Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 15**»

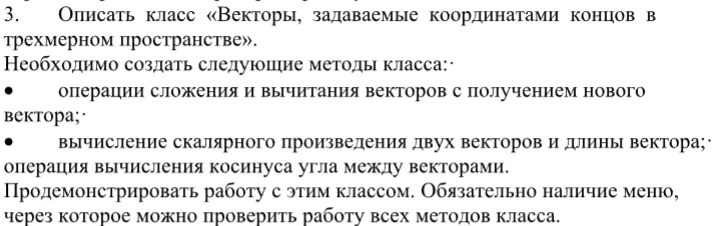
Выполнил: Сагутдинова Илона Ильнаровна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

****

**Входные данные:**

X1 – точка первого вектора, дробное число, double.

X2 – точка второго вектора, дробное число, double.

Y1 – точка первого вектора, дробное число, double.

Y2 – точка второго вектора, дробное число, double.

Z1 – точка первого вектора, дробное число, double.

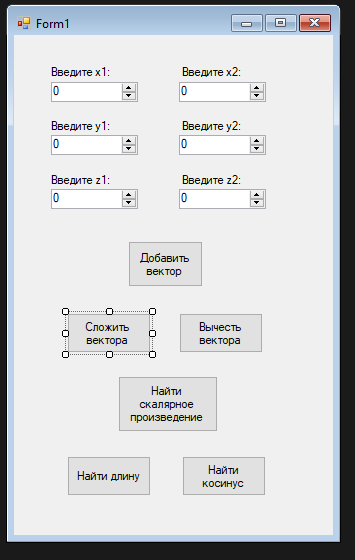
Z2 – точка второго вектора, дробное число, double.

**Выходные данные:**

Расчет суммы, разности, скалярного произведения, длины, косинуса угла между векторами.

**Блок-схема:**

**Листинг программы:**

****

internal class Vector

{

private double x1;

private double y1;

private double z1;

//открытие полей

public double GetX1()

{

return x1;

}

public void SetX1 (double x)

{

x1 = x;

}

public double GetY1()

{

return y1;

}

public void SetY1(double x)

{

y1 = x;

}

public double GetZ1()

{

return z1;

}

public void SetZ1(double x)

{

z1 = x;

}

//метод для вычисления суммы векторов

public string Sum (double x1, double x2, double y1, double y2, double z1, double z2)

{

double x3 = x1 + x2;

double y3 = y1 + y2;

double z3 = z1 + z2;

string rez = $"Вектор суммы: ({x3} : {y3} : {z3})";

return rez;

}

// метод для вычисления разности векторов

public string Razn(double x1, double x2, double y1, double y2, double z1, double z2)

{

double x3 = x1 - x2;

double y3 = y1 - y2;

double z3 = z1 - z2;

string rez = $"Вектор разности: ({x3} : {y3} : {z3})";

return rez;

}

// метод для вычисения скалярного произведения векторов

public double Scal (double x1, double x2, double y1, double y2, double z1, double z2)

{

double x3 = x1 \* x2;

double y3 = y1 \* y2;

double z3 = z1 \* z2;

double skal = x3 + y3 + z3;

return skal;

}

// метод для вычисления длины векторов

public double Length(double x1, double y1, double z1)

{

double leng = Math.Sqrt(Math.Pow(x1, 2) + Math.Pow(y1, 2) + Math.Pow(z1, 2));

double length = Math.Round(leng, 4);

return length;

}

// метод для вычисления косинуса между углами векторов

public double Cos(double x1, double x2, double y1, double y2, double z1, double z2)

{

double cos = Math.Round(Scal(x1, x2, y1, y2, z1, z2) / (Length(x1, y1, z1) \* Length(x2,y2,z2)), 4);

return cos;

}

}

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//Создание объектов

Vector v1 = new Vector();

Vector v2 = new Vector();

//Кнопка для сложения векторов

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

{

v1.SetX1((double)numericUpDown1.Value);

v2.SetX1((double)numericUpDown2.Value);

v1.SetY1((double)numericUpDown3.Value);

v2.SetY1((double)numericUpDown4.Value);

v1.SetZ1((double)numericUpDown5.Value);

v2.SetZ1((double)numericUpDown6.Value);

MessageBox.Show($"{v1.Sum(v1.GetX1(), v2.GetX1(), v1.GetY1(), v2.GetY1(), v1.GetZ1(), v2.GetZ1())}");

}

//Кнопка для добавления вектора

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

{

v1.SetX1((double)numericUpDown1.Value);

v2.SetX1((double)numericUpDown2.Value);

v1.SetY1((double)numericUpDown3.Value);

v2.SetY1((double)numericUpDown4.Value);

v1.SetZ1((double)numericUpDown5.Value);

v2.SetZ1((double)numericUpDown6.Value);

if (v1.GetX1() == 0 && v2.GetX1() == 0 && v1.GetY1() == 0

&& v2.GetY1() == 0 && v1.GetZ1() == 0 && v2.GetZ1() == 0)

{

MessageBox.Show("Введен нуевой вектор");

}

else

{

addBut.Enabled = true;

subBut.Enabled = true;

skalBut.Enabled = true;

lengthBut.Enabled = true;

cosBut.Enabled = true;

AddVec.Enabled = false;

}

}

//Кнопка для рассчета разности векторов

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

{

v1.SetX1((double)numericUpDown1.Value);

v2.SetX1((double)numericUpDown2.Value);

v1.SetY1((double)numericUpDown3.Value);

v2.SetY1((double)numericUpDown4.Value);

v1.SetZ1((double)numericUpDown5.Value);

v2.SetZ1((double)numericUpDown6.Value);

MessageBox.Show($"{v1.Razn(v1.GetX1(), v2.GetX1(), v1.GetY1(), v2.GetY1(), v1.GetZ1(), v2.GetZ1())}");

}

//Кнопка для рассчета скалярного произведения векторов

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

{

v1.SetX1((double)numericUpDown1.Value);

v2.SetX1((double)numericUpDown2.Value);

v1.SetY1((double)numericUpDown3.Value);

v2.SetY1((double)numericUpDown4.Value);

v1.SetZ1((double)numericUpDown5.Value);

v2.SetZ1((double)numericUpDown6.Value);

MessageBox.Show($"Скалярное произведение = {v1.Scal(v1.GetX1(), v2.GetX1(), v1.GetY1(), v2.GetY1(), v1.GetZ1(), v2.GetZ1())}");

}

//Кнопка для рассчета дины векторов

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

{

v1.SetX1((double)numericUpDown1.Value);

v2.SetX1((double)numericUpDown2.Value);

v1.SetY1((double)numericUpDown3.Value);

v2.SetY1((double)numericUpDown4.Value);

v1.SetZ1((double)numericUpDown5.Value);

v2.SetZ1((double)numericUpDown6.Value);

MessageBox.Show($"Длина первого вектора = {v1.Length(v1.GetX1(), v1.GetY1(), v1.GetZ1())}" +

$"\nДлина второго вектора = {v2.Length(v2.GetX1(), v2.GetY1(), v2.GetZ1())}");

}

//Кнопка дя рассчета косинуса между векторами

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

{

v1.SetX1((double)numericUpDown1.Value);

v2.SetX1((double)numericUpDown2.Value);

v1.SetY1((double)numericUpDown3.Value);

v2.SetY1((double)numericUpDown4.Value);

v1.SetZ1((double)numericUpDown5.Value);

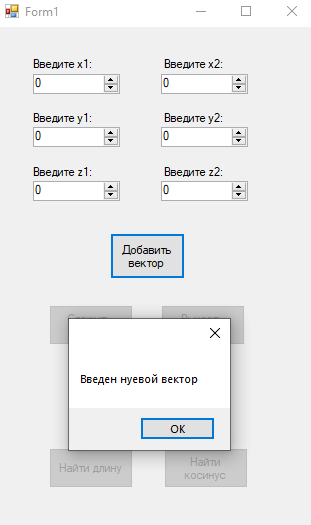
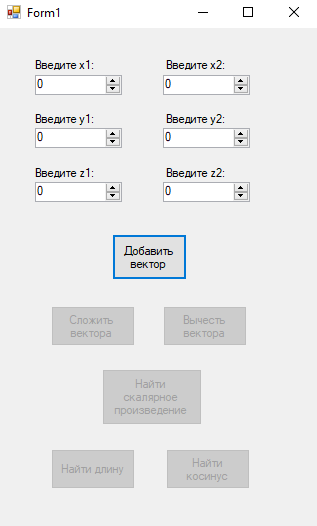
v2.SetZ1((double)numericUpDown6.Value);

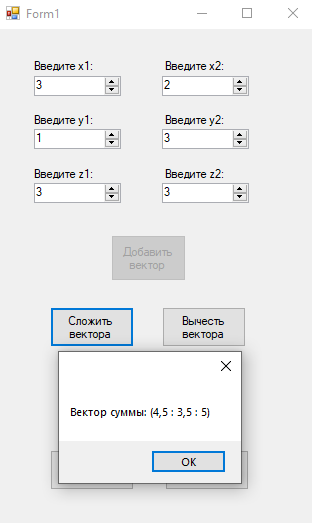
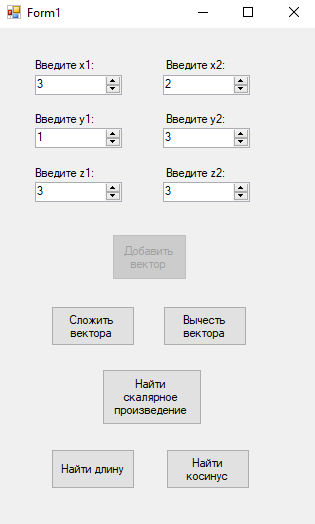
MessageBox.Show($"Косинус между углами = {v1.Cos(v1.GetX1(), v2.GetX1(), v1.GetY1(), v2.GetY1(), v1.GetZ1(), v2.GetZ1())}");

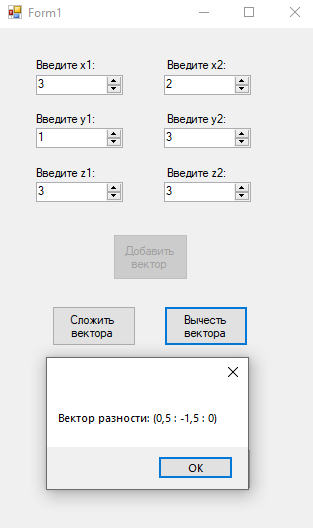
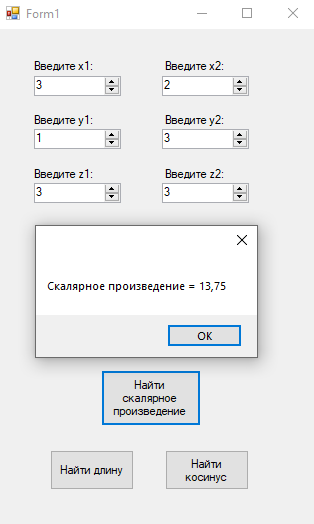
}

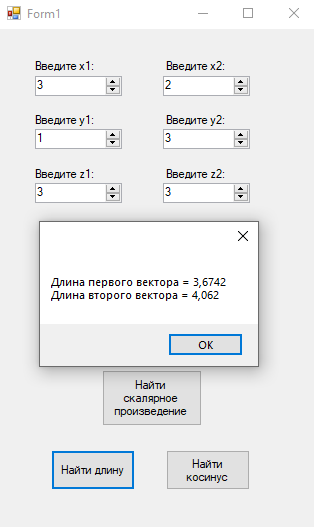
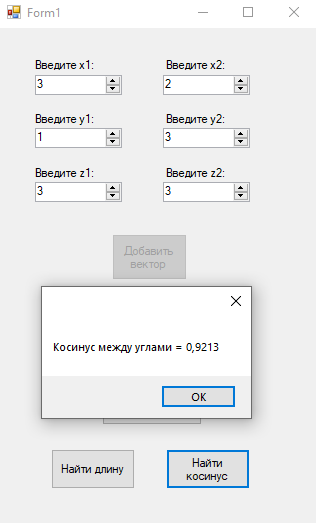
}

**Тестовые ситуации:**

****

****

**** ****

**** ****