

Лабораторная работа 1.

В работе определяется абстрактный базовый класс и два производных класса. В одном производном классе данные хранятся в массивах, в другом – в одном из типов коллекций стандартной библиотеки .NET. Кроме того, в работе определяется класс, содержащий коллекцию из элементов производных классов.

Привязка классов к данным измерений некоторых физических величин условная, так как типы, которые определяются в лабораторных работах, это “учебные” типы для изучения синтаксических конструкций языка C# и стандартной библиотеки .NET.

Производные классы можно рассматривать как два разных формата для хранения данных измерений некоторой векторной физической величины (например, двух компонент электромагнитного поля) на одномерной сетке. Узел сетки задается его координатой. Координату узла сетки можно трактовать, например, как момент времени измерения поля или как координату точки на прямой, в которой измеряется поле. С каждым узлом сетки связаны комплексные значения двух компонент поля.

Типы, определенные в лабораторной работе, используются в следующих лабораторных работах, в том числе в следующем семестре.

Вариант 1

В лабораторной работе надо определить следующие типы:

- **struct Dataltem** для хранения данных, связанных с одним узлом сетки;
- абстрактный базовый класс **V1Data** и два производных от него класса **V1DataList** и **V1DataArray**; в классе **V1DataArray** данные измерений хранятся в одномерных массивах, в классе **V1DataList** данные хранятся в коллекции **List<Dataltem>**;
- класс **V1MainCollection** для коллекции объектов типа **V1DataList** и **V1DataArray**;
- делегат **void FValues (double x,**
ref System.Numerics.Complex y1,
ref System.Numerics.Complex y2);
- делегат **Dataltem FDI (double x);**

Делегаты FValues и FDI

Экземпляры делегатов **FValues** и **FDI** используется как параметры конструкторов при инициализации объектов типа **V1DataArray** и **V1DataList**.

Методы, отвечающие делегатам **FValues** и **FDI**, можно определить в отдельном статическом классе или в классах **V1DataArray** и **V1DataList**.

Структура Dataltem

Структура **Dataltem** содержит открытые автореализуемые свойства

- типа **double** с координатой x узла сетки;
- два свойства типа **System.Numerics.Complex** для двух комплексных значений поля $\{y_1, y_2\}$ в этом узле сетки.

В структуре **Dataltem** определены открытые

- конструктор
Dataltem (double x, System.Numerics.Complex y1, System.Numerics.Complex y2) для инициализации данных структуры;
- метод **string ToString(string format)**, возвращающий строку, которая содержит значение координаты узла сетки и значения поля; параметр **format** задает формат вывода чисел с плавающей запятой;
- перегруженная (**override**) версия виртуального метода **string ToString()**.

Класс V1Data

Абстрактный базовый класс **V1Data** содержит открытые

- автореализуемое свойство **Key** типа **string**;
- автореализуемое свойство **Date** типа **DateTime**;
- конструктор с параметрами типа **string** и **DateTime**;
- абстрактное свойство **xLength** типа **int** (только с методом **get**);
- абстрактное свойство **MinMaxDifference** типа кортеж **(double, double)** (только с методом **get**);
- абстрактный метод **string ToLongString(string format)**;
- перегруженную (**override**) версию виртуального метода **string ToString()**;

Класс V1DataList

Класс **V1DataList** является производным от класса **V1Data**. В классе **V1DataList** данные измерений хранятся в коллекции **List<Dataltem>**. Среди элементов **Dataltem**, входящих в коллекцию **List<Dataltem>**, не должно быть элементов с совпадающими значениями координат x узлов сетки.

В классе **V1DataList** определены открытые

- автореализуемое свойство типа **List<Dataltem>**;
- конструктор **V1DataList (string key, DateTime date)**
для инициализации данных базового класса; в этом конструкторе распределяется память для коллекции **List<Dataltem>**;
- конструктор
V1DataList (string key, DateTime date, double[] x, FDI F),

в который через параметр **double[] x** передается ссылка на массив с координатами узлов сетки; в конструкторе для каждого элемента массива вызывается метод **F**, который вычисляет значения поля $\{y_1, y_2\}$; создается и добавляется в коллекцию **List<DataItem>** элемент **DataItem**; для равных элементов массива **x** в коллекцию добавляется только один элемент **DataItem**;

- реализацию абстрактного свойства **xLength** типа **int**, которое возвращает число узлов сетки (число элементов в коллекции **List<DataItem>**);
- реализацию абстрактного свойства **MinMaxDifference** типа кортеж **(double, double)**, которое возвращает минимальное и максимальные значения модуля разности компонент поля $\{y_1, y_2\}$ среди всех элементов **List<DataItem>**;
- оператор преобразования типа **V1DataList** к типу **V1DataArray** – метод **public static explicit operator V1DataArray (V1DataList source);** преобразующий данные, которые хранятся в объекте типа **V1DataList**, в данные в формате **V1DataArray**;
- перегруженную (override) версию виртуального метода **string ToString()**, который возвращает строку с именем типа объекта, данными базового класса и числом элементов в списке **List<DataItem>**;
- реализацию абстрактного метода **string ToLongString(string format)**, который возвращает строку с такими же данными, что и метод **ToString()**, и дополнительно информацию о каждом элементе из **List<DataItem>** – координату узла сетки и значения поля; параметр **format** задает формат вывода чисел с плавающей запятой.

Класс V1DataArray

Класс **V1DataArray** определяется как производный от абстрактного базового класса **V1Data**. В классе **V1DataArray** в массиве типа **double[]** хранятся координаты **x** узлов сетки, в массиве типа **System.Numerics.Complex[]** хранятся измеренные значения поля. Число элементов массива **System.Numerics.Complex[]** в два раза больше, чем число узлов сетки. В первых двух элементах массива типа **System.Numerics.Complex[]** хранятся значения компонент поля для первого узла сетки, в следующих двух элементах массива хранятся значения для второго узла сетки и так далее.

Класс **V1DataArray** содержит открытые

- автореализуемое свойство типа **double[]** для массива узлов сетки;
- автореализуемое свойство типа **System.Numerics.Complex []** для массива со значениями поля в узлах сетки;
- конструктор

V1DataArray (string key, DateTime date)

для инициализации данных базового класса; в этом конструкторе распределяется память для массивов, в которых хранятся данные, с нулевым числом элементов;

- конструктор

V1DataArray (string key, DateTime date, double[] x, FValues F),

в который через параметр **double[] x** передается массив с координатами узлов сетки, распределяется память для массива **System.Numerics.Complex[]**, и для каждого

элемента массива **x** вызывается метод **F** типа **FValues**, который вычисляет значения поля в узле;

- индексатор типа **DatalItem?** (только с методом **get**) с целочисленным индексом; индексатор возвращает объект **DatalItem**, отвечающий узлу сетки с соответствующим номером, и значение **null**, если значение индекса выходит за пределы диапазона номеров узлов сетки;
- реализацию абстрактного свойства **xLength** типа **int**, которое возвращает число узлов сетки;
- реализацию абстрактного свойства **MinMaxDifference** типа кортеж **(double, double)**, которое возвращает минимальное и максимальные значения модуля разности компонент поля $\{y_1, y_2\}$ в узлах сетки;
- перегруженную (**override**) версию виртуального метода **string ToString ()**, который возвращает строку с именем типа объекта и данными базового класса;
- реализацию абстрактного метода **string ToLongString (string format)**, который возвращает строку с такими же данными, что и метод **ToString ()**, и дополнительно информацию о каждом узле сетки – координату узла сетки и значения поля; параметр **format** задает формат вывода чисел с плавающей запятой.

Класс V1MainCollection

Класс **V1MainCollection** определяется как производный от класса **List<V1Data>**. Коллекция базового класса содержит элементы типа **V1DataList** и **V1DataArray**.

Класс **V1MainCollection** содержит

- индексатор типа **V1Data** (только с методом **get**) с параметром типа **string**; индексатор возвращает ссылку на элемент коллекции, если в коллекции есть элемент с заданным значением свойства **Key** базового класса и значение **null** в противном случае;
- открытый метод **bool Add (V1Data v1Data)**, который добавляет в коллекцию новый элемент (**V1DataList** или **V1DataArray**); элемент добавляется только в том случае, когда в коллекции нет элемента с такими же значениями свойств **Key** и **Date** в базовом классе, как в параметре **v1Data** (совпадают и **Key** и **Date**); метод возвращает значение **true**, если элемент был добавлен, и значение **false** в противном случае;
- конструктор **V1MainCollection (int nA, int nL)**, в котором в коллекцию добавляются **nA** элементов типа **V1DataArray** и **nL** элементов типа **V1DataList**; объекты типа **V1DataArray** и **V1DataList** создаются в самом конструкторе; для параметров конструкторов выбираются некоторые значения; конструктор используется для отладки;
- открытый метод **string ToLongString (string format)**, который возвращает строку с информацией о каждом элементе коллекции; при создании строки для каждого элемента коллекции вызывается метод **ToLongString (string format)**;
- перегруженную(**override**) версию виртуального метода **string ToString()**, который возвращает строку с информацией о каждом элементе коллекции; при создании строки для каждого элемента вызывается метод **ToString()**.

В методе **Main()**

1. Создать объект типа **V1DataList**, вывести его данные с помощью метода **ToLongString (string format)**. С помощью оператора преобразования, определенного в классе **V1DataList**, преобразовать его в объект типа **V1DataArray** и вывести данные **V1DataArray** с помощью метода **ToLongString (string format)**.
2. Создать объект типа **V1DataArray** и вывести значения индексатора типа **DataItem?** для двух значений индекса; одно значение индекса попадает в диапазон номеров сетки, другое находится вне диапазона.
3. Создать объект типа **V1MainCollection** и вывести данные объекта **V1MainCollection** с помощью метода **ToLongString (string format)**.
4. Для каждого элемента из **V1MainCollection** вывести значения свойств **xLength** и **MinMaxDifference**.
5. Для **V1MainCollection** вывести значения индексатора с индексом типа **string** для двух значений индекса – для одного значения в коллекции есть элемент с заданным значением, для другого нет.

Срок сдачи лабораторной работы:

15 октября – группы 302, 309

17 октября – группы 301, 341/2