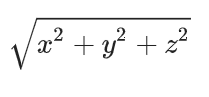
**Самостоятельная работа от 11.02.2020**

**Тема: Определение классов**

**Задание 2**

Создать в Visual Studio проект консольного типа.

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)**

Класс Vector3D:

class Vector3D

{

public double X;

public double Y;

public double Z;

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

{

X = x;

Y = y;

Z = z;

}

public double LengthVector() **// ДЛИНА ВЕКТОРА**

{

return Math.Sqrt(Math.Pow(X, 2) + Math.Pow(Y, 2) + Math.Pow(Z, 2));

}

public double ScalarProduct(Vector3D Vector) **// СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ**

{

return (X \* Vector.X + Y \* Vector.Y + Z \* Vector.Z);

}

public Vector3D VectorProduct(Vector3D Vector) **// ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ**

{

Vector3D res = new Vector3D(0.0, 0.0, 0.0);

res.X = Y \* Vector.Z - Z \* Vector.Y;

res.Y = Z \* Vector.X - X \* Vector.Z;

res.Z = X \* Vector.Y - Y \* Vector.X;

return res;

}

public override string ToString()

{

return $"vec(X: {X}, Y: {Y}, Z: {Z})";

}

}

Main():

static void Main(string[] args)

{

Vector3D myVector = new Vector3D(35.50, 87.90, 35.10);

Vector3D otherVector = new Vector3D(50.50, 32.35, 12.70);

Console.WriteLine("Вывод изначального вектора: " + myVector.ToString());

Console.WriteLine("Длина вектора: " + myVector.LengthVector());

Console.WriteLine("Cкалярное произведение: " + myVector.ScalarProduct(otherVector));

Console.WriteLine("Векторное произведение: " + myVector.VectorProduct(otherVector));

Console.ReadKey();

}

Результат работы программы:

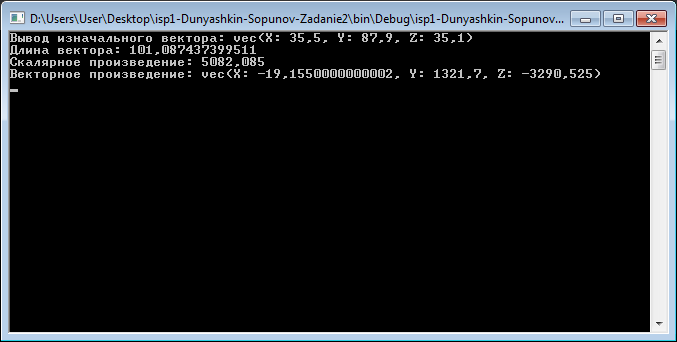


Рисунок 1 - результат описания класса Vector3D