Структури от данни- масиви, списъци, опашки, стекове

Масивите са винаги с фиксиран размер

Масивите by default се сравняват по референция, т.е. :

int[] numbers = {1, 2, 3};

int[] anotherNumber = {1, 2, 3};

Console.WriteLint(numbers == anotherNumbers); // false

За разлика от другите езици, в C# има някаква оптимизация, при която случая:

for (int i =0; i < arr.Length; i++)

Е по-бърз от:

for (int i =0; i < 10; i++)

В другите езици е обратното

Функционално програмиране – когато подаваме функция като параметър на дадена друга функция

Може да вмъкваме поредица от елементи в даден лист чрез метода InsertRange(int index, IEnumerable<T> collection)

Класа List в C#, представлява имплементация на ArrayList (т.е. List вътрешно е имплементиран чрез обикновен масив), а LinkedList представлява, различни node-ове, които имат връзка по между си(първия node сочи към втория, втория node - към третия, и т.н.)

Stack<T>

* Count - връща броя на елементите на колекцията
* Peek() – връща последния елемент
* Pop() – връща последния елемент и след това го премахва от колекцията

Елементите в стека се пълнят един върху друг, т.е. последния по ред елемент, който сме вкарали ще бъде първия, който ще изкараме

Първият, който влиза, е последният, който излиза

Queue<T> - first-in-first-out collection

Първият, който влиза е първият, който излиза

Stack<T> and Queue<T> provide LIFO and FIFO lists

LINQ extension методите предоставят допълнителни функционалности за колекции