

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1**

**з дисципліни «Алгоритми та структури даних»**

**на тему: «Розробка програм методом покрокової деталізації»**

Виконав:

студент гр. ПЗ2011 Грива Я. А.

Прийняла:

Демидович І. М.

Дніпро, 2022

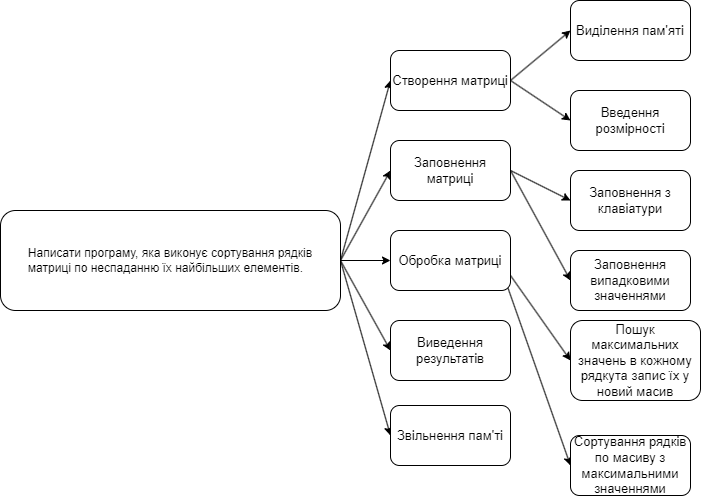
**Тема:** Розробка програм методом покрокової деталізації.

**Мета:** Ознайомитися з методами розробки алгоритмів. Набути практичних навичок розробки програм методом покрокової реалізації.

**Завдання**

7. Написати програму, яка виконує сортування рядків матриці по неспаданню їх найбільших елементів.

**Структурна схема розв’язання завдання**



**Текст програми**

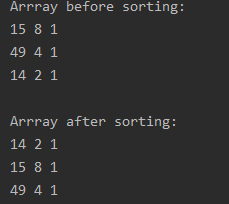
**Matrix.java**

package com.company;  
  
public class Matrix  
{  
 private int[][] data;  
 private int rows;  
 private int columns;  
 private int MaxValue;  
  
 public Matrix(int rows, int columns)  
 {  
 this.rows = rows;  
 this.columns = columns;  
 data = new int[rows][columns];  
 fillZero();  
 Max();  
 }  
  
 public Matrix(int [][] data)  
 {  
 this.data = data;  
 rows = data.length;  
 columns = data.length;  
 }  
  
 public int getMaxFloorWithMan()  
 {  
 int MaxFloor = 0;  
 int sumMan = 0;  
 int maxSumMan = getSumMan(0);  
 for (int i = 1; i < rows; i++)  
 {  
 sumMan = getSumMan(i);  
 if(sumMan > maxSumMan)  
 {  
 MaxFloor = i;  
 maxSumMan = sumMan;  
 }  
 }  
 return MaxFloor;  
 }  
  
 private int getSumMan(int row) {  
 int sum = 0;  
 for (int i = 0; i < columns; i+=2) {  
 sum += data[row][i];  
 }  
 return sum;  
 }  
  
 public int getMaxElementInString(int stringNumber)  
 {  
 int max = data[stringNumber][0];  
  
 for (int i = 0; i < data.length; i++)  
 {  
 if(max <= data[stringNumber][i])  
 max = data[stringNumber][i];  
 }  
 return max;  
 }  
  
  
 //Написати програму, яка виконує сортування рядків матриці по неспаданню їх найбільших елементів.  
 public void SortString() {  
 int[] maxMass = new int[rows];  
 int max = getMaxElementInString(0);  
 for (int i = 0; i < data.length; i++) {  
 maxMass[i] = getMaxElementInString(i);  
 }  
  
  
 for (int i = 0; i < data.length - 1; i++) {  
 for (int j = 0; j < data.length - i - 1; j++) {  
  
 if (maxMass[j] > maxMass[j+1])  
 {  
 int temp = maxMass[j];  
 maxMass[j] = maxMass[j+1];  
 maxMass[j+1] = temp;  
  
 SwapString(data, j, j+1);  
  
 }  
 }  
 }  
 System.out.println();  
 }  
  
 private void SwapString(int[][] mass, int string1, int string2)  
 {  
 for (int i = 0; i < data.length; i++) {  
 int temp = mass[string1][i];  
 mass[string1][i] = mass[string2][i];  
 mass[string2][i] = temp;  
 }  
 }  
 public int getMaxFloorWithWoman()  
 {  
 int MaxFloor = 0;  
 int sumWoman = 0;  
 int maxSumWoman = getSumWoman(0);  
 for (int i = 1; i < rows; i++)  
 {  
 sumWoman = getSumWoman(i);  
 if(sumWoman > maxSumWoman)  
 {  
 MaxFloor = i;  
 maxSumWoman = sumWoman;  
 }  
 }  
 return MaxFloor;  
 }  
  
 private int getSumWoman(int row) {  
 int sum = 0;  
 for (int i = 1; i < columns; i+=2) {  
 sum += data[row][i];  
 }  
 return sum;  
 }  
  
 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();  
 }  
 }  
  
 public void printMas(int [] mass)  
 {  
 for (int i = 0; i < mass.length; i++)  
 {  
 System.out.print(mass[i]+ " ");  
 }  
 }  
 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);  
 }  
 }  
 }  
  
 public void fillRandomRange(int from, int to)  
 {  
 for(int i = 0; i < rows; i++)  
 {  
 for(int j = 0; j < columns; j++)  
 {  
 data[i][j] = (int)(Math.random() \* (to - from) + from);  
 }  
 }  
 }  
  
 public void manualFill()  
 {  
 for(int i = 0; i < rows; i++)  
 {  
 for(int j = 0; j < columns; j++)  
 {  
 System.out.print("Введіть значення для " + i + " " + j + ": ");  
 data[i][j] = Integer.parseInt(System.console().readLine());  
 }  
 }  
 }  
  
 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];  
 }  
 }  
 }  
 MaxValue = max;  
 }  
  
 public int getRows()  
 {  
 return rows;  
 }  
  
 public int getColumns()  
 {  
 return columns;  
 }  
  
 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;  
 }  
}

**Main.java**

package com.company;  
  
public class Main {  
  
 public static void main(String[] args) {  
  
 int [][] mass = {{15, 8, 1}, {49, 4, 1}, {14, 2,1}};  
  
 Matrix matrix = new Matrix(mass);  
 System.out.println("Arrray before sorting:");  
 matrix.print();  
 matrix.SortString();  
 System.out.println("Arrray after sorting:");  
 matrix.print();  
 }  
}

**Аналіз результатів**

****

На скріншоті видно результат роботи програми. Все працює корректно.

**Висновок**

Під час виконання лабораторної роботи розробляв програму методом покрокової деталізації. Його суть полягає в розбиванні складної задачі на більш прості підзадачі(модулі). Завдяки цьому я можу коли завгодно вносити правки до коду не порушуючи правильності роботи програми. Для цього зв'зок між модулями повинен бути мінімальним.