

# 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 2-1

В структуру **DatalItem** добавить

- метод **string ToString(string format)**, который возвращает строку, содержащую координаты точки, в которой измеряется поле, комплексное значение поля в этой точке и модуль значения поля, и использует параметр **format** для чисел с плавающей запятой.

В структуру **Grid1D** добавить

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

В класс **V2Data** добавить

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

В классе **V2DataOnGrid** определить открытый конструктор **public V2DataOnGrid (string filename)**, который инициализирует объект данными из файла **filename**:

- В файле **filename** в текстовом виде хранится вся информация об объекте **V2DataOnGrid** – данные базового класса, данные объектов **Grid1D** и значения 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 **V2DataOnGrid**.

В класс **V2DataOnGrid** добавить

- реализацию интерфейса **IEnumerable<DatalItem>**, который перечисляет все данные на сетке как экземпляры **DatalItem** - для каждого узла сетки вычисляет координаты узла и отвечающее этому узлу значение **Complex**;
- реализацию абстрактного (или виртуального) метода **string ToLongString(string format)**, который возвращает строку с именем типа, данными базового класса, данными **Grid1D**

и информацию о каждом узле сетки ( координаты узла сетки, комплексное значение в узле, модуль значения) и использует параметр **format** для чисел с плавающей запятой.

В класс **V2DataCollection** добавить

- реализацию интерфейса **IEnumerable<DataItem>**, который перечисляет все элементы **DataItem** из списка **List<DataItem>**;
- реализацию абстрактного (или виртуального) метода **string ToLongString(string format)**, который возвращает строку с именем типа, данными базового класса, информацию для каждого элемента **List<DataItem>** (координаты точек измерения, значение поля и модуль значения) и использует параметр **format** для чисел с плавающей запятой.

В класс **V2MainCollection** добавить

- метод **string ToLongString(string format)**, в котором для создания строки с данными для каждого элемента **List<V2Data>** вызывается метод **ToLongString(string format)**.

В классе **V2MainCollection** 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. Свойство типа **double**, возвращающее среднее значение модуля поля для всех результатов измерений в коллекции **V2MainCollection**.
2. Свойство типа **DataItem**, возвращающее объект **DataItem**, в котором значение модуля поля ближе всего к среднему значению модуля поля среди всех результатов измерений. Если в коллекции **V2MainCollection** несколько таких элементов, свойство возвращает любой из них.
3. Свойство **IEnumerable<Vector2>**, который перечисляет как экземпляры **Vector2** точки измерения поля, которые встречаются в каждом элементе типа **V2DataCollection** в коллекции **V2MainCollection**.

Результат измерений – это данные для одного узла сетки (как элемент **DataItem**) для элементов коллекции **V2MainCollection**, которые имеют тип **V2DataOnGrid**, и элемент **DataItem** в списке **List<DataItem>** для элементов, которые имеют тип **V2DataCollection**. Число результатов измерений в элементах **V2DataCollection** – это число элементов в списке **List<DataItem>**. Число результатов измерений в элементах **V2DataOnGrid** – это число узлов сетки.

В методе **Main()**

1. Создать один объект типа **V2DataOnGrid**, 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 **V2MainCollection** добавить такие данные, чтобы можно было отладить все запросы **LINQ**. Можно дополнительно определить конструкторы в классах **V2DataOnGrid** и **V2DataCollection**. Среди элементов коллекции должны быть элемент типа **V2DataCollection**, у которого в списке **List<DataItem>** нет элементов, и элемент типа **V2DataOnGrid**, в котором число узлов сетки равно 0.

3. Создать объект типа **V2MainCollection**. Вызвать метод **AddDefaults()** и вывести данные объекта **V2MainCollection**. Вызвать все перечисленные выше свойства класса **V2MainCollection** с запросами LINQ 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.Complex** есть методы:

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

Ссылки 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 справочную информацию для System.Numerics.Complex

<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.complex?view=netframework-4.8>

Ссылки на справочную информацию для System.Numerics.Complex

<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.complex?view=netframework-4.8>