Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 8, варіант 23

З дисципліни « ПРОГРАМУВАННЯ, ЧАСТИНА 2 (ОБ’ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ) »

На тему: «Шаблони»

Виконав: ст. гр. КІ-15

Пилипів Р. С.

Перевірив: викладач

Козак Н.Б.

Львів – 2020

**Завдання.**

**Варіант 23.**

У матриці розташувати додатні елементи в перших стовпцях, а від‘ємні – в останніх.

**Код програми:**

//main.cpp

#include <iostream>

#include <iomanip>

#include "matrix.h"

using namespace std;

int main(void)

{

int row, col;

cout << "Enter row: ";

cin >> row;

cout << "Enter col: ";

cin >> col;

Matrix<int> m(row, col);

cout << m;

m.count\_null();

return 0;

}

//Matrix.h

#ifndef matrix\_h\_\_

#define matrix\_h\_\_

#include <iostream>

#include <iomanip>

using namespace std;

template<class T>

class Matrix

{

private:

T\*\* m\_iArr;

int size\_row;

int size\_col;

public:

//конструктор за замовчуванням

Matrix();

//конструктор з параметрами

Matrix(int size\_row\_BuUser, int size\_col\_BuUser);

//деструктор

~Matrix();

//перевантаження оператора виводу

friend ostream& operator<< <>(ostream&, Matrix<T>&);

//підрахунок нулів в матриці

int count\_null();

};

//конструктор за замовчуванням

template <class T>

Matrix<T>::Matrix()

{

size\_col = 0;

size\_row = 0;

m\_iArr = new T \* [size\_row];

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

{

m\_iArr[i] = new T[size\_col];

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

{

m\_iArr[i][j] = 0;

}

}

}

//конструктор з параметрами

template <class T>

Matrix<T>::Matrix(int size\_row\_BuUser, int size\_col\_BuUser)

{

size\_row = size\_col\_BuUser;

size\_col = size\_row\_BuUser;

m\_iArr = new T \* [size\_row];

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

{

m\_iArr[i] = new T[size\_col];

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

{

m\_iArr[i][j] = 0;

}

}

cout << "Matrix for random enter - 1\nMatrix for user enter - 2" << endl << "Please enter:";

int l = 1;

int b = 0;

cin >> l;

switch (l)

{

case 1:

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

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

{

m\_iArr[i][j] = rand() % 99;

b = rand() % 2;

if (b == 0)

{

m\_iArr[i][j] = -1 \* m\_iArr[i][j];

}

}

break;

case 2:

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

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

{

cout << "a[" << i << "]" << "[" << j << "]" << "=";

cin >> m\_iArr[i][j];

}

break;

}

}

template<class T>

Matrix<T>::~Matrix()

{

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

{

delete m\_iArr[j];

}

delete[] m\_iArr;

}

template <class T>

ostream& operator<<(ostream& output, Matrix<T>& matr)

{

int res[99][99];

int a = 0, b = 0;

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

{

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

{

if (m\_iArr[i][j] > 0)

{

res[a][b] = m\_iArr[i][j];

a++;

if (a == size\_row)

{

b++;

a = 0;

}

}

}

}

if (a != 0)

{

while (a < size\_row)

{

res[a][b] = 0;

a++;

}

}

a = 0;

b++;

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

{

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

{

if (m\_iArr[i][j] < 0)

{

res[a][b] = m\_iArr[i][j];

a++;

if (a == size\_row)

{

b++;

a = 0;

}

}

}

}

if (a != 0)

{

while (a < size\_row)

{

res[a][b] = 0;

a++;

}

}

b++;

cout << endl << endl << "Result:" << endl << endl;

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

{

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

{

cout << setw(5) << res[i][j] << " ";

}

cout << endl;

}

return 0;}

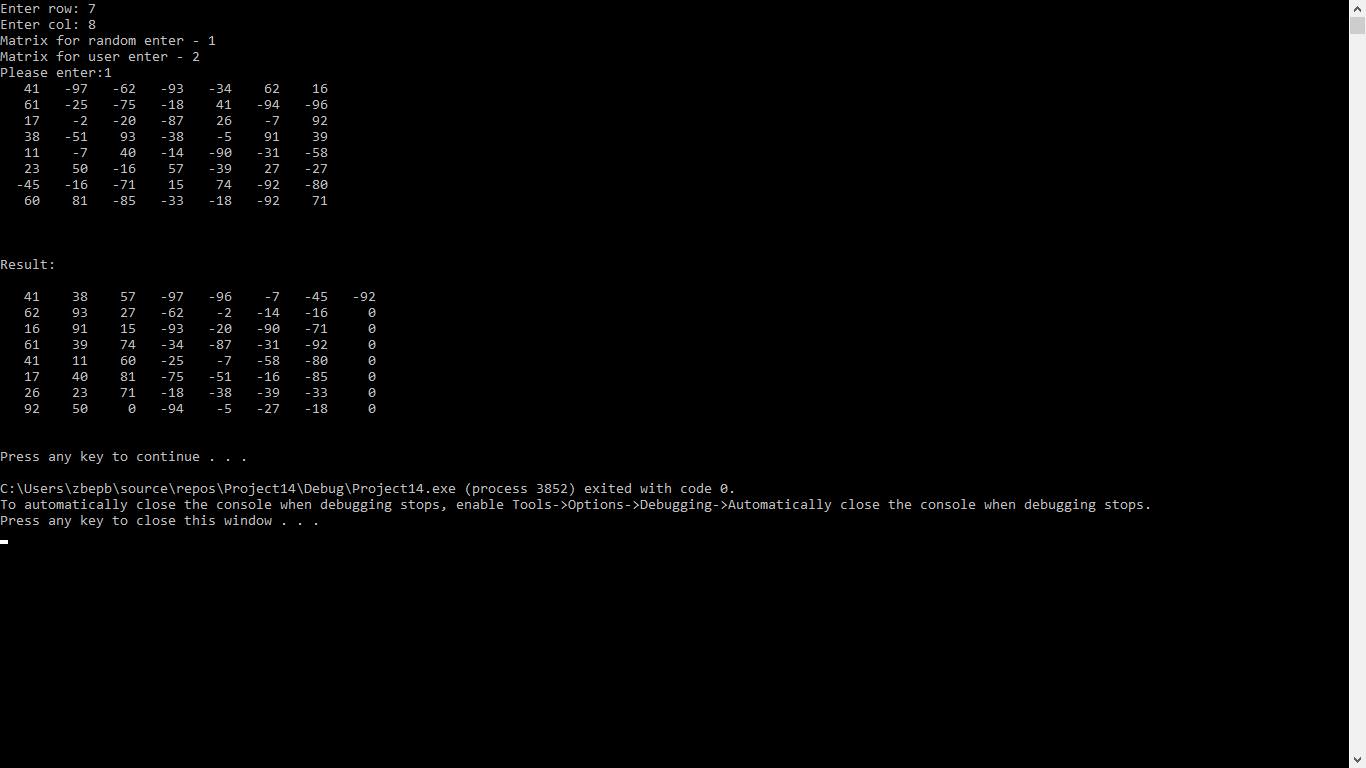
#endif

**Результат:**

1)Введення матриці в ручну:



2) Введення матриці за допомогою рандому:



**Висновок:** На цій лабораторній роботі я познайомився із створенням шаблонів.