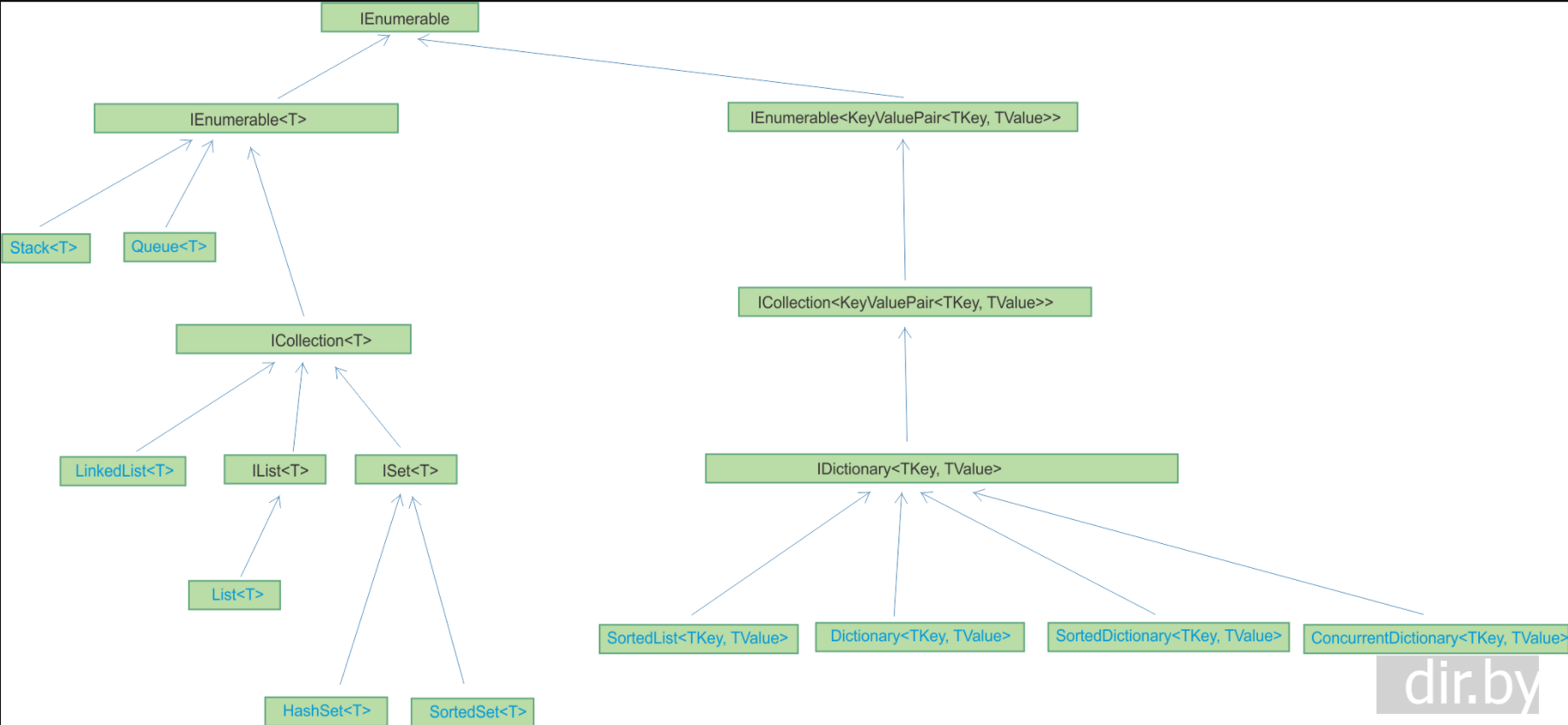
**Коллекция** **в** **C#** - это группа связанных объектов, хранящихся в структуре.

Иерархия



Во главе стоит IEnumerable, Интерфейс имеет метод, возвращающий ссылку на другой интерфейс - перечислитель. А интерфейс IEnumerator определяет функционал для перебора внутренних объектов в контейнере

Метод MoveNext() перемещает указатель на текущий элемент на следующую позицию в последовательности. Свойство Current возвращает объект в последовательности, на который указывает указатель. Метод Reset() сбрасывает указатель позиции в начальное положение.

К IEnumerable идет список IEnumerable<T> и обобщенный список IEnumerable<KeyValuePair<TKey, TValue>>. IEnumerable<T> предоставляет перечислитель. Содержит метод GetEnumerator(), который возвращает перечислитель. Имеется свойство Current, которое получает текущий элемент в коллекции. Методы: Dispose() - освобождение, удаление неуправляемых ресурсов; MoveNext() - Перемещает перечислитель к следующему элементу коллекции; Reset() - Устанавливает перечислитель в его начальное положение, т. е. перед первым элементом коллекции.

IEnumerable<T> состоит из Stack<T>, Queue<T>, ICollection<T>. Stack<T> это класс, который использует алгоритм "последний вошел - первый вышел". Первым извлекается элемент, который находится в самом верху стека. Stack<T> относится к типизированной коллекции, то есть все элементы одного типа. Методы: Clear(); Contains(); CopyTo(); Push(); ToArray(). Queue<T> это класс, который использует алгоритм "первый вошел - первый вышел". Каждый добавленный элемент помещается в конец. Первым извлекается элемент, который находится в начале. ICollection<T> это коллекция объектов. У этой коллекции можно узнать размер, скопировать элементы, удалить элемент, добавить элемент, очистить всю коллекцию. Свойства: Count - Получает число элементов, содержащихся в коллекции, IsReadOnly -Получает значение, указывающее, является ли коллекция доступная только для чтения.Методы: Add - Добавляет элемент в коллекцию, Clear - удаляет все элементы из коллекции, Contains - Определяет, содержит ли коллекция указанное значение, CopyTo - Копирует элементы коллекции в массив Array, начиная с указанного индекса массива Array, Remove - Удаляет первый найденный объект из коллекции.

ICollection<T> состоит из LinkedList<T>, IList<T>, ISet<T>. IList<T> это коллекция объектов. У этой коллекции можно получить элемент по индексу, получить индекс по элементу, удалить элемент, вставить элемент. Свойства this - Возвращает или задает элемент по указанному индексу, методы: IndexOf - Определяет индекс заданного элемента в коллекции, Insert - Вставляет элемент по указанному индексу в коллекцию, RemoveAt - Удаляет элемент, расположенный по указанному индексу из коллекции. ISet<T> это уникальный набор объектов. У этого набора есть методы: объединение, пересечение, разность и другие методы для работы с множествами как в математике. Методы: Add - Добавляет элемент в текущий набор и возвращает значение, указывающее, что элемент был добавлен успешно, ExceptWith - Удаляет элементы из текущего набора, которые совпадают с элементами в переданной коллекции, IntersectWith - Удаляются элементы в текущем наборе, IsProperSubsetOf - Определяет, является ли текущий набор подмножеством заданной коллекции, SetEquals - Определяет, содержат ли текущий набор и переданная коллекция одни и те же элементы. LinkedList<T> это двухсвязный список, в котором каждый элемент хранит 2 ссылки: ссылка на следующий элемент, ссылка на предыдущий элемент.

IList<T> содержит List<T>. List<T> это класс, который реализовывает массив с динамическим добавлением и удалением элементов, обращение по индексу. Так же есть методы поиска и сортировки элементов.

ISet<T> содержит HashSet<T> и SortedSet<T>. HashSet<T> содержит неповторяющуюся коллекцию неупорядоченных элементов. HashSet<T> имеет методы: UnionWith, IntersectWith (пересечение элементов с другим HashSet<T>), ExceptWith (разность элементов с другим HashSet<T>), SymmetricExceptWith (симетрическая разность элементов с другим HashSet<T>). SortedSet<T> содержит неповторяющуюся коллекцию упорядоченных элементов. SortedSet<T> - имеет методы Add, Remove, Contains, UnionWith (объединение), IntersectWith (пересечение), ExceptWith (разность), SymmetricExceptWith (симетрическая разность)

IEnumerable<KeyValuePair<TKey, TValue>> содержит ICollection<KeyValuePair<TKey,TValue>>, которая включает в себя IDictionary<TKey,TValue>. IDictionary это коллекция пар (ключ - значение). У этой коллекции можно получить значение по ключу, удалить элемент, добавить элемент. Она состоит из SortedList<TKey,TValue>, Dictionary<TKey,TValue>, SortedDictionary<TKey,TValue>, ConcurrentDictionary<TKey,TValue>. Dictionary<TKey,TValue> класс храняющий набор пар "ключ-значение". SortedDictionary<TKey,TValue> - Это коллекция пар "ключ-значение", упорядоченных по ключу. ConcurrentDictionary<TKey,TValue> - Это потокобезопасная коллекция пар "ключ-значение", доступ к которой могут одновременно получать несколько потоков.

Типы коллекций

Различают три вида коллекций:

* необобщенные;
* обобщенные;
* специальные.

Необобщенные коллекции

1. **List** – класс коллекции, предоставляющий доступ к элементам по индексу. Содержит методы изменения коллекции, сортировки элементов и поиска по элементам.
2. **Dictionary** – класс коллекции, в основе которой лежат пара ключ/значение. Элементы упорядочены по ключу. Доступ к элементам производится через ключ.
3. **Queue** – класс коллекции, реализующий модель FIFO (первым пришел – первым ушел).
4. **Stack** – класс коллекции, реализующий модель LIFO (последним пришел – первым вышел).
5. **SortedList** – класс коллекции, дополняющий функциональность типа List возможностью реализовать свой механизм сортировки путем реализации класса интерфейса IComparer. Более подробно каждый тип коллекции мы рассматриваем в соответствующих разделах.

Обобщенные коллекции

1. Queue - класс коллекции, реализующий модель FIFO (первым пришел – первым ушел).
2. Stack – класс коллекции, реализующий модель LIFO (последним пришел – первым вышел).
3. ArrayList – динамический массив, способен увеличивать размер при добавлении нового элемента.
4. HashTable – класс коллекции, в основе которой лежат пара ключ/значение. Значения упорядочены по хэш-коду.

**Наблюдаемые коллекции ObservableCollection<T>**.

ObservableCollection<T> - наблюдаемая коллекция, которая изменяется со срабатыванием события. Пользовательский интерфейс получает информацию об изменениях коллекции, унаследован от Collection<T>, использует внутри себя List<T>, INotifyCollectionChanged

Методы базового класса SetItem() и RemoveItem() переопределены для инициации события CollectionChanged. Клиенты этого класса могут регистрироваться на это событие, используя интерфейс INotifyCollectionChanged.

CollectionChanged

Событие CollectionChanged предоставляется интерфейсом INotifyCollectionChanged, оно должно вызываться при каждом изменении базовой коллекции.

Словари

Словарь (dictionary) представляет собой сложную структуру данных, позволяющую обеспечить доступ к элементам по ключу. Ключи должны быть с универсальными названиями. Главное свойство словарей — быстрый поиск на основе ключей. Можно также свободно добавлять и удалять элементы, подобно тому, как это делается в List<T>, но без накладных расходов производительности, связанных с необходимостью смещения последующих элементов в памяти.

Реализуются интерфейсы IDictionary, IDictionary<TKey, TValue>, ICollection, ICollection<KeyValuePair<TKey, TValue>>, IEnumerable, IEnumerable<KeyValuePair<TKey, TValue>>, ISerializable и IDeserializationCallback.

Методы: Add(), ContainsKey(), ContainsValue(), Remove().

Свойства: Comparer, Keys, Values.

Множество списков

Список может быть объявлен следующим образом:

var someList = new List<[тип данных]>();

// Объявление нового списка.

var someList = new List<string>();

// Добавление новых элементов в список.

someList.Add("Первая строка в списке");

someList.Add("Вторая строка в списке");

someList.Add("Третья строка в списке");

Двунаправленный (или двусвязный) список

Двунаправленный (или двусвязный) список – это список, который состоит из последовательности элементов, каждый из которых содержит информационную часть (данные) и два указателя на соседние элементы (на следующий и предыдущий элементы). В пространстве имен .NET C# System.Collections.Generic содержится класс LinkedList<T>, реализующий двунаправленный (двусвязный список).

IEnumerable

Интерфейс IEnumerable имеет метод, возвращающий ссылку на другой интерфейс – перечислитель.

А интерфейс IEnumerator определяет функционал для перебора внутренних объектов в контейнере.

Метод MoveNext() перемещает указатель на текущий элемент на следующую позицию в последовательности. Если последовательность еще не закончилась, то возвращает true. Если же последовательность закончилась, то возвращается false.

Свойство Current возвращает объект в последовательности, на который указывает указатель.

Метод Reset() сбрасывает указатель позиции в начальное положение.

Например, без использования цикла foreach перебирем коллекцию с помощью интерфейса IEnumerator

Коллекции со словом list

IList<T> это коллекция объектов. У этой коллекции можно получить элемент по индексу, получить индекс по элементу, удалить элемент, вставить элемент.

SortedList - класс коллекции, хранящей наборы пар "ключ-значение", отсортированных по ключу.

LinkedList<T> - Двусторонний упорядоченный список, оптимизация - вставка и удаление с любого конца, поддерживает произвольный доступ

List<T> - доступ по индексу, поиск и сортировка

ArrayList представляет коллекцию объектов. В классе ArrayList реализуются интерфейсы ICollection, IList, IEnumerable, ICloneable. В классе ArrayList определяется ряд собственных методов, помимо тех, что уже объявлены в интерфейсах, которые в нем реализуются. Коллекцию класса ArrayList можно отсортировать, вызвав метод Sort().

List - упорядоченная последовательность элементов, тогда как Set - это отдельный список неупорядоченных элементов  
 3. В чем разница между ArrayList и Array?

Основное различие между Array и ArrayList состоит в том, что Array является структурой данных фиксированной длины, а ArrayList является классом Collection переменной длины. Вкратце, Array имеет статическую природу, тогда как ArrayList имеет динамическую природу.