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 1- 2

В структуру **DataItem** добавить

* метод **string ToString(string format)**, который возвращает строку, содержащую значение времени **t,** значения компонент поля в момент времени **t** и длину вектора поля, и использует параметр **format** для чисел с плавающей запятой.

В структуру **Grid** добавить

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

В класс **V1Data** добавить

* абстрактный (или виртуальный) метод **string** **ToLongString(****string format)**, который возвращает строку с данными класса и использует параметр **format** для чисел с плавающей запятой.

В класс **V1DataOnGrid** добавить

* реализацию интерфейса **IEnumerable<DataItem>**, который перечисляет все данные на сетке как экземпляры **DataItem** - для каждого узла сетки вычисляет координату и отвечающее этому узлу значение **Vector3;**

* реализацию абстрактного (или виртуального) метода **string ToLongString**(**string format**), который возвращает строку с именем типа, данными базового класса, данными свойства **Grid** и информацию о каждом узле сетки – координату, значения компонент поля, ***длину вектора*** и использует параметр **format** для чисел с плавающей запятой.

В классе **V1Data****Collection** определить открытый конструктор **public V1DataCollection (string filename),** который инициализирует объект данными из файла **filename:**

* В файле **filename** в текстовом виде хранится вся информация об объекте **V1DataCollection** – данные базового класса, значения t и векторзначений поля в этих точках.
* 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 **V1DataCollection**.

В класс **V1DataCollection** 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>**  (значение **t,** компоненты поля и ***длину вектора***) и использует параметр **format** для чисел с плавающей запятой.

В класс **V1MainCollection** добавить

* метод string **ToLongString(string format)**, в котором для создания строки с данными для каждого элемента **List<V1Data>** вызывается метод **ToLongString(string format)**.

В классе **V1MainCollection** 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 **int**, возвращающее максимальное число результатов измерений поля в элементах **V1Data** из списка **List<V1Data>** коллекции **V1MainCollection.**
2. Свойство типа **IEnumerable<DataItem>**, возвращающее переменную запроса для перечисления в порядке убывания длины вектора поля всех результатов измерений в коллекции **V1MainCollection** как экземпляры **DataItem.**
3. Cвойство типа **IEnumerable<float>,** котороеперечисляет все значения **t**, которые встречаются среди всех результатов измерений в коллекции **V1MainCollection** только один раз.

Результат измерений – это данные для одного узла сетки (как элемент **DataItem)** для элементов коллекции **V1MainCollection**, которые имеют тип **V1DataOnGrid**, и элемент **DataItem** в списке **List<DataItem>** для элементов, которые имеют тип **V1DataCollection**. Число результатов измерений в элементах **V1DataCollection** – это число элементов в списке **List<DataItem>**. Число результатов измерений в элементах **V1DataOnGrid** – это число узлов сетки.

В методе **Main()**

1. Создать один объект типа **V1DataCollection**, 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 **V1MainCollection** добавить такие данные, чтобы можно было отладить все запросы LINQ. Можно дополнительно определить конструкторы в классах **V1DataOnGrid** и **V1DataCollection.** Среди элементов коллекции должен быть элемент типа **V1DataCollection**, у которого в списке **List<DataItem>** нет элементов, и элемент типа **V1DataOnGrid**, в котором число узлов сетки равно 0.
3. Создать объект типа **V1MainCollection**. Вызвать метод **AddDefaults()** и вывести данные объекта **V1MainCollection**. Вызвать все перечисленные выше свойства класса **V1MainCollection** 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.Vector3** 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.Vector3 value1, System.Numerics.Vector3 value2); | возвращает евклидово расстояние между двумя заданными точками; |
| public static  float DistanceSquared (System.Numerics.Vector3 value1, System.Numerics.Vector3 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.Vector3

<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.vector3?view=netframework-4.8>

Ссылки на 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>