**Collections .Net**

2012-12-08

ToDo: Add Collection<T> and ReadOnlyCollection<T> !!!

Add OrderedDictionary, collection of keys is accessible by index (but not generic…)

|  |  |
| --- | --- |
| Access by index | Integer index |
| Arrays | Lower bound can be different than 0  Support multiple dimensions |
| ArrayList | Read-only wrapper available  Object elements imply boxing for value types  IndexOf method in O(n) |
| List<T> | Read-only wrapper available  Generic equivalent of ArrayList, as equivalent as ArrayList for reference elements, but more efficient for value types  Find method in O(n), Indexed access in O(1)  As efficient as array if no memory reallocation occurs.  Accepts null as a valid value for reference types and allows duplicate elements. |
|  |  |
| Hashed collections | Key must be unique and cannot be null |
| Hastable | Collection of key/value pairs  Object elements imply boxing for value types |
| Dictionary<TK,TV> | Retrieving values using its key in O(1) |
| ConcurrentDictionary<TK,TV> | Should be used when multiple threads might be accessing the collection simultaneously |
| HashSet<T> | Represents a set of values, no duplicates. Since FW 3.5  High-performance set operations, in addition to extension methods provided by Enumerable class (UnionWith, IntersectWith, ExceptWith, SymmetricExceptWith).  Provides methods for determining set equality, overlap of sets, and whether a set is a subset or superset of another set.  No particular order. |
|  |  |
| Sorted collections | Key must be unique and cannot be null |
| SortedList | Represents a collection of key/value pairs (object/object) that are sorted by the keys and are accessible by key and by index.  Index of elements may change after adding values.  Maintains internally two arrays.  Slower than Hashtable but more flexible (indexed access).  Insertion in O(n) except if elements are added in sorted order.  Retrieval in O(log n). |
| SortedList<TK,TV> | Generic version of SortedList  Uses less memory than SortedDictionary<TK, TV>.  Efficient indexed retrieval of keys and values through the collections returned by the Keys and Values properties.  Insertion in O(n) except if elements are added in sorted order.  Retrieval in O(log n). |
| SortedDictionary<TK,TV> | Faster insertion and removal operations for unsorted data, O(log n) as opposed to O(n) for SortedList<TK, TV>.  Insertion and retrieval in O(log n) |
| SortedSet<T> | Collection of objects that is maintained in sorted order. Since FW 4.  No duplicates. |
|  |  |
| Queues | FIFO collections.  Accepts null as a valid value for reference types and allows duplicate elements. |
| Queue | Object based |
| Queue<T> | Generic version |
| ConcurrentQueue<T> | Thread-safe version |
|  |  |
| Stacks | LIFO collections |
| Stack | Object based |
| Stack<T> | Generic version |
| ConcurrentStack<T> | Thread-safe version |
|  |  |
| Other |  |
| LinkedList<T> | Doubly linked list of values (Previous/Next navigation).  Does not support splitting or cycles.  Insertion and removal are O(1) operations.  Accepts null as a valid Value property for reference types and allows duplicate values. |
| SynchronizedCollection<T> | Thread-safe collection. |

System.Collections.Specialized

Bit collections