МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №4**

**з дисципліни «Основи програмної інженерії»**

**на тему: «Розробка програми з меню»**

Виконав: студент гр. ПЗ2011

Горбань М.О.

Прийняла: к.т.н., доцент кафедри КІТ: Горбова О.В.

Дніпро, 2021

**Тема.** Розробка програми з меню.

**Мета.** Ознайомитися з поняттям сценарію та структури діалогу. Набути практичних навичок розробки програм зі структурою діалогу «меню».

**Завдання**

Розробіть програму для вирішення задачі з індивідуального завдання лабораторної роботи № 1, використавши структуру діалогу «меню». Перед розробкою виконайте проектування меню, аналогічно до прикладу,представленого у теоретичних відомостях.

Вимоги до написання програми:

– матриця представлена у вигляді структури (struct);

– кожна принципово нова дія представлена у вигляді функції. Обов’язковий набір функцій:

* main – головна функція програми;
* create – створення матриці (зазначення розмірності, виділення пам’яті);
* delete – очищення пам’яті, яку займає матриця;
* processing – обробка матриці, не повинна містити операторів введення та/або виведення;
* manualFilling, randomFilling, readFile – заповнення матриці з клавіатури, генератором випадкових чисел та з файлу відповідно;
* show – виведення матриці на екран;

– функція/ї обробки матриці містить коментарі, виходячи з методу покрокової деталізації;

– усі функції мають коротку та змістовну специфікацію, усі ідентифікатори (назви змінних, функцій, структур тощо) відображають суть названої програмної сутності;

– використовується лише одна матриця, використання додаткових матриць (масивів) заборонено;

– наявність меню у вигляді списку об'єктів, які обирають прямим зазначенням номера. Можливість багаторазового виклику усіх пунктів меню у довільному порядку. Можливість вибору способу заповнення матриці (з клавіатури, гератором випадкових чисел, з файлу). При успішному заповненні генератором випадкових чисел та з файлу виводиться повідомлення про успішне виконання операції;

– для організації роботи меню використовується цикл з постумовою та оператор множинного вибору;

– не використовуються рекурсивні функції та оператор goto;

– відсутність витоку пам’яті;

– при заповненні матриці з файлу, ім’я файлу вказується на етапі виконання.

При відсутності вказаного файлу виводиться повідомлення для користувача;

– файл, з якого заповнюється матриця, містить інформацію щодо її розмірності та значення елементів;

– перевірка коректності введення номера пункту меню, розмірності матриці (з точки зору належності області визначення).

Індивідуальне завдання:

6. В рядках, в яких сума першого і останнього елементів перевищує значення діагонального елемента, поміняти місцями пари елементів (перший з другим, третій з четвертим і так далі).

**1) Проект меню**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пункт (рівень 1) | № | Пункт (рівень 2) | Пункти, які треба виконати до |
| 1 | Створити матрицю | 1.1 | Заповнити випадковим чином | - |
| 1.2 | Заповнити з клавіатури |
| 1.3 | Заповнити з файлу |
| 1.4 | Повернутися до головного меню |
| 1.5 | Відчистка консолі |
| 2 | Обробка матриці |  |  | 1 |
| 3 | Вивести матрицю | 3.1 | Вивести матрицю | 1 |
| 3.2 | Записати у файл |
| 3.3 | Повернутится до головного меню |
| 3.4 | Відчистка консолі |
| 4 | Вихід із програми |  |  | - |
| 5 | Відчистка консолі |  | - | - |

**2) Текст програми**

Файл Source.cpp

#include <iostream>

#include<Windows.h>

#include "func.h"

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Matrix array = { 0, 0, nullptr };

while (true)

{

outputMenu();

int choice;

cout << "Введіть цифру необхідного пункту меню!" << endl;

cin >> choice;

switch (choice) //головне меню

{

case 1: //створити матрицю

{

bool exit = true;

while (exit)

{

outputCreateLevel2();

cin >> choice;

switch (choice) //меню створення матриці 2-й ур

{

case 1: //Заповнення генератором випадкових чисел

{

input(array);

create(array);

randomFilling(array);

break;

}

case 2: //Заповнення вручну

{

input(array);

create(array);

manualFilling(array);

break;

}

case 3: //Заповнення вручну із файлу

{

fileFilling(array);

break;

}

case 4: //вихід в головне меню

{

exit = false;

break;

}

case 5: //відчистка консолі

{

system("cls");

break;

}

default:

{

cout << "Такого пункту меню немає, спробуйте ще раз!" << endl;

}

}

}

break;

}

case 2: //обработка матрицы

{

if (array.arr == nullptr)

{

cout << "Матриця ще не створена!" << endl;

}

else

{

processing(array);

cout << "Матриця успішно оброблена!" << endl;

}

break;

}

case 3: //вивід матриці

{

bool exit = true;

while (exit)

{

outputMatrixLevel2();

cin >> choice;

switch (choice) //2-й рівень меню друку матриці.

{

case 1: //вивести матрицю в консоль

{

if (array.arr == nullptr)

{

cout << "Матриця ще не створена!" << endl;

}

show(array);

break;

}

case 2: //записати матрицю в файл

{

if (array.arr == nullptr)

{

cout << "Матриця ще не створена!" << endl;

}

fileInput(array);

break;

}

case 3: //повернутися до головного меню

{

exit = false;

break;

}

case 4: //відчистити консоль

{

system("cls");

break;

}

default:

{

cout << "Такого пункту меню немає, спробуйте ще раз!" << endl;

}

}

}

break;

}

case 4: //вихід із програми

{

delete\_matrix(array);

exit(1);

break;

}

case 5: //відчистка консолі

{

system("cls");

break;

}

default:

{

cout << "Такого пункту меню немає, спробуйте ще раз!" << endl;

}

}

}

}Файл func.cpp

#include "func.h"

#include <iostream>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

//головне меню

void outputMenu()

{

cout << "================ МЕНЮ =================" << endl;

cout << "1. Створити матрицю." << endl;

cout << "2. Обробка матриці." << endl;

cout << "3. Вивести матрицу." << endl;

cout << "4. Вихід." << endl;

cout << "5. Відчистити консоль!" << endl;

cout << "=======================================" << endl;

}

//Ввід розмірності матриці

void input(Matrix& array)

{

cout << "Введіть кількість рядків: "; cin >> array.row;

cout << "Введіть кількість стовпців: "; cin >> array.column;

}

//Виділення пам'яті під матрицю

void create(Matrix& array)

{

array.arr = new int\* [array.row];

for (int i = 0; i < array.row; i++)

{

array.arr[i] = new int[array.column];

}

}

//звільнення пам'яті

void delete\_matrix(Matrix& array)

{

for (int i = 0; i < array.row; i++)

{

delete[] array.arr[i];

}

delete[] array.arr;

array.arr == nullptr;

}

//Заповнення вручну

void manualFilling(Matrix& array)

{

for (int i = 0; i < array.row; i++)

{

for (int j = 0; j < array.column; j++)

{

cout << "array" << '[' << i << ']' << '[' << j << "]= ";

cin >> array.arr[i][j];

cout << endl;

}

}

}

//Заповнення генератором випадкових чисел

void randomFilling(Matrix& array)

{

srand(time(NULL));

for (int i = 0; i < array.row; i++)

{

for (int j = 0; j < array.column; j++)

{

array.arr[i][j] = rand() % 20;

}

}

cout << "Матриця успішно заповнена." << endl;

}

//друк матриці

void show(Matrix& array)

{

for (int i = 0; i < array.row; i++)

{

for (int j = 0; j < array.column; j++)

{

cout << setw(4) << array.arr[i][j] << ' ';

}

cout << endl << endl;

}

}

// порівняння сум першого і останнього елемента з діагональним

bool calculateSum(Matrix& array, int row)

{

int sum = array.arr[row][0] + array.arr[row][array.column - 1];

if (sum > array.arr[row][row])

return 1;

else return 0;

}

//обробка матриці

void processing(Matrix& array)

{

for (int i = 0; i < array.row; i++)

{

bool swaping = calculateSum(array, i);

if (swaping)

{

for (int j = 0; j < array.column - 1; j += 2)

{

swap(array.arr[i][j], array.arr[i][j + 1]);

}

}

}

}

//2-й рівень меню заповнення матриці

void outputCreateLevel2()

{

cout << "=============== МЕНЮ ================" << endl;

cout << "1. Заповнити випадковими числами." << endl;

cout << "2. Заповнити вручну." << endl;

cout << "3. Заповнити з файла." << endl;

cout << "4. Повернутися до головного меню." << endl;

cout << "5. Відчистити консоль." << endl;

cout << "=====================================" << endl;

}

//2-й рівень меню друку матриці

void outputMatrixLevel2()

{

cout << "=============== МЕНЮ ================" << endl;

cout << "1. Вивести мавтрицю в консоль." << endl;

cout << "2. Записати матрицю в файл." << endl;

cout << "3. Повернутися до головного меню." << endl;

cout << "4. Відчистити консоль." << endl;

cout << "=====================================" << endl;

}

//заповнення матриці із файлу

void fileFilling(Matrix& array)

{

char string[50];

cout << "Введіть шлях до файлу з матрицею." << endl;

cin >> string;

fstream f(string);

if (f.fail())

{

cout << "Помилка відкриття файлу!" << endl;

}

f >> array.row;

f >> array.column;

if (array.arr != nullptr)

{

delete\_matrix(array);

}

array.arr = new int\* [array.row];

for (int i = 0; i < array.row; i++)

{

array.arr[i] = new int[array.column];

}

for (int i = 0; i < array.row; i++)

{

for (int j = 0; j < array.column; j++)

{

f >> array.arr[i][j];

}

}

f.close();

}

//запис матриці в файл

void fileInput(Matrix& array)

{

char string[50];

cout << "Введіть шлях до файлу для запису матриці матриці." << endl;

cin >> string;

fstream f(string, ios::in | ios::out | ios::app);

if (f.fail())

{

cout << "Помилка відкриття файлу!" << endl;

}

f << array.row;

f << array.column;

for (int i = 0; i < array.row; i++)

{

f << endl;

for (int j = 0; j < array.column; j++)

{

f << array.arr[i][j] << '\t';

}

}

f.close();

}

Файл func.h

#pragma once

struct Matrix

{

int row;

int column;

int\*\* arr;

};

void outputMenu();

void outputCreateLevel2();

void outputMatrixLevel2();

void fileFilling(Matrix& array);

void fileInput(Matrix& array);

void create(Matrix& array);

void delete\_matrix(Matrix& array);

void manualFilling(Matrix& array);

void randomFilling(Matrix& array);

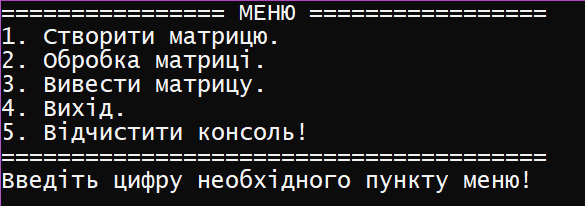
void show(Matrix& array);

void input(Matrix& array);

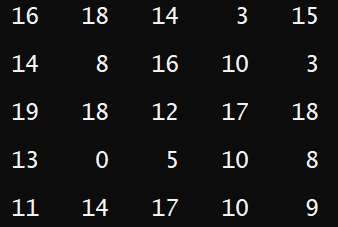
void processing(Matrix& array);

**3)Аналіз результатів**

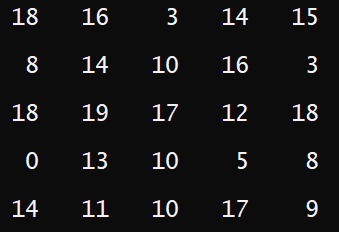
Меню



Матриця до:



Матриця після:



**4) Висновок**

При виконанні лабораторної роботи проектував меню користувача. За допомогою меню, користувач може зручно користуватись програмою. Меню може бути однорівневим та багаторівневим і кожен пункт може мати свої підфункції. Це робе програму більш зручною у користуванні.