Міністерство науки і освіти України

Кломийський політехнічний коледж

Національного університету «Львівська політехніка»

**Циклова комісія: Інженерія програмного забезпечення**

**Практична робота №7**

**З дисципліни**

**«Об’єктно-орієнтоване програмування»**

Виконав студент групи П-31

Ракута Любомир

Коломия 2018

Теорія

Перегрузка операторів – потужний інструмент С++, за допомогою якого різні оператори можуть виконуватися відносно користувацьких типів даних та структур даних. Їх можна перегрузити за допомогою слова operator@ де @ - оператор, який потрібно перегрузити.

Є два способи перегрузки операторів. Через дружні функції та методи самого класу.

Задача

1. Створити клас і для нього перегрузити оператори +,+=,==
2. Створити клас координат і для нього перегрузити оператор \* для множення об’єктів між собою та на ціле число

Код програми 1.

#include "pch.h"

#include <iostream>

#include <conio.h>

class Integer

{

private:

int value;

public:

int getValue() // метод, що повертає значення value

{

return value;

}

Integer(int i) { value = i; } //конструктор

//об'явлення дружньої функції для перегрузки оператора +

friend const Integer operator+(const Integer& left, const Integer& right);

//об'явлення дружньої функції для перегрузки оператора +=

friend Integer operator+=(Integer& left, const Integer& right);

//об'явлення дружньої функції для перегрузки оператора ==

friend bool operator==(const Integer& left, const Integer& right);

};

//перегрузка оператора +

const Integer operator+(const Integer& left, const Integer& right)

{

// повернення об'єкта Integer що містить суму полів лівого і правого операнда

return Integer(left.value + right.value);

}

//перегрузка оператора +=

Integer operator+=(Integer& left, const Integer& right)

{

// виконується оператор +=

left.value += right.value;

//повертається об'єкт, в якого записано результат

return left;

}

//перегрузка оператора ==

bool operator==(const Integer& left, const Integer& right)

{

//повертається результат порівняння правого та лівого операндів

return left.value == right.value;

}

using namespace std;

int main()

{

Integer a(5),b(6); // створюються два об'єкти з значеннями 5 та 6

Integer c = a + b;// виконується перегружений + і результат записується в с

cout << "c = a + b = " << c.getValue() << endl;

a += b; // виконується перегружений +=

cout << "a += b = " << a.getValue() << endl;

if (a == b) //якщо а = b тоді вивід буде true а якщо ні то false

{

cout << "a == b true" << endl;;

}

else

{

cout << "a == b false" << endl;

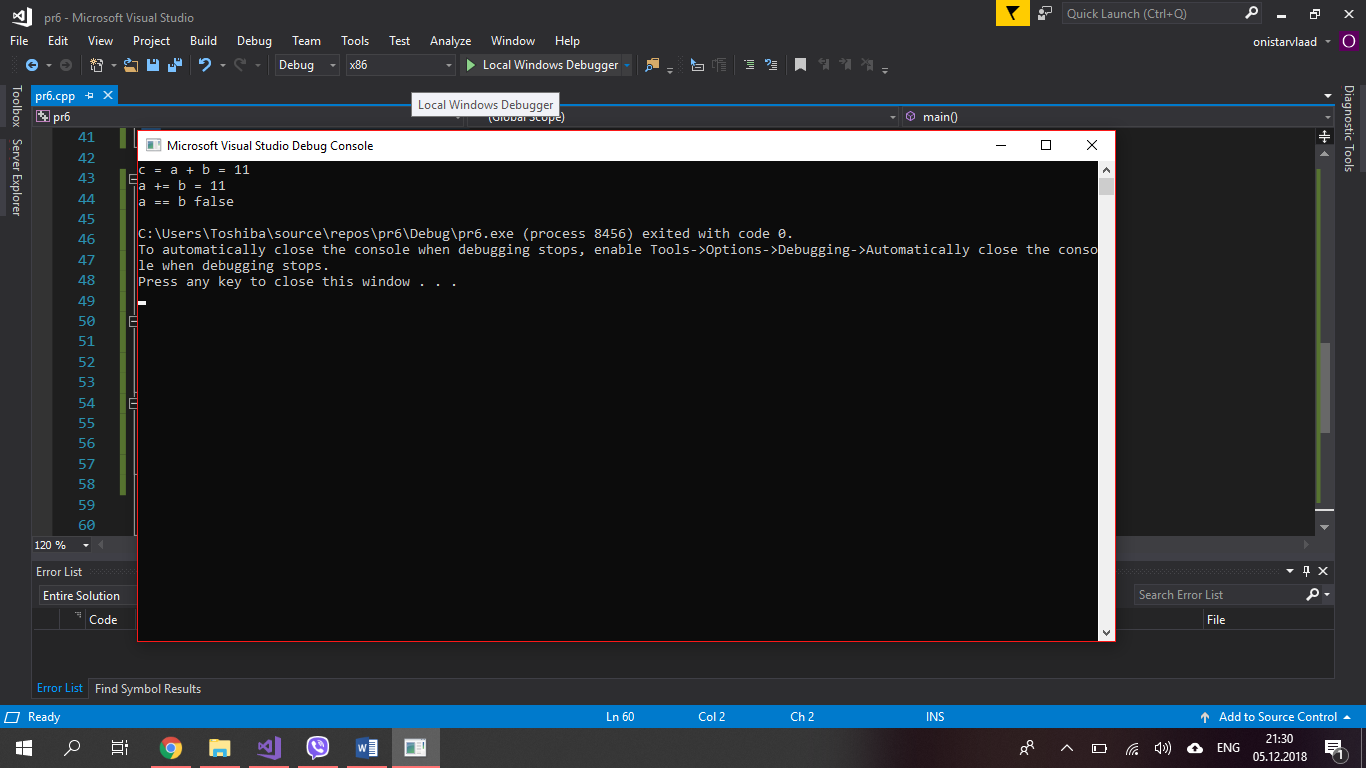
}

\_getch();

return 0;

}

Консоль 1



Код програми 2.

#include "pch.h"

#include <iostream>

#include <conio.h>

using namespace std;

class Coord

{

int x, y, z;

public:

Coord() {} // конструктор без параметрів

Coord(int x, int y, int z) // конструктор з параметрами

{

this->x = x;

this->y = y;

this->z = z;

}

void showCoords() // метод що виводить значення координат

{

cout << "x = " << this->x << endl

<< "y = " << this->y << endl

<< "z = " << this->z << endl;

cout << endl;

}

//об'явлення дружньої функції перегрузки \* що дає змогу множити два об'єкти класу Coord

friend Coord operator\*(const Coord& left, const Coord& right);

//об'явлення дружньої функції перегрузки \* що дає змогу множити об'єкт Coord та ціле число

friend Coord operator\*(const Coord& left, int right);

};

//перегрузка оператора \* де перемножуються між собою координати двох об'єктів

Coord operator\*(const Coord& left, const Coord& right)

{

Coord temp;

temp.x = left.x \* right.x;

temp.y = left.y \* right.y;

temp.z = left.z \* right.z;

return temp;

}

//перегрузка оператора \* де перемножуються координати об'єкта та ціле число

Coord operator\*(const Coord& left, int right)

{

Coord temp;

temp.x = left.x \* right;

temp.y = left.y \* right;

temp.z = left.z \* right;

return temp;

}

using namespace std;

int main()

{

//створюються два об'єкти з координатами

Coord A(4, 5, 9);

Coord B(8, 2, 3);

//створюється третій об'єкт, в який зберігається результат множення А \* B

Coord C = A \* B;

C.showCoords();

C = A \* 3;

C.showCoords();

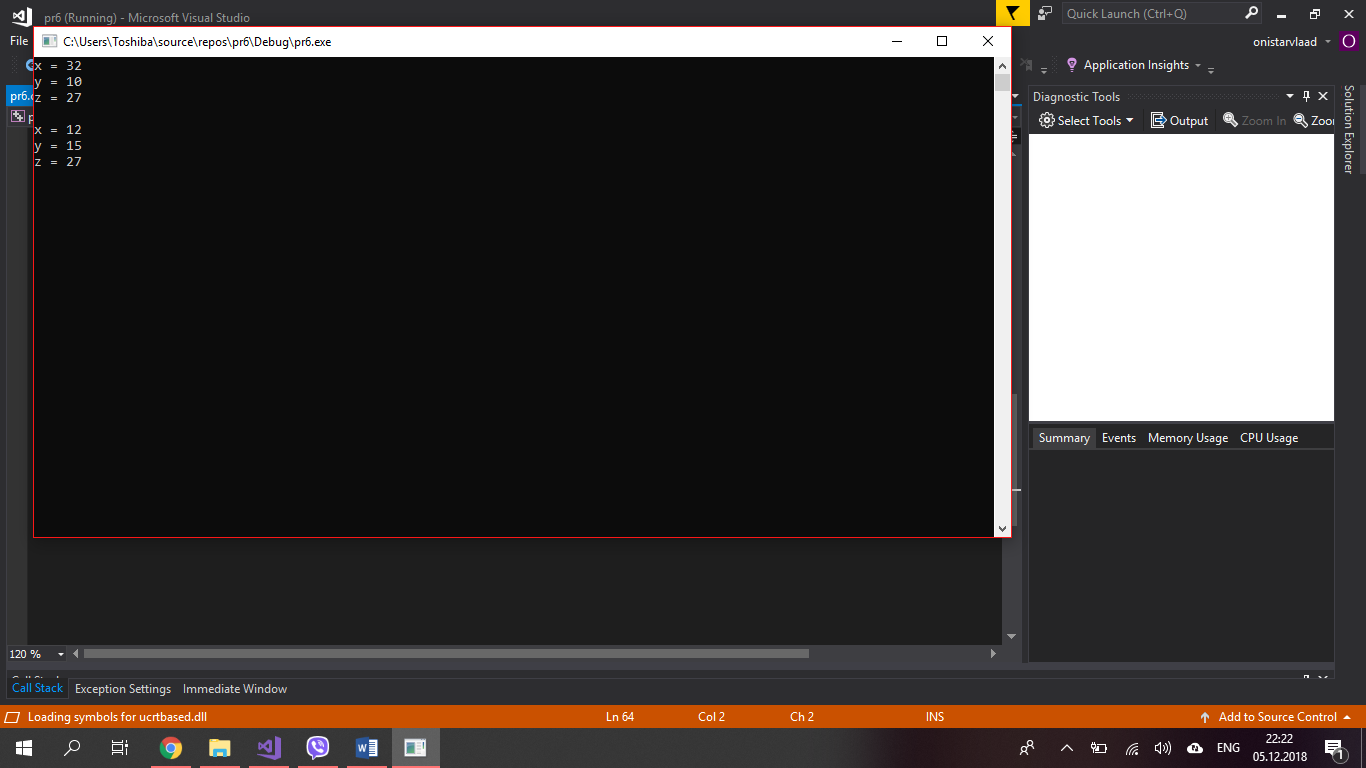
//C = 3 \* A; //так не можна тому, що нема перегруженого оператора з таким порядком параметрів

\_getch();

return 0;

}

Консоль 2



**Висновок**

На даній практичній роботі я оволодів навиками роботи з перегрузкою операторів, рекомендований метод перегрузки яких є перегрузка через дружні функціїю