Міністерство освіти і науки України

Київський політехнічний інститут ім. Ігоря Сікорського

Теплоенергетичний факультет

Кафедра АПЕПС

ЗВІТ

про виконання лабораторної роботи №5

з дисципліни «Технології конструювання програмного забезпечення»

**«Построение графических интерфейсов с помощью инструментов Swing»**

**Виконала:**

Студентка 2 курсу, групи ТІ-01

Круть Катерина Олександрівна

Дата: 20.12.21

**Перевірив:**

доцент, к.ф.-м.н.

Тарнавський Ю.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Варіант №14

Завдання: 3,5,19,29

**Завдання №3:**

import java.awt.Color;  
import java.awt.Graphics;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.JCheckBoxMenuItem;  
import javax.swing.JFrame;  
import javax.swing.JMenu;  
import javax.swing.JMenuBar;  
import javax.swing.JMenuItem;  
import javax.swing.JPanel;  
  
public class task\_3 extends JFrame implements ActionListener {  
 private JPanel Print;  
 private JMenuBar menuBar;  
 private JMenu parameter;  
 private JMenu change;  
 private JMenuItem increase;  
 private JMenuItem reduce;  
 private JCheckBoxMenuItem width;  
 private JCheckBoxMenuItem height;  
 protected int rect\_width;  
 protected int rect\_height;  
  
 public task\_3() {  
 super("Rectangle");  
 this.$$$setupUI$$$();  
 this.menuBar = new JMenuBar();  
 this.parameter = new JMenu("parameter");  
 this.change = new JMenu("change");  
 this.increase = new JMenuItem("Увеличить");  
 this.reduce = new JMenuItem("Уменьшить");  
 this.width = new JCheckBoxMenuItem("Ширина");  
 this.height = new JCheckBoxMenuItem("Высота");  
 this.setDefaultCloseOperation(3);  
 this.add(this.Print);  
 this.add(this.menuBar, "North");  
 this.pack();  
 this.setSize(500, 500);  
 this.rect\_width = 150;  
 this.rect\_height = 150;  
 this.parameter.add(this.width);  
 this.parameter.add(this.height);  
 this.change.add(this.increase);  
 this.change.add(this.reduce);  
 this.menuBar.add(this.parameter);  
 this.menuBar.add(this.change);  
 this.width.setActionCommand("Ширина");  
 this.height.setActionCommand("Высота");  
 this.increase.setActionCommand("Увеличить");  
 this.reduce.setActionCommand("Уменьшить");  
 this.height.addActionListener(this);  
 this.increase.addActionListener(this);  
 this.reduce.addActionListener(this);  
 this.width.addActionListener(this);  
 this.Print.setBackground(new Color(255, 255, 255));  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 String command = e.getActionCommand();  
 if (command.equals("Увеличить")) {  
 if (this.width.isSelected()) {  
 this.rect\_width += 10;  
 }  
  
 if (this.height.isSelected()) {  
 this.rect\_height += 10;  
 }  
 }  
  
 if (command.equals("Уменьшить")) {  
 if (this.width.isSelected()) {  
 this.rect\_width -= 10;  
 }  
  
 if (this.height.isSelected()) {  
 this.rect\_height -= 10;  
 }  
 }  
  
 this.repaint();  
 }  
  
 private void createUIComponents() {  
 this.Print = new task\_3.imagePanel();  
 }  
  
 public static void main(String[] args) {  
 (new task\_3()).setVisible(true);  
 }  
  
 class imagePanel extends JPanel {  
 imagePanel() {  
 }  
  
 public void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 g.drawRect(50, 50, task\_3.this.rect\_width, task\_3.this.rect\_height);  
 }  
 }  
}

**Завдання №5:**

//3. Изменение размера фигуры в графическом окне. В окне определена строка меню  
// (JMenuBar), в которой определены меню (JMenu) "Параметр" и "Изменение". В меню  
// "Параметр" определены два пункта меню (JCheckBoxMenuItem): "Ширина" (по умолчанию  
// включена) и "Высота" (по умолчанию включена). В меню "Изменение" определены два  
// пункта меню (JMenuItem): "Увеличить" и "Уменьшить". В центре панели (JPanel) окна  
// "Вывод изображения" в графическом контексте задается (с помощью метода drawRect())  
// прямоугольник. При выборе одного из пунктов второго меню фигура скачкообразно  
// (на 10 пикселей) увеличивается или уменьшается для заданного параметра или параметров  
// в первом меню и перерисовывается. Если в первом меню не включен ни один из пунктов,  
// фигура не меняется.  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class task\_3 extends JFrame implements ActionListener {  
 private JPanel Print;  
 private JMenuBar menuBar = new JMenuBar();  
 private JMenu parameter = new JMenu("parameter");  
 private JMenu change = new JMenu("change");  
 private JMenuItem increase = new JMenuItem("Увеличить");  
 private JMenuItem reduce= new JMenuItem("Уменьшить");  
 private JCheckBoxMenuItem width = new JCheckBoxMenuItem("Ширина");  
 private JCheckBoxMenuItem height= new JCheckBoxMenuItem("Высота");  
  
 protected int rect\_width, rect\_height;  
  
 public task\_3() {  
 super("Rectangle");  
 this.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 add(Print);  
 add(menuBar, BorderLayout.*NORTH*);  
 pack();  
 setSize(500, 500);  
 rect\_width = 150;  
 rect\_height = 150;  
 parameter.add(width);  
 parameter.add(height);  
 change.add(increase);  
 change.add(reduce);  
 menuBar.add(parameter);  
 menuBar.add(change);  
 width.setActionCommand("Ширина");  
 height.setActionCommand("Высота");  
 increase.setActionCommand("Увеличить");  
 reduce.setActionCommand("Уменьшить");  
  
 height.addActionListener(this);  
 increase.addActionListener(this);  
 reduce.addActionListener(this);  
 width.addActionListener(this);  
  
 Print.setBackground(new java.awt.Color(255, 255, 255));  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 String command = e.getActionCommand();  
 if(command.equals("Увеличить")){  
 if(width.isSelected()){  
 rect\_width += 10;  
 }  
 if(height.isSelected()){  
 rect\_height += 10;  
 }  
 }  
 if(command.equals("Уменьшить")){  
 if(width.isSelected()){  
 rect\_width -= 10;  
 }  
 if(height.isSelected()){  
 rect\_height -= 10;  
 }  
 }  
 repaint();  
 }  
  
 private void createUIComponents() {Print = new imagePanel();}  
  
 class imagePanel extends JPanel {  
 @Override  
 public void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 g.drawRect(50, 50, rect\_width, rect\_height);  
 }  
 }  
 public static void main(String[] args) {  
 new task\_3().setVisible(true);  
 }  
}

**Завдання №19:**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.Objects;  
  
//19. Изменение фигуры в графическом окне. В центре верхней панели (JPanel)  
// "Вывод треугольника" в графическом контексте рисуется (с помощью метода  
// drawPolygon()) треугольник. В нижней панели (JPanel) "Параметры треугольника"  
// задаются следующие компоненты: надпись (JLabel) "Цвет:", раскрывающееся меню  
// (JComboBox) с пунктами: "Черный" (цвет по умолчанию), "Красный", "Зеленый" и  
// "Синий", надпись (JLabel) "Номер вершины:", раскрывающееся меню (JComboBox)  
// с пунктами "1", "2" и "3", надпись (JLabel) "Координата X:", вращающийся  
// список (JSpinner), надпись (JLabel) "Координата Y:", вращающийся список  
// (JSpinner) и кнопка (JButton) "Изменить треугольник". При наборе данных во  
// вращающихся списках (в пикселях) и при нажатии кнопки "Изменить треугольник"  
// треугольник перерисовывается заданным цветом и с заданным новым положением  
// одной из вершин. Диапазон изменения вращающихся списков – от 0 до 150 с шагом 1.  
public class task19 extends JFrame implements ActionListener {  
 private JPanel menu;  
 private JPanel Print;  
 private JComboBox choose\_color;  
 private JSpinner choose\_x;  
 private JSpinner choose\_y;  
 private JButton change;  
 private JComboBox choose\_point;  
 private JLabel col;  
 private JLabel numb;  
 private JLabel label\_x;  
 private JLabel label\_y;  
 protected String[] Colors = {"Black", "Red", "Green", "Blue" };  
 protected Integer[] points = {1, 2, 3};  
 protected Color color;  
 protected int x[] = {100, 200, 400};  
 protected int y[] = {100, 400, 200};  
 protected int new\_x, new\_y;  
  
 public task19(){  
 super("Triangle");  
 this.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 choose\_color.setModel(new DefaultComboBoxModel(Colors));  
 choose\_color.setSelectedIndex(0);  
 choose\_point.setModel(new DefaultComboBoxModel(points));  
 add(Print);  
 add(menu, BorderLayout.*SOUTH*);  
 pack();  
 setSize(700, 700);  
 color = new Color(0,0,0);  
 choose\_x.setModel(new SpinnerNumberModel(75, 0, 150, 1));  
 choose\_y.setModel(new SpinnerNumberModel(75, 0, 150, 1));  
  
 choose\_color.setActionCommand("Цвет");  
 choose\_point.setActionCommand("Вершина");  
 change.setActionCommand("Изменить");  
 choose\_color.addActionListener(this);  
 choose\_point.addActionListener(this);  
 change.addActionListener(this);  
 Print.setBackground(new java.awt.Color(255, 255, 255));  
 choose\_x.addChangeListener(e -> {  
 JSpinner spinner = (JSpinner) e.getSource();  
 new\_x = (int)spinner.getValue();  
 System.*out*.println("X is " + new\_x);  
 });  
 choose\_y.addChangeListener(e -> {  
 JSpinner spinner = (JSpinner) e.getSource();  
 new\_y = (int)spinner.getValue();  
 System.*out*.println("Y is " + new\_y);  
 });  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String command = e.getActionCommand();  
 switch (command) {  
 case "Цвет" -> {  
 String color1 = Objects.*requireNonNull*(choose\_color.getSelectedItem()).toString();  
 switch(color1) {  
 case "Black" -> color = Color.*BLACK*;  
 case "Red" -> color = Color.*RED*;  
 case "Green" -> color = Color.*GREEN*;  
 case "Blue" -> color = Color.*BLUE*;  
 }  
 }  
 case "Вершина" -> {  
 Integer point = (Integer) Objects.*requireNonNull*(choose\_point.getSelectedItem());  
 switch (point) {  
 case 1 -> {  
 x[0] = new\_x;  
 y[0] = new\_y;  
 }  
 case 2 -> {  
 x[1] = new\_x;  
 y[1] = new\_y;  
 }  
 case 3 -> {  
 x[2] = new\_x;  
 y[2] = new\_y;  
 }  
 }  
 }  
 }  
 if (command.equals("Изменить")) {  
 repaint();  
 }  
  
 }  
 private void createUIComponents() {Print = new imagePanel();}  
  
 class imagePanel extends JPanel {  
 @Override  
 public void paintComponent(Graphics g) {  
 super.paintComponents(g);  
 g.setColor(color);  
 g.drawPolygon(x, y, 3);  
 }  
 }  
  
  
 public static void main(String[] args) { new task19().setVisible(true); }  
}

**Завдання №29:**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
/\*29. Изменение размера изображения в графическом окне. В верхней панели  
 окна (JPanel) "Размер изображения" заданы следующие компоненты: надпись  
 (JLabel) "Ширина:", текстовое поле (JTextField), надпись (JLabel)  
 "Высота:", текстовое поле (JTextField) и кнопка (JButton): "Вывести  
 изображение". В нижней панели окна (JPanel) окна "Вывод изображения" в  
 компоненте (JLabel) задается произвольное изображение. При вводе размера  
 изображения по ширине и высоте и нажатии кнопки "Вывести изображение" в  
 нижней панели выводится масштабированное изображение с заданными размерами  
 (в пикселях). Первоначально изображение имеет «естественный» размер. Перед  
 выводом изображения выполняется проверка, заданы ли значения ширины и высоты  
 и являются ли они целыми числами.\*/  
public class task29 extends JFrame implements ActionListener {  
 private JTextField width;  
 private JTextField height;  
 private JButton print\_image;  
 private JLabel l\_width;  
 private JLabel l\_height;  
 private JPanel menu;  
 private JPanel Print;  
 private JLabel l\_image;  
 protected int img\_width = 250, img\_height = 250;  
  
 public task29 (){  
 super("Image");  
 this.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 add(Print);  
 add(menu, BorderLayout.*SOUTH*);  
 pack();  
 setSize(800, 800);  
 print\_image.addActionListener(this);  
 Print.setBackground(new java.awt.Color(255, 255, 255));  
  
 }  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if(!width.getText().isEmpty() && !height.getText().isEmpty() && Integer.*parseInt*(width.getText()) > 0 &&  
 Integer.*parseInt*(height.getText()) > 0) {  
 img\_width = Integer.*parseInt*(width.getText());  
 img\_height = Integer.*parseInt*(height.getText());  
  
 }  
 Image i = new ImageIcon("stitch.png").getImage().getScaledInstance(img\_width,  
 img\_height, Image.*SCALE\_DEFAULT*);  
 ImageIcon icon = new ImageIcon(i);  
 icon.getImage().flush();  
 l\_image.setIcon(icon);  
 l\_image.setVisible(true);  
 }  
 private void createUIComponents() {  
 Image icon = new ImageIcon("stitch.png").getImage().getScaledInstance(img\_width,  
 img\_height, Image.*SCALE\_DEFAULT*);  
 l\_image = new JLabel(new ImageIcon(icon));  
 l\_image.setVisible(false);  
 }  
 public static void main(String[] args){  
 new task29().setVisible(true);  
 }  
  
}