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 22

Виконав студент _____ІП-15_Мєшков_Андрій_Ігорович_____

(шифр, прізвище, ім'я, по батькові)

Перевірив _____Вечерковська Анастасія Сергіївна_____
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 4
Перевантаження операторів

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

Варіант 22

Завдання.

22. Розробити клас "Вектор у просторі", який задається координатами його кінця. Реалізувати для нього декілька конструкторів, геттери, метод обчислення довжини вектора. Перевантажити оператори "+" та "*" для знаходження суми і скалярного добутку векторів відповідно. Створити три вектори (M1, M2, M3), використовуючи різні конструктори. Визначити вектор M3 як суму векторів M1 та M2. Знайти довжину вектору M3, а також скалярний добуток векторів M1 та M3.

Код C++
lab4.cpp

```cpp
#include <iostream>

#include "function.hpp"
using namespace std;
int main() {

    VectorInSpace M1(0), M2(0), M3(0);

    M1=new_vector1();
    cout<<"M1 = ";
    M1.getVector();

    M2=new_vector2();
    cout<<"M2 = ";
    M2.getVector();

    M3=M1+M2;
    cout<<"M3 = M1 + M2 = ";
    M3.getVector();

    cout<<"|M3| = "<< M3.module()<<endl<<endl;

    cout<<"M1 * M2 = "<<M1*M2<<endl<<endl;

}

```

function.hpp

```hpp
#ifndef function_hpp
#define function_hpp
#include "class.hpp"
#include <stdio.h>
using namespace std;
VectorInSpace new_vector1();
VectorInSpace new_vector2();

#endif

```

function.cpp

```cpp
#include "function.hpp"
#include <iostream>
#include <string>
using namespace std;

VectorInSpace new_vector1(){
    double x,y,z;
    cout<<"M1"<<endl;
    cout<< "x = ";cin>>x;
    cout<< "y = ";cin>>y;
    cout<< "z = ";cin>>z;
    VectorInSpace v(x,y,z);
    return v;
}
VectorInSpace new_vector2(){
    cin.ignore();
    string vector;
    cout<<"M2"<<endl;
    cout<< "(x, y, z) = ";
    getline(cin, vector);

    VectorInSpace v(vector);
    return v;
}

```

class.hpp

```cpp
#ifndef class_hpp
#define class_hpp
#include <iostream>
#include <stdio.h>
#include <string>
using namespace std;

class VectorInSpace{
    double x, y, z;
public:
    VectorInSpace(){}
    VectorInSpace(double xx=0, double yy=0, double zz=0): x(xx), y(yy), z(zz){}
    VectorInSpace(string vector){
        int num1 = vector.find(', ');
        int num2 = vector.rfind(', ');
        x=atof(vector.substr(0, num1).c_str());
        y=atof(vector.substr(num1+1, num2).c_str());
        z=atof(vector.substr(num2+1).c_str());
    }
    VectorInSpace(VectorInSpace& obj){
        x=obj.x;
        y=obj.y;
        z=obj.z;
    }
    void getVector();
    double getX(){return x;}
    double getY(){return y;}
    double getZ(){return z;}
    double module();
    VectorInSpace operator+(const VectorInSpace obj);
    double operator*(const VectorInSpace obj);
    ~VectorInSpace(){}


};

#endif

```

class.cpp

```cpp
#include "class.hpp"
#include "cmath"
using namespace std;


void VectorInSpace::getVector(){
    cout<<"("<<x<<", "<< y<< ", "<<z<<")"<<endl<<endl;
}

double VectorInSpace::module(){
    return sqrt(x*x + y*y + z*z);
}

VectorInSpace VectorInSpace::operator+(const VectorInSpace obj){
    VectorInSpace tmp(0);
    tmp.x = x + obj.x;
    tmp.y = y + obj.y;
    tmp.z = z + obj.z;
    return tmp;
}
double VectorInSpace::operator*(const VectorInSpace obj){
    return x * obj.x + y * obj.y + z * obj.z;
}


```

Результат
C++

```
M1
x = -3
y = 4.24
z = 5
M1 = (-3, 4.24, 5)

M2
(x, y, z) = -1.23, 0, 10
M2 = (-1.23, 0, 10)

M3 = M1 + M2 = (-4.23, 4.24, 15)

|M3| = 16.1515

M1 * M2 = 53.69

Program ended with exit code: 0
```