

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 Кулик С. В.

Прийняла:

Демидович І. М.

Дніпро, 2022

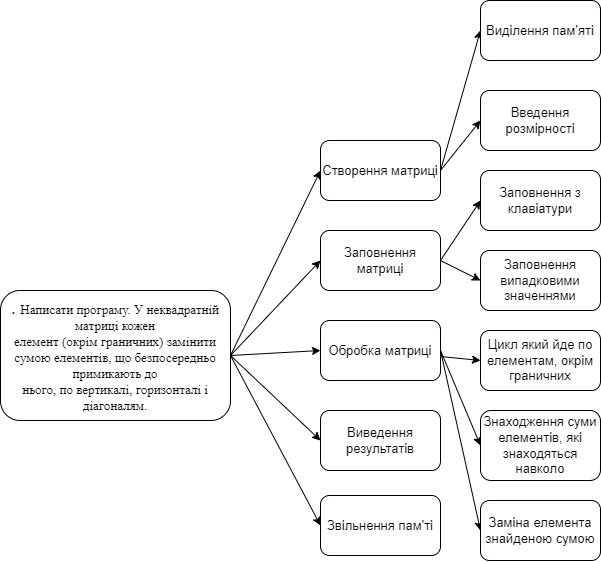
**Тема:** Розробка програм методом покрокової деталізації.

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

**Завдання**

11. Написати програму. У неквадратній матриці кожен елемент (окрім граничних) замінити сумою елементів, що безпосередньо примикають до нього, по вертикалі, горизонталі і діагоналям.

**Структурна схема розв’язання завдання**



**Текст програми**

/\*Написати програму. У неквадратній матриці кожен

елемент (окрім граничних) замінити сумою елементів,

що безпосередньо примикають до нього, по вертикалі,

горизонталі і діагоналям.\*/

#include <iostream>

using namespace std;

struct Matrix

{

int rows;

int cols;

int\*\* elements;

};

/// <summary>

/// Створення матриці

/// </summary>

/// <param name="\_m">кількість рядків</param>

/// <param name="\_n">кількість стовбців</param>

/// <returns>матриця</returns>

Matrix Create(int \_m, int \_n)

{

Matrix tmp;

tmp.rows = \_m;

tmp.cols = \_n;

tmp.elements = new int\* [tmp.rows];

for (int i = 0; i < tmp.rows; i++)

{

tmp.elements[i] = new int[tmp.cols];

}

return tmp;

}

/// <summary>

/// Видалення матриці з пам'яті

/// </summary>

/// <param name="\_matrix">посилання на матрицю</param>

void Delete(Matrix& \_matrix)

{

for (int i = 0; i < \_matrix.rows; i++)

{

delete[] \_matrix.elements[i];

}

delete[] \_matrix.elements;

}

/// <summary>

/// Заповнення матриці випадковими числами від 1 до 100

/// </summary>

/// <param name="\_matrix">матриця для заповнення</param>

void RandFilling(Matrix \_matrix)

{

for (int i = 0; i < \_matrix.rows; i++)

{

for (int j = 0; j < \_matrix.cols; j++)

{

\_matrix.elements[i][j] = rand() % 100 + 1;

}

}

}

/// <summary>

/// Заповнення матриці числами, введеними з клавіатури

/// </summary>

/// <param name="\_matrix">матриця для заповнення</param>

void ManualFilling(Matrix \_matrix)

{

for (int i = 0; i < \_matrix.rows; i++)

{

for (int j = 0; j < \_matrix.cols; j++)

{

cin >> \_matrix.elements[i][j];

}

}

}

/// <summary>

/// вивдення матриці на екран

/// </summary>

/// <param name="\_matrix">матриця для виведення</param>

void Show(Matrix \_matrix)

{

for (int i = 0; i < \_matrix.rows; i++)

{

for (int j = 0; j < \_matrix.cols; j++)

{

cout << \_matrix.elements[i][j] << "\t";

}

cout << endl;

}

}

/// <summary>

/// Обробка матриці, заміняє не граничні елементи в матриці

/// сумою елементів, які знаходяться навколо цього елемента

/// </summary>

/// <param name="\_matrix"></param>

/// <returns></returns>

Matrix\* ProcessMatrix(Matrix \_matrix)

{

//якщо кількість рядків або стовбців менше 3, вийти з функції

if (\_matrix.rows < 3 || \_matrix.cols < 3)

{

return nullptr;

}

/\*цикл який перебирає всі елементи окрім граничних\*/

for (int i = 1; i < \_matrix.rows - 1; i++)

{

for (int j = 1; j < \_matrix.cols - 1; j++)

{

//Підрахунок суми елементів, які знаходяться навколо поточного

int sum = 0;

sum += \_matrix.elements[i - 1][j - 1];

sum += \_matrix.elements[i - 1][j];

sum += \_matrix.elements[i - 1][j + 1];

sum += \_matrix.elements[i][j - 1];

sum += \_matrix.elements[i][j + 1];

sum += \_matrix.elements[i + 1][j - 1];

sum += \_matrix.elements[i + 1][j];

sum += \_matrix.elements[i + 1][j + 1];

//заміна поточного елемента сумою

\_matrix.elements[i][j] = sum;

}

}

return &\_matrix;

}

int main()

{

Matrix a = Create(4, 5);

ManualFilling(a);

cout << endl << endl;

cout << "Matrix before: " << endl << endl;

Show(a);

cout << '\n';

ProcessMatrix(a);

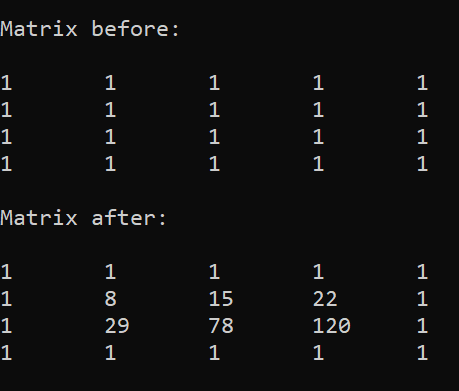
cout << "Matrix after: " << endl << endl;

Show(a);

Delete(a);

}

**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.