

## فایل ها

### فایل های متنی

فایل های متنی، نوعی از فایل ها هستند که اطلاعات را به صورت متنی ساده ذخیره می کنند و معمولاً از کاراکترهای **ASCII** یا **Unicode** تشکیل شده اند. این فایل ها به راحتی قابل ویرایش و خواندن هستند و به دلیل سادگی ساختارشان، در بسیاری از برنامه ها و سیستم ها مورد استفاده قرار می گیرند. فایل های متنی معمولاً با پسوند هایی مانند **txt**، **csv** و **log** شناخته می شوند و می توانند شامل اطلاعات متنوعی از جمله داده های جدولی، گزارش ها، یا متن های ساده باشند. به دلیل عدم وجود فرمت پیچیده، این فایل ها به راحتی توسط کاربران و برنامه های مختلف قابل دسترسی و پردازش هستند، و از این رو در تبادل داده ها، ذخیره سازی اطلاعات، و برنامه نویسی به عنوان یک ابزار کلیدی شناخته می شوند.

### کلاس File

کلاس **File** در زبان برنامه نویسی سی شارپ، بخشی از فضای نام **System.IO** است و به عنوان ابزاری قدرتمند برای مدیریت و کار با فایل ها در سیستم عامل ها عمل می کند. این کلاس شامل متدهای متنوعی است که امکان انجام عملیات مختلفی مانند ایجاد، خواندن، نوشتن، کپی، و حذف فایل ها را فراهم می آورد.

### خصوصیات کلاس File

کلاس **File** در سی شارپ دارای چندین خصوصیت (**Property**) است که به کاربر اجازه می دهد تا اطلاعات مفیدی درباره فایل ها به دست آورد. از جمله مهم ترین خصوصیات این کلاس می توان به **Attributes** اشاره کرد که مشخصات و ویژگی های فایل مانند حالت خواندنی، نوشتنی، مخفی و سیستم را نمایش می دهد.

## متدهای کلاس File

کلاس **File** در سی شارپ شامل مجموعه‌ای از متدها است که برای انجام عملیات مختلف بر روی فایل‌ها طراحی شده‌اند. برای خواندن محتویات فایل‌ها، متدهای **ReadAllText** و **ReadAllLines** به ترتیب محتوای کامل فایل یا خطوط آن را به عنوان رشته یا آرایه‌ای از رشته‌ها برمی‌گردانند. از سوی دیگر، متدهای **WriteAllText** و **WriteAllLines** برای نوشتن متن یا آرایه‌ای از رشته‌ها به فایل‌ها به کار می‌روند. همچنین، متدهای **Copy** و **Move** برای کپی و انتقال فایل‌ها به مکان‌های جدید استفاده می‌شوند.

مثال : برنامه ای بنویسید که عبارت **Hello Seraj** را در فایلی بنام **output.txt** بنویسد.

```
C# Program.cs
using System.IO;
string filePath = "output.txt";
string data="Hello Seraj";
File.WriteAllText(filePath,data);
```

مثال : برنامه ای بنویسید که داده های موجود در فایلی بنام **output.txt** را در خروجی نمایش دهد.

```
C# Program.cs
using System.IO;
string filePath = "output.txt";
string s = File.ReadAllText(filePath);
Console.WriteLine(s);
```

نکته : این برنامه ها با فرض موجود بودن و وجود داده در فایل نوشته شده اند در محیط واقعی باید وجود فایل و خالی نبودن آن را بررسی کنید.

مثال : برنامه ای بنویسید که نام های موجود در آرایه ای بنام **students** را در فایلی بنام **output.txt** بنویسید.

```
C# Program.cs
string filePath = "output.txt";
string[] students=new string[]{
    "hadi","farhad","ali"};
File.WriteAllLines(filePath,students);
```

مثال : برنامه ای بنویسید که با فرض موجود بودن نام تعدادی دانشجو در فایلی بنام **output.txt** آنها را در خروجی نمایش دهد.

```
C# Program.cs
using System.IO;
string filePath = "output.txt";
string[] s=File.ReadAllLines(filePath);
foreach (string name in s){
    Console.WriteLine(name);
}
```

مثال : برنامه ای بنویسید ده عدد تصادفی دورقمی تولید و آنها را در فایلی متنی بنام **output.txt** بنویسد.

```
C# Program.cs
using System.IO;
string filePath = "output.txt";
string[] nums=new string[10];
Random r=new Random();
for(int i=0;i<10;i++){
    nums[i]=r.Next(10,101).ToString();}
File.WriteAllLines(filePath,nums);
```

مثال : بافرض موجود بودن تعدادی عدد در فایلی بنام **output.txt** برنامه ای بنویسید که دوبرابر عدد های موجود در فایل را در خروجی نمایش دهد.

```
C# Program.cs
using System.IO;
string filePath = "output.txt";
string[] nums=File.ReadAllLines(filePath);
foreach(string n in nums){
    int num=Convert.ToInt32(n);
    Console.WriteLine(num*2);}
```

مثال : کلاسی بنام **Test** بنویسید که شامل متدی بنام **genNums** باشد ، این متد عدد صحیحی را بعنوان پارامتر گرفته و به تعداد آن ، عدد تصادفی دورقمی در فایلی بنام **output.txt** بنویسد.

```
using System.IO;
class Test{
    private string filePath="output.txt";
    public void genNums(int n){
        Random r=new Random();
        string[] nums=new string[n];
        for(int i=0;i<n;i++){
            nums[i]=r.Next(10,100).ToString();
        }
        File.WriteAllLines(filePath,nums);}}
```

مثال : به کلاس **Test** متدی بنام **showNums** اضافه کنید که عدد های موجود در فایل را در خروجی نمایش دهد.

```
class Test{
    public void showNums(){
        string[] nums=File.ReadAllLines(filePath);
        foreach(string n in nums){
            Console.Write("{0} ",n);
        }
    }
}
```

مثال : به کلاس **Test** متدی بنام **showSum** اضافه نمایید که مجموع عدد های موجود در فایل را در خروجی نمایش دهد.

```
class Test{
    public void showSum(){
        int sum=0;
        string[] nums=File.ReadAllLines(filePath);
        foreach(string n in nums){
            sum=sum+Convert.ToInt32(n);
        }
        Console.WriteLine("Sum={0}",sum);}
}
```

مثال : به کلاس **Test** متدی بنام **showCount** اضافه نمایید که عددی را بعنوان پارامتر گرفته و تعداد آن عدد در فایل را بازگرداند.

```
class Test{
    public int showCount(int n){
        int count=0;
        string data=n.ToString();
        string[] nums=File.ReadAllLines(filePath);
        foreach(string s in nums){
            if (s==data){count++;}
        }
        return count;}
}
```

مثال : به کلاس **Test** متدی بنام **showEven** اضافه نمایید که عدد های زوج موجود در فایل را در خروجی نمایش دهد.

```
class Test{
    public void showEven(){
        string[] nums=File.ReadAllLines(filePath);
        foreach(string n in nums){
            if (Convert.ToInt32(n)%2==0){
                Console.Write("{0} ",n);}
        }
    }
}
```

مثال : به کلاس **Test** متدی بنام **evenCount** اضافه نمایید که تعداد عدد های زوج موجود در فایل را در خروجی نمایش دهد.

```
class Test{
    public void evenCount(){
        string[] nums=File.ReadAllLines(filePath);
        int count=0;
        foreach(string n in nums){
            if (Convert.ToInt32(n)%2==0){
                count++;}
            Console.Write("count={0} ",count);
        }
    }
}
```

مثال : بافرض موجود بودن فایلی بنام **output.txt** برنامه ای بنویسید که عددی از کاربر گرفته و به انتهای فایل اضافه نماید.

```
C# Program.cs
using System.IO;
string filePath ="output.txt";
string num=Console.ReadLine();
File.AppendAllText(filePath,num+"\n");
```

تمرین ۸ : برنامه ای بنویسید که نام و موبایل دانشجویی را گرفته و در فایلی بنام **output.txt** ذخیره سازی نماید.

```
C# Program.cs
using System.IO;
List<string> ls=new List<string>();
ls.Add(Console.ReadLine());
ls.Add(Console.ReadLine());
string filePath="output.txt";
File.WriteAllLines(filePath,ls);
```