Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

Звіт

З лабораторної роботи №8

З предмету «Об’єктно орієнтоване програмування»

Виконав

Студент ІІ курсу

Група ПЗ-154

Линник В.Ю.

Перевірив

Викладач кафедри

Програмного забезпечення автоматизованих систем

Крайовий В. М.

Виконав

Студент І курсу

Група ПЗ-154

Кравченко Артем Олегович

ЧЕРКАСИ 2016

**Тема:** Композиція.

**Мета:** Зрозуміти принципи та призначення композиції та навчитись використовувати її при побудові класів.

**Постановка завдання:**

В даній роботі будуть використовуватись результати попередніх робіт.

1. Взяти один клас з попередніх робіт та розробити другий клас відповідно до варіанту. При розробці другого класу використати принцип композиції, зробивший його полем об'єкт або масив чи колекцію об’єктів першого класу.

2. Розробити третій клас, в якому реалізувати інтерфейс роботи з об’єктом другого класу. В ньому реалізувати функції:

- відображення даного об’єкту;

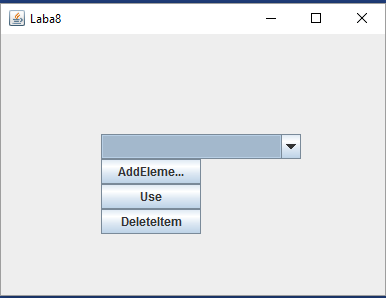
- додавання до нього об’єкту першого класу;

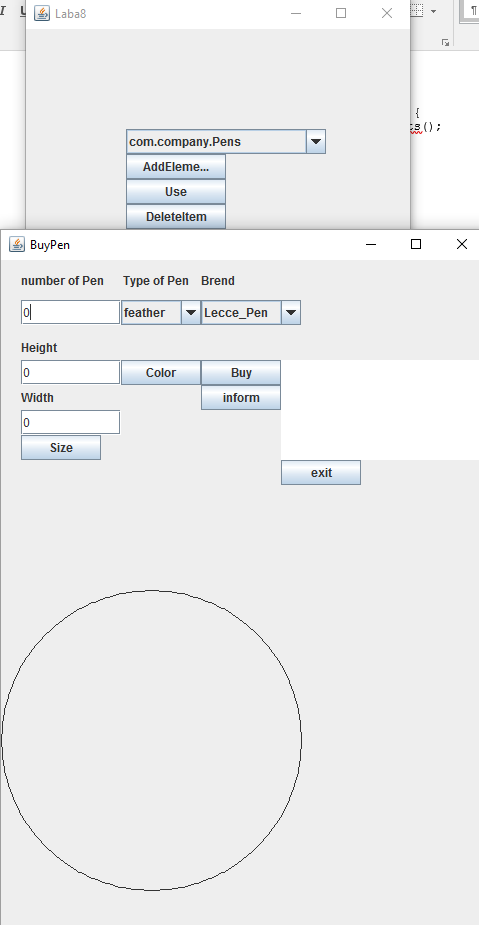
- проведення дій над об’єктом першого класу як частини другого класу.

Індивідуальне завдання: Наручний годинник та Людина

**Виконання лабораторної роботи:**

package com.company;  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.util.\*;  
import java.util.List;  
import java.lang.String;  
class Pens extends JPanel {  
  
 Color color;  
 String Brend;  
 String TypeOfPen;  
 int numbOfPen;  
 int Width;  
 int Height;  
  
 Pens() {  
 numbOfPen = 0;  
 Brend = "Lecce\_Pen";  
 TypeOfPen = "feather";  
 Height = 300;  
 Width = 300;  
  
 }  
  
  
  
  
 public String toString() {  
 return " Pen : " + "\n" +  
 "Number of Pen = " + numbOfPen + "\n" + " Type = " + TypeOfPen + "\n" + "Brend = " + Brend + "\n";  
 }  
  
 void setPenWidth(int PenWingt) {  
 this.Width = PenWingt;  
 }  
  
 void setPenHeight(int PenHeight) {  
 this.Height = PenHeight;  
 }  
  
 void setColor(Color col) {  
 this.color = col;  
 }  
  
  
  
 public void paint(Graphics g) {  
 int h = Height;  
 int w = Width;  
 g.setColor(color);  
 g.drawOval(0, 30, h, w);  
  
  
 }  
  
  
  
  
 void setAll(int Numb , String Bren , String Type)  
 {  
 numbOfPen = Numb ;  
 TypeOfPen = Type;  
 Brend = Bren;  
 }  
  
 String information()  
 {  
 String i = "Number of Pen = " + numbOfPen + "\n" + " Type = " + TypeOfPen + "\n" + "Brend = " + Brend + "\n";  
 return i ;  
 }  
  
 public void draw\_window(Pens pen ) {  
 String[] Brend = {  
 "Lecce\_Pen",  
 "Parker\_Pen",  
 "AURORA"  
 };  
  
  
 String[] Type = {  
 "feather",  
 "rollers",  
 "ball-point"  
 };  
  
  
 JFrame windowTwo = new JFrame("BuyPen");  
 windowTwo.setSize(500, 800);  
  
 windowTwo.setLocationRelativeTo(null);  
 windowTwo.setLayout(null);  
 windowTwo.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
  
 //btBuy  
 JButton btBuy = new JButton("Buy");  
 btBuy.setBounds(200, 100, 80, 25);  
 windowTwo.add(btBuy);  
  
 //inform  
 JButton btInf = new JButton("inform");  
 btInf.setBounds(200, 125, 80, 25);  
 windowTwo.add(btInf);  
  
 // btSize  
 JButton btSize = new JButton("Size");  
 btSize.setBounds(20, 175, 80, 25);  
 windowTwo.add(btSize);  
  
 //draw  
 pen.setBounds(0, 300, 800, 800);  
 windowTwo.add(pen);  
  
 //Height  
 JLabel HLabel = new JLabel("Height");  
 HLabel.setBounds(20, 75, 100, 25);  
 windowTwo.add(HLabel);  
  
 JTextField TxtHeight = new JTextField("0");  
 TxtHeight.setBounds(20, 100, 100, 25);  
 windowTwo.add(TxtHeight);  
 //Width  
 JLabel WLabel = new JLabel("Width");  
 WLabel.setBounds(20, 125, 100, 25);  
 windowTwo.add(WLabel);  
  
 JTextField TxtWidth = new JTextField("0");  
 TxtWidth.setBounds(20, 150, 100, 25);  
 windowTwo.add(TxtWidth);  
  
 //color  
  
 JButton btColor = new JButton("Color");  
 btColor.setBounds(120, 100, 80, 25);  
 windowTwo.add(btColor);  
  
  
 JLabel LPen = new JLabel("number of Pen");  
 LPen.setBounds(20, 5, 100, 30);  
 windowTwo.add(LPen);  
  
 JTextField numbPen = new JTextField("0");  
 numbPen.setBounds(20, 40, 100, 25);  
 windowTwo.add(numbPen);  
  
  
 //type  
 JLabel LType = new JLabel("Type of Pen");  
 LType.setBounds(122, 5, 80, 30);  
 windowTwo.add(LType);  
  
 JComboBox BoxTypeVar = new JComboBox(Type);  
 BoxTypeVar.setBounds(120, 40, 80, 25);  
 windowTwo.add(BoxTypeVar);  
  
 //brend  
 JLabel LBrend = new JLabel("Brend");  
 LBrend.setBounds(200, 5, 100, 30);  
 windowTwo.add(LBrend);  
  
 JComboBox BoxBreVar = new JComboBox(Brend);  
 BoxBreVar.setBounds(200, 40, 100, 25);  
 windowTwo.add(BoxBreVar);  
  
 //inform  
 JTextArea Information = new JTextArea();  
 Information.setBounds(280, 100, 200, 100);  
 windowTwo.add(Information);  
  
 //btExit  
 JButton btExit = new JButton("exit");  
 btExit.setBounds(280, 200, 80, 25);  
 windowTwo.add(btExit);  
  
  
 btExit.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 windowTwo.dispose();  
 //wind.setVisible(true);  
  
 }  
 });  
  
  
 btColor.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 Color choosenColor = JColorChooser.*showDialog*(null, "Choose color", Color.*white*);  
 if (choosenColor != null) {  
 pen.setColor(choosenColor);  
 pen.setOpaque(true);  
  
 windowTwo.repaint();  
  
 }  
 }  
 });  
  
 btSize.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
  
 pen.setPenHeight(Integer.*valueOf*(TxtHeight.getText().toString()));  
 pen.setPenWidth(Integer.*valueOf*(TxtWidth.getText().toString()));  
 windowTwo.repaint();  
 }  
 });  
  
  
 btBuy.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
  
 pen.setAll(Integer.*valueOf*(numbPen.getText().toString()),BoxBreVar.getSelectedItem().toString(),BoxTypeVar.getSelectedItem().toString());  
 }  
 });  
  
 btInf.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 Information.setText(null);  
  
 Information.append( pen.information() + "");  
 }  
 });  
 windowTwo.setVisible(true);  
 }  
  
  
  
  
  
}  
  
  
  
class Pocket  
{  
 Pens Pensil = new Pens();  
  
  
 void draw()  
 {  
 Pensil.draw\_window(Pensil);  
 }  
}  
  
  
class All\_elements {  
 void draw\_all() {  
 JFrame window = new JFrame("Laba8");  
 window.setSize(400, 300);  
 window.setLocationRelativeTo(null);  
  
 window.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
  
 window.setLayout(null);  
  
 JButton btAddElement = new JButton("AddElement");  
 btAddElement.setBounds(100, 125, 100, 25);  
 window.add(btAddElement);  
  
  
 JComboBox ComboBoxV = new JComboBox();  
 ComboBoxV.setBounds(100, 100, 200, 25);  
 window.add(ComboBoxV);  
  
  
 JButton btDelItem = new JButton("DeleteItem");  
 btDelItem.setBounds(100, 175, 100, 25);  
 window.add(btDelItem);  
  
 JButton btUseItem = new JButton("Use");  
 btUseItem.setBounds(100, 150, 100, 25);  
 window.add(btUseItem);  
  
  
 window.setVisible(true);  
  
  
 List Pensils = new ArrayList();  
 Pocket pocket = new Pocket();  
  
  
 btAddElement.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int i = 0;  
 Pensils.add(pocket.Pensil);  
 ComboBoxV.addItem(Pensils.get(i).getClass().getName());  
 i++;  
  
 }  
 });  
  
  
 btUseItem.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
  
 if(ComboBoxV.getSelectedItem() != null)  
 {  
  
 ((Pens)Pensils.get(ComboBoxV.getSelectedIndex())).draw\_window(pocket.Pensil);  
 }  
 }  
 });  
  
 btDelItem.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if(ComboBoxV.getSelectedItem() != null)  
 {  
 ComboBoxV.removeItem(ComboBoxV.getSelectedItem());  
 Pensils.remove(ComboBoxV.getSelectedItem());  
 }  
 }  
 });  
 }  
}  
public class Main {  
  
 public static void main(String[] args) {  
 All\_elements elem = new All\_elements();  
 elem.draw\_all();  
 }}

**Результат виконання програми:** 



**Висновок:** під час виконання цієї лабораторної роботи, я навчився застосовувати композицію у своїй програмі.