
Основи програмування – 2

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.
Методології програмування»

«Перевантаження операторів»

Варіант 29

Виконав студент ПІ-13 Романюк Діана Олексіївна
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 4

Мета – вивчити механізми створення класів з використанням перевантаження операторів (операцій).

Варіант 29

Постановка задачі:

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.

Код програми на C++:

Header.h

```
#pragma once
#include <iostream>
#include <string>
#include <cmath>
#include <vector>

using namespace std;

class Vector {
private:
    double Ro;
    int Theta;
    int Phi;
public:
    Vector(double _Ro, int _Theta, int _Phi);
    Vector(string line);
    Vector(Vector& pt);

    double GetRo();
    int GetTheta();
    int GetPhi();

    double FindX();
    double FindY();
    double FindZ();

    Vector operator+= (int deg);
    friend bool operator!= (Vector& a, Vector& b);
};

void FindAngle(int& degree);

std::vector<std::string> split(std::string);
```

Lab4.cpp

```
#include "Header.h"
using namespace std;
```

Основи програмування – 2

```
int main() {
    double _ro;
    int _theta, _phi;
    cout << "1) Enter coordinates: ";
    cout << "\tRo: "; cin >> _ro; cout << endl;
    cout << "\t\t\tTheta: "; cin >> _theta; cout << endl;
    cout << "\t\t\tPhi: "; cin >> _phi; cout << endl;
    Vector B1(_ro, _theta, _phi);
    cout << "Vector B1: (" << B1.GetRo() << ", " << B1.GetTheta() << ", " << B1.GetPhi() <<
    ")\\n";
    string line;
    cout << "2) Enter coords: ";
    cin.ignore();
    getline(cin, line, '\\n');
    Vector B2(line);
    cout << "Vector B2: (" << B2.GetRo() << ", " << B2.GetTheta() << ", " << B2.GetPhi() <<
    ")\\n";
    Vector B3(B1);
    cout << "Vector B3: (" << B3.GetRo() << ", " << B3.GetTheta() << ", " << B3.GetPhi() <<
    ")\\n";
    int value;
    cout << "Enter value to increase Theta of B1: "; cin >> value;
    B1 += value;
    cout << "Changed B1: (" << B1.GetRo() << ", " << B1.GetTheta() << ", " << B1.GetPhi() <<
    ")\\n";
    cout << "Cartesian coords of B1: (" << B1.FindX() << ", " << B1.FindY() << ", " <<
    B1.FindZ() << ")\\n";
    cout << "Is B2 collinear to B3? " << (B2 != B3) << endl;
}
```

Header.cpp

```
#include "Header.h"
using namespace std;

Vector::Vector(double _Ro, int _Theta, int _Phi) {
    Ro = _Ro;
    Theta = _Theta;
    Phi = _Phi;
}

Vector::Vector(string line) {
    Ro = stod(split(line)[0]);
    Theta = stoi(split(line)[1]);
    Phi = stoi(split(line)[2]);
}

Vector::Vector(Vector& vec) {
    Ro = vec.Ro;
    Theta = vec.Theta;
    Phi = vec.Phi;
}

double Vector::GetRo() {
    return Ro;
}

int Vector::GetTheta() {
    return Theta;
}

int Vector::GetPhi() {
    return Phi;
}
```

Основи програмування – 2

```
double Vector::FindX() {
    return (Ro * sin(Theta)) * cos(Phi);
}

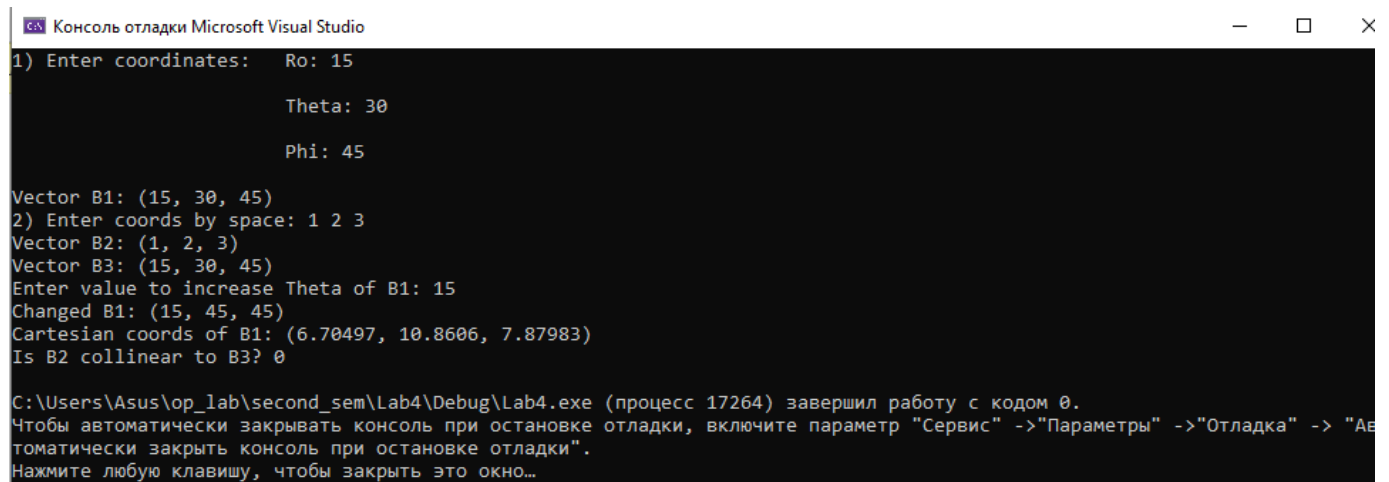
double Vector::FindY() {
    return (Ro * sin(Theta)) * sin(Phi);
}

double Vector::FindZ() {
    return Ro * cos(Theta);
}

Vector Vector::operator+= (int deg) {
    this->Theta += deg;
    return *this;
}

bool operator!= (Vector& a, Vector& b) {
    return (a.GetRo() / b.GetRo() > 0) && (a.GetTheta() == b.GetTheta()) && (a.GetPhi() ==
b.GetPhi());
}

vector<string> split(string line) {
    vector<string> coords;
    char sep = ' ';
    line += sep;
    string buffer = "";
    for (int i = 0; i < line.length(); i++) {
        if (line[i] == sep) {
            if (buffer.length() > 0)
                coords.push_back(buffer);
            buffer = "";
        }
        else
            buffer += line[i];
    }
    return coords;
}
```



```
Консоль отладки Microsoft Visual Studio
1) Enter coordinates:  Ro: 15
                     Theta: 30
                     Phi: 45
Vector B1: (15, 30, 45)
2) Enter coords by space: 1 2 3
Vector B2: (1, 2, 3)
Vector B3: (15, 30, 45)
Enter value to increase Theta of B1: 15
Changed B1: (15, 45, 45)
Cartesian coords of B1: (6.70497, 10.8606, 7.87983)
Is B2 collinear to B3? 0

C:\Users\Asus\op_lab\second_sem\Lab4\Debug\Lab4.exe (процесс 17264) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Висновки: під час лабораторної роботи ми вивчили механізми створення класів з використанням перевантаження операторів (операцій).