**Лабораторна №3**

**Тема:** Перевантаження операцій класу

**Мета:** ознайомитись зі способами перевантаження операцій та навчитись використовувати їх при роботі з об’єктами.

1. Створити клас – вектор, який має у закритій частині вказівник на дані цілого типу та кількість елементів. Визначити необхідні конструктори, деструктор. Перевантажити потокові операції введення і виведення елементів вектора, операції + , – , \* , = та [ ]

#include <iostream>

using namespace std;

class Vector

{

private:

int x;

int y;

int z;

public:

Vector(int x, int y, int z)

{

this->x = x;

this->y = y;

this->z = z;

}

Vector() {

x = 1;

y = 1;

z = 1;

}

Vector operator -(Vector&);

Vector operator \*(Vector&);

bool operator =(Vector&);

void Output();

Vector operator +(Vector&);

void Input();

}rez, a1, a2, a3;

void Vector::Output() {

cout << "x= " << x << "y= " << y << "z= " << z;

}

void Vector::Input() {

cout << "Введіть х";

cin >> x;

cout << "Введіть y";

cin >> y;

cout << "Введіть z";

cin >> z;

}

Vector Vector::operator- (Vector& c1)

{

return Vector(x - c1.x, y - c1.y, z - c1.z);

}

Vector Vector::operator\* (Vector& c1)

{

return Vector(x \* c1.x, y \* c1.y, z \* c1.z);

}

bool Vector::operator= (Vector& c1)

{

bool rez;

if (x == c1.x || y == c1.y || z == c1.z)

{

rez = true;

}

else {

rez = false;

}

return rez;

}

Vector Vector::operator+ (Vector& c1) {

return Vector(x + c1.x, y + c1.y, z + c1.z);

}

int main()

{

setlocale(LC\_CTYPE, "ukr");

Vector aa1(1, 1, 1);

Vector aa2(1, 1, 1);

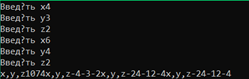
Vector rez3(0, 0, 0);

aa2.Input();

rez3 = aa1 + aa2;

rez3.Output();

}



**Висновок:** я ознайомився зі способами перевантаження операцій та навчитись використовувати їх при роботі з об’єктами.