

Міністерство освіти і науки України  
Національний технічний університет України “Київський політехнічний  
інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4 з дисципліни  
«Основи програмування 2. Модульне програмування»

«Перевантаження операторів»

Варіант 29

Виконав студент

ПІ-15 Рибалка Ілля Сергійович  
(шифр, прізвище, ім'я, по батькові)

Перевірив

Всечерковська Анастасія Сергіївна  
( прізвище, ім'я, по батькові)

## Лабораторна робота 3

### Перевантаження операторів

**Мета** - вивчити механізм створення класів з використанням перевантажених операторів.

#### Індивідуальне завдання

##### 1.1.1 Вимоги до програми

- 1) Реалізація коду відповідно до модульного підходу.
- 2) Виведення усіх вхідних, проміжних і вихідних даних.

#### Варіант 29

29. Розробити клас "Вектор", членами якого є сферичні координати вектора у просторі. Реалізувати для нього декілька конструкторів, геттери, метод обчислення його декартових координат. Перевантажити оператори: "+" – для збільшення полярного кута на вказану величину (у градусах), "!=" – для перевірки неколінеарності векторів. Створити три вектори (B1, B2, B3), використовуючи різні конструктори. Збільшити полярний кут вектора B1 на вказану величину. Знайти декартові координати зміненого вектора B1. Перевірити колінеарність векторів B2 і B3.

#### Розв'язання

#### Код

##### main.cpp

```
#include "func.h"

int main()
{
    int deg, r, phi, thet;

    std::string str;

    inputdat(&str, &r, &phi, &thet);

    Vector B1(str), B2(r, phi, thet), B3;

    inputdeg(&deg);
```

```
B1+=deg;

B1.deccord();

findout(B2, B3);

return 0;

}
```

### func.h

```
#include "Vector.h"

void inputdat(std::string *, int *, int *, int *); // Введення даних про вектори

void inputdeg(int *); // Введення кута

void findout(Vector, Vector); // Перевірка векторів на колінеарність
```

### func.cpp

```
#include "func.h"

void inputdat(std::string *str, int *r, int *phi, int *thet)
{
    std::string ts;

    int tr, tp, tt;

    std::cout << "Введіть значення першого вектора (Напр. - 30,20,40) - ";

    std::getline(std::cin, ts);

    *str = ts;

    std::cout << "Введіть значення другого вектора (Напр. - 30,20,40) - ";

    std::getline(std::cin, ts);

    int n1 = ts.find(','), n2 = ts.find(',', n1+1);

    tr = std::stoi(ts.substr(0,n1));
```

```

    tp = std::stoi(ts.substr(n1+1,n2-n1-1));

    tt = std::stoi(ts.substr(n2+1,ts.size()-n2));

    *r = tr;

    *phi = tp;

    *thet = tt;

}

void inputdeg(int *d)

{

    int deg;

    std::cout << "Введіть величину на яку збільшити полярний кут = ";

    std::cin >> deg;

    *d = deg;

}

void findout(Vector B2, Vector B3)

{

    if(B2!=B3) std::cout << "Вектори не колінеарні" << std::endl;

    else std::cout << "Вектори колінеарні" << std::endl;

}

```

## Vector.h

```

#pragma once

#include <iostream>

#include <string>

#include <cmath>

```

```

class Vector
{
private:
    int r; // Відстань від початку координат

    int phi; // Азимутальний кут

    int thet; // Полярний кут

public:
    Vector(); // Створення вектора

    Vector(int, int, int); // Створення вектора за введеними даними

    Vector(std::string); // Створення вектора за введеною строкою

    Vector operator+=(int); // Збільшити полярний кут на задану величину

    bool operator!=(Vector &); // Перевірити на неколінеарність

    void deccord(); // Отримати декартові координати на вивід

    void deccord(float *, float *, float *); // Отримати декартові координати у задані змінні

    int Getr(); // Отримати відстань від початку координат

    int Getphi(); // Отримати азимутальний кут

    int Getthet(); // Отримати полярний кут

};

```

## Vector.cpp

```

#include "Vector.h"

Vector::Vector()
{
    this->r = 10;
    this->phi = 10;
    this->thet = 10;
}

```

## Основи програмування 2. Модульне програмування

```
Vector::Vector(int r, int phi, int thet)
{
    if(r<0) throw std::invalid_argument("Неправильно вказана відстань");
    this->r = r;
    if(phi<0 || phi>=360) throw std::invalid_argument("Неправильно вказаний азимутальний кут");
    this->phi = phi;
    if(thet<0 || thet>180) throw std::invalid_argument("Неправильно вказаний полярний кут");
    this->thet = thet;
}

Vector::Vector(std::string str)
{
    std::string r, phi, thet;
    int n1 = str.find(','), n2 = str.find(',', n1+1);
    r = str.substr(0,n1);
    phi = str.substr(n1+1,n2-n1-1);
    thet = str.substr(n2+1,str.size()-n2);
    this->r = std::stoi(r);
    this->phi = std::stoi(phi);
    this->thet = std::stoi(thet);
}

Vector Vector::operator+=(int deg)
{
    Vector temp;
    temp.r = this->r;
    temp.phi = this->phi;
    int t = this->thet+deg;
    while(t>180) t-=180;
    temp.thet = t;
    return temp;
}

bool Vector::operator!=(Vector &oth)
{
    float n1, n2, n3;
    float x1, y1, z1;
    this->deccord(&x1, &y1, &z1);
    float x2, y2, z2;
    oth.deccord(&x2, &y2, &z2);
    if(x1==0 || x2==0 || y1==0 || y2==0 || z1==0 || z2==0) throw
std::invalid_argument("Координати одного з векторів дорівнюють 0");
}
```

```

        n1 = x1/x2;
        n2 = y1/y2;
        n3 = z1/z2;

        if(n1!=n2 && n2!=n3 && n1!=n3) return true;
        else return false;
    }

void Vector::deccord()
{
    float x, y, z;

    x = this->r*sin(this->thet)*cos(this->phi);
    z = this->r*cos(this->thet);
    y = z*sin(this->phi);

    std::cout << "Декартові координати:" << std::endl << "x - " << x << std::endl << "y
- " << y << std::endl << "z - " << z << std::endl;
}

void Vector::deccord(float *x, float *y, float *z)
{
    *x = this->r*sin(this->thet)*cos(this->phi);
    *z = this->r*cos(this->thet);
    *y = this->r*cos(this->thet)*sin(this->phi);
}

int Vector::Gettr()
{
    return this->r;
}

int Vector::Getphi()
{
    return this->phi;
}

int Vector::Getthet()
{
    return this->thet;
}

```

## Тестування

```
Введіть значення першого вектора (Напр. – 30,20,40) – 20,30,40  
Введіть значення другого вектора (Напр. – 30,20,40) – 10,10,10  
Введіть величину на яку збільшити полярний кут = 60  
Декартові координати:  
x – 2.2987  
y – 13.1791  
z – -13.3388  
Вектори колінеарні
```

### Висновок

Я дослідив механізм створення класів з використанням перевантажених операторів. В ході роботи над лабораторною роботою було створено програму для обробки 3 векторів в сферичних координатах.