**Самостоятельная работа 2**

**Тема: Определение классов**

**Задание 1**

Создайте в Visual Studio проект консольного типа.

Определите в проекте класс, описывающий вектор в трёхмерном пространстве.

Класс должен включать:

конструктор с параметрами в виде списка координат x, y, z

метод, вычисляющий длину вектора по формуле:   


метод, вычисляющий скалярное произведение этого вектора с другим по формуле:  


метод, вычисляющий векторное произведение с другим вектором:  


Переопределение метода ToString(), возвращающий строковое представление вектора в формате: **vec (x:число ; y:число ; z:число)**

В методе Main() создайте два экземпляра класса, описывающего вектор. Используя конструктор задайте для каждого экземпляра, его каждому полю/свойству значения. Используя метод ToString() получите строковое представление каждого объекта. Выведите в консоль полученные строковые представления объектов.  
Рассчитайте модули (длины) векторов и выведите в консоль результат с текстовым сообщением, поясняющим, что это длины векторов.  
Рассчитайте скалярное произведение векторов и выведите в консоль результат с текстовым сообщением, поясняющим, что это результат скалярного произведения векторов.

Рассчитайте векторное произведение векторов и выведите в консоль результат с текстовым сообщением, поясняющим, что это результат векторного произведения векторов.

Листинг Vector3D.cs

using System;

namespace ConsoleApp1

{

class Vector3D

{

public double X { get; set; }

public double Y { get; set; }

public double Z { get; set; }

public double VectorLength

{

get => Math.Sqrt(Math.Pow(X, 2) + Math.Pow(Y, 2) + Math.Pow(Z, 2));

}

public double ScalarProduct(Vector3D guestVector)

{

return ScalarProduct(guestVector.X, guestVector.Y, guestVector.Z);

}

public double ScalarProduct(double xGuest, double yGuest, double zGuest)

{

return X \* xGuest + Y \* yGuest + Z \* zGuest;

}

public Vector3D VectorProduct(Vector3D guestVector)

{

return VectorProduct(guestVector.X, guestVector.Y, guestVector.Z);

}

public Vector3D VectorProduct(double xGuest, double yGuest, double zGuest)

{

Vector3D result = new Vector3D();

result.X = Y \* zGuest - Z - yGuest;

result.Y = Z \* xGuest - X \* zGuest;

result.Z = X \* yGuest - Y \* xGuest;

return result;

}

public override string ToString()

{

return $"vec (x: {X}; y: {Y}; z: {Z})";

}

public Vector3D(double x = 0, double y = 0, double z = 0)

{

X = x;

Y = y;

Z = z;

}

}

}

Листинг Program.cs

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Vector3D vector1 = new Vector3D(2, 4, 6);

Vector3D vector2 = new Vector3D(3, 6, 9);

Console.WriteLine($"вектор 1: {vector1.ToString()}");

Console.WriteLine($"вектор 2: {vector2.ToString()}");

Console.WriteLine($"Длина 1 вектора: {vector1.VectorLength}");

Console.WriteLine($"Длина 2 вектора: {vector2.VectorLength}");

Console.WriteLine($"Скалярное произведение: " +

$"{vector1.ScalarProduct(vector2)}");

Console.WriteLine($"Векторное произведение: " +

$"{vector1.VectorProduct(vector2)}");

}

}

}