**Практика «Readonly Vector»**

Помните класс Vector из позапрошлой практики? Скорее всего, он был написан ужасно, с открытыми полями и всем прочим.

Как правило, такие структуры данных делают read-only.

В пространстве имен ReadOnlyVectorTask сделайте класс ReadOnlyVector с двумя публичными readonly-полями X и Y, которые устанавливаются в конструкторе.

ReadOnlyVector должен содержать метод Add(ReadOnlyVector other), который возвращает сумму векторов.

При работе с readonly классами часто хочется изготовить вектор "такой же, но с другим значением поля X или Y". Обеспечьте такую функциональность с помощью методов WithX(double) и WithY(double)

// Вставьте сюда финальное содержимое файла ReadOnlyVector.cs

**Программа:**

**Содержимое файла Program.cs:**

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ReadOnlyVectorTask

{

class Program

{

public static void Main(string[] args)

{

ReadOnlyVector v1 = new ReadOnlyVector(3.2, 8.5);

ReadOnlyVector v2 = new ReadOnlyVector(v1.WithX(10.5).X, 8.5);

ReadOnlyVector v3 = new ReadOnlyVector(3.2, v1.WithY(7.3).Y);

ReadOnlyVector v4 = v1.Add(v2);

ReadOnlyVector v5 = v1.Add(v3);

Console.WriteLine("Вектор v1 х={0} y={1}", v1.X, v1.Y);

Console.WriteLine("Вектор v2 WithX х={0} y={1}", v2.X, v2.Y);

Console.WriteLine("Вектор v3 WithY х={0} y={1}", v3.X, v3.Y);

Console.WriteLine("Суммирующий вектор v4 х={0} y={1}", v4.X, v4.Y);

Console.WriteLine("Суммирующий вектор v5 х={0} y={1}", v5.X, v5.Y);

Console.ReadKey();

}

}

}

**Содержимое файла ReadOnlyVector.cs:**

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ReadOnlyVectorTask

{

public class ReadOnlyVector

{

public readonly double X;

public readonly double Y;

public ReadOnlyVector(double x, double y)

{

X = x;

Y = y;

}

public ReadOnlyVector Add(ReadOnlyVector otherVector)

{

return new ReadOnlyVector(X + otherVector.X, Y + otherVector.Y);

}

public ReadOnlyVector WithX(double x)

{

return new ReadOnlyVector(x, Y);

}

public ReadOnlyVector WithY(double y)

{

return new ReadOnlyVector(X, y);

}

}

}