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 4_1

Реализация интерфейса IEnumerable<DataItem>

Абстрактный класс **V4Data** объявить как реализующий интерфейс **IEnumerable<DataItem>.**

В классах **V4DataList** и **V4DataArray** реализовать интерфейс **IEnumerable<DataItem>.**

- в классе **V4DataArray** реализация интерфейса **IEnumerable<DataItem>** перечисляет все элементы **DataItem** из списка **List<DataItem>;**
- в классе **V4DataList** реализация интерфейса **IEnumerable<DataItem>** перечисляет все данные на сетке как экземпляры **DataItem** − для каждого узла сетки создается экземпляр **DataItem** с координатами узла сетки и значением типа **Vector2** в узле сетки**.**

Запросы LINQ

В классе **V4MainCollection** 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**.

Результат измерений – это данные для одного узла сетки (как элемент **DataItem)** для элементов коллекции **V4MainCollection**, которые имеют тип **V4DataArray**, и элемент **DataItem** в списке **List<DataItem>** для элементов, которые имеют тип **V4DataList**.
Точка, в которой измерено поле, – это узел сетки в типе **V4DataArray** и точка с координатами **(x,y)**, которые хранятся в свойстве типа **Vector2**, в элементах **DataItem** в типе **V4DataList.**
Число результатов измерений в элементах **V4DataList** – это число элементов в списке **List<DataItem>**. Число результатов измерений в элементах **V4DataArray** – это число узлов сетки.

В классе **V4MainCollection** определить

1. Свойство типа **float**, возвращающее максимальное значение значение модуля поля среди всех результатов измерений в элементах коллекции **V4MainCollection**, которые имеют тип **V4DataArray**. Если в коллекции нет элементов, свойство возвращает значение **float.NaN**.

2. Свойство типа **IEnumerable<DataItem>**, возвращающее переменную запроса для перечисления всех результатов измерения поля (как элеменов **DataItem**) из **V4MainCollection** в порядке убывания расстояния от начала координат до точки, где измерено поле. Если в коллекции нет элементов, свойство возвращает значение **null**.

3. Свойство типа **IEnumerable<Vector2>**, которое перечисляет без повторов все точки, в которых измерено поле (как объекты **Vector2**), такие, что они есть в элементах типа **V4DataArray,** но их нет в элементах **V4DataList.**

Запись и восстановление данных

В абстрактном базовом классе **V4Data** изменить доступ к открытым автореализуемым свойствам. Класс **V1Data** содержит открытые автореализуемые свойства

- типа **string** с методами **get** и **protected set** для идентификации объекта данных;
- типа **DateTime** с методами **get** и **protected set** для даты измерений поля.

В классе **V4DataArray** в открытые автореализуемые свойства для параметров сетки и в открытое свойство типа **Vector2 [,]** для двумерного прямоугольного массива добавить метод **private set.**

В класс **V4DataArray** добавить

- экземплярный метод **bool SaveBinary (string filename)** или статический метод **bool SaveBinary (string filename, V4DataArray v4);**
- экземплярный метод **bool LoadBinary (string filename, ref V4DataArray v4)** или статический метод **bool LoadBinary (string filename, ref V4DataArray v4).**

Метод **SaveBinary** сохраняет все данные объекта в файле с именем **filename** в двоичном виде**.**
Метод **LoadBinary** восстанавливает все данные объекта из файла с именем **filename.** Для сохранения/восстановления объекта типа **V4DataList** можно использовать бинарную сериализацию из класса **BinaryFormatter** или методы для записи/чтения из классов **BinaryWriter/BinaryReader** (на ваш выбор).

В класс **V4DataList** добавить

- экземплярный метод **bool SaveAsText (string filename)** или статический метод **bool SaveAsText (string filename, V4DataList v4);**
- экземплярный метод **bool LoadAsText (string filename, ref V4DataList v4)** или статический метод **bool LoadAsText (string filename, ref V4DataList v4).**

Метод **SaveAsText** сохраняет все данные объекта в файле с именем **filename** в текстовом виде**.** Метод **LoadAsText** восстанавливает все данные объекта из файла с именем **filename.** Для сохранения/восстановления объекта типа **V4DataList** можно использовать текстовую сериализацию из класса **JsonFormatter** или методы для записи/чтения из класса **StreamWriter/StreamReader** или комбинацию этих способов сохранения данных (на ваш выбор).

Коды, которые сохраняют данные в файле и читают данные из файла и преобразуют их в объекты соответствующего типа, должны находиться в блоке **try-catch-finally** и обрабатывать все исключения, которые могут быть брошены при записи и чтении из файла.

Отладка программы

В классе, который содержит статический метод **Main()**, определить два статических метода для отладки чтения/записи данных в файл и для отладки свойств класса **V4MainCollection** с запросами **LINQ**. Эти методы вызываются из метода **Main.**

В одном методе

1. Создать объект **V4DataArray.** Сохранить его в файле. Восстановить объект из файла и вывести исходный и сохраненный объекты.

2. Создать объект **V4DataList.** Сохранить его в файле. Восстановить объект из файла и вывести исходный и сохраненный объекты.

Во втором методе

1. Создать объект типа **V4MainCollection**, добавить элементы в коллекцию **List<V4Data>** и вывести всю коллекцию. В коллекцию надо добавить такие элементы, чтобы можно было проверить, что все запросы **LINQ** работают правильно. Среди элементов коллекции должен быть элемент типа **V4DataList**, у которого в списке **List<DataItem>** нет элементов, и элемент типа **V4DataArray**, в котором число узлов сетки равно 0.

2. Вызвать все перечисленные выше свойства класса **V4MainCollection** 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.

В программе должны быть обработаны все исключения, которые могут быть брошены при выполнении приложения.

Срок сдачи лабораторной работы 16 ноября