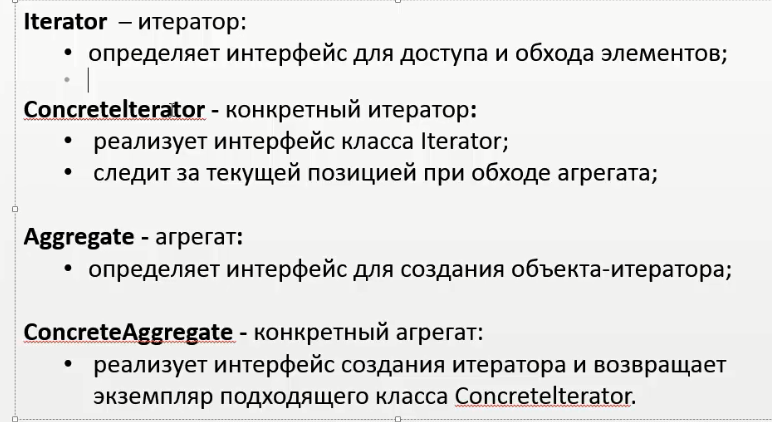
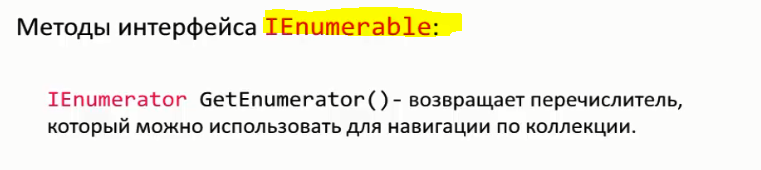


Коллекция является агрегатом и у любой коллекции мы можем получить итератор

ПАТЕРН Iterator





Т.е. возвращает ссылку на объект реализующий IEnumerator (в моих примерах это класс и передается this приведенная или нет к IEnumerator)

Можно осуществить перебор элементов

коллекции с помощью foreach без реализации интерфейсов IEnumerator и IEnumerable, для этого необходимо чтобы класс КОЛЛЕКЦИЯ содержал в себе открытые методы интерфейсов:

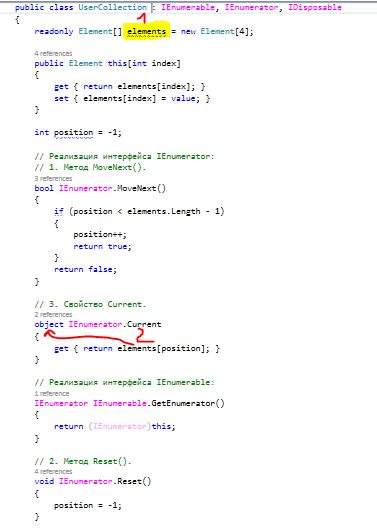
IEnumerator (public bool MoveNext(),public void Reset(), свойство - public object Current)

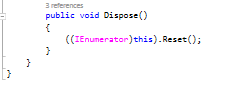
IEnumerable (public UserCollection GetEnumerator() – где имя UserCollection имя класса коллекции)

В ДАННОЙ пользовательской коллекции имеются недостатки:

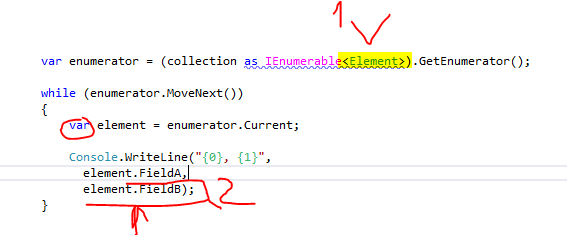
1.строго заданный тип элемента

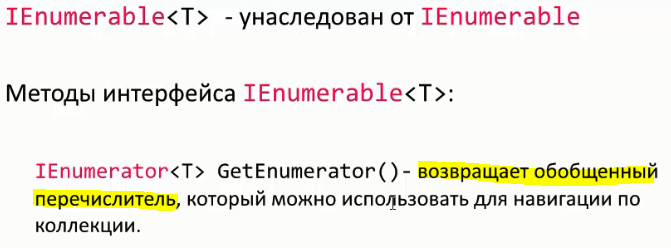
2.при возврате элемента он приводится (происходит UpCast) к типу object, а после DownCast к нужному типу

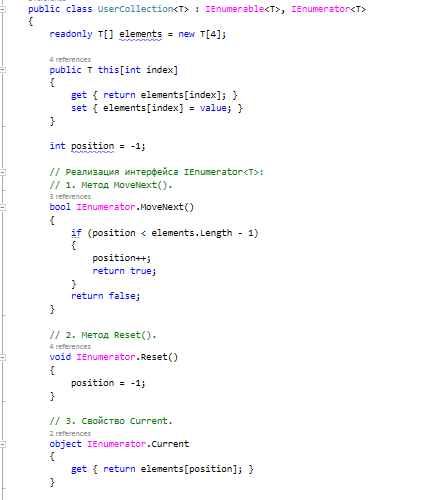


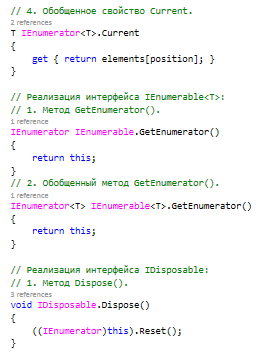


В случае если метод IEnumerator.Current возвращает object, то коллекция можно привести к параметризированному типу 1. где <УКАЗАНИЕ ТИПА ЭЛЕМЕНТА>, и тогда не нужно будет применять DownCast в возвращающем значении метода IEnumerator.Current 2.



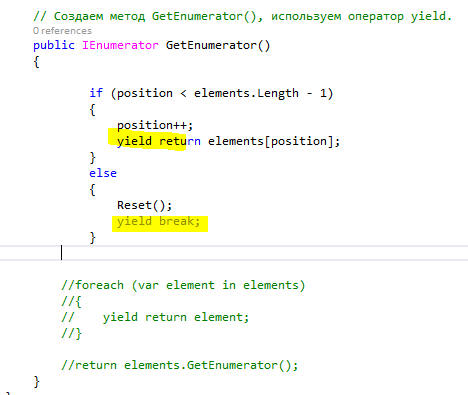




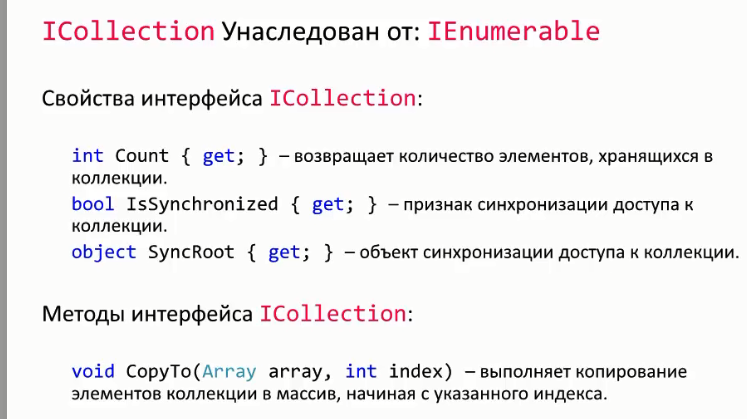


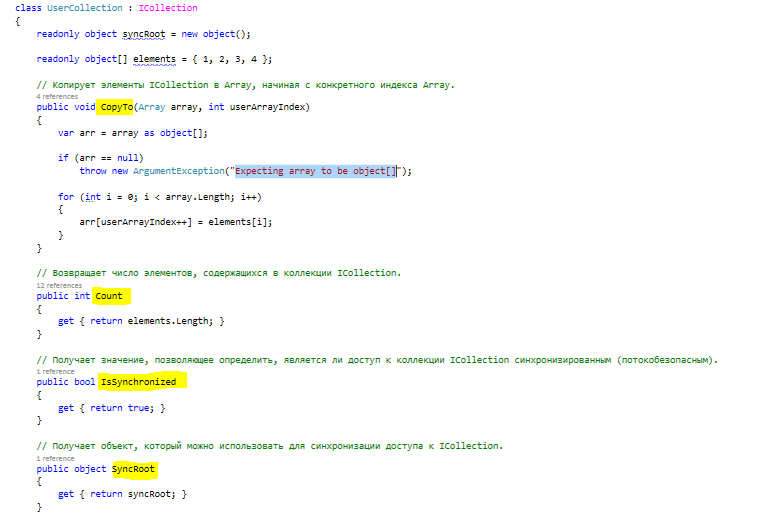
Возможно осуществить перечисления не реализовывая интерфейсы IEnumerator и IEnumerable, а в методе public IEnumerator GetEnumerator() использовать оператор yield

Метод должен обязательно возвращать IEnumerator

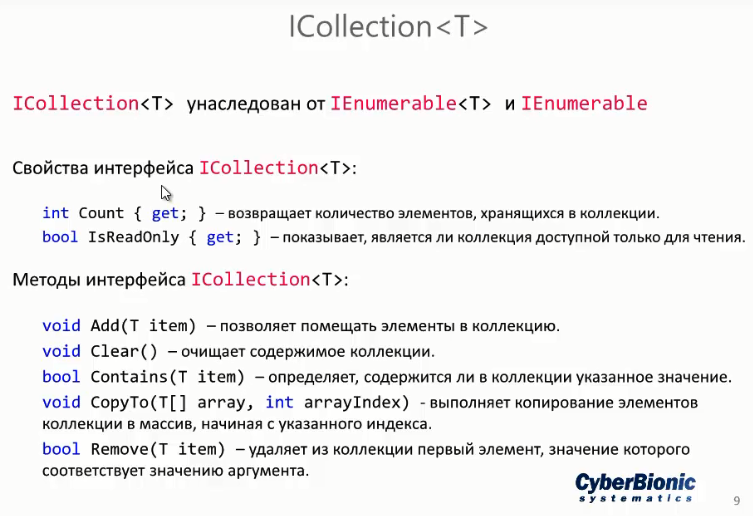


Интерфейс ICollection





Параметризированная версия интерфейса ICollection<T>



Если имеется в пользовательской коллекции имеется метод public void Add(T item)то при инициализации коллекции можно использовать блок инициализации для добавления элементов



Плюс заключается в том что объект создается с проинициализированным массивом элементом (пустая коллекция не создастся и если кто то (подписан на создание экземпляра коллекции) будет работать с другого потока то он будет работать с элементами а не с пустышкой, а при Add(T item) элементы добавляются поочередно)

