Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

# Отчёт по программе «Инициализация классов и конструкторы»

Выполнил: Плоских Александр Павлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание**

Описать класс «Векторы, задаваемые координатами концов в

трехмерном пространстве».

Необходимо создать следующие методы класса:

* операции сложения и вычитания векторов с получением нового

вектора;·

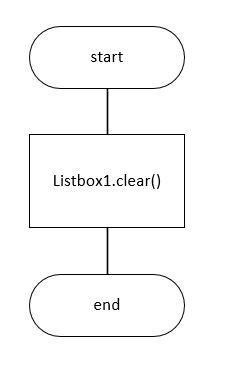
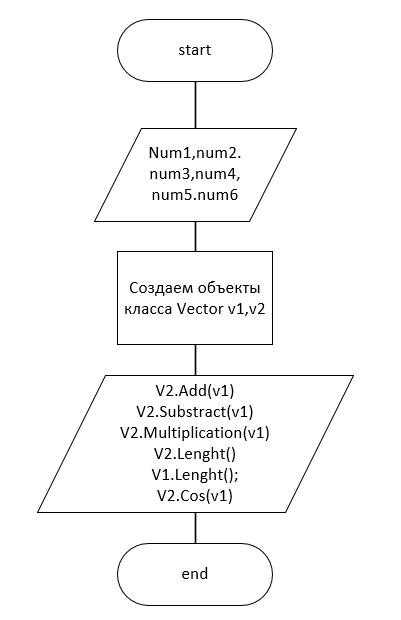
* вычисление скалярного произведения двух векторов и длины вектора;

операция вычисления косинуса угла между векторами.

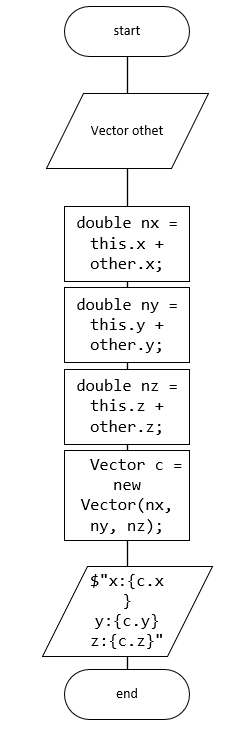
**Входные и выходные данные**

Свойства класса Vector

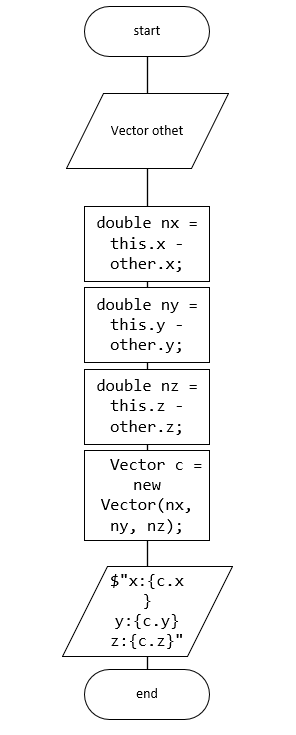
**Блок-схема**

****

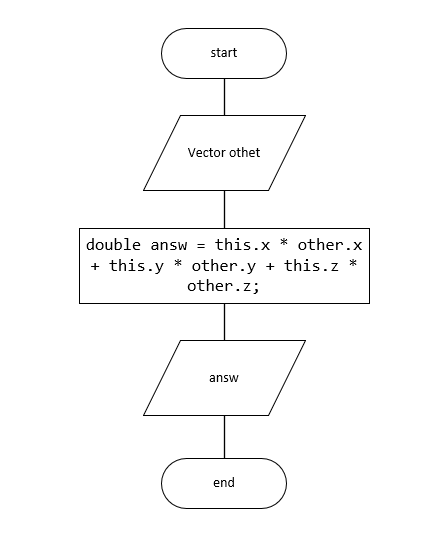
**Add**

****

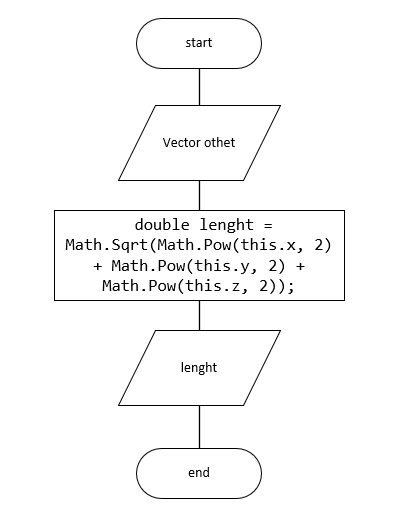
**Subsctract**

****

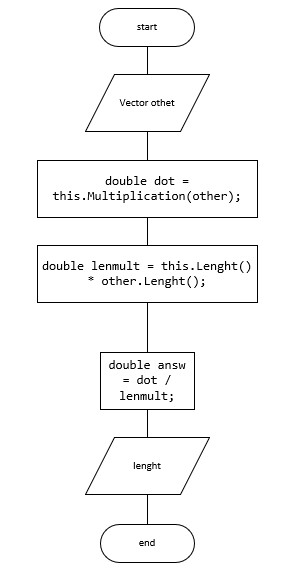
**Multiplication**

****

**lenght**

****

**Cos**

****

**Листинг программы**

*Основная программа*

private void button1\_Click(object sender, EventArgs e)

{

Vector v1 = new Vector((double)numericUpDown1.Value, (double)numericUpDown2.Value, (double)numericUpDown3.Value);

Vector v2 = new Vector((double)numericUpDown6.Value, (double)numericUpDown5.Value, (double)numericUpDown4.Value);

listBox1.Items.Add("add:");

listBox1.Items.Add(v2.Add(v1));

listBox1.Items.Add("subtract:");

listBox1.Items.Add(v2.Subtract(v1));

listBox1.Items.Add("Multiplication:");

listBox1.Items.Add(v2.Multiplication(v1));

listBox1.Items.Add("lenght v2:");

listBox1.Items.Add(v2.Lenght());

listBox1.Items.Add("lenght v1:");

listBox1.Items.Add(v1.Lenght());

listBox1.Items.Add("Cos:");

listBox1.Items.Add(v2.Cos(v1));

}

private void button2\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

}

*Класс*

class Vector

{

private double x;

private double y;

private double z;

public Vector(double a,double b,double c)

{

X = a;

Y = b;

Z = c;

}

public double X

{

get

{

return x;

}

set

{

x = value;

}

}

public double Y

{

get

{

return y;

}

set

{

y = value;

}

}

public double Z

{

get

{

return z;

}

set

{

z = value;

}

}

public string Add(Vector other)

{

double nx = this.x + other.x;

double ny = this.y + other.y;

double nz = this.z + other.z;

Vector c = new Vector(nx, ny, nz);

return $"x:{c.x} y:{c.y} z:{c.z}";

}

public string Subtract(Vector other)

{

double nx = this.x - other.x;

double ny = this.y - other.y;

double nz = this.z - other.z;

Vector c=new Vector(nx, ny, nz);

return $"x:{c.x} y:{c.y} z:{c.z}";

}

public double Multiplication(Vector other)

{

double answ = this.x \* other.x + this.y \* other.y + this.z \* other.z;

return answ;

}

public double Lenght()

{

double lenght = Math.Sqrt(Math.Pow(this.x, 2) + Math.Pow(this.y, 2) + Math.Pow(this.z, 2));

return lenght;

}

public double Cos(Vector other)

{

double dot = this.Multiplication(other);

double lenmult = this.Lenght() \* other.Lenght();

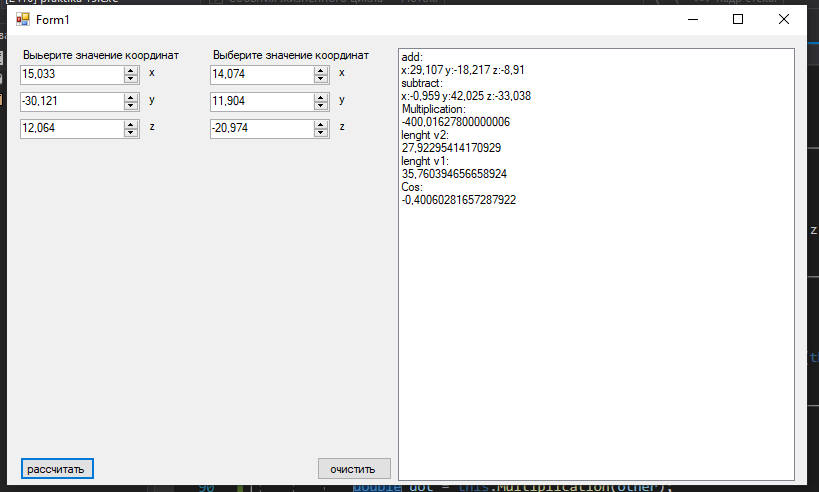
double answ = dot / lenmult;

return lenmult;

}

}

**Тестовые ситуации**

****