**Федеральное государственное бюджетное общеобразовательное учреждение   
высшего образования**

**Национальный исследовательский университет**

**«МЭИ»**

**Лабораторная работа № 4**

по курсу «Технология программирования»

**По теме: «ООП C++. ПЕРЕГРУЗКА ОПЕРАТОРОВ»**

**Выполнил:**

студент: Нестеров А.C.

группа: Аэ-21-23  
вариант: 8

**Москва, 2024**

*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 целых чисел (long). Определить оператор -, как функцию-член и + как дружественную функцию.

**1.Постановка задачи**

Создать оконное приложение, перегрузить операторы и продемонстрировать их работу.

Функции:

1)

* Перегруженные инкремент и декремент
* Перегруженный «-», меняющий знак обеих координат

2)

* Перегруженные + и -

Входные данные:

1)

* Два числа по модулю <1000 – начальные координаты

2)

* Два числа по модулю <2 000 000 000 – начальный класс чисел

Выходные данные:

1)

* Два числа по модулю <=1000 – измененные координаты

2)

* Два числа по модулю <2 050 688 107– измененный класс чисел

Вид приложения:

* Консоль c++, VS 2019

**2. Разработка программы**

2.1 Описание классов, свойств и методов классов

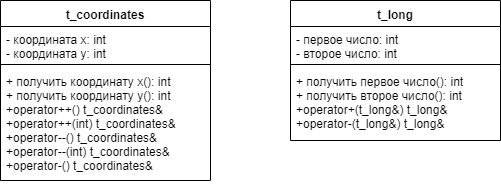
Создаем 2 класса:  
Первый класс - координаты, в нем перегружаем операторы инкремента и декремента, а так же оператор «-», предназначенный для изменения знака обеих координат   
Второй класс – два целых числа, в нем перегружаем операторы «+» и «-», предназначенные для работы с таким же классом

Таб. 1. Описание структур данных

|  |  |  |
| --- | --- | --- |
| Имя | Описание | Структура |
| t\_coordinates | класс координат | Класс |
| x\_ | координата x | Свойство класса |
| y\_ | координата y | Свойство класса |
| x | возвращение координаты x | Метод класса |
| y | возвращение координаты y | Метод класса |
| -- | уменьшает значение обеих координат на 1 | Перегрузка класса |
| ++ | увеличивает значение обеих координат на 1 | Перегрузка класса |
| - | меняет знак обеих координат | Перегрузка класса |
| t\_long | класс чисел | Класс |
| long1\_ | первое число класса | Свойство класса |
| long2\_ | второе число класса | Свойство класса |
| long1 | возвращение первого числа | Метод класса |
| long2 | возвращение второго числа | Метод класса |
| + | Прибавляет значение к каждому числу класса | Перегрузка класса |
| - | Вычитает значение из каждого числа класса | Перегрузка класса |
| R1 | Случайное число |R1| < 50 688 107 Для изменения класса чисел | локальная переменная |
| R2 | Случайное число |R2| < 50 688 107 Для изменения класса чисел | локальная переменная |

Продолжение таб1

Таб. 2. Диаграмма классов



2.2 Разработка схемы алгоритма   
см. приложение 1

2.3 Описание пользовательского интерфейса

Категории пользователей:

* Обычный пользователь ПК

Функции пользователя:

* Ввод 4 чисел

Диалог с пользователем показан на рис.1

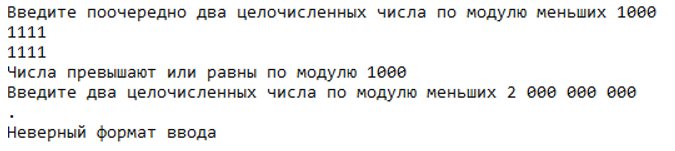


Рис.1 Диалог с пользователем

**3. Реализация и тестирование программы**

3.1. Описание разработанной программы

Программа представляет собой 2 класса с перегруженными операторами.

* В первом классе хранятся 2 координаты  
  Перегружены унарные операторы инкремент и декремент (постфиксные и префиксные), изменяющие обе координаты при вызове

t\_coordinates& operator++ ()

{

x\_++;

y\_++;

return \*this;

};

t\_coordinates operator++ (int)

{

t\_coordinates temp = \*this;

++\* this;

return temp;

};

friend t\_coordinates& operator-- (t\_coordinates& T);

friend t\_coordinates& operator--(t\_coordinates& T, int);

t\_coordinates& operator-- (t\_coordinates& T)

{

T.x\_--;

T.y\_--;

return T;

};

t\_coordinates& operator--(t\_coordinates& T, int)

{

t\_coordinates temp = T;

--T.x\_;

--T.y\_;

return temp;

};

и оператор «-», меняющий знак у обеих координат.

t\_coordinates& operator-()

{

x\_ \*= -1;

y\_ \*= -1;

return \*this;

};

};

* Во втором классе хранятся 2 числа

Перегружены бинарные операторы «+» и «-», работающие с таким же классом, и применяющие стандартные операторы суммы и разности к соответствующим полям класса.

t\_long operator+ (t\_long& num)

{

long1\_ += num.long1\_;

long2\_ += num.long2\_;

return \*this;

};

friend t\_long& operator- (t\_long& num1, t\_long& num2);

t\_long& operator- (t\_long& num1, t\_long& num2)

{

num1.long1\_ -= num2.long1\_;

num1.long2\_ -= num2.long2\_;

return num1;

};

Листинг – приложение 2

3.2. Тестирование программы

* Объект испытаний – lab4
* Цель испытаний – проверка характеристик программы
* Порядок испытаний – проверка работоспособности, проверка обработки исключительных ситуаций
* Методы испытаний - функциональное тестирование

Тесты см. Рисунок 2 a-d

1. Исключение для первого и второго классов на неверный формат ввода
2. Исключение для первого и второго классов на превышение допустимых чисел
3. Граничные условия + Противоположные по знаку
4. Ноль

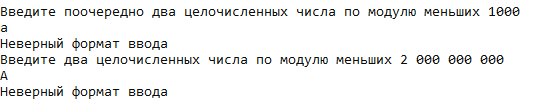


Рис2a

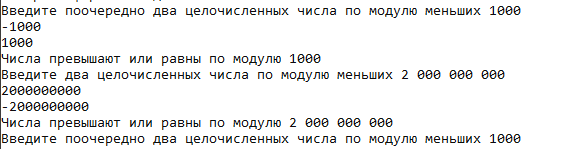


Рис2b

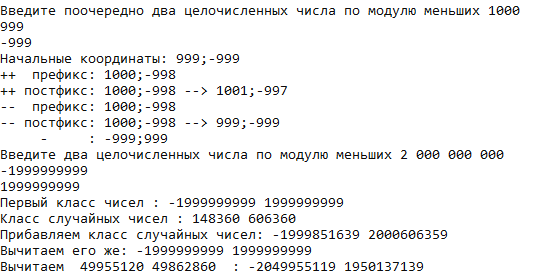


Рис2c

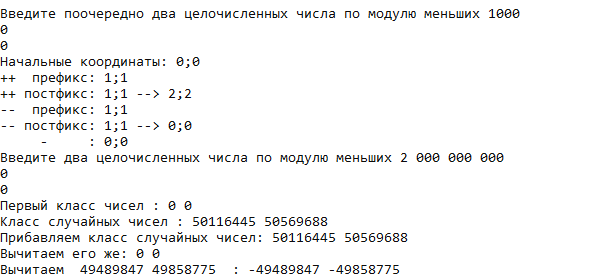


Рис2d