

آرایه ها

آرایه ها در زبان برنامه نویسی سی شارپ نوعی ساختار داده ای هستند که به منظور ذخیره سازی مجموعه ای از عناصر هم نوع استفاده می شوند. آرایه ها می توانند شامل انواع مختلفی از داده ها مانند عددی، کاراکتری یا رشته ای باشند و اندازه آن ها در زمان ایجاد تعیین می شود و بعد از آن قابل تغییر نیست ، دسترسی به عناصر آرایه ها با استفاده از اندیس ها انجام می شود، به طوری که اولین عنصر در اندیس صفر قرار دارد.

خصوصیات و متدهای کلاس

کلاس آرایه دارای مجموعه ای از متدها و خصوصیات است که یکی از مهم ترین ویژگی های این کلاس، خاصیت **Length** بوده که تعداد عناصر موجود در آرایه را برمی گرداند. متد **Sort** به برنامه نویسان امکان می دهد تا عناصر آرایه را به صورت صعودی یا نزولی مرتب کنند. همچنین، متد **Reverse** می تواند ترتیب عناصر آرایه را معکوس کند. متد **IndexOf** به جستجوی یک عنصر خاص در آرایه کمک می کند و موقعیت آن را برمی گرداند.. متد **Clear** نیز می تواند برای پاک کردن عناصر یک آرایه استفاده شود.

مثال : آرایه ای به نام **names** با نام سه دانشجو تعریف کرده و داده های آن را در خروجی نمایش دهید.

```
string[] names=new string[3]
{"hadi","farhad","ali"};
foreach(string s in names){
    Console.WriteLine(s);}
```

مثال : کلاسی بنام **Test** با متدی بنام **sum** تعریف کنید که یک آرایه از اعداد صحیح را بعنوان پارامتر گرفته و مجموع تمامی عناصر موجود در آرایه را محاسبه و بازگرداند.

```
class Test{  
    public int Sum(int[] numbers){  
        int total = 0;  
        foreach (int number in numbers)  
        {  
            total += number;  
        }  
        return total;  
    }  
}
```

مثال: کلاسی بنام **Test** با متدی بنام **avg** تعریف کنید که یک آرایه از اعداد صحیح را بعنوان پارامتر گرفته و میانگین عدد های موجود در آن را بازگرداند.

```
class Test{  
    public double Avg(int[] numbers)  
    {  
        int total = 0;  
        foreach (int number in numbers)  
        {  
            total += number;  
        }  
        return (double)total / numbers.Length;  
    }  
}
```

مثال : کلاسی بنام **Test** با متدی بنام **showcount** تعریف کنید که یک آرایه از نام دانشجویان و نام دانشجویی را بعنوان پارامتر گرفته و تعداد تکرار آن نام در آرایه را بازگرداند.

```
class Test
{
    public int ShowCount(string[] studentNames,
        string targetName){
        int count = 0;
        foreach (string name in studentNames)
        {
            if (name==targetName)){count++;}
        }
        return count;}}}
```

مثال : کلاسی بنام **Test** با متدی بنام **evencount** تعریف کنید که یک آرایه از اعداد صحیح را بعنوان پارامتر گرفته و تعداد عدد های زوج موجود در آن آرایه را بازگرداند.

```
class Test{
    public int EvenCount(int[] numbers)
    {
        int count = 0;
        foreach (int number in numbers)
        {
            if (number % 2 == 0){count++;}
        }
        return count;}}}
```

مثال: کلاسی بنام **Test** با متدی بنام **issymmetric** تعریف کنید که یک آرایه از اعداد صحیح را بعنوان پارامتر گرفته و در صورت متقارن بودن آرایه **true** و در غیر اینصورت **false** را بازگرداند.

```
class Test{
    public bool IsSymmetric(int[] numbers){
        int length = numbers.Length;
        for (int i = 0; i < length / 2; i++)
        {
            if (numbers[i] != numbers[length - 1 - i])
            {
                return false;
            }
        }
        return true;
    }
}
```

مثال: کلاسی بنام **Test** با متدی بنام **bigcount** تعریف کنید که یک آرایه از اعداد صحیح و عددی را بعنوان پارامتر گرفته و تعداد اعداد بزرگتر از آن عدد را بازگرداند.

```
class Test{
    public int BigCount(int[] numbers, int x){
        int count = 0;
        foreach (int number in numbers){
            if (number > x){count++;}
        }
        return count;
    }
}
```

مثال : کلاسی بنام **Test** با متدی بنام **addtoarray** تعریف کنید که یک آرایه از اعداد صحیح و عددی را بعنوان پارامتر گرفته و آن عدد را به تمامی عناصر آرایه اضافه نماید و آرایه را بازگرداند.

```
class Test{
    public int[] AddToArray(int[] numbers,
        int valueToAdd){
        for (int i = 0; i < numbers.Length; i++)
        {
            numbers[i] += valueToAdd;
        } return numbers;}}}
```

مثال: کلاسی بنام **Test** با متدی بنام **rev** تعریف کنید که یک آرایه از اعداد صحیح را بعنوان پارامتر گرفته و بدون استفاده از امکانات زبان برنامه نویسی آن را معکوس کرده و بازگرداند.

```
class Test{
    public int[] Rev(int[] numbers){
        int length = numbers.Length;
        int[] reversedArray = new int[length];
        for (int i = 0; i < length; i++){
            reversedArray[i] = numbers[length - 1 - i];
        } return reversedArray;}}
```