

راهنمایی درباره نحوه برنامه‌نویسی پروژه:

- ۱- هنگامیکه برنامه‌تان را اجرا می‌کنید، در ابتدای اجرا، برنامه شما از هر فایل، اطلاعات اشیا را خوانده و در آرایه‌ای از اشیا یی مربوطه ذخیره کند.
 - ۲- هنگام نمایش اطلاعات، برنامه باید از آرایه فوق‌الذکر بخواند (در یک حلقه تکرار هر بار اطلاعات یک شیء را با استفاده از متد (Show) از کلاس شیء مربوطه نمایش دهد).
 - ۳- هنگام درج اطلاعات یک شیء جدید، ابتدا اطلاعات را در آرایه فوق‌الذکر ذخیره کند، سپس از همین آرایه بخواند و در فایل مربوطه بنویسد.
 - ۴- هنگام ویرایش اطلاعات یک شیء، ابتدا اطلاعات را در آرایه فوق‌الذکر تغییر دهد، سپس از همین آرایه بخواند و در فایل مربوطه بنویسد (فایل را update (به روز) کند).
 - ۵- هنگام حذف اطلاعات یک شیء، ابتدا شیء مورد نظر را از آرایه فوق‌الذکر حذف کند، سپس از همین آرایه بخواند و در فایل مربوطه بنویسد (فایل را update (به روز) کند).
 - ۶- هنگام جستجو، آرایه‌ای از اشیا را جستجو کند و در صورت یافت شدن، اطلاعات شیء یافته شده را از همین آرایه بخواند و نمایش دهد.
- برای درک بهتر، مطالب را بر روی یک سیستم مثالی مطرح می‌کنیم.

مثال: سیستم اطلاعات یک کتابخانه

در این سیستم، اشیا عبارتند از: کتاب‌ها، مجلات، اعضاء، امانات، کارکنان، و بنابر این در ابتدای اجرا، برنامه باید از فایل کتاب‌ها اطلاعات را بخواند و در آرایه‌ای از کتاب‌ها ذخیره کند. از فایل اعضا بخواند و در آرایه‌ای از اشیا (از نوع عضو کتابخانه) ذخیره کند و به همین ترتیب.

به عنوان مثال class ای داریم به نام Book:

```
class Book
{
    string title ; // نام کتاب
    string author; // نام نویسنده
    MyDate yearOfPublication; // سال انتشار
    string edition; // ویرایش
    string subject; // موضوع کتاب
```

}

فایلی به نام **books** ایجاد می کنیم که در هر سطر آن اطلاع یک کتاب نوشته شده باشد و داده‌های مربوط به فیلدهای مختلف یک کتاب با یک کاما (,) از هم جدا شده باشند.
به عنوان مثال اگر اطلاعات یک کتاب عبارت باشد از:

title : “The Psychology of the Child”
author: “Jean Piaget”
yearOfPublication: 1972
edition: “2nd”
subject: Psychology

و اطلاعات کتاب دوم از قرار زیر باشد:

title: “The C Programming Language”
author: “Dennis M. Ritchie”
yearOfPublication: 1988
edition: “2nd”
subject: “Programming Languages”

می توان اطلاعات کتاب‌ها را به شکل زیر در فایل **Books** ذخیره کرد:

The Psychology of the child, Jean Piaget, 1972, 2nd, Psychology, ...
The C Programming Language, Dennis M. Ritchie, 1988, 2nd, Programming Languages, ...

.
.
.

وقتی برنامه شما شروع به اجرا می کند، ابتدا باید اطلاعات کتاب‌ها را از فایل **books** خوانده و در آرایه‌ای از کتاب‌ها (آرایه‌ای از اشیایی از نوع **Book**) (مثلا به نام **booksArray**) ذخیره کند. پس هر عنصر این آرایه یک **reference** به یک شیء از نوع **book** است.
در خصوص انواع عملیاتی که برنامه باید بتواند انجام دهد:

۱- اگر کاربر نمایش اطلاعات کتاب‌ها را انتخاب کند، برنامه شما باید در یک حلقه جمله‌ی `booksArray[i].Show()` را اجرا کند. (پس عملیات برنامه شما روی آرایه‌ای از کتاب‌ها انجام می‌شود، نه اینکه مستقیم از فایل بخوانید و نمایش دهید، یعنی تفکر `object-oriented` است.)

۲- اگر کاربر جستجوی یک کتاب خاص را انتخاب کند، برنامه باید `booksArray` را جستجو کند و اگر یافت شد، اطلاعات شیء یافت شده در آرایه را بخواند و نمایش دهد. (نمایش با استفاده از متد `Show()` از کلاس `Book`، جستجو با استفاده از متد استاتیک `Search` از کلاس `Book`)

۳- اگر کاربر عمل درج اطلاعات یک کتاب جدید را انتخاب کند، اطلاعات را ابتدا در یک شیء جدید از نوع `Book` ذخیره کرده و `reference` آن را در آرایه‌ی کتاب‌ها اضافه کنید. یعنی:

```
booksArray[currentIndex] = new Book(...);
```

سپس اطلاعات کتاب جدید را از همین آرایه بخوانید و به انتهای فایل `Books` اضافه کنید (عمل اضافه کردن به انتهای فایل `Append` نام دارد.)

۴- به‌طور مشابه اگر کاربر عمل ویرایش اطلاعات یک کتاب را انتخاب کند، ابتدا اطلاعات کتاب مذکور را در آرایه `booksArray` تغییر دهید. سپس اطلاعات فایل `books` را به‌روزرسانی کنید. (برای این کار می‌توانید در یک حلقه تکرار هر بار اطلاعات یک کتاب را از `booksArray` خوانده و در یک سطر فایل `books` ذخیره کنید. کافی است `filemode` را `create` انتخاب کنید تا فایل `books` به‌طور کامل رونویسی (`overwrite`) شود.)

۵- به‌طور مشابه اگر کاربر عمل حذف اطلاعات یک کتاب را انتخاب کند، ابتدا اطلاعات کتاب مذکور را از آرایه `booksArray` حذف کنید. سپس اطلاعات فایل `books` را به‌روزرسانی کنید. (برای این کار می‌توانید در یک حلقه تکرار هر بار اطلاعات یک کتاب را از `booksArray` خوانده و در یک سطر فایل `books` ذخیره کنید. کافی است `filemode` را `create` انتخاب کنید تا فایل `books` به‌طور کامل رونویسی (`overwrite`) شود.)

ضمناً درخصوص نوشتن در فایل و خواندن از فایل می‌توانید به مثال ۳ (StreamWriter) صفحات ۱۲۹ و ۱۳۰ جزوه و مثال ۴ (StreamReader) صفحات ۱۳۰ و ۱۳۱ مراجعه کنید.

فرض کنید می‌خواهیم اولین سطر فایل Books را بخوانیم و در یک شیء از نوع Book ذخیره کنیم.

```
FileStream fs1 = new FileStream(@"d:\myFolder\Books.txt", FileMode.Open);  
StreamReader sr = new StreamReader(fs1);
```

```
int firstIndexOfComma;  
int secondIndexOfComma;  
int thirdIndexOfComma;  
...  
string title, author, yearOfPublication, edition, subject, ... ;
```

```
string str = sr.Read();  
int i = 0; // index of booksArray  
while(str != null)  
{  
    // now str = "The Psychology of the child,Jean Piaget,1972,2nd,Psychology, ..."  
    firstIndexOfComma = str.IndexOf(',');  
    secondIndexOfComma = str.IndexOf(',', firstIndexOfComma + 1);  
    thirdIndexOfComma = str.IndexOf(',', SecondIndexOfComma + 1);  
    .  
    .  
    .  
    title = str.Substring(0, firstIndexOfComma);  
    author = str.Substring( firstIndexOfComma + 1 , SecondIndexOfComma  
        - firstIndexOfComma -1 );  
    ...  
    booksArray[i] = new Book(title, author, yearOfPublication, edition, subject, ...);  
    ...  
    str = sr.Read();  
    i++;  
}
```

نتیجه‌گیری: تفکر و برنامه‌نویسی object-oriented است. برنامه شما اطلاعات را از فایل books می‌خواند و در booksArray می‌ریزد. خواندن اطلاعات کتاب‌ها از booksArray انجام می‌شود و عملیات درج یا تغییر یا حذف اطلاعات، ابتدا بر روی booksArray انجام می‌شود و سپس اطلاعات از booksArray خوانده شده و در فایل books ذخیره می‌شود. یعنی عملیات برنامه شما روی

آرایه‌ای از کتاب‌ها انجام می‌شود، نه اینکه مستقیم از فایل بخوانید و نمایش دهید، یا مستقیم در فایل بنویسید.)

البته برنامه شما عملیات نمایش، درج، ویرایش، حذف، و جستجوی اطلاعات اعضاء، کارمندان، امانات و ... را نیز خواهد داشت و آرایه‌های `membersArray` و `employeesArray` و ... و متناظر آن‌ها فایل‌های `Members` و `Employees` و ... نیز وجود خواهند داشت.

توجه: در برنامه خود از اسامی قابل درک و `comment` استفاده کنید و رسوم برنامه‌نویسی را رعایت نمایید.