

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №14**

**з дисципліни «Об’єктно орієнтованне програмування»**

## на тему:

**«Розробка Java-програм з графічним інтерфейсом.»**

Виконав:

студент гр. ПЗ2011

Кулик С. В.

Прийняв:

Андрющенко В. О.

Дніпро, 2022

**Тема:** Розробка Java-програм з графічним інтерфейсом.

**Мета:** Отримати навички розробки Java-програм з графічним інтерфейсом.

**Завдання**

Написати об'єктно-орієнтовану програму на мові Java. Програма повинна мати графічний інтерфейс взаємодії з користувачем. В програмі повинно бути не менше трьох класів.

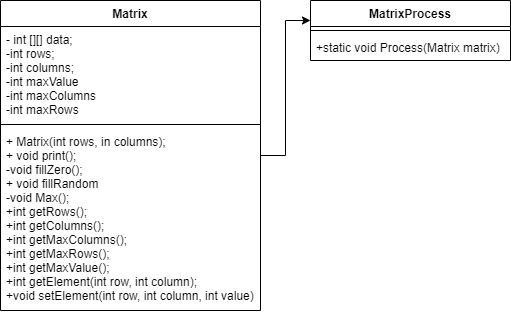
Індивідуальні завдання взяти з попередньої лабораторної роботи.

**Індивідуальне завдання**

Дана дійсна матриця. Переставляючи її рядки і стовпці, домогтися того, щоб

найбільший елемент (один з них) виявився в верхньому лівому кутку.

**Діаграма класів**

****

**Текст програми**

**Файл Main.java**

package com.company;  
  
public class Main {  
  
 public static void main(String[] args) {  
 FrameGUI app = new FrameGUI();  
 app.setVisible(true);  
 }  
}

**Файл FrameGUI.java**

package com.company;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class FrameGUI extends JFrame {  
 private Matrix matrix;  
  
 private JTextField rowsTextField = new JTextField(10);  
 private JTextField columnsTextField = new JTextField(10);  
  
  
 private JButton fillMatrixButton = new JButton("Заповнити матрицю");  
  
 private JTable matrixTable;  
 private JScrollPane matrixScroll;  
 private JButton processMatrix = new JButton("Обробка матриці");  
  
 private JPanel panel = new JPanel();  
  
 public FrameGUI() {  
 super("Робота над матрицею");  
 setSize(500, 400);  
 setLocationRelativeTo(null);  
 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 this.setResizable(false);  
  
 panel.setLayout(null);  
 this.add(panel);  
  
 JLabel matrixSize = new JLabel("Розмір матриці");  
 matrixSize.setBounds(190, 5, 100, 20);  
 panel.add(matrixSize);  
  
 JLabel rowsLabel = new JLabel("Рядків");  
 rowsLabel.setBounds(20, 40, 50, 20);  
 rowsTextField.setBounds(100, 40, 100, 20);  
 rowsTextField.setText("3");  
  
 JLabel columnsLabel = new JLabel("Стовпців");  
 columnsLabel.setBounds(240, 40, 60, 20);  
 columnsTextField.setBounds(340, 40, 100, 20);  
 columnsTextField.setText("5");  
  
  
 panel.add(rowsLabel);  
 panel.add(rowsTextField);  
 panel.add(columnsLabel);  
 panel.add(columnsTextField);  
 rowsTextField.setColumns(10);  
 columnsTextField.setColumns(10);  
  
  
  
 fillMatrixButton.setBounds(30, 330, 200, 20);  
 panel.add(fillMatrixButton);  
  
 processMatrix.setBounds(250, 330, 200, 20);  
 panel.add(processMatrix);  
  
 fillMatrixButton.addActionListener(new EventListener());  
 processMatrix.addActionListener(new EventListener());  
 }  
  
 class EventListener implements ActionListener {  
 @Override  
 public void actionPerformed(ActionEvent event) {  
 if (event.getSource() == fillMatrixButton) {  
 if (matrixTable != null) {  
 DefaultTableModel model = (DefaultTableModel) matrixTable.getModel();  
 for (int i = 0; i < model.getRowCount(); i++)  
 model.removeRow(i);  
 panel.remove(matrixTable);  
 panel.remove(matrixScroll);  
 }  
  
 //створення нової матриці  
 int rows;  
 int columns;  
 try {  
 rows = Integer.parseInt(rowsTextField.getText());  
 columns = Integer.parseInt(columnsTextField.getText());  
 if (rows < 1 || columns < 1) {  
 JOptionPane.showMessageDialog(null,  
 "Кількість рядків і стовпців матриці має бути більше 0", "Помилка!",  
 JOptionPane.ERROR\_MESSAGE);  
 return;  
 }  
 matrix = new Matrix(rows, columns);  
 } catch (Exception e) {  
 JOptionPane.showMessageDialog(null, "Ви ввели некоректні значення кількості рядків/стовпців!",  
 "Помилка!", JOptionPane.ERROR\_MESSAGE);  
 return;  
 }  
 matrix.fillRandom();  
  
 //виведення таблиці  
 matrixTable = new JTable(rows, columns);  
 matrixScroll = new JScrollPane(matrixTable);  
 matrixTable.setBounds(20, 120, 450, 200);  
 panel.add(matrixScroll);  
 panel.add(matrixTable);  
 matrixTable = createTable(matrix, matrixTable);  
 }  
 if (event.getSource() == processMatrix) {  
 if (matrixTable == null) {  
 JOptionPane.showMessageDialog(null, "Спочатку заповніть матрицю!", "Помилка!",  
 JOptionPane.ERROR\_MESSAGE);  
 return;  
 }  
 MatrixProcess.Process(matrix);  
 //оновлення таблиці  
 matrixTable = createTable(matrix, matrixTable);  
  
 //виведення результату  
  
 }  
  
 }  
  
 public JTable createTable(Matrix matrix, JTable fieldMatrix) {  
 DefaultTableModel model = (DefaultTableModel) fieldMatrix.getModel();  
 for (int i = 0; i < model.getRowCount(); i++) {  
 for (int j = 0; j < model.getColumnCount(); j++) {  
 model.setValueAt(matrix.getElement(i, j), i, j);  
 }  
 }  
 return new JTable(model);  
 }  
 }  
}

**Файл Matrix.java**

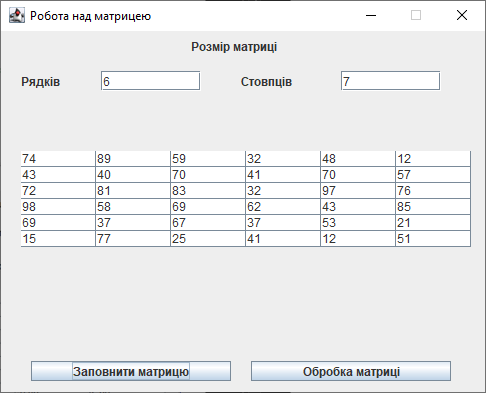
package com.company;  
  
public class Matrix  
{  
 private int[][] data;  
 private int rows;  
 private int columns;  
 private int MaxValue;  
 private int maxColumns;  
 private int maxRows;  
  
 public Matrix(int rows, int columns)  
 {  
 this.rows = rows;  
 this.columns = columns;  
 data = new int[rows][columns];  
 fillZero();  
 Max();  
 }  
   
 public void print()  
 {  
 for (int i = 0; i < rows; i++)  
 {  
 for (int j = 0; j < columns; j++)  
 {  
 System.out.print(data[i][j] + " ");  
 }  
 System.out.println();  
 }  
 }  
  
 private void fillZero()  
 {  
 for(int i = 0; i < rows; i++)  
 {  
 for(int j = 0; j < columns; j++)  
 {  
 data[i][j] = 0;  
 }  
 }  
 }  
  
 public void fillRandom()  
 {  
 for(int i = 0; i < rows; i++)  
 {  
 for(int j = 0; j < columns; j++)  
 {  
 data[i][j] = (int)(Math.random() \* 100);  
 }  
 }  
 }  
  
 private void Max()  
 {  
 int max = 0;  
 for (int i = 0; i < rows; i++)  
 {  
 for (int j = 0; j < columns; j++)  
 {  
 if (data[i][j] > max)  
 {  
 max = data[i][j];  
 maxColumns = j;  
 maxRows = i;  
 }  
 }  
 }  
 MaxValue = max;  
 }  
  
 public int getRows()  
 {  
 return rows;  
 }  
  
 public int getColumns()  
 {  
 return columns;  
 }  
  
 public int getMaxColumns()  
 {  
 Max();  
 return maxColumns;  
 }  
  
 public int getMaxRows()  
 {  
 Max();  
 return maxRows;  
 }  
  
 public int getMaxValue()  
 {  
 return MaxValue;  
 }  
 public int getElement(int row, int column)  
 {  
 return data[row][column];  
 }  
  
 public void setElement(int row, int column, int value)  
 {  
 data[row][column] = value;  
 }  
}

**Файл MatrixProcess.java**

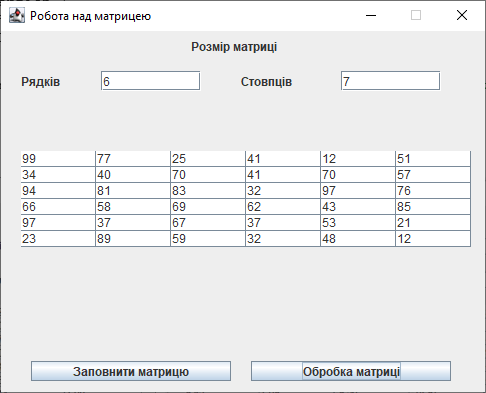
package com.company;  
  
public class MatrixProcess  
{  
 public static void Process(Matrix matrix)  
 {  
 int maxRow = matrix.getMaxRows();  
 int maxCol = matrix.getMaxColumns();  
  
 for (int i = 0; i < matrix.getColumns(); i++)  
 {  
 int temp = matrix.getElement(0, i);  
 matrix.setElement(0, i, matrix.getElement(maxRow, i));  
 matrix.setElement(maxRow, i, temp);  
  
 }  
 for (int i = 0; i < matrix.getRows(); i++)  
 {  
 int temp = matrix.getElement(i, 0);  
 matrix.setElement(i, 0, matrix.getElement(i, maxCol));  
 matrix.setElement(i, maxCol, temp);  
 }  
 }  
}

**Результати програми**

*Матриця до обробки*

****

*Матриця після обробки*



**Висновок**

Під час виконання лабораторної роботи було отримано практичні навички роботи з графічними інтерфейсами в мові програмування Джава. Було створенно об’єктно-орієнтовану програму, яка виконує поставленне завдання. Взаємодія з користувачем здійснюється за допомогою графічного інтерфейсу.

Також для виконання завдання було використано роботу з інших дисциплін, так, для створення матриці в даній роботі було лише підключено раніше створену бібліотеку з матрицями та операціями над ними.