Министерство образования и науки РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №5

ИССЛЕДОВАНИЕ МЕХАНИЗМА ДРУЖЕСТВЕННОСТИ

по дисциплине «Объектно-ориентированное программирование»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Тимофеев И.С.

г. Севастополь 2018

**Цель работы:**

Исследование основных средств описания дружественных функций и дружественных классов, и особенности использования их при написании объектно-ориентированных программ.

**Вариант №10**

Создать два класса: Матрица\_целых\_чисел (int \*\*) и Матрица\_вещественных\_чисел (double \*\*). Для каждого класса описать конструктор по умолчанию и конструктор с параметрами. Описать дружественную функцию, вычисляющую поэлементную сумму матриц и выводящую результат на экран. Учесть проверку соответствия размерностей.

**Текст программы:**

#include "stdafx.h"

#include <iostream>

#include <cstdlib>

#include <string>

using namespace std;

class MatrixDoub;

class MatrixInt {

private:

int x;

int y;

int m[20][20];

void OutputMat() {

cout << "Enter MatrInt" << endl;

int i, j;

cout << "Matr:" << endl;

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

for (j = 0; j < y; j++) {

cout << m[i][j] << " ";

}

cout << "\n";

}

};

public:

MatrixInt() {

int i, j;

x = y = 2;

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

cout << "Enter " << i + 1 << " stroky" << endl;

for (j = 0; j < y; j++) {

cout << "Enter " << j + 1 << " element " << i + 1 << " stroki" << endl;

cin >> m[i][j];

}

}

};

MatrixInt(int \_x, int \_y) {

int i, j;

x = \_x;

y = \_y;

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

cout << "Enter " << i + 1 << " stroky" << endl;

for (j = 0; j < y; j++) {

cout << "Enter " << j + 1 << " element " << i + 1 << " stroki" << endl;

cin >> m[i][j];

}

}

};

friend void Output(MatrixInt ob1, MatrixDoub ob2);

};

class MatrixDoub {

private:

int x;

int y;

double m[20][20];

void OutputMat() {

cout << "Enter MatrDoub" << endl;

int i, j;

cout << "Matr:" << endl;

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

for (j = 0; j < y; j++) {

cout << m[i][j] << " ";

}

cout << "\n";

}

};

public:

MatrixDoub() {

int i, j;

x = y = 2;

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

cout << "Enter " << i + 1 << " stroky" << endl;

for (j = 0; j < y; j++) {

cout << "Enter " << j + 1 << " element " << i + 1 << " stroki" << endl;

cin >> m[i][j];

}

}

};

MatrixDoub(int \_x, int \_y) {

int i, j;

x = \_x;

y = \_y;

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

cout << "Enter " << i + 1 << " stroky" << endl;

for (j = 0; j < y; j++) {

cout << "Enter " << j + 1 << " element " << i + 1 << " stroki" << endl;

cin >> m[i][j];

}

}

};

friend void Output(MatrixInt ob1, MatrixDoub ob2);

};

void Output(MatrixInt ob1, MatrixDoub ob2) {

int i, j;

cout << "Matr1:" << endl;

ob1.OutputMat();

cout << "Matr2:" << endl;

ob2.OutputMat();

cout << "Sum Matr1+Matr2:" << endl;

if ((ob1.x = ob2.y) && (ob1.y = ob2.y)) {

for (i = 0; i < ob1.x; i++) {

for (j = 0; j < ob1.y; j++) {

cout << ob1.m[i][j] + ob2.m[i][j] << " ";

}

cout << "\n";

}

};

};

int main() {

MatrixInt M1;

MatrixDoub M2;

Output(M1, M2);

return 0;

}

**Тестовые примеры:**

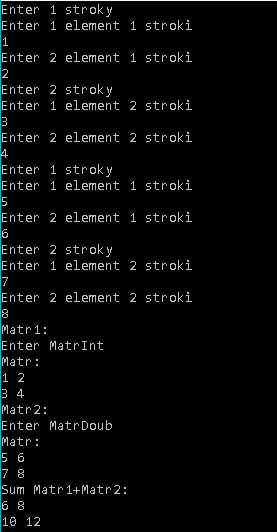


рис.1 – пример работы программы

**Вывод:**

В данной лабораторной работе были исследованы основные средства описания дружественных функций и дружественных классов, и особенности использования их при написании объектно-ориентированных программ.