

Лабораторная работа 2. Вариант 2

Дополнительные возможности в механизме привязки данных в Windows Presentation Foundation

В классе **V2DataCollection** надо реализовать интерфейс **System.Collections.Specialized.INotifyCollectionChanged** и добавить метод **void Add(DataItem item)**. Метод **void Add(DataItem item)** добавляет элемент **item** в список **List<DataItem>** и бросает событие **CollectionChanged**.

В пользовательский интерфейс приложения для работы с коллекцией **V2MainCollection** из лабораторной работы 1 надо добавить

- элементы управления для ввода данных для нового элемента **DataItem** в коллекцию **List<DataItem>** элемента **V2DataCollection**, который пользователь выбрал в элементе управления **ListBox** с представлением коллекции;
- проверку корректности данных в привязке с использованием интерфейса **IDataErrorInfo**;
- команды.

В меню главного окна приложения надо добавить элемент **Add DataItem**.

В главное окно приложения добавить кнопки **Open**, **Save**, **Add DataItem** и **Remove** (или элемент управления **ToolBar** с кнопками **Open**, **Save**, **Add DataItem** и **Remove**).

В главное окно приложения добавить элементы управления для ввода данных для нового элемента **DataItem**:

- **TextBox** для координаты x точки измерения поля;
- **TextBox** для координаты y точки измерения поля;
- **TextBox** для действительной части значения поля;
- **TextBox** для мнимой части значения поля.

Можно вместо отдельных элементов **TextBox** для каждой из координат точки измерения поля использовать только один элемент **TextBox**, в который пользователь через разделитель вводит оба значения. В этом случае в программе необходимо определить и использовать в привязке преобразователь типа.

В проект **WPF** добавить класс, в котором надо определить свойства для использования в привязках к элементам управления для ввода данных для элемента **DataItem**.

В классе надо определить конструктор с параметром типа **V2DataCollection** (ссылка на элемент, который выбрал пользователь).

В этом классе надо реализовать интерфейс **System.IDataErrorInfo** для проверки корректности данных в привязках и интерфейс

System.ComponentModel.INotifyPropertyChanged. Когда происходит событие **PropertyChanged**, в элементах управления с привязкой к соответствующему свойству обновляются данные. Событие **PropertyChanged** интерфейса **INotifyPropertyChanged** дает возможность выполнить проверку корректности данных в интерфейсе **System.IDataErrorInfo** в случае, когда допустимость значения одного свойства зависит от значения других свойств.

В классе надо определить открытый метод для создания и добавления нового элемента **Dataltem** в список **List<Dataltem>** элемента **V2DataCollection**. При создании **Dataltem** используются значения свойств, участвующих в привязках к элементам управления **TextBox** для ввода данных.

Проверка корректности данных в привязке

Реализация интерфейса **IDataErrorInfo** в классе должна проверять, что

- точка измерения поля в новом элементе **Dataltem** не совпадает ни с одной точкой измерения поля из списка **List<Dataltem>** элемента **V2DataCollection**, который был передан в класс через параметр конструктора;
- модуль значения поля больше 0.

Команды

С элементом меню и кнопкой **Open** надо связать команду **ApplicationCommands.Open**.

- В обработчике события **Execute** команды проверяется, что пользователь сохранил объект **V2MainCollection**, и десериализуется новый объект **V2MainCollection** из файла, который пользователь выбрал в стандартном диалоге **OpenFileDialog**.

С элементом меню и кнопкой **Save** надо связать команду **ApplicationCommands.Save**.

- В обработчике события **CanExecute** команды проверяется значение булевского поля класса **V2MainCollection** с информацией о том, что в процессе работы приложения пользователь изменил данные объекта **V2MainCollection**.
- В обработчике события **Execute** команды коллекция **V2MainCollection** сериализуется в файл, который пользователь выбрал в стандартном диалоге **SaveFileDialog**.

С элементом меню и кнопкой **Remove** надо связать команду **ApplicationCommands.Delete**.

- В обработчике события **CanExecute** команды **ApplicationCommands.Delete** проверяется, что в элементе управления **ListBox** со всей коллекцией **V2MainCollection** есть выбранный элемент.
- В обработчике события **Execute** команды из коллекции удаляется элемент, выбранный пользователем в элементе управления **ListBox** со всей коллекцией.

Определить пользовательскую команду и связать ее с элементом меню **Add Dataltem** и кнопкой **Add Dataltem**.

- Обработчик события **CanExecute** команды должен проверить, что введенные пользователем данные для нового элемента **DataItem** не содержат ошибок. Проверяется корректность данных в привязках элементов управления к свойствам, которые контролируются в **IDataErrorInfo**.
- Обработчик события **Execute** команды добавляет новый элемент **DataItem** в список **List<DataItem>** элемента **V2DataCollection**, который выбрал пользователь.

Сохранение данных

Как и в первой лабораторной работе информация о том, что в процессе работы пользователь изменил коллекцию **V2MainCollection**, хранится в поле булевского типа, которому присваивается значение **true** каждый раз, когда пользователь изменяет объект **V2MainCollection** и значение **false** после сохранения (сериализации) данных объекта **V2MainCollection** в файл.

Если перед выбором элементов меню **New** или **Open** или перед выходом из приложения пользователь изменил коллекцию **V2MainCollection** и не сохранил их (не сериализовал в файл), он получает предупреждение о том, что данные будут потеряны. Предупреждение выводится с помощью стандартного диалога **System.Windows.MessageBox**, в котором пользователю предлагается выбрать – сохранить в файл измененные данные или выполнить соответствующую операцию без сохранения результатов. Если пользователь выбрал сохранение данных, то вызывается стандартный диалог **Microsoft.Win32.SaveFileDialog** для выбора имени файла, в который будут сериализованы данные объекта **V2MainCollection**.

Обработка исключений

Все исключения, которые могут возникать при обработке некорректного ввода пользователя, должны обрабатываться приложением.

Независимо от того, корректно были введены данные или при вводе были допущены ошибки, все файловые потоки должны быть закрыты.

Приложение должно оставаться в рабочем состоянии до тех пор, пока пользователь не закроет главное окно приложения.

Срок сдачи лабораторной работы 16 апреля