**Non – Generic Collections**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | List | ArrayList | Stack | Queue |
| Creation |  | ArrayList arrayname = new ArrayList(); | Stack myStack = new Stack(); | Queue myQ = new Queue(); |
| Deleting |  | Arrayname.Remove(“element”); | Console.WriteLine( "(Pop)\t\t{0}", myStack.Pop() ); | Console.WriteLine( "(Dequeue)\t{0}", myQ.Dequeue() ); |
| Updating |  | Arrayname.Add(“This”) | myStack.Push( "The" ); | myQ.Enqueue( "quick" ); |
| Reading |  | foreach (var item in arrayname)  {console.WriteLine(item);} | PrintValues( myStack, '\t' ); | PrintValues( myQ ); |
| Readonly |  | ArrayList readonlyvar = ArrayList.ReadOnly( arrayname ); |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Generic Collections**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | List | ArrayList | Stack | Queue |
| Creation |  |  | Stack<string> numbers = new Stack<string>(); | Queue<string> numbers = new Queue<string>(); |
| Deleting |  |  | Console.WriteLine("Popping '{0}'", numbers.Pop()); | Console.WriteLine("\nDequeuing '{0}'", numbers.Dequeue()); |
| Updating |  |  | numbers.Push("one"); | numbers.Enqueue("three"); |
| Reading |  |  | foreach( string number in numbers ){ Console.WriteLine(number);  } | foreach( string number in queueCopy ){ Console.WriteLine(number);  } |
| Readonly |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |