**Practice «Readonly Vector»**

Remember the Vector class from the practice before last? Most likely, it was written badly, with open margins and all.

Typically, such data structures are read-only.

In the ReadOnlyVectorTask namespace, make a ReadOnlyVector class with two public readonly fields, X and Y, that are set in the constructor.

ReadOnlyVector must contain an Add (ReadOnlyVector other) method that returns the sum of the vectors.

When working with readonly classes, you often want to make a vector "the same, but with a different X or Y field value". Provide this functionality with the WithX (double) and WithY (double) methods

// Paste the final content of the ReadOnlyVector.cs file here

**Code:**

**Contents of Program.cs file:**

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();

}

}

}

**Contents ReadOnlyVector.cs file:**

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);

}

}

}