

# Tema 8

Sr	Array	ArrayList
1	Array is strongly typed. This means that an array can store only specific type of items\elements	ArrayList can store any type of items\elements.
2	In arrays we can store only one datatype either int, string, char etc...	In arraylist we can store all the datatype values
3	Array cant accept null	ArrayList collection accepts null
4	Arrays belong to System.Array namespace using System;	Arraylist belongs to System.Collection namespaces using System.Collections;
5	Example - <code>int[] intArray=new int[]{2}; intArray[0] = 1; intArray[2] = 2;</code>	Example - <code>ArrayList Arrlst = new ArrayList(); Arrlst.Add("Sagar"); Arrlst.Add(1); Arrlst.Add(null);</code>

## ARRAY VS ARRAYLIST

**SINTAKSA ZA ARRAYS**

**STATICKA DEKLARACIJA**

`Tip[] imeNiza = {skup vrednosti}`

```
int[] niz1 = {2, 4, 5, 6};  
double[] niz2 = {1.3, 1.7, 8.4};  
string[] niz3 = {"test", "pismeni"}
```

**DINAMICKA DEKLARACIJA**

```
Tip[] imeNiza;  
int[] niz1;  
niz1 = new int[5];  
niz[1] = 54;...
```

## PRIMER 1 - Osnovni nacin deklarisanja nizova u C# - ARRAYS, pristup elementima i osnovne funkcionalnosti sa nizovima.

```
using System;  
using System.Linq;  
  
namespace Nizovi1  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            // deklaracija promenljivih  
  
            int[] evenNums = { 2, 4, 6, 8, 10 };  
  
            string[] cities = { "Mumbai", "London", "New York" };  
        }  
    }  
}
```

```

// pristupanje

Console.WriteLine(evenNums[0]); //prints 2
Console.WriteLine(evenNums[1]); //prints 4

// pristupanje elementima kroz petlju
for (int i = 0; i < evenNums.Length; i++)
    Console.WriteLine(evenNums[i]);

for (int i = 0; i < evenNums.Length; i++)
    evenNums[i] = evenNums[i] + 10; // update the value of each
element by 10

// foreach
foreach (var item in evenNums)
    Console.WriteLine(item);

foreach (var city in cities)
    Console.WriteLine(city);

// metode iz using System.Linq;
Console.WriteLine(evenNums.Min()); // returns 6
Console.WriteLine(evenNums.Sum()); // returns 55
Console.WriteLine(evenNums.Average()); // returns 55
Console.WriteLine(evenNums.ElementAt(3));
Console.WriteLine(evenNums.Max());
Console.WriteLine(evenNums.Count());

    }
}
}

```

**NAPOMENA 1:** Skrenuti paznju na dodatni namespace koji se ukljuuje.

**NAPOMENA 2:** Povuci paralelu sa Pythonom

**NAPOMENA 3:** Objasnjenje sta je var

## **PRIMER 2: Dinamicki nacini deklaracije nizova u C#.**

```

static void Main(string[] args)
{
    // zakasnela deklaracija, dinamička deklaracija
    int[] oddNums;

    oddNums = new int[5];
    oddNums[0] = 1;
    oddNums[1] = 3;
    oddNums[2] = 5;
    oddNums[3] = 7;
    oddNums[4] = 9;
    // or
    // oddNums = new int[] { 2, 4, 6, 8, 10 };
}

```

```

// metode iz biblioteke
Console.WriteLine(oddNums.Length);
Array.Sort(oddNums);
foreach (var item in oddNums)
    Console.WriteLine(item);
Array.Reverse(oddNums);
foreach (var item in oddNums)
    Console.WriteLine(item);
Console.WriteLine(Array.BinarySearch(oddNums, 5));

// Create an array of four elements, and add values later
string[] cars1 = new string[4];

// Create an array of four elements and add values right away
string[] cars2 = new string[4] { "Volvo", "BMW", "Ford", "Mazda"
};

// Create an array of four elements without specifying the size
string[] cars3 = new string[] { "Volvo", "BMW", "Ford", "Mazda"
};

// Create an array of four elements, omitting the new keyword,
and without specifying the size
string[] cars4 = { "Volvo", "BMW", "Ford", "Mazda" };

// Declare an array
string[] cars5;

// Add values, using new
cars5 = new string[] { "Volvo", "BMW", "Ford" };

// Add values without using new (this will cause an error)
// cars5 = { "Volvo", "BMW", "Ford"};
}

```

**NAPOMENA 1:** Isprobati sta sve uzrokuje gresku.

**NAPOMENA 2:** Objasniti dinamicko i staticko kreiranje elemenata nizova

**NAPOMENA 3:** Objasniti sta se desava u memoriji kada se stavi ovo 'new'

**PRIMER 3:** a) Primer sa nizom, kako se spajaju dva niza

```

using System.Linq; // ovo mora da bude ukljuceno za union

static void Main(string[] args)
{
    string[] animals = { "Cat", "Alligator", "Fox", "Donkey", "Cat"
};
    string[] birds = { "Sparrow", "Peacock", "Dove", "Crow" };

    var all = animals.Union(birds).ToArray();
    foreach (var element in all) {
        Console.WriteLine(element);
    }
}

```

```

int[] num1 = { 1, 2, 3, 4, 3, 55, 23, 2 };
int[] num2 = { 55, 23, 45, 50, 80 };

var ukupno = num1.Union(num2).ToArray();
foreach (var element in ukupno)
{
    Console.WriteLine(element);
}
}

```

**NAPOMENA 1:** Ne zaboraviti da se doda using System. Linq

**NAPOMENA 2:** Objasniti da mora ovo toArray() jer je ta promenljiva definisana kao var pa se posle mora reci koji je tip da bi se moglo dalje raditi sa njom

b)

```

static void Main(string[] args){

    int[] nums = { 1, 2, 3, 4, 3, 55, 23, 2 };
    int[] dist1 = nums.Distinct().ToArray();
    foreach (var element in dist1)
    {
        Console.WriteLine(element);
    }

    string[] animals = { "Cat", "Alligator", "Fox", "Donkey", "Cat"
};

    string[] dist2 = animals.Distinct().ToArray();
    foreach (var element in dist2)
    {
        Console.WriteLine(element);
    }
}

```

**PRIMER 4:** Primer sa koristenjem ArrayList kolekcije u C#

```

// mora da se ukljuci using System.Collections;

static void Main(string[] args)
{
    ArrayList arr = new ArrayList();
    arr.Add("One");
    arr.Add("Two");
    arr.Add("Three");
    arr.Add("Four");

    Console.WriteLine("ArrayList...");
    foreach (string str in arr)
    {
        Console.WriteLine(str);
    }
}

```

```

    }
    arr.Remove("Three");
    Console.WriteLine("ArrayList after removing an element...");

    foreach (string str in arr)
    {
        Console.WriteLine(str);
    }
    arr.Insert(0, "zero");
    Console.WriteLine("ArrayList after adding zero element...");

    foreach (string str in arr)
    {
        Console.WriteLine(str);
    }
    arr.Sort();

    Console.WriteLine("ArrayList after sorting...");
    foreach (string str in arr)
    {
        Console.WriteLine(str);
    }
    arr.Reverse();

    Console.WriteLine("ArrayList after reverse sorting...");

    foreach (string str in arr)
    {
        Console.WriteLine(str);
    }
    if (arr.Contains("zero"))
    {
        Console.WriteLine("Sadrzi");
    }else
    {
        Console.WriteLine("Ne sadrzi");
    }
    arr.Clear();
    Console.WriteLine(arr.Count);
}

```

**NAPOMENA 1:** Objasniti razliku izmedju metode i propertija u klasi

**NAPOMENA 2:** Podsetiti se klasa u Pythonu

**PRIMER 5:** Primer sa Array listama - dinamički se setuju elementi liste, pa se onda radi njihov zbir.

```
using System.Collections;
```

```

static void Main(string[] args)
{
    int broj = -1;
    ArrayList niz = new ArrayList();

    int suma = 0;

    while (broj != 0)
    {
        Console.WriteLine("Unesi naredni broj");
        broj = Convert.ToInt32(Console.ReadLine());
    }
}

```

```

        if (broj != 0)
        {
            niz.Add(broj);
            suma += broj;
        }
    }

    niz.Sort();
    // Console.WriteLine("Niz je: ({0})", String.Join(", ",
niz.Cast<int>()));

    foreach (int element in niz)
    {
        Console.WriteLine(element);
    }
}

```

**NAPOMENA 1:** Uraditi prvo varijantu sa fiksnih 5 elemenata niza.

**NAPOMENA 2:** Zakomentarisanu liniju odkomentarisati na kraju i objasniti kako radi taj ispis u jednom redu.

**NAPOMENA 3:** Za taj CAST treba da se ukljuci **using System.Linq;**

## **PRIMER 6: RAZLIKA IZMEDJU ARRAY I ARRAYLIST**

```

static void Main(string[] args)
{

    ArrayList niz = new ArrayList();
    int[] _niz = { 5, 6, 7, 9 };

    niz.Add(23);
    niz.Add("45");
    niz.Add(null);

    foreach (var element in niz)
    {
        if (element != null)
        {
            Console.WriteLine(element.ToString());
        }
        else
        {
            Console.WriteLine("null");
        }
    }

    _niz[3] = 55;
    Array.Resize(ref _niz, _niz.Length + 1);
    _niz[4] = 100;
    for (int i = 0; i < _niz.Length; i++)
    {
        Console.WriteLine(_niz[i]);
    }
}

```