Институт информационных технологий и управления в технических системах

Кафедра информационных технологий и компьютерных систем

ОТЧЁТ

по лабораторной работе №5

«ИССЛЕДОВАНИЕ МЕХАНИЗМА ДРУЖЕСТВЕННОСТИ»

по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ПИН/б-19-1-о

Мельник А.С.

Проверил ассистент

Тимофеев И.С.

Севастополь

2020

**1. ЦЕЛЬ РАБОТЫ**

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

**2. ОПИСАНИЕ ПРОЕКТА**

Разработать программу на языке С++, в которой необходимо создать два класса и дружественную функцию обоим классам. Для каждого класса описать конструктор с параметрами и без. В дружественной функции обратиться к закрытым полям обоих классов.

Разработать тестовые примеры.

Выполнить отладку программы.

Сформулировать выводы.

Оформить отчет по проделанной работе.

**3. АНАЛИЗ ЗАДАЧИ**

Для варианта задания, полученного в лабораторной работе №5, необходимо выполнить следующее:

Описать заданные по варианту классы (содержащие private поля и методы). Для каждого класса описать конструктор по умолчанию и конструктор с параметрами, а также деструктор (по необходимости). Создать функцию, дружественную обоим классам, и в ней обратиться к их закрытым полям и методам.

Вариант 3

Создать два класса: Вектор (int \*) и Матрица (int \*\*). Описать функцию, определяющую сумму их максимальных значений, как дружественную

**4. ТЕКСТ С++ ПРОГРАММЫ, ЗАДАННОЙ ВАРИАНТОМ ЗАДАНИЯ**

#include <iostream>

using std::cout;

using std::endl;

class Matrix;

class Vector

{

private:

int v[5];

public:

void Output() { cout << v << endl; }

void OutputS()

{

for (int i = 0; i < 5; i++)

{

cout << v[i];

}

}

Vector() {}

Vector(int \_v[])

{

for (int i = 0; i < 5; i++)

{

v[i] = \_v[i];

}

}

~Vector() {};

friend int Func(Vector v1, Matrix m1);

};

class Matrix

{

private:

int m[5][2];

public:

void Output() { cout << m << endl; }

void OutputS()

{

for (int i = 0; i < 5; i++)

{

cout << m[i][i];

}

}

Matrix() {}

Matrix(int \_m[5][2])

{

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 2; j++)

{

m[i][j] = \_m[i][j];

}

}

}

~Matrix() {};

friend int Func(Vector v1, Matrix m1);

};

int Func(Vector v1, Matrix m1)

{

int maxV = 1;

int maxM = 1;

for (int i = 0; i < 5; i++)

{

if (maxV < (int)v1.v[i]) maxV = (int)v1.v[i];

}

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 2; j++)

{

if (maxM < (int)m1.m[i][j]) maxM = (int)m1.m[i][j];

}

}

return maxV + maxM;

}

int main()

{

int vect[5] = { 10, 17, -10, 54, 61 };

int matr[5][2] = { {17, 0}, {48, 24}, {100, 446}, {-16, 101}, {555, 32} };

Vector\* vector = new Vector(vect);

Matrix\* matrix = new Matrix(matr);

cout << Func(\*vector, \*matrix);

}

**5. СВЕДЕНИЯ ОБ ОТЛАДКЕ ПРОГРАММЫ**

****

**6. ВЫВОД**

В ходе лабораторной работы исследованы основные механизмы дружественности.