

**Министерство науки и высшего образования  
Российской Федерации**

**Федеральное государственное автономное  
образовательное учреждение высшего образования**

**«Национальный исследовательский университет  
ИТМО»**

**Факультет информационных технологий и  
программирования**

Лабораторная работа №2

Вариант 3

*Классы*

**Выполнил студент группы № М3111  
Соловьев Михаил Александрович.**

Санкт-Петербург  
2023

В данном задании необходимо согласно варианту, описать указанные классы (вектор в  $R^3$ ). Написать программу, использующую описанные классы: инициализация переменных, выполнение действий с экземплярами класса (в зависимости от дальнейшего ввода пользователя)

Описание и реализация должны находиться в разных файлах. Доступ к полям класса — только через методы. Внешние функции для работы с данными класса не допускаются.

Код:

```
class.cpp
#include <iostream>
#include <cmath>
#include "class.h"

Vector::Vector() {                // default constructor
    x = 0;
    y = 0;
    z = 0;
}

Vector::Vector(double x_, double y_, double z_) { // float constructor
    x = x_;
    y = y_;
    z = z_;
}

Vector::Vector(const Vector &v) {    // copy constructor
    x = v.x;
    y = v.y;
    z = v.z;
}

double Vector::GetX() const {        // Getters
    return x;
}

double Vector::GetY() const {
    return y;
}

double Vector::GetZ() const {
    return z;
}

void Vector::SetX(double x_) {        // Setters
    x = x_;
}

void Vector::SetY(double y_) {
```

```

    y = y_;
}

void Vector::SetZ(double z_) {
    z = z_;
}

Vector Vector::add(const Vector &other) const { // Functionality
    return {x + other.x, y + other.y, z + other.z};
}

Vector Vector::sub(const Vector &other) const {
    return {x - other.x, y - other.y, z - other.z};
}

double Vector::len() const {
    return (sqrt((pow(x, 2) + pow(y, 2) + pow(z, 2))));
}

double Vector::angle(const Vector &other) const {
    return acos((x * other.x + y * other.y + z * other.z) /
                ((sqrt((pow(x, 2) + pow(y, 2) + pow(z, 2)))) *
                 sqrt((pow(other.x, 2) + pow(other.y, 2) + pow(other.z, 2)))))
}

void Vector::print() const {
    std::cout << "Vector: (" << x << ", " << y << ", " << z << ")\n";
}

```

```

class.h
#ifndef COURSE_C__CLASS_H
#define COURSE_C__CLASS_H

class Vector {
private:

public:
    double x;
    double y;
    double z;

    Vector();
    Vector(double x_, double y_, double z_);
    Vector(const Vector &v);

    double GetX() const;
    double GetY() const;
    double GetZ() const;

```

```

void SetX(double x_);
void SetY(double y_);
void SetZ(double z_);

Vector add(const Vector &other) const;
Vector sub(const Vector &other) const;
double len() const;
double angle(const Vector &other) const;
void print() const;
};

#endif //COURSE_C___CLASS_H

main.cpp
#include <iostream>
#include "class.h"
#include "menu.h"

int main() {
    std::string command;
    Vector vector(1, 2, 3);

    std::cout << "1 - create \n2 - length \n3 - add \n4 - sub \n5 - angle \n6 - print \n";

    while (true) {
        std::cin >> command;
        Menu menu;

        if (command == "1") {
            menu.create(vector);
            continue;
        }
        else if (command == "2") {
            menu.len(vector);
            continue;
        }
        else if (command == "3") {
            menu.sum(vector);
            continue;
        }
        else if (command == "4") {
            menu.sub(vector);
            continue;
        }
        else if (command == "5") {
            menu.sub(vector);
            continue;
        }
    }
}

```

```

        else if (command == "6") {
            menu.output(vector);
            continue;
        }
    }
}

```

menu.cpp

```

#include <iostream>
#include "class.h"
#include "menu.h"

```

```

void Menu::create(Vector &vector) {
    std::cout << "Input x, y, z \n";
    int x, y, z;
    std::cin >> x >> y >> z;
    Vector vector_new(x, y, z);
    vector = vector_new;
    std::cout << "Vector created! \n";
}

```

```

void Menu::sum(Vector &vector) {
    std::cout << "Input x, y, z \n";
    int x, y, z;
    std::cin >> x >> y >> z;
    Vector vector_to_sum(x, y, z);
    vector = vector.add(vector_to_sum);
    std::cout << "Operation complete!" << "\n";
}

```

```

void Menu::sub(Vector &vector) {
    std::cout << "Input x, y, z \n";
    int x, y, z;
    std::cin >> x >> y >> z;
    Vector vector_to_sub(x, y, z);
    vector = vector.sub(vector_to_sub);
    std::cout << "Operation complete!" << "\n";
}

```

```

void Menu::angle(Vector &vector) {
    std::cout << "Input x, y, z \n";
    int x, y, z;
    std::cin >> x >> y >> z;
    Vector angle_vector(x, y, z);
    std::cout << "Vector angle: " << vector.angle(angle_vector) << "\n";
}

```

```

void Menu::len(Vector &vector) {

```

```
std::cout << "Vector length: " << vector.len() << "\n";  
}
```

```
void Menu::output(Vector &vector) {  
    vector.print();  
}
```

menu.h

```
#ifndef COURSE_C__MENU_H  
#define COURSE_C__MENU_H
```

```
#include <iostream>
```

```
class Menu {  
private:
```

```
public:
```

```
    void create(Vector &vector);  
    void sum(Vector &vector);  
    void sub(Vector &vector);  
    void angle(Vector &vector);  
    void output(Vector &vector);  
    void len(Vector &vector);  
};
```

```
#endif
```