**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: Цыкин Иван

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата: 23.09.2020

Оценка:

Москва, 2020

1.     Постановка задачи:

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.

2. Описание программы:

Программа состоит из 4 файлов: 1 – библиотека Vector3D.hpp, в которой описывается класс; 2 – описание функций в файле Vector3D.cpp, 3 – main.cpp – непосредственно файл с программой, 4 – Cmakefile.txt связующий файл.

3. Набор тестов

|  |  |  |
| --- | --- | --- |
|  | Входные данные\*: | Выходные данные\*\*: |
| тест 1 | 1  1  2  1  X: 2  Y: 4  Z: 6  1  1  3  1  4  1  2  2  2  X: 0  Y: 0  Z: 4  1  2  1  1  5  6  7  8  9  10  11  0 | ВЕКТОР: {0,0,0}  ВЕКТОР: {2,4,6}  ВЕКТОР: {4,8,12}  ВЕКТОР: {0,0,4}  ВЕКТОР: {2,4,6}  ВЕКТОР: {2,4,10}  ВЕКТОР: {2,4,2}  ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ: 8  СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ: 24  НЕ СОВПАДАЮТ  ДЛИНА ГЛАВНОГО ВЕКТОРА БОЛЬШЕ  УГОЛ МЕЖДУ ВЕКТОРАМИ: 36.6992 |
| тест 2 | 2  1  X: 1  Y: 1  Z: 0  2  2  X: 0  Y: 0  Z: 1  8  7  11  2  1  X: 1  Y: 1  Z: 1  2  2  X: 2  Y: 2  Z: 2  9  10  2  1  X: 2  Y: 2  Z: 2  10  0 | СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ: 0  ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ: 0  УГОЛ МЕЖДУ ВЕКТОРАМИ: 90  2 ВЕКТОРА СОВПАДАЮТ  ДЛИНА ГЛАВНОГО ВЕКТОРА МЕНЬШЕ  ДЛИНЫ РАВНЫ |

4. Результаты выполнения тестов.

**Тест 1:**

Windows PowerShell

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\itsyk\.vscode> & 'c:\Users\itsyk\.vscode\extensions\ms-vscode.cpptools-1.0.1\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-i0askc3n.zsr' '--stdout=Microsoft-MIEngine-Out-adponyfp.v0x' '--stderr=Microsoft-MIEngine-Error-rlsosizn.fnv' '--pid=Microsoft-MIEngine-Pid-knw5nmcq.hwm' '--dbgExe=C:\MinGW\mingw32\bin\gdb.exe' '--interpreter=mi'

БЫЛ СОЗДАН ВЕКТОР

БЫЛ СОЗДАН ВЕКТОР

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| |

| МЕНЮ |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

| |

|1 - ВЫВЕСТИ ВЕКТОР |

|2 - ИЗМЕНИТЬ ВЕКТОР |

|3 - ДЛИННА ВЕКТОРА |

|4 - УМНОЖИТЬ НА ЧИСЛО |

|5 - СЛОЖЕНИЕ 2 ВЕКТОРОВ |

|6 - ВЫЧИТАНИЕ 2 ВЕКТОРОВ |

|7 - ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ |

|8 - СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ |

|9 - СРАВНЕНИЕ НА СОВПАДЕНИЕ|

|10 - СРАВНЕНИЕ ДЛИН ВЕКТОРОВ|

|11 - УГОЛ МЕЖДУ ВЕКТОРАМИ |

|0 - КОНЕЦ |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

1

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ВЕКТОР: {0,0,0}

2

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА

X: 2

Y: 4

Z: 6

1

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ВЕКТОР: {2,4,6}

3

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ДЛИНА ВЕКТОРА: 7.48331

4

ВЫБИРИТЕ ВЕКТОР(1 или 2):

2

ВВЕДИТЕ ЧИСЛО: 1

ВЕКТОР: {0,0,0}

4

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ВВЕДИТЕ ЧИСЛО: 2

ВЕКТОР: {4,8,12}

2

ВЫБИРИТЕ ВЕКТОР(1 или 2):

2

ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА

X: 0

Y: 0

Z: 4

1

ВЫБИРИТЕ ВЕКТОР(1 или 2):

2

ВЕКТОР: {0,0,4}

1

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

5

ВЕКТОР: {2,4,10}

6

ВЕКТОР: {2,4,2}

7

ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ: 8

8

СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ: 24

9

НЕ СОВПАДАЮТ

10

ДЛИНА ГЛАВНОГО ВЕКТОРА БОЛЬШЕ

11

УГОЛ МЕЖДУ ВЕКТОРАМИ: 36.6992

0

**Тест 2:**

PS C:\Users\itsyk\.vscode> & 'c:\Users\itsyk\.vscode\extensions\ms-vscode.cpptools-1.0.1\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-4wf1ueqv.veu' '--stdout=Microsoft-MIEngine-Out-1xruvb5t.d30' '--stderr=Microsoft-MIEngine-Error-ezdlrrze.eeg' '--pid=Microsoft-MIEngine-Pid-wo3qwbo3.qmi' '--dbgExe=C:\MinGW\mingw32\bin\gdb.exe' '--interpreter=mi'

БЫЛ СОЗДАН ВЕКТОР

БЫЛ СОЗДАН ВЕКТОР

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| |

| МЕНЮ |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

| |

|1 - ВЫВЕСТИ ВЕКТОР |

|2 - ИЗМЕНИТЬ ВЕКТОР |

|3 - ДЛИННА ВЕКТОРА |

|4 - УМНОЖИТЬ НА ЧИСЛО |

|5 - СЛОЖЕНИЕ 2 ВЕКТОРОВ |

|6 - ВЫЧИТАНИЕ 2 ВЕКТОРОВ |

|7 - ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ |

|8 - СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ |

|9 - СРАВНЕНИЕ НА СОВПАДЕНИЕ|

|10 - СРАВНЕНИЕ ДЛИН ВЕКТОРОВ|

|11 - УГОЛ МЕЖДУ ВЕКТОРАМИ |

|0 - КОНЕЦ |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

2

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА

X: 1

Y: 1

Z: 0

2

ВЫБИРИТЕ ВЕКТОР(1 или 2):

2

ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА

X: 0

Y: 0

Z: 1

8

СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ: 0

7

ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ: 0

11

УГОЛ МЕЖДУ ВЕКТОРАМИ: 90

2

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА

X: 1

Y: 1

Z: 1

2

ВЫБИРИТЕ ВЕКТОР(1 или 2):

2

ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА

X: 2

Y: 2

Z: 2

9

2 ВЕКТОРА СОВПАДАЮТ

10

ДЛИНА ГЛАВНОГО ВЕКТОРА МЕНЬШЕ

2

ВЫБИРИТЕ ВЕКТОР(1 или 2):

1

ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА

X: 2

Y: 2

Z: 2

10

ДЛИНЫ РАВНЫ

0

5. Листинг программы:

Vector3D.hpp

#ifndef Vector3D\_hpp

#define Vector3D\_hpp

#include <iostream>

using namespace std;

class Vector3D{

    public:

        double x, y, z;

        Vector3D(){

            x = 0;

            y = 0;

            z = 0;

            cout << "БЫЛ СОЗДАН ВЕКТОР" << endl;

        }

        void add();

        void print\_vec() const;

        void func\_l() const;

        void sum(Vector3D v);

        void dif(Vector3D v);

        void mult\_a(double a);

        void vec\_pro(Vector3D v) const;

        void scalar\_pro(Vector3D v) const;

        void comp(Vector3D v) const;

        void comp\_l(Vector3D v) const;

        void angle(Vector3D v) const;

};

#endif //Vector3D\_hpp

Vector3D.cpp

#include <iostream>

#include <cmath>

#include "Vector3D.hpp"

using namespace std;

#define Pi 3.14159265358

void Vector3D::add(){

    cout << "ВВЕДИТЕ КООРДИНАТЫ ВЕКТОРА" << endl;

        cout << "X: ";

        cin >> x;

        cout << "Y: ";

        cin >> y;

        cout << "Z: ";

        cin >> z;

}

void Vector3D::print\_vec() const {

        cout << "ВЕКТОР: {" << x << "," << y << "," << z << "}" << endl;

    }

void Vector3D::func\_l() const{

        cout << "ДЛИНА ВЕКТОРА: " << sqrt((x\*x)+(y\*y)+(z\*z)) << endl;

    }

void Vector3D::sum(Vector3D v){

        cout << "ВЕКТОР: {" << x+v.x << "," << y+v.y << "," << z+v.z << "}" << endl;

    }

void Vector3D::dif(Vector3D v){

        cout << "ВЕКТОР: {" << x-v.x << "," << y-v.y << "," << z-v.z << "}" << endl;

    }

void Vector3D::mult\_a(double a){

        cout << "ВЕКТОР: {" << x\*a << "," << y\*a << "," << z\*a << "}" << endl;

    }

void Vector3D::vec\_pro(Vector3D v) const{

        cout << "ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ: " << ((y \* v.z - z \* v.y) - (x \* v.z - z \* v.x) + (x \* v.y - y \* v.x)) << endl;

    }

void Vector3D::scalar\_pro(Vector3D v) const{

        cout << "СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ: " << (x \* v.x + y \* v.y + z \* v.z) << endl;

    }

void Vector3D::comp(Vector3D v) const{

        double n = x/v.x;

        if (y/v.y == n && z/v.z == n){

            cout << "2 ВЕКТОРА СОВПАДАЮТ" << endl;

        } else{

            cout << "НЕ СОВПАДАЮТ" << endl;

        }

    }

void Vector3D::comp\_l(Vector3D v) const{

        if (sqrt((x\*x)+(y\*y)+(z\*z)) > sqrt((v.x\*v.x)+(v.y\*v.y)+(v.z\*v.z))){

            cout << "ДЛИНА ГЛАВНОГО ВЕКТОРА БОЛЬШЕ" << endl;

        } else if(sqrt((x\*x)+(y\*y)+(z\*z)) < sqrt((v.x\*v.x)+(v.y\*v.y)+(v.z\*v.z))){

            cout << "ДЛИНА ГЛАВНОГО ВЕКТОРА МЕНЬШЕ" << endl;

        } else if(sqrt((x\*x)+(y\*y)+(z\*z)) == sqrt((v.x\*v.x)+(v.y\*v.y)+(v.z\*v.z)){

            cout << "ДЛИНЫ РАВНЫ" << endl;

        }

    }

void Vector3D::angle(Vector3D v) const{

        double a = (x \* v.x + y \* v.y + z \* v.z)/(sqrt((x\*x)+(y\*y)+(z\*z))\*sqrt((v.x\*v.x)+(v.y\*v.y)+(v.z\*v.z)));

        cout << "УГОЛ МЕЖДУ ВЕКТОРАМИ: " << acos(a)\*180/Pi << endl;

    }

main.cpp

#include <iostream>

#include "Vector3D.cpp"

using namespace std;

void print\_menu(){

    cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    cout << "|                            |" << endl;

    cout << "|            МЕНЮ            |" << endl;

    cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

    cout << "|                            |" << endl;

    cout << "|1  - ВЫВЕСТИ ВЕКТОР         |" << endl;

    cout << "|2  - ИЗМЕНИТЬ ВЕКТОР        |" << endl;

    cout << "|3  - ДЛИННА ВЕКТОРА         |" << endl;

    cout << "|4  - УМНОЖИТЬ НА ЧИСЛО      |" << endl;

    cout << "|5  - СЛОЖЕНИЕ 2 ВЕКТОРОВ    |" << endl;

    cout << "|6  - ВЫЧИТАНИЕ 2 ВЕКТОРОВ   |" << endl;

    cout << "|7  - ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ |" << endl;

    cout << "|8  - СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ |" << endl;

    cout << "|9  - СРАВНЕНИЕ НА СОВПАДЕНИЕ|" << endl;

    cout << "|10 - СРАВНЕНИЕ ДЛИН ВЕКТОРОВ|" << endl;

    cout << "|11 - УГОЛ МЕЖДУ ВЕКТОРАМИ   |" << endl;

    cout << "|0  - КОНЕЦ                  |" << endl;

    cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

}

int main(){

    int n;

    setlocale(LC\_ALL, "Rus");

    Vector3D v1{};

    Vector3D v2{};

    int var;

    print\_menu();

    for(;;){

        cin >> var;

        switch(var){

            case 1:

                cout <<"ВЫБИРИТЕ ВЕКТОР(1 или 2): "<< endl;

                cin >> n;

                if(n == 1){

                    v1.print\_vec();

                    break;

                }else if(n == 2){

                    v2.print\_vec();

                    break;

                } else{

                    cout << "НЕВЕРНО ВВЕДЕН НОМЕР ВЕКТОРА!" << endl;

                    break;

                }

            case 2:

                cout <<"ВЫБИРИТЕ ВЕКТОР(1 или 2): "<< endl;

                cin >> n;

                if(n == 1){

                    v1.add();

                    break;

                }else if(n == 2){

                    v2.add();

                    break;

                } else{

                    cout << "НЕВЕРНО ВВЕДЕН НОМЕР ВЕКТОРА!" << endl;

                    break;

                }

            case 3:

                cout <<"ВЫБИРИТЕ ВЕКТОР(1 или 2): "<< endl;

                cin >> n;

                if(n == 1){

                    v1.func\_l();

                    break;

                }else if(n == 2){

                    v2.func\_l();

                    break;

                } else{

                    cout << "НЕВЕРНО ВВЕДЕН НОМЕР ВЕКТОРА!" << endl;

                    break;

                }

            case 4:

                double a;

                cout <<"ВЫБИРИТЕ ВЕКТОР(1 или 2): "<< endl;

                cin >> n;

                if(n == 1){

                    cout << "ВВЕДИТЕ ЧИСЛО: ";

                    cin >> a;

                    v1.mult\_a(a);

                    break;

                }else if(n == 2){

                    cout << "ВВЕДИТЕ ЧИСЛО: ";

                    cin >> a;

                    v2.mult\_a(a);

                    break;

                } else{

                    cout << "НЕВЕРНО ВВЕДЕН НОМЕР ВЕКТОРА!" << endl;

                    break;

                }

            case 5:

                v1.sum(v2);

                break;

            case 6:

                v1.dif(v2);

                break;

            case 7:

                v1.vec\_pro(v2);

                break;

            case 8:

                v1.scalar\_pro(v2);

                break;

            case 9:

                v1.comp(v2);

                break;

            case 10:

                v1.comp\_l(v2);

                break;

            case 11:

                v1.angle(v2);

                break;

            case 0:

                return 0;

            default:

                cout << "ОШИБКА, ПОПРОБУЙТЕ ЕЩЕ РАЗ" << endl;

                break;

        }

    }

}

6. Вывод:

Во время лабораторной работы мной были структура класса и его методы, изучена кроссплатформенная система автоматизации сборки программного обеспечения из исходного кода. C помощью полученных знаний составил программу, которая решает поставленную задачу.

7. Список литературы.

1. Побитовые операции С++ [Электронный ресурс]. URL: https://ravesli.com/urok-45-pobitovye-operatory