Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Институт информационных технологий

Современные платформы программирования

Лабораторная № 1

«Графический интерфейс java-приложений. Swing и AWT»

Выполнил: Фут Д.С.

Проверил: Калитеня И.Л.

Минск 2017

Лабораторная № 1

«Графический интерфейс java-приложений. Swing и AWT»

1. Задать движение по экрану строк (одна за другой) из массива строк. Направление движения по апплету и значение каждой строки выбирается случайным образом.

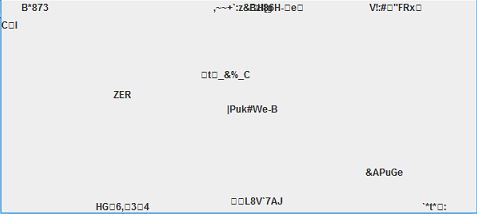


Рис. 1 – Скриншот программы

Листинг:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Random;

class JLabelStorage {

JLabel label;

int direction;

JLabelStorage() {

this.label = new JLabel();

this.direction = 0;

}

JLabelStorage(String title) {

this.label = new JLabel(title);

this.direction = 0;

}

JLabelStorage(String title, int direction) {

this.label = new JLabel(title);

this.direction = direction;

}

JLabel getLabel() {

return this.label;

}

int getDirection(){

return this.direction;

}

void setDirection(int direction){

this.direction = direction;

}

}

public class lab1\_1 {

public static void main(String args[]){

JFrame frame = new JFrame("Фут Лаба 1.1");

Random randomer = new Random();

//frame init

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

frame.setVisible(true);

frame.setSize(600,300);

frame.setResizable(false);

frame.setLayout(new FlowLayout());

frame.setLocationRelativeTo(null);

//Strings & labels init

int sArrayLength = randomer.nextInt(15) + 5;

JLabelStorage[] labels = new JLabelStorage[sArrayLength];

for (int i = 0; i < sArrayLength; i++) {

int curStringLength = randomer.nextInt(10) + 3;

String curString = "";

for (int j = 0; j < curStringLength; j++) {

int charCode = randomer.nextInt(122) + 33;

curString += Character.toString((char) charCode);

}

int curDirection = randomer.nextInt(7);

labels[i] = new JLabelStorage(curString,curDirection);

frame.add(labels[i].getLabel());

}

Timer tm = new Timer(10, new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

for (int i = 0; i < labels.length; i ++) {

JLabel curLabel = labels[i].getLabel();

int curX = curLabel.getX(),

curY = curLabel.getY(),

dir = labels[i].getDirection();

boolean moved = false,

dirChanged = false;

while (!moved) {

switch (dir) {

case 0:

if (curY <= 0) {

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curY -= 1;

moved = true;

}

break;

case 1:

if (curY <= 0 || (curX + curLabel.getWidth()) >= frame.getContentPane().getWidth()){

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curY -= 1;

curX += 1;

moved = true;

}

break;

case 2:

if ((curX + curLabel.getWidth()) >= frame.getContentPane().getWidth()){

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curX += 1;

moved = true;

}

break;

case 3:

if ((curY + curLabel.getHeight()) >= frame.getContentPane().getHeight() || (curX + curLabel.getWidth()) >= frame.getContentPane().getWidth()){

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curY += 1;

curX += 1;

moved = true;

}

break;

case 4:

if ((curY + curLabel.getHeight()) >= frame.getContentPane().getHeight()){

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curY += 1;

moved = true;

}

break;

case 5:

if ((curY + curLabel.getHeight()) >= frame.getContentPane().getHeight() || curX <= 0){

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curY += 1;

curX -= 1;

moved = true;

}

break;

case 6:

if (curX <= 0){

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curX -= 1;

moved = true;

}

break;

case 7:

if (curY <= 0 || curX <= 0){

dir = randomer.nextInt(7);

dirChanged = true;

}

else {

curY -= 1;

curX -= 1;

moved = true;

}

break;

}

}

if (dirChanged)

labels[i].setDirection(dir);

curLabel.setLocation(curX,curY);

}

}});

tm.start();

}

}

1. Задать движение окружности по апплету так, чтобы при касании границы окружность отражалась от нее с эффектом упругого сжатия.

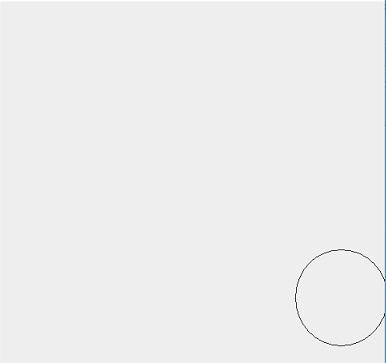


Рис. 2 – Скриншот программы

Листинг:

import java.awt.\*;

import java.awt.event.\*;

import java.awt.geom.\*;

import javax.swing.\*;

public class lab1\_2 {

public static void main(String[] args) {

JFrame mainFrame = new JFrame("Лаба 1");

mainFrame.setPreferredSize(new Dimension(500, 500));

final JPanel mainPanel = new JPanel();

mainFrame.add(mainPanel);

mainFrame.setVisible(true);

mainFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

mainFrame.pack();

Timer mainTimer = new Timer(17, new ActionListener() {

int ellipseRadius = 60,

ellipsePosX = 0, //windowWidth/2 - ellipseRadius,

ellipsePosY = 0,

posXDirection = 5,

posYDirection = 5,

ellipseWidth = ellipseRadius\*2,

ellipseHeight = ellipseRadius\*2,

compressDirectionX = 5,

compressDirectionY = 5;

boolean wasCompressedX = false,

wasCompressedY = false,

stopMoveX = false,

stopMoveY = false;

@Override

public void actionPerformed(ActionEvent e) {

Graphics2D graphicsObject = (Graphics2D) mainPanel.getRootPane().getGraphics();

mainPanel.update(graphicsObject);

GeneralPath ellipsePath = new GeneralPath();

if (((ellipsePosX + ellipseWidth) >= mainPanel.getWidth() && posXDirection > 0) || (ellipsePosX <= 0 && posXDirection < 0)) {

if (wasCompressedX) {

stopMoveY = false;

wasCompressedX = false;

posXDirection \*= -1;

compressDirectionX \*= -1;

} else {

stopMoveY = true;

if ((ellipsePosX + ellipseWidth) >= mainPanel.getWidth()) {

ellipseWidth -= compressDirectionX;

if (ellipseWidth > ellipseRadius\*2/5\*4) {

ellipsePosX += compressDirectionX;

} else {

compressDirectionX \*= -1;

}

} else if(ellipsePosX <= 0) {

ellipseWidth -= compressDirectionX;

if (ellipseWidth <= ellipseRadius\*2/5\*4) {

compressDirectionX \*= -1;

}

}

if (ellipseWidth == ellipseRadius \* 2) {

wasCompressedX = true;

}

}

} else {

if (!stopMoveX) {

ellipsePosX += posXDirection;

}

}

if (((ellipsePosY + ellipseHeight) >= mainPanel.getHeight() && posYDirection > 0) || (ellipsePosY <= 0 && posYDirection < 0)) {

if (wasCompressedY) {

stopMoveX = false;

wasCompressedY = false;

posYDirection \*= -1;

compressDirectionY \*= -1;

} else {

stopMoveX = true;

if ((ellipsePosY + ellipseHeight) >= mainPanel.getHeight()) {

ellipseHeight -= compressDirectionY;

if (ellipseHeight > ellipseRadius\*2/5\*4) {

ellipsePosY += compressDirectionY;

} else {

compressDirectionY \*= -1;

}

} else if(ellipsePosY <= 0) {

ellipseHeight -= compressDirectionY;

if (ellipseHeight <= ellipseRadius\*2/5\*4) {

compressDirectionY \*= -1;

}

}

if (ellipseHeight == ellipseRadius \* 2) {

wasCompressedY = true;

}

}

} else {

if (!stopMoveY) {

ellipsePosY += posYDirection;

}

}

ellipsePath.append((new Ellipse2D.Float(0, 0, ellipseWidth, ellipseHeight)), true);

graphicsObject.translate(ellipsePosX, ellipsePosY);

graphicsObject.draw(ellipsePath);

}});

mainTimer.start();

}

}

1. Изобразить в апплете приближающийся издали шар, удаляющийся шар. Шар должен двигаться с постоянной скоростью.

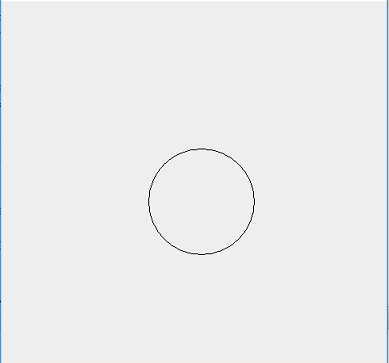


Рис. 3 – Скриншот программы

Листинг:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.geom.\*;

public class lab1\_3 {

public static void main(String[] args) {

JFrame frame = new JFrame("Фут Лаба 1.3");

frame.setSize(500,500);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

Timer tm = new Timer(50, new ActionListener(){

double scaleVal = 0.01,

step = 0.01;

@Override

public void actionPerformed(ActionEvent arg0) {

Graphics2D ball = (Graphics2D)frame.getRootPane().getGraphics();

frame.update(ball);

if (scaleVal == 0.01)

step = 0.01;

else if (scaleVal == 1)

step = -0.01;

scaleVal += step;

scaleVal = (double)Math.round(scaleVal \* 100)/100;

GeneralPath ballPath = new GeneralPath();

ballPath.append(new Ellipse2D.Float(

frame.getWidth()/2 - (200\*(float)scaleVal)/2,

frame.getHeight()/2 - (200\*(float)scaleVal)/2,

200\*(float)scaleVal,

200\*(float)scaleVal), true);

ball.draw(ballPath);

}});

tm.start();

}

}

1. Изобразить в окне приложения отрезок, вращающийся в плоскости экрана вокруг одной из своих концевых точек. Цвет прямой должен изменяться при переходе от одного положения к другому.

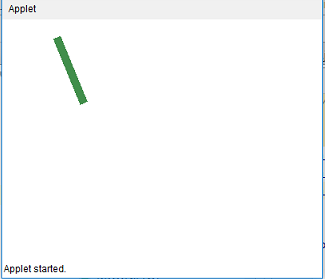


Рис. 4 – Скриншот программы

Листинг:

import java.applet.\*;

import java.awt.\*;

import java.util.Random;

public class lab1\_4 extends Applet implements Runnable{

Thread runner;

int posXcenter=100;

int posYcenter=100;

int posX=0;

int posY=0;

float radius=80;

float time=0;

Random rand=new Random();

public void run() {

while (runner != null) {

repaint();

try {

Thread.sleep( 20 );

} catch ( InterruptedException e ) {

}

}

}

public void start() {

// user visits the page, create a new thread

if ( runner == null ) {

runner = new Thread( this );

runner.start();

}

}

public void paint(Graphics g) {

Graphics2D g2D = (Graphics2D)g;

time+=0.1;

posX=(int)(radius \*Math.cos(time));

posY=(int)(radius \*Math.sin(time));

float \_r = rand.nextFloat();

float \_g = rand.nextFloat();

float \_b = rand.nextFloat();

BasicStroke pen1 = new BasicStroke(10);

g2D.setStroke(pen1);

g2D.setColor(new Color(\_r,\_g,\_b));

g2D.drawLine(posXcenter,posYcenter,posXcenter+posX,posYcenter+posY);

}

}

1. Изобразить в окне приложения отрезок, вращающийся в плоскости фрейма вокруг точки, движущейся по отрезку.

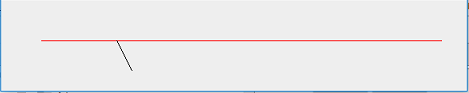


Рис. 5 – Скриншот программы

Листинг:

import java.awt.\*;

import java.awt.event.\*;

import java.awt.geom.\*;

import javax.swing.\*;

public class lab1\_5 {

public static void main(String[] args) {

JFrame fr=new JFrame("Изобразить в окне приложения отрезок, вращающийся в плоскости фрейма вокруг точки, движущейся по отрезку");

fr.setPreferredSize( new Dimension(600,120));

final JPanel pan= new JPanel();

fr.add(pan);

fr.setVisible(true);

fr.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

fr.pack();

Timer tm= new Timer(100, new ActionListener(){

int i=0;

int q=0;

int z = 1;

@Override

public void actionPerformed(ActionEvent arg0) {

int start=50; int end=550;//start and end of line

int ls=50; int le=80;

Graphics2D gr = (Graphics2D) pan.getRootPane().getGraphics();

pan.update(gr);

gr.setColor(Color.RED);

gr.draw(new Line2D.Double(start, start, end, start));

gr.setColor(Color.black);

GeneralPath path = new GeneralPath();

ls=ls+i;

le=le+i;

path.append(new Line2D.Double(ls, 50, le, 80), true);

int x = ls, y = 50;

AffineTransform tranforms = AffineTransform.getRotateInstance((i) \* 0.07, x, y);

if (ls==end){z = 0;} else if(ls==start){z = 1;}

if(z == 1){i++;}else{i--;}

gr.transform(tranforms);

gr.draw(path);

}

});

tm.start();

}

}

1. Изобразить четырехугольник, вращающийся в плоскости апплета вокруг своего центра тяжести.

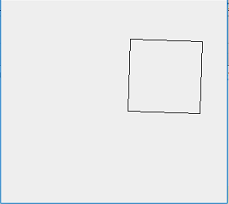


Рис. 6 – Скриншот программы

Листинг:

import java.awt.\*;

import java.awt.event.\*;

import java.awt.geom.\*;

import javax.swing.\*;

public class lab1\_6 {

public static void main(String[] args) {

JFrame fr=new JFrame("Фут Лаба 1.6");

fr.setPreferredSize( new Dimension(300,300));

final JPanel pan= new JPanel();

fr.add(pan);

fr.setVisible(true);

fr.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

fr.pack();

Timer tm= new Timer(50, new ActionListener(){

int i=0;

@Override

public void actionPerformed(ActionEvent arg0) {

Graphics2D gr=(Graphics2D)pan.getRootPane().getGraphics();

pan.update(gr);

GeneralPath path=new GeneralPath();

path.append(new Polygon(new int []{10,100,100,10},new int[]{-10,-10,-100,-100},4),true);

int x=(10+100+100+10)/4,y=(-10-10-100-100)/4;

gr.translate(150, 150);

AffineTransform tranforms = AffineTransform.getRotateInstance((i++)\*0.07, x, y);

gr.transform(tranforms);

gr.draw(path);

}});

tm.start();

}

}

1. Создать фрейм с областью для рисования «пером». Создать меню для выбора цвета и толщины линии.

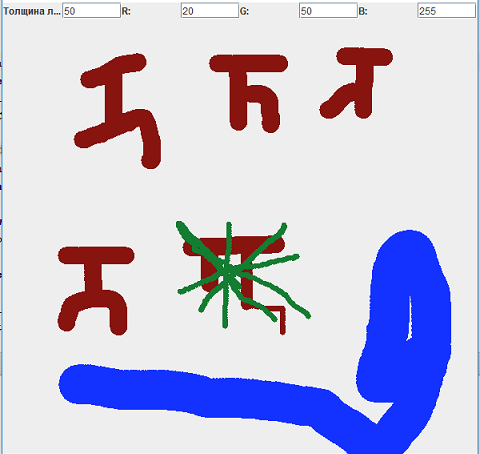


Рис. 7 – Скриншот программы

Листинг:

import org.w3c.dom.css.RGBColor;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.geom.Line2D;

import java.util.logging.ConsoleHandler;

public class lab1\_7 {

public static void main (String args[]){

JFrame frame = new JFrame("Фут лаба 1.7");

frame.setResizable(false);

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

frame.setSize(600,600);

frame.setLocationRelativeTo(null);

frame.setLayout(new BorderLayout());

JPanel menu = new JPanel(new GridLayout());

frame.add(menu,BorderLayout.NORTH);

JLabel text1 = new JLabel("Толщина линии:");

menu.add(text1);

JTextField lineWidthField = new JTextField("22");

menu.add(lineWidthField);

JLabel text2 = new JLabel("R:");

menu.add(text2);

JTextField redField = new JTextField("0");

menu.add(redField);

JLabel text3 = new JLabel("G:");

menu.add(text3);

JTextField greenField = new JTextField("0");

menu.add(greenField);

JLabel text4 = new JLabel("B:");

menu.add(text4);

JTextField blueField = new JTextField("0");

menu.add(blueField);

frame.setVisible(true);

JPanel canvas = new JPanel();

canvas.setPreferredSize(new Dimension(frame.getContentPane().getWidth(),frame.getContentPane().getHeight() - menu.getHeight()));

frame.add(canvas,BorderLayout.SOUTH);

canvas.setCursor(new Cursor(Cursor.CROSSHAIR\_CURSOR));

canvas.addMouseMotionListener(new MouseMotionListener() {

@Override

public void mouseDragged(MouseEvent e) {

try {

if (canvas.contains(e.getX(),e.getY() - 20)) {

Graphics2D frameGraph = (Graphics2D)canvas.getRootPane().getGraphics();

frameGraph.setColor(

new Color(

Integer.parseInt(redField.getText()),

Integer.parseInt(greenField.getText()),

Integer.parseInt(blueField.getText())

)

);

frameGraph.setStroke(

new BasicStroke(

Float.parseFloat(lineWidthField.getText()),

BasicStroke.CAP\_ROUND,

BasicStroke.JOIN\_ROUND

)

);

frameGraph.draw(

new Line2D.Float(

(float)e.getX(),

(float)e.getY() + 20,

(float)e.getX(),

(float)e.getY() + 20

)

);

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(

null,

"Введите корректные данные для кисти.",

"Ошибка",

JOptionPane.ERROR\_MESSAGE

);

}

}

@Override

public void mouseMoved(MouseEvent e) {

}

});

/\*

frame.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

drawEnable[0] = true;

}

@Override

public void mouseReleased(MouseEvent e) {

drawEnable[0] = false;

}

@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

});\*/

}

}

1. Составить программу для управления скоростью движения точки по апплету. Одна кнопка увеличивает скорость, другая – уменьшает. Каждый щелчок изменяет скорость на определенную величину.



Рис. 8 – Скриншот программы

Листинг:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.geom.Ellipse2D;

import java.awt.geom.GeneralPath;

class SpeedController{

int speed;

SpeedController(){

this.speed = 1;

}

int getSpeed(){

return this.speed;

}

void incSpeed() {

if (this.speed > 30) {

JOptionPane.showMessageDialog(

null,

"Скорость не возможно больше увеличить.",

"Ошибка",

JOptionPane.ERROR\_MESSAGE

);

} else

this.speed++;

}

void decSpeed() {

if (this.speed < 1) {

JOptionPane.showMessageDialog(

null,

"Скорость не возможно больше уменьшить.",

"Ошибка",

JOptionPane.ERROR\_MESSAGE

);

} else

this.speed--;

}

}

public class lab1\_8 {

public static void main(String args[]) {

JFrame frame = new JFrame("Фут Лаба 1.8");

frame.setSize(500,500);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setResizable(false);

frame.setLocationRelativeTo(null);

SpeedController speedControl = new SpeedController();

JPanel buttonHolder = new JPanel();

frame.add(buttonHolder,BorderLayout.NORTH);

JButton plusButton = new JButton("+");

buttonHolder.add(plusButton);

plusButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

speedControl.incSpeed();

}

});

JButton minusButton = new JButton("-");

buttonHolder.add(minusButton);

minusButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

speedControl.decSpeed();

}

});

JPanel actionPanel = new JPanel();

frame.add(actionPanel,BorderLayout.CENTER);

frame.setVisible(true);

Timer tm = new Timer(50, new ActionListener(){

int position = 10,

dir = 1;

@Override

public void actionPerformed(ActionEvent arg0) {

Graphics2D ball = (Graphics2D)actionPanel.getGraphics();

actionPanel.update(ball);

if (position + 10 >= actionPanel.getWidth())

dir = -1;

else if (position <= 0)

dir = 1;

position += speedControl.getSpeed()\*dir;

GeneralPath ballPath = new GeneralPath();

ballPath.append(new Ellipse2D.Float(

position,

0,

10,

10), false);

ball.draw(ballPath);

}});

tm.start();

}

}

1. Изобразить в окне гармонические колебания точки вдоль некоторого горизонтального отрезка. Если длина отрезка равна q, то расстояние от точки до левого конца в момент времени t можно считать равным q(1 + cos(wt))/2, где w – некоторая константа. Предусмотреть поля для ввода указанных величин и кнопку для остановки и пуска процесса.



Рис. 9 – Скриншот программы

Листинг:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.geom.Ellipse2D;

import java.awt.geom.GeneralPath;

public class lab1\_9 {

public static void main (String args[]){

JFrame frame = new JFrame("Фут Лаба 1.9");

frame.setSize(500,500);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(new BorderLayout());

JPanel menu = new JPanel(new FlowLayout());

frame.add(menu,BorderLayout.NORTH);

JLabel text1 = new JLabel("Q:");

menu.add(text1);

JTextField qField = new JTextField("5");

qField.setPreferredSize(new Dimension(50,20));

menu.add(qField);

JLabel text2 = new JLabel("W:");

menu.add(text2);

JTextField wField = new JTextField("5");

wField.setPreferredSize(new Dimension(50,20));

menu.add(wField);

JButton control = new JButton("Старт");

menu.add(control);

frame.setVisible(true);

JPanel canvas = new JPanel();

canvas.setPreferredSize(new Dimension(frame.getContentPane().getWidth(),frame.getContentPane().getHeight() - menu.getHeight()));

frame.add(canvas,BorderLayout.SOUTH);

control.addActionListener(new ActionListener() {

Timer tm;

@Override

public void actionPerformed(ActionEvent e) {

try {

if (control.getText() == "Старт") {

control.setText("Закончить");

tm = new Timer(50, new ActionListener(){

int q = Integer.parseInt(qField.getText()),

w = Integer.parseInt(wField.getText()),

t = 0;

@Override

public void actionPerformed(ActionEvent arg0) {

Graphics2D frameGraph = (Graphics2D)canvas.getRootPane().getGraphics();

//canvas.update(frameGraph);

frameGraph.fillOval(t,canvas.getHeight()/2 +(int)(q\*(1 + Math.cos(w \* t))/2),5,5);

t++;

}});

tm.start();

}

else {

control.setText("Старт");

tm.stop();

Graphics2D frameGraph = (Graphics2D)canvas.getRootPane().getGraphics();

canvas.update(frameGraph);

}

}

catch (Exception ex) {

JOptionPane.showMessageDialog(

null,

"Введите корректные данные.",

"Ошибка",

JOptionPane.ERROR\_MESSAGE

);

}

}

});

}

}

1. Создать апплет со строкой, движущейся по диагонали. При достижении границ апплета все символы строки случайным образом меняют регистр. При этом шрифт меняется на шрифт, выбранный из списка.

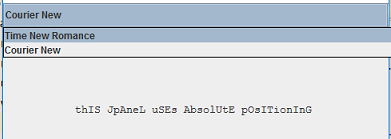


Рис. 10 – Скриншот программы

Листинг:

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Random;

import javax.swing.\*;

public class lab1\_10 {

public static int direction = 0;

public static String fontName = "";

public static void main(String... args) {

JFrame frame = new JFrame("Absolute Layout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel contentPane = new JPanel();

contentPane.setLayout(null);

JLabel label = new JLabel("This JPanel uses Absolute Positioning", JLabel.CENTER);

label.setSize(300, 30);

label.setLocation(0, 30);

String[] fontsArray = {"Time New Romance", "Courier New"};

JComboBox fonts = new JComboBox(fontsArray);

fonts.setSize(500, 30);

fonts.setSelectedIndex(0);

fontName = fonts.getItemAt(0).toString();

fonts.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JComboBox cb = (JComboBox) e.getSource();

fontName = (String) cb.getSelectedItem();

}

});

contentPane.add(label);

contentPane.add(fonts);

frame.setContentPane(contentPane);

frame.setSize(500, 500);

frame.setLocationByPlatform(true);

frame.setVisible(true);

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

Timer timer = new Timer(50, new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

LabelMovie(label);

}

});

timer.start();

}

});

}

public static void LabelMovie(JLabel label) {

int widthPane = label.getParent().getWidth();

int heightPane = label.getParent().getHeight();

int widthLabel = label.getWidth();

int heightLbel = label.getHeight();

int x = label.getX();

int y = label.getY();

int x\_right = x + widthLabel;

int y\_bottom = y + heightLbel;

String text = label.getText();

if (x\_right == widthPane || y\_bottom == heightPane) {

direction = 1;

text = changeReg(text);

} else if (x == 0 || y == 10) {

text = changeReg(text);

direction = 0;

}

label.setText(text);

label.setFont(new Font(fontName, Font.PLAIN, 14));

if (direction == 0) {

label.setLocation(x + 1, y + 1);

} else {

label.setLocation(x - 1, y - 1);

}

}

private static String changeReg(String text) {

int length = text.length();

Random random = new Random();

String new\_text = "";

for (int i = 0; i < length; i++) {

char el = text.charAt(i);

boolean uppeercase = random.nextBoolean();

if (uppeercase) {

new\_text += Character.toString(el).toUpperCase();

} else {

new\_text += Character.toString(el).toLowerCase();

}

}

return new\_text;

}

}

1. Создать апплет со строкой, движущейся горизонтально, отражаясь от границ апплета и меняя при этом свой цвет, на цвет, выбранный из выпадающего списка.

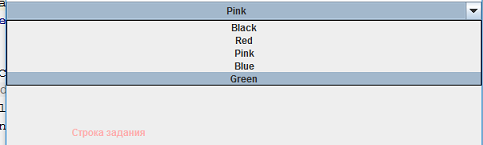


Рис. 11 – Скриншот программы

Листинг:

import org.omg.CORBA.StringHolder;

import javax.swing.\*;

import javax.swing.border.Border;

import javax.swing.event.ListDataListener;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class ColorWithName {

Color color;

String name;

public ColorWithName(){

this.color = new Color(0,0,0);

this.name = "Black";

}

public ColorWithName(int r, int g, int b, String name){

this.color = new Color(r,g,b);

this.name = name;

}

public ColorWithName(Color color, String name){

this.color = color;

this.name = name;

}

String getName(){

return this.name;

}

Color getColor(){

return this.color;

}

}

class ColorsComboBoxRenderer extends JLabel implements ListCellRenderer {

public ColorsComboBoxRenderer() {

setOpaque(true);

setHorizontalAlignment(CENTER);

setVerticalAlignment(CENTER);

}

public Component getListCellRendererComponent(

JList list,

Object value,

int index,

boolean isSelected,

boolean cellHasFocus) {

if (isSelected) {

setBackground(list.getSelectionBackground());

setForeground(list.getSelectionForeground());

} else {

setBackground(list.getBackground());

setForeground(list.getForeground());

}

//Set the icon and text. If icon was null, say so.

ColorWithName active = (ColorWithName)value;

setText(active.getName());

setFont(list.getFont());

return this;

}

}

class ColorsComboBoxModel extends DefaultComboBoxModel<ColorWithName> {

public ColorsComboBoxModel(ColorWithName[] items) {

super(items);

}

@Override

public ColorWithName getSelectedItem() {

ColorWithName selectedColor = (ColorWithName) super.getSelectedItem();

return selectedColor;

}

}

public class lab1\_11 {

public static void main (String args[]) {

JFrame frame = new JFrame("Фут Лаба 1.11");

//frame init

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

frame.setSize(600,300);

frame.setResizable(false);

frame.setLocationRelativeTo(null);

ColorWithName[] colors = new ColorWithName[]{

new ColorWithName(Color.BLACK,"Black"),

new ColorWithName(Color.RED,"Red"),

new ColorWithName(Color.PINK,"Pink"),

new ColorWithName(Color.BLUE,"Blue"),

new ColorWithName(Color.GREEN,"Green"),

};

ColorsComboBoxModel colorsModel = new ColorsComboBoxModel(colors);

ColorsComboBoxRenderer colorsRender = new ColorsComboBoxRenderer();

JComboBox colorSelect = new JComboBox(colorsModel);

colorSelect.setRenderer(colorsRender);

frame.add(colorSelect,BorderLayout.NORTH);

JPanel stringHolder = new JPanel();

frame.add(stringHolder, BorderLayout.CENTER);

JLabel stringPanel = new JLabel("Строка задания");

stringHolder.add(stringPanel);

frame.setVisible(true);

Timer tm = new Timer(10, new ActionListener(){

int sDir = 1,

sX = 0,

sY = frame.getContentPane().getHeight()/2;

@Override

public void actionPerformed(ActionEvent arg0) {

if (sX + stringPanel.getWidth() >= frame.getContentPane().getWidth()) {

sDir = -1;

stringPanel.setForeground(colorsModel.getSelectedItem().getColor());

}

else if (sX <= 0 ) {

sDir = 1;

stringPanel.setForeground(colorsModel.getSelectedItem().getColor());

}

sX += sDir;

stringPanel.setLocation(sX,sY);

}});

tm.start();

}

}

1. Промоделировать вращение спутника вокруг планеты по эллиптической орбите. Когда скрывается за планетой – спутник не виден.



Рис. 12 – Скриншот программы

Листинг:

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class lab1\_12 {

public static Image createThumb(String srcImage, int w, int h) {

ImageIcon imageIcon = new ImageIcon(srcImage);

Image image = imageIcon.getImage();

Image rszImage = image.getScaledInstance(w,h, Image.SCALE\_SMOOTH);

return rszImage;

}

public static void main(String[] args) {

int workspaceWidth = 600,

workspaceHeight = 600,

earthImageWidth = 250,

earthImageHeight = 250,

moonImageWidth = 100,

moonImageHeight = 100;

String earthImagePath = "out/production/ssp\_lab1/images/earth.png",

moonImagePath = "out/production/ssp\_lab1/images/MOON.PNG";

JFrame fr = new JFrame("Фут Лаба 1.12");

fr.setPreferredSize(new Dimension(workspaceWidth,workspaceHeight));

fr.setResizable(false);

final JPanel pan = new JPanel(new BorderLayout());

fr.add(pan);

fr.setVisible(true);

fr.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//add planet image to panel

ImageIcon earthImage = new ImageIcon(lab1\_12.createThumb(earthImagePath,earthImageWidth,earthImageHeight));

JLabel earthLabel = new JLabel("",earthImage,JLabel.CENTER);

pan.add(earthLabel,BorderLayout.CENTER);

ImageIcon moonImage = new ImageIcon(lab1\_12.createThumb(moonImagePath,moonImageWidth,moonImageHeight));

JLabel moonLabel = new JLabel("",moonImage,JLabel.CENTER);

moonLabel.setPreferredSize(new Dimension(200,200));

pan.add(moonLabel, BorderLayout.EAST);

fr.pack();

earthLabel.setLocation(90,0);

Timer tm = new Timer(10, new ActionListener(){

int minX = 180,

minY = 0,

r = 200;

float i = 0.00f;

@Override

public void actionPerformed(ActionEvent arg0) {

moonLabel.setLocation((int)(minX + r\*Math.cos(i)),(int)(minY + r\*Math.sin(i)/1.5));

i += 0.01f;

i = (float)Math.round(i \* 100)/100;

if (i % 0.28 > 0 && i % 0.28 < 0.1) {

if (i % 6.28 > 4.71 || i % 6.28 < 1.57)

moonImage.setImage(lab1\_12.createThumb(moonImagePath,moonImage.getIconWidth() + 1,moonImage.getIconHeight() + 1));

else

moonImage.setImage(lab1\_12.createThumb(moonImagePath,moonImage.getIconWidth() - 1,moonImage.getIconHeight() - 1));

}

if (i % 6.28 > 0.1 && i % 6.28 < 0.2) {

pan.removeAll();

pan.add(moonLabel, BorderLayout.EAST);

pan.add(earthLabel, BorderLayout.CENTER);

} else if (i % 6.28 > 3.1 && i % 6.28 < 3.2) {

pan.add(earthLabel,BorderLayout.CENTER);

pan.add(moonLabel,BorderLayout.EAST);

}

}});

tm.start();

}

}

1. Промоделировать аналоговые часы (со стрелками) с кнопками для увеличения/уменьшения времени на час/минуту.

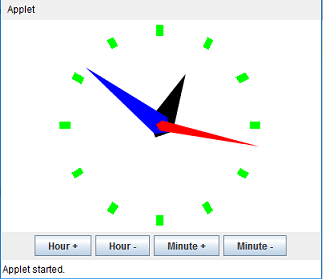


Рис. – Скриншот программы

Листинг:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.image.\*;

import java.awt.geom.\*;

import java.util.Calendar;

public class CircleClock extends java.applet.Applet implements Runnable {

private Needle hNeedle, mNeedle, sNeedle;

private ClockDial dial;

private int w, h;

private BufferedImage bi;

private Graphics2D big;

private boolean stop = false;

private Thread timer = null;

private JPanel btnPanel, clockPanel;

private JButton plusHourBtn, minusHourBtn,

plusMinuteBtn, minusMinuteBtn;

private int hourShift, minuteShift;

private Color dialColor = Color.WHITE;

private Color hairLineColor = Color.GREEN;

private Color hoursColor = Color.BLACK;

private Color minutesColor = Color.BLUE;

private Color secondsColor = Color.RED;

public void init() {

try {

hourShift = minuteShift = 0;

this.setLayout(new BorderLayout());

btnPanel = new JPanel(new FlowLayout());

this.add(btnPanel, BorderLayout.SOUTH);

clockPanel = new JPanel();

plusHourBtn = new JButton("Hour +");

btnPanel.add(plusHourBtn);

plusHourBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

hourShift++;

}

});

minusHourBtn = new JButton("Hour -");

btnPanel.add(minusHourBtn);

minusHourBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

hourShift--;

}

});

plusMinuteBtn = new JButton("Minute +");

btnPanel.add(plusMinuteBtn);

plusMinuteBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

minuteShift++;

}

});

minusMinuteBtn = new JButton("Minute -");

btnPanel.add(minusMinuteBtn);

minusMinuteBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

minuteShift--;

}

});

Dimension dim = getSize();

w = dim.width;

h = dim.height - 35;

setColors();

int radius = 0;

if(w < h)

radius = w/2;

else

radius = h/2;

mNeedle = new Needle(radius, radius/5);

hNeedle = new Needle(2d/3d\*radius, radius/4);

sNeedle = new Needle(radius, radius/10);

dial = new ClockDial(radius\*2, dialColor, hairLineColor);

bi = (BufferedImage)createImage(w, h);

big = bi.createGraphics();

big.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,

RenderingHints.VALUE\_ANTIALIAS\_ON);

timer = new Thread(this);

timer.start();

}

catch(Exception error) {

System.out.println(error.getMessage());

}

}

public void update(Graphics g) {

try {

Graphics2D g2 = (Graphics2D)g;

grawClock();

g2.drawImage(bi, 0, 0, this);

big.setTransform(new AffineTransform());

}

catch(Exception error) {

System.out.println(error.getMessage());

}

}

private void grawClock() {

Calendar curTime = Calendar.getInstance();

curTime.setTimeInMillis(System.currentTimeMillis());

double hour = curTime.get(Calendar.HOUR\_OF\_DAY) + hourShift;

double min = curTime.get(Calendar.MINUTE) + minuteShift;

double sec = curTime.get(Calendar.SECOND);

big.setBackground(Color.WHITE);

big.clearRect(0, 0, w, h);

Point luCorner = new Point(0, 0);

if(w > h)

{

luCorner.x = (w - h)/2;

luCorner.y = 0;

}

else

{

luCorner.x = 0;

luCorner.y = (h - w)/2;

}

big.drawImage(dial.getClockDial(), null, luCorner.x, luCorner.y);

big.setColor(hoursColor);

AffineTransform hat = new AffineTransform();

hat.translate(w/2 - hNeedle.getRotationCenter().x,

h/2 - hNeedle.getRotationCenter().y);

if(hour >= 12)

hour -= 12;

double theta = (-90 + (30\*hour + 0.5\*min))\*Math.PI/180;

hat.rotate(theta, hNeedle.getRotationCenter().x,

hNeedle.getRotationCenter().y);

big.setTransform(hat);

big.fill(hNeedle.getNeedle());

big.setColor(minutesColor);

AffineTransform mat = new AffineTransform();

mat.translate(w/2 - mNeedle.getRotationCenter().x,

h/2 - mNeedle.getRotationCenter().y);

theta = (-90 + (6\*min + 0.1\*sec))\*Math.PI/180;

mat.rotate(theta, mNeedle.getRotationCenter().x,

mNeedle.getRotationCenter().y);

big.setTransform(mat);

big.fill(mNeedle.getNeedle());

big.setColor(secondsColor);

AffineTransform sat = new AffineTransform();

sat.translate(w/2 - sNeedle.getRotationCenter().x,

h/2 - sNeedle.getRotationCenter().y);

theta = (-90 + (6\*sec))\*Math.PI/180;

sat.rotate(theta, sNeedle.getRotationCenter().x,

sNeedle.getRotationCenter().y);

big.setTransform(sat);

big.fill(sNeedle.getNeedle());

}

private void setColors() {

try {

Long dc = new Long(Long.parseLong(getParameter("dialColor"), 16));

Long hlc = new Long(Long.parseLong(getParameter("hairLineColor"), 16));

Long hnc = new Long(Long.parseLong(getParameter("hoursNeedleColor"), 16));

Long mnc = new Long(Long.parseLong(getParameter("minutesNeedleColor"), 16));

Long snc = new Long(Long.parseLong(getParameter("secondsNeedleColor"), 16));

dialColor = new Color(dc.intValue(), true);

hairLineColor = new Color(hlc.intValue(), true);

hoursColor = new Color(hnc.intValue(), true);

minutesColor = new Color(mnc.intValue(), true);

secondsColor = new Color(snc.intValue(), true);

}

catch(Exception error) {

System.out.println(error.getMessage());

}

}

public void run() {

while(!stop) {

try {

repaint();

Thread.currentThread().sleep(1000);

}

catch(Exception err) {}

}

}

public void stop() {

super.stop();

stop = true;

}

public void start() {

super.start();

stop = false;

if(timer == null) {

timer = new Thread(this);

timer.start();

}

}

public void destroy() {

super.destroy();

stop = true;

Thread.currentThread().yield();

}

public String[][] getParameterInfo() {

String[][] retValue = {

{"dialColor", "hex", "alfa, red, green, blue (4 bytes, from 00 to FF)"},

{"hairLineColor", "hex", "alfa, red, green, blue (4 bytes, from 00 to FF)"},

{"hoursNeedleColor", "hex", "alfa, red, green, blue (4 bytes, from 00 to FF)"},

{"minutesNeedleColor", "hex", "alfa, red, green, blue (4 bytes, from 00 to FF)"},

{"secondsNeedleColor", "hex", "alfa, red, green, blue (4 bytes, from 00 to FF)"}

};

return retValue;

}

public String getAppletInfo() {

String retValue;

retValue = "Circle Clock";

return retValue;

}

}