

فایل های باینری

فایل های باینری در سی شارپ به نوعی از فایل ها اطلاق می شوند که داده ها به صورت غیر متنی و به شکل بایت ها ذخیره می شوند. این نوع فایل ها به دلیل قابلیت ذخیره سازی انواع داده ها، از جمله اعداد صحیح، اعداد اعشاری، رشته ها و اشیاء پیچیده، بسیار کارآمد هستند. برای کار با فایل های باینری در سی شارپ، می توان از کلاس های **FileStream**, **BinaryReader** و **BinaryWriter** استفاده کرد. استفاده از فایل های باینری معمولاً برای ذخیره سازی داده های بزرگ یا پیچیده، مانند تصاویر، صداها و پایگاه های داده، مناسب تر از فایل های متنی است، زیرا آن ها فضای کمتری را اشغال می کنند و سرعت دسترسی به داده ها را افزایش می دهند.

روش های دسترسی به فایل

در کلاس **FileStream** در سی شارپ، **FileMode** یک نوع **Enum** است (در فصل های آینده با این نوع از داده ها آشنا خواهیم شد) که نحوه باز کردن یا ایجاد یک فایل را مشخص می کند. این **Enum** شامل چندین حالت مختلف است که هر کدام کاربرد خاص خود را دارند. به عنوان مثال، **FileMode.Create** برای ایجاد یک فایل جدید استفاده می شود و اگر فایل با همین نام وجود داشته باشد، آن را بازنویسی می کند. **FileMode.Open** برای باز کردن یک فایل موجود به کار می رود و در صورت عدم وجود فایل، یک استثنا ایجاد می کند. **FileMode.Append** به شما این امکان را می دهد که داده ها را به انتهای یک فایل موجود اضافه کنید بدون اینکه محتوای قبلی آن را از بین ببرید.

مثال : برنامه ای بنویسید که نام ، شماره دانشجویی و معدل دانشجویی فرضی را در فایل بیابری بنام **seraj.dat** ذخیره سازی نماید.

```
C# Program.cs
using System.IO;
string name = "Farhad";
int studentId = 1111;
double avg = 17.5;
string fileName = "seraj.dat";
FileStream fs = new FileStream(fileName, FileMode.Create);
BinaryWriter writer = new BinaryWriter(fs);
writer.Write(name);
writer.Write(studentId);
writer.Write(avg);
writer.Close(); fs.Close();
```

مثال : برنامه ای بنویسید که داده های دانشجویی شامل نام ، شماره دانشجویی و معدل را از فایل بیابری بنام **seraj.dat** خوانده و در خروجی نمایش دهد.

```
C# Program.cs
using System.IO;
string fileName = "seraj.dat";
FileStream fs = new FileStream(fileName, FileMode.Open);
BinaryReader reader = new BinaryReader(fs);
string name = reader.ReadString();
int studentId = reader.ReadInt32();
double avg = reader.ReadDouble();
Console.WriteLine(name);
Console.WriteLine(studentId);
Console.WriteLine(avg);
```

مثال : برنامه ای بنویسید که ده عدد صحیح تصادفی تک رقمی تولید کرده و در فایل باینری بنام **seraj.dat** بنویسد.

```
using System.IO;
string fName = "seraj.dat";
Random random = new Random();
FileStream fs = new FileStream(fName, FileMode.Create);
BinaryWriter writer = new BinaryWriter(fs);
for (int i = 0; i < 10; i++){
    int randomNumber = random.Next(0, 10);
    writer.Write(randomNumber);}
writer.Close();fs.Close();
```

مثال : با فرض موجود بودن تعدادی عدد صحیح در فایل باینری بنام **seraj.dat** برنامه ای بنویسید که عدد های موجود در فایل را در خروجی نمایش دهد.

```
using System.IO;
string fileName = "seraj.dat";
FileStream fs = new FileStream(fileName, FileMode.Open);
BinaryReader reader = new BinaryReader(fs);
while (fs.Position < fs.Length){
    int number = reader.ReadInt32();
    Console.WriteLine(number);}
reader.Close();fs.Close();
```

تمرین ۱ : کلاسی بنام **Test** بنویسید که شامل متدی بنام **addStudent** باشد ، این متد نام و نمره دانشجویی را گرفته و در فایلی باینری بنام **seraj.dat** ذخیره سازی کند.

تمرین ۲ : متدی بنام **showNames** به کلاس **Test** اضافه نمایید که نام دانشجویان موجود در فایل را در خروجی نمایش دهد.

تمرین ۳ : متدی بنام **showPass** به کلاس **Test** اضافه نمایید که تعداد دانشجویانی که نمره آنها بیشتر از ۱۰ است را در خروجی نمایش دهد.

تمرین ۴ : متدی بنام **findMark** بنویسید که نام دانشجویی را بعنوان پارامتر گرفته و نمره او را بازگرداند.