Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

# Отчёт по программе «Индивидуальные задания по теме «Коллекции в с#»»

Выполнил: Плоских Александр Павлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание**

Описать класс «Векторы, задаваемые координатами концов в

трехмерном пространстве». Выполнить с использованием коллекции ArrayList.

Необходимо создать следующие методы класса:

* операции сложения и вычитания векторов с получением нового

вектора;·

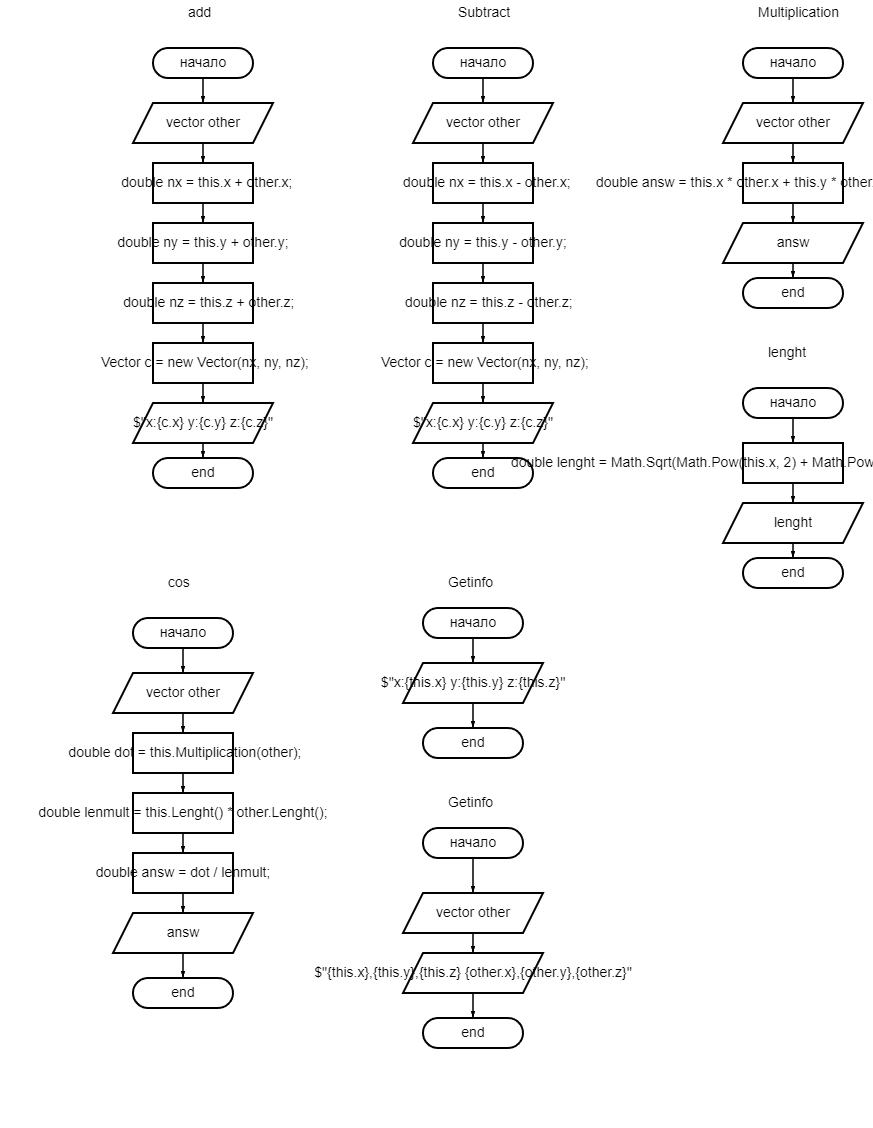
* вычисление скалярного произведения двух векторов и длины вектора;

операция вычисления косинуса угла между векторами.

**Входные и выходные данные**

Свойства класса Vector с типом данных double

**Блок-схема**

****

**Листинг программы**

*Основная программа*

if(new FileInfo("vector.txt").Length == 0)

{

try

{

Console.Write("X:");

double x = Convert.ToDouble(Console.ReadLine());

Console.Write("y:");

double y = Convert.ToDouble(Console.ReadLine());

Console.Write("z:");

double z = Convert.ToDouble(Console.ReadLine());

Console.Write("X:");

double x1 = Convert.ToDouble(Console.ReadLine());

Console.Write("y:");

double y1 = Convert.ToDouble(Console.ReadLine());

Console.Write("z:");

double z1 = Convert.ToDouble(Console.ReadLine());

Vectora v1 = new Vectora(x, y, z);

Vectora v2 = new Vectora(x1, y1, z1);

Console.WriteLine("add");

Console.WriteLine(v1.Add(v2));

Console.WriteLine("sub:");

Console.WriteLine(v1.Subtract(v2));

Console.WriteLine("Multi:");

Console.WriteLine(v1.Multiplication(v2));

Console.WriteLine("length1:");

Console.WriteLine(v1.Lenght());

Console.WriteLine("length2:");

Console.WriteLine(v2.Lenght());

Console.WriteLine("Cos:");

Console.WriteLine(v1.Cos(v2));

StreamWriter sw = File.CreateText("vector.txt");

sw.WriteLine(v1.Getinfo(v2));

sw.Close();

}

catch (FormatException)

{

Console.WriteLine("No");

}

}

else

{

try

{

List<Vectora> al = File.ReadAllLines("vector.txt").SelectMany(line => line.Split(' ')).SelectMany(ss => ss.Split(',')).Select(nn => double.Parse(nn)).Select((nn, index) => new { nn, index }).GroupBy(x => x.index / 3).Select(g => g.Select(x => x.nn).ToArray()).Select(nn => new Vectora(nn[0], nn[1], nn[2])).ToList();

for(int i = 0; i < (al.Count-1); i++)

{

Vectora v1 = (Vectora)al[i];

Vectora v2 = (Vectora)al[i + 1];

Console.WriteLine("v2:");

Console.WriteLine(v2.Getinfo());

Console.WriteLine("v1:");

Console.WriteLine(v1.Getinfo());

Console.WriteLine("add");

Console.WriteLine(v1.Add(v2));

Console.WriteLine("sub:");

Console.WriteLine(v1.Subtract(v2));

Console.WriteLine("Multi:");

Console.WriteLine(v1.Multiplication(v2));

Console.WriteLine("length1:");

Console.WriteLine(v1.Lenght());

Console.WriteLine("length2:");

Console.WriteLine(v2.Lenght());

Console.WriteLine("Cos:");

Console.WriteLine(v1.Cos(v2));

Console.WriteLine("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

}

catch (FormatException)

{

Console.WriteLine("no");

}

}

Console.ReadLine();

*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();

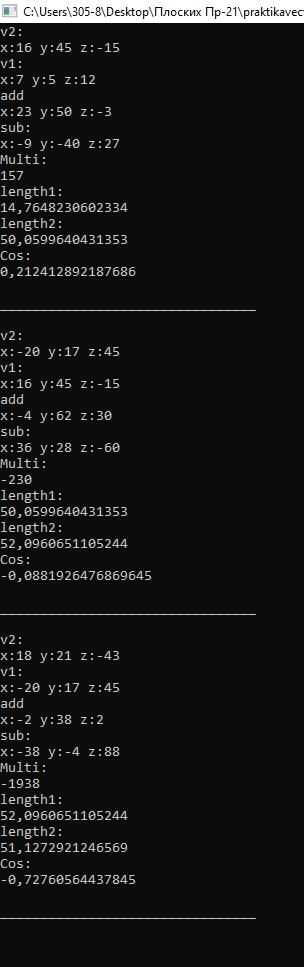
double answ = dot / lenmult;

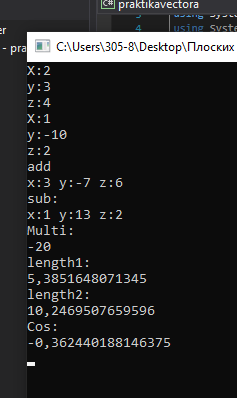
return lenmult;

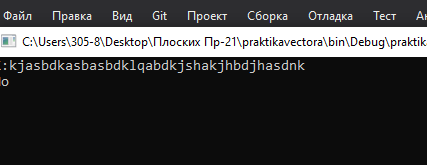
}

}

**Тестовые ситуации**

****

****

****

**Задание**

Выполнить с использованием коллекции ArrayList.

Описать класс «Почтовый адрес организации».

Необходимо создать следующие методы класса:·

 изменение составных частей адреса;·

 создание и уничтожение объектов этого класса.

Продемонстрировать работу с этим классом. Обязательно наличие меню,

через которое можно проверить работу всех методов класса.

**Входные и выходные данные**

Свойства класса Adress с типом данных string

**Блок-схема**

**Листинг программы**

Основная программа

ArrayList al = new ArrayList();

private void button5\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

int numbers = 1;

//pokazat

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

{

numbers = 1;

if (new FileInfo("adress.txt").Length != 0)

{

StreamReader sr = File.OpenText("adress.txt");

al.Clear();

comboBox1.Items.Clear();

while (!sr.EndOfStream)

{

string[] s = sr.ReadLine().Split(',');

Adress a = new Adress();

a.City = s[0];

a.Street = s[1];

a.Code = s[2];

if (a.Prov() == true)

{

listBox1.Items.Add(a.Getinfo());

al.Add(a);

comboBox1.Items.Add(numbers);

numbers++;

}

else

{

MessageBox.Show("Некоторые данные в файле не правильные");

}

}

sr.Close();

}

else

{

try

{

Adress a1 = new Adress();

a1.City = textBox1.Text;

a1.Street = textBox2.Text;

a1.Code = textBox3.Text;

if (a1.Prov() != false)

{

listBox1.Items.Add(a1.Getinfo());

comboBox1.Items.Add(numbers);

al.Add(a1);

numbers++;

StreamWriter sw = File.AppendText("adress.txt");

sw.WriteLine(a1.Getinfo());

sw.Close();

}

else

{

MessageBox.Show("Вы ввели некорректные данные попробуйте еще раз");

}

}

catch (FormatException)

{

}

}

}

//izmenenie

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

{

try

{

Adress a1 = (Adress)al[comboBox1.SelectedIndex];

a1.Update(textBox1.Text, textBox2.Text, textBox3.Text);

al[comboBox1.SelectedIndex] = a1;

StreamWriter sw = File.CreateText("adress.txt");

for (int i = 0; i < al.Count; i++)

{

Adress a2 = (Adress)al[i];

sw.WriteLine(a2.Getinfo());

}

sw.Close();

}

catch (FormatException) { }

}

//delete

private void button4\_Click(object sender, EventArgs e)

{

al.RemoveAt(comboBox1.SelectedIndex);

comboBox1.Items.Remove(comboBox1.SelectedItem);

StreamWriter sw = File.CreateText("adress.txt");

for (int i = 0; i < al.Count; i++)

{

Adress a2 = (Adress)al[i];

sw.WriteLine(a2.Getinfo());

}

sw.Close();

}

//dobav

private void button3\_Click(object sender, EventArgs e)

{

try

{

Adress a1 = new Adress();

a1.City = textBox1.Text;

a1.Street = textBox2.Text;

a1.Code = textBox3.Text;

if (a1.Prov() != false)

{

listBox1.Items.Add(a1.Getinfo());

comboBox1.Items.Add(numbers);

al.Add(a1);

numbers++;

StreamWriter sw = File.CreateText("adress.txt");

for (int i = 0; i < al.Count; i++)

{

Adress a2 = (Adress)al[i];

sw.WriteLine(a2.Getinfo());

}

sw.Close();

}

else

{

MessageBox.Show("Вы ввели некорректные данные попробуйте еще раз");

}

}

catch (FormatException)

{

}

}

Класс

private string city;

private string street;

private string code;

public string City

{

get { return city; }

set { city = value; }

}

public string Street

{

get { return street; }

set { street = value; }

}

public string Code

{

get { return code; }

set { code = value; }

}

public void Update(string city, string street, string code)

{

this.street = street;

this.city = city;

this.code = code;

}

public string Getinfo()

{

return $"{this.city},{this.street},{this.code}";

}

public bool Prov()

{

for (int i = 0; i < city.Length; i++)

if (!char.IsLetter(city[i])) return false;

for (int i = 0; i < street.Length; i++)

if (!char.IsLetter(street[i])) return false;

for (int i = 0; i < code.Length; i++)

if (!char.IsDigit(code[i])||code.Length!=6) return false;

return true;

}

**Тестовые ситуации**