

Лабораторная работа 2. Вариант 2

Наследование. Интерфейсы. Делегаты

В лабораторной работе определяются классы, которые дополнительно к данным, заданным на равномерной прямоугольной сетке (как в лабораторной работе 1), содержат данные, заданные на некотором множестве точек с координатами (x, y) . Точки из дополнительного множества не совпадают с узлами прямоугольной сетки.

Как и в лабораторной работе 1 декартовы координаты узла равномерной прямоугольной сетки $\{x_i, y_j\}, i = 0, 1, \dots, N, j = 0, 1, \dots, M$ вычисляются по формуле $x_i = ih_x, y_j = jh_y$. h_x – заданный шаг сетки по оси Ox , h_y – по оси Oy .

В лабораторной работе определяются следующие типы:

- **struct DataItemD** для хранения данных, связанных с одним узлом прямоугольной сетки;
- класс **V2JDataArray**; в классе **V2JDataArray** данные хранятся в двумерном ступенчатом (jagged) массиве типа **(double, double)[,][,]**;
- класс **V2JList**, производный от класса **V2JDataArray**; в базовом классе хранятся данные, заданные на равномерной прямоугольной сетке; данные для дополнительных точек хранятся в коллекции **System.Collections.Generic.List<DataItemD>** производного класса;
- класс **V2JDict**, который содержит ссылку на класс **V2JDataArray**; данные для дополнительных точек хранятся в коллекции **System.Collections.Generic.Dictionary<(double, double), DataItemD>**; координаты (x, y) дополнительных точек используются как ключ в коллекции-словаре **Dictionary<(double, double), DataItemD>**;
- статический класс **Funcs**, который содержит статические функции для инициализации данных, отвечающих делегатам, которые являются параметрами конструкторов классов **V2JDataArray**, **V2JList** и **V2JDict**;
- **interface IDataInfo**
{ string Key {get; set;}
int Count {get;}
double MaxDifference {get;}
DataItemD Farthest (double x, double y);
}

Классы **V2JDataArray**, **V2JList** и **V2JDict** реализуют интерфейсы **IDataInfo** и **System.Collections.Generic.IEnumerable<DataItemD>**.

Структура **DatalItemD** (из лабораторной работы 1 без изменений)

Структура **DatalItemD** содержит открытые автореализуемые свойства

- типа **double** с координатой *x* узла сетки;
- типа **double** с координатой *y* узла сетки;
- типа **(double, double)** для измеренных данных в узле сетки.

В структуре **DatalItemD** определены открытые

- конструктор
DatalItemD (double x, double y, (double, double) v) для инициализации данных структуры;
- перегруженная (**override**) версия виртуального метода **string ToString()**, возвращающего строку, которая содержит координаты узла сетки и измеренные данные.

Класс **V2JDataArray** (новая версия).

В новой версии класса измеренные данные хранятся в двумерном ступенчатом (jagged) массиве типа **(double, double) [][[]]**.

Класс **V2JDataArray** содержит поле типа **(double, double) [][[]]** с типом доступа **protected**.

Класс **V2JDataArray** содержит открытые

- автореализуемое свойство **Key** типа **string**;
- автореализуемое свойство **XYgrid** типа **(int, double, int, double)** с методами **get** и **private set**; первые два элемента кортежа содержат число узлов сетки и шаг сетки по оси *Ox*; следующие два элемента кортежа – число узлов и шаг сетки по оси *Oy*;
- конструктор
V2JDataArray (string key, (int, double, int, double) XYgrid, System.Func<double, double, (double, double)> FJ);
в конструкторе распределяется память для массива **(double, double) [][[]]**, в котором хранятся данные, и элементам массива присваиваются значения, которые вычисляются с помощью функции **FJ**;
- перегруженную (**override**) версию виртуального метода **string ToString()**; метод возвращает строку со значением свойства **Key** и параметрами сетки;

- виртуальный метод **virtual string ToLongString()**; метод возвращает строку с такой же информацией, как и метод **ToString()**, и дополнительно для каждого узла равномерной сетки выводятся значения индексов узла сетки, координаты узла сетки и значения компонент **(double, double)** в узле сетки.

Класс **V2JDataArray** реализует интерфейсы **IDataInfo** и **IEnumerable<DataItemD>**.

Реализация интерфейса **IDataInfo**:

- **string Key {get; set;}**;
- **int Count {get;}**; свойство возвращает число узлов сетки;
- **double MaxDifference {get;}**; возвращает максимальное значение модуля разности двух компонент измеренных данных (для узла сетки) массива **(double, double)[][]**;
- **DataItemD Farthest (double x, double y)**; среди всех узлов сетки выбирается узел, который находится на максимальном расстоянии от точки с координатами **(x, y)**; метод возвращает **DataItemD** с координатами узла и значением **(double, double)** в этом узле.

Реализация интерфейса **IEnumerable<DataItemD>**:

В классе **V2JDataArray** реализация интерфейса **IEnumerable<DataItemD>** перечисляет все данные на сетке как экземпляры **DataItemD** – для каждого узла сетки создается экземпляр **DataItemD** с координатами **(x, y)** узла сетки и значениями данных **(double, double)** в этом узле.

Класс V2JList

Класс **V2JList** определяется как производный от класса **V2JDataArray**.

Класс **V2JList** содержит закрытое поле типа **List <DataItemD>**.

Класс **V2JList** содержит открытые

- конструктор

**V2JList (string key, (int, double, int, double) XYgrid,
System.Func<double, double, (double, double)> FJ,
int nList, System.Func<int, DataItemD> FL);**

параметры **key**, **XYgrid**, **FJ** используются при инициализации данных базового класса; в конструкторе распределяется память для коллекции **List <DataItemD>** с **nList** элементами, элементам коллекции присваиваются значения, которые вычисляются с помощью функции **FL**;

- метод **void Add(DataItemD item)**, который добавляет в коллекцию **List <DataItemD>** новый элемент **item**;
- перегруженную (**override**) версию виртуального метода **string ToString()**; метод возвращает строку со значением свойства **Key**, параметрами сетки и числом элементов в списке **List <DataItemD>**;
- перегруженную (**override**) версию виртуального метода **string ToLongString()**; метод возвращает строку с такой же информацией. как и метод **ToString()**, дополнительно для каждого узла равномерной сетки выводятся значения индексов узла сетки, координаты узла сетки и значения компонент **(double, double)** в узле сетки и, кроме того, информация для каждого элемента списка **List <DataItemD>**.

Класс **V2JList** реализует интерфейсы **IDataInfo** и **IEnumerable<DataItemD>**.

Реализация интерфейса **IDataInfo**:

- **string Key {get; set;}**;
- **int Count {get;}**; свойство возвращает число узлов прямоугольной сетки плюс число элементов в коллекции **List <DataItemD>**;
- **double MaxDifference {get;}**; возвращает максимальное значение модуля разности двух компонент измеренных данных (для узла сетки) среди всех элементов массива **(double, double)[][]** и коллекции **List <DataItemD>**;
- **DataItemD Farthest (double x, double y)**; среди всех узлов сетки и координат узлов из **List <DataItemD>** выбирается узел, который находится на максимальном расстоянии от точки с заданными координатами **(x, y)**; метод возвращает **DataItemD** с координатами узла и значениями данных **(double, double)** в этом узле.

Реализация интерфейса **IEnumerable<DataItemD>**:

В классе **V2JList** реализация интерфейса **IEnumerable<DataItemD>** перечисляет все данные на сетке и в коллекции **List <DataItemD>** как экземпляры **DataItemD**.

Класс V2JDict

Класс **V2JDict** содержит закрытые поля типов

- **V2JDataArray**;
- **Dictionary < (double, double), DataItemD >**.

Класс **V2JDict** содержит открытые

- конструктор

**V2JDict (string key, (int, double, int, double) XYgrid,
System.Func<double, double, (double, double)> FJ,
int nDict, System.Func<int, ((double, double), DataItemD)> FD);**

в конструкторе распределяется память для **V2JDataArray** и для коллекции **Dictionary <(double, double), DataItemD>** с **nDict** элементами; параметры **key, XYgrid, FJ** используются как параметры конструктора **V2JDataArray**; коллекция **Dictionary <(double, double), DataItemD>** инициализируется с помощью функции **FD**;

- метод **void Add(DataItemD item)**, который добавляет в коллекцию **Dictionary <(double, double), DataItemD>** новый элемент **item** с ключом **(x,y)**, **x,y** – координаты узла в элементе **item**;
- перегруженную (**override**) версию виртуального метода **string ToString()**; метод возвращает строку со значением свойства **Key**, параметрами сетки и числом элементов в **Dictionary <(double, double), DataItemD >**;
- метод **string ToLongString()**; метод возвращает строку с такой же информацией, как и метод **ToString()**, дополнительно для каждого узла равномерной сетки выводятся значения индексов узла сетки, координаты узла сетки и значения компонент **(double, double)** в узле сетки и, кроме того, информация для каждого элемента коллекции **Dictionary <(double, double), DataItemD>**.

Класс **V2JDict** реализует интерфейсы **IDataInfo** и **IEnumerable<DataItemD>**.

Реализация интерфейса **IDataInfo**:

- **string Key {get; set;}**;
- **int Count {get}**; свойство возвращает число узлов прямоугольной сетки плюс число элементов в коллекции **Dictionary <(double, double), DataItemD>**;
- **double MaxDifference {get}**; возвращает максимальное значение модуля разности двух компонент измеренных данных (для узла сетки) среди всех элементов массива **(double, double)[][]** и коллекции **Dictionary <(double, double), DataItemD>**;
- **DataItemD Farthest (double x, double y)**; среди всех узлов сетки и координат узлов из **Dictionary <(double, double), DataItemD>** выбирается узел, который находится на максимальном расстоянии от точки с заданными координатами **(x, y)**; свойство возвращает **DataItemD** с координатами узла и значениями данных **(double, double)** в этом узле.

Реализация интерфейса **IEnumerable<DataItemD>**:

В классе **V2JDict** реализация интерфейса **IEnumerable<DataItemD>** перечисляет все данные на сетке и в коллекции **Dictionary <(double, double), DataItemD>** как экземпляры **DataItemD**.

В методе **Main()**

1. Создать объект типа **V2JDataArray**, вывести его данные с помощью метода **ToLongString ()**.
2. Вывести все данные объекта **V2JDataArray**, перечислив их с помощью интерфейса **IEnumerable<DataItemD>**. Вывести значения свойств **Key, Count, MaxDifference** и функции **Farthest**.
3. Создать объект типа **V2JList**, вывести его данные с помощью метода **ToLongString ()**.
4. Вывести все данные объекта **V2JList**, перечислив их с помощью интерфейса **IEnumerable<DataItemD>**. Вывести значения свойств **Key, Count, MaxDifference** и функции **Farthest**.
5. Создать объект типа **V2JDict**, вывести его данные с помощью метода **ToLongString ()**.
6. Вывести все данные объекта **V2JDict**, перечислив их с помощью интерфейса **IEnumerable<DataItemD>**. Вывести значения свойств **Key, Count, MaxDifference** и функции **Farthest**.
7. Создать массив типа **IDataInfo[3]**; присвоить элементам массива ссылки на объекты **V2JDataArray, V2JList, V2JDict**. В цикле для всех элементов массива **IDataInfo[3]** вывести значения свойств **Key, Count, MaxDifference** и функции **Farthest**.

Срок сдачи лабораторной работы:

11 ноября – группы 302, 309

7 ноября – группы 301, 341/2