Коллекция – класс для хранения совокупности данных и выполнения простейших операций над ними.

1. Самая базовая – массив. Является классом, но в с# для него используется упрощенный синтаксис.

Существует класс Array (все массивы – его экземпляры), и у этого класса есть набор полезных статических методов.

1. Array.IndexOf(массив, значение); - производит линейный поиск указанный в значения в массиве, возвращает – индекс.
2. Array.Sort(массив) – выполняется пузырьковая сортировка (не самый быстрый алгоритм сортировки).
3. Array.BinarySearch(массив, значение) – двоичный поиск, самый быстрый поиск, работает эффективно, когда массив отсортирован.
4. Array.Reverse(массив) – меняет порядок элементов на обратный.
5. Array.Clear(массив) – очистка массива нулями.
6. Array.Copy(массив1, индекс1, количество, массив2, индекс2) – копирует массив.
7. Array.Resize(ref массив, количество) – меняет размер массива.
8. Не статический метод массив.Clone() – клонирует массив.

Все эти массивы работают с одномерными массивами.

1. Список – класс ArrayList

Отличия от базового массива:

1. Содержит множество методов: вставка, удаления и добавление новых элементов.
2. Хранит элементы типа object – в списке можно хранить элементы разного типа потому что работает в Boxing.

Методы Arraylist (экземплярные - не статические):

1. Есть все методы, что и у массива, но не статические: .IndexOf(), .Sort(), .BinarySearch(), .Clear(), .Clone(), .CopyTo()
2. .Add(значение) – добавляет новый элемент в конец списка (размер увеличивается).
3. .AddRange(массив) – добавляет массив или список элементов в конец списка.
4. .Insert(значение, индекс) – вставляет новый элемент со сдвигом в указанный индекс списка.
5. .InsertRange(массив, индекс) – вставляет массив или список элементов в указанный индекс.
6. .Remove(Значение) – ищет линейным поиском и удаляет первое вхождение со сдвигом.
7. .RemoveAt(индекс) – удаляет значение по индексу со сдвигом.
8. .RemoveRange(индекс, количество)
9. .Contains(значение) – возвращает true/false – присутствует ли значение в списке.

Класс ArrayList является оберткой для обычного массива (внутри происходит операции создания нового массива и копирования данных)

1. Stack – работает по принципу LIFO (Last Input First Output Последний Зашел Первый Вышел) в этом классе всего лишь два метода:
2. .Push(значение) – “Толкать” значение помещается на вершину стэка.
3. .Pop() – “извелекать/выталкивать” – значение извлекается/удаляется с вершины стэка.
4. Queue – работает по принципу FIFO (First Input First Output Первый Зашел Первый Вышел)
5. .Enqueue(значение) – значение помещается в очередь.
6. .Dequeue(значение) – значение удаляется из очереди.

Где и зачем очередь в программирование?

1. Очередь команд для процессора.
2. PlayList

На основе стэка процессор запоминает вызовы методов

Все эти стандартные коллекции содержат элементы object