

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 № 3
з дисципліни «Основи програмування –
2. Методології програмування»

«Перевантаження операторів»

Варіант 11

Виконав студент ПІ-13 Дем'янчук Олександр Петрович
(шифр, прізвище, ім'я, по батькові)

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

Київ 2022

Лабораторна робота 3

Варіант 11

Мета роботи - вивчити механізми створення класів з використанням перевантажених операторів (операцій).

Завдання: Визначити клас "Точка в просторі", членами якого є координати точки в циліндричній системі координат. Реалізувати для нього декілька конструкторів, геттери, метод обчислення відстані від точки до початку системи координат. Перевантажити оператори: префіксний "++" — для збільшення азимутального кута на 1° , "+=" - для збільшення радіальної відстані r на вказану величину, "==" – для визначення рівності відповідних координат двох точок. Створити три точки (P1, P2, P3), використовуючи різні конструктори. Інкрементувати азимутальний кут точки P1, а радіальну відстань r точки P2 збільшити на вказану величину. З'ясувати, чи рівні між собою відповідні координати цих двох точок (P1 і P2). Визначити відстань від точки P3 до початку системи координат.

Код програми

C++

main.cpp

```
#include "lib.h"

int main() {
    setlocale(LC_CTYPE, "rus");
    double ro, z = 0;
    int phi = 0;
    cout << "1) Введіть координати: ";
    cout << "\tro: "; cin >> ro;
    cout << "\t\t\tphi: "; cin >> phi;
    cout << "\t\t\tz: "; cin >> z;
    cout << endl;
    cPoint P1(ro, phi, z);
    printf("P1 (%f %d %f)\n", P1.GetRo(), P1.GetPhi(), P1.GetZ());
    string line;
    cout << "2) Введіть координати через пробіл:";
    cin.ignore();
    getline(cin, line, '\n');
    cPoint P2(line);
    printf("P2 (%f %d %f)\n", P2.GetRo(), P2.GetPhi(), P2.GetZ());
    cPoint P3(P1);
    printf("P3 (%f %d %f)\n", P3.GetRo(), P3.GetPhi(), P3.GetZ());
    cout << "Інкрементуємо P1:\n";
    ++P1;
    printf("P1 (%f %d %f)\n", P1.GetRo(), P1.GetPhi(), P1.GetZ());
    int value; cout << "Введіть значення: "; cin >> value; cout << endl;
    P2 += value;
    printf("%f %d %f\n", P2.GetRo(), P2.GetPhi(), P2.GetZ());
    cout << "P1 рівна P2? " << (P1 == P2) << endl;
    cout << "Відстань від точки P3: " << P3.FindDistance() << endl;
}
```

lib.h

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

using namespace std;

class cPoint {
private:
    double ro = 1;
    int phi = 45;
    double z = 1;
public:
    cPoint(double Ro = 1, int Phi = 45, double Z = 1);
    cPoint(string line);
    cPoint(cPoint& pt);

    double GetRo();
    int GetPhi();
    double GetZ();

    double FindDistance();

    cPoint operator++ ();
    cPoint operator+= (double val);
    friend bool operator== (cPoint& a, cPoint& b);
};

vector<string> splitCoords(string);
```

lib.cpp

```
#include "lib.h"

cPoint::cPoint(double Ro, int Phi, double Z) {
    this->ro = Ro;
    this->phi = Phi;
    while (phi > 360) { phi -= 360; }
    this->z = Z;
}

cPoint::cPoint(string line) {
    ro = stod(splitCoords(line)[0]);
    phi = stoi(splitCoords(line)[1]);
    while (phi > 360) { phi -= 360; }
    z = stod(splitCoords(line)[2]);
}

cPoint::cPoint(cPoint& pt) {
    this->ro = pt.ro;
    this->phi = pt.phi;
    this->z = pt.z;
}

int cPoint::GetPhi() {
    return phi;
}
```

```

double cPoint::GetRo() {
    return ro;
}

double cPoint::GetZ() {
    return z;
}

double cPoint::FindDistance() {
    return sqrt(ro * ro + z * z);
}

cPoint cPoint::operator++ () {
    this->phi++;
    this->phi = (this->phi <= 360) ? this->phi : this->phi - 360;
    return *this;
}

cPoint cPoint::operator+= (double val) {
    this->ro += val;
    return *this;
}

bool operator== (cPoint& one, cPoint& other) {
    return ((one.ro == other.ro) && (one.phi == other.phi) && (one.z == other.z));
}

vector<string> splitCoords(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;
}

```

Тестування



```
Консоль отладки Microsoft Visual Studio
1) Введіть координати:  rho: 12.5
                        phi: 60
                        z: 4

P1 (12.500000 60 4.000000)
2) Введіть координати через пробіл:11.1 45 23
P2 (11.100000 45 23.000000)
P3 (12.500000 60 4.000000)
інкрементуємо P1:
P1 (12.500000 61 4.000000)
Введіть значення: 4

15.100000 45 23.000000
P1 рівна P2? 0
Відстань від точки P3: 13.1244

C:\Users\demya\source\repos\lab3cpp\x64\Debug\lab3cpp.exe (процесс 11824) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Висновки:

На роботі вивчив і використав на практиці механізми створення і використання класів з використанням перевантажених операторів (операцій), використав різні види конструкторів, та використовував геттери.