Лабораторная работа 2.

В лабораторной работе 2 в типы из лабораторной работы 1 надо добавить новые методы и свойства, связанные с чтением данных из файла и запросами LINQ к данным, которые хранятся в типах.

Вариант 5-1

Определить структуру **DataItem**, котораясодержит открытые автореализуемые свойства

* типа **System.Numerics.Vector2** с координатами двумерной точки;
* типа **System.Numerics.Vector2** для значения электромагнитного поля.

В структуре **DataItem** определить

* конструктор с двумя параметрами типа **System.Numerics.Vector2** для инициализации данных типа;
* перегруженную (**override**) версию виртуального метода **string ToString();**
* метод **string ToString(string format)**, который возвращает строку, содержащую координаты точки, в которой измеряется поле, значение поля в этой точкеидлину вектора поля,и использует параметр **format** для чисел с плавающей запятой.

В структуру **Grid2D** добавить

* метод **string ToString(string format)**, который возвращает строку с данными структуры и использует параметр **format** для чисел с плавающей запятой.

В класс **V5Data** добавить

* абстрактный (или виртуальный) метод **string** **ToLongString(****string format),** который использует параметр **format** для чисел с плавающей запятой;

В классе **V5DataOnGrid** определить открытый конструктор **public V5DataOnGrid (string filename),** который инициализирует объект данными из файла **filename:**

* В файле **filename** в текстовом виде хранится вся информация об объекте **V5DataOnGrid** – данные базового класса, данные объекта **Grid2D** изначения поля в узлах сетки.
* Код, который читает текстовые данные из файла и преобразует их в объект соответствующего типа, должен находиться в блоке **try-catch-finally** и обрабатывать все исключения, которые могут быть брошены при разборе файла.
* Формат хранения данных в текстовом виде выбирайте сами – порядок, в котором хранятся данные, разделители, какие данные находятся в одной строке и т.д. Описание выбранного формата надо сделать в отдельном файле (docx или pdf) или как комментарий в исходном коде с конструктором класса **V5DataOnGrid**.

В класс **V5DataOnGrid** добавить

* реализацию интерфейса **IEnumerable<DataItem>**, который перечисляет все данные на сетке как экземпляры **DataItem** - для каждого узла сетки вычисляет координаты узла и отвечающее этому узлу значение **Vector2;**
* реализацию абстрактного (или виртуального) метода **string ToLongString**(**string format**), который возвращает строку с именем типа, данными базового класса, данными **Grid2D** и информацию о каждом узле сетки (координаты узла сетки и значение в узле) и использует параметр **format** для чисел с плавающей запятой.

В класс **V5DataCollection** добавить

* реализацию интерфейса **IEnumerable<DataItem>**, который перечисляет все элементы из коллекции **Dictionary<Vector2,Vector2>** как экземпляры **DataItem;**
* реализацию абстрактного (или виртуального) метода **string ToLongString**(**string format**), который возвращает строку с именем типа, данными базового класса, информацию для каждого элемента **Dictionary<Vector2,Vector2>** (координаты узлов сеткии значение поля) и использует параметр **format** для чисел с плавающей запятой.

В класс **V5MainCollection** добавить

* метод string **ToLongString(string format)**, в котором для создания строки с данными для каждого элемента **List<V5Data>** вызывается метод **ToLongString(string format)**.

В классе **V5MainCollection** определить свойства (только с методом **get**) для выполнения операций с данными, использующие интегрированные в язык C# запросы LINQ. В этих свойствах не должно быть операторов foreach или операторов цикла, только запросы LINQ.

1. Cвойство типа **float**, возвращающее минимальное значение длины вектора поля среди всех измеренных значений поля в коллекции **V5DataCollection.**
2. Cвойство типа **IEnumerable<****DataItem>,** перечисляющее как объекты **DataItem** результаты измерений изколлекции **V5MainCollection** с минимальным значением длины вектора поля.
3. Cвойство типа **IEnumerable<Vector2>,** перечисляющее все точки (как объекты **Vector2**), в которых измерено поле, такие, что они есть в элементах типа **V5OnDataGrid,** но их нет в элементах **V5DataCollection.**

Результат измерений – это данные для одного узла сетки (как элемент **DataItem)** для элементов из коллекции **V5MainCollection**, которые имеют тип **V5DataOnGrid**, и данные одного элемента в коллекции **Dictionary<****Vector2, Vector2>** (как элемент **DataItem)** для элементов, которые имеют тип **V5DataCollection**. Число результатов измерений в элементах **V5DataCollection** – это число элементов в коллекции **Dictionary<Vector2, Vector2>**. Число измерений в элементах **V5DataOnGrid** – это число узлов сетки.

В методе **Main()**

1. Создать один объект типа **V5DataOnGrid**, инициализировать его с помощью конструктора, который читает данные для инициализации из файла, и вывести его с помощью метода **ToLongString(string format).**
2. В методе **AddDefaults()** в коллекцию **V5MainCollection** добавить такие данные, чтобы можно было отладить все запросы LINQ. Можно дополнительно определить конструкторы в классах **V5DataOnGrid** и **V5DataCollection.** Среди элементов коллекции должен быть элемент типа **V5DataCollection**, у которого в списке **List<DataItem>** нет элементов, и элемент типа **V5DataOnGrid**, в котором число узлов сетки равно 0.
3. Создать объект типа **V5MainCollection**. Вызвать метод **AddDefaults()** и вывести данные объекта **V5MainCollection**. Вызвать все перечисленные выше свойства класса **V5MainCollection** с запросамиLINQ и вывести результаты выполнения запросов. Вывод должен быть подписан - перед выводом результата выполнения каждого запроса должна быть выведена информация с описанием запроса. При выводе результатов запросов надо использовать методы **ToLongString** с форматированием.
4. В программе должны быть обработаны все исключения, которые могут быть брошены при выполнении приложения.

**Срок сдачи лабораторной работы 23 ноября.**

**Дополнительная информация.**

Ссылка на справочную информацию о стандартных числовых форматах:

<https://docs.microsoft.com/ru-ru/dotnet/standard/base-types/standard-numeric-format-strings>

В классах **System.Numerics.Vector2** и **System.Numerics.Double** есть методы:

|  |  |
| --- | --- |
| public string ToString(string format); | возвращает строковое представление текущего экземпляра, используя заданную строку форматирования для форматирования отдельных элементов; |
| public float Length (); | возвращает длину вектора; |
| public float LengthSquared (); | возвращает длину вектора в квадрате; обеспечивает лучшую производительность, чем метод Length; |
| public static  float Distance (System.Numerics.Double value1, System.Numerics.Double value2); | возвращает евклидово расстояние между двумя заданными точками; |
| public static  float DistanceSquared (System.Numerics.Double value1, System.Numerics.Double value2); | возвращает квадрат евклидова расстояния между двумя заданными точками; |

Ссылки на на справочную информацию для System.Numerics.Vector2

<https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.vector2?view=netcore-3.1>

<https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.vector2?view=netframework-4.8>

Ссылки на на справочную информацию для System.Numerics.Double

[https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.Double?view=netcore-3.1](https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.vector3?view=netcore-3.1)

[https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.Double?view=netframework-4.8](https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.vector3?view=netframework-4.8)

Ссылки на на справочную информацию для System.Numerics.Double

[https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.double?view=netcore-3.1](https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.complex?view=netcore-3.1)

[https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.double?view=netframework-4.8](https://docs.microsoft.com/ru-ru/dotnet/api/system.numerics.complex?view=netframework-4.8)