**Ламбда изрази (lambda expressions)**

**(анонимни функции) + LINQ - заявки**

* Анонимни функции, които съдържат изрази или последователност от оператори, използват ламбда оператора **=>**
* Лява страна - входните параметри
* дясна страна - израз
* евентуално връща някакъв резултат.

**a => a > 5;** Чете се: **„*а* goes to** *a* **larger than 5”**

**Func() –** начин за наименоване на метод от анонимен тип

(e.g static int Func(x) {return x / 2;}

**LINQ:** var numbers = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

**разширяващи методи:**

* **Min();** int minValue = numbers.Min(); // 1
* **Max();** int maxValue = numbers.Min(); // 10
* **Sum();**
* **Average();**
* **Where(); -** var newList = numbers.Where(x => x > 0).ToList(); - връща Ienumerable, затова трябва изрично да му зададем тип – ToList();
* **Foreach(); - само за List! Void Method – не връща резултат!**

numbers.Where(x => x % 2 == 0).ToList().ForEach(x => Console.WriteLine(x));

* **Sort(); -** voidnumbers.Sort((x1, x2) => x1.Title.CompareTo(x2));
* **OrderBy();** връщат резултат
* **ThanBy();**
* **OrderByDescending();**
* **Select(); -** извърши някакво действие върху всеки един елемент!(Връща резултат) var catNumbers = numbers.Select(x => "Cat" + " " + x).ToList(); // Cat1 Cat2 Cat3...
* **Take(); / TakeLast();**
* **Skip();**  var result = numbers.Skip(3);
* **ToList();**
* **ToArray();**
* **ToDictionary();**

var newDict = someDict

.Where(x => x.Value == 5)

.ToDictionary(x => x.Key, y => y.Value)

**Тернарен оператор:** int a = 5;

string result = a > 10 ? "Big" : "Small";

**Форматиране на стринг с плейсхолдъри:**

String output = “{0} => {1}”;

Console.WriteLine(string.Join(Enviroment.NewLine, userPlateNumber.Select(x => string.Format(otput, x.Key, x.Value))));

Ako искам да използвам войд медтод в комбинация с други методи, които връщат резултат, мога преди него да напиша .AsEnumerable - > .Sort() !!!