Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

Звіт

З лабораторної роботи №1

З предмету «Об’єктно орієнтоване програмування»

Перевірив

Викладач кафедри

Програмного забезпечення автоматизованих систем

Долецький К. Ю.

Виконав

Студент І курсу

Група ПЗ-154

Кравченко Артем Олегович

Виконав

Студент ІІ курсу

Група ПЗ-154

Кравченко А. О.

ЧЕРКАСИ 2017

**Тема:** Використання моделей конструювання та планування конструювання при розробці програм. Створення об'єктно-орієнтованих програм у відповідності з правилами написання якісного коду.

**Мета:** Набути навичок побудови програм у відповідності з правилами написання якісного коду.

**Постановка завдання:**

Створити програми, які реалізують наступну функціональність. При створенні програм користуватись правилами написання якісного коду.

1. Квадрат, який рухається всередині вікна по діагоналі і "відбивається" від стінок під кутом, "дзеркальним" до кута руху до стінки.
2. На екрані з’являються різнокольорові прямокутники з однаковою шириною і різною висотою, яка задається генератором випадкових чисел. При натисненні кнопки вишикувати дані прямокутники за зростанням висоти.

**Теоретичні відомості.**

Принципи правильної побудови коду:

Рівень класів:

* розумна абстракція;
* якісний інтерфейс класу;
* розумне об’єднання класів в пакети.

Рівень методів:

* розумна причина для створення методу;
* вдале ім’я методу;
* розумний розмір методу;
* розумний список параметрів методу;
* обгрунтований тип значення, що повертається;
* обгрунтоване використання ключового слова final.

Рівень змінних:

* грамотне оголошення змінних;
* грамотна ініціалізація змінних;
* поміщення змінної у відповідну їй область видимості;
* одиничність мети кожної змінної;
* грамотне іменування змінної.

Рівень операторів:

* грамотний вибір умовного оператора (if або case);
* грамотний вибір оператора циклу;
* відповідне використання нестандартних структур керування (наприклад, рекурсії);
* подолання занадто великої вкладеності.

Рівень констант:

* завжди підставляти константи на місце "магічних чисел";
* вдале ім’я константи.

Загальний рівень:

* форматування коду.

**Виконання лабораторної роботи:**

**Лістинг програми:**

Клас MainFrame – найперша форма, яку бачить користувач. Призначена для вибору подальшого шляху роботи програми.

package lab1.util;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class MainFrame {  
  
 public static void main(String[] args) {  
 JFrame mainWindow = new JFrame("LAB 1| MENU");  
 mainWindow.setLayout(null);  
 mainWindow.setSize(250,300);  
 mainWindow.setResizable(false);  
 mainWindow.getContentPane().setBackground(Color.*cyan*);  
 mainWindow.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JButton task1Button = new JButton("TASK 1");  
 task1Button.setSize(150,30);  
 task1Button.setLocation(50,70);  
 task1Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 lab1.task1.Window winTask1 = new lab1.task1.Window();  
 winTask1.drawWindow();  
 mainWindow.dispose();  
 }  
 });  
  
 JButton task2Button = new JButton("TASK 2");  
 task2Button.setSize(150,30);  
 task2Button.setLocation(50,170);  
 task2Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 lab1.task2.Window winTask2 = new lab1.task2.Window();  
 winTask2.drawWindow();  
 mainWindow.dispose();  
 }  
 });  
  
  
 mainWindow.add(task1Button);  
 mainWindow.add(task2Button);  
 mainWindow.setVisible(true);  
  
 }//End of method  
}//End of class

Клас Rectungle – клас, призначений для малювання і руху квадрату по певній площині.

package lab1.task1;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.Random;  
  
  
public class Rectungle extends JPanel {  
 int coordX, coordY, width, height;  
 Point maxCoordinates;  
 Component formForRepaint;  
  
 public Rectungle(Point startCoord, Point maxCoord, Component form){  
 coordX = (int)startCoord.getX();  
 coordY = (int)startCoord.getY();  
 maxCoordinates = maxCoord;  
 width = 45;  
 height = 45;  
 formForRepaint = form;  
 }//End of constructor  
  
 @Override  
 public void paint(Graphics g){  
 g.fillRect(coordX, coordY,width,height);  
 }//End of method  
  
  
 int incX = 1, incY = 1;  
 public void moveRectangle(){  
 Timer timer = new Timer(10,null);  
  
 ActionListener timerAction = new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 //Move of the Rectungle  
 coordX += incX;  
 coordY += incY;  
  
 //Valid of the X coordinate  
 if(coordX + width >= maxCoordinates.getX()) {  
 incX = -1;  
 changeFieldColor();  
 }  
 if(coordX <= 0) {  
 incX = 1;  
 changeFieldColor();  
 }  
  
 //Valid of the Y coordinate  
 if(coordY + height >= maxCoordinates.getY()) {  
 incY = -1;  
 changeFieldColor();  
 }  
 if(coordY <= 0) {  
 incY = 1;  
 changeFieldColor();  
 }  
  
  
 formForRepaint.repaint();  
 formForRepaint.validate();  
 }  
 };  
  
 timer.addActionListener(timerAction);  
 timer.start();  
 }//End of method  
  
  
  
 private void changeFieldColor(){  
 Color [] fieldColor = {Color.*RED*, Color.*BLUE*, Color.*cyan*, Color.*DARK\_GRAY*, Color.*green*, Color.*YELLOW*};  
 Random random = new Random();  
  
 formForRepaint.setBackground(fieldColor[random.nextInt(6)]);  
 }//End of method  
  
}//End of class

Клас Window – клас, що призначений для відображення результату роботи класу Rectungle для завдання №1.

package lab1.task1;  
  
  
import lab1.util.InputDialog;  
  
import javax.swing.\*;  
import javax.swing.border.Border;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Window {  
  
 public void drawWindow(){  
 JFrame window = new JFrame("lab\_1");  
 window.setLayout(null);  
 window.setSize(900,536);  
 window.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 //The util panel for moving of the cube  
 JPanel panel = new JPanel(new BorderLayout());  
 panel.setSize(550,400);  
 panel.setLocation(30,60);  
 Border lineBorder = BorderFactory.*createLineBorder*(Color.*BLUE*);  
 panel.setBorder(lineBorder);  
 panel.setOpaque(true);  
 panel.setBackground(Color.*RED*);  
  
  
 //Preparing the coordinates for the rectungle.  
 Point startPosition = new Point();  
 startPosition.setLocation(0,0);  
 Point coordEndOfField = new Point();  
 coordEndOfField.setLocation(panel.getWidth(),panel.getHeight());  
  
  
 //Info labels  
 JLabel viewportLabel = new JLabel("VIEWPORT:");  
 viewportLabel.setFont( new Font("Verdana", Font.*PLAIN*, 20));  
 viewportLabel.setSize(viewportLabel.getPreferredSize());  
 viewportLabel.setLocation(90, 25);  
  
 JLabel settingLabel = new JLabel("SETTING:");  
 settingLabel.setFont( new Font("Verdana", Font.*PLAIN*, 20));  
 settingLabel.setSize(settingLabel.getPreferredSize());  
 settingLabel.setLocation(645, 25);  
  
 JLabel inputXLabel = new JLabel("<html>Please, input X coordinate:<br><br>Now X is </html>");  
 inputXLabel.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 inputXLabel.setSize(inputXLabel.getPreferredSize());  
 inputXLabel.setLocation(620, 100);  
  
 JLabel inputYLabel = new JLabel("<html>Please, input Y coordinate:<br><br>Now Y is </html>");  
 inputYLabel.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 inputYLabel.setSize(inputYLabel.getPreferredSize());  
 inputYLabel.setLocation(620, 250);  
  
  
 //Creation of Buttons  
 JButton inputXButton = new JButton("" + (int)startPosition.getX());  
 inputXButton.setSize(130,30);  
 inputXButton.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 inputXButton.setLocation(710,138);  
 inputXButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 InputDialog xCoordDialog = new InputDialog();  
 startPosition.setLocation(xCoordDialog.showInputDialog(), startPosition.getY());  
 inputXButton.setText("" + startPosition.getX());  
 }  
 });  
  
 JButton inputYButton = new JButton("" + (int)startPosition.getY());  
 inputYButton.setSize(130,30);  
 inputYButton.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 inputYButton.setLocation(710,288);  
 inputYButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 InputDialog yCoordDialog = new InputDialog();  
 startPosition.setLocation(startPosition.getX(), yCoordDialog.showInputDialog());  
 inputYButton.setText("" + startPosition.getY());  
 }  
 });  
  
 JButton startButton = new JButton("START");  
 startButton.setSize(220,40);  
 startButton.setFont( new Font("Verdana", Font.*PLAIN*, 24));  
 startButton.setLocation(620,385);  
 startButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 Rectungle cube = new Rectungle(startPosition,coordEndOfField, panel);  
 panel.add(cube);  
 cube.moveRectangle();  
 window.repaint();  
 window.validate();  
 }  
 });  
  
  
 //add component to the Window  
 window.add(panel);  
  
 window.add(viewportLabel);  
 window.add(settingLabel);  
 window.add(inputXLabel);  
 window.add(inputYLabel);  
  
 window.add(inputXButton);  
 window.add(inputYButton);  
 window.add(startButton);  
  
 window.setVisible(true);  
 }//End of method  
  
}//End of class

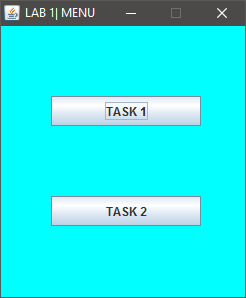
Клас Diagram – призначений для створення діаграми із прямокутників. Кожний прямокутник має випадкову висоту, але однакову ширину. Клас виконує сортування діаграми за висотою в окремому методі.

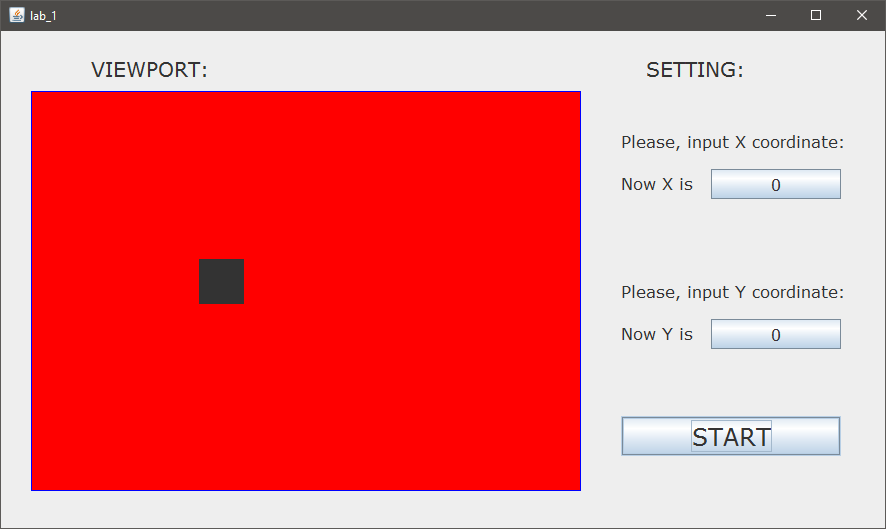
package lab1.task2;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.util.Random;  
  
  
public class Diagram extends JPanel{  
  
 int numberOfCols;  
 int [] heightCol;  
 int widthCol, widthSpace, heightSpace;  
  
 Diagram(int numberCols, int widthSpace, int heightSpace){  
 this.numberOfCols = numberCols;  
 this.widthSpace = widthSpace;  
 this.heightSpace = heightSpace;  
  
  
 Random rand = new Random();  
  
 widthCol = widthSpace/numberCols;  
 heightCol = new int[numberCols];  
 for (int i = 0; i < numberCols; i++) {  
 heightCol[i] = rand.nextInt(heightSpace);  
 }  
 }//End of Constructor  
  
 @Override  
 public void paint(Graphics g){  
 for(int i = 0, xCoord = 0;i < numberOfCols; i++, xCoord +=(widthCol+1) ){  
 g.fillRect(xCoord,heightSpace-heightCol[i],widthCol,heightCol[i]);  
 }  
 }//End of method  
  
 void sortRectungleByheight(){  
 int bufer;  
 int i;  
  
 for(int j = 1; j < numberOfCols; j++) {  
 for (i = 0; i < numberOfCols - j; i++) {  
 if (heightCol[i] > heightCol[i + 1]) {  
 bufer = heightCol[i];  
 heightCol[i] = heightCol[i + 1];  
 heightCol[i + 1] = bufer;  
 }  
 }  
 }  
 }//End of method  
  
}//End of class

Клас Window – клас, що призначений для відображення результату роботи класу Diagram для завдання №2.

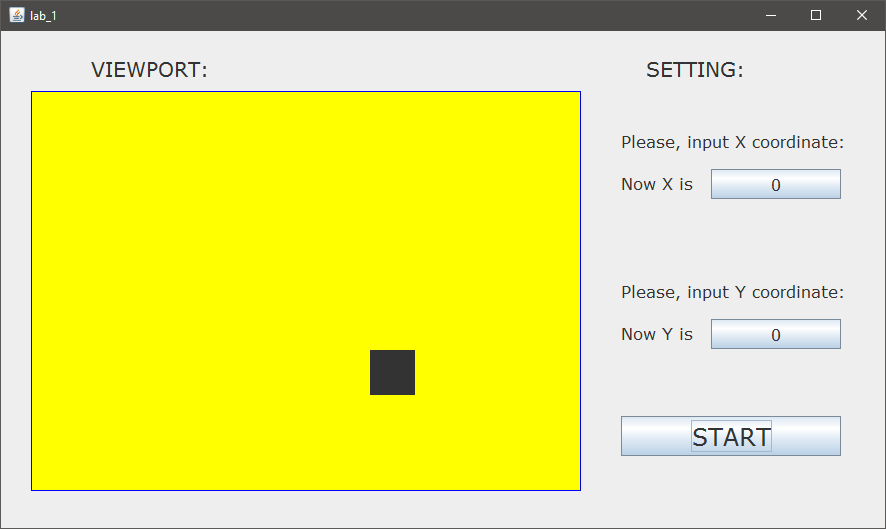
package lab1.task2;  
  
import lab1.util.InputDialog;  
  
import javax.swing.\*;  
import javax.swing.border.Border;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
  
public class Window {  
  
 int numOfCols;  
 public Window(){  
 numOfCols = 10;  
 }  
  
 public void drawWindow(){  
 JFrame window = new JFrame("lab\_1");  
 window.setLayout(null);  
 window.setSize(900,536);  
 window.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 //The util panel for moving of the cube  
 JPanel panel = new JPanel(new BorderLayout());  
 panel.setSize(550,400);  
 panel.setLocation(30,60);  
 Border lineBorder = BorderFactory.*createLineBorder*(Color.*BLUE*);  
 panel.setBorder(lineBorder);  
 panel.setOpaque(true);  
 panel.setBackground(Color.*RED*);  
  
  
  
 //Info labels  
 JLabel viewportLabel = new JLabel("VIEWPORT:");  
 viewportLabel.setFont( new Font("Verdana", Font.*PLAIN*, 20));  
 viewportLabel.setSize(viewportLabel.getPreferredSize());  
 viewportLabel.setLocation(90, 25);  
  
 JLabel settingLabel = new JLabel("SETTING:");  
 settingLabel.setFont( new Font("Verdana", Font.*PLAIN*, 20));  
 settingLabel.setSize(settingLabel.getPreferredSize());  
 settingLabel.setLocation(645, 25);  
  
 JLabel inputXLabel = new JLabel("<html>Please, input number of cols:<br><br>COLUMN </html>");  
 inputXLabel.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 inputXLabel.setSize(inputXLabel.getPreferredSize());  
 inputXLabel.setLocation(620, 100);  
  
 JLabel inputYLabel = new JLabel("<html>Sort cols by inc the height<br><br>Press here </html>");  
 inputYLabel.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 inputYLabel.setSize(inputYLabel.getPreferredSize());  
 inputYLabel.setLocation(620, 250);  
  
  
 //Creation of Buttons  
 JButton inputColButton = new JButton("" + numOfCols);  
 inputColButton.setSize(130,30);  
 inputColButton.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 inputColButton.setLocation(710,138);  
 inputColButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 InputDialog input = new InputDialog();  
 numOfCols = input.showInputDialog();  
 inputColButton.setText("" + numOfCols);  
 }  
 });  
  
  
 Diagram diagram = new Diagram(numOfCols, panel.getWidth(), panel.getHeight());  
 JButton sortButton = new JButton("SORT");  
 sortButton.setEnabled(false);  
 sortButton.setSize(130,30);  
 sortButton.setFont( new Font("Verdana", Font.*PLAIN*, 16));  
 sortButton.setLocation(710,288);  
 sortButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 diagram.sortRectungleByheight();  
 window.repaint();  
 window.validate();  
 }  
 });  
  
  
 JButton startButton = new JButton("START");  
 startButton.setSize(220,40);  
 startButton.setFont( new Font("Verdana", Font.*PLAIN*, 24));  
 startButton.setLocation(620,385);  
 startButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 sortButton.setEnabled(true);  
 startButton.setEnabled(false);  
 inputColButton.setEnabled(false);  
  
 panel.add(diagram);  
 window.repaint();  
 window.validate();  
 }  
 });  
  
  
  
 //add component to the Window  
 window.add(panel);  
  
 window.add(viewportLabel);  
 window.add(settingLabel);  
 window.add(inputXLabel);  
 window.add(inputYLabel);  
  
 window.add(inputColButton);  
 window.add(sortButton);  
 window.add(startButton);  
  
 window.setVisible(true);  
 }//End of method  
  
}//End of class

**Результат виконання програми**

Головне меню програми. Користувач обирає який функціонал має запускатися. Завдання 1 чи завдання 2.



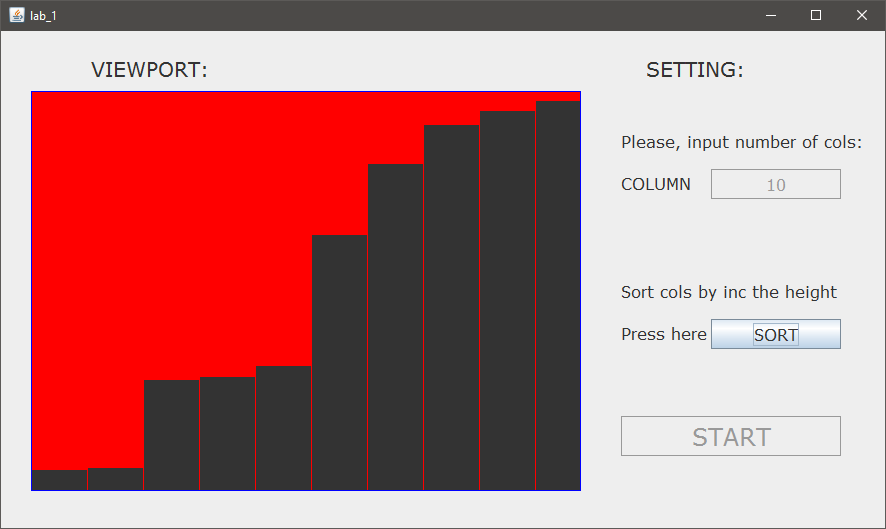
Форма відображує рух квадрату по панелі. При натисканні кнопки старт можна додавати необмежену кількість фігур.



При дотиканні до стінок панелі, вона міняє свій колір на випадковий. Всього кольорів шість.



Форма для відображення діаграм. При натисканні кнопки старт створюється діаграма. Кількість стовпців за замовчуванням 10.



При натисканні кнопки «sort» дана діаграма сортується за висотою стовпців.

**Висновок:** під час виконання лабораторної роботи, я навчився писати код більш зрозумілим і наочним. Використовуючи правила написання якісного коду, код став набагато зрозумілішим, а тому легшим у пошуку помилок і доданні нового функціоналу.