int[] array = new int[] { 1, 2, 3, 4, 5 }; // Създаване на масив от цели числа  
string[] stringArray = new string[] { "a", "b", "c" }; // Масив от низове  
  
var list = new List<int> { 1, 2, 3, 4, 5 }; // Създаване на списък от цели числа  
list.Add(6); // Добавяне на елемент към списъка  
  
var dictionary = new Dictionary<string, int>  
{  
   { "one", 1 },  
   { "two", 2 },  
   { "three", 3 }  
}; // Създаване на речник от низове и цели числа  
  
var set = new HashSet<int> { 1, 2, 3, 3, 4 }; // Създаване на множество от цели числа  
// Стойността "3" е добавена два пъти, но ще се появи само веднъж в множеството  
  
var queue = new Queue<int>(); // Създаване на опашка от цели числа  
queue.Enqueue(1); // Добавяне на елемент в опашката  
queue.Enqueue(2);  
queue.Enqueue(3);  
var firstElement = queue.Dequeue(); // Изваждане на първия елемент (1)  
  
var stack = new Stack<int>(); // Създаване на стек от цели числа  
stack.Push(1); // Добавяне на елемент в стека  
stack.Push(2);  
stack.Push(3);  
var lastElement = stack.Pop(); // Изваждане на последния елемент (3)

int[] array = { 1, 2, 3, 4, 5 };  
int length = array.Length; // Връща 5  
int firstElement = array[0]; // Връща първия елемент (1)  
array[1] = 10; // Променя втория елемент на 10  
  
var list = new List<int> { 1, 2, 3 };  
list.Add(4); // Добавя 4 към списъка  
bool containsTwo = list.Contains(2); // Проверява дали списъкът съдържа 2 (true)  
list.Remove(2); // Премахва 2 от списъка  
  
var dictionary = new Dictionary<string, int> { { "a", 1 }, { "b", 2 } };  
dictionary.Add("c", 3); // Добавя нова двойка  
bool containsA = dictionary.ContainsKey("a"); // Проверява дали ключът "a" съществува  
int value;  
if (dictionary.TryGetValue("b", out value))  
{  
   Console.WriteLine(value); // Показва 2  
}  
  
var set = new HashSet<int> { 1, 2, 3 };  
set.Add(4); // Добавя 4  
bool containsThree = set.Contains(3); // Проверява дали множеството съдържа 3  
set.Remove(1); // Премахва 1  
  
var queue = new Queue<int>();  
queue.Enqueue(1); // Добавя 1 в опашката  
queue.Enqueue(2);  
int first = queue.Peek(); // Връща първия елемент без да го премахва (1)  
int dequeued = queue.Dequeue(); // Премахва първия елемент (1)  
  
var stack = new Stack<int>();  
stack.Push(1); // Добавя 1 в стека  
stack.Push(2);  
int top = stack.Peek(); // Връща последния елемент без да го премахва (2)  
int popped = stack.Pop(); // Премахва последния елемент (2)

[Yesterday 20:14] Robart Dimitrov Syulev

string sentence = "Hello World! This is C#.";  
var words = sentence.Split(' '); // Разделя по празно място  
  
var numbers = new List<int> { 1, 2, 3 };  
var first = numbers.FirstOrDefault(); // Връща 1  
var emptyList = new List<int>();  
var defaultResult = emptyList.FirstOrDefault(); // Връща 0 (стойност по подразбиране)  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
var evenNumbers = numbers.Where(n => n % 2 == 0); // Връща 2 и 4  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
var squares = numbers.Select(n => n \* n); // Връща квадрати на числата (1, 4, 9, 16, 25)  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
var sum = numbers.Aggregate((a, b) => a + b); // Сумира всички числа (15)  
  
var numbers = new List<int> { 5, 3, 2, 1, 4 };  
var sortedNumbers = numbers.OrderBy(n => n); // Сортира по възходящ ред (1, 2, 3, 4, 5)  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
var max = numbers.Max(); // Връща 5  
var min = numbers.Min(); // Връща 1  
  
string text = "Hello, World!";  
bool containsWorld = text.Contains("World"); // Връща true  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
bool hasEvenNumbers = numbers.Any(n => n % 2 == 0); // Връща true  
var emptyList = new List<int>();  
bool hasElements = emptyList.Any(); // Връща false  
  
var numbers = new List<int> { 2, 4, 6, 8 };  
bool allEven = numbers.All(n => n % 2 == 0); // Връща true  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
int count = numbers.Count(); // Връща 5  
int evenCount = numbers.Count(n => n % 2 == 0); // Връща броя на четните числа (2)  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
var reversed = numbers.Reverse(); // Връща (5, 4, 3, 2, 1)  
  
var numbers = new List<int> { 1, 2, 2, 3, 4, 4, 5 };  
var distinctNumbers = numbers.Distinct(); // Връща (1, 2, 3, 4, 5)  
  
var list1 = new List<int> { 1, 2, 3 };  
var list2 = new List<int> { 4, 5, 6 };  
var concatenated = list1.Concat(list2); // Връща (1, 2, 3, 4, 5, 6)  
  
var list1 = new List<string> { "a", "b", "c" };  
var list2 = new List<int> { 1, 2, 3 };  
var zipped = list1.Zip(list2, (s, n) => $"{s}{n}"); // Връща ("a1", "b2", "c3")  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
var skipped = numbers.Skip(2); // Пропуска първите два елемента (3, 4, 5)  
var taken = numbers.Take(3); // Взима първите три елемента (1, 2, 3)  
  
var people = new List<string> { "Alice", "Bob", "Charlie", "Alice", "Bob" };  
var grouped = people.GroupBy(name => name); // Групира по име  
foreach (var group in grouped)  
{  
   Console.WriteLine($"Name: {group.Key}, Count: {group.Count()}");  
}  
  
var customers = new List<(int Id, string Name)> { (1, "Alice"), (2, "Bob") };  
var orders = new List<(int CustomerId, string Product)> { (1, "Product1"), (2, "Product2"), (1, "Product3") };  
var customerOrders = customers.Join(orders,   
                                   c => c.Id,   
                                   o => o.CustomerId,   
                                   (c, o) => $"{c.Name} ordered {o.Product}");  
foreach (var co in customerOrders)  
{  
   Console.WriteLine(co); // Връща "Alice ordered Product1", "Alice ordered Product3", "Bob ordered Product2"  
}  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
var array = numbers.ToArray(); // Преобразува в масив  
var list = numbers.ToList(); // Преобразува в списък  
  
var people = new List<(string Name, int Age)> { ("Alice", 30), ("Bob", 25) };  
var dictionary = people.ToDictionary(p => p.Name, p => p.Age);  
Console.WriteLine(dictionary["Alice"]); // Връща 30

В този пример, лямбда изразът, който се предава на ForEach(), съдържа for цикъл, който отпечатва стойностите от 0 до num-1. Така комбинираме циклите с лямбда изрази.  
  
Action<int> printNumbers = n =>  
{  
   for (int i = 0; i < n; i++)  
   {  
       Console.WriteLine(i);  
   }  
};  
  
printNumbers(5); // Ще отпечата 0, 1, 2, 3, 4  
  
var numbers = new List<int> { 1, 2, 3, 4, 5 };  
  
numbers.ForEach(num =>  
{  
   for (int i = 0; i < num; i++)  
   {  
       Console.WriteLine(i);  
   }  
});