МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1**

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

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

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Куроп’ятник О. С.

Дніпро, 2020

**Лабораторна робота №1**

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

**Постановка задачі**

1.Індивідуальна постановка: 7. У таблиці містяться оцінки студентів за результатами сесії. Кожен рядок відповідає успішності студента, кожен стовпець – окремій дисципліні. Знайти 40% студентів, які будуть отримувати стипендію. Стипендію отримують студенти з найвищим середнім балом, які не мають незадовільних оцінок.



**Текст програми**

matrix.h

#ifndef \_\_MATRIX\_H\_\_

#define \_\_MATRIX\_H\_\_

#include <random>

#include <iostream>

struct matrix {

int x;

int rows;

int cols;

int\*\* arr;

};

matrix Create(int Rows, int Cols);//створення структури

void Delete(matrix& \_m);//видалення матриці

void RandFilling(matrix \_m);//заповнення рандомно

void ManualFilling(matrix \_m);//заповнення вручну

void Show(matrix \_m);//вивід матриці

void ShowMean(int\* \_m, int x);//вивід студениів на степендію

int\* Selection(matrix &\_m);//виповнення завдання

#endif

matrix.cpp

#include "matrix.h"

//Вхідні дані: розмір матриці

//Вихідні дані: структура

matrix Create(int Rows, int Cols) {

matrix tmp;

tmp.rows = Rows;

tmp.cols = Cols;

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

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

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

return tmp;

}

//Вхідні дані: структура

//Вихідні дані: видалення

void Delete(matrix & \_m) {

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

delete[] \_m.arr[i];

delete[] \_m.arr;

}

//Вхідні дані: структура

//Вихідні дані: заповнення матриці

void RandFilling(matrix \_m) {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<int> dis(1, 5);

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

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

\_m.arr[i][j] = dis(gen);

}

//Вхідні дані: структура

//Вихідні дані: заповнення матриці

void ManualFilling(matrix \_m) {

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

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

\_m.arr[i][j] = 0;

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

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

Show(\_m);

std::cin >> \_m.arr[i][j];

system("cls");

}

}

//Вхідні дані: структура

//Вихідні дані: заповнена матриця

void Show(matrix \_m) {

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

std::cout << i + 1 << "ст.\t";

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

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

std::cout << std::endl;

}

}

//Вхідні дані: структура

//Вихідні дані: заповнена матриця

void ShowMean(int \*\_m, int x) {

std::cout << std::endl << "Степендію отримають: " << std::endl;

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

std::cout << "Студент номер " << \_m[i] << std::endl;

}

//Вхідні дані: структура

//Вихідні дані: номери студентів

int\* Selection(matrix &\_m) {

bool check = true;

int tmp = 0;

int SIZE = 0; //розмір масива середнього значення

for (int i = 0; i < \_m.rows; check = true, i++) {

for (int j = 0; j < \_m.cols && check == true; j++) {

if (\_m.arr[i][j] < 3)

check = false;

}

if (check == true)

SIZE++;

}

double \*mean = new double[SIZE];//середне значення

int \*number = new int[SIZE];//номер студента

int k = 0;

for (int i = 0; i < \_m.rows; tmp = 0, check = true, i++) {

for (int j = 0; j < \_m.cols && check == true; j++) {

if (\_m.arr[i][j] > 2)

tmp += \_m.arr[i][j];

else

check = false;

}

if (check == true) {

mean[k] = (double)tmp / \_m.cols;//середній бал

number[k] = i + 1;//запамятовуємо номер студента

k++;

}

}

/\*std::cout << "До Відсортований" << std::endl;

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

std::cout << "Студент номер " << number[i] << " із середнім балом: " << mean[i] << std::endl;\*/

int temp, item;//для масиму mean

int temp\_, item\_;//для масиву number

for (int counter = 1; counter < SIZE; counter++) {

temp\_ = number[counter];

item\_ = counter - 1; // запоминаем индекс предыдущего элемента массива

temp = mean[counter]; // инициализируем временную переменную текущим значением элемента массива

item = counter - 1; // запоминаем индекс предыдущего элемента массива

while (item >= 0 && mean[item] < temp) {// пока индекс не равен 0 и предыдущий элемент массива больше текущего

mean[item + 1] = mean[item];

mean[item] = temp;

number[item + 1] = number[item];

number[item] = temp\_;

item--;

}

}

/\*std::cout << "Відсортований" << std::endl;

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

std::cout << "Студент номер " << number[i] << " із середнім балом: " << mean[i] << std::endl;\*/

\_m.x = (SIZE \* 40) / 100;

/\*std::cout << "x = " << \_m.x << std::endl;

std::cout << "Степендію отримають: " << std::endl;

for (int i = 0; i < \_m.x; i++)

std::cout << "Студент номер " << number[i] << " із середнім балом: " << mean[i] << std::endl;

std::cout << std::endl;\*/

return number;

}

main.cpp

#include <iostream>

#include <Windows.h>

#include "matrix.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int rows, cols, menu1, menu2;

std::cout << "Кількість студентів: ";

std::cin >> rows;

std::cout << "Кількість предметів: ";

std::cin >> cols;

matrix array = Create(rows, cols);

matrix &a = array;

do {

std::system("cls");

std::cout << "1.Заповнити матрицю" << std::endl;

std::cout << "2.Вивести матрицю" << std::endl;

std::cout << "3.Виконати підрахунки" << std::endl;

std::cout << "4.Вихід" << std::endl;

std::cin >> menu1;

switch (menu1) {

case 1: {

do {

system("cls");

std::cout << "1.Рандомно" << std::endl;

std::cout << "2.В ручну" << std::endl;

std::cout << "3.Вихід в головне меню" << std::endl;

std::cout << "4.Вихід з програми" << std::endl;

std::cin >> menu2;

switch (menu2) {

case 1: {

system("cls");

RandFilling(array);

Show(array);

system("pause");

break;

}

case 2: {

system("cls");

ManualFilling(array);

Show(array);

system("pause");

break;

}

case 4:

return 0;

default:

break;

}

} while (menu2 != 3);

break;

}

case 2: {

system("cls");

Show(array);

system("pause");

break;

}

case 3: {

system("cls");

Show(array);

int\* Arr = Selection(a);

ShowMean(Arr, array.x);

system("pause");

break;

}

case 4:

Delete(array);

return 0;

default:

break;

}

} while (menu1!=4);

}

Висновок: метод покрокової деталізації полягає в тому, що вихідна задача розбивається на кілька взаємозалежних під задач, кожна з яких, натомість, розбивається на під задачі і та ін. Цей про­цес завершується, коли розв'язування вихідної задачі зводиться до розв'язування ряду простих задач, для яких легко скласти ал­горитм. На кожному кроці цього процесу відбувається деталізація, тобто перехід від більш загальних задач до менш загальних, котрі, в свою чергу, допускають деталізацію до більш конкретних під задач. Схему цього процесу можна зобразити у вигляді дерева.