

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

Лабораторна робота № 14

з дисципліни «Об'єктно-орієнтоване програмування»

на тему: «Розробка Java-програм з графічним інтерфейсом»

Виконав:
студент гр.ПЗ1911
Сафонов Д. Є.
Прийняла:
Демидович І.М.

Дніпро, 2021

Тема. Розробка 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. 1. Дана дійсна матриця. Переставляючи її рядки і стовпці, домогтися того, щоб найбільший елемент (один з них) виявився в верхньому лівому кутку.

Діаграма класів.

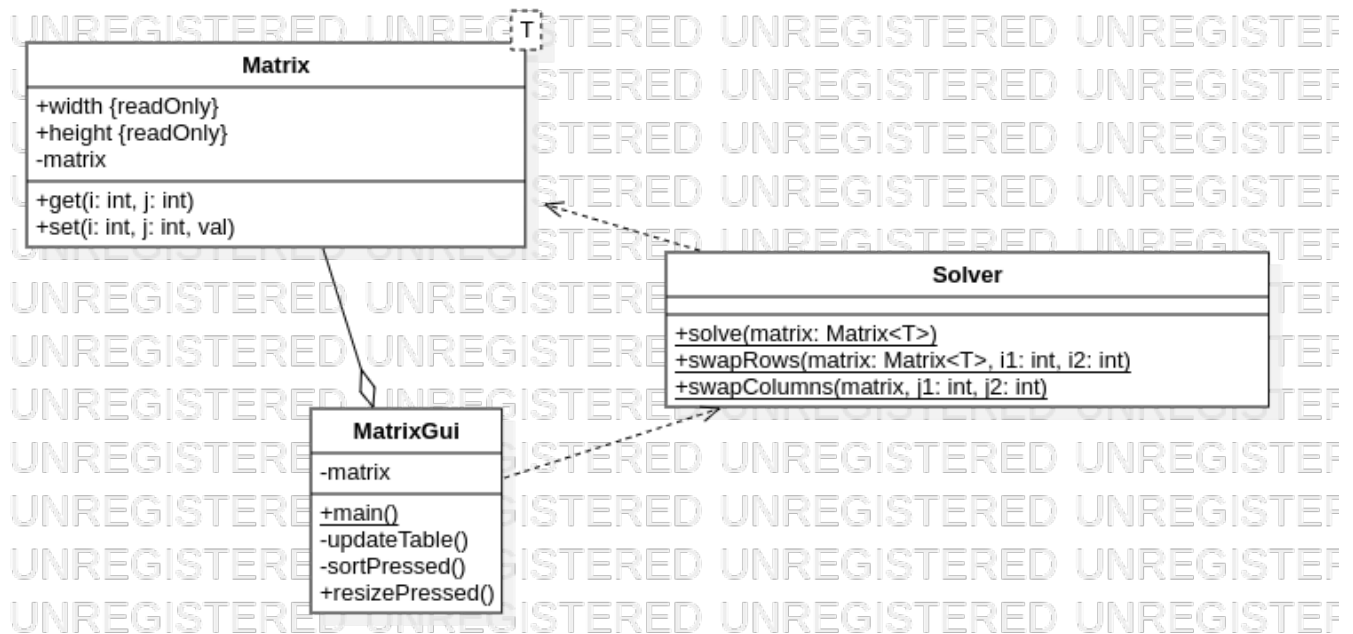


Рисунок 1

Опис елементів інтерфейсу, які використовуються в програмі.

Label - Ярлик із текстом, для спрощення навігації.

Button - Кнопка, для виклику операцій.

TextField - Поле для вводу текстової інформації.

Table - Таблиця для організації пов'язаної текстової інформації.

Текст програми.

“MatrixGui.java”

```
public class MatrixGui extends javax.swing.JFrame {
    public MatrixGui() {
        initComponents();// generated function
        //
        updateTable();
        //
    }

    private void updateTable() {
        var m = new Integer[matrix.height][matrix.width];
        for (int i = 0; i < matrix.height; i++) {
            m[i] = new Integer[matrix.width];
            for (int j = 0; j < matrix.width; j++) {
                m[i][j] = matrix.get(i, j);
            }
        }

        var names = new String[matrix.width];
        for (int j = 0; j < matrix.width; j++) {
            names[j] = String.valueOf(j);
        }
        table.setModel(new javax.swing.table.DefaultTableModel(m, names));
        table.setShowGrid(true);
        table.setTableHeader(null);
        jScrollPane1.setViewportViewView(table);
    }

    private void sortButtonActionPerformed(java.awt.event.ActionEvent evt) {
        for (int i = 0; i < matrix.height; i++) {
            for (int j = 0; j < matrix.width; j++) {
                var mij = Integer.parseInt(table
                    .getModel()
                    .getValueAt(i, j)
                    .toString());
                matrix.set(i, j, mij);
            }
        }

        var indices = com.dazzlemon.oop14.Solver.solve(matrix);
        var x = indices.x;
        var y = indices.y;
    }
}
```

```

        com.dazzlemon.oop14.Solver.swapRows(matrix, 0, x);
        com.dazzlemon.oop14.Solver.swapColumns(matrix, 0, y);

        updateTable();
    }

```

```

private void resizeButtonActionPerformed(java.awt.event.ActionEvent evt) {
    int w = Integer.parseInt(columnsText.getText());
    int h = Integer.parseInt(rowsText.getText());
    var newMatrix = new com.dazzlemon.oop14.Matrix<>(h, w, 0);
    for (int i = 0; i < h && i < matrix.height; i++) {
        for (int j = 0; j < w && j < matrix.width; j++) {
            var mij = Integer.parseInt(table
                .getModel()
                .getValueAt(i, j)
                .toString());
            newMatrix.set(i, j, mij);
        }
    }
    matrix = newMatrix;
    updateTable();
}

```

```

private com.dazzlemon.oop14.Matrix<Integer> matrix = new
com.dazzlemon.oop14.Matrix<Integer>(4, 4, 0);
}

```

“Matrix.java”

```
package com.dazzlemon.oop14;

import java.util.ArrayList;
import java.util.List;

public class Matrix <T> {
    public final int width;
    public final int height;
    private List<List<T>> matrix;

    public Matrix(int height, int width, T filler) {
        this.height = height;
        this.width = width;

        this.matrix = new ArrayList<>(height);
        for (int i = 0; i < height; i++) {
            this.matrix.add(i, new ArrayList<>(width));
            for (int j = 0; j < width; j++) {
                this.matrix.get(i).add(filler);
            }
        }
    }

    public void set(int i, int j, T val) {
        this.matrix.get(i).set(j, val);
    }

    public T get(int i, int j) {
        return this.matrix.get(i).get(j);
    }
}
```

“Solver.java”

```
package com.dazzlemon.oop14;

import java.awt.Point;

public class Solver {
    public static <T extends Comparable<T>> Point solve(Matrix<T> matrix) {
        int imax = 0;
        int jmax = 0;
        for (int i = 0; i < matrix.height; i++) {
            for (int j = 0; j < matrix.width; j++) {
                if (matrix.get(i, j).compareTo(matrix.get(imax, jmax)) > 0) {
                    imax = i;
                    jmax = j;
                }
            }
        }
        return new Point(imax, jmax);
    }

    public static <T> void swapRows(Matrix<T> matrix, int i1, int i2) {
        for (int j = 0; j < matrix.width; j++) {
            var tmp = matrix.get(i1, j);
            matrix.set(i1, j, matrix.get(i2, j));
            matrix.set(i2, j, tmp);
        }
    }

    public static <T> void swapColumns(Matrix<T> matrix, int j1, int j2) {
        for (int i = 0; i < matrix.height; i++) {
            var tmp = matrix.get(i, j1);
            matrix.set(i, j1, matrix.get(i, j2));
            matrix.set(i, j2, tmp);
        }
    }
}
```

Приклад роботи програми.

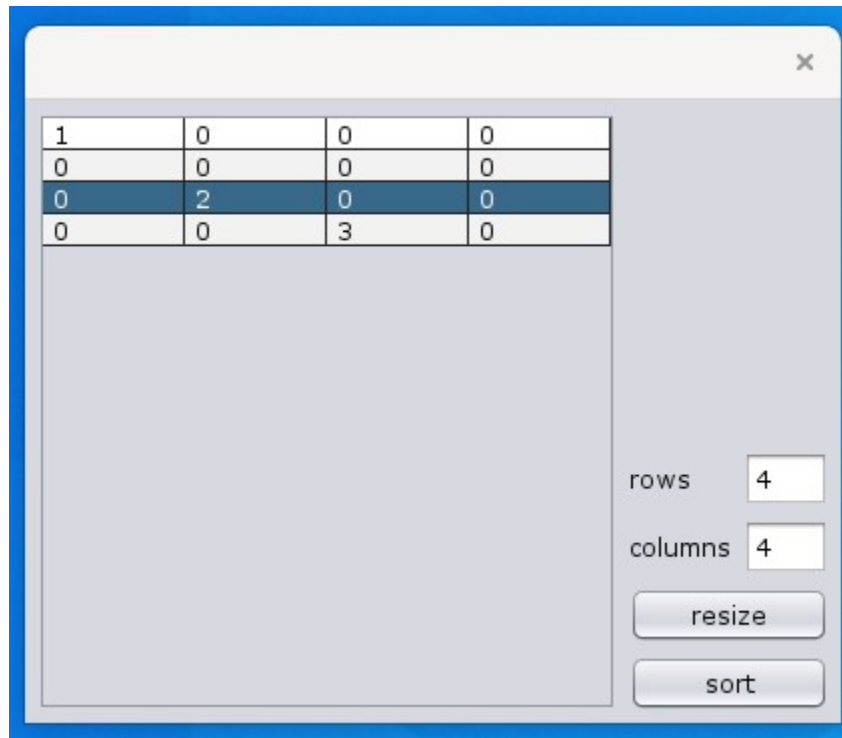


Рисунок 2(Тест 1. Підготовка)

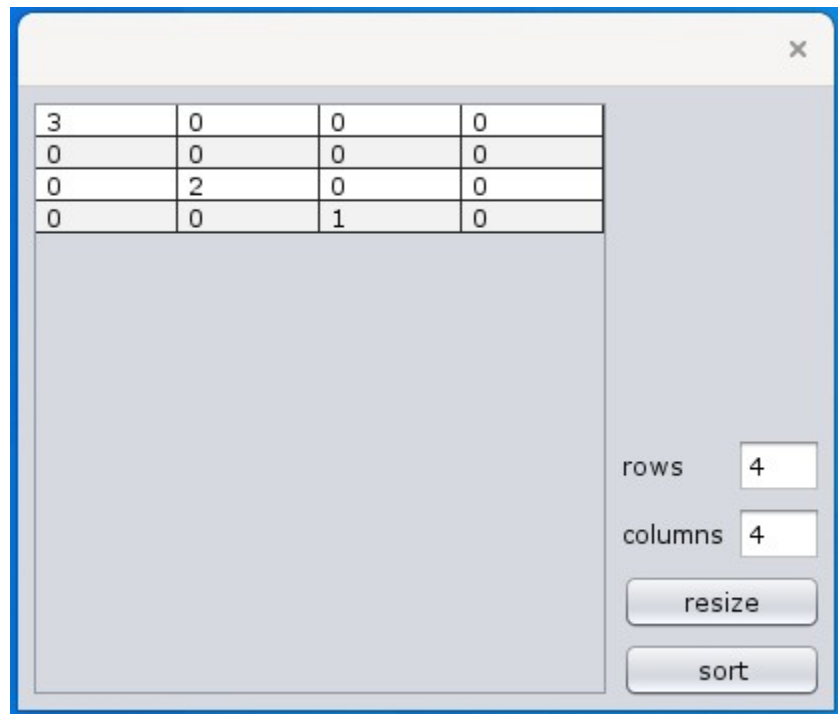


Рисунок 3(Тест 1. Стовпці та Рядки були замінені так, що найбільший елемент знаходиться за індексами 0, 0)

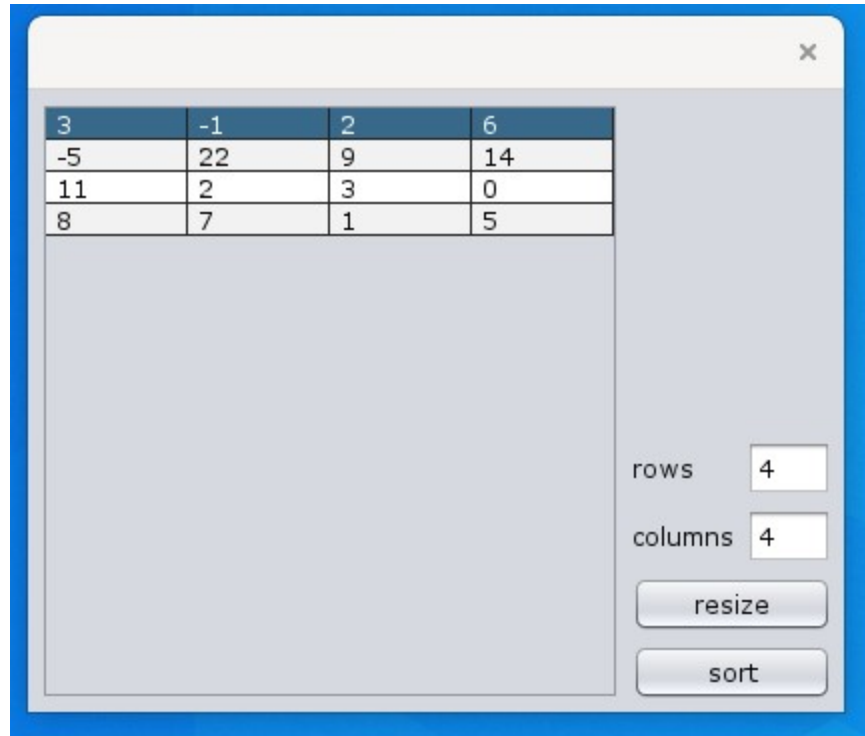


Рисунок 4(Тест 2. Підготовка)

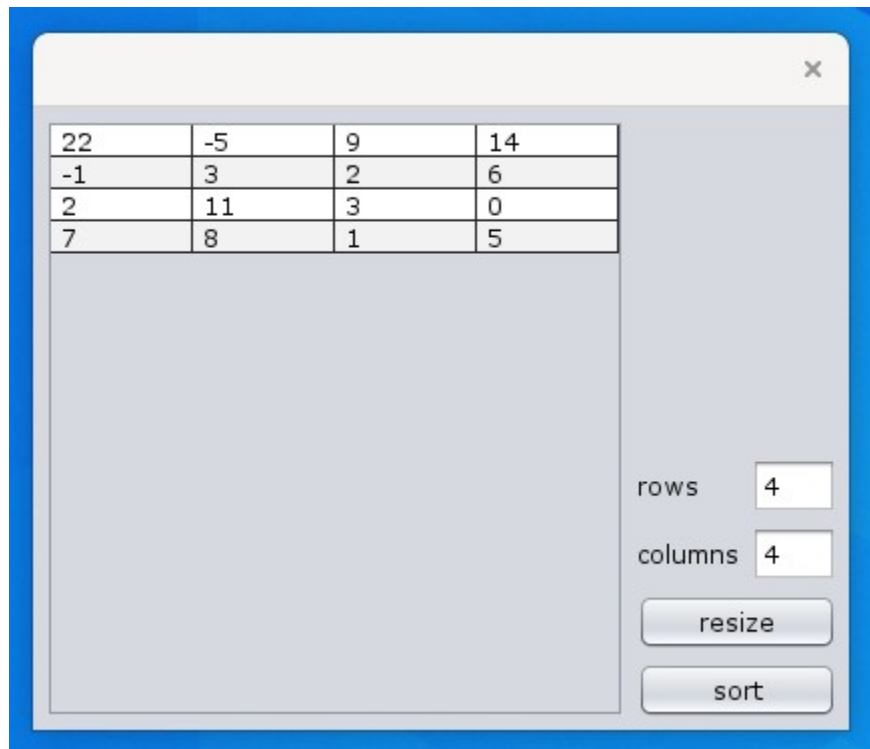


Рисунок 5(Тест 2. Найбільший елемент за індексом 0, 0)

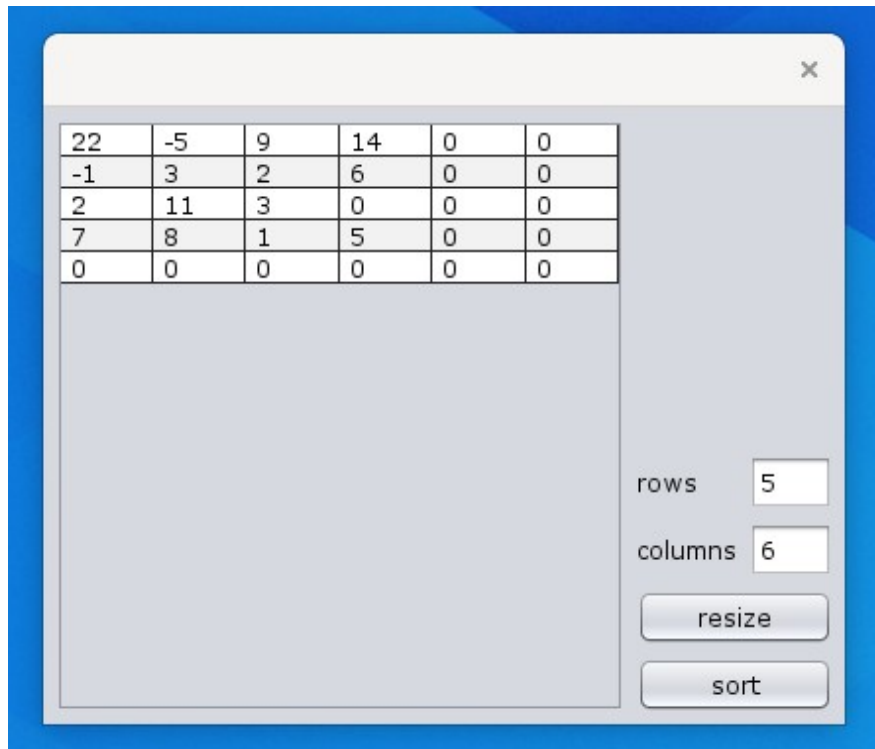


Рисунок 6(збільшення розміру)

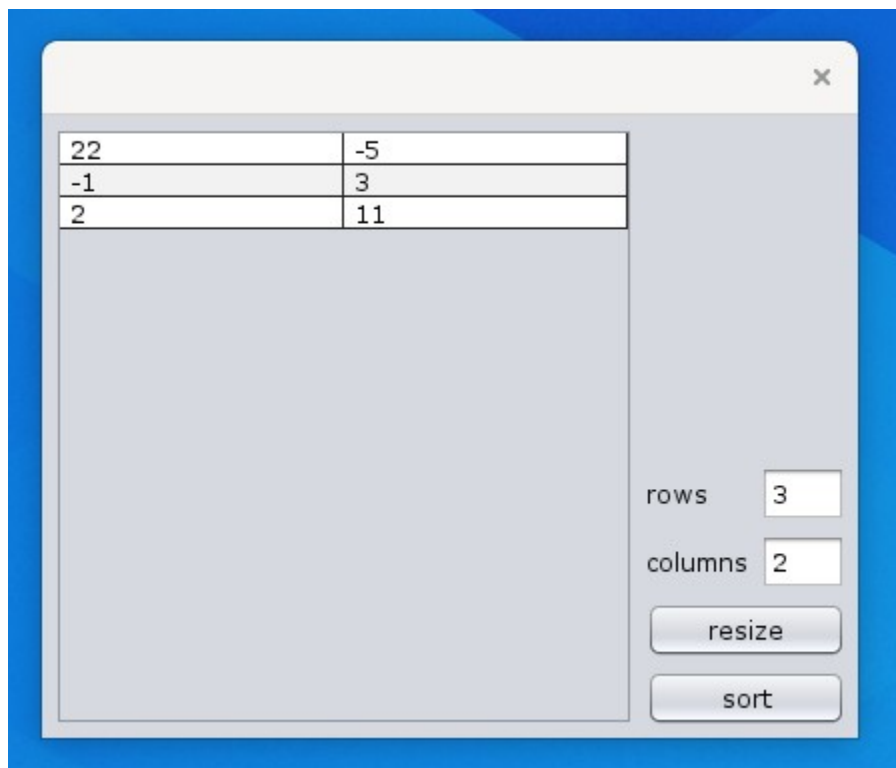


Рисунок 7(зменшення розміру)

Висновки.

Програма була розроблена на основі класів з попередньої лабораторної роботи(був змінений тільки інтерфейс взаємодії з користувачем). Для розробки була використана черета розробки Apache NetBeans. Хоча з конструктором розробляти програму із використанням Swing набагато зручніше ніж у текстовому режимі, вона програє у дуже багатьох аспектах графічним бібліотекам (наприклад Qt5), але скоріш за все більшість мінусів можна пояснити особливостями мови Java.