VERTEX_ARRAY);
  gl.glEnableClientState(GL2.GL_COLOR_ARRAY);

  gl.glColorPointer(3, GL2.GL_FLOAT, 12, cBuffer);
  gl.glVertexPointer(3, GL2.GL_FLOAT, 12, vBuffer);
```

```
  gl.glDrawElements(GL2.GL_TRIANGLES, 3 , GL.GL_UNSIGNED_INT, iBuffer);


  gl.glDisableClientState(GL2.GL_COLOR_ARRAY);
  gl.glDisableClientState(GL2.GL_VERTEX_ARRAY);
}


//Using VBO (Vertex Buffer Object)
static void drawTriangleWithVertexBufferObject(GL2 gl)
{
        FloatBuffer cBuffer = Buffers.newDirectFloatBuffer(9);


   cBuffer.put(1).put(0).put(0);
   cBuffer.put(0).put(1).put(0);
   cBuffer.put(0).put(0).put(1);
   cBuffer.flip();


   FloatBuffer vBuffer = Buffers.newDirectFloatBuffer(9);
   vBuffer.put(-0.5f).put(-0.5f).put(0.0f);
   vBuffer.put(+0.5f).put(-0.5f).put(0.0f);
   vBuffer.put(+0.5f).put(+0.5f).put(0.0f);
   vBuffer.flip();


  IntBuffer ib = Buffers.newDirectIntBuffer(2);


  gl.glGenBuffers(2, ib);
```

```
int vHandle = ib.get(0);
int cHandle = ib.get(1);

gl.glEnableClientState(GL2.GL_VERTEX_ARRAY);
gl.glEnableClientState(GL2.GL_COLOR_ARRAY);

gl.glBindBuffer(GL2.GL_ARRAY_BUFFER, vHandle);
gl.glBufferData(GL2.GL_ARRAY_BUFFER, 12 * 3 ,vBuffer, GL2.GL_STATIC_DRAW);
gl.glVertexPointer(3, GL2.GL_FLOAT, 12, 0L);

gl.glBindBuffer(GL2.GL_ARRAY_BUFFER, cHandle);
gl.glBufferData(GL2.GL_ARRAY_BUFFER, 12 *3 ,cBuffer, GL2.GL_STATIC_DRAW);
gl.glColorPointer(3, GL2.GL_FLOAT, 12, 0L);

gl.glDrawArrays(GL2.GL_TRIANGLES, 0, 3);

gl.glBindBuffer(GL2.GL_ARRAY_BUFFER, 0);

gl.glDisableClientState(GL2.GL_COLOR_ARRAY);
gl.glDisableClientState(GL2.GL_VERTEX_ARRAY);

// cleanup VBO handles
ib.put(0, vHandle);
ib.put(1, cHandle);
gl.glDeleteBuffers(2, ib);
```

```
}

//Using VBO + Indices
static void drawTriangleWithIndicesAndVertexBufferObject(GL2 gl)
{
        FloatBuffer cBuffer = Buffers.newDirectFloatBuffer(9);

   cBuffer.put(1).put(0).put(0);
   cBuffer.put(0).put(1).put(0);
   cBuffer.put(0).put(0).put(1);
   cBuffer.flip();

   FloatBuffer vBuffer = Buffers.newDirectFloatBuffer(9);
   vBuffer.put(-0.5f).put(-0.5f).put(0.0f);
   vBuffer.put(+0.5f).put(-0.5f).put(0.0f);
   vBuffer.put(+0.5f).put(+0.5f).put(0.0f);
   vBuffer.flip();

   IntBuffer indices = Buffers.newDirectIntBuffer(3);
   indices.put(0);
   indices.put(1);
   indices.put(2);
   indices.flip();

  IntBuffer ib = Buffers.newDirectIntBuffer(3);
```

```
//Generate buffers for vertex, color and index data
gl.glGenBuffers(3, ib);
int vHandle = ib.get(0);
int cHandle = ib.get(1);
int iHandle = ib.get(2);

//Load data into buffers
gl.glBindBuffer(GL2.GL_ARRAY_BUFFER, vHandle);
gl.glBufferData(GL2.GL_ARRAY_BUFFER, 12 * 3 ,vBuffer, GL2.GL_STATIC_DRAW);
gl.glVertexPointer(3, GL2.GL_FLOAT, 12, 0L);

gl.glBindBuffer(GL2.GL_ARRAY_BUFFER, cHandle);
gl.glBufferData(GL2.GL_ARRAY_BUFFER, 12 *3 ,cBuffer, GL2.GL_STATIC_DRAW);
//Old code, use glVertexAttribPointer instead
gl.glColorPointer(3, GL2.GL_FLOAT, 12, 0L);

gl.glBindBuffer(GL2.GL_ELEMENT_ARRAY_BUFFER, iHandle);
gl.glBufferData(GL2.GL_ELEMENT_ARRAY_BUFFER, indices.limit() * 4, indices,
GL2.GL_STATIC_DRAW);

//Old Code... replaced by glEnableVertexAttribArray
gl.glEnableClientState(GL2.GL_VERTEX_ARRAY);
gl.glEnableClientState(GL2.GL_COLOR_ARRAY);

gl.glDrawElements(GL2.GL_TRIANGLES, 3 , GL.GL_UNSIGNED_INT, 0);
```

```
        gl.glBindBuffer(GL2.GL_ARRAY_BUFFER, 0);


        gl.glDisableClientState(GL2.GL_COLOR_ARRAY);
        gl.glDisableClientState(GL2.GL_VERTEX_ARRAY);


        // cleanup VBO handles
        ib.put(0, vHandle);
        ib.put(1, cHandle);
        ib.put(2, iHandle);
        gl.glDeleteBuffers(3, ib);
    }




}
```