**ЛАБОРАТОРНАЯ РАБОТА №4 «ООП C++. ПЕРЕГРУЗКА ОПЕРАТОРОВ»**

**Татарников Максим А-07-22 Вариант №17**

**Задание**

Задание 1. Унарная операция

Создать класс - координаты с унарным ++ и --, -. ++ и -- постфиксная и префиксная. - меняет знак у обеих координат. ++ как функция-член, -- как дружественная функция.

Задание 2. Бинарная операция

Создать класс вещественных чисел (double). Определить оператор \*, как функцию-член и + как дружественную функцию.

**Постановка задачи**

1. Реализовать перегрузку операторов.
2. Написать код из 2 классов – один увеличивает (уменьшает) и изменяет знак координат (перегрузка ++, --, -) (пост- пре- фиксный оператор). Второй – выводит сумму и произведение двух чисел типа double.
3. Продемонстрировать работу перегрузок в коде с помощью пункт 2.
4. Обработать все исключения (выход из пределов значений переменных) (неправильный тип данных).

**Разработка программы**

**Описание классов**

Разработаны 2 класса:

1. Класс координат, разработан для проверки перегрузки оператора инкремента и декремента.
2. Класс вещественных чисел с проверкой перегрузки сложения и умножения.

**Point**

Поля

Содержит 2 переменные

Int x

Int y

Конструкторы

Конструктор с параметрами

Point(int x\_val, int y\_val)

Конструктор копирования для реализации некоторых перегрузок

Point(const Point& other)

Геттеры

int GetX()

int GetY()

Перегрузки

Point operator = (const Point& other)

Point operator ++()

Point operator ++(int value)

Point operator -()

friend void operator --(Point& value);

friend Point operator--(Point& value, int);

**Real**

Поля

Double a

Конструкторы

Real(double a\_val)

Real(const Real& other)

Геттер

double GetA()

Сеттер

void SetA(double a\_val)

Перегрузки

Real operator =(const Real& other)

Real operator \*(const Real& other)

friend ostream& operator << (ostream& os, const Real& r);

friend Real operator +(Real val, const Real& other);

**Описание пользовательского интерфейса**

При запуске программы пользователь должен ввести данные классов.

Введите значения X и Y:

54 65

Введите значения A и B:

12 85

После данные выводятся и открывается выбор задания 1 и задания 2:

X = 55 Y = 55

A = 66 B = 45

1. Унарные операторы '++', '--', '-'.

2. Бинарные операторы '+', '\*'.

После выбора пользователь переходит в одно из основных меню проверки кода и может уже проверить работоспособность перегрузок:

X = 54 Y = 65

1. Постфикс '++'

2. Постфикс '--'

3. Префикс '++'

4. Префикс '--'

5. Минус '-'

0. Выход

A = 12 B = 85

A + B = 97 A \* B = 1020

Введите 2 числа (0 0 - выход):

**Листинг**

#include <iostream>

#include <string>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

class Point {

private:

int x;

int y;

public:

Point(int x\_val, int y\_val) { x = x\_val; y = y\_val; }

Point(const Point& other) { this->x = other.x; this->y = other.y; };

int GetX() { return x; }

int GetY() { return y; }

Point operator = (const Point& other) { this->x = other.x; this->y = other.y; return \*this; }

Point operator ++() { this->x++; this->y++; return \*this; }

Point operator ++(int value) { Point temp(\*this); this->x++; this->y++; return temp; }

Point operator -() { this->x = -this->x; this->y = -this->y; return \*this; }

friend void operator --(Point& value);

friend Point operator--(Point& value, int);

};

void operator --(Point& value) { --value.x; --value.y; }

Point operator --(Point& value, int) { Point temp(value); --value; return temp; }

class Real

{

private:

double a;

public:

Real(double a\_val) { a = a\_val; }

Real(const Real& other) { this->a = other.a; };

double GetA() { return a; }

void SetA(double a\_val) { a = a\_val; }

Real operator =(const Real& other) { this->a = other.a; return \*this; }

Real operator \*(const Real& other) { Real num(\*this); num.a \*= other.a; return num; }

friend ostream& operator << (ostream& os, const Real& r);

friend Real operator +(Real val, const Real& other);

};

ostream& operator << (ostream& os, const Real& r) { return os << r.a; }

Real operator +(Real val, const Real& other) { Real num(val.a); num.a += other.a; return num; }

int main(){

setlocale(LC\_ALL, "ru");

int x\_val, y\_val;

string x\_v, y\_v, a\_v, b\_v;

double a\_val, b\_val;

cout << "Введите значения X и Y: " << endl;

cin >> x\_v >> y\_v;

try {

x\_val = stoi(x\_v);

y\_val = stoi(y\_v);

if (abs(x\_val) > 100 || abs(y\_val) > 100) throw std::out\_of\_range("Превышен диапазон");}

catch (const std::out\_of\_range&) {

x\_val = 1; y\_val = 1;

cout << "Неверный диапазон, X=1, Y=1" << endl;

\_getch();}

catch (const invalid\_argument& e) {

x\_val = 1; y\_val = 1;

cout << "Неверный тип данных, X=1, Y=1" << endl;

\_getch();}

Point pnt(x\_val, y\_val);

cout << "Введите значения A и B: " << endl;

cin >> a\_v >> b\_v;

try {

a\_val = stod(a\_v);

b\_val = stod(b\_v);

if (fabs(a\_val) > 100 || fabs(b\_val) > 100) throw std::out\_of\_range("Превышен диапазон");}

catch (const std::out\_of\_range&) {

a\_val = 1; b\_val = 1;

cout << "Неверный диапазон, A=1, B=1" << endl;

\_getch();}

catch (const invalid\_argument& e) {

a\_val = 1; b\_val = 1;

cout << "Неверный тип данных, A=1, B=1" << endl;

\_getch();}

Real first(a\_val);

Real second(b\_val);

system("cls");

int k = 1;

while (k){

cout << "X = " << pnt.GetX() << "\t" << "Y = " << pnt.GetY() << endl;

cout << "A = " << first.GetA() << "\t" << "B = " << second.GetA() << endl;

cout << "1. Унарные операторы \'++\', \'--\', \'-\'." << endl;

cout << "2. Бинарные операторы \'+\', \'\*\'." << endl;

cin >> k;

switch (k){

case 1:{

int n = 1;

while (n) {

system("cls");

cout << "X = " << pnt.GetX() << "\t" << "Y = " << pnt.GetY() << endl;

cout << "1. Постфикс \'++\'" << endl;

cout << "2. Постфикс \'--\'" << endl;

cout << "3. Префикс \'++\'" << endl;

cout << "4. Префикс \'--\'" << endl;

cout << "5. Минус \'-\'" << endl;

cout << "0. Выход" << endl;

string n\_string;

cin >> n\_string;

try { n = stoi(n\_string); }

catch (const invalid\_argument& e) { n = 10; cout << "Неверный тип данных" << endl \_getch(); }

switch (n) {

case 1: {pnt++; break; }

case 2: {pnt--; break; }

case 3: {++pnt; break; }

case 4: {--pnt; break; }

case 5: {pnt = -pnt; break; }

case 0: {system("cls"); break; }

default:break;}}break;}

case 2:{

int n = 1;

while (n) {

system("cls");

cout << "A = " << first.GetA() << "\t" << "B = " << second.GetA() << endl;

cout << "A + B = " << first + second << "\t" << "A \* B = " << first \* second << endl;

cout << "Введите 2 числа (0 0 - выход): " << endl;

cin >> a\_v >> b\_v;

try {a\_val = stod(a\_v); b\_val = stod(b\_v);}

catch (const invalid\_argument& e) {cout << "Неверный тип данных" << endl; \_getch();}

if ((a\_val == 0) & (b\_val == 0)) { system("cls"); break; }

first.SetA(a\_val);

second.SetA(b\_val);}break;}

case 0: {exit(0); break; }

default: system("cls"); break;}} }

**Тестирование**

1.

Input 12 64 98 02 1 1 2 2 0 26 45 0 0

Output

X = 11 Y = 63

A = 98 B = 2

Input 0 0 0 0 0 0 0 0 0 0 0 0 0

Output

X = 0 Y = 0

A = 0 B = 0

Input 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0

Output

X = 6 Y = 6

A = 1 B = 1

Input sfhg sdh fgj sdf fgj gh 5454hdg56 6h54 6sd54hdf6g j54dg654 df6gj54 d6g 0 0 0

Output

X = 1 Y = 1

A = 1 B = 1

Input 654 98 3 49 64 68 46541 684 9 84 984 5 4 98 16 54 6 84 4 3 8 84 6 84 6 84 84 9 49 81 0 0 0

Output

X = 1 Y = 1

A = 3 B = 49