

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** **4**

з дисципліни “Математичні та алгоритмічні основи комп’ютерної графіки”

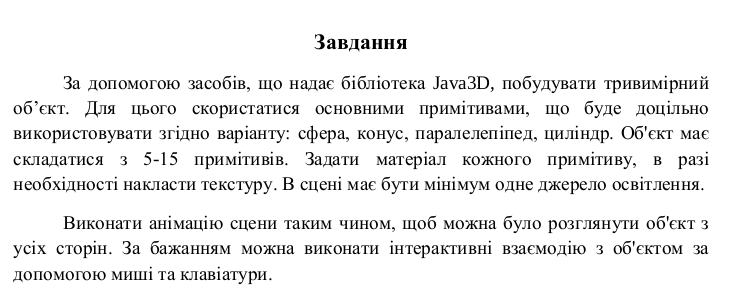
|  |  |  |  |
| --- | --- | --- | --- |
| Виконав  студент III курсу  групи КП-81  Черняєв Іван  (*прізвище, ім’я, по батькові*)  варіант № 19 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною (*прізвище, ім’я, по батькові*) |

Київ 2021

**Варіант завдання**

**Завдання**: Створити малюнок за варіантом користуючись графічними примітивами бібліотеки JavaFX.

**Варіант 19**



**Лістинг коду програми**

package com.company;import com.sun.j3d.utils.behaviors.mouse.MouseRotate;import com.sun.j3d.utils.behaviors.mouse.MouseTranslate;import com.sun.j3d.utils.behaviors.mouse.MouseZoom;import com.sun.j3d.utils.geometry.Box;import com.sun.j3d.utils.geometry.Cylinder;import com.sun.j3d.utils.geometry.Primitive;import com.sun.j3d.utils.image.TextureLoader;import com.sun.j3d.utils.universe.SimpleUniverse;import javax.media.j3d.\*;import javax.swing.\*;import javax.vecmath.\*;import java.awt.\*;import java.awt.event.ActionEvent;import java.awt.event.ActionListener;import java.awt.event.WindowAdapter;import java.awt.event.WindowEvent;import java.io.File;public class Main implements ActionListener { private BoundingSphere bounds = new BoundingSphere(new Point3d(0, 0, 0), 100.0); private TransformGroup pcTransformGroup = new TransformGroup(); private Transform3D pcTransform3D = new Transform3D(); private PCBuilder builder = new PCBuilder(); private Timer timer; private float angle = 0; public static void main(String[] args) { new Main(); } private Main() { timer = new Timer(25, this); timer.start(); BranchGroup root = createScene(); Frame frame = new Frame(); frame.setSize(1400, 900); // set window properties frame.addWindowListener(new WindowAdapter() { public void windowClosing(WindowEvent e) { System.*exit*(0); } }); // init universe Canvas3D canvas = new Canvas3D(SimpleUniverse.*getPreferredConfiguration*()); SimpleUniverse universe = new SimpleUniverse(canvas); universe.addBranchGraph(root); universe.getViewingPlatform().setNominalViewingTransform(); frame.add(BorderLayout.*CENTER*, canvas); frame.setVisible(true); } private BranchGroup createScene() { BranchGroup objRoot = new BranchGroup(); pcTransform3D.setTranslation(new Vector3d(0, 0, -10)); buildPC(); // -------------- mouse movements ---------------------- // allow runtime modification pcTransformGroup.setCapability(TransformGroup.*ALLOW\_TRANSFORM\_READ*); pcTransformGroup.setCapability(TransformGroup.*ALLOW\_TRANSFORM\_WRITE*); // rotate with mouse MouseRotate rotate = new MouseRotate(pcTransformGroup); rotate.setFactor(0.02); rotate.setSchedulingBounds(bounds); pcTransformGroup.addChild(rotate); // zoom with mouse MouseZoom zoom = new MouseZoom(pcTransformGroup); zoom.setFactor(0.02); zoom.setSchedulingBounds(bounds); pcTransformGroup.addChild(zoom); // move with mouse MouseTranslate translate = new MouseTranslate(pcTransformGroup); translate.setFactor(0.02); translate.setSchedulingBounds(bounds); pcTransformGroup.addChild(translate); objRoot.addChild(pcTransformGroup); Background background = new Background(new Color3f(1.0f, 1.0f, 1.0f)); BoundingSphere sphere = new BoundingSphere(new Point3d(0, 0, 0), 100000); background.setApplicationBounds(sphere); objRoot.addChild(background); BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0); Color3f light1Color = new Color3f(1.0f, 0.5f, 0.4f); Vector3f light1Direction = new Vector3f(.8f, .8f, .0f); DirectionalLight light1 = new DirectionalLight(light1Color, light1Direction); light1.setInfluencingBounds(bounds); objRoot.addChild(light1); Color3f ambientColor = new Color3f(1.0f, 1.0f, 1.0f); AmbientLight ambientLightNode = new AmbientLight(ambientColor); ambientLightNode.setInfluencingBounds(bounds); objRoot.addChild(ambientLightNode); return objRoot; } private void buildPC() { Box leg1 = builder.getPart(.02f, .3f, .3f); Transform3D body1T = new Transform3D(); body1T.setTranslation(new Vector3f(.4f, .0f, .0f)); TransformGroup body1TG = new TransformGroup(); body1TG.setTransform(body1T); body1TG.addChild(leg1); pcTransformGroup.addChild(body1TG); Box leg2 = builder.getPart(.02f, .3f, .3f); Transform3D body2T = new Transform3D(); body2T.setTranslation(new Vector3f(-.4f, .0f, .0f)); TransformGroup body2TG = new TransformGroup(); body2TG.setTransform(body2T); body2TG.addChild(leg2); pcTransformGroup.addChild(body2TG); Box table = builder.getPart(.6f, .01f, .34f); Transform3D body3T = new Transform3D(); body3T.setTranslation(new Vector3f(.0f, .3f, .0f)); TransformGroup body3TG = new TransformGroup(); body3TG.setTransform(body3T); body3TG.addChild(table); pcTransformGroup.addChild(body3TG); Box PC\_Box = builder.getPCBox(.08f, .2f, .28f); Transform3D body4T = new Transform3D(); body4T.setTranslation(new Vector3f(.2f, -.1f, .025f)); TransformGroup body4TG = new TransformGroup(); body4TG.setTransform(body4T); body4TG.addChild(PC\_Box); pcTransformGroup.addChild(body4TG); Box PC\_Box\_back = builder.getPCBack(.078f, .168f, .005f); Transform3D body5T = new Transform3D(); body5T.setTranslation(new Vector3f(.2f, -.07f, -.26f)); TransformGroup body5TG = new TransformGroup(); body5TG.setTransform(body5T); body5TG.addChild(PC\_Box\_back); pcTransformGroup.addChild(body5TG); Cylinder PC\_Monitor\_bottom = builder.getMonitor(.1f, .03f); Transform3D body6T = new Transform3D(); body6T.setTranslation(new Vector3f(.0f, .32f, .0f)); TransformGroup body6TG = new TransformGroup(); body6TG.setTransform(body6T); body6TG.addChild(PC\_Monitor\_bottom); pcTransformGroup.addChild(body6TG); Cylinder PC\_Monitor\_leg = builder.getMonitor(.013f, .08f); Transform3D body7T = new Transform3D(); body7T.setTranslation(new Vector3f(.0f, .37f, .0f)); TransformGroup body7TG = new TransformGroup(); body7TG.setTransform(body7T); body7TG.addChild(PC\_Monitor\_leg); pcTransformGroup.addChild(body7TG); Box PC\_Monitor = builder.getMonitor(.3f, .18f, 0.015f); Transform3D body8T = new Transform3D(); body8T.setTranslation(new Vector3f(.0f, .59f, .0f)); TransformGroup body8TG = new TransformGroup(); body8TG.setTransform(body8T); body8TG.addChild(PC\_Monitor); pcTransformGroup.addChild(body8TG); Box PC\_Screen = builder.getScreen(.29f, .165f, 0.001f); Transform3D body9T = new Transform3D(); body9T.setTranslation(new Vector3f(.0f, .59f, .015f)); TransformGroup body9TG = new TransformGroup(); body9TG.setTransform(body9T); body9TG.addChild(PC\_Screen); pcTransformGroup.addChild(body9TG); Cylinder PC\_Push = builder.getPush(.05f, 0.001f); Transform3D body10T = new Transform3D(); body10T.rotX(Math.*PI* / 2); body10T.setTranslation(new Vector3f(.2f, -.19f, .305f)); TransformGroup body10TG = new TransformGroup(); body10TG.setTransform(body10T); body10TG.addChild(PC\_Push); pcTransformGroup.addChild(body10TG); Box PC\_CD1 = builder.getCD(.06f, .008f, 0.001f); Transform3D body11T = new Transform3D(); body11T.setTranslation(new Vector3f(.2f, -0.f, .305f)); TransformGroup body11TG = new TransformGroup(); body11TG.setTransform(body11T); body11TG.addChild(PC\_CD1); pcTransformGroup.addChild(body11TG); Box PC\_CD2 = builder.getCD(.06f, .008f, 0.001f); Transform3D body12T = new Transform3D(); body12T.setTranslation(new Vector3f(.2f, 0.02f, .305f)); TransformGroup body12TG = new TransformGroup(); body12TG.setTransform(body12T); body12TG.addChild(PC\_CD2); pcTransformGroup.addChild(body12TG); Box PC\_CD3 = builder.getCD(.06f, .008f, 0.001f); Transform3D body13T = new Transform3D(); body13T.setTranslation(new Vector3f(.2f, 0.04f, .305f)); TransformGroup body13TG = new TransformGroup(); body13TG.setTransform(body13T); body13TG.addChild(PC\_CD3); pcTransformGroup.addChild(body13TG); Box PC\_USB1 = builder.getButton(.013f, .005f, 0.001f); Transform3D body14T = new Transform3D(); body14T.setTranslation(new Vector3f(.22f, -.02f, .305f)); TransformGroup body14TG = new TransformGroup(); body14TG.setTransform(body14T); body14TG.addChild(PC\_USB1); pcTransformGroup.addChild(body14TG); Box PC\_USB2 = builder.getButton(.013f, .005f, 0.001f); Transform3D body15T = new Transform3D(); body15T.setTranslation(new Vector3f(.18f, -.02f, .305f)); TransformGroup body15TG = new TransformGroup(); body15TG.setTransform(body15T); body15TG.addChild(PC\_USB2); pcTransformGroup.addChild(body15TG); } @Override public void actionPerformed(ActionEvent e) {// pcTransform3D.rotY(angle);// pcTransform3D.setScale(0.4);// pcTransformGroup.setTransform(pcTransform3D);// angle += 0.01; } public class PCBuilder { public PCBuilder() { } public Box getPart(float x, float y, float z) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Box(x, y, z, primitive\_flags, getTableAppearance()); } public Box getPCBox(float x, float y, float z) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Box(x, y, z, primitive\_flags, get\_PC\_box\_Appearance()); } public Box getPCBack(float x, float y, float z) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Box(x, y, z, primitive\_flags, get\_PC\_box\_back\_Appearance()); } public Box getMonitor(float x, float y, float z) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Box(x, y, z, primitive\_flags, get\_Monitor\_Appearance()); } public Cylinder getMonitor(float radius, float height) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Cylinder(radius, height, primitive\_flags, get\_Monitor\_Appearance()); } public Cylinder getPush(float radius, float height) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Cylinder(radius, height, primitive\_flags, get\_Push\_Appearance()); } public Box getScreen(float x, float y, float z) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Box(x, y, z, primitive\_flags, get\_Screen\_Appearance()); } public Box getButton(float x, float y, float z) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Box(x, y, z, primitive\_flags, get\_Button\_Appearance()); } public Box getCD(float x, float y, float z) { int primitive\_flags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*; return new Box(x, y, z, primitive\_flags, get\_CD\_Appearance()); } private Appearance getTableAppearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/wood.jpg", "LUMINANCE", new Container()); Appearance ap = new Appearance(); Texture texture = loader.getTexture(); texture.setBoundaryModeS(Texture.*WRAP*); texture.setBoundaryModeT(Texture.*WRAP*); texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f)); TextureAttributes texAttr = new TextureAttributes(); texAttr.setTextureMode(TextureAttributes.*MODULATE*); ap.setTexture(texture); ap.setTextureAttributes(texAttr); Color3f emissive = new Color3f(new Color(69, 29, 0)); Color3f ambient = new Color3f(new Color(119, 41, 0)); Color3f diffuse = new Color3f(); Color3f specular = new Color3f(new Color(0, 0, 0)); ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f)); return ap; } private Appearance get\_PC\_box\_Appearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/plastic\_for\_pc.jpg", "LUMINANCE", new Container()); Appearance ap = new Appearance(); Texture texture = loader.getTexture(); texture.setBoundaryModeS(Texture.*WRAP*); texture.setBoundaryModeT(Texture.*WRAP*); texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f)); TextureAttributes texAttr = new TextureAttributes(); texAttr.setTextureMode(TextureAttributes.*MODULATE*); ap.setTexture(texture); ap.setTextureAttributes(texAttr); Color3f emissive = new Color3f(new Color(99, 99, 101)); Color3f ambient = new Color3f(new Color(0, 0, 0, 210)); Color3f diffuse = new Color3f(); Color3f specular = new Color3f(new Color(0, 0, 0)); ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f)); return ap; } private Appearance get\_PC\_box\_back\_Appearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/metal.jpg", "LUMINANCE", new Container()); Appearance ap = new Appearance(); Texture texture = loader.getTexture(); texture.setBoundaryModeS(Texture.*WRAP*); texture.setBoundaryModeT(Texture.*WRAP*); texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f)); TextureAttributes texAttr = new TextureAttributes(); texAttr.setTextureMode(TextureAttributes.*MODULATE*); ap.setTexture(texture); ap.setTextureAttributes(texAttr); Color3f emissive = new Color3f(new Color(71, 71, 73)); Color3f ambient = new Color3f(new Color(179, 179, 179, 210)); Color3f diffuse = new Color3f(); Color3f specular = new Color3f(new Color(0, 0, 0)); ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f)); return ap; } private Appearance get\_Monitor\_Appearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/plastic.jpg", "LUMINANCE", new Container()); Appearance ap = new Appearance(); Texture texture = loader.getTexture(); texture.setBoundaryModeS(Texture.*WRAP*); texture.setBoundaryModeT(Texture.*WRAP*); texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f)); TextureAttributes texAttr = new TextureAttributes(); texAttr.setTextureMode(TextureAttributes.*MODULATE*); ap.setTexture(texture); ap.setTextureAttributes(texAttr); Color3f emissive = new Color3f(new Color(71, 71, 73)); Color3f ambient = new Color3f(new Color(6, 6, 6, 210)); Color3f diffuse = new Color3f(); Color3f specular = new Color3f(new Color(0, 0, 0)); ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f)); return ap; } private Appearance get\_Push\_Appearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/push.jpg", "LUMINANCE", new Container()); return getAppearance(loader); } private Appearance getAppearance(TextureLoader loader) { Appearance ap = new Appearance(); Texture texture = loader.getTexture(); texture.setBoundaryModeS(Texture.*WRAP*); texture.setBoundaryModeT(Texture.*WRAP*); texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f)); TextureAttributes texAttr = new TextureAttributes(); texAttr.setTextureMode(TextureAttributes.*MODULATE*); ap.setTexture(texture); ap.setTextureAttributes(texAttr); Color3f emissive = new Color3f(new Color(3, 3, 4)); Color3f ambient = new Color3f(new Color(173, 135, 255, 255)); Color3f diffuse = new Color3f(); Color3f specular = new Color3f(new Color(0, 0, 0)); ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f)); return ap; } private Appearance get\_CD\_Appearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/cd.jpg", "LUMINANCE", new Container()); return getAppearance(loader); } private Appearance get\_Button\_Appearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/buttons.jpg", "LUMINANCE", new Container()); Appearance ap = new Appearance(); Texture texture = loader.getTexture(); texture.setBoundaryModeS(Texture.*WRAP*); texture.setBoundaryModeT(Texture.*WRAP*); texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f)); TextureAttributes texAttr = new TextureAttributes(); texAttr.setTextureMode(TextureAttributes.*MODULATE*); ap.setTexture(texture); ap.setTextureAttributes(texAttr); Color3f emissive = new Color3f(new Color(87, 87, 87)); Color3f ambient = new Color3f(new Color(101, 74, 140, 246)); Color3f diffuse = new Color3f(); Color3f specular = new Color3f(new Color(0, 0, 0)); ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f)); return ap; } private Appearance get\_Screen\_Appearance() { TextureLoader loader = new TextureLoader("/home/ivan/workspace/Lab4/src/textures/screen.png", "LUMINANCE", new Container()); Appearance ap = new Appearance(); Texture texture = loader.getTexture(); texture.setBoundaryModeS(Texture.*WRAP*); texture.setBoundaryModeT(Texture.*WRAP*); texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f)); TextureAttributes texAttr = new TextureAttributes(); texAttr.setTextureMode(TextureAttributes.*MODULATE*); ap.setTexture(texture); ap.setTextureAttributes(texAttr); Color3f emissive = new Color3f(new Color(0, 3, 119)); Color3f ambient = new Color3f(new Color(0, 3, 50, 238)); Color3f diffuse = new Color3f(); Color3f specular = new Color3f(new Color(0, 0, 0)); ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f)); return ap; } }}

**Результат**

