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

Тихолаз А.А.

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 C++, которая демонстрирует использование дружественной функции. Программа должна содержать описание двух классов и вызов дружественной функции:

- Разработать тестовые примеры.

- Выполнить отладку программы.

- Получить результаты работы программы и исследовать еѐ свойства для различных режимов работы, 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 11**

Создать два класса: Матрица (int \*\*) и Координаты (две пары чисел). Описать дружественную функцию, которая меняет местами два элемента, положение которых задается координатами. Предусмотреть проверку соответствия координат и размерности матрицы.

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

#include <iostream>

using std::cout;

using std::endl;

class Matrix;

class Coord

{

private:

int first[2], second[2];

public:

void Output()

{

for (int i = 0; i < 4; i++)

{

cout << first[i];

}

}

Coord() {}

Coord(int \_f[2], int \_s[2])

{

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

{

first[i] = \_f[i];

second[i] = \_s[i];

}

}

~Coord() {};

friend void FriendlyFunction(Coord c1, Matrix m1);

};

class Matrix

{

private:

int m[4][2], lenghtX = 4, lenghtY = 2;

public:

void Output()

{

for (int i = 0; i < 4; i++)

{

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

{

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

}

cout << "\n";

}

}

Matrix() {}

Matrix(int \_m[4][2])

{

for (int i = 0; i < 4; i++)

{

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

{

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

}

}

}

~Matrix() {};

friend void FriendlyFunction(Coord c1, Matrix m1);

};

void FriendlyFunction(Coord c1, Matrix m1)

{

if (c1.first[0] > m1.lenghtX or c1.first[1] > m1.lenghtY)

cout << "Ошибка при вводе";

else if (c1.second[0] > m1.lenghtX or c1.second[1] > m1.lenghtY)

cout << "Ошибка при вводе";

int temp = m1.m[c1.first[0]][c1.first[1]];

m1.m[c1.first[0]][c1.first[1]] = m1.m[c1.second[0]][c1.second[1]];

m1.m[c1.second[0]][c1.second[1]] = temp;

for (int i = 0; i < 4; i++)

{

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

{

cout << m1.m[i][j] << " ";

}

cout << "\n";

}

cout << endl << "Успех!" << endl;

}

int main()

{

setlocale(LC\_ALL, "");

int first[2] = {3, 1};

int second[2] = {2, 0};

int matr[4][2] = { {47, 13}, {36, 52}, {29, 38}, {74, 23} };

Coord\* vector = new Coord(first, second);

Matrix\* matrix = new Matrix(matr);

cout <<"Вывод матрицы:" << endl;

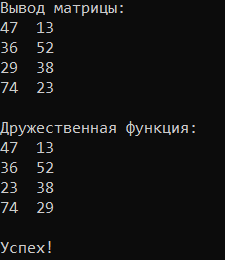
matrix->Output();

cout << endl << "Дружественная функция:" << endl;

FriendlyFunction(\*vector, \*matrix);

}

**5. СВЕДЕНИЯ ОБ ОТЛАДКЕ ПРОГРАММЫ**



**6. ВЫВОД**

В ходе работы исследованы основные средства описания дружественных функций и дружественных классов, и особенности использования их при написании объектно-ориентированных программ.