Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра АСОІУ

ЗВІТ

про виконання комп’ютерного практикуму № 9

з дисципліни

“Об’єктно-орієнтоване програмування”

Тема: DLL С++

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Головченко М.М. |  | студент 2-го курсу  гр. ІП-52 ФІОТ  Набоков Е.М.  Варіант №10 |

Київ 2016

**ЗМІСТ**

[1 ПОСТАНОВКА ЗАДАЧІ 3](#_Toc470516599)

[2 МЕТА РОБОТИ 4](#_Toc470516600)

[3 АНАЛІТИЧНІ ВИКЛАДКИ 5](#_Toc470516601)

[4 КОД ПРОГРАМИ 6](#_Toc470516602)

[5 ПРИКЛАДИ РОБОТИ ПРОГРАМИ 9](#_Toc470516603)

[ВИСНОВКИ 10](#_Toc470516604)

# ПОСТАНОВКА ЗАДАЧІ

Розробити DLL, що містить клас для вирішення завдання згідно варіанту. Розробити тестове додаток для демонстрації роботи бібліотеки. Передбачити в класі обробку будь-яких спеціальних випадків і стандартних помилок.

Ввести квадратную матрицу и число k. Разделить k-ю строку на диагональный элемент этой строки. Вывести исходную и полученную матрицы.Підключення створеної бібліотеки здійснювати динамічно.

# МЕТА РОБОТИ

Вивчити особливості використання DLL, розглянути принципи статичного і динамічного підключення.

# АНАЛІТИЧНІ ВИКЛАДКИ

Найпростіший спосіб обміну інформацією між програмами це використання бібліотек. DLL Строго кажучи, бібліотека.

DLL це не програма, а сховище програмного коду (наприклад, функцій) і ресурсів (наприклад, форм). Вона підключається до програми динамічно, після того як програма запущена. Використовувати бібліотеки.

DLL дуже зручно. Нехай, наприклад, є програма прогнозування якогось значення на підставі даних, що вводяться користувачем. Створимо для користувача інтерфейс, а сам алгоритм розрахунку прогнозу (досить складний) виділимо в окрему функцію і помістимо в бібліотеку. DLL, підключаємо до програми.

# КОД ПРОГРАМИ

// OOP9.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <windows.h>

#include <iostream>

#include "../Library/Interface.h"

using namespace std;

int main()

{

system("color F0");

setlocale(LC\_ALL,"rus");

typedef Interface\*(\*classDll)();

typedef double(\*funcDll)(double, double);

HINSTANCE hDllInstance = LoadLibrary(L"Library.dll");

classDll func = (classDll) GetProcAddress(hDllInstance, "myFunc");

Interface\* test\_ptr = func();

if (test\_ptr != NULL) {

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

test\_ptr->print();

try {

test\_ptr->Change();

}

catch (...) {

cout << "Ho-HO" << endl;

}

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

test\_ptr->print();

delete test\_ptr;

}

FreeLibrary(hDllInstance);

system("pause");

return 0;

}

#include "stdafx.h"

#include "MyLib.h"

#include <stdexcept>

using namespace std;

namespace MathFuncs

{

MyMathFuncs::MyMathFuncs()

{

char symbols[] = { '0','1','2','3','4','5','6','7','8','9','-' };

cout << "Enter number of row and column: ";

char ch[65536];

gets\_s(ch, 65536);

if (strcmp(ch, "") == 0) throw invalid\_argument("n is invalid");

for (int i = 0; i < strlen(ch); i++) //check for not allowed symbols

{

bool flag = false;

for (int j = 0; j < 11; j++)

if (ch[i] == symbols[j]) flag = true;

if (!flag) throw invalid\_argument("n is invalid");

}

n = atoi(ch);

if (n < 0 || n > 100)

{

throw invalid\_argument("n cannot be negative or too big!");

}

m = n;

matrix = new int\*[n];

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

matrix[i] = new int[m];

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

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

{

cout << "Enter element (" << i << "," << j << ") matrix: ";

gets\_s(ch, 65536);

if (strcmp(ch, "") == 0) throw invalid\_argument("number is invalid");

for (int i = 0; i < strlen(ch); i++) //check for not allowed symbols

{

bool flag = false;

for (int j = 0; j < 11; j++)

if (ch[i] == symbols[j]) flag = true;

if (!flag) throw invalid\_argument("number is invalid");

}

matrix[i][j] = atoi(ch);

}

}

void MyMathFuncs::Change()

{

int k;

char symbols[] = { '0','1','2','3','4','5','6','7','8','9','-' };

char ch[65536];

gets\_s(ch, 65536);

cout << "Enter k:"; cin >> ch;

if (strcmp(ch, "") == 0) throw invalid\_argument("n is invalid");

for (int i = 0; i < strlen(ch); i++) //check for not allowed symbols

{

bool flag = false;

for (int j = 0; j < 11; j++)

if (ch[i] == symbols[j]) flag = true;

if (!flag) throw invalid\_argument("n is invalid");

}

k = atoi(ch);

if (k < 0 || k > 100){

throw invalid\_argument("n cannot be negative or too big!");

}

if (k < 0 || k > n)

throw invalid\_argument("row with such axis does not exist");

if (matrix[k - 1][k - 1] == 0)

throw invalid\_argument("division by zero");

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

matrix[k - 1][i] /= matrix[k - 1][k - 1];

}

}

void MyMathFuncs::print()

{

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

{

cout << endl;

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

{

printf("%4d", matrix[i][j]);

}

}

cout << endl;

}

MyMathFuncs::~MyMathFuncs()

{

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

delete[] matrix[i];

delete[] matrix;

}

}

#pragma once

class Interface

{

public:

// Returns a + b

virtual void Change() = 0;

virtual void print() = 0;

virtual ~Interface() {};

};

#pragma once

#ifdef LIBRARY\_EXPORTS

#define LIBRARY\_API \_\_declspec(dllexport)

#else

#define LIBRARY\_API \_\_declspec(dllimport)

#endif

#include "Interface.h"

#include <iostream>

using namespace std;

namespace MathFuncs

{

class MyMathFuncs: public Interface

{

int n;

int m;

int\*\* matrix;

public:

LIBRARY\_API MyMathFuncs();

// Returns a + b

LIBRARY\_API void Change();

LIBRARY\_API void print();

LIBRARY\_API ~MyMathFuncs();

};

}

extern "C" LIBRARY\_API Interface\* myFunc() {

try{

return new MathFuncs::MyMathFuncs();

}

catch (const invalid\_argument &e)

{

cout << "Caught exception: " << e.what() << endl;

return NULL;

}

}

# ПРИКЛАДИ РОБОТИ ПРОГРАМИ

Фотокопія роботи програми (Рисунок 5.1).

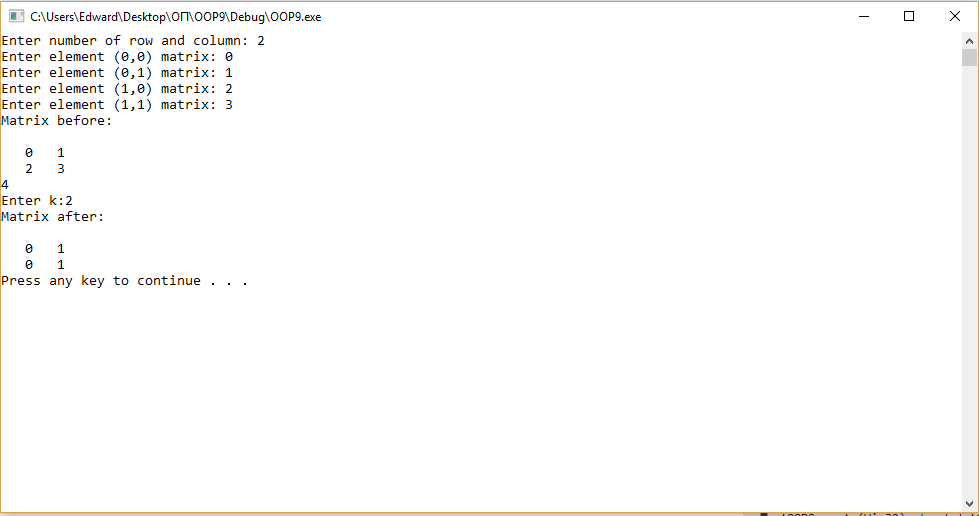


Рисунок 5.1 – Приклад виконання програми

# 

# ВИСНОВКИ

У даній лабораторній роботі, я вивчив особливості використання DLL, розглянув принципи статичного і динамічного підключення. Була створена бібліотека, яку було динамічно підключено до основної програми. Тестування програми помилок не виявило, тому можна вважати що бібліотека написана коректно.