**Dictionaries**

**(Associative arrays)(Maps)**

- всеки елемент - двойка **ключ (Key)** и **стойност (Value)**

- ключът е уникален

**1.Създаване на Речник - Using System.Collection.Generics**

Dictionary<string, double> studentMarks = new Dictionary<string, double>();

tudentMarks["Vasil"] = 3.50;

studentMarks["Petya"] = 6.00;

foreach (var item in studentMarks)

{

Console.WriteLine(item.Key);

Console.WriteLine(item.Value);

}

Ако напишем само Console.WriteLine(item); ще върне резултата като *KeyValuePair*

//[Vasil, 3.5]

//[Petya, 6]

**2. Използване на For – loop –**  Може да се използва само с ElementAt във формат *KeyValuePair*

for (int i = 0; i < studentMarks.Count; i++)

{

Console.WriteLine(studentMarks.ElementAt(i)); //[Vasil, 3.5]

//[Petya, 6]

}

**3. Операции**

* Add(key, value) - добавя в речника зададената наредена двойка(Бавна операция!). **При добавяне на вече съществуващ ключ, се хвърля грешка -** System.ArgumentException
* bool Remove(key) – премахва стойността за този ключ (цялата двойка) от речника; връща дали е премахнат елемент от речника.
* bool ContainsKey(key) – връща true или false

bool ContainsValue(value)

* Clear() – void метод, т.е не връща резултат – трие целият списък studentMarks.Clear();
* int Count – връща броя елементи (двойки ключ-стойност) в речника

**SortedDictionary<key, value>** - имплементация с балансирано (червено-черно) дърво – Balanced Tree (малко по-бавно работи)

**Dictionary** – имплементация с хеш-таблица