Додатки

**Додаток А**

**Блок-схема алгоритму розв'язку задачі**

**Рисунок А.1 – Блок-схема алгоритму**

**Додаток Б**

**Файл класу Main**

import java.io.\*;  
  
public class Main {  
 public static void main(String[] args) throws IOException {  
 Matrix a = new Matrix();  
 a.readFromFile("Data.txt");  
 a.print();  
 a.sort();  
 a.print();  
 *additionalTask*(a.getMatrix());  
 }  
  
 private static void additionalTask(int[][] matrix) throws IOException {  
 int countEvenElements = 0;  
 int sumOfEvenElements = 0;  
  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 if (matrix[i][j] % 2 == 0) {  
 countEvenElements++;  
 sumOfEvenElements += matrix[i][j];  
 }  
 }  
 }  
  
 int averageOfEvenElements = sumOfEvenElements / countEvenElements;  
  
 System.*out*.println("Середнє парних елементів: " + averageOfEvenElements);  
  
 FileWriter fileWriter = new FileWriter("Data.txt");  
 fileWriter.write("Середнє парних елементів: " + averageOfEvenElements);  
 fileWriter.close();  
 }  
}

**Файл класу Matrix**

import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
class Matrix {  
 private int[][] matrix;  
  
 int[][] getMatrix() {  
 return this.matrix;  
 }  
  
 void readFromFile(String pathname) throws FileNotFoundException {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("N = ");  
 int n = Integer.*parseInt*(scanner.nextLine());  
 System.*out*.print("M = ");  
 int m = Integer.*parseInt*(scanner.nextLine());  
  
 int[][] matrix = new int[n][m];  
  
 File file = new File(pathname);  
 Scanner fileScanner = new Scanner(file);  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < m; j++) {  
 matrix[i][j] = fileScanner.nextInt();  
 }  
 }  
 fileScanner.close();  
  
 this.matrix = matrix;  
 }  
  
 void print() {  
 for (int i = 0; i < this.matrix.length; i++) {  
 for (int j = 0; j < this.matrix[0].length; j++) {  
 System.*out*.print(this.matrix[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 }  
  
 void sort() {  
 int min = findMinElement();  
 int max = findMaxElement();  
  
 for (int i = 0; i < this.matrix.length; i++) {  
 boolean hasMinOrMax = false;  
 for (int j = 0; j < this.matrix[0].length; j++) {  
 if (this.matrix[i][j] == min || this.matrix[i][j] == max){  
 hasMinOrMax = true;  
 break;  
 }  
 }  
  
 if (hasMinOrMax) {  
 this.matrix[i] = sortRow(i);  
 }  
 }  
 }  
  
 private int[] sortRow(int rowNumber) {  
 int[] row = this.matrix[rowNumber];  
 for (int i = 1; i < row.length; i++) {  
 int temp = row[i];  
 for (int j = i; j > 0; j--) {  
 if (row[j-1] > temp) {  
 row[j] = row[j-1];  
 row[j-1] = temp;  
 } else {  
 break;  
 }  
 }  
 }  
  
 return row;  
 }  
  
 private int findMinElement() {  
 int min = this.matrix[0][0];  
 for (int i = 0; i < this.matrix.length; i++) {  
 for (int j = 0; j < this.matrix[0].length; j++) {  
 if (this.matrix[i][j] < min) min = this.matrix[i][j];  
 }  
 }  
 return min;  
 }  
  
 private int findMaxElement() {  
 int max = matrix[0][0];  
  
 for (int i = 0; i < this.matrix.length; i++) {  
 for (int j = 0; j < this.matrix[0].length; j++) {  
 if (this.matrix[i][j] > max) max = this.matrix[i][j];  
 }  
 }  
 return max;  
 }  
}