زنگ سی‌شارپ – قسمت سی و دوم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در قسمت قبل، در حل تمرین شماره ۱۴ توانستید تک‌آهنگ‌های هر هنرمند را ویراش کنید، در این قسمت قصد داریم آلبوم‌های ذخیره شده را ویراش و همین‌طور تک‌آهنگ‌ها و آهنگ‌های آلبوم هر هنرمند را Play و Stop کنیم.

با ویرایش آلبوم شروع می‌کنیم. پس از این‌که برای هر هنرمند تعدادی آلبوم ذخیره کردید، کاربر باید بتواند تماماً این آلبوم را ویرایش کند. یعنی نام آلبوم، سال انتشار، سبک و آهنگ‌های آلبوم (با تمام جزئیات هر آهنگ) باید قابل ویرایش باشد و در نهایت باید این قابلیت وجود داشته باشد که آلبوم یا آهنگ‌های آن را بتوان حذف کرد.

بعد از آن‌که هنرمند مورد نظر را در برنامه انتخاب کردید، گزینه‌های مختلفی را مشاهده می‌کنید که به شرح زیر است:

```
1. Edit      2. Delete      3. Add Album      4. Add Single Tune
5. Play Single Tunes        6. Play Albums    7. Back

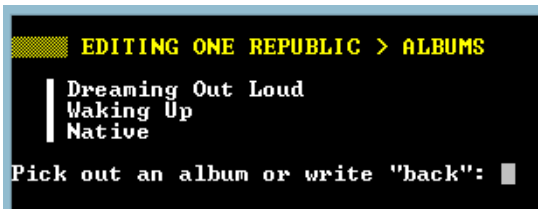
Pick out an action(choose correspond number):
```

در قسمت قبل، در گزینه‌های ۵ و ۶ عبارات View Single Tunes و View Albums قرار داشت که در این قسمت آن‌ها را به Play Single Tunes و Play Albums تغییر دادیم. در قسمت قبل با انتخاب گزینه‌های ۵ و ۶ می‌توانستید آلبوم‌ها و تک‌آهنگ‌های ذخیره شده را مشاهده کنید، تفاوت در این‌جاست که در این قسمت پس از مشاهده‌ی آن‌ها می‌توانید آهنگ مورد نظر خود را انتخاب و به آن گوش کنید. در انتهای این مقاله در مورد Play و Stop کردن موزیک‌ها بحث خواهیم کرد. وقتی گزینه‌ی شماره ۱ را انتخاب می‌کنید، گزینه‌های بعدی برای شما نمایش داده می‌شود:

```
1. Edit name and family      2. Edit Albums
3. Edit single tunes         4. Back

Pick out an action(choose correspond number):
```

در قسمت قبل با گزینه‌های ۱ و ۳ آشنا شدید و دانستید که به چه صورت این ویرایش‌ها انجام می‌شوند. در این قسمت به سراغ گزینه‌ی ۲، Edit Albums، می‌رویم. با فرض این‌که هنرمندی به‌نام OneRepublic داشته و سه آلبوم برای آن داشته باشیم، پس از انتخاب گزینه‌ی ۲ آلبوم‌های OneRepublic را می‌بینیم که باید به‌منظور Edit یکی را انتخاب کنیم:



برای نمونه، آلبوم Native را انتخاب می‌کنیم. پس از آن‌که Native را نوشته و Enter زدیم، گزینه‌های زیر برای ما نمایش داده می‌شود:



در این‌جا شما می‌توانید تمام جزئیات یک آلبوم را ویرایش کنید. به‌منظور ویرایش و دیدن اطلاعات هر آلبوم، تعدادی متد برای کلاس Album در نظر گرفتیم. به کلاس Album مجدداً توجه کنید:

```
using System;
namespace MusicBoxPractice
{
    class Album
    {
        // Fields
        private string AlbumName;
        private string AlbumOwner;
        private string AlbumGenre;
        private ushort AlbumYear;
        private Tune[] Tunes;

        // Constructor
        public Album(string albumName, Artist artist,
            string albumGenre, ushort albumYear, Tune[] tunes)
        {
            AlbumName = albumName;
            AlbumOwner = artist.GetArtistFullName();
            AlbumGenre = albumGenre;
            AlbumYear = albumYear;
            Tunes = tunes;
        }

        public string GetAlbumName() { return AlbumName; }
        public void SetAlbumeName(string name) { AlbumName = name; }

        public string GetAlbumGenre() { return AlbumGenre; }
        public void SetAlbumGenre(string genre) { AlbumGenre = genre; }

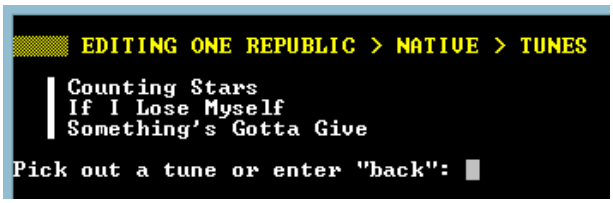
        public ushort GetAlbumYear() { return AlbumYear; }
        public void SetAlbumYear(ushort year) { AlbumYear = year; }

        public Tune[] GetTunes() { return Tunes; }
```

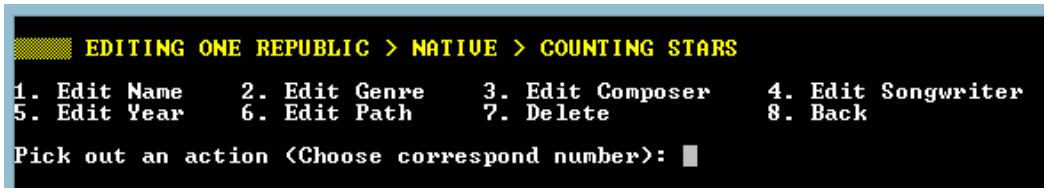
```
        public void RemoveTune(byte index)
        {
            Tunes[index] = null;
        }
    }
}
```

به‌این ترتیب برای ویرایش (مثلاً) نام آلبوم کافی است که نام جدید را از کاربر دریافت و توسط متد SetAlbumName() نام را جدید را ثبت کنیم. سال و سبک آلبوم نیز به همین روش تغییر می‌یابد. برای حذف کردن یک آهنگ از آلبوم کافی است که index آن خانه از آرایه که آهنگ مورد نظر در آن ذخیره شده است را به متد RemoveTune() بدهید.

گزینه‌ی ۴، Edit Tunes، برای ویرایش آهنگ‌های آلبوم است. برای آلبوم Native از OneRepublic، سه آهنگ در نظر گرفته‌ایم. بعد از انتخاب گزینه‌ی چهار، این سه آهنگ را می‌بینید و به‌منظور ویرایش، باید یکی را انتخاب کنید:



با فرض این‌که آهنگ Counting Stars را انتخاب کنیم، بعد از انتخاب آن گزینه‌های زیر را می‌بینیم:



در این‌جا، ویرایش آهنگ‌های آلبوم مانند ویرایش تک‌آهنگ‌ها است. کافی است که (مثلاً) با انتخاب گزینه‌ی ۲، ژانر جدید را وارد و آن را جایگزین ژانر قبلی کنید.

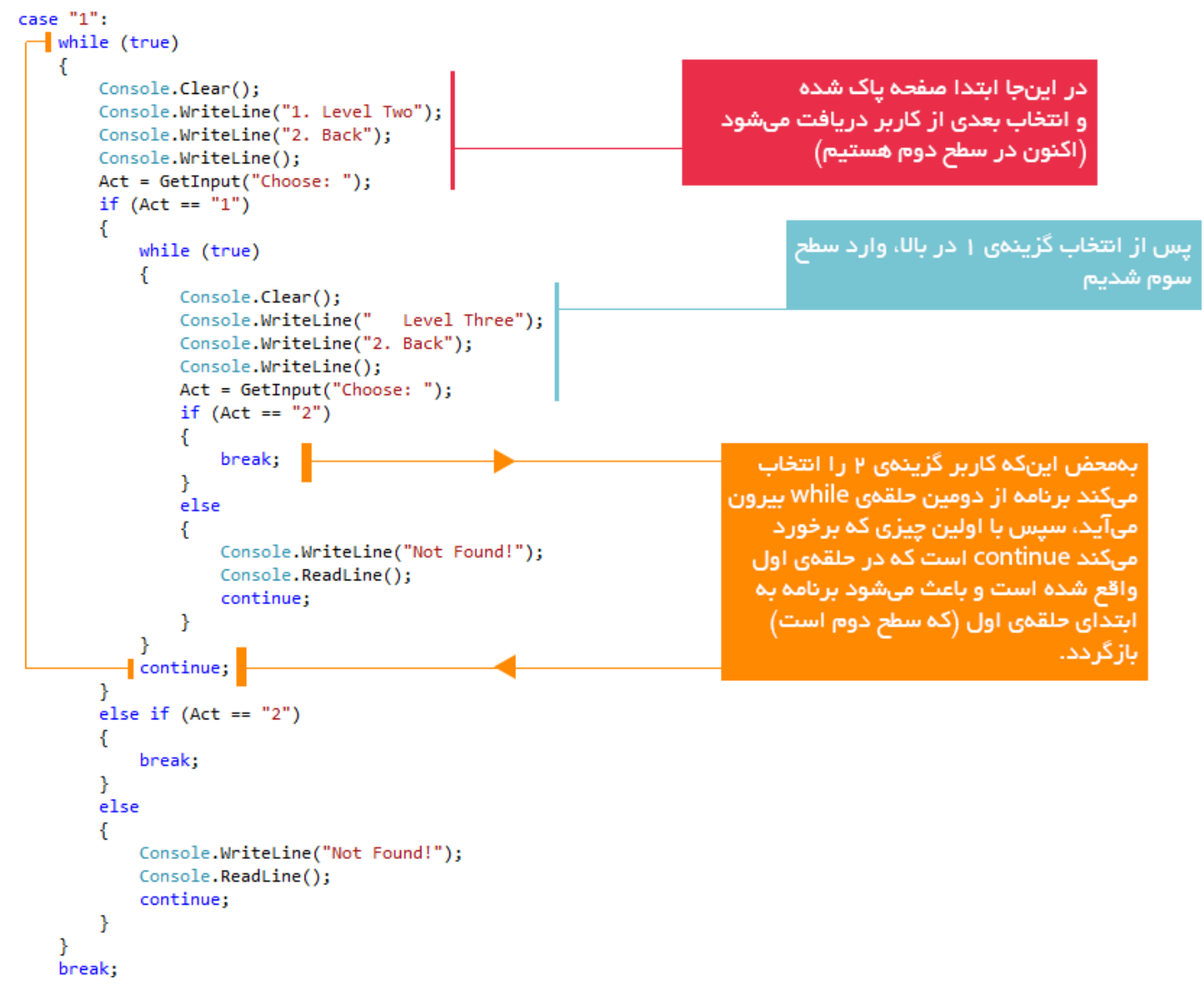
نکته‌ی قابل توجه در این برنامه گزینه‌ی Back است. می‌بینید که با انتخاب بعضی از گزینه‌ها، صفحه‌ی Console پاک شده و موارد دیگری برای شما نمایش داده می‌شود و در واقع شما یک سطح جلوتر می‌روید و با انتخاب Back یک سطح به عقب برمی‌گردید. برای این‌که با این روش آشنا شوید به برنامه‌ی ساده زیر که به‌این منظور نوشته شده است، دقت کنید:

```
using System;
class MyClass
{
    static void Main()
    {
        Test ob = new Test();
        while (true)
        {
```

```
            ob.Proccess(ob.ShowMenu());
        }
    }
}
class Test
{
    string Act;
    public string ShowMenu()
    {
        Console.Clear();
        Console.WriteLine("1. Level One");
        Console.WriteLine("2. Exit");
        Console.WriteLine();
        Console.Write("Choose: ");
        return Console.ReadLine();
    }
    public void Proccess(string choice)
    {
        switch (choice)
        {
            case "1":
                while (true)
                {
                    Console.Clear();
                    Console.WriteLine("1. Level Two");
                    Console.WriteLine("2. Back");
                    Console.WriteLine();
                    Act = GetInput("Choose: ");
                    if (Act == "1")
                    {
                        while (true)
                        {
                            Console.Clear();
                            Console.WriteLine("   Level Three");
                            Console.WriteLine("2. Back");
                            Console.WriteLine();
                            Act = GetInput("Choose: ");
                            if (Act == "2")
                            {
                                break;
                            }
                            else
                            {
                                Console.WriteLine("Not Found!");
                                Console.ReadLine();
                                continue;
                            }
                        }
                        continue;
                    }
                    else if (Act == "2")
                    {
                        break;
                    }
                    else
                    {
                        Console.WriteLine("Not Found!");
                        Console.ReadLine();
                        continue;
                    }
                }
```

```
                break;
            case "2":
                Environment.Exit(0);
                break;
        }
    }
    public string GetInput(string message)
    {
        Console.Write(message);
        return Console.ReadLine();
    }
}
```

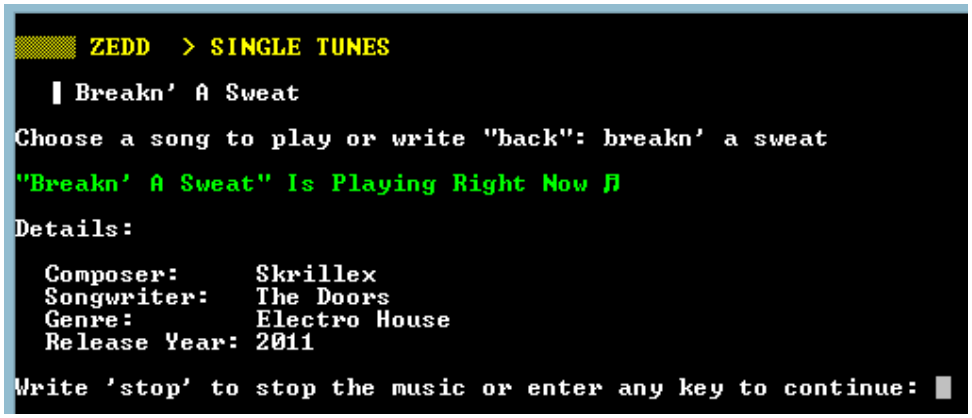
اگر برنامه بالا را اجرا کنید متوجه می‌شوید که سه سطح در این برنامه وجود دارد. وقتی برنامه اجرا می‌شود کاربر دو گزینه دارد، یا باید سطح یک را انتخاب کند یا از برنامه خارج شود. اگر کاربر گزینه‌ی ۱ انتخاب کند وارد سطح دوم می‌شود و سپس:



به منظور پخش تک‌آهنگ‌هایی که ذخیره کردید، گزینه‌ی ۵ را انتخاب می‌کنید. با فرض این‌که برای هنرمندی به اسم Zedd تک‌آهنگی به‌نام Breakn' A Sweat افزوده باشید، با انتخاب گزینه‌ی ۵ کلیه‌ی تک‌آهنگ‌های Zedd برای شما نمایش داده می‌شود، در این‌جا تنها یک تک‌آهنگ (Breakn' A Sweat) برای این هنرمند در نظر گرفته‌ایم:



بعد از انتخاب تک‌آهنگ مورد نظر، موزیک Play شده و جزئیات آن نمایش داده می‌شود:



برای Stop کردن موزیک کافی است که کلمه‌ی Stop را وارد کنید، در غیر این‌صورت با وارد کردن هر کلید دیگری برنامه به مرحله‌ی قبل باز می‌گردد و موسیقی نیز درحال پخش می‌ماند. موسیقی شما باید دارای پسوند wav. باشد تا Play شود زیرا Net Framework. به‌صورت پیش‌فرض از پسوندهای دیگر مثل mp3. پشتیبانی نمی‌کند و در این‌جا نیز ما از همین پسوند برای پخش کردن موزیک استفاده می‌کنیم. برای پخش موسیقی باید `using System.Media`را به کلاس UI اضافه کنید تا بتوانید به کلاس SoundPlayer برای Play کردن موسیقی دسترسی داشته باشید.

به نمونه‌ی زیر دقت کنید:

```
using System;
using System.Media;
class Program
{
    static void Main(string[] args)
    {
        SoundPlayer myPlayer = new SoundPlayer(@"C:\Daft Punk\Instant Crush.wav");
        myPlayer.Play();
        //myPlayer.Stop();
        Console.ReadLine();
    }
```

```
}
```

همان‌طور که می‌بینید در constructor کلاس SoundPlayer آدرس آهنگ مورد نظر را قرار داده‌ایم که در پروژه این آدرس در متغیر Path قرار می‌گیرد. توجه کنید که قبل از رشته آدرس باید از علامت @ استفاده کنید تا علائمی مثل \ به‌عنوان بخشی از آدرس در نظر گرفته شده و به‌عنوان escape sequence در نظر گرفته نشوند. توسط متد ()Play آهنگ پخش و توسط متد ()Stop آهنگ متوقف می‌شود. در گزینه‌ی ۶، Play Albums، ابتدا لیستی از آلبوم‌ها نشان داده شده و سپس با انتخاب آلبوم و سپس انتخاب یکی از آهنگ‌های آن آلبوم، موزیک مورد نظر Play می‌شود.

یکی دیگر از متدهایی که در این برنامه از آن استفاده کرده‌ایم، متد ()Capitalize است:

```
public static string Capitalize(string title)
{
        CultureInfo cultureInfo = Thread.CurrentThread.CurrentCulture;
        TextInfo textInfo = cultureInfo.TextInfo;
        return textInfo.ToTitleCase(title);
}
```

دلیل استفاده از این متد این بود که بتوانیم حروف اول یک عبارت را به‌صورت بزرگ نمایش دهیم. به‌عنوان مثال عبارت this is a text به‌صورت This Is A Text نمایش داده شود.

حل تمرین شماره ۱۴ در این قسمت به پایان رسید و نسخه‌ی نهایی پروژه را می‌توانید از این‌جا دانلود کنید. قابل ذکر است که این پروژه با توجه به مواردی که تا قبل از این تمرین گفته شده بود انجام شده است و بدون شک روش‌های بهتر و زیباتری برای انجام چنین تمرین‌های وجود دارد که با دنبال کردن قسمت‌های بعدی زنگ سی‌شارپ با آن‌ها آشنا خواهید شد. لطفاً پروژه را دانلود کرده و تا آن‌جا که می‌توانید کدها را برای خودتان تجزیه و تحلیل کنید هرچند که زمان‌بر باشد زیرا تا زمانی‌که روی این موضوعات و تجزیه و تحلیل کدهای نمونه‌ای که می‌بینید و می‌نویسید وقت و انرژی صرف نکنید نمی‌توانید توقع داشته باشید که خودتان هم روزی پروژه‌ی بزرگی انجام دهید. تاکید می‌شود که برای بهتر شدن در برنامه‌نویسی، زبان انگلیسی خود را قوی کنید و پیوسته مطالعه و تمرین را در کنار هم قرار دهید.

کلیه حقوق مادی و معنوی برای وب‌سایت وب‌تارگت محفوظ است.
استفاده از این مطلب در سایر وب‌سایت‌ها و نشریات چاپی تنها با ذکر و درج لینک منبع مجاز است.