Институт информационных технологий и управления в технических системах

Кафедра информационных технологий и компьютерных систем

ОТЧЁТ

по лабораторной работе №6

«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 назначения и средств создания перегруженных операторов при написании объектно-ориентированных программ.

**2. ОПИСАНИЕ ПРОЕКТА**

Разработать программу на языке C++, которая должна содержать:

- описание заданного класса с переопределенными для него операторами;

- два объекта полученного класса;

- демонстрацию работы унарных операторов на одном объекте;

- демонстрацию работы бинарных операторов над созданными объектами. Ввод и вывод данных должен осуществляться перегруженными операторами работы с потоками.

Разработать тестовые примеры.

Выполнить отладку программы.

Получить результаты работы программы и исследовать её свойства для различных режимов работы, сформулировать выводы.

Оформить отчет по проделанной работе

**3. АНАЛИЗ ЗАДАЧИ**

Для заданного по варианту класса выполнить следующие действия:

- описать конструкторы и деструктор (по необходимости);

- переопределить оператор вывода в поток <<;

- переопределить оператор ввода из потока >>;

- переопределить заданные по варианту операторы;

- предусмотреть обработку ошибок. Создать два объекта заданного по варианту класса и на их примере продемонстрировать корректную работу всех перегруженных операторов.

**Вариант 11**

Создать класс координаты Coords (содержит две пары чисел (x1, y1) и (x2, y2)). Перегрузить операторы:

1) ! как унарный метод класса, проверяющий на совпадение пары координат (x1 на равенство с x2 и аналогично y1 и y2 — одновременно);

2) ++ как унарную дружественную функцию, увеличивающую первую пару координат на случайное число;

3) \* как бинарный метод класса, умножающий координаты на заданное число;

4) != как бинарную дружественную функцию сравнения координат двух объектов

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

#include <iostream>

#include <ctime>

using std::cout;

using std::endl;

class Coords

{

public:

int x[2];

int y[2];

bool operator !()

{

return (x[0] == y[0] and x[1] == y[1]) ? true : false;

}

friend Coords& operator ++(Coords& C);

Coords& operator \* (int num)

{

this->x[0] \*= num;

this->x[1] \*= num;

this->y[0] \*= num;

this->y[1] \*= num;

return \*this;

}

friend bool operator != (Coords& C, Coords& D);

friend std::ostream& operator << (std::ostream& out, const Coords& F);

friend std::istream& operator >> (std::istream& in, Coords& F);

Coords(int obj1[2], int obj2[2])

{

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

{

x[i] = obj1[i];

y[i] = obj2[i];

}

}

};

Coords& operator ++(Coords& C)

{

C.x[0] += rand() % 20;

C.x[1] += rand() % 20;

return C;

}

bool operator != (Coords& C, Coords& D)

{

return (C.x[0] != D.x[0] or

C.x[1] != D.x[1] or

C.y[0] != D.y[0] or

C.y[1] != D.y[1]);

}

std::ostream& operator << (std::ostream& out, const Coords& C)

{

out << C.x[0] << "/" << C.x[1] << endl << C.y[0] << "/" << C.y[1] << endl;

return out;

}

std::istream& operator >> (std::istream& in, Coords& C)

{

in >> C.x[0];

in >> C.x[1];

in >> C.y[0];

in >> C.y[1];

return in;

}

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(0));

int l1[] = { 18,10 };

int l2[] = { 14,16 };

int l3[] = { 7,0 };

Coords\* coord1 = new Coords(l1, l2);

Coords\* coord2 = new Coords(l3, l2);

cout << endl << "Проверка:" << endl << !\*coord1;

cout << endl << "Сравнение:" << endl << (\*coord1 != \*coord2);

cout << endl << "Инкремент:" << endl << ++(\*coord1);

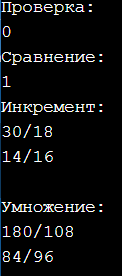
cout << endl << "Умножение:" << endl << coord1->operator\*(6);

delete coord1;

delete coord2;

}

**5. СВЕДЕНИЯ ОБ ОТЛАДКЕ ПРОГРАММЫ**



**6. ВЫВОД**

В ходе лабораторной работы были исследованы назначения и средства создания перегруженных операторов при написании объектно-ориентированных программ.