МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Разработка классов на С++.

Дисциплина: «Объектно-ориентированное программирование»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

|  |  |
| --- | --- |
| Руководитель | Сорокин А.Н. |
| Выполнили студенты | Пчелкина О.С. |
| Группа, курс | ВМ-41 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

2022 г.

**Цель работы**: научиться разрабатывать классы на языке С++.

# Вариант 1

1. Разработать класс для работы с многомерными векторами. Класс должен содержать следующие элементы:

Поля (скрытые):

размерность вектора

указатель на компоненты вектора

Методы:

конструктор по умолчанию, задающий 3-мерный нулевой вектор

конструктор, формирующий нулевой вектор заданной размерности

конструктор-копировщик

деструктор

метод для задания размерности вектора

метод для задания отдельных компонентов вектора

методы для ввода вектора с клавиатуры

метод для вывода координат вектора на экран в общепринятом формате «(X1,X2,…)»

метод-функция для вычисления длины вектора

метод для нормализации вектора

2. Реализовать для разработанного класса методы-операторы:

+, - (плюс и минус**)**  - для сложения и вычитания двух векторов

\* (умножить) – для нахождения скалярного произведения двух векторов

== (два знака равенства подряд) - для сравнения двух векторов, результатом этой операции должен быть 0, если два вектора неравны, и любое ненулевое значение в противном случае

= (знак равенства) - присваивание

3. Реализовать класс в виде шаблона, позволяющего хранить данные различных типов.

4. Разработать программу, которая демонстрировала бы работоспособность всех методов данного класса.

Код программы:

#include<iostream>

using namespace std;

const float EPSILON = 0.001f;

template <class T>

class Vector

{

public:

Vector();

Vector(Vector& v);

Vector(int size);

~Vector();

void SetSize(int size);

void SetComponent(int num, T component);

void InputVector();

void OutputVector();

float Magnitude();

void Normalize();

Vector operator+(Vector& v);

Vector operator\* (Vector& v);

Vector<T>& operator= (Vector& v);

bool operator== (Vector& v);

private:

int \_size;

T\* \_point;

};

template<typename T>

Vector<T>::Vector()

{

\_point = new T[3];

\_size = 3;

\_point[0] = 0;

\_point[1] = 0;

\_point[2] = 0;

}

template<typename T>

Vector<T>::Vector(Vector& v)

{

\_size = v.\_size;

T\* point = new T[\_size];

for (int i = 0; i < \_size; i++)

point[i] = v.\_point[i];

\_point = point;

}

template<typename T>

Vector<T>::Vector(int size)

{

\_size = size;

\_point = new T[size];

for (int i = 0; i < size; i++)

\_point[i] = 0;

}

template<typename T>

Vector<T>::~Vector()

{

delete \_point;

}

template<typename T>

void Vector<T>:: SetSize(int size)

{

T\* point = new T[size];

for (int i = 0; i < size; i++)

{

if (i < \_size)

point[i] = \_point[i];

else point[i] = 0;

}

delete[] \_point;

\_size = size;

\_point = point;

}

template<typename T>

void Vector<T>::SetComponent(int num, T component)

{

if (num > \_size - 1) {

cout << "Индекс вне границ массива" << endl;

}

else

\_point[num] = component;

}

template<typename T>

void Vector<T>::InputVector()

{

for (int i = 0; i < \_size; i++)

cin >> \_point[i];

}

template<typename T>

void Vector<T>::OutputVector()

{

for (int i = 0; i < \_size; i++)

cout << \_point[i] << "\t";

}

template<typename T>

float Vector<T>:: Magnitude()

{

auto sum = 0;

for (int i = 0; i < \_size; i++)

sum += pow(\_point[i], 2);

return sqrt(sum);

}

template<typename T>

void Vector<T>:: Normalize()

{

float magnitude = Magnitude();

if (magnitude > 0)

{

float invertedMag = (float)1 / magnitude;

for (int i = 0; i < \_size; i++)

\_point[i] \*= invertedMag;

}

}

template<typename T>

Vector<T> Vector<T>::operator+ (Vector& v)

{

for (int i = 0; i < \_size; i++)

\_point[i] += v.\_point[i];

return \*this;

}

template<typename T>

Vector<T> Vector<T>:: operator\* (Vector& v)

{

for (int i = 0; i < \_size; i++)

\_point[i] \*= v.\_point[i];

return \*this;

}

template<typename T>

bool Vector<T>::operator== (Vector& v)

{

if (\_size != v.\_size)

return false;

for (int i = 0; i < \_size; i++)

{

if (fabs(\_point[i] - v.\_point[i]) > EPSILON)

return false;

}

return true;

}

template<typename T>

Vector<T>& Vector<T>:: operator= (Vector& v)

{

if (&\_point != &v.\_point)

{

\_size = v.\_size;

T\* point = new T[\_size];

for (int i = 0; i < \_size; i++)

{

point[i] = v.\_point[i];

}

delete[] \_point;

\_point = point;

}

return \*this;

}

Программа, демонстрирующая работу класса:

#include <iostream>

#include "Vector.cpp"

#include <vector>

using namespace std;

int main()

{

setlocale(LC\_ALL, "");

Vector<float> v;

cout << "Вектор по умолчанию:" << endl;

v.OutputVector();

cout << "\n" << endl;

Vector<float> v1(7);

cout << "Вектор заданной размерности:" << endl;

v1.OutputVector();

cout << "\n" << endl;

int size = 0;

cout << "Введите размерность вектора: ";

cin >> size;

v1.SetSize(size);

int num = 0;

int component = 0;

cout << "Введите номер компонента: ";

cin >> num;

cout << "Введите компонент: ";

cin >> component;

v1.SetComponent(num, component);

cout << "Новый вектор:" << endl;

v1.OutputVector();

cout << "\n" << endl;

cout << "Введите вектор:" << endl;

v1.InputVector();

cout << "Новый вектор:" << endl;

v1.OutputVector();

cout << "\n" << endl;

cout << "Длина вектора: ";

auto length = v1.Magnitude();

cout << length << endl;

cout << endl;

cout << "Вектор после нормализации:" << endl;

v1.Normalize();

v1.OutputVector();

cout << "\n" << endl;

Vector<float> v2(7);

cout << "Введите вектор:" << endl;

v2.InputVector();

cout << "Сложение векторов: v1 + v2 = v3" << endl;

cout << "v1: ";

v1.OutputVector();

cout << endl;

cout << "v2: ";

v2.OutputVector();

cout << endl;

cout << "v3: ";

v1 + v2;

v1.OutputVector();

cout << endl;

cout << "Умножение векторов: v1 + v2 = v3" << endl;

cout << "v1: ";

v1.OutputVector();

cout << endl;

cout << "v2: ";

v2.OutputVector();

cout << endl;

cout << "v3: ";

v1 \* v2;

v1.OutputVector();

cout << endl;

cout << "Сравнение векторов: v1 == v2" << endl;

cout << "v1: ";

v1.OutputVector();

cout << endl;

cout << "v2: ";

v2.OutputVector();

cout << endl;

bool flag = v1 == v2;

if (flag) cout << "Ответ: Равны" << endl;

else cout << "Ответ: Неравны" << endl;

cout << endl;

cout << "Приравнивание векторов: v1 = v2" << endl;

cout << "v1: ";

v1.OutputVector();

cout << endl;

cout << "v2: ";

v2.OutputVector();

cout << endl;

v1 = v2;

cout << "Вектор после приравнивания:" << endl;

cout << "v1: ";

v1.OutputVector();

cout << endl;

}

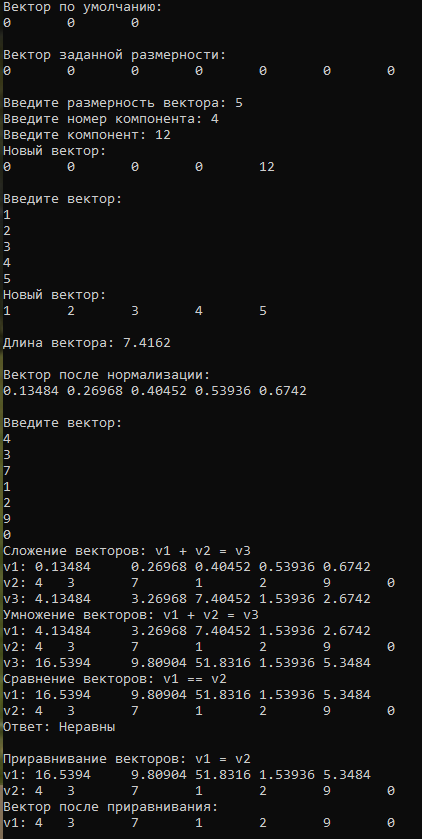


Рис. 1 – Работа программы

# Контрольные вопросы

1. Что такое «константные методы»? Какие методы разработанного класса можно сделать константными?

*Константный метод — это метод, который гарантирует, что не будет изменять объект или вызывать неконстантные методы класса.*

OutputVector(), Magnitude().

2. Что такое «статическое поле» и «статический метод» класса? Для чего применяются статические свойства класса?

*Статическое поле класса – это член статического класса, либо поле.*

*Статический метод класса – метод, не имеющий доступа к состоянию (полям) объекта, то есть к переменной this.*

*Объявление с помощью ключевого слова static, доступ к статическому полю/методу производится через имя класса, а не объекта.*

3.Что такое параметр типа для шаблона класса?

*Параметр типа – это имя типа*.

**Вывод:** в ходе лабораторной работы ознакомились с основными принципами ООП, научились создавать классы, изучили работу конструкторов, деструкторов, перегрузки операторов. Разработали класс на языке С++ и программу для демонстрации работы класса.