**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Бирюков Виктор Владимирович

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи. Вариант 11

Создать класс vector3D, задаваемый тройкой координат. Обязательно должны быть реализованы: операции сложения и вычитания векторов, векторное произведение векторов, скалярное произведение векторов, умножения на скаляр, сравнение векторов на совпадение, вычисление длины вектора, сравнение длины векторов, вычисление угла между векторами.

1. Описание программы

В программе реализован класс vector3D, задающий вектор в трехмерном пространстве при помощи координат - трех целых чисел. Для взаимодействия с вектором реализованы следующие методы:

* add - складывает переданный вектор с текущим, возвращает результат
* sub - вычитает переданный вектор из текущего, возвращает результат
* mul - умножает вектор на целое число, возвращает результат
* dot - скалярное произведение переданного вектора с текущим
* cross - векторное произведение переданного вектора с текущим
* equal - сравнивает переданный вектор с текущим покоординатно
* len - длина вектора
* len\_eq - сравнивает переданный вектор с текущим по длине
* angle - вычисляет угол (в радианах) между переданным вектором и текущим

Также в классе описаны конструкторы: конструктор по умолчанию, создающий вектор с нулевыми координатами; конструктор принимающий значения координат; конструктор копирования.

Программа получает на вход координаты векторов a и b в виде трех целых чисел, разделенных пробелом, и целое число n. Программа выводит результаты следующих операций: сложения a и b, вычитания b из a, умножения b на число n, скалярного произведения a и b, векторного произведения a и b, сравнения a и b, длину a, длину b, сравнения длин a и b, угол между a и b.

1. Набор тестов

Тест 1:

4 20 7

16 9 1

6

Тест 2:

3 5 9

3 5 9

40

Тест 3:

12 -9 5

9 -5 -12

123

Тест 4:

56 90 3

0 0 0

451

1. Результаты выполнения тестов

Тест 1:

a + b = (20, 29, 8)

a - b = (-12, 11, 6)

b \* n = (96, 54, 6)

(a, b) = 251

[a, b] = (-43, 108, -284)

(a == b) = 0

|a| = 21.563859

|b| = 18.384776

(|a| == |b|) = 0

angle(a, b) = 0.885214 rad

Тест 2:

a + b = (6, 10, 18)

a - b = (0, 0, 0)

b \* n = (120, 200, 360)

(a, b) = 115

[a, b] = (0, 0, 0)

(a == b) = 1

|a| = 10.723805

|b| = 10.723805

(|a| == |b|) = 1

angle(a, b) = 0.000000 rad

Тест 3:

a + b = (21, -14, -7)

a - b = (3, -4, 17)

b \* n = (1107, -615, -1476)

(a, b) = 93

[a, b] = (133, 189, 21)

(a == b) = 0

|a| = 15.811388

|b| = 15.811388

(|a| == |b|) = 1

angle(a, b) = 1.189634 rad

Тест 4:

a + b = (56, 90, 3)

a - b = (56, 90, 3)

b \* n = (0, 0, 0)

(a, b) = 0

[a, b] = (0, 0, 0)

(a == b) = 0

|a| = 106.042444

|b| = 0.000000

(|a| == |b|) = 0

angle(a, b) = 0.000000 rad

1. Листинг программы

// Бирюков М8О-207Б-19 вариант 11

#include <cmath>

#include <cstdio>

using namespace std;

class vector3D {

public:

int x;

int y;

int z;

vector3D(int \_x, int \_y, int \_z) {

x = \_x;

y = \_y;

z = \_z;

}

vector3D(): vector3D(0, 0, 0) {};

vector3D(const vector3D &v): vector3D(v.x, v.y, v.z) {};

vector3D add(vector3D a) {

return vector3D(x + a.x, y + a.y, z + a.z);

}

vector3D sub(vector3D a) {

return vector3D(x - a.x, y - a.y, z - a.z);

}

vector3D mul(int n) {

return vector3D(x \* n, y \* n, z \* n);

}

int dot(vector3D a) {

return x\*a.x + y\*a.y + z\*a.z;

}

vector3D cross(vector3D a) {

return vector3D(y\*a.z - z\*a.y, z\*a.x - x\*a.z, x\*a.y - y\*a.x);

}

bool equal(vector3D a) {

return x == a.x && y == a.y && z == a.z;

}

double len() {

return sqrt(this->dot(\*this));

}

bool len\_eq(vector3D a) {

return this->len() == a.len();

}

double angle(vector3D a) {

if (this->len() == 0 || a.len() == 0) return 0;

return acos(this->dot(a) / (this->len() \* a.len()));

}

};

int main() {

vector3D a;

printf("a = ");

scanf("%d %d %d", &a.x, &a.y, &a.z);

int x, y, z;

printf("b = ");

scanf("%d %d %d", &x, &y, &z);

vector3D b(x, y, z);

int n;

printf("n = ");

scanf("%d", &n);

vector3D c(a.add(b));

printf("a + b = (%d, %d, %d)\n", c.x, c.y, c.z);

c = a.sub(b);

printf("a - b = (%d, %d, %d)\n", c.x, c.y, c.z);

c = b.mul(n);

printf("b \* n = (%d, %d, %d)\n", c.x, c.y, c.z);

printf("(a, b) = %d\n", a.dot(b));

c = a.cross(b);

printf("[a, b] = (%d, %d, %d)\n", c.x, c.y, c.z);

printf("(a == b) = %d\n", a.equal(b));

printf("|a| = %lf\n", a.len());

printf("|b| = %lf\n", b.len());

printf("(|a| == |b|) = %d\n", a.len\_eq(b));

printf("angle(a, b) = %lf rad\n", a.angle(b));

}

1. Выводы

В ходе лабораторной работы получены навыки создания простого класса на языке C++, реализация методов и конструкторов.

1. Литература

1 Справочник по языку C++ [Электронный ресурс]. URL: https://ru.cppreference.com