

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

Реализация интерфейса **IEnumerable<DataItem>**

Абстрактный класс **V1Data** объявить как реализующий интерфейс **IEnumerable<DataItem>**.

В классах **V1DataList** и **V1DataArray** реализовать интерфейс **IEnumerable<DataItem>**.

- в классе **V1DataList** реализация интерфейса **IEnumerable<DataItem>** перечисляет все элементы **DataItem** из списка **List<DataItem>**;
- в классе **V1DataArray** реализация интерфейса **IEnumerable<DataItem>** перечисляет все данные на сетке как экземпляры **DataItem** – для каждого узла сетки создается экземпляр **DataItem** с координатами узла сетки и комплексным значением в узле сетки.

Запросы LINQ

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**.

Результат измерений – это данные для одного узла сетки (как элемент **DataItem**) для элементов коллекции **V1MainCollection**, которые имеют тип **V1DataArray**, и элемент **DataItem** в списке **List<DataItem>** для элементов, которые имеют тип **V1DataList**.

Точка, в которой измерено поле, – это узел сетки в типе **V1DataArray** и точка с координатами **(x,y)** в элементах **DataItem** в типе **V1DataList**.

Число результатов измерений в элементах **V1DataList** – это число элементов в списке **List<DataItem>**. Число результатов измерений в элементах **V1DataArray** – это число узлов сетки.

В классе **V1MainCollection** определить открытые

1. Свойство типа **DateTime?**, которое возвращает минимальное значение времени измерений (значение свойства типа **DateTime** базового класса **V1Data**) среди всех элементов **V1Data** в **V1MainCollection**. Если в коллекции **V1MainCollection** нет элементов, свойство возвращает значение **null**.
2. Свойство типа **IEnumerable<V1DataList>**, которое перечисляет элементы коллекции **V1MainCollection**, которые имеют тип **V1DataList**, в порядке убывания среднего значения (среднего арифметического) модуля поля. Если в коллекции нет элементов, свойство возвращает значение **null**.

3. Свойство типа **IEnumerable<V1Data>**, которое перечисляет все элементы **V1Data** коллекции **V1MainCollection** с максимальным числом результатов измерений поля. Если в коллекции нет элементов, свойство возвращает значение **null**.

Запись и восстановление данных

В абстрактном базовом классе **V1Data** 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 **V1DataArray** в открытые автореализуемые свойства для параметров сетки и в открытое свойство типа **Complex [,]** для двумерного прямоугольного массива добавить метод **private set**.

В класс **V1DataArray** добавить

- экземплярный метод **bool SaveAsText (string filename)** или статический метод **bool SaveAsText (string filename, V1DataArray v1)**;
- экземплярный метод **bool LoadAsText (string filename, ref V1DataArray v1)** или статический метод **bool LoadAsText (string filename, ref V1DataArray v1)**.

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 **V1DataList** 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).

В класс **V1DataList** добавить

- экземплярный метод **bool SaveBinary (string filename)** или статический метод **bool SaveBinary (string filename, V1DataArray v1)**;
- экземплярный метод **bool LoadBinary (string filename, ref V1DataList v1)** или статический метод **bool LoadBinary (string filename, ref V1DataList v1)**.

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 **V1DataList** можно использовать бинарную сериализацию из класса **BinaryFormatter** или методы для записи/чтения из классов **BinaryWriter/BinaryReader** (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 **V1MainCollection** с запросами **LINQ**. Эти методы вызываются из метода **Main**.

В одном методе

1. Создать объект **V1DataArray**. Сохранить его в файле. Восстановить объект из файла и вывести исходный и сохраненный объекты.
2. Создать объект **V1DataList**. Сохранить его в файле. Восстановить объект из файла и вывести исходный и сохраненный объекты.

Во втором методе

1. Создать объект типа **V1MainCollection**, добавить элементы в коллекцию **List<V1Data>** 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 **V1DataList**, у которого в списке **List<DataItem>** нет элементов, и элемент типа **V1DataArray**, в котором число узлов сетки равно 0.
2. Вызвать все перечисленные выше свойства класса **V1MainCollection** 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