МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №5**

**з дисципліни «Основи програмної інженерії»**

**на тему: «Розробка та створення діалогу типу «питання-відповідь»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Куроп'ятник О. С.

Дніпро, 2020

**Лабораторна робота №3**

Тема. Розробка та створення діалогу типу «питання-відповідь».

Мета. Отримати практичні навички створення діалогу типу «питання-відповідь».

**Постановка задачі**

1.Загальна постановка:

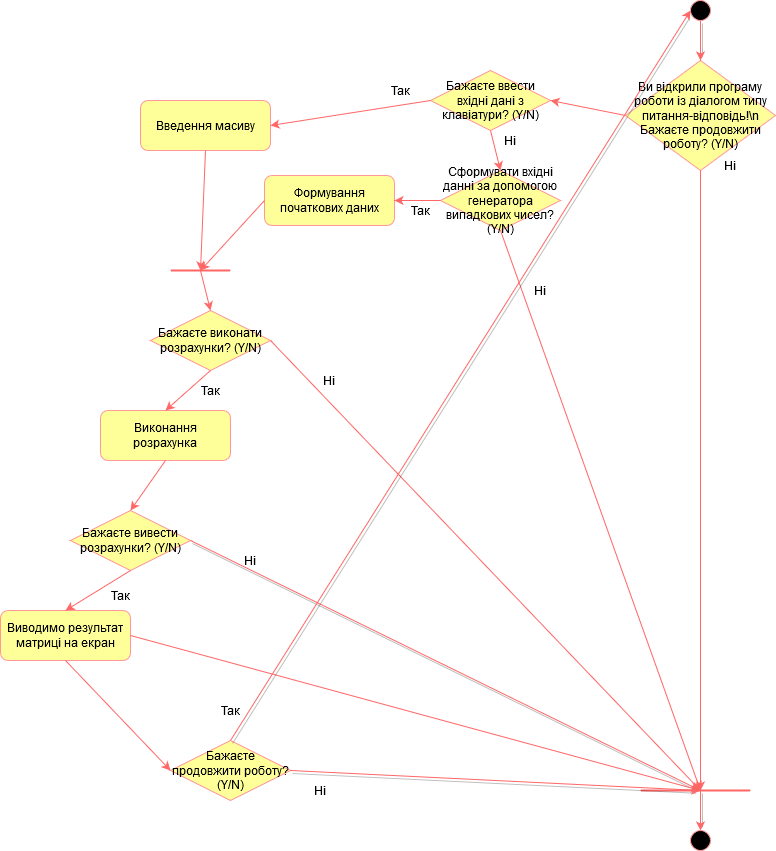
-побудуйте сценарій ведення діалог;

**-**розробіть алгоритм та напишіть програму з діалогом типу «питання-відповідь» згідно розробленого сценарію.

2.Індивідуальна постановка:

18. Якщо максимум серед елементів, що розташовані вище бічної діагоналі квадратної матриці А, менше за мінімум серед елементів, що розташовані нижче головної діагоналі, то транспонуйте матрицю, інакше знайдіть А2.

**Сценарій діалогу програми**



**Текст програми**

Файл source.cpp

#include <iostream>

#include <conio.h>

#include "Header.h"

#include "Windows.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int min, max;

int \*\*Array;

int \*\*tmparr;

int size;

char choice, choiceExit;

do {

std::system("cls");

std::cout << "Ви відкрили програму роботи із діалогом типу питання-відповідь!\n Бажаєте продовжити роботу? (Y/N)" << std::endl;

choiceExit = \_getch();

if ((choiceExit == 'Y') || (choiceExit == 'y')) {

std::system("cls");

std::cout << "Бажаєте ввести вхідні дані з клавіатури? (Y/N)" << std::endl;

choice = \_getch();

if ((choice == 'Y') || (choice == 'y')) {

std::system("cls");

std::cout << "Введіть розмір матриці: ";

size = input\_size();

int n = 1;

Array = New2DArray(size);

for (int i = 0; i < size; i++)//заповнення масива нулями

for (int j = 0; j < size; j++)

Array[i][j] = 0;

for (int m = 0; m < size; m++) {//заповнення масива з клавіатури

for (int n = 0; n < size; n++) {

std::system("cls");

PrintArray(Array, size);

std::cout << "arr[" << m << "][" << n << "] = ";

std::cin >> Array[m][n];

}

}

std::system("cls");

std::cout << "Бажаєте виконати розрахунки? (Y/N)" << std::endl;

choice = \_getch();

if ((choice == 'Y') || (choice == 'y')) {

max = FindingMax(Array, size);

min = FindingMin(Array, size);

if (max < min)

tmparr = NewTransposeArray(Array, size);

else

SquareArray(Array, size);

std::system("cls");

std::cout << "Бажаєте вивести результати? (Y/N)" << std::endl;

choice = \_getch();

if ((choice == 'Y') || (choice == 'y')) {

std::system("cls");

if (max < min) {

std::cout << "Так як максимальне число меньше мінімального, то ми транспонуємо матрицю" << std::endl;

PrintArray(tmparr, size);

Delete2DArray(tmparr, size);

}

else {

std::cout << "Так як максимальне число більше мінімального, то ми матрицю підносимо у квадрат" << std::endl;

std::cout << "Матриця в квадраті:" << std::endl;

PrintArray(Array, size);

Delete2DArray(Array, size);

}

system("Pause");

std::cout << "Бажаєте продовжити роботу? (Y/N)" << std::endl;

choiceExit = \_getch();

}

}

}

else {

std::system("cls");

std::cout << "Бажаєте сформувати вхідні дані за допомогою генератора випадкових чисел? (Y/N)" << std::endl;

choice = \_getch();

if ((choice == 'Y') || (choice == 'y')) {

std::system("cls");

std::cout << "Введіть розмір матриці: ";

size = input\_size();

Array = New2DArray(size);

FillingArray(Array, size);//заповнення масива випадковими числами

std::system("cls");

std::cout << "Бажаєте виконати розрахунки? (Y/N)" << std::endl;

choice = \_getch();

if ((choice == 'Y') || (choice == 'y')) {

max = FindingMax(Array, size);

min = FindingMin(Array, size);

if (max < min)

tmparr = NewTransposeArray(Array, size);

else

SquareArray(Array, size);

std::system("cls");

std::cout << "Бажаєте вивести результати? (Y/N)" << std::endl;

choice = \_getch();

if ((choice == 'Y') || (choice == 'y')) {

std::system("cls");

if (max < min) {

std::cout << "Так як максимальне число меньше мінімального, то ми транспонуємо матрицю" << std::endl;

PrintArray(tmparr, size);

Delete2DArray(tmparr, size);

}

else {

std::cout << "Так як максимальне число більше мінімального, то ми матрицю підносимо у квадрат" << std::endl;

std::cout << "Матриця в квадраті:" << std::endl;

PrintArray(Array, size);

Delete2DArray(Array, size);

}

system("Pause");

std::cout << "Бажаєте продовжити роботу? (Y/N)" << std::endl;

choiceExit = \_getch();

}

}

}

}

}

} while ((choiceExit != 'N') && (choiceExit != 'n'));

return 0;

}Файл function.h

#ifndef \_\_HEADER\_H\_\_

#define \_\_HEADER\_H\_\_

int input\_size();

int\*\* New2DArray(const int size);

void Delete2DArray(int\*\* Arr, const int size);

void PrintArray(int \*\*arr, const int size);

void FillingArray(int \*\*arr, const int size);

int FindingMax(int \*\*arr, const int size);

int FindingMin(int \*\*arr, const int size);

int\*\* NewTransposeArray(int \*\*arr, const int size);

void SquareArray(int \*\*arr, const int size);

#endif

Файл function.cpp

#include <iostream>

#include <random>

#include "Header.h"

int input\_size() {

system("cls");

int size;

while (true) {

std::cout << "Введіть розмір матриці: ";

std::cin >> size;

if ((std::cin.peek() != '\n') || !(size > 1)) {

std::cin.clear();

std::cin.ignore(32767, '\n');

std::cout << "На жаль, цей введення неправильне. Розмір масива повинен бути більше за 1. Повторіть знову." << std::endl;

}

else

return size;

}

}

int\*\* New2DArray(const int size) {//створення масива

int\*\* Arr = new int\*[size];

for (int i = 0; i < size; ++i)

Arr[i] = new int[size];

return Arr;

}

void Delete2DArray(int\*\* Arr, const int size) {//видилення масива

for (int i = 0; i < size; ++i)

delete[] Arr[i];

delete[] Arr;

}

void PrintArray(int\*\* arr, const int size) {//друк масива

for (int i = 0; i < size; i++){

for (int j = 0; j < size; j++) {

std::cout << arr[i][j] << "\t";

}

std::cout << std::endl;

}

}

void FillingArray(int \*\*arr, const int size) {//заповнення масива випадковими числами

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<int> dis(0, 10);

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

arr[i][j] = dis(gen);

}

int FindingMax(int \*\*arr, const int size) {//знаходження максимального числа

int max, m = 1;

max = arr[0][0];

for (int i = 0; i < size - 1; i++, m++)

for (int j = 0; j < size - m; j++)

if (arr[i][j] > max)

max = arr[i][j];

return max;

}

int FindingMin(int \*\*arr, const int size) {//знаходження мінімального числа

int min, m = 1;

min = arr[1][0];

for (int i = 1; i < size - 1; i++, m++)

for (int j = 0; j < m; j++)

if (arr[i][j] < min)

min = arr[i][j];

return min;

}

int\*\* NewTransposeArray(int \*\*arr, const int size) {//транспортування матриці

int\*\* res = New2DArray(size);

for (int i = 0; i < size; ++i)

for (int j = 0; j < size; ++j)

res[j][i] = arr[i][j];

return res;

}

void SquareArray(int \*\*arr, const int size) {//матриця в квадраті

int \*\*temp = New2DArray(size);

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

temp[i][j] = 0;

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

for (int k = 0; k < size; k++)

temp[i][j] += arr[i][k] \* arr[k][j];

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

arr[i][j] = temp[i][j];

temp[i][j] = 0;

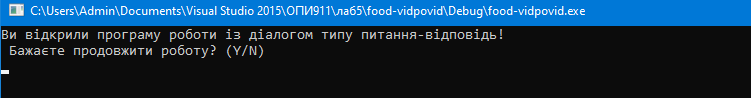
}

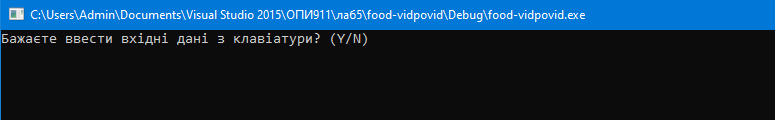
}

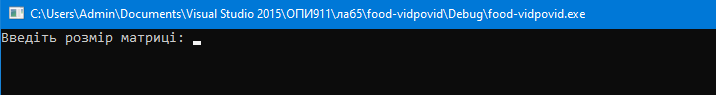
}

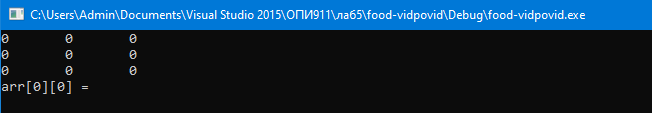
**Результати виповнення програми**

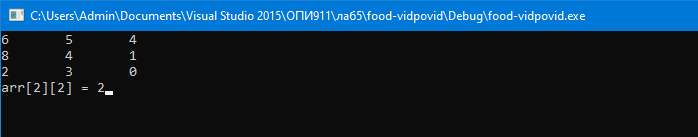
Прохід по версії програми з введенням діних з клавіатури

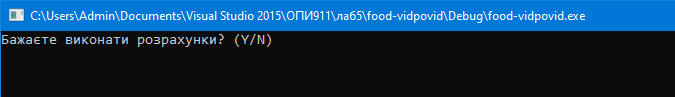


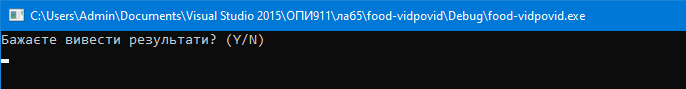


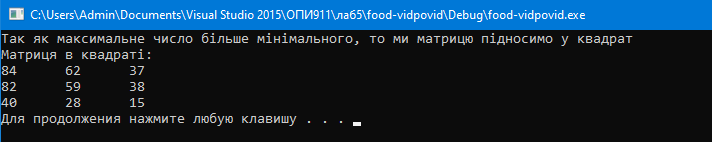


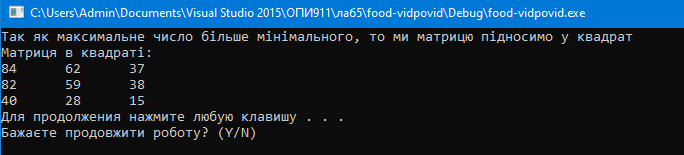






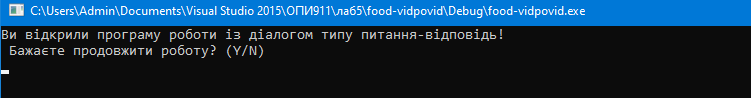


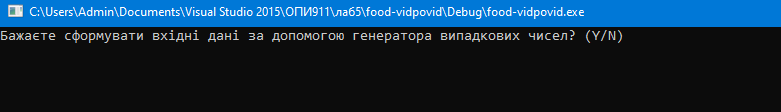


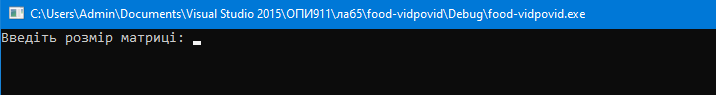


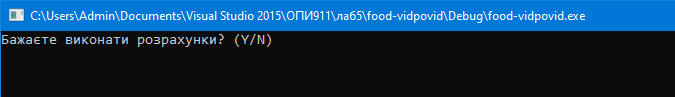
При виборі негативній відповіді вам буде видано таке питання

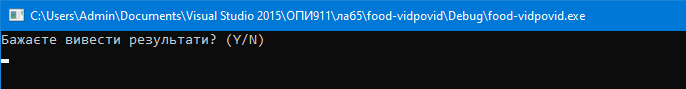
Прохід по версії програми з згенерованими даними

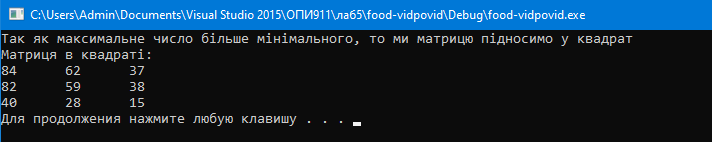


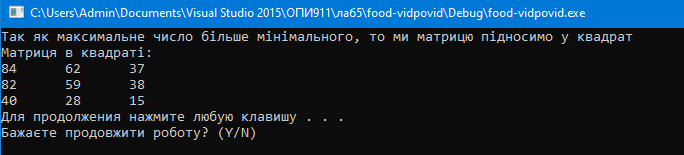












**Висновок**: з появою графічного інтерфейсу структура Q&a дещо застаріла, проте, у неї є певні переваги. Ця структура може задовольнити вимоги різних користувачів і типів даних. Зокрема, така структура особливо доречна при реалізації діалогу з безліччю «відгалужень», тобто в тих випадках, коли на кожне питання передбачається велике число відповідей, кожний з яких впливає на те, яке питання буде поставлено наступним. З цієї причини структура Q&a часто використовується в експертних системах.