

Лабораторная работа 1.

В работе определяется абстрактный базовый класс и два производных класса. В одном производном классе данные хранятся в массиве, в другом – в одном из типов коллекций стандартной библиотеки .NET. Привязка типов 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.

Типы, определенные в лабораторной работе, будут использоваться в других лабораторных работах, в том числе в следующем семестре.

Вариант 1

В лабораторной работе надо определить типы для работы с данными измерений комплексного значения электромагнитного поля на множестве точек на двумерной плоскости:

- **struct DataItem** для координат (x, y) точки и значения поля в этой точке;
- абстрактный базовый класс **V1Data** и два производных от него класса **V1DataList** и **V1DataArray**:
- **class V1DataList** для значений поля на множестве точек на двумерной плоскости, которые хранятся в виде коллекции **List<DataItem>**;
- **class V1DataArray** для значений поля на равномерной сетке, которые хранятся в прямоугольном двумерном массиве типа **System.Numerics.Complex[,]**;
- **class V1MainCollection** для коллекции объектов типа **V1DataList** и **V1DataArray**;
- статический класс с методами для вычисления значений поля в точке на двумерной плоскости.

Структура **DataItem** содержит открытые автореализуемые свойства

- типа **double** с координатой x двумерной точки;
- типа **double** с координатой y двумерной точки;
- типа **System.Numerics.Complex** для комплексного значения поля в этой точке.

В структуре **DataItem** определены открытые

- конструктор с параметрами типа **double** (координата x), **double** (координата y), **Complex** (значение поля) для инициализации данных структуры;
- метод **string ToLongString(string format)**, возвращающий строку, которая содержит значения координат точки измерения, комплексное значение поля и значение модуля поля; параметр **format** задает формат вывода чисел с плавающей запятой;
- перегруженная (**override**) версия виртуального метода **string ToString()**.

Определить делегат **Complex FdblComplex (double x, double y)**.

Абстрактный базовый класс **V1Data** имеет открытые автореализуемые свойства

- типа **string** с методом **get** для идентификации объекта данных;
- типа **DateTime** с методом **get** для даты измерений поля.

Класс **V1Data** содержит открытые

- конструктор с параметрами типа **string** и **DateTime**;
- абстрактное свойство **Count** типа **int** (только с методом **get**);
- абстрактное свойство **AverageValue** типа **double** (только с методом **get**);
- абстрактный метод **string ToLongString(string format)**;
- перегруженную (**override**) версию виртуального метода **string ToString()**;

Класс **V1DataList** является производным от класса **V1Data**. В классе **V1DataList** данные измерений хранятся в коллекции **List<DataItem>**. Каждый элемент коллекции **DataItem** содержит координаты точки, в которой измерено поле, и комплексное значение поля.

В классе **V1DataList** определить открытые

- автореализуемое свойство типа **List<DataItem>** с методом **get**;
- конструктор с параметрами типа **string** и **DateTime** для инициализации данных базового класса; в этом конструкторе распределяется память для **List<DataItem>**;
- метод **bool Add(DataItem newItem)** для добавления в список **List<DataItem>** нового элемента **newItem**; метод добавляет новый элемент только в том случае, когда в коллекции **List<DataItem>** нет элемента, у которого координаты точки, в которой измерено поле, совпадают с координатами точки измерения элемента **newItem**; в этом случае метод возвращает значение **true**; в противном случае новый элемент не добавляется и возвращается значение **false**;
- метод **int AddDefaults(int nItems, FdblComplex F)**, в котором (по какому-нибудь алгоритму) выбираются **nItems** точек на плоскости, для этих точек с помощью метода **F** вычисляются комплексные значения поля, создаются элементы **DataItem** и добавляются в список **List<DataItem>**; при добавлении элементов **DataItem** надо использовать метод **bool Add(DataItem newItem)**; метод возвращает число элементов, добавленных в список **List<DataItem>**; число добавленных элементов может быть меньше, чем **nItems**, если в списке уже есть данные для выбранных по умолчанию точек измерения;
- реализацию абстрактного свойства **Count**, которое возвращает число точек, в которых измерено поле (число элементов в **List<DataItem>**);
- реализацию абстрактного свойства **AverageValue** типа **double**, которое возвращает среднее значение модуля;
- перегруженную (**override**) версию виртуального метода **string ToString()**, который возвращает строку с именем типа, 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>** – координаты точки измерения,

комплексное значение поля и значение модуля поля; параметр **format** задает формат вывода чисел с плавающей запятой.

Класс **V1DataArray** является производным от класса **V1Data**. В классе **V1DataArray** данные измерений на двумерной прямоугольной сетке с равномерным шагом по осям **Ox** и **Oy** хранятся в двумерном прямоугольном массиве типа **Complex[,]**. В классе хранятся параметры сетки – шаг по оси **Ox**, шаг по оси **Oy**, число узлов по оси **Ox**, число узлов по оси **Oy**. Предполагается, что координаты первых узлов сетки по осям **Ox** и **Oy** равны нулю.

Класс **V1DataArray** содержит открытые

- автореализуемые свойства для параметров сетки с методом **get** – типа **int** для числа узлов по оси **Ox**, типа **int** для числа узлов по оси **Oy**, типа **double** для шага сетки по оси **Ox**, типа **double** для шага сетки по оси **Oy**;
- автореализуемое свойство с методом **get** типа **Complex [,]** для двумерного прямоугольного массива значений поля в узлах сетки;
- конструктор с параметрами типа **string**, **DateTime** для инициализации данных базового класса; в этом конструкторе распределяется память для массива **Complex [,]** с нулевым числом элементов;
- конструктор с параметрами типа
 - **string**, **DateTime** для инициализации данных базового класса;
 - **int** (число узлов по оси **Ox**), **int** (число узлов по оси **Oy**), **double** (шаг по оси **Ox**), **double** (шаг по оси **Oy**) для инициализации сетки;
 - **FdblComplex** для вычисления комплексных значений поля в узлах сетки;в конструкторе распределяется память для массива **Complex [,]** и для каждого узла сетки вызывается метод типа **FdblComplex**, который вычисляет комплексное значения поля в узле;
- реализацию абстрактного свойства **Count**, которое возвращает число точек, в которых измерено поле (число узлов сетки);
- реализацию абстрактного свойства **AverageValue** типа **double**, которое возвращает среднее значение модуля;
- перегруженную (**override**) версию виртуального метода **string ToString ()**, который возвращает строку с именем типа, данными базового класса и параметрами сетки;
- 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 – координаты, значение поля и значение модуля поля; параметр **format** задает формат вывода чисел с плавающей запятой;
- оператор преобразования типа **V1DataArray** к типу **V1DataList**; в коде оператора преобразования для каждого узла сетки создается объект типа **DataItem**, который добавляется в список **List<DataItem>** класса **V1DataList**.

Класс **V1MainCollection** содержит коллекцию элементов **V1DataList** и **V1DataArray**.

Класс **V1MainCollection** содержит

- закрытое поле типа **List<V1Data>**;

- открытое свойство **Count** типа **int**, которое возвращает число элементов в списке **List<V1Data>**;
- открытый индексатор типа **V1Data** с целочисленным индексом (только с методом **get**), который возвращает элемент списка **List<V1Data>** с заданным индексом;
- открытый метод **bool Contains (string ID)**, который возвращает значение **true**, если среди элементов списка **List<V1Data>** есть элемент, у которого значение свойства типа **string** для идентификации объекта **V1Data** равно ID и **false** в противном случае;
- открытый метод **bool Add (V1Data v1Data)** для добавления в коллекцию **List<V1Data>** нового элемента; метод добавляет новый элемент только в том случае, когда в коллекции **List<V1Data>** нет элемента с таким же значением свойства типа **string** для идентификации объекта, как и у значения параметра **v1Data** метода; метод возвращает значение **true**, если элемент был добавлен, и значение **false** в противном случае;
- открытый метод **string ToLongString (string format)**, который возвращает строку с информацией о каждом элементе списка **List<V1Data>**; при создании строки для каждого элемента из списка **List<V1Data>** вызывается метод **ToLongString (string format)**;
- перегруженную(**override**) версию виртуального метода **string ToString()**, который возвращает строку с информацией о каждом элементе списка **List<V1Data>**; при создании строки для каждого элемента из списка **List<V1Data>** вызывается метод **ToString()**.

Статический класс содержит статические методы, отвечающие делегату **FdblComplex**. Эти методы используются как фактические значения параметров типа **FdblComplex** в методе **AddDefaults(int nItems, FdblComplex F)** класса **V1DataList** и в конструкторе класса **V1DataArray**.

В методе **Main()**

1. Создать объект типа **V1DataArray**, вывести его данные с помощью метода **ToLongString (string format)**. С помощью оператора преобразования, определенного в классе **V1DataArray**, преобразовать его в объект типа **V1DataList**, вывести данные **V1DataList** с помощью метода **ToLongString (string format)**. Для исходного объекта **V1DataArray** и для объекта **V1DataList**, полученного в результате преобразования, вывести значения свойств **Count** и **AverageValue**.
2. Создать объект типа **V1MainCollection**, с помощью метода **bool Add (V1Data v1Data)** добавить в коллекцию два элемента типа **V1DataArray** и два элемента типа **V1DataList** и вывести данные объекта **V1MainCollection** с помощью метода **ToLongString (string format)**.
3. Для всех элементов из **V1MainCollection** вызвать свойства **Count** и **AverageValue** и вывести их значения.

Срок сдачи лабораторной работы
12 октября