





## پودمان پنجم

### واحد یادگیری ۹ و ۱۰

#### مدیریت پایگاه داده

## واحد یادگیری ۹

### شایستگی کار با پرونده‌ها، پوشه‌ها و درایوها

#### مقدمات تدریس

#### الف) مفاهیم کلیدی

مفاهیم کلیدی			
پرونده متنی	پرونده دودویی	ایجاد پرونده	نوشتن در پرونده
خواندن از پرونده	حذف پرونده	کپی پرونده	مدیریت استثنا
فضای نام System.IO	کلاس File	کلاس Directory	کلاس Path

#### ب) مراحل انجام کار واحد یادگیری

مراحل انجام کار	
۱	ایجاد پرونده متنی و نوشتن در آن
۲	خواندن از پرونده متنی
۳	عملیات روی پرونده‌ها و پوشه‌ها در برنامه

#### ج) تجهیزات لازم

الزامات نرم‌افزاری:

■ نرم‌افزار Visual Studio Express 2012

تجهیزات سخت‌افزاری:

■ پردازنده حداقل ۱/۶ گیگاهرتز

■ حافظه RAM حداقل ۱ گیگابایت

■ حداقل فضای موجود در دیسک سخت ۴ گیگابایت

■ کارت ویدئویی متناسب با DirectX9 و قابل اجرا در رزولوشن  $۱۰۲۴ \times ۷۶۸$  و بالاتر

#### د) بودجه بندی

برای بودجه‌بندی فصل پنجم می‌توانید از نمونه پیشنهادی استفاده کنید.

جلسه	واحد یادگیری	کارگاه (موضوع)	شماره صفحات	اهداف توانمندسازی	فعالیت‌های تکمیلی
۲۹	۹	آشنایی با پرونده و کارگاه ۱ و ۲ و ۳ و ۴	۲۱۸ تا ۲۲۷	<ul style="list-style-type: none"> <li>- آشنایی با مفهوم پرونده و انواع آن</li> <li>- آشنایی با متدهای کلاس File برای نوشتن در پرونده</li> <li>- تشخیص متد مورد نیاز برای نوشتن در پرونده متنی</li> <li>- آشنایی با متدهای کلاس File برای خواندن از پرونده</li> <li>- تشخیص متد مورد نیاز برای خواندن از پرونده متنی</li> <li>- تشخیص وجود پرونده در مسیر مورد نظر</li> </ul>	نوشتن در پرونده به فرمت xml و inf توسعه برنامه Notepad و نوشتن در قالب RTF
۳۰	۹	کارگاه ۵ و ۶ و ۷	۲۲۸ تا ۲۳۴	<ul style="list-style-type: none"> <li>- آشنایی با استثناها در برنامه</li> <li>- مدیریت استثنا در برنامه</li> <li>- آشنایی با کنترل نگهبان پرونده و اطلاع از تغییرات پرونده</li> <li>- نوشتن و خواندن از یک پرونده متنی مشترک و استفاده از آن</li> </ul>	توسعه پروژه تالار گفتگو و افزودن منو و قابلیت ارسال استیکر به آن
۳۱	۹	کارگاه ۸ و ۹ و ۱۰	۲۳۵ تا ۲۴۵	<ul style="list-style-type: none"> <li>- آشنایی با متدهای کلاس Directory و کلاس Path برای مشاهده فهرست پرونده‌ها و پوشه‌ها</li> <li>- مشاهده فهرست پرونده‌ها و پوشه‌ها</li> <li>- آشنایی با متدهای کلاس Directory و کلاس File برای حذف، کپی و جابجایی پرونده‌ها</li> <li>- انجام عملیات حذف، کپی و جابجایی پرونده‌ها</li> </ul>	توسعه کارگاه ۸ و افزودن قابلیت باز و بسته کردن درایو CD یا DVD به برنامه افزودن قابلیت پخش صوت و نمایش تصویر به کارگاه ۹
۳۲	۹	پروژه سیستم خودپرداز و توسعه آن	۲۴۶ تا ۲۴۹	ثبت وقایع برنامه در یک پرونده متنی	نوشتن برنامه ماشین حساب با درج محاسبات در پرونده متنی
	۱۰	طراحی پایگاه داده	۲۵۰ تا ۲۵۳	طراحی پایگاه داده مورد نیاز	از هنرجویان بخواهید به صورت گروهی در مورد تجهیزات هنرستان با سرپرست کارگاه‌ها مصاحبه کنند و پایگاه داده مناسب را طراحی کنند.

جلسه	واحد یادگیری	کارگاه (موضوع)	شماره صفحات	اهداف توانمندسازی	فعالیت‌های تکمیلی
۳۳	۱۰	کارگاه ۱ و ۲ و ۳ و ۴ و ۵	۲۶۱ تا	<ul style="list-style-type: none"> <li>- آشنایی با ابزارهای ارتباط با پایگاه داده</li> <li>- آشنایی با ویژگی‌های کنترل DataGridView</li> <li>- آشنایی با اتصال کادرمتن و DataGridView به منبع داده</li> <li>- آشنایی با متدهای کلاس BindingSource برای حرکت بین رکوردهای جدول</li> <li>- دسترسی به خانه‌های DataGridView</li> </ul>	<p>نحوه ایجاد منبع داده (Data Source) در VS2012 و اضافه کردن به فرم</p> <p>- بررسی متدهای دیگر کنترل BindingSource و کنترل DataGridView</p>
۳۴	۱۰	کارگاه ۶ و ۷ و ۸ و ۹ و ۱۰	۲۶۲ تا ۲۷۰	<ul style="list-style-type: none"> <li>- جستجو دقیق و مشابه واژه در جدول</li> <li>- جستجو ترکیبی در جدول</li> <li>- آشنایی با ایجاد متدهای درج و حذف و ویرایش رکورد و فراخوانی آن در برنامه</li> </ul>	<p>- بررسی عملگر Like در زبان SQL و استفاده در جدول مشتری</p> <p>- بررسی ایمن بودن اعتبارسنجی کاربر به وسیله هنرجویان</p> <p>- به وسیله هنرجویان بررسی شود که اگر جدول جدیدی در پایگاه داده فروشگاه ایجاد شود چگونه متدهای درج و حذف و ویرایش برای آن ایجاد گردد.</p>
۳۵	۱۰	کارگاه ۱۱ و ۱۲ و ۱۳	۲۷۱ تا ۲۷۶	<ul style="list-style-type: none"> <li>- آشنایی با روش‌های درج و حذف سطر از کنترل DataGridView</li> <li>- کنترل خطا در برنامه پایگاه داده</li> <li>- آشنایی با نحوه درج همزمان چندین رکورد در یک پایگاه داده ارتباطی</li> </ul>	<p>امکانات درج و ویرایش در کنترل DataGridView قرار گیرد.</p> <p>پیام خطای مناسب هر خطا در برنامه نمایش داده شود.</p> <p>از هنرجویان بخواهید بررسی کنند چگونه می‌توان یک برگه خرید را در پایگاه داده فروشگاه حذف کرد؟</p>

## زمان‌بندی واحد یادگیری ۹

ردیف	اهداف توانمندسازی		سهم زمان	
	تئوری	عملی	تئوری	عملی
۱	آشنایی با مفهوم پرونده و انواع آن		۲۰ دقیقه	۱
۲	آشنایی با متدهای کلاس File برای نوشتن در پرونده		۲۰ دقیقه	
۳	تشخیص متد مورد نیاز برای نوشتن در پرونده متنی		۲۰ دقیقه	۲
۴	آشنایی با متدهای کلاس File برای خواندن از پرونده		۲۰ دقیقه	
۵	تشخیص متد مورد نیاز برای خواندن از پرونده متنی		۲۰ دقیقه	۲
۶	بررسی وجود پرونده در مسیر مورد نظر		۲۰ دقیقه	۱
۷	آشنایی با استثناها در برنامه		۲۰ دقیقه	۱
۸	مدیریت استثنا در برنامه		۲۰ دقیقه	۲
۹	آشنایی با کنترل نگهبان پرونده و اطلاع از تغییرات پرونده		۲۰ دقیقه	۱
۱۰	نوشتن و خواندن از یک پرونده متنی مشترک و استفاده از		۲۰ دقیقه	۳
۱۱	آشنایی با متدهای کلاس Directory و کلاس Path برای		۲۰ دقیقه	۱
۱۲	مشاهده فهرست پرونده‌ها و پوشه‌ها		۲۰ دقیقه	۳
۱۳	آشنایی با متدهای کلاس Directory و کلاس File برای		۲۰ دقیقه	۱
۱۴	انجام عملیات حذف، کپی و جابجایی پرونده‌ها		۲۰ دقیقه	۲
۱۵	ثبت وقایع برنامه در یک پرونده متنی		۲۰ دقیقه	۳
۱۶	ارزشیابی پایانی		-	۲
مجموع ساعت			۵	۲۵

در زمان‌های ذکرشده حداکثر ۲۰ دقیقه به توضیحات هنرآموز و مابقی زمان به کار عملی هنرجویان و رفع مشکلات عملی اختصاص می‌یابد.

## ه) فیلم‌ها و پرونده‌های پیوست

شماره فیلم	هدف
111h46	پروژه تالار گفتگو و توسعه آن
111h47	پروژه شبیه ساز سیستم خودپرداز بانکی و توسعه آن
111h48	برنامه نقاشی

نکته: برای آشنایی با کد پروژه‌ها و سایر مباحث تکمیلی می‌توانید از پرونده پیوست «مطالب و فعالیت‌های تکمیلی» استفاده کنید.

## و) ورود به بحث

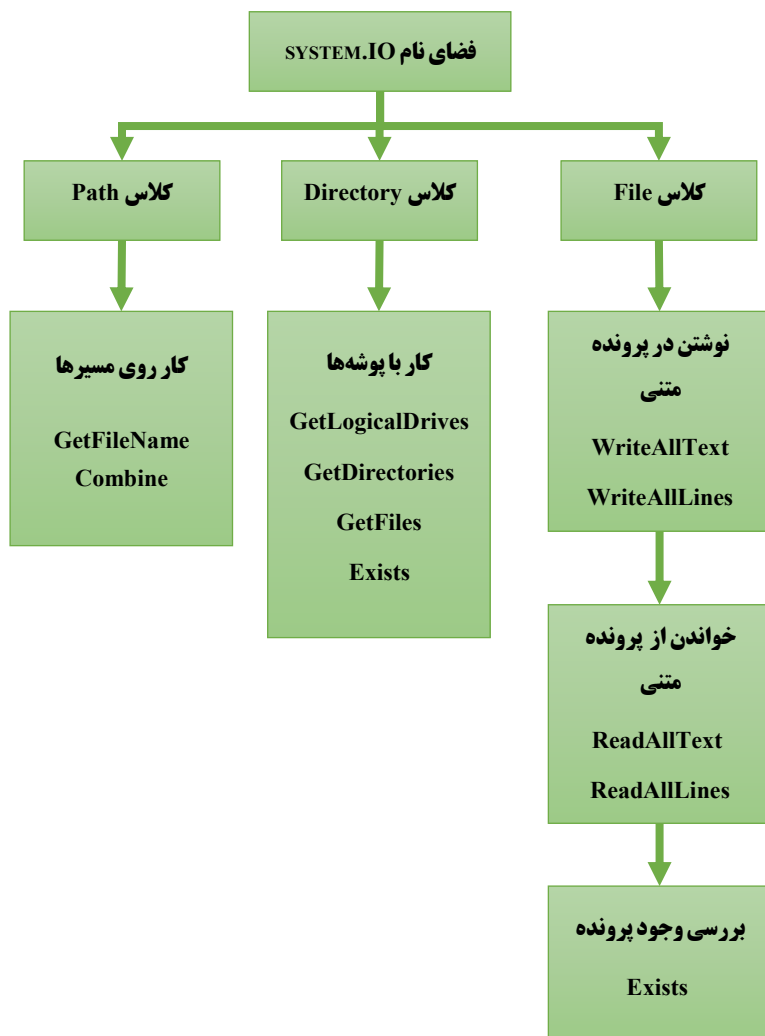
شیوه پیشنهادی برای ارائه‌ی این واحد یادگیری روش تدریس پروژه محور است. در این روش هنرآموز، هنرجویان را به گروه‌هایی تقسیم کرده، بعد از مشخص نمودن هدف درس و بیان پایه‌های تدریس، آنها با فعالیت گروهی، کارکرد گزینه‌های مختلف را کشف می‌کنند. می‌توانید هر قسمت از درس را به یک گروه محول کرده و پس از پایان زمان تعیین‌شده، یکی از اعضاء گروه مفاهیم کلیدی را که فراگرفته‌اند، به دیگران توضیح دهد. در انتها بهتر است هنرآموز جمع‌بندی و تکمیل مبحث را انجام دهد.

### سؤالات تعیین سطح

- اگر در زمان کار با یک نرم‌افزار مانند ویژوال استودیو کار خود را ذخیره نکرده باشیم و سیستم را خاموش کنیم چه اتفاقی برای اطلاعات ما رخ می‌دهد؟
- چگونه پس از خاموش شدن سیستم، اطلاعات حفظ می‌شوند؟
- نوشته‌ها چگونه در سیستم ذخیره می‌شوند؟
- تصویر، صوت و ویدیو چگونه در سیستم ذخیره می‌شوند؟
- اگر یک پرونده تصویری را در برنامه Notepad بازکنیم، چه چیزی خواهیم دید؟

در این واحد یادگیری هنرجویان با مقدمات ذخیره پایدار داده‌های متنی در حافظه جانبی و بازیابی داده‌ها از حافظه جانبی آشنا می‌شوند. چگونگی ایجاد پرونده متنی، نوشتن داده‌ها در پرونده متنی و خواندن داده‌ها از پرونده متنی، نحوه کار با پوشه‌ها شامل ایجاد پوشه، حذف پوشه و پرونده‌های آن و همچنین ایجاد فهرست از محتویات درایوها، پوشه‌ها و پرونده‌های موجود در یک مسیر مشخص، در قالب کارگاه‌های مختلف آموزش داده شده و تمرین می‌شود.

کلاس‌ها و متدهایی که در این واحد یادگیری معرفی شده و به کارگرفته می‌شوند در نمودار آمده است.



پیش نیازهای لازم برای انجام فعالیت‌های این واحد یادگیری

- ۱) آشنایی با نحوه طراحی واسط کاربری
- ۲) آشنایی با کنترل‌های پرکاربرد مانند کنترل برچسب، کادر متنی و ...
- ۳) آشنایی با مفهوم پوشه و پرونده و توانایی مدیریت پوشه‌ها و پرونده‌ها در سیستم عامل
- ۴) توانایی ایجاد و ذخیره پرونده متنی در ویرایشگر Notepad



## پرونده و انواع آن

### متن ساده (Plain text)

در علوم رایانه، متن ساده (Plain text) به داده‌هایی مانند محتویات پرونده گفته می‌شود که فقط از کاراکترهای قابل خوانده شدن تشکیل شده‌اند و هیچ نمایش گرافیکی مربوط به کاراکترها و یا اشیاء دیگر مانند تصاویر، قلم و نظایر آن را شامل نمی‌شود. متن ساده همچنین می‌تواند شامل تعداد محدودی کاراکترهای کنترلی مثل کاراکتر انتهای خط یا کاراکتر جدول بندی (tab) باشد که چیدمان ساده متن را کنترل می‌کنند. متن ساده با متن قالب بندی شده که اطلاعات مربوط به سبک و قالب بندی را شامل می‌شود و همچنین با پرونده‌های دودویی که رکوردهای آنها باید به صورت فیلدهای دودویی متفاوتی تفسیر شوند (مثلاً اعداد صحیح، اعداد حقیقی، رشته متنی، تصاویر و...) تفاوت دارد. به طور سنتی در کدگذاری متن ساده از استاندارد اسکی (ASCII) با محدودیت کد ۷ یا ۸ بیت استفاده می‌شود. هرچند کدگذاری یونیکد مانند UTF-8 یا UTF-16 در حال جایگزینی با کدهای اسکی است.

روشی که برای تمیز دادن پرونده‌های متنی ساده از پرونده‌های دودویی در اینجا پیشنهاد شده است، استفاده از یک ویرایشگر متنی ساده مانند Notepad است. البته باید توجه داشت که این روش یک ملاک و معیار قطعی برای تشخیص پرونده متنی از پرونده دودویی نیست. به عنوان مثال پرونده‌های از نوع rtf یک قالب ترکیبی از متن و محتویات دودویی شامل تصاویر هستند. در ضمن باید این نکته را هم در نظر داشت که یک واژه پرداز مانند word محتوای پرونده‌های متنی ساده را به درستی نشان می‌دهد اما اگر پرونده‌ای در واژه پرداز word به درستی نشان داده شد، این پرونده لزوماً یک پرونده متنی ساده نیست. به عنوان مثال پرونده‌های از نوع pdf که پرونده دودویی محسوب می‌شوند توسط واژه پرداز word نسخه ۲۰۱۳ به بعد باز می‌شوند.

### پاسخ به فعالیت‌ها

در جدول زیر با کمک هم گروهی خود مشخص کنید، کدام پرونده‌ها از نوع دودویی و کدام پرونده‌ها از نوع متنی هستند. سعی کنید این نوع از پرونده‌ها را با برنامه Notepad باز کنید. چه تفاوتی مشاهده می‌کنید؟

MP3 Audio File	C# source code	Microsoft Word Document	Adobe pdf Document	html
دودویی	متنی	دودویی	دودویی	متنی

فعالیت گروهی  
ص ۲۲۰

از هنرجوها بخواهید که ابتدا نوع پرونده را از نظر محتوا حدس بزنند؛ سپس در یک پوشه انواع پرونده‌های خواسته شده را گردآوری کرده، سعی کنند با برنامه Notepad آن‌ها را باز کنند.

علاوه بر این می‌توانید از هنرجوها بخواهید که انواع دیگری از پرونده‌ها مانند پرونده‌های اجرایی (exe,dll)، ویدیویی (avi,mpg,mp4)، تصویری (jpg,tiff,gif) و پرونده‌های متنی با پسوندهای log, inf, bat, xml را روی سیستم خود جستجو کرده، آن‌ها را به این ترتیب امتحان کنند.

از نظر نحوه دسترسی، پرونده‌ها به دو دسته با دسترسی ترتیبی و دسترسی مستقیم تقسیم می‌شوند. ذخیره‌سازی اطلاعات و دسترسی به پرونده‌های متنی، ترتیبی است در صورتی که برای پرونده‌های دودویی از روش ذخیره‌سازی و دسترسی مستقیم استفاده می‌شود.

از هنرجوها بپرسید که آیا نوار کاست و یا نوار VHS را دیده‌اند؟ از آنها بپرسید که انتخاب آهنک در نوار کاست با لوح فشرده چه تفاوتی دارد؟

#### فعالیت گروهی ص ۲۲۰

شیوه دسترسی به داده‌های حافظه یا محتویات رسانه‌ای را در هر یک از موارد زیر تعیین کنید.

حافظه دیجیتال یا رسانه ذخیره	نحوه دسترسی به حافظه یا محتویات
حافظه فلش	دسترسی مستقیم
لوح فشرده	دسترسی مستقیم
نوار کاست	دسترسی ترتیبی
نوار ویدیویی VHS	دسترسی ترتیبی

متدهایی که برای خواندن از پرونده متنی و نوشتن در پرونده متنی استفاده می‌شوند به صورت ترتیبی عمل می‌کنند. برای خواندن یک خط مشخص از پرونده متنی می‌توان از متد `File.ReadLines` به صورت زیر استفاده کرد.

```
string line = File.ReadLines(FileName).Skip(23).Take(1).First();
```

این دستور ۲۳ خط ابتدای پرونده با نام `FileName` را رد کرده، محتوای خط شماره ۲۴ آن را خوانده و در متغیر `line` ذخیره می‌کند. باید توجه داشت که این دستور نیز به صورت ترتیبی عمل می‌کند.

#### پژوهش

ص ۲۲۰

درباره روش‌های ذخیره‌سازی ابری تحقیق کنید و نتیجه را در کلاس ارائه دهید.

پاسخ: فضای ابری مجموعه‌ای از کامپیوترهای به هم متصل است که به عنوان یک سیستم واحد عمل می‌کنند. فضاهای ابری یک یا چند سرویس مختلف مانند

سرویس فضای ذخیره سازی، سرویس ارسال و دریافت داده و یا امکان استفاده از نرم افزارهای مختلف را به کاربران راه دور عرضه می کنند. در آینده استفاده از این سرویس ها بسیار گسترده و فراگیر خواهد شد. از سرویس های معروف ذخیره سازی ابری (Cloud storage) می توان به گوگل درایو (Google Drive)، وان درایو (Microsoft OneDrive) و دراپ باکس (Dropbox) اشاره کرد. دسترسی به فضای ذخیره سازی ابری نیازمند یک اتصال اینترنتی با سرعت مناسب است. ذخیره سازی ابری به کاربران این امکان را می دهد تا به اطلاعاتشان در هر زمان و مکانی دسترسی داشته باشند.

## کارگاه ۱ – ایجاد پرونده و نوشتن در آن

در این کارگاه برای نوشتن داده رشته ای در پرونده متنی دو متد WriteAllText و AppendAllText معرفی شده، تفاوت آنها بررسی می شود. در زمان طراحی فرم دقت شود که فرم طراحی شده هنجاریان با محتوای فرم تناسب داشته باشد و کنترل ها به صورت منظم و مطابق شکل ارائه شده چیده شوند. از دیگر موارد مطرح شده در این کارگاه استفاده از دنباله "\r\n" به عنوان نویسه خط جدید (New Line) و "\t" برای ایجاد یک Tab فاصله در پرونده متنی است. در این کارگاه با زدن دکمه ثبت، اطلاعات فردی شامل نام، نام خانوادگی، نام پدر و کد ملی در پرونده متنی ثبت می شود.

```
string newContact = txtName.Text + "\t" +  
    txtFamily.Text + "\t" +  
    txtFatherName.Text + "\t" + txtNationalCode.Text;  
File.AppendAllText("Info.txt", newContact + "\r\n");
```

برای توسعه این کارگاه می توانید از هنجاریان بخواهید برای هر رکورد از اطلاعات فردی و نیز برای هر کدام از فیلدها یک برچسب (tag) به صورت زیر ایجاد کنند.

[Person]

[Name]علی

[LastName]احمدی

[FatherName]رضا

[NID]۱۲۵۴۷۸۹۶۵۴

برای این منظور یک متغیر رشته ای با نام newRecord برای ذخیره رکورد جدید تعریف کرده و آن را مطابق قالب مشخص شده، با داده های ورودی مقداردهی کنید.

```
string newRecord = "[Person]\r\n" +  
    "[Name]" + txtName.Text + "\r\n" +  
    "[LastName]" + txtFamily.Text + "\r\n" +
```

```
"[FatherName]" + txtFatherName.Text + "\r\n" +
"[NID]" + txtNationalCode.Text;
```

```
File.WriteAllText("ContactList.ini", newRecord + "\r\n");
```

سیستم عامل ویندوز از روشی مشابه برای نگهداری برخی داده‌های مرتبط با پیکربندی خود در قالب پرونده‌هایی با پسوند ini یا inf استفاده می‌کند. به عنوان مثال پرونده‌های win.ini و system.ini را در پوشه نصب ویندوز ببینید.

کنجکاو: اگر بخواهیم اطلاعات در یک مسیر مشخص ذخیره شود، متدهای نوشتن در پرونده را چگونه فراخوانی می‌کنید؟

باید آدرس پرونده را به عنوان ورودی به این متدها داد. آدرس ورودی می‌تواند مطلق یا نسبی باشد.

در روش آدرس دهی مطلق، آدرس از ریشه درایو داده می‌شود. به فرض اگر شما در هنرستان هستید و می‌خواهید به دوست خود آدرس یک سینما را بدهید، می‌توانید آدرس سینما را از میدان مرکزی شهر مشخص کنید. دستور زیر نحوه آدرس دهی مطلق را نشان می‌دهد:

```
File.WriteAllText(@"C:\Users\Admin\Desktop>ContactList.txt",
newContact + "\r\n");
```

در روش آدرس دهی نسبی، آدرس از مکان پوشه کاری جاری داده می‌شود. به فرض اگر محل فعلی شما را داخل هنرستان فرض کنیم شما می‌توانید آدرس یک مجتمع ورزشی را که چند کوچه یا خیابان با هنرستان فاصله دارد از محل هنرستان مشخص کنید. دستور زیر یک پرونده را در پوشه data که در مسیر پوشه کاری جاری قرار دارد ایجاد می‌کند.

```
File.WriteAllText("data\\ContactList.txt", newContact + "\r\n");
```

برای گرفتن آدرس پوشه کاری جاری می‌توان از متد GetCurrentDirectory از کلاس Directory استفاده کرد. همچنین در کلاس Path متدهایی برای کار با آدرس پرونده‌ها و پوشه‌ها وجود دارد که در ادامه این واحد یادگیری هنجریان با برخی از متدهای آن‌ها آشنا می‌شوند.

## کارگاه ۲- پروژه Notepad

در این کارگاه نحوه ایجاد یک واسط کاربری استاندارد برای ذخیره پرونده تمرین شده، از ابزارهای منوی استاندارد و کنترل saveFileDialog برای ایجاد برنامه‌ای مشابه Notepad ویندوز استفاده شده است. چگونگی برنامه‌نویسی برای گزینه‌های save، save as و new از اهداف اصلی این کارگاه است. این گزینه‌ها در اغلب برنامه‌های ویندوزی کاربرد دارند. رعایت جزئیات پیاده‌سازی نظیر استفاده از فیلتر

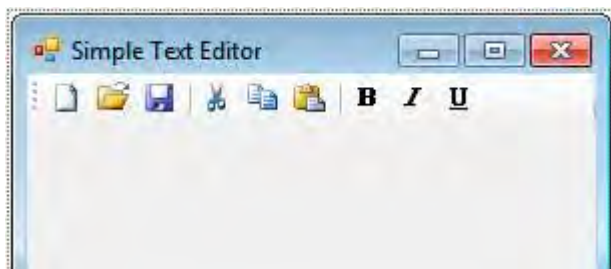
مناسب برای پنجره محاوره‌ای save و تطبیق منوی برنامه با منوی برنامه Notepad از نکاتی است که در این کارگاه باید مورد توجه قرار گیرد.

مسیر ذخیره پرونده در یک متغیر رشته‌ای به نام documentFilePath قرار می‌گیرد. در ابتدای اجرای برنامه، مقدار این رشته تهی است. تهی بودن این رشته نشان دهنده این است که هنوز پرونده ذخیره نشده است. با هر بار زدن گزینه Save As محاوره‌ای saveFileDialog نشان داده شده و رشته مسیر پرونده از نو مقداردهی می‌شود. با زدن گزینه save در صورتی که رشته مسیر تهی باشد مانند روال گزینه save as عمل می‌شود و رشته مسیر مقدار می‌گیرد. در دفعات بعدی استفاده از گزینه save، پرونده متنی در مسیر مشخص شده با رشته مسیر documentFilePath ذخیره می‌شود. گزینه new محتوای کادر متنی سند را پاک کرده، رشته مسیر documentFilePath را نیز با رشته تهی "" مقداردهی می‌کند.

برای توسعه این کارگاه می‌توانید از هنجریان بخواهید که به جای استفاده از کنترل کادرممتنی ساده از یک کنترل کادرممتنی غنی RichTextBox استفاده کنند. کنترل RichTextBox برای نمایش دادن متن‌های قالب بندی شده در قالب SaveFile (RTF) Rich Text Format به کار می‌رود. این کنترل متدی با نام SaveFile دارد که به سادگی می‌توان از آن برای ذخیره پرونده استفاده کرد. به عنوان مثال در رخدادهای کلیک گزینه save as می‌توان به شکل زیر پرونده را با پسوند doc یا rtf ذخیره کرد.

```
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
    // پرونده مسیر روزرسانی به
    documentFilePath = saveFileDialog1.FileName;
    // پرونده در متن نوشتن
    richTextBox1.SaveFile(documentFilePath);
}
```

می‌توان با اضافه کردن یک نوار ابزار toolStrip به برنامه، امکانات قالب‌بندی متن نظیر تغییر رنگ، تغییر قلم و ضخیم یا مورب کردن متن انتخاب شده را به برنامه افزود(شکل ۱).



شکل ۱- اضافه کردن نوار ابزار toolStrip

فعالیت کارگاهی  
ص ۲۲۴

برنامه‌ای بنویسید که قابلیت ذخیره مخاطبان در دفترچه تلفن را داشته باشد.



شکل ۲- فرم دفترچه تلفن

پاسخ: هدف از این فعالیت کارگاهی تمرین کار با متد `AppendAllText` برای افزودن متن به انتهای پرونده است. با زدن دکمه "افزافه به فهرست" محتویات کادرهای متنی همزمان به انتهای کادر فهرست و پرونده متنی اضافه می‌شوند.

```
string newContact = textBoxName.Text + "\t" +
                    textBoxLastName.Text + "\t" +
                    textBoxPhone.Text + "\t" +
                    textBoxAddress.Text;

File.AppendAllText("ContactList.txt", newContact + "\r\n");
listBoxContact.Items.Add(newContact);
```

برای توسعه این فعالیت کارگاهی می‌توانید از هنرجوها بخواهید که رکورد داده‌های افراد را در پرونده `ContactList.xml` با قالب `xml` به شکل زیر ذخیره کنند.

```
<person>
  <name>رضا</name>
  <surname>فرهادی</surname>
  <tel>۴۳۵۰۹۰۷۰</tel>
  <address>تبریز</address>
</person>
```

برای این منظور یک متغیر رشته‌ای با نام `newRecord` برای ذخیره رکورد جدید تعریف کرده، آن را مطابق قالب `xml` با داده‌های ورودی مقارنه کنید. از رشته‌های `"\r\n"` و `"\t"` برای قالب‌بندی متن استفاده کنید.

```
string newRecord = "<person>\r\n" +
```

```

        "\t<name>" + textBoxName.Text + "</name>\r\n"
+
        "\t<surname>" + textBoxLastName.Text +
"</surname>\r\n" +
        "\t<tel>" + textBoxPhone.Text + "</tel>\r\n"
+
        "\t<address>" + textBoxAddress.Text +
"</address>\r\n" +
        "</person>";

File.AppendAllText("ContactList.xml", newRecord + "\r\n");

```

XML مخفف زبان نشانه‌گذاری قابل گسترش (eXtensible Markup Language) است. XML کاربردهای فراوانی دارد. با XML می‌توان داده‌ها را بین دو سیستم ناهمسان مبادله کرد، فایل‌های متن ساده را برای به اشتراک گذاشتن داده‌ها به کار برد، فایل‌های متن ساده را برای ذخیره داده‌ها به کار گرفت و داده‌ها را در دسترس کاربران بیشتری قرار داد. تشخیص رکوردها و فیلدها و ذخیره و بازیابی رکوردها در قالب پرونده متنی مقدمه‌ای برای ورود به بحث ارتباط با پایگاه داده است.

فعالیت منزل  
ص ۲۲۴

بازی سرعت تایپ یا اعصاب‌سنج در فصل ۴ را توسعه دهید تا پس از پایان بازی نام‌کاربر و امتیازش را در یک پرونده متنی ذخیره کند.

پاسخ: هنجاریان می‌توانند با استفاده از متد AppendAllText نام‌کاربر و امتیازش را در یک پرونده متنی به صورت خط به خط اضافه کنند. توصیه می‌شود برای ذخیره داده‌ها از قالب xml استفاده شود.

```

string newRecord = "<player>\r\n" +
        "\t<name>" + userName + "</name>\r\n" +
        "\t<score>" + score + "</score>\r\n" +
        "</player>";

File.AppendAllText("score.xml", newRecord + "\r\n");

```

### کارگاه ۳ – خواندن پرونده متنی با متد ReadAllText

در تکمیل پروژه Notepad هنجاریان در این کارگاه عملکرد گزینه Open را فعال می‌کنند.

تکمیل کارگاه  
ص ۲۲۶

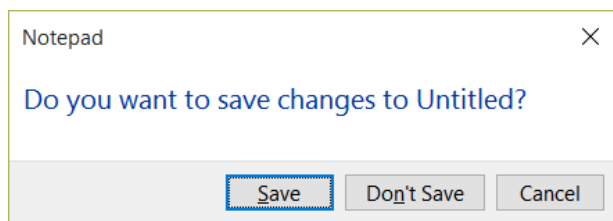
عملکرد رویداد کلیک گزینه Open چیست؟

پاسخ: در رخداد کلیک گزینه Open از منوی برنامه، ابتدا با فراخوانی متد ShowDialog از کنترل OpenFileDialog پنجره باز کردن پرونده نمایش داده می‌شود. در صورتی که کاربر پرونده‌ای را برای باز کردن انتخاب کند و دکمه

Open را بزند، آدرس پرونده در متغیر سراسری `documentFilePath` ذخیره شده، محتوای پرونده با متد `ReadAllText` خوانده و در کادر متنی نمایش داده می‌شود.

- کد مرحله ۳ را تغییر دهید تا قبل از باز کردن پرونده جدید، از کاربر برای ذخیره پرونده باز شده قبلی، سؤال کند.

در برنامه `notepad` ویندوز اگر آخرین تغییرات متنی ذخیره نشده باشد، هنگام خروج از برنامه یا باز کردن پرونده جدید، پیامی مبنی بر عدم ذخیره تغییرات به کاربر نشان داده می‌شود (شکل ۳).



شکل ۳- پیام ذخیره تغییرات

در این حالت به کاربر فرصت داده می‌شود که قبل از خروج، تغییرات انجام شده را ذخیره کند. می‌خواهیم این قابلیت را به پروژه `notepad` اضافه کنیم.

(۱) در پروژه `notepad` یک متغیر از نوع `bool` با نام `saveStatus` و با مقدار اولیه `true` به کلاس فرم اصلی اضافه کنید. وظیفه این متغیر یا پرچم، نشان دادن وضعیت ذخیره شدن محتوای کادر متنی است. اگر مقدار `saveStatus` برابر با `true` باشد یعنی محتوا قبلاً ذخیره شده است و برنامه می‌تواند بسته شود. در غیر اینصورت باید از بسته شدن پرونده جلوگیری شود و منتظر تأیید کاربر برای خروج بماند.

(۲) هر بار که نوشته کادر متنی تغییر می‌کند مقدار متغیر `saveStatus` باید `false` شود. بنابراین در رخداد `TextChanged` کادر متنی، `saveStatus` را `false` می‌کنیم.

```
private void mainDocument_TextChanged(object sender, EventArgs e)
{
    saveStatus = false;
}
```

(۳) در متدهای `save` و `save as`، پس از نوشتن موفقیت آمیز داده‌ها در پرونده، مقدار `saveStatus` را دوباره `true` می‌کنیم.

```
private void SaveAsAction()
{
```



```

        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            documentFilePath = saveFileDialog.FileName;
            File.WriteAllText(documentFilePath, mainDocument.Text);
            saveStatus = true;
        }
    }
    private void SaveAction()
    {
        if (documentFilePath == "")
        {
            SaveAsAction();
        }
        else
        {
            File.WriteAllText(documentFilePath, mainDocument.Text);
            saveStatus = true;
        }
    }
}

```

۴) رخداد کلیک گزینه Open را به صورت زیر تغییر می‌دهیم.

```

private void openToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (!saveStatus)
    {
        DialogResult result = MessageBox.Show(this,
            "Do you want to save changes?", "Notepad",
            MessageBoxButtons.YesNoCancel);
        if (result == DialogResult.Yes)
        {
            SaveAction();
        }
        if (result == DialogResult.Cancel)
        {
            return;
        }
    }
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        documentFilePath = openFileDialog.FileName;
        mainDocument.Text = File.ReadAllText(documentFilePath);
        saveStatus = true;
    }
}

```

۵) در رخداد FormClosing از فرم اصلی وضعیت ذخیره شدن متن را بررسی می‌کنیم.

۶) برای جلوگیری از بسته شدن فرم، در رخداد FormClosing مقدار e.Cancel را true کنید.

```
private void mainForm_FormClosing(object sender,
FormClosingEventArgs e)
{
    if(!saveStatus)
    {
        DialogResult result =
            MessageBox.Show(this, "Do you want to
            save changes?",
                                "Notepad", MessageBoxButtons.YesNoCancel);

        switch(result)
        {
            case DialogResult.Yes:
                SaveAction();
                e.Cancel = !saveStatus;
                break;
            case DialogResult.Cancel:
                e.Cancel=true;
                break;
        }
    }
}
```

### فعالیت منزل ص ۱۷۵

با تنظیم خاصیت راست به چپ فرم و ترجمه گزینه‌های منو به زبان فارسی، برنامه Notepad را فارسی کنید.

پاسخ: در این تمرین هنجاریان باید به معادل‌های متعارف زبان فارسی برای واژه‌های استفاده شده در منوی برنامه توجه کنند. برای این منظور می‌توان از واژه‌های مصوب فرهنگستان زبان و ادب پارسی استفاده کرد.

معادل فارسی	واژه	معادل فارسی	واژه
برش	Cut	سند نو، سند تازه، جدید	New
رونوشت	Copy	بازکردن سند	Open
چسباندن	Paste	حفظ سند، ذخیره سند	Save
قلم	Font	حفظ با نام، ذخیره در	Save as

- برای گزینه‌های منوی Edit برنامه Notepad کدنویسی کنید.

با استفاده از متدهای Cut، SelectAll، Copy و Paste از کادر متنی، به ترتیب قابلیت گزینه‌های Cut، Copy، Paste در منوی Edit را می‌توان به برنامه اضافه کرد.

```
private void copyToolStripMenuItem_Click(object sender,
EventArgs e)
{
```

```

    mainDocument.Copy();
}

private void pasteToolStripMenuItem_Click(object sender,
EventArgs e)
{
    mainDocument.Paste();
}

private void cutToolStripMenuItem_Click(object sender,
EventArgs e)
{
    mainDocument.Cut();
}

private void selectAllToolStripMenuItem_Click(object sender,
EventArgs e)
{
    mainDocument.SelectAll();
}

```

## — کارگاه ۴ – خواندن پرونده متنی با متد ReadAllLines

در این کارگاه عملکرد متدهای ReadAllLines و ReadAllText در خواندن داده‌های پرونده با هم مقایسه می‌شود. همچنین برای اطمینان از وجود پرونده در مسیر مشخص شده متد File.Exists معرفی می‌شود.

تکمیل کارگاه  
ص ۲۲۷

س: برنامه را اجرا کنید. چه اطلاعاتی را در فهرست مشاهده می‌کنید؟  
 ج: اطلاعات تماس ذخیره شده در پرونده ContactList.txt  
 س: رشته fileContent حاوی چه مقادیری است؟  
 ج: محتویات پرونده ContactList.txt  
 س: عملکرد متد ReadAllLines چیست؟  
 ج: خواندن خطوط پرونده متنی و ذخیره آن‌ها در یک آرایه از رشته‌ها.  
 س: عملکرد متد AddRange چیست؟  
 ج: تمام محتویات یک آرایه از رشته‌ها را خط به خط به کادر فهرست اضافه می‌کند.  
 س: برنامه را اجرا کنید. تفاوت متد ReadAllLines با متد ReadAllText در چیست؟  
 ج: متد ReadAllLines تمام محتویات پرونده متنی را خط به خط خوانده، این خطوط را به ترتیب در آرایه‌ای از رشته‌ها قرار می‌دهد، اما متد ReadAllText تمام محتویات پرونده را در یک متغیر رشته‌ای برمی‌گرداند.

از متد `ReadAllLines` در جایی استفاده می‌شود که بخواهیم به خطوط پرونده متنی به طور مستقیم دسترسی پیدا کرده، پرونده را خط به خط ویرایش کنیم. به عنوان مثال اگر بخواهید خطوط پنجم و هفتم یک پرونده متنی را با هم جابجا کنید کافی است که متد `ReadAllLines` را فراخوانی کرده، در آرایه خروجی جای عنصر پنجم و هفتم را جابجا کنید، سپس با فراخوانی متد `WriteAllLines` می‌توان محتویات این آرایه را در پرونده بازنویسی کرد.

```
string[] str = File.ReadAllLines(fileName);
string temp = str[4];
str[4] = str[6];
str[6] = temp;
File.WriteAllLines(fileName, str);
```

س: پرونده `ContactList.txt` را از پوشه `Debug` برنامه حذف کنید. برنامه را اجرا کنید. چه خطایی رخ می‌دهد؟ چرا؟

ج: یک خطای استثنای مدیریت نشده (`FileNotFoundException`) رخ می‌دهد و نشان می‌دهد که در مسیر مشخص شده پرونده `ContactList.txt` وجود نداشته است.

س: نوع خروجی متد `Exists` چیست؟

ج: خروجی این متد از نوع `bool` است. اگر پرونده در مسیر `path` وجود داشته باشد مقدار `true` و در غیر اینصورت مقدار `false` برگردانده می‌شود.

```
public static bool Exists(string path);
```

## کارگاه ۵ – مدیریت خطا با دستور try-catch

برای اداره کردن استثناها در `C#` از کلاس `Exception` استفاده می‌شود. این کلاس، کلاس پایه تمام استثناها است و استثناهای دیگر از این کلاس مشتق می‌شوند. کلاس `Exception` ویژگی‌ها و متدهای مختلفی دارد که می‌توان در کتابخانه `MSDN` آن‌ها را مشاهده کرد. یکی از پرکاربردترین ویژگی‌های این کلاس، ویژگی `Message` است. این ویژگی یک پیام متنی است و توضیحی در مورد استثنایی که رخ داده، در اختیار برنامه‌نویس قرار می‌دهد. علاوه بر استثناهای تعریف شده در کتابخانه `.Net`، برنامه‌نویسان می‌توانند با استفاده از کلمه کلیدی `throw` در زبان `C#`، استثناهای خاص خود را تولید کنند.

```
try
{
    throw new Exception("استثنای تولید شده توسط برنامه نویس");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "خطا");
}
```

```
}
```

در حالتی که ممکن است در بخش try بیش از یک حالت استثنا ایجاد شود، برنامه‌نویس می‌تواند حالت‌های استثنایی مختلف را یک به یک و به صورت مجزا در بلوک‌های catch متفاوت مدیریت کند. ساختار کلی دستور try-catch برای مدیریت استثنای چندگانه به صورت زیر است:

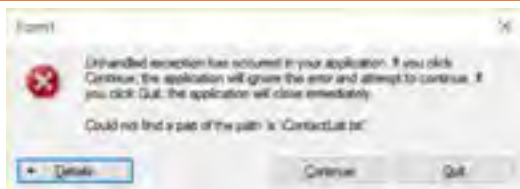
```
try
{
    // قسمتی از کد که ممکن است استثنا ایجاد کند
}
catch (ExceptionName1 e1)
{
    // کدهای مدیریت خطا برای استثنای e1
}
catch (ExceptionName2 e2)
{
    // کدهای مدیریت خطا برای استثنای e2
}
catch (ExceptionNameN eN)
{
    // کدهای مدیریت خطا برای استثنای eN
}
finally
{
    // دستوراتی که در نهایت اجرا می‌شوند
}
```

اگر استثنایی رخ دهد کنترل برنامه به قسمت catch مربوط به آن استثنا منتقل می‌شود. در هر صورت اگر استثنایی رخ دهد و یا رخ ندهد در انتها بلوک دستورات finally اجرا خواهد شد. به عنوان مثال اگر شما پرونده‌ای را برای خواندن یا نوشتن باز کرده‌اید، در هر صورت چه استثنایی رخ دهد چه رخ ندهد باید پرونده را در انتهای کار ببندید.

س: برنامه را اجرا کنید. چه خطایی مشاهده می‌کنید؟ با کمک هنرآموز خود پیام نشان داده شده در پنجره خطا را به فارسی معنی کنید.

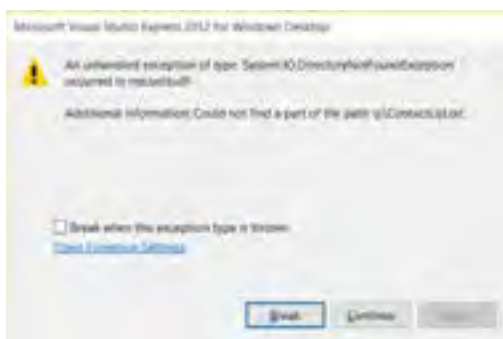
ج: اگر برنامه را با Ctrl+F5 و بدون دیباگ کردن اجرا کنید، پنجره خطا به شکل زیر نمایش داده می‌شود. این حالت مانند وقتی است که با دوبار کلیک روی پرونده exe برنامه را اجرا می‌کنید.

تکمیل گارگاه  
ص ۲۲۹



شکل ۴- پنجره خطا بدون دیباگ

یک استثنای مدیریت نشده در برنامه رخ داده است. اگر بر روی دکمه Continue کلیک کنید، برنامه از این خطا چشم پوشی کرده، سعی خواهد کرد که اجرای خود را ادامه دهد. اگر بر روی Quite کلیک کنید، برنامه بلافاصله بسته خواهد شد. بخشی از مسیر "a:\ContactList.txt" یافت نشد. اگر برنامه را با F5 و در حالت دیباگ اجرا کنید، پنجره خطا به شکل زیر خواهد بود (شکل ۵).



شکل ۵- پنجره خطا در حالت دیباگ

س: برنامه را اجرا کنید. اجرای برنامه در این حالت چه تفاوتی با حالت قبل دارد؟  
ج: به جای پیام خطای قبلی، پیام زیر مشاهده می شود (شکل ۶).



شکل ۶- پنجره پیام

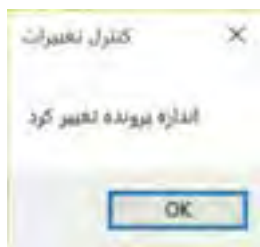
## کارگاه ۶ - اطلاع از تغییرات پرونده

این روزها با گسترش زیرساخت‌های ارتباطی و افزایش میزان دسترسی مردم به اینترنت، استفاده از نرم‌افزارهای پیام‌رسان به شکل وسیعی مورد توجه قرار گرفته است. بسیاری از افراد جامعه برای برقراری ارتباط با دیگران و ارسال و دریافت پیام، از نرم‌افزارهای پیام‌رسان روی گوشی‌های هوشمند و یا رایانه‌های شخصی خود استفاده می‌کنند. در کارگاه ۶ هنرجویان با کنترل FileSystemWatcher آشنا می‌شوند و سپس با استفاده از این ابزار در کارگاه ۷ یک تالار گفتگوی مجازی ایجاد می‌کنند.

از کنترل FileSystemWatcher برای مشاهده تغییرات در یک پوشه مشخص استفاده می‌شود. می‌توان تغییرات در پرونده‌ها و زیر شاخه‌های پوشه مشخص شده را زیر نظر گرفت. از این ابزار برای زیرنظر گرفتن پرونده‌های یک کامپیوتر محلی، درایو شبکه یا یک کامپیوتر از راه دور استفاده می‌شود.

برای زیرنظر گرفتن تغییرات تمام پرونده‌ها، ویژگی Filter را به یک رشته خالی ("") تنظیم کنید. همچنین می‌توانید از نشانه‌های عام (wildcards) مانند ("\*") نیز استفاده کنید. برای زیرنظر گرفتن یک پرونده خاص، ویژگی Filter را به نام فایل تنظیم کنید. به عنوان مثال، برای زیرنظر گرفتن تغییرات در پرونده MyDoc.txt، ویژگی Filter را به "MyDoc.txt" تنظیم کنید. همچنین می‌توانید تغییرات در یک نوع خاص از پرونده را زیرنظر بگیرید. برای مثال، برای تحت نظر گرفتن تغییرات در پرونده‌های متنی، ویژگی فیلتر را به "\*.txt" تنظیم کنید.

انواع تغییرات را می‌توان در یک پوشه یا پرونده زیرنظر گرفت. به عنوان مثال، می‌توانید تغییر تاریخ و زمان انجام آخرین عمل نوشتن در پرونده یا تغییر اندازه پوشه یا پرونده را زیر نظر بگیرید. این کار با تنظیم ویژگی NotifyFilter به یکی از مقادیر NotifyFilters انجام می‌شود. برای اطلاعات بیشتر در مورد نوع تغییراتی که می‌توانید تحت نظر بگیرید، FileSystemWatcher و NotifyFilters را در کتابخانه MSDN جستجو کنید.



شکل ۷

س: برنامه را اجرا کنید. روی دکمه کلیک کنید و در حالی که برنامه در حال اجرا است، پرونده MyFile.txt را در برنامه notepad باز کنید. نام خود را در پرونده بنویسید و آن را ذخیره کنید. برنامه شما چه واکنشی به این تغییر نشان می‌دهد؟

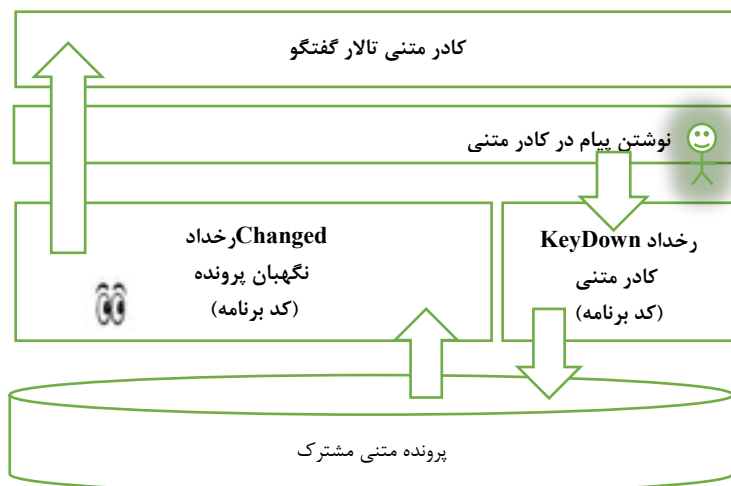
ج: پیامی مبنی بر تغییر اندازه پرونده مشاهده می‌شود (شکل ۷).

تکمیل کارگاه  
ص ۲۳۰

به نظر شما این برنامه چه کاربردی می‌تواند داشته باشد؟ آیا به کمک این برنامه می‌توان از ویروس‌ها، شدن پرونده‌ها مطلع شد؟ پاسخ: یکی از کاربردهای این ابزار می‌تواند تشخیص تغییر پرونده در زمان ویرایش باشد. به عنوان مثال اگر شما در ویرایشگر کد برنامه ویژوال استودیو یک پرونده را باز کرده باشید و همزمان محتویات این پرونده را در یک ویرایشگر دیگر مانند Notepad تغییر دهید، هنگام بازگشت به پنجره ویرایشگر کد در ویژوال استودیو، به شما پیامی مبنی بر تشخیص تغییر محتویات پرونده نشان داده می‌شود. می‌توان با زیر نظر گرفتن محتویات یک پوشه از نظر تغییر در اندازه پرونده‌ها و یا تغییر در زمان آخرین نوشتن در پرونده از تغییر محتویات پرونده‌ها به وسیله برخی ویروس‌ها مطلع شد.

## کارگاه ۷ - نوشتن و خواندن از یک پرونده متنی مشترک

در این کارگاه هنرجویان با نوشتن در یک پرونده متنی مشترک و خواندن از این پرونده یک تالار گفتگوی مجازی ایجاد می‌کنند. کارهای اصلی برنامه در دو رخداد KeyDown کادر متنی و رخداد Changed نگهبان پرونده انجام می‌شود. عملکرد رخداد KeyDown کادر متنی: اضافه کردن پیام نوشته شده در کادر متنی به انتهای پرونده متنی  
عملکرد رخداد Changed نگهبان پرونده: خواندن پرونده متنی و نوشتن آن در کادر متنی تالار گفتگو



شکل ۸- عملکرد برنامه تالار گفتگو



در عمل برای پیاده‌سازی برنامه‌های پیام‌رسانی مانند این برنامه، از روش‌های دیگری مانند وب سرویس‌ها (Web Services) یا برنامه‌نویسی سوکت (Socket Programming) استفاده می‌کنند.

س: عملکرد متد رویداد `FileSystemWatcher_Changed` چیست؟

ج: خواندن محتوای پرونده متنی و نوشتن آن در کادر متنی تالار گفتگو

عملکرد متد `ScrollToCaret` کنترل کادرمتن چیست؟

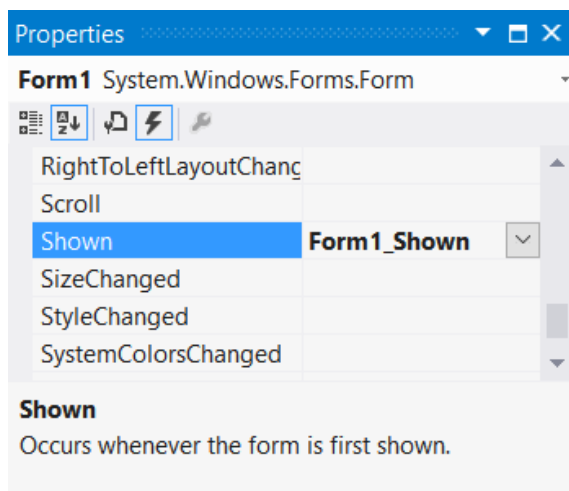
این متد محتوای کنترل را تا جایی که نشانگر درج متن در دیدرس قرار گیرد، اسکرول می‌کند. از این متد می‌توان در یک کادرمتنی چندخطی برای اطمینان حاصل کردن از اینکه متن ورودی فعلی در دیدرس قرار دارد، استفاده کرد.

س: برنامه را اجرا کنید. در پوشه `Debug` برنامه `PublicChat.exe` را چند بار اجرا کنید و در هر یک از برنامه‌ها پیام‌های دلخواهی بنویسید. چه چیزی مشاهده می‌کنید؟

ج: متن ارسال شده در تمام نسخه‌های در حال اجرای برنامه مشاهده می‌شود.

س: با توجه به راهنمای رویدادها در `VS` عملکرد رویداد `Form1_Shown` چیست؟

این رویداد وقتی اتفاق می‌افتد که فرم برای اولین بار نمایش داده می‌شود.



شکل ۹- رویداد `shown`

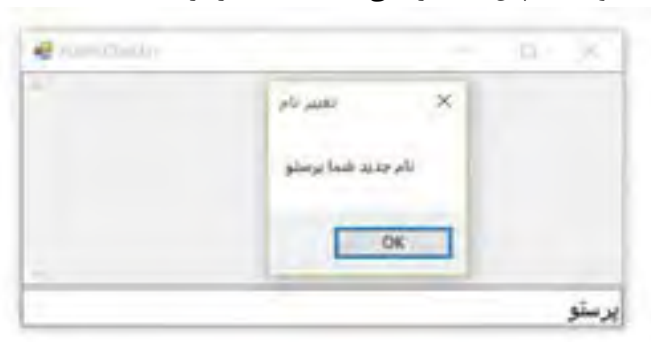
س: در کد رویداد `txtPost_KeyDown` امکان تغییر نام کاربر را با کلیدهای

Ctrl+Enter فراهم کنید.

ج: رخداد KeyDown کادر متنی txtPost را به شکل زیر تغییر می‌دهیم.

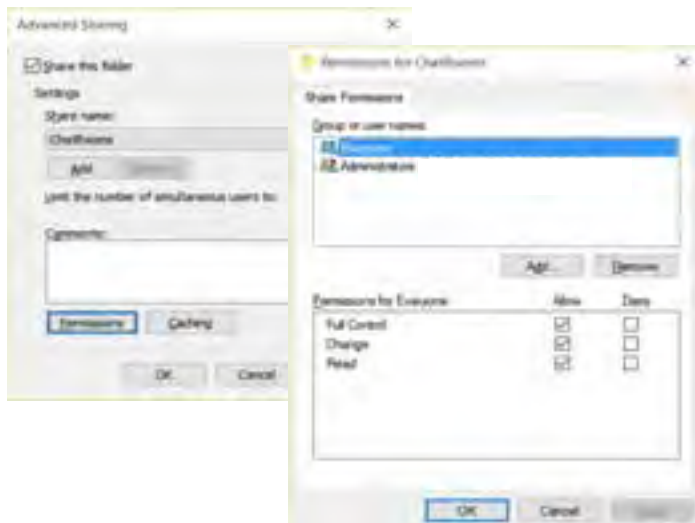
```
private void txtPost_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        if (e.Control)
        {
            myName = txtPost.Text;
            MessageBox.Show("شما جدید نام" + " " + myName, "تغییر نام");
            txtPost.Clear();
        }
        else
        {
            try
            {
                File.AppendAllText(sharedDir + roomFile,
                    DateTime.Now.ToString("HH:mm:ss") + " " +
                    myName + " : " + txtPost.Text + "\r\n");
                txtPost.Clear();
            }
            catch
            {
                MessageBox.Show("نشد ارسال پیام", "خطا");
            }
        }
    }
}
```

به این ترتیب کاربر می‌تواند برای تغییر نام خود، ابتدا اسمی را در کادر متنی txtPost نوشته، سپس کلید ترکیبی Ctrl+Enter را بزند.



شکل ۱۰- کادر پیام تایید تغییر نام کاربر

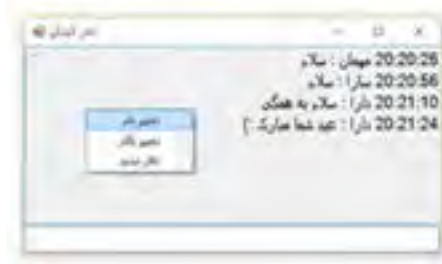
برنامه تالار گفتگو را می‌توان در شبکه اجرا کرد و هنجرویان می‌توانند از طریق شبکه با یکدیگر گفتگو کنند. هنرآموز محترم می‌تواند پوشه‌ای دلخواه را در شبکه به اشتراک گذاشته و به کاربران دسترسی کامل بدهد.



مسیر پوشه اشتراکی در شبکه در متغیر sharedDir قرار می‌گیرد.  
`string sharedDir = @"\\TEACHER-PC\ChatRooms\";`

توسعه پروژه اتاق گفتگو با افزودن منو به برنامه پاسخ: هنرآموز می‌تواند رخداد کلیک گزینه تغییر تالار را خود در کلاس به طور کامل انجام داده، مراحل کار را برای هنجرویان شرح دهد و در ادامه کدنویسی رخداد کلیک گزینه تالار جدید را به عنوان تکلیف منزل به هنجرویان واگذار کند.

فعالیت منزل  
 ص ۲۳۴



شکل ۱۱- منوی زمینه‌ای تالار گفتگو

یک کنترل `contextMenuStrip` شامل گزینه‌های تغییر نام، تغییر تالار و تالار جدید به برنامه اضافه کرده، متد رویداد کلیک آن‌ها را بنویسید.

رخداد کلیک گزینه تغییر نام:

با استفاده از کادر متنی `txtPost` نام جدید کاربر دریافت می‌شود.

```
private void ToolStripMenuItemChangeName_Click(object sender,
EventArgs e)
{
    myName = txtPost.Text;
    MessageBox.Show("نام جدید شما" + " " + myName, "تغییر نام");
    txtPost.Clear();
}
```

رخداد کلیک گزینه تغییر تالار:

یک کنترل `openFileDialog` به برنامه اضافه کرده، در رخداد کلیک گزینه تغییر تالار، کادر محاوره‌ای باز کردن پرونده را به کاربر نشان می‌دهیم. در صورتی که کاربر پرونده‌ای را انتخاب کند، در مسیر قبلی پرونده اتاق، پیام خروج کاربر نوشته می‌شود و سپس متغیرهای `roomFile` و `sharedDir` به ترتیب با نام پرونده انتخاب شده و مسیر پوشه پرونده انتخاب شده، مقداردهی می‌شوند. در انتها در مسیر جدید پرونده اتاق، پیام ورود کاربر نوشته می‌شود.

```
private void ToolStripMenuItemChangeRoom_Click(object sender,
EventArgs e)
{
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        try
        {
            File.AppendAllText(sharedDir + roomFile,
                DateTime.Now.ToString("HH:mm:ss") + " " +
myName + " خارج شد " + "\r\n");
        }
        catch
        {
            MessageBox.Show("خطا", "پیام ارسال نشد");
        }
        roomFile = Path.GetFileName(openFileDialog.FileName);
        sharedDir =
Path.GetDirectoryName(openFileDialog.FileName) + @"\\";
        this.Text = roomFile;
        fileSystemWatcher.Filter = roomFile;
        fileSystemWatcher.Path = sharedDir;
        try
        {
            File.AppendAllText(sharedDir + roomFile,
                DateTime.Now.ToString("HH:mm:ss") + " " +
myName + " وارد شد " + "\r\n");
        }
    }
}
```

```

        catch
        {
            MessageBox.Show("خطا", "پیام ارسال نشد");
        }
    }
}

```

رخداد کلیک گزینه تالار جدید:

یک کنترل `saveFileDialog` به برنامه اضافه کرده و در رخداد کلیک گزینه تالار جدید، کادر محاوره‌ای ذخیره کردن پرونده را به کاربر نشان می‌دهیم. در صورتی که کاربر پرونده‌ای را انتخاب کند، در مسیر قبلی پرونده اتاق، پیام خروج کاربر نوشته می‌شود و سپس متغیرهای `roomFile` و `sharedDir` به ترتیب با نام پرونده انتخاب شده و مسیر پوشه پرونده انتخاب شده، مقداردهی می‌شوند. در انتها در مسیر جدید پرونده اتاق، پیام ایجاد اتاق نوشته شده و پرونده ایجاد می‌شود.

```

private void ToolStripMenuItemNewRoom_Click(object sender,
EventArgs e)
{
    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        try
        {
            File.AppendAllText(sharedDir + roomFile,
                DateTime.Now.ToString("HH:mm:ss") + " " +
myName + " خارج شد " + "\r\n");
        }
        catch
        {
            MessageBox.Show("خطا", "پیام ارسال نشد");
        }
        roomFile = Path.GetFileName(saveFileDialog.FileName);
        sharedDir =
Path.GetDirectoryName(saveFileDialog.FileName) + @"\";
        this.Text = roomFile;
        fileSystemWatcher.Filter = roomFile;
        fileSystemWatcher.Path = sharedDir;
        File.WriteAllText(sharedDir + roomFile, "اتاق ایجاد شد" +
"\r\n");
    }
}

```

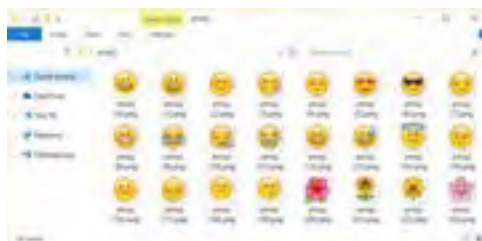
س: چگونه می‌توان قابلیت نمایش استیکر و ایموجی (Emoji) نمادها را به برنامه افزود؟ چه نوع پرونده‌ای برای نمایش این شکل‌ها در کنار متن مناسب است؟  
 ج: برای این منظور می‌توان به جای کنترل کادر متنی از یک کنترل مرورگر وب (`webBrowser`) و به جای پرونده متنی ساده از پرونده `html` برای نمایش پیام‌ها استفاده کرد. به این ترتیب کافی است که پیام‌ها را در قالب کدهای `html` در پرونده متنی مشترک با پسوند `htm` نوشته، هر بار که محتوای این پرونده

رخداد Form\_Load بر نامه به شکل زیر خواهد بود.

برای نوشتن پیام در قالب `html`، در کد رویداد `txtPost_KeyDown` می‌توان دستور زیر را جایگزین دستور `AppendAllText` قبلی کرد. هر جای دیگر نیز که متنی در پرونده نوشته می‌شود مطابق این الگو عمل شود.

کد رویداد `fileSystemWatcher_Changed` نیز به شکل زیر ساده خواهد شد.

در داخل پوشه‌ای که پرونده html اشتراکی قرار دارد یک پوشه با نام emoji ایجاد کرده، تمام تصویرهایی که می‌خواهید به عنوان استیکر از آن‌ها استفاده کنید را در این پوشه قرار دهید. پرونده‌های تصویری را به صورت emoji.png (x) نامگذاری کنید که در آن x شماره پرونده تصویری است و از صفر شروع می‌شود.



---

برچسب‌ها را انتخاب و برای رخداد کلیک آن‌ها این متد را انتخاب کنید.

```
private void emojiLabel_Click(object sender, EventArgs e)
{
    try
    {
        string stickerPath = "emoji\\emoji (" +
            ((Label)sender).ImageIndex.ToString() + ").png";

        File.AppendAllText(sharedDir + roomFile,
            "<h2 dir=\"rtl\" align=\"right\">" +
            DateTime.Now.ToString("HH:mm:ss") + " " +
            myName + " : " + txtPost.Text + "<br><br><br>\r\n"
+
            "<img align=\"right\" src=\"" + stickerPath +
            "\" height=\"50\"
width=\"50\"><br><br><br></h2>\r\n");
        txtPost.Clear();
    }
    catch
    {
        MessageBox.Show("نشد ارسال پیام", "خطا");
    }
}
```



به جای رخداد Form1\_Shown، رخداد DocumentCompleted از کنترل webBrowser را به صورت زیر اضافه کنید.

```
private void webBrowser1_DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e)
{
    webBrowser1.Document.Window.ScrollTo(0,
webBrowser1.Document.Body.ScrollRectangle.Height);
}
```

محتویات پرونده html به شکل زیر خواهد بود.

```
<meta charset="UTF-8">
<h2 dir="rtl" align="right">خوش آمدید</h2>
<h2 dir="rtl" align="right">سلام : مهمان 23:01:46</h2>
<h2 dir="rtl" align="right">مهمان 23:01:48 : <br><br><br>
<br><br><br></h2>
```

## مشاهده فهرست پرونده‌ها و پوشه‌ها

بسیاری از اوقات کار با پرونده‌ها در یک برنامه مستلزم انجام عملیاتی نظیر ایجاد، حذف، ایجاد نسخه مشابه، جابه‌جایی و برخی کارهای دیگر روی پوشه‌ها و یا استخراج پسوند یا نام پرونده از یک مسیر است. در این بخش با معرفی کلاس‌های Directory و Path و آشنایی با برخی از متدهای پرکاربرد آن‌ها هنرجویان آماده می‌شوند که برنامه‌ای شبیه به مرورگر پرونده (FileExplorer) ویندوز بنویسند.

فعالیت کارگاهی  
ص ۲۳۵

با استفاده از MSDN جدول متدهای کلاس Directory را تکمیل کنید.

نام متد	نوع ورودی	نوع خروجی	عملکرد
GetLogicalDrives	ندارد	آرایه رشته‌ای	نام همه درایوها را در یک آرایه برمی‌گرداند.
GetDirectories	رشته (مسیر)	آرایه رشته‌ای	نام و مسیر تمام زیرپوشه‌های مسیر ورودی را در یک آرایه برمی‌گرداند.
GetFiles	رشته (مسیر)	آرایه رشته‌ای	نام و مسیر همه پرونده‌های مسیر ورودی را در یک آرایه برمی‌گرداند.
Exists	رشته (مسیر)	bool	تشخیص می‌دهد که آیا مسیر ورودی به پوشه‌ای که روی دیسک وجود دارد اشاره می‌کند یا خیر.

باید توجه شود که برخی متدها مانند متد GetDirectories و متد GetFiles دارای چند ریخت مختلف با تعداد ورودی‌ها و یا نوع خروجی متفاوت هستند. در زبان‌های برنامه‌نویسی شیء‌گرا مانند C# می‌توان دو و یا بیشتر از دو متد هم نام را با پارامترهای ورودی یا خروجی متفاوت در یک کلاس تعریف کرد. به این کار Overload کردن متدها گفته می‌شود. Overloading یکی از روش‌های پشتیبانی از چند ریختی (polymorphism) در زبان‌های برنامه‌نویسی است.

از دیگر روش‌های پشتیبانی از چندریختی در زبان‌های برنامه‌نویسی شیء‌گرا، Override کردن متدها است. Overriding قابلیت است که در آن یک کلاس فرزند می‌تواند پیاده‌سازی خاص خود را از متدهای کلاس پدر داشته باشد.



با استفاده از MSDN جدول متدهای کلاس Path را تکمیل کنید.

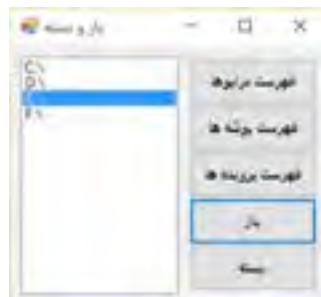
نام متد	نوع ورودی	نوع خروجی	عملکرد
GetFileName	یک رشته (مسیر)	یک رشته	نام و پسوند پرونده مسیر ورودی را برمی گرداند.
Combine	دو رشته (مسیر) یا بیشتر	یک رشته (مسیر)	آرایه‌ای از رشته‌ها را با هم ترکیب کرده و از ترکیب آنها یک مسیر برمی گرداند.

## کارگاه ۸ – کاربرد کلاس Path و Directory

س: برنامه را اجرا کنید. عملکرد متد AddRange کنترل کادر فهرست چیست؟  
ج: یک آرایه از رشته‌ها را یکباره به کنترل کادر فهرست اضافه می‌کند.  
س: کد را تغییر دهید تا مسیر **directory** را از کادر متن دریافت کند.  
ج: `string directory = txtPath;`

### توسعه کارگاه ۸

به عنوان توسعه این کارگاه می‌توانید قابلیت باز و بسته کردن درایو CD یا DVD را به این برنامه اضافه کنید.



شکل ۱۲- فرم توسعه کارگاه ۸

در ابتدای برنامه فضای نام مناسب را به برنامه اضافه کنید.

`using System.Runtime.InteropServices;`

در داخل کلاس فرم کد زیر را اضافه کنید. این کد تابع `mciSendString` را به عنوان یک متد خارجی به کلاس فرم اضافه می‌کند. تابع `mciSendString` یکی از API‌های ویندوز برای کار با تجهیزات MCI است.

```
[DllImport("winmm.dll", EntryPoint = "mciSendStringA", CharSet =
CharSet.Ansi)]
protected static extern int mciSendString(string lpstrCommand,
StringBuilder lpstrReturnString,
int uReturnLength,
IntPtr hwndCallback);
```

در رخداد کلیک دکمه "باز" کد زیر را بنویسید. دستور `mciSendString` اول، درایو را مشخص می‌کند و دستور دوم فرمان باز یا بسته شدن را به درایو انتخاب شده ارسال می‌کند.

```
string driveLetter = listBox1.SelectedItem.ToString();
StringBuilder returnString = new StringBuilder();
mciSendString("open " + driveLetter + ": type CAudio alias
drive" + driveLetter,
returnString, 0, IntPtr.Zero);
mciSendString("set drive" + driveLetter + " door open",
returnString, 0, IntPtr.Zero);
```

در رخداد کلیک دکمه "بسته" کردن کد زیر را بنویسید.

```
string driveLetter = listBox1.SelectedItem.ToString();
StringBuilder returnString = new StringBuilder();
mciSendString("open " + driveLetter + ": type CAudio alias
drive" + driveLetter,
returnString, 0, IntPtr.Zero);
mciSendString("set drive" + driveLetter + " door closed",
returnString, 0, IntPtr.Zero);
```

## کارگاه ۹ – پروژه مرورگر پرونده (FileExplorer)

در این کارگاه سطح پیشرفته‌تری از کار با پوشه‌ها و پرونده‌ها تمرین می‌شود. انتخاب مسیر در این برنامه شبیه برنامه مرورگر پرونده در ویندوز صورت می‌گیرد.

تکمیل کارگاه  
ص ۲۴۰

س: حلقه `foreach` اول و دوم در مرحله ۵ کارگاه، هر کدام چه عناصری را به فهرست اضافه می‌کنند؟  
ج: حلقه `foreach` اول پوشه‌ها و حلقه دوم پرونده‌های موجود در مسیر را به کادر فهرست اضافه می‌کند.  
س: در مرحله ۶ از ساختار `try-catch` برای مدیریت حالت‌های استثنا استفاده کنید.

ج: دستورات `showDirectory` متد را در داخل قسمت `try` نوشته و در قسمت `catch` پیام مناسبی برای نشان دادن حالت استثنا بنویسید. در صورتی که نخواهید پیامی نشان داده شود، قسمت `catch` را خالی بگذارید.

فعالیت کارگاهی  
ص ۲۴۱

دکمه `btnUp` ، مسیر جاری را یک پوشه بالا می برد. کد زیر را برای رویداد کلیک این دکمه تکمیل کنید. در چه حالتی ممکن است استثنا رخ دهد؟ برای این حالت پیام مناسبی به کاربر نشان دهید.

ج: در این قسمت از متد `GetParent` استفاده شده است. برای تشخیص انواع استثناهایی که ممکن است هنگام استفاده از این متد در برنامه پیش آید به توضیحات متد در کتابخانه MSDN مراجعه می کنیم.

#### Exceptions

Exception	Condition
<code>IOException</code>	The directory specified by <code>path</code> is read-only.
<code>UnauthorizedAccessException</code>	The caller does not have the required permission.
<code>ArgumentException</code>	<code>path</code> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by <code>InvalidPathChars</code> .
<code>ArgumentNullException</code>	<code>path</code> is <b>null</b> .
<code>PathTooLongException</code>	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters and file names must be less than 260 characters.
<code>DirectoryNotFoundException</code>	The specified path was not found.

اگر پوشه مشخص شده به وسیله مسیر ورودی `path` فقط خواندنی باشد، استثنای `IOException` رخ می دهد.

اگر برنامه ای که این متد را فراخوانی کرده است، مجوز لازم برای دسترسی به پوشه را نداشته باشد، استثنای `UnauthorizedAccessException` رخ می دهد.

اگر طول رشته ورودی صفر باشد، رشته ورودی تنها شامل کاراکتر فاصله باشد یا دارای یک یا بیشتر کاراکتر غیر مجاز باشد در این موارد استثنای `ArgumentException` رخ می دهد.

اگر رشته ورودی null باشد، استثنای `ArgumentNullException` رخ می‌دهد.  
 اگر طول مسیر ورودی، نام پرونده یا هردو از حداکثر طول تعریف شده برای سیستم بیشتر باشد، استثنای `PathTooLongException` رخ می‌دهد.  
 اگر مسیر ورودی پیدا نشود، استثنای `DirectoryNotFoundException` رخ می‌دهد.  
 برای تمام حالت‌های استثنا می‌توان یک پیام کلی در نظر گرفت.

```
try
{
    string newPath =
Directory.GetParent(textBoxAddress.Text).FullName;
    showDirectory(newPath);
}
catch
{
    MessageBox.Show("خطا", "دسترسی به این پوشه امکان پذیر نیست");
}
```

در غیر این صورت برای تفکیک حالت‌های استثنا و نشان دادن پیام‌های جداگانه باید برای هر استثنا یک `catch` جدا نوشت.

به برنامه یک کادر تصویر اضافه کنید. با کلیک روی پرونده‌های تصویری، تصویر آن‌ها در کادر تصویر نمایش داده شود و اگر پرونده صوتی با قالب wav بود پخش شود.

برای نمایش تصویر و پخش صوت کد زیر را در رخداد `SelectedIndexChanged` از کادر فهرست بنویسید.

```
string filePath = Path.Combine(currentDir,
listBox2.SelectedItem.ToString());
string ext = Path.GetExtension(filePath);
if ((ext == ".jpg") || (ext == ".bmp") || (ext == ".gif") ||
(ext == ".png"))
{
    pictureBox1.Load(filePath);
}
if ((ext == ".wav"))
{
    System.Media.SoundPlayer audio = new
System.Media.SoundPlayer(filePath);
    audio.Play();
}
```

## کارگاه ۱۰ – مدیریت پرونده‌ها و پوشه‌ها

برای عمل درج در مقصد اگر نسخه برداری از پوشه باشد باید همه پوشه و زیرپوشه‌ها در مقصد ایجاد شود و سپس از پرونده‌ها نسخه برداری شود. کد را کامل کرده، برنامه را اجرا کنید. از ساختار try-catch برای مدیریت حالت‌های استثنا استفاده کنید.

```
case OperationType.Copy:
    try
    {
        if (Directory.Exists(sourcePath))
        {
            // ایجاد تمام پوشه‌ها و زیر پوشه‌ها
            foreach (string dirPath in
Directory.GetDirectories(sourcePath, "*",
SearchOption.AllDirectories))
Directory.CreateDirectory(dirPath.Replace(sourcePath
, destPath));

            // نسخه برداری از تمام پرونده‌ها و بازنویسی پرونده‌های هم نام
            foreach (string newPath in
Directory.GetFiles(sourcePath, "*.*",
SearchOption.AllDirectories))
File.Copy(newPath,
newPath.Replace(sourcePath, destPath), true);
        }
        if (File.Exists(sourcePath))
File.Copy(sourcePath, destPath);
    }
    catch
    {
        MessageBox.Show("خطا", "نیست پذیر امکان کپی",
MessageBoxButtons.OK,
MessageBoxIcon.Warning);
    }
    break;
```

فعالیت منزل  
ص ۲۴۵

برای ایجاد پوشه جدید، یک دکمه به برنامه کارگاه قبلی اضافه کنید. در رویداد کلیک این دکمه از متد `CreateDirectory`، برای ایجاد یک پوشه جدید با نام `New folder` در مسیر جاری استفاده کنید.

```
private void buttonNewFolder_Click(object sender, EventArgs e)
{
    Directory.CreateDirectory(Path.Combine(currentDir, "New
folder"));
    showDirectory(currentDir);
}
```

یک دکمه برای تغییر نام پرونده و پوشه به برنامه اضافه کنید. (راهنمایی از متد `Move` استفاده کرده، ورودی دوم متد را نام جدید قرار دهید).

```
private void buttonRename_Click(object sender, EventArgs e)
{
    try
    {
        string name = Path.Combine(textBoxAddress.Text,
listBoxDirectory.Text);
        string newName = Path.Combine(textBoxAddress.Text,
textBoxNewName.Text);
        if (Directory.Exists(name))
            Directory.Move(name, newName);
        if (File.Exists(name))
            File.Move(name, newName);
        showDirectory(currentDir);
    }
    catch
    {
        MessageBox.Show("تغییر نام امکان پذیر نیست", "خطا",
        MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    }
}
```



شکل ۱۳- فرم مدیریت پرونده‌ها و پوشه‌ها

## ثبت وقایع برنامه در یک پرونده

لاگ فایل‌ها یک ابزار استاندارد در دست مدیران و توسعه‌دهندگان سیستم‌های رایانه‌ای هستند. آن‌ها "چه چیزی، چه زمانی و به وسیله چه کسی رخ داد"های سیستم را در خود ضبط می‌کنند. ضبط این داده‌ها می‌تواند به تشخیص ایرادهای سیستم کمک کند. با ثبت وقایع و رخدادها می‌توان نفوذکنندگان به سیستم و حفره‌های امنیتی را شناسایی کرد و در بازبینی و واریسی سیستم از آن‌ها بهره گرفت. متداول‌ترین روش برای ثبت وقایع برنامه، نوشتن این وقایع در یک پرونده متنی است. خوبی این روش این است که رخدادها ثبت شده در پرونده متنی، همیشه و همه جا در دسترس هستند حتی وقتی که سیستم کاملاً از کار افتاده باشد. به دلیل کاربرد فراوان و گسترده این روش در بین توسعه‌دهندگان سیستم‌های نرم‌افزاری، ابزارهای متعددی برای آن به وجود آمده است. از جمله معروفترین لاگرها یا ابزارهای لاگ در محیط .Net می‌توان log4net و NLog را نام برد. به عنوان نمونه چند خط از یک لاگ فایل تولید شده توسط log4net را ببینید.

```
2016-08-10 11:54:26,798 [10] INFO Log4Net.Program [(null)] - Info logging
2016-08-10 11:54:26,824 [10] ERROR Log4Net.Program [(null)] - This is my error
System.Exception: Exception!
   at Log4Net.Program.Main(String[] args) in
C:\Test\Log4Net\Program.cs:line 14
```

### پروژه سامانه خودپرداز

در این پروژه هنجاریان یک برنامه شبیه‌ساز سیستم خودپرداز بانکی می‌نویسند. در این برنامه، فعالیت‌های صورت گرفته توسط کاربران سیستم خودپرداز بانکی، شامل هر کلیدی که توسط کاربر زده می‌شود، در یک پرونده متنی به نام ATM.LOG با تاریخ و ساعت ثبت می‌شود.

این خودپرداز ۵ صفحه مختلف دارد. هنجاریان می‌توانند برای طراحی صفحات خودپرداز از نرم‌افزار powerpoint استفاده کنند. هر صفحه در قالب یک اسلاید طراحی شده، در انتها اسلایدها به صورت پرونده تصویری ذخیره می‌شوند.

هنجاریان پس از طراحی رابط کاربری در ادامه برای رخداد کلیک دکمه‌های شبیه‌ساز کدنویسی می‌کنند. این برنامه به صورت یک ماشین حالت (State Machine) کار می‌کند. بر اساس رویدادهای صورت گرفته به وسیله کاربر، صفحات خودپرداز تغییر می‌کنند. برای تغییر صفحات، تصویر صفحه جدید جایگزین تصویر قبلی در کادر تصویر می‌شود. از یک کادر متنی که روی تصویر قرار داده شده، برای گرفتن گذرواژه کاربر استفاده شده است.

```
namespace ATM
{
```

```

enum PageName
{
    Page1, Page2, Page3, Page4, Page5
}
public partial class ATMForm : Form
{
    PageName currentPage = PageName.Page1;
    string password = "4386";

    public ATMForm()
    {
        InitializeComponent();
        textBoxPassword.PasswordChar = '*';
    }
    private void buttonInsrtCard_Click(object sender,
EventArgs e)
    {
        if (currentPage == PageName.Page1)
        {
            File.AppendAllText("ATM.LOG",
DateTime.Now.ToString("yyy/MM/dd HH:mm:ss") + ":" + "کارت ورود" +
"\r\n");
            pictureBoxScreen.Image =
Properties.Resources.EnterPassword;
            currentPage = PageName.Page2;
            textBoxPassword.Visible = true;
            textBoxPassword.Focus();
        }
    }
    private void buttonGetCard_Click(object sender, EventArgs
e)
    {
        if (currentPage == PageName.Page4)
        {
            pictureBoxScreen.Image =
Properties.Resources.EnterCard;
            currentPage = PageName.Page1;
        }
    }
    private void buttonCancel_Click(object sender, EventArgs
e)
    {
        if (currentPage != PageName.Page1)
        {
            pictureBoxScreen.Image =
Properties.Resources.TakeCard;
            currentPage = PageName.Page4;
            textBoxPassword.Visible = false;
        }
    }
    private void buttonClear_Click(object sender, EventArgs
e)

```



```

    {
        textBoxPassword.Clear();
        textBoxPassword.Focus();
    }
    private void buttonEnter_Click(object sender, EventArgs
e)
    {
        switch (currentPage)
        {
            case PageName.Page2:
                if (textBoxPassword.Text == password)
                {
                    pictureBoxScreen.Image =
Properties.Resources.SelectService;
                    textBoxPassword.Visible = false;
                    currentPage = PageName.Page3;
                }
                else
                {
                    pictureBoxScreen.Image =
Properties.Resources.EnterPassword;
                    textBoxPassword.Visible = true;
                    textBoxPassword.Clear();
                    textBoxPassword.Focus();
                    currentPage = PageName.Page5;
                }
                break;
            case PageName.Page5:
                if (textBoxPassword.Text == password)
                {
                    pictureBoxScreen.Image =
Properties.Resources.SelectService;
                    textBoxPassword.Visible = false;
                    currentPage = PageName.Page3;
                }
                else
                {
                    pictureBoxScreen.Image =
Properties.Resources.EnterCard;
                    textBoxPassword.Visible = false;
                    currentPage = PageName.Page1;
                }
                break;
        }
    }
}

```

## پس از تدریس

## فعالیت‌های تکمیلی

## توسعه پروژه سیستم خودپرداز

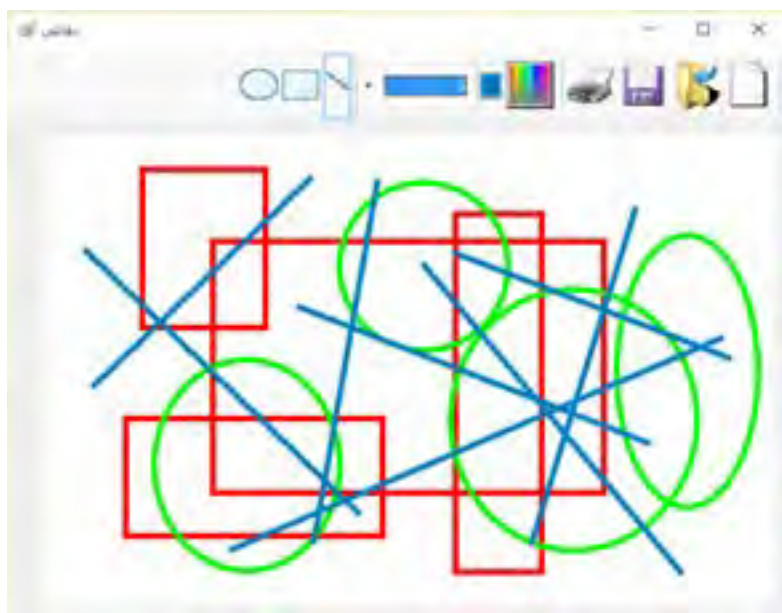
برای توسعه این پروژه می‌توان صفحات جدیدی به آن افزود. مثلاً صفحه پرداخت قبض و صفحه دریافت یا انتقال وجه. همچنین می‌توان به جای استفاده از کنترل کادر تصویر، از یک کنترل مرورگر وب (webBrowser) برای نمایش صفحات خودپرداز استفاده کرد. در این حالت هر صفحه به صورت یک پرونده html طراحی می‌شود. همچنین می‌توان مدت زمان توقف در هر صفحه را با استفاده از تایمر محدود کرد.

## پروژه‌های بیشتر برای هنرجویان علاقمند

در هر کلاس هنرجویانی هستند که به نسبت علاقمندی بیشتری به مباحث برنامه‌نویسی دارند و ممکن است از قبل هم تا حدودی با برنامه‌نویسی آشنا بوده باشند. معمولاً روند عادی کلاس برای این هنرجویان کند و ملال آور بوده و این امر می‌تواند باعث سرخوردگی و از بین رفتن انگیزه آن‌ها شود. هنرآموزان اغلب برای حفظ روحیه و ایجاد انگیزه در این هنرجویان، در گروه‌بندی‌های کلاسی آن‌ها را به عنوان سرگروه انتخاب می‌کنند. یک راهکار مؤثر دیگر برای تشویق این هنرجویان و ایجاد انگیزه مضاعف در آن‌ها، تعریف و مشخص کردن برخی پروژه‌های ساده برای آن‌ها است. در انجام این پروژه‌ها هنرجویان دیگر هم می‌توانند به صورت گروهی مشارکت داشته باشند. این پروژه‌ها را هنرجویان در منزل انجام می‌دهند و برای رفع اشکال به هنرآموز خود مراجعه می‌کنند. هنرآموزان باید دقت داشته باشند که متناسب با میزان آمادگی هنرجویان پروژه‌ها را تعریف کنند و در نظر داشته باشند که سپردن یک پروژه سنگین که خارج از توان هنرجو است نتیجه عکس خواهد داد و موجب افت انگیزه در هنرجویان خواهد شد. هنرآموزان می‌توانند فیلم آموزشی از پیاده‌سازی این پروژه‌ها را در اختیار هنرجویان قرار دهند.

## فیلم پروژه نقاشی (Paint)

در این آموزش هنرجویان برنامه‌ای نظیر برنامه نقاشی ویندوز ایجاد خواهند کرد. کنترل‌های استفاده شده در این برنامه همگی برای هنرجو آشنا هستند و تنها موارد جدید در این برنامه استفاده از کلاس‌های Graphics و Bitmap برای ایجاد اشکال گرافیکی خط، دایره و مستطیل است. این برنامه تمرینی برای کار با پرونده‌های تصویری در برنامه است.



شکل ۱۴- فرم برنامه paint

### فیلم پروژه بازی مارپله (Snake & Ladder)

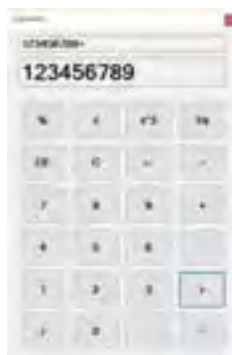
بازی‌ها همیشه جذابیت خود را دارند و نوشتن یک بازی جذابیتی مضاعف برای هنرجویان خواهد داشت. این پروژه فرصتی برای هنرجویان است که از آموخته‌های خود برای نوشتن یک بازی استفاده کنند. حالت ساده این بازی بدون استفاده از تایمر است. در حالت پیشرفته‌تر می‌توان از کنترل تایمر برای پویانمایی حرکت مهره‌ها استفاده کرد.



شکل ۱۵- فرم بازی مار و پله

### فیلم پروژه ماشین حساب (Calculator)

در این آموزش با بهره از متد Compute از کلاس DataTable یک ماشین حساب ساده ایجاد خواهد شد.



شکل ۱۶- فرم برنامه ماشین حساب

## واحد یادگیری ۱۰

### شایستگی کار با پایگاه داده در برنامه نویسی

#### مقدمات تدریسی

#### الف) مفاهیم کلیدی

مفاهیم کلیدی			
مقیدسازی	کنترل DataGridView	ارتباط با پایگاه داده از طریق wizard	طراحی پایگاه داده
کنترل BindingNavigator	کنترل BindingSource	کنترل TableAdaptor	کنترل DataSet
	کنترل خطا در برنامه پایگاه داده	متدهای TableAdaptor	پرونده با قالب xsd

#### ب) مراحل انجام کار واحد یادگیری

مراحل انجام کار	
۱	مقیدسازی واسط کاربری
۲	جستجوی داده ها
۳	انجام عملیات روی داده در محیط برنامه نویسی
۴	رفع اشکالات احتمالی برنامه

## ج) بودجه بندی

زمان بندی پیشنهادی واحد یادگیری:

ردیف	اهداف توانمندسازی	سهم زمان	
		تتوری	عملی
۱	ضرورت به کارگیری سیستم مدیریت پایگاه داده و طراحی پایگاه داده مناسب برای پروژه	۲	۲
۲	اتصال یک DataGridView به پایگاه داده و سفارشی کردن کنترل DataGridView	۱	۳
۳	و مقیدسازی کنترل ها و برنامه نویسی برای پیمایش رکوردها در کنترل DataGridView	۱	۳
۴	جستجوی دقیق و مشابه و ترکیبی در پایگاه داده	۱	۳
۵	اضافه و حذف و ویرایش رکورد در جدول	۱	۳
۶	اضافه و حذف کردن سطر از کنترل DataGridView با کدنویسی و رفع خطای احتمالی پروژه	۱	۴
۷	توسعه پروژه پایگاه داده	۰	۳
مجموع		۷	۲۱

## طرح درس پیشنهادی

طرح درس روزانه (هفتگی) پیشنهادی				
پایه: یازدهم			درس: توسعه برنامه‌سازی و پایگاه‌داده	
پیام جلسه (هدف کلی): ارتباط با پایگاه‌داده از طریق Wizard و سفارشی کردن DataGridView				
اهداف یادگیری		فعالیت‌ها		زمان
فعالیت	طبقه هدف: حیطه عاطفی / شناختی / روانی – حرکتی	کار هنرآموز	کار هنرجو	مدت (دقیقه)
ارزشیابی رفتار ورودی	سنجش میزان آگاهی هنرجویان از ایجاد پروژه ویندوزی و آشنایی با مفاهیم پایگاه‌داده	معطوف کردن توجه هنرجویان به ابزارهایی که در برنامه ویندوزی برای ارتباط با پایگاه‌داده نیاز هست مثل کنترل DataGridView	از هنرجو بخواهید که کنترل DataGridView را در جعبه ابزار پیدا کنند و مشخصات آن را بررسی کنند.	۱۰
ایجاد انگیزه	ایجاد توجه و تمرکز برای ورود به بحث ایجاد ارتباط اولیه پایگاه‌داده نمونه به پروژه ویندوزی و تنظیم ویژگی‌های کنترل DataGridView	با استفاده از ویدئوپروژکتور یا اجرای چند پروژه نمونه نمایش‌های مختلف داده‌ها را در DataGridView و دیگر کنترل‌ها مقید نمایش دهد.	از هنرجو بخواهید که نمونه برنامه‌هایی در محیط اطراف خود را نام ببرد که داده‌ها را نمایش می‌دهد.	۲۰
رانه مفاهیم کلیدی (توضیح هنرآموز)	توضیح کامل مفاهیم کلیدی (دانشی) و ایجاد علاقه و انگیزه در هنرجویان (بیشی)	پس از ایجاد پایگاه‌داده نمونه(Dictionary) و جدول (computerTbl) فیلم نمایش داده‌های جدول در DataGridView را نمایش دهد و یک نمونه ارتباط ایجاد کند.	هنرجو با دقت فیلم و توضیحات هنرآموز را ببیند.	۳۰

طرح درس روزانه (هفتگی) پیشنهادی				
پایه: یازدهم			درس: توسعه برنامه‌سازی و پایگاه‌داده	
پیام جلسه (هدف کلی): ارتباط با پایگاه‌داده از طریق Wizard و سفارشی کردن DataGridView				
اهداف یادگیری		فعالیت‌ها		زمان
انجام فعالیت کارگاهی شماره ۱ (تمرین هنجویان)	هنرجو باید یک ارتباط ویزاردی با پایگاه‌داده انجام دهد و داده‌های یک جدول را در یک کنترل DataGridView نمایش دهد.	هنرآموز توضیح کامل در خصوص فعالیت کارگاهی ۱ به هنرجو بدهد و از هنرجو بخواهد به سوالات کارگاه ۱ پاسخ دهند.		۶۰
نظارت بر عملکرد هنجویان و ارزیابی فعالیت‌ها	بررسی نقاط ضعف هنجویان در درک مفاهیم کلیدی	رفتار هنرجو را در هنگام اجرای تکالیف زیر نظر داشته باشد و در صورت لزوم از هنجویان مستعدتر برای آموزش مفاهیم به آن‌ها کمک بگیرد		۳۰
انجام فعالیت کارگاهی شماره ۲ (تمرین هنجویان)	هنرجو باید کنترل DataGridView را بشناسد و آن را سفارشی کند	هنرآموز توضیح کامل در خصوص فعالیت کارگاهی ۲ به هنرجو بدهد و جدول ۴ را تکمیل کنند.		۶۰
نظارت بر عملکرد هنجویان و ارزیابی فعالیت‌ها	بررسی نقاط ضعف هنجویان در درک مفاهیم کلیدی	رفتار هنرجو را در هنگام اجرای تکالیف زیر نظر داشته باشد و در صورت لزوم از هنجویان مستعدتر برای آموزش مفاهیم به آن‌ها کمک بگیرد		۳۰
دریافت بازخود از تدریس	هنرجو بتواند فعالیت منزل صفحه ۲۵۷ کتاب را انجام دهد	تقسیم‌بندی وظایف کار فعالیت در منزل بین اعضای گروه و تحویل در زمان‌بندی تعیین‌شده		۶۰
ابزارهای موردنیاز		ویدئو پروژکتور، رایانه، تخته آموزشی، دفتر یادداشت		



## ه) فیلم‌ها و پرونده‌های پیوست

شماره فیلم	هدف
111h49	طراحی و برنامه‌نویسی فرم جدول مشتری در پایگاه‌داده فروشگاه
111h50	طراحی و برنامه‌نویسی فرم اخذ رأی برنامه انتخابات شورای دانش‌آموزی
111h51	توسعه پروژه فروشگاه
111h52	برنامه‌نویسی چند لایه در پایگاه‌داده برای مدیریت مشتریان

## و) ورود به بحث

هدف از این واحد شایستگی، چگونگی طراحی و برنامه‌نویسی برای یک پایگاه‌داده در یک محیط عملیاتی است. هنرجو ابتدا باید پایگاه‌داده را طراحی کند؛ سپس از طریق ویزارد یک ارتباط با پایگاه‌داده ایجاد کند و پس از آن می‌تواند عملیات مختلف را روی پایگاه‌داده انجام دهد. در نهایت هنرجو باید بتواند عمل کنترل خطا را در یک برنامه پایگاه‌داده انجام دهد.

برای آموزش برنامه‌نویسی پایگاه‌داده در سی شارپ این سوال برای هنرجویان مطرح می‌شود که با این که در فصل اول مدیریت پایگاه‌داده با Access آموزش داده شد، چه نیازی به مدیریت پایگاه‌داده با سی شارپ است.

۱) از هنرجویان بپرسید آیا همه کاربران نهایی می‌توانند پایگاه‌داده را در Access مدیریت کنند.

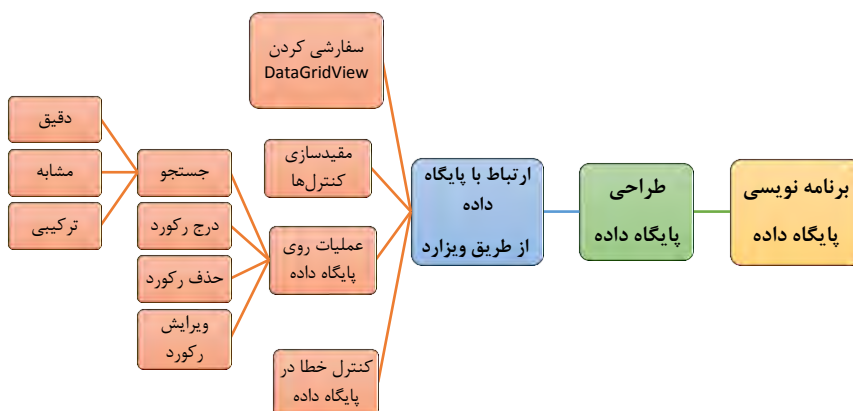
۲) با نمایش برخی از برنامه‌های کاربردی مختلف ویندوزی و تلفن‌های هوشمند، از هنرجویان بپرسید که کدام برنامه‌ها کاربرپسندتر هستند؟ مدیریت برنامه در Access یا ایجاد یک برنامه جذاب ویندوزی برای مدیریت پایگاه‌داده

مزیت‌های استفاده از برنامه پایگاه‌داده نسبت به پرونده‌های ترتیبی و دودویی در کلاس بیان شود.

با مراجعه به کارگاه دفترچه تلفن واحد ۹ کتاب تفاوت برنامه با پرونده‌ای ترتیبی و برنامه پایگاه‌داده را نشان دهید.

برای آموزش این واحد یادگیری هنرجو باید با مفاهیم اولیه پایگاه‌داده نظیر طراحی پایگاه‌داده و عملیات روی پایگاه‌داده و زبان پرس‌وجوی SQL آشنایی اولیه داشته باشد.

نمودار زیر نقشه مفهومی و ذهنی این واحد یادگیری را نمایش می‌دهد و شما می‌توانید با داشتن یک دورنما از مطالب و محتوای آموزشی، تدریس خود را به شکل بهتری مدیریت کنید.



نمودار ۱- نقشه مفهومی و ذهنی واحد یادگیری ۱۰

## مفاهیم اولیه پایگاه داده

یکی از مهمترین بخش‌های برنامه نویسی پایگاه داده، طراحی پایگاه داده مربوطه است که اگر طراحی مناسب صورت نگیرد، برنامه دچار مشکلات بزرگی خواهد شد. برای استخراج گزارش‌های کاربردی و جلوگیری از مشکلاتی نظیر وجود داده‌های تکراری و تداخل و ... باید پایگاه داده دقیق و ارتباطی ایجاد گردد. جهت ایجاد پایگاه داده مناسب پودمان اول کتاب برای هنرجو به اختصار یادآوری شود و پایگاه داده فروشگاه به صورت گام به گام به هنرجو آموزش داده شود و برای تقویت طراحی، فعالیت‌های این واحد انجام شود.

### پاسخ به فعالیت‌ها

به کمک هم‌گروهی خود یک پایگاه‌داده برای ذخیره مخاطبان ایجاد کنید و به سوالات زیر پاسخ دهید.

فعالیت کارگاهی  
ص ۲۵۱

■ چگونه پایگاه‌داده را در Access ایجاد می‌کنید؟

بعد از اجرای برنامه Access 2016 و انتخاب گزینه Blank desktop database، تعیین مسیر مناسب برای ایجاد پایگاه داده، نام پایگاه‌داده را PhoneNotebook قرار دهید و دکمه Create را کلیک کنید.

■ چگونه در پایگاه‌داده، جدول مخاطبان را ایجاد می‌کنید؟

از سربرگ Create گزینه Table را انتخاب کنید. روی جدول ایجاد شده در سمت چپ کلیک راست کرده، گزینه Design View را انتخاب کنید. نام جدول را Contacts قرار دهید و فیلدهای مناسب را در جدول قرار دهید.

■ چه فیلدهایی را برای جدول مخاطبان تعریف می‌کنید؟ نوع داده این فیلدها چیست؟

در جدول زیر حداقل فیلدها برای جدول مخاطبان تعریف شده است:

Field Name	Data Type	
CountryCode	Short Text	کد کشور
MobileNumber	Short Text	تعداد شماره
ContactFName	Short Text	نام مخاطب
ContactLName	Short Text	نام خانوادگی مخاطب
Organization	Short Text	شرکت
Email	Short Text	ایمیل
ContactPicture	Short Text	آدرس تصویر مخاطب

■ چگونه رکورد جدید در جدول مخاطبان وارد می‌کنید؟

با دابل کلیک روی جدول می‌توان در صفحه گسترده (DataSheet View) باز شده، زیر هر ستون مقدار مناسب را وارد کرد و یک رکورد جدید با کامل کردن یک سطر صفحه گسترده ایجاد می‌شود.

■ چگونه یک رکورد را حذف می‌کنید؟

با کلیک راست روی سطر دلخواه در صفحه گسترده و با انتخاب Delete Record می‌توان یک رکورد دلخواه را حذف کرد.

#### کنجکاوی ص ۲۵۱

چند نرم‌افزار پیرامون خود را نام ببرید که از پایگاه‌داده برای ذخیره و بازیابی اطلاعات استفاده می‌کنند.

پاسخ: نرم‌افزار دانا مدارس - نرم‌افزار حسابداری گردو  
نام بازی‌هایی را بنویسید که فکر می‌کنید از پایگاه‌داده برای ذخیره اطلاعات بازی استفاده می‌کنند.

پاسخ: Soccer Stars و Pes2016

چند نرم‌افزار فارسی، تلفن هوشمند را نام ببرید که از پایگاه‌داده استفاده می‌کند.

پاسخ: بازار - لغت نامه معین - اسنپ

آیا امکانات موجود در نرم‌افزارهای مدیریت پایگاه‌داده نظیر Access برای تولید یک نرم‌افزار که کار ذخیره و بازیابی اطلاعات را انجام دهد، کافی نیست؟ چرا؟

پاسخ: خیر. چون برای استفاده کامل از امکانات Access کاربر باید حرفه‌ای باشد؛ بنابراین برای مدیریت بهتر باید نرم‌افزار کاربردی طراحی کرد تا کاربر معمولی بتواند به سادگی از آن استفاده کند.

یک بازی آنلاین که مشخصات و سطح و امتیاز کاربران در آن ذخیره می‌شود را بررسی کنید. فکر می‌کنید پایگاه‌داده این بازی از چه جدول‌هایی تشکیل شده است؟

پاسخ: جدول کاربران، جدول بازی‌ها، جدول ابزار و....

#### فعالیت کارگاهی ص ۲۵۲

جدول بالا را در Access پیاده‌سازی کنید و رکوردهای بالا را در جدول وارد کنید. اگر دو کارمند فرهنگی نام و نام خانوادگی یکسانی داشته باشند، آیا برای صورت‌حساب آن‌ها مشکلی پیش می‌آید؟ چرا؟ بله، مغایرت در داده‌ها پیش می‌آید.

مشکل را چگونه حل می‌کنید؟

پاسخ: با تعریف کلید اصلی برای جدول.

برگ خرید شامل چه داده‌هایی می‌شود؟  
برای نمایش کالاها در برگ خرید نیاز به «جدول کالا» داریم. این جدول شامل چه فیلدهایی می‌شود؟

Goods جدول کالا

Field Name	Data Type	
GoodsCode	Number	کد کالا
GoodsName	Short Text	نام کالا
UnitPrice	Number	قیمت واحد
Stock	Number	موجودی

نکته: در جدول بالا می‌تواند فیلدهای دیگری نیز تعریف شود. کد کالا می‌تواند از نوع AutoNumber هم تعریف شود.

برای نمایش اطلاعات مربوط به کارمند فرهنگی در برگ خرید نیاز به «جدول مشتری» داریم. این جدول شامل چه فیلدهایی می‌شود؟

Customers جدول مشتری

Field Name	Data Type	
CustomerCode	Number	کد مشتری
CustomerName	Short Text	نام
CustomerFamily	Short Text	نام خانوادگی
mobile	Short Text	شماره همراه

نکته: در جدول بالا می‌تواند فیلدهای دیگری نیز تعریف شود. کد مشتری می‌تواند از نوع AutoNumber هم تعریف شود.

برای ایجاد هر برگ خرید نیاز به «جدول برگه خرید» داریم که مشخص کند، مشتری در چه تاریخی چه کالاهایی خریده است. این جدول شامل چه فیلدهایی می‌شود؟

Factor جدول برگه خرید

Field Name	Data Type	
FactorNO	Number	شماره برگه خرید
CustomerCode	Number	کد مشتری
FactorDate	Date/Time	تاریخ برگه خرید

نکته: شماره برگه خرید می‌تواند از نوع AutoNumber هم تعریف شود.  
مشتری در هر بار مراجعه ممکن است چندین قلم کالا خریداری کند؛ بنابراین نمی‌توان کالاهای خریداری‌شده را در «جدول کالا» ذخیره کرد به همین دلیل، جدول دیگری به نام «جدول ثبت کالا در برگه خرید» ایجاد می‌کنیم. این جدول شامل چه فیلدهایی می‌شود؟

فصل پنجم: مدیریت پایگاه داده

Registerfactor جدول ثبت کالا در برگه خرید		
Field Name	Data Type	
FactorNO	Number	شماره برگه خرید
GoodsCode	Number	کد کالا
Amount	Number	تعداد کالا برای خرید

فعالیت منزل  
ص ۲۵۳

فرض کنید در فروشگاه فرهنگیان بیش از یک فروشنده کار فروش کالا را انجام می‌دهند، اگر قرار باشد در برگ خرید، نام فروشنده نیز قید شود، چه تغییری باید در ساختار پایگاه‌داده فروشگاه صورت گیرد؟



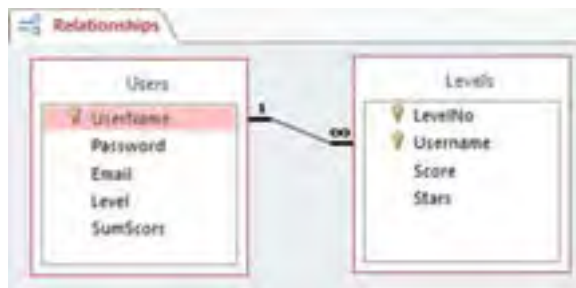
جدول Seller جدول فروشنده است.

- معاون پرورشی هنرستان جابرین حیان از هنرجویان رشته شبکه و نرم‌افزار رایانه درخواست برنامه‌ای برای سامان‌دهی امانت کتاب، در کتابخانه مدرسه کرده است. هنرجویان ابتدا می‌خواهند پایگاه‌داده کتابخانه را ایجاد کنند. با فرض اینکه هر هنرجو می‌تواند در هر بار مراجعه، چند کتاب را از کتابخانه امانت بگیرد، طراحی پایگاه‌داده کتابخانه هنرستان جابرین حیان را انجام دهید.



- در یک بازی می‌توان با ایجاد یک نام کاربری وارد بازی شد. هر کاربر با

کسب حداقل امتیاز به سطح بالاتر می‌رود و با توجه به امتیاز خود ستاره کسب می‌کند. پایگاه داده مناسب را برای این بازی طراحی کنید تا نشان دهد که هر کاربر در هر سطح چند امتیاز و چند ستاره دارد.



## کارگاه ۱ – ارتباط با پایگاه داده از طریق Wizard

ساده‌ترین راه برای ایجاد ارتباط با پایگاه داده در سی‌شارپ، یک ارتباط ویزارد است که می‌توان با قرار دادن یک کنترل مقید نظیر DataGridview این ارتباط را ایجاد کرد. با ایجاد این ارتباط، نسخه‌ای از پرونده پایگاه داده در مسیر جاری پروژه قرار می‌گیرد و پرونده با پسوند xsd (نام+DataSet.xsd) پایگاه داده) در بخش Solution Explorer قرار می‌گیرد. با ایجاد ارتباط با پایگاه داده به صورت ویزارد، سه کنترل (DataSet و BindingSource و TableAdaptor) به صورت خودکار به فرم اضافه می‌شود. یک کنترل DataSet برای یک پایگاه داده و برای هر جدول یا دید یک کنترل BindingSource و TableAdaptor متناظر اضافه می‌شود. البته اگر شما از طریق مقیدسازی کنترلی به جدول اقدام نکرده‌اید کنترل‌های BindingSource و TableAdaptor اضافه نمی‌شوند.

### کلاس DataSet

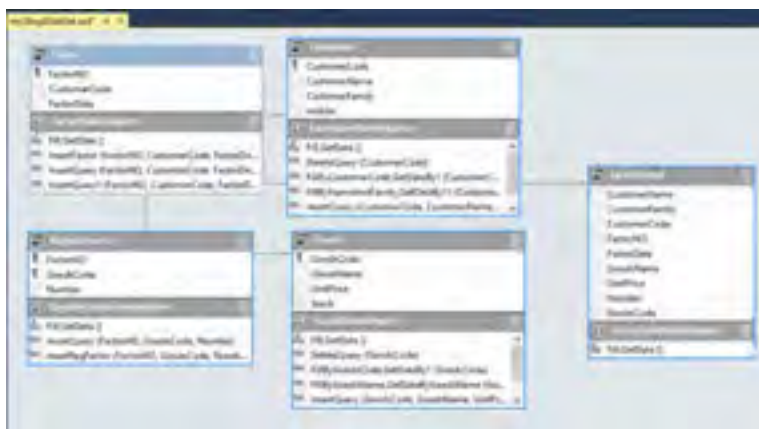
کلاس DataSet برای نگهداری اطلاعات به دست آمده از پایگاه داده در حافظه بکار می‌رود. این کلاس شامل مجموعه‌ای از جدول‌ها، رابطه‌ها، قید و شرط‌ها و دیگر مواردی است که از پایگاه داده خوانده شده است. این کلاس خود همانند یک موتور کوچک پایگاه داده عمل می‌کند که می‌تواند داده‌ها را در درون خود در جدولی مجزا نگهداری کرده، به کاربر اجازه دهد که آن‌ها را ویرایش کند و با استفاده از کلاس DataView پرس‌وجوهایی را روی داده‌های موجود در آن اجرا کند.

داده‌هایی که در این کنترل قرار دارند، ارتباطی با پایگاه داده ندارند. در طول برنامه می‌توان داده‌های موجود در آن را حذف، ویرایش یا اضافه کرد و بعد از اتمام

تغییرات مورد نظر دوباره با استفاده از DataAdaptor به پایگاه داده متصل شده، تغییرات را در پایگاه داده ذخیره کرد.

کلاس DataSet از ساختار XML برای ذخیره داده‌ها استفاده می‌کند. به این ترتیب می‌توانید داده‌های موجود در یک نمونه از کلاس DataSet را به سادگی در یک فایل ذخیره کرده، یا آن را با استفاده از شبکه به کامپیوتر دیگری منتقل کنید. البته هنگام برنامه‌نویسی و کار با DataSet لازم نیست با آن‌ها در قالب XML رفتار کنید، بلکه کافی است تمام کارهای موردنظر خود را با استفاده از ویژگی‌ها و متدهای موجود در DataSet انجام دهید.

مانند هر سند XML یک DataSet نیز دارای یک الگو است. پرونده‌ای با پسوند XSD ساختارهای داده‌ای درون یک یا چند پرونده XML را شرح می‌دهد. شکل زیر پرونده myShopDataSet.xsd را نمایش می‌دهد.

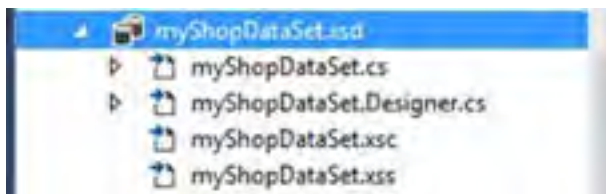


این پرونده (myShopDataSet.xsd) حاوی الگوی XML است که داده‌های myShopDataSet را نگهداری می‌کند. VS به وسیله این پرونده، یک کلاس از DataSet مشتق می‌کند تا بتواند داده‌های دریافت شده از پایگاه داده را در نمونه کلاس نگهداری کند. البته تمام موارد نیز از دید برنامه‌نویس پنهان می‌ماند و توسط DataSet انجام می‌شود.

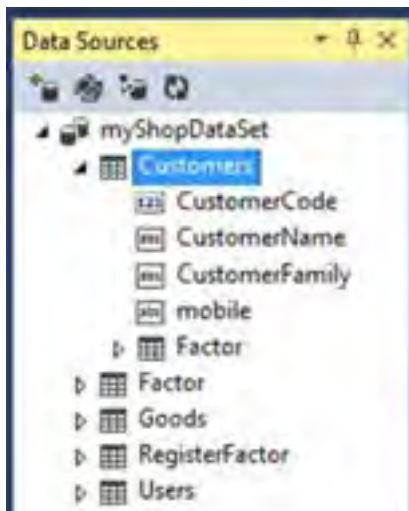
یکی از متدهای کلاس DataSet متد WriteXmlSchema است که می‌توان این شیما را در قالب یک پرونده xsd ایجاد کرد. این پرونده کاربردهای بسیاری دارد. برای همه عملیات روی پایگاه داده از جمله پرس و جوی‌های انتخابی (جستجو و نمایش) و عملیاتی (درج، حذف و ویرایش) و تجمعی می‌توان یک متد روی جدول یا TableAdaptor مربوط به جدول ایجاد کرد و در برنامه به آسانی آن را فراخوانی کرد. برای اضافه کردن متد روی TableAdaptor جدول موردنظر کلیک راست کرده، گزینه Add Query را انتخاب کنید و با توجه به فهرست پرس و جوها،



پرسوجو مورد نظر را به صورت ویزارد طراحی و با کلیک روی دکمه Query Builder به راحتی ویرایش کنید. نام مناسب برای این متدها قرار دهید. با کلیک روی فلش پرونده xsd در بخش Solution Explorer می‌توان پرونده‌های دیگری را مشاهده کرد که پس زمینه عملیات ایجاد این شما است.



برای مشاهده همه منابع داده از منوی View گزینه Other Windows و انتخاب گزینه Data Sources استفاده کنید و یا با فشار دادن همزمان کلیدهای Shift+Alt+D پنجره منابع داده را نمایش دهید. در منبع داده فهرست همه جدول‌ها یا دیدها به همراه فیلدهای جدول وجود دارد که با درگ کردن جدول یک کنترل DataGridView به همراه کنترل BindingNavigator برای پیمایش رکوردها به فرم اضافه می‌شود.



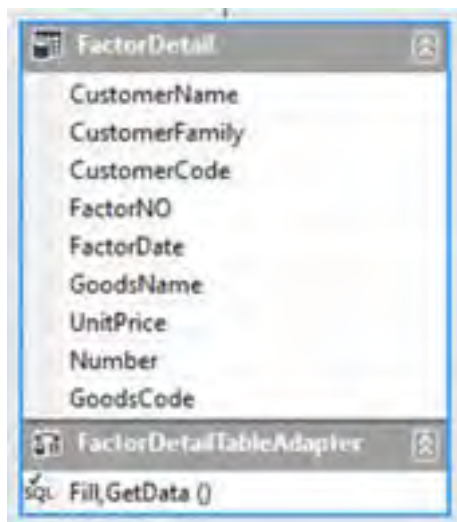
بدون استفاده از کنترل‌های مقید نظیر DataGridView می‌توانید از آیکن جمع پنجره Data Sources گزینه Add New Data Source یک منبع داده جدید اضافه کنید.

## ایجاد پرونده xsd

برای ایجاد کردن یک پرونده با قالب xsd به صورت دستی ابتدا از طریق منوی **Add New Item** → **Project** عنصر **DataSet** را انتخاب کرده، با نوشتن نام مناسب دکمه **Add** را کلیک کنید. یک پرونده با نام xsd در بخش **Solution Explorer** اضافه می‌شود. با دابل کلیک روی این پرونده مشاهده می‌شود که این **DataSet** حاوی چیزی نیست. برای قرار دادن جدول و عناصر دیگر از منوی **View** پنجره **Server Explorer** را فعال کنید و از طریق گزینه **Add Connection** یک پایگاه‌داده جدید به این پنجره اضافه کنید و در جدول‌های پایگاه‌داده مورد نظر یک جدول را انتخاب کرده، روی پرونده xsd بکشید و رها کنید. مشاهده می‌شود که جدول به همراه فیلدها به این پرونده اضافه می‌شود و می‌توانید متدهای مورد نیاز را ایجاد کرده، در برنامه فراخوانی کنید.

## افزودن یک TableAdaptor جدید

اگر بخواهیم داده‌هایی از چند جدول مرتبط به هم در یک پایگاه‌داده را مدیریت کنیم باید در پرونده xsd کلیک راست کرده و از گزینه **Add**، گزینه **TableAdaptor** را انتخاب کنید و با ایجاد یک اتصال جدید و از طریق **Query Builder** یک پرس‌وجوی جدید از چند جدول مرتبط به هم ایجاد کنید. با زدن دکمه **Finish** یک جدول جدید به همراه یک **TableAdaptor** جدید ایجاد می‌شود و می‌توان آن را همانند دیگر **TableAdaptor**ها مدیریت کرد.



جدول بالا از فیلدهای چهار جدول مرتبط به هم ایجاد شده است.

## پاسخ به فعالیت‌ها

تکمیل کارگاه  
ص ۲۵۴

چه کنترل‌هایی به فرم برنامه اضافه شده است؟  
پاسخ: بعد از اتصال کنترل DataGridView به منبع داده سه کنترل dictionaryDataSet و computerTblTableAdapter و computerBindingSource به فرم اضافه می‌شوند.

-کلید F7 را فشار دهید. آیا کدی به برنامه اضافه شده است؟  
پاسخ: در رویداد Form\_Load کدی به صورت زیر به برنامه اضافه می‌شود.  
`this.computertblTableAdapter.Fill(this.dictionaryDataSet.computertbl);`

## کارگاه ۲ - سفارشی کردن کنترل DataGridView

هدف از کارگاه ۲ آموزش استفاده از ویژگی‌های کاربردی کنترل DataGridView برای نمایش سفارشی و جذاب‌تر این کنترل است. از هنجارو بخواهید فهرست ویژگی‌های این کنترل را ببینید و با آزمون و خطا به کاربرد برخی از این ویژگی‌ها پی ببرید.

## پاسخ به فعالیت‌ها

تکمیل کارگاه  
ص ۲۵۶

مرحله ۲: فیلد ID باید از ستون‌های DataGridView حذف شود.

مرحله ۳: ستون‌ها فارسی می‌شوند.

مرحله ۴: کنترل DataGridView تمام فرم را در برمی‌گیرد.

مرحله ۵: با تنظیم این ویژگی می‌توانید اندازه ستون‌ها را متناسب با سلیقه خود تنظیم کنید. با استفاده از ویژگی FillWeight هر ستون می‌توانید درصد عرض ستون را مشخص کنید.

مرحله ۶: همه ستون‌های کنترل DataGridView حذف می‌شود.

جدول ۴ برخی از ویژگی‌های کنترل DataGridView را نمایش می‌دهد. این جدول را تکمیل کنید.

فعالیت گارگاهی  
ص ۲۵۶

نام ویژگی	شرح
Dock	مشخص می‌کند که کدام گوشه از کنترل به فرم محدود شود.
AutoSizeColumnsMode	وضعیت اندازه خودکار برای ستون را مشخص می‌کند.

DefaultCellStyle	قالب بندی خانه‌های کنترل DataGridView
DataSource	نام جدول داده یا نام کنترل BindingSource
HeaderText	عنوان در DataGridView
ColumnHeadersDefaultCellStyle	قالب بندی خانه‌های عنوان کنترل DataGridView
EnableHeadersVisualStyles	این ویژگی برای اعمال قالب بندی باید False شود.

### فعالیت منزل ص ۲۵۶

جدول مخاطبان تلفن همراه را روی فرم نمایش دهید.

پاسخ: بعد از ایجاد پایگاه داده و جدول مخاطبان Contacts و یک پروژ ویندوزی به نام PhoneNoteBook یک کنترل DataGridView به فرم اضافه کنید و از طریق ویزارد یک منبع داده ایجاد کنید و کنترل DataGridView جدول مخاطبان را به منبع داده مقید کنید. با کلیک راست روی DataGridView و انتخاب گزینه Edit Columns فیلدهای اضافی را حذف کنید و نام سرستون‌ها را ویرایش کنید. ویژگی Dock را Fill قرار دهید. ویژگی AutoSizeColumnsMode را Fill قرار دهید. به کمک ویژگی ColumnHeadersDefaultCellStyle و EnableHeadersVisualStyles قالب بندی سرستون‌ها را انجام دهید و با ویژگی DefaultCellStyle قالب بندی خانه‌های DataGridView را انجام دهید. برای عدم نمایش خطوط عمودی DataGridView ویژگی CellBorderStyle را SingleHorizontal قرار دهید.



شماره همراه	نام	نام خانوادگی
۰۹۱۱۷۷۶۹۰۹۰	رضا	محمدی
۰۹۱۲۳۴۰۶۰۵۰	باقر	فرهادی
۰۹۱۵۹۰۰۷۰۸۰	غلام	علوی
۰۹۱۷۸۰۰۵۰۹۰	بهرام	بهرامی
۰۹۱۸۷۰۰۶۰۹۰	رضا	یوسفی

اگر بخواهید نام و نام خانوادگی را در قالب یک ستون نمایش دهید. پرس و جوی متد Fill را به صورت زیر ویرایش کنید.

```
SELECT MobileNumber, CountryCode, ContactFName+''
```

+ContactLname AS ContactFullName, Oragnization, Email,  
ContactPicture

FROM            Contacts

نکته: در بخش Edit Column ویژگی DataPropertyName در کنترل  
DataGridView با نام مستعار ContactFullName تغییر یابد و ستون «نام  
خانوادگی» حذف شود.

از پایگاه داده فروشگاه، جدول کالا را روی فرم نمایش دهید.

همانند نمایش جدول مخاطبان روی فرم اقدام کنید.



در مورد ویژگی‌های دیگر کنترل DataGridView تحقیق کنید.

پژوهش  
ص ۲۵۷

به کاربر اجازه می‌دهد به DataGridView سطر جدید اضافه کند.	AllowUserToAddRow
به کاربر اجازه می‌دهد سطر دلخواه را انتخاب و با دکمه Delete صفحه کلید سطر انتخاب شده را از DataGridView حذف کند. نکته اینکه این سطر فقط از DataGridView حذف می‌شود و از پایگاه داده حذف نمی‌شود.	AllowUserToDeleteRow
به کاربر اجازه می‌دهد که اندازه (عرض) ستون‌ها را با استفاده از ماوس تغییر دهد.	AllowUserToResizeColumns
به کاربر اجازه می‌دهد که بتواند اندازه (ارتفاع) سطرها را با استفاده از ماوس تغییر دهد.	AllowUserToResizeRows

با استفاده از این ویژگی می‌توانید ظاهر DataGridView (از جمله رنگ زمینه، قلم، رنگ متن، رنگ سطر انتخاب شده، تنظیمات متن و...) را تغییر دهید.	AlternatingRowsDefaultCellStyle
با تنظیم این ویژگی می‌توانید اندازه ستون‌ها را متناسب با سلیقه خود تنظیم کنید.	AutoSizeColumnsMode
با تنظیم این ویژگی می‌توانید اندازه سطرها را متناسب با سلیقه خود تنظیم کنید.	AutoSizeRowsMode
این ویژگی رنگ پشت زمینه (بدنه) DataGridView را تعیین می‌کند.	BackgroundColor
این ویژگی نوع خط کادر DataGridView را تعیین می‌کند.	BorderStyle
این گزینه، نوع خطوط بین سلول‌های DataGridView را تعیین می‌کند (از جمله نوع‌های موجود در این ویژگی خط ساده، برجسته افقی، برجسته عمودی و...).	CellBorderStyle
این گزینه، نوع خطوط بین سلول‌های عنوان ستون‌ها DataGridView را تعیین می‌کند.	ColumnHeadersBorderStyle
این ویژگی، اندازه ارتفاع عنوان ستون‌ها را تعیین می‌کند.	ColumnHeaderHeight
با False کردن این ویژگی، عنوان ستون‌ها مخفی می‌شوند.	ColumnHeadersVisible
با استفاده از این ویژگی می‌توانید ستون‌های DataGridView را مدیریت (اضافه، حذف و ویرایش) کنید.	columns
این ویژگی، شکل آیکن ماوس را برای زمانی که وارد DataGridView می‌شود، تعیین می‌کند. (به عنوان مثال با انتخاب علامت Hand، وقتی ماوس روی DataGridView قرار می‌گیرد آیکن ماوس به شکل دست تغییر می‌کند.)	Cursor
با استفاده از این ویژگی می‌توانید ظاهر سلول‌های سطرهای فرد DataGridView را تغییر دهید (از جمله رنگ زمینه، قلم، رنگ متن، تنظیمات متن و...).	DefaultCellStyle
با استفاده از این ویژگی می‌توانید اندازه کنترل DataGridView را متناسب با اندازه فرم تنظیم کنید. (به عنوان مثال می‌توانید تعیین کنید که کنترل	Dock

<p>DataGridView به طور کامل سمت چپ فرم را بپوشاند و با تغییر اندازه فرم، اندازه DataGridView نیز تغییر کند)</p>	
<p>با استفاده از این ویژگی می توانید تعیین کنید که آیا کاربر می تواند یک سطر یا بیش از یک سطر را همزمان انتخاب (Select) کند.</p>	MultiSelect
<p>با True کردن این ویژگی، کاربر نمی تواند متن درون سلول ها را ویرایش کند و تنها می تواند متن درون DataGridView را مشاهده کند.</p>	ReadOnly
<p>با True کردن این ویژگی، سلول های درون DataGridView، در سمت راست کنترل DataGridView قرار می گیرند. ( در صورتی که متن درون DataGridView فارسی است، این گزینه را روی Yes قرار دهید)</p>	RightToLeft
<p>این گزینه، نوع خطوط بین سلول های عنوان سطرها (علامت ستاره کنار هر سطر) DataGridView را تعیین می کند.</p>	RowHeadersBorderStyle
<p>با False کردن این ویژگی، سلول های خالی که در اول هر سطر قرار دارند، مخفی می شوند.</p>	RowHeaderVisible
<p>این ویژگی، طول سلول های خالی که در اول هر سطر قرار دارند را تعیین می کند.</p>	RowHeaderWidth
<p>این ویژگی تعیین می کند که اگر تعداد سطرها و یا ستون های DataGridView بیشتر از اندازه طول و یا عرض کنترل شد، آیا نوار پیمایش در پایین و یا سمت راست کنترل نمایش داده شود یا خیر. یا اینکه تنها نوار افقی نمایش داده شود و یا تنها نوار عمودی و یا هر دوی آن ها و یا هیچکدام.</p>	ScrollBars
<p>این ویژگی تعیین می کند، زمانی که روی یکی از سلول ها کلیک شد، آن سطر یا ستون به طور کامل انتخاب شود.</p>	SelectionMode

## کارگاه ۳ – مقیدسازی کنترل‌ها

هدف از این کارگاه توضیح مفهوم مقیدسازی و نحوه مقیدسازی کنترل‌ها با ویزارد و کدنویسی است. از هنرجو بخواهید که کنترل‌ها مقید و نامقید را در جعبه ابزار مشخص کند.

### پاسخ به فعالیت‌ها

تکمیل کارگاه  
ص ۲۵۷

مرحله ۳: چه مقداری در txtCIName قرار می‌گیرد؟ مقدار واژه که روی سطر آن کلیک شده است.

مرحله ۴: برای مقیدسازی کنترل کادرفهرست چگونه ویژگی DataSource و DisplayMember را تغییر می‌دهید؟

از طریق فهرست DataSource کنترل کادرفهرست computerTblBindingSource را انتخاب و از طریق فهرست DisplayMember یکی از فیلدهای جدول computerTbl را انتخاب کنید.

مرحله ۶: ویژگی DataSource کنترل BindingNavigator را computerTblBindingSource قرار دهید. با اجرای برنامه به وسیله این کنترل می‌توان رکوردها را روی کنترل‌های مقید نظیر DataGridView و listBox پیمایش کرد

کنجکاوی  
ص ۲۵۸

چه کنترل‌هایی را می‌توان مقید کرد؟ نام چند کنترل را نام ببرید.  
پاسخ: کنترل‌هایی که ویژگی DataBinding دارند مقید هستند مثل checkbox و radioButton

چگونه با کدنویسی می‌توان یک کنترل را نامقید کرد؟  
پاسخ: برای نامقید کردن کنترل‌ها باید مقدار DataSource را null کنید.

`listBox1.DataSource = null;`

برای نامقید کردن کنترل‌هایی که ویژگی DataSource ندارند باید از طریق متد Clear، DataBinding اقدام کرد.

`txtCIName.DataBindings.Clear();`

## کارگاه ۴ – بارگذاری داده و پیمایش رکوردها در DataGridView

هدف این کارگاه آموزش استفاده از متد Fill کنترل TableAdaptor برای بارگذاری داده و متدهای پیمایش (Move) کنترل BindingSource برای پیمایش رکوردها در برنامه نویسی است.



هنگام ایجاد اتصال به صورت ویزارد دو کنترل TableAdaptor و BindingSource به فرم اضافه می‌شوند. در رویداد Form\_Load نیز کد زیر به صورت خودکار اضافه می‌شود.

```
this.goodsTableAdapter.Fill(this.myShopDataSet.Goods);
```

با دابل کلیک روی پرونده xsd متد Fill جدول Goods را به هنرجو نشان دهید و نحوه فراخوانی این متد را در کدنویسی آموزش دهید.

### پاسخ به فعالیت‌ها

- کنترل MyShopDataSet شامل چه جدول‌های دیگری است؟ اگر در هنگام ایجاد منبع داده به صورت ویزارد همه جدول‌ها انتخاب شده باشد، جدول Customers و Factor و RegisterFactor نیز در MyShopDataSet وجود دارد.

- آیا می‌توان برای مقیدسازی کنترل goodsDataGridView دستوری دیگری نوشت؟

```
this.goodsTableAdapter.Fill(this.myShopDataSet.Goods);
```

مرحله ۴: برای رفتن به آخرین رکورد (کالا) چه دستوری می‌نویسید؟

```
goodsBindingSource.MoveLast();
```

مرحله ۵: برای رفتن به رکورد بعدی (کالا) چه دستوری می‌نویسید؟

```
goodsBindingSource.MoveNext();
```

کنجکاوی  
ص ۲۵۹

## کارگاه ۵ - دسترسی به خانه‌های DataGridView

هدف از این کارگاه دسترسی به خانه‌های کنترل DataGridView از طریق کدنویسی است. اهمیت دسترسی به خانه‌های این کنترل برای نمایش در کادرمتن یا برچسب و یا انجام برخی محاسبات روی خانه‌های DataGridView لازم است. برای یادگیری بهتر این کارگاه، مفهوم Row و Cell را در کنترل DataGridView به هنرجو آموزش دهید. در این کارگاه برای دسترسی به رکورد جاری ویژگی CurrentCell.RowIndex و مفهوم رویداد CellClick نیز آموزش داده شود.

### پاسخ به فعالیت‌ها

مقدار اندیس (شماره) اولین رکورد چیست؟

پاسخ: صفر

کنجکاوی  
ص ۲۵۹

## کارگاه ۶ - جستجو در جدول

هدف از این کارگاه جستجوی دقیق و مشابه واژه‌ها در جدول است.

- از هنجرو در مورد کاربرد دستور Select که در پودمان اول فراگرفته، سوال کنید و بخواهید ساختار آن را یادآوری کند و سپس بخواهید وارد پرونده xsd شود و روی جدول یا TableAdaptor کلیک راست کرده، مشاهدات خود را بگوید و یک متد جدید را با پرس‌وجو Select ایجاد کرده، نام مناسب برای متد قرار دهد.
- در مورد دکمه Query Builder برای ایجاد پرس‌وجوی انتخابی با شرط توضیح دهید. در پنجره Query Builder نحوه اجرای یک پرس‌وجو را نشان دهید.
- عملکرد عملگر Like در پرس‌وجوی Select را با چند مثال در Access به هنجرو نشان دهید و از او بخواهید که متد جدیدی برای جستجوی واژه‌های مشابه ایجاد نماید.
- نحوه فراخوانی متدهای Select را در برنامه به صورت زیر به هنجرو آموزش دهید.

(ورودی‌های متد، نام جدول . نام DataSet) نام متد . نام TableAdaptor

### کار با داده‌ها با استفاده از DataView

در ADO.NET برای انجام عملیات (جستجو، مرتب سازی و فیلتر کردن) روی داده‌ها در یک DataTable دو روش وجود دارد:

- (۱) استفاده از متد Select شیء DataTable
- (۲) استفاده از شیء DataView

شیء DataView دارای توانایی بیشتری نسبت به متد Select شیء DataTable است و کار با آن نیز ساده‌تر است. شیء DataView مانند یک منبع برای اتصال کنترل‌های دیگر به آن مورد استفاده قرار می‌گیرد. شیء DataView نمایی از یک شیء DataTable است که برای جستجو، فیلتر و مرتب‌سازی و حتی دستکاری داده‌های آن شیء DataTable مورد استفاده قرار می‌گیرد.

هر شیء DataView شامل یک مجموعه DataRowView است که مشابه شیء DataRow در DataTable هستند. این اشیاء دارای ویژگی‌هایی هستند که در زیر به اختصار توضیح داده شده‌اند:

- **DataView شیء:** DataView را برمی‌گرداند که سطر جاری به آن تعلق دارد.
- **Item:** مقدار یک ستون از سطر فعلی را برمی‌گرداند.
- **Row:** شیء DataRow را برمی‌گرداند که در حال حاضر در حال بررسی است.

اشیاء `DataGridView` در داخل یک شیء دیگر به نام `DataGridViewManager` قرار می‌گیرند. رابطه بین این دو شیء مانند رابطه بین `DataSet` و `DataTable` است، با این تفاوت که امکان دارد یک شیء `DataTable` دارای چند شیء `DataGridView` باشد که برای انجام جستجو و فیلتر کردن مختلف روی داده‌های آن استفاده می‌شوند. در حقیقت در زمان اتصال یک کنترل به یک شیء `DataSet` در پروسه `Data-Binding` آن کنترل به شیء `DataGridView` مربوط به آن `DataSet` متصل می‌شود که با ویژگی `DefaultViewManager` شیء `DataSet` مشخص می‌شود.

برای ساخت یک شیء `DataGridView` از متد `New` استفاده می‌شود در مثال زیر یک شیء `DataGridView` می‌سازیم:

```
DataGridView myDataGridView = new DataGridView();
myDataGridView.Table = dictionaryDataSet.computertbl;
listBoxName.DataSource = myDataGridView;
```

در مثال بالا ابتدا یک شیء `DataGridView` جدید به نام `myDataGridView` ایجاد شده است. سپس ویژگی `Table` آن برابر با جدول کنترل `DataSet` که در برنامه کاربردی ما موجود است، قرار داده شده است. شیء `DataGridView` دارای ویژگی‌هایی است که در زیر به اختصار توضیح داده شده‌اند:

`AllowDelete` و `AllowEdit` و `AllowNew`: با تنظیم این مقادیر `Boolean` می‌توان با دستکاری و اضافه و حذف مقادیر در `DataGridView` مقادیر متناظر آن‌ها را در شیء `DataTable` مربوط تغییر داد.

■ `Count`: تعداد اشیاء `RowView` موجود در `DataGridView` جاری را برمی‌گرداند (مشابه ویژگی `Count` از شیء `DataTable`).

■ `Item`: شیء `DataRowView` را برمی‌گرداند که با اندیس به وسیله پارامتر به آن ارسال شده است.

■ `RowFilter`: عبارتی که برای فیلتر کردن سطرها درون `DataGridView` استفاده می‌شود.

■ `Sort`: عبارتی که برای مرتب کردن سطرها درون `DataGridView` بکار می‌رود.

■ `Table`: شیء `DataTable` متناظر با `DataGridView` را مشخص می‌کند.

برای استفاده از ویژگی‌ها `RowFilter` و `Sort` عبارت‌هایی هستند که مشابه عبارت‌های محاسباتی برای متد `Select` از شیء `DataTable` مورد استفاده قرار می‌گیرند. برای تنظیم مقادیر ویژگی `Sort` که مربوط به روش مرتب‌سازی سطرها `DataRow` است، لازم است مقادیر ستون‌های مورد نظر جهت مرتب‌سازی را به ترتیب در داخل علامت " قرار دهیم. برای مثال برای مرتب‌سازی یک `DataGridView` به ترتیب بر حسب ستون‌های `Date` و `Author` به روش زیر عمل می‌کنیم:

```
myDataGridView.Sort = "CIName";
```

برای تنظیم مقدار RowFilter می‌توان از عبارات DataColumn استفاده کرد. این عبارات شامل یک سری توابع، عملگرها، عبارات و نام ستون‌ها است که جهت فیلترکردن مورد استفاده قرار می‌گیرند در این عبارات می‌توان از تمام عملیات دستکاری روی رشته‌ها و متغیرهای رشته‌ای استفاده کرد. به علاوه برای استفاده از مقادیر تاریخ باید آن‌ها را در بین علامات # # قرار داد.

برای استفاده از یک ستون خاص در این عبارات تنها کافیست، از نام آن ستون استفاده کنید. برای استفاده از سطرهای فرزند از عبارت Child قبل از آن‌ها و برای استفاده از سطر مادر نیز از عبارت Parent قبل از آن‌ها استفاده کنید.

بعضی از توابع قابل استفاده در این عبارات عبارتند از: Sum و Avg و Min و Max و Count و بعضی از عملگرهای منطقی مورد استفاده عبارتند از: And و Or و Not و = و < و In و Like.

### برنامه جستجوی واژه (کارگاه ۶) با استفاده از DataView

ابتدا نمونه‌ای از کلاس DataView در بخش کلاس فرم ایجاد کنید.

```
public partial class Form1 : Form
{
```

```
    DataView myDataView=new DataView();
```

در رویداد Form\_Load کدهای زیر را برای مقیدسازی کنترل کادرفهرست به DataView بنویسید.

```
private void Form1_Load(object sender, EventArgs e)
{
```

```
    this.computertblTableAdapter.Fill(this.dictionaryDataSet.computertbl);
    myDataView.Table = dictionaryDataSet.computertbl;
    listBoxName.DataSource = myDataView;
}
```

برای جستجوی واژه می‌توان فقط از نمونه کلاس DataView استفاده کرد.

```
private void textBoxSearch_TextChanged(object sender, EventArgs e)
{

    myDataView.RowFilter = "";
    if (txtSearchWord.Text.Trim().Length > 0)
        myDataView.RowFilter = "CIName Like '" + txtSearchWord.Text +
        "'";
    listBoxName.DataSource = myDataView;
    textBoxMeans.DataBindings.Clear();
    textBoxMeans.DataBindings.Add("Text", myDataView, "means");
}
```

## پاسخ به فعالیت‌ها

روی چه کنترل‌هایی می‌توان کنترل SplitContainer را قرار داد؟  
پاسخ: روی کنترل‌هایی که ظرف (container) هستند مثل groupBox و Panel

کنجکاوی  
ص ۲۶۳

عملکرد دستور SELECT چیست؟  
پاسخ: انتخاب رکوردهایی که نام واژه آن برابر با ورودی باشد.

کنجکاوی  
ص ۲۶۲

مرحله ۸: عملکرد دستور if چیست؟ در صورتی جستجو انجام می‌شود که در کادر متن txtSearchWord متنی نوشته شده باشد.

نوع ویژگی DataSource کنترل computertblBindingSource چیست؟  
پاسخ: DataSet

کنجکاوی  
ص ۲۶۴

مرحله ۱۰: عملکرد دستور Like چیست؟  
جستجوی بر اساس شباهت ابتدای واژه صورت می‌گیرد.

قابلیت جستجو براساس فیلد نام و کد کالا را برای فرم کالا ایجاد کنید.  
پاسخ: ابتدا متدی به نام FillByGoodsCode با پرس‌وجوی زیر برای جستجو بر اساس کد کالا ایجاد می‌کنیم:

```
SELECT GoodsCode, GoodsName, UnitPrice, Stock FROM  
Goods  
WHERE (GoodsCode = ?)
```

سپس متدی به نام FillByGoodsName با پرس‌وجوی زیر برای جستجوی مشابه بر اساس نام کالا ایجاد می‌کنیم:

```
SELECT GoodsCode, GoodsName, UnitPrice, Stock FROM  
Goods  
WHERE (GoodsName like ? + '%')
```

در فرم جستجو، نام کادر متن مربوط به کد کالا txtSGoodsCode و نام کادر متن مربوط به نام کالا txtSGoodsName است.

```
private void txtSGoodsCode_TextChanged(object sender,  
EventArgs e)  
{  
    if (txtSGoodsCode.Text.Length > 0)  
        goodsTableAdapter.FillByGoodsCode(myShopDataSet.Goods,  
            int.Parse(txtSGoodsCode.Text));  
    else
```

فعالیت منزل  
ص ۲۶۴

```

        goodsTableAdapter.Fill(myShopDataSet.Goods);
    }
    private void txtSGoodsName_TextChanged(object sender, EventArgs
e)
    {
        if (txtSGoodsName.Text.Length > 0)
            goodsTableAdapter.FillByGoodsName(myShopDataSet.Goods,
txtSGoodsName.Text);
        else
            goodsTableAdapter.Fill(myShopDataSet.Goods);
    }

```

#### فعالیت منزل ص ۲۶۵

قابلیت جستجو براساس نام و شماره همراه را برای جدول مخاطبان ایجاد کنید. با دابل کلیک روی PhoneNotebookDataSet.xsd یک متد برای ContactsTableAdapter با نام FillByContactFullName با پرسوجوی زیر ایجاد کنید.

پاسخ:

```

SELECT      MobileNumber, ContactFName, ContactLName,
Oragnization, Email, ContactPicture, CountryCode FROM
Contacts
WHERE      (ContactFName LIKE '%' + ? + '%') AND
(ContactLName LIKE '%' + ? + '%')

```

دو کنترل txtFname و txtLname به فرم اضافه کنید و در رویداد TextChanged کد زیر را بنویسید.

```

contactsTableAdapter.FillByContactFullName(
    phoneNotebookDataSet.Contacts,          txtFname.Text,
txtLname.Text);

```

## کارگاه ۷ - جستجوی ترکیبی در جدول

هدف از این کارگاه جستجو در جدول بر اساس چند فیلد است که در این کارگاه جستجو بر اساس دو فیلد نام کاربری و گذرواژه انجام شده است. با استفاده از دکمه Query Builder به هنجرو نشان دهید که به آسانی می‌توان یک پرسوجوی انتخابی با ترکیب چند شرط ایجاد کرد. ویژگی Count که تعداد رکوردهای جدول DataSet را نمایش می‌دهد، با اجرای گام به گام به هنجرو توضیح دهید.

### پاسخ به فعالیت‌ها

#### تکمیل کارگاه ص ۲۶۶

مرحله ۵: مفهوم شرط دستور if چیست؟  
پاسخ: اگر نتیجه اجرای پرسوجو حداقل یک رکورد داشته باشد، به این معنا است که کاربر معتبر و فرم MainForm نمایش داده می‌شود و گرنه پیام نامعتبر بودن کاربر

نمایش داده می‌شود.

برنامه را اجرا کنید. عملکرد متد `btnLogon_Click` چیست؟

پاسخ: متد `FillByUsernameAndPass` پرس‌وجوی انتخابی با شرط نام کاربری و گذرواژه است و اگر تعداد سطرهای حاصل از اجرای پرس‌وجو بیش از صفر باشد کاربر معتبر و گرنه نامعتبر است.

فعالیت منزل  
ص ۲۶۶

برنامه ورود به فروشگاه را طوری تغییر دهید که با زدن تیک کادر علامت، نام کاربری برای ورود بعدی ذخیره شود و تصویر کاربر نیز نمایش داده شود. پاسخ: متد زیر را پس از ورود به فرم اصلی فراخوانی کنید.

```
private void SaveUN()
{
    if (checkBoxSaveUN.Checked)
        File.WriteAllText("SaveUN.txt", txtUsername.Text);
    else
        if (File.Exists("SaveUN.txt"))
            File.Delete("SaveUN.txt");
}
```

در رویداد `Form_Load` بررسی کنید اگر فایل ذخیره کاربری موجود است نام کاربر در کادر متن نام کاربری نمایش داده شود و تصویر کاربر در کادر تصویر نمایش داده شود.

یک متد با نام `FillByUserName` ایجاد کنید که جستجو را فقط بر اساس نام کاربری انجام دهد.

```
private void Form1_Load(object sender, EventArgs e)
{
    if (File.Exists("SaveUN.txt"))
    {
        checkBoxSaveUN.Checked = true;
        txtUsername.Text = File.ReadAllText("SaveUN.txt");
        usersTableAdapter.FillByUserName(
            myShopDataSet.Users, txtUsername.Text.Trim());
    }
    string strFilename =
        myShopDataSet.Users.Rows[0]["PicturePath"].ToString();
    if (File.Exists(strFilename))
        picUser.ImageLocation = strFilename;
}
```

نکته: در جدول کاربران می‌تواند چندین کاربر وجود داشته باشد. این برنامه تمرینی برای یک برنامه در شبکه است که نام کاربری هر فرد روی سیستم خودش ذخیره شود و به راحتی بدون ورود نام کاربری به برنامه چندکاربره وارد شود.

## پژوهش

ص ۲۶۶

- با بازکردن جدول کاربران در پایگاه داده هر فردی می تواند نام کاربری و گذرواژه همه کاربران را ببیند و به به برنامه فروشگاه وارد شود. برای امنیت برنامه از روش های رمزگذاری (Encoding) گذرواژه استفاده می کنند. درمورد این روش ها پژوهش کنید.

پاسخ: یکی از مباحث مهم در امنیت داده ها استفاده از روش های رمزگذاری (Encoding) است. رمزگذاری چیست؟ رمزگذاری یا درهم ریزی داده ها (Hashing) روشی برای غیرقابل خواندن کردن پرونده ها و نوشته ها است که معمولاً از یک زوج الگوریتم تشکیل می شود یکی برای رمزگذاری و دیگری برای رمزگشایی (Decoding). به طور مثال در یک حلقه for تک تک کاراکترها را با ۲۵۶ xor کرده و جواب را می نویسد و بخش رمزگشایی را با الگوریتم دوم که قرینه الگوریتم اول هست به حالت اول برمی گردانیم. البته استفاده از XOR به مثال ساده است و برای مسائل مهم کسی از این روش استفاده نمی کند یکی از الگوریتم های معروف در زمینه کد کردن الگوریتم base64 و MD5 است.

## الگوریتم MD5 (منبع MSDN مایکروسافت):

```
static string GetMd5Hash(MD5 md5Hash, string input)
{
    // Convert the input string to a byte array and compute the
    // hash.
    byte[] data =
    md5Hash.ComputeHash(Encoding.UTF8.GetBytes(input));
    // Create a new StringBuilder to collect the bytes
    // and create a string.
    StringBuilder sBuilder = new StringBuilder();
    // Loop through each byte of the hashed data
    // and format each one as a hexadecimal string.
    for (int i = 0; i < data.Length; i++)
    {
        sBuilder.Append(data[i].ToString("x2"));
    }
    // Return the hexadecimal string.
    return sBuilder.ToString();
}
```

متد GetMd5Hash ورودی input را با md5Hash کدگذاری و در یک رشته sBuilder قرار می دهد.

```
// Verify a hash against a string.
static bool VerifyMd5Hash(MD5 md5Hash, string input, string hash)
{
    // Hash the input.
    string hashOfInput = GetMd5Hash(md5Hash, input);
    // Create a StringComparer and compare the hashes.
```



```
StringComparer comparer = StringComparer.OrdinalIgnoreCase;
if (0 == comparer.Compare(hashOfInput, hash))
{
    return true;
}
else
{
    return false;
}
}
```

متد VerifyMd5Hash بررسی می‌کند که اگر ورودی با input با md5Hash کد شود، آیا رشته کدگذاری شده hash معادل است یا خیر.

### الگوریتم base64 :

رمزگذاری یک رشته:

```
private static string Base64Encode(string plainText)
{
    var plainTextBytes =
    System.Text.Encoding.UTF8.GetBytes(plainText);
    return System.Convert.ToBase64String(plainTextBytes);
}
```

رمزگشایی یک رشته کدگذاری شده:

```
private static string Base64Decode(string base64EncodedData)
{
    var base64EncodedBytes =
    System.Convert.FromBase64String(base64EncodedData);
    return System.Text.Encoding.UTF8.GetString(base64EncodedBytes);
}
```

ابتدا گذرواژه را با متد Base64Encode رمزگذاری کرده، نتیجه را در فیلد گذرواژه در جدول کاربران قرار دهید و متد جستجو را به صورت زیر فراخوانی کنید.

```
usersTableAdapter.FillByUsernameAndPass(myShopDataSet.Users,
txtUsername.Text, Base64Encode(txtPassword.Text));
```

## کارگاه ۸ – درج یک رکورد در جدول

هدف از این کارگاه نحوه ایجاد متد درج در جدول و فراخوانی آن در برنامه و آشنایی با دستور INSERT INTO است. از هنرجو بخواهید ساختار دستور INSERT INTO را در پودمان اول فراگرفته توضیح دهد. سپس از او بخواهید متدی را در TableAdaptor جدول کالا در پرونده xsd ایجاد کند که دستور درج رکورد را انجام دهد. نحوه فراخوانی متد پرس و جوی عملیاتی به صورت زیر به هنرجو آموزش دهید.

(ورودی‌های متد) نام متد • نام TableAdaptor

## پاسخ به فعالیت‌ها

کنجکاوی  
ص ۲۶۷

ساختار دستور INSERT INTO در زبان SQL چیست؟  
پاسخ: مقدار VALUES .... , نام فیلد ۲, نام فیلد ۱ نام جدول INSERT INTO  
..... مقدار فیلد ۲, فیلد ۱

مرحله ۲:

- متد InsertQuery دارای چند آرگومان ورودی است؟ ۴ آرگومان.
- چرا برخی از ورودی‌ها عددی است؟ چون نوع فیلد در جدول عددی است.
- چرا از دستور if در برنامه استفاده شده است؟ چون ممکن است کاربر برخی از ورودی‌هایی که لازم است مقدار داشته باشند را خالی گذاشته باشد.
- عملکرد متد رویداد btnAddGoods\_Click چیست؟ در صورت ورود نام، قیمت و کد کالا یک کالای جدید به جدول Goods اضافه می‌کند.
- عملکرد دستور goodsTableAdapter.InsertQuery چیست؟ متدی برای اضافه کردن رکورد جدید به جدول Goods است.
- آیا رکورد جدید در goodsDataGridView نمایش داده می‌شود؟ چرا؟ بله. چون بعد از اضافه شدن رکورد در جدول متد Fill نمایش داده‌ها را در goodsDataGridView بروز می‌کند.
- آیا رکورد اضافه شده در goodsDataGridView وجود دارد؟ خیر.
- جدول Goods پایگاه‌داده MyShop را باز کنید. آیا رکورد اضافه شده در جدول وجود دارد؟ خیر

مرحله ۳:

- آیا در جدول Goods این کالا اضافه شده است؟ بله

فعالیت منزل  
ص ۲۶۸

- برای جدول مشتری برنامه درج رکورد را بنویسید. (فیلم)
  - برای جدول مخاطبان برنامه ثبت مخاطب جدید را بنویسید.
- پاسخ: با دابل کلیک روی PhoneNotebookDataSet.xsd در بخش Solution Explorer یک متد برای اضافه کردن مخاطب جدید ایجاد کنید. روی جدول Contacts راست‌کلیک کنید. گزینه Query از منوی Add را انتخاب کنید و سپس گزینه Use SQL statement را انتخاب و دکمه Next را کلیک کنید، سپس گزینه INSERT را انتخاب و دکمه Next را کلیک کنید. فرم زیر را برای اضافه کردن مخاطب جدید ایجاد کنید.

برای دکمه بارگذاری تصویر کدهای زیر را بنویسید.

```
private void btnLoadPicture_Click(object sender, EventArgs e)
{
    if (!IsValidMobileNO(txtMobileNO.Text))
    {
        MessageBox.Show("شماره موبایل معتبر نیست");
        return;
    }
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        string strFileName = openFileDialog1.FileName;
        picContact.ImageLocation = strFileName;
        fileName = txtMobileNO.Text +
            System.IO.Path.GetExtension(strFileName);
    }
}
```

نکته: متغیر fileName در بخش کلاس فرم تعریف شود.

```
private bool IsValidMobileNO(string strMobileNO)
{
    long mobile;
    if (!long.TryParse(strMobileNO, out mobile))
        return false;
    if (strMobileNO[0] != '0') strMobileNO = "0" + strMobileNO;
    if (strMobileNO.Length == 11)
        return true;
    return false;
}
```

به فرم ثبت مخاطب یک کنترل DataGridView اضافه کنید و آن را مقید کنید تا کنترل‌های مورد نیاز به فرم اضافه شود.

با کلیک روی دکمه btnAddContact مخاطب جدید به فرم اضافه می‌شود.

```
private void btnAddContact_Click(object sender, EventArgs e)
{
```

```

if (isValidMobileNO(txtMobileNO.Text) && txtFname.TextLength >
0 &&
txtLname.TextLength > 0)
{
    if (txtMobileNO.TextLength == 10)
        txtMobileNO.Text = "0" + txtMobileNO.Text;
    contactsTableAdapter.InsertQuery(txtMobileNO.Text,
txtCountyCode.Text,txtFname.Text,
txtLname.Text,
txtOrg.Text,
txtEmail.Text,fileName);
    string appPath = Application.StartupPath + "\\\" + fileName;
    //کپی تصویر در مسیر برنامه
    System.IO.File.Copy(openFileDialog1.FileName, appPath);
    MessageBox.Show("ثبت شد");
}
else
    MessageBox.Show("مخاطب ثبت نشد");
}

```

## کارگاه ۹ – حذف یک رکورد در جدول

هدف از این کارگاه نحوه ایجاد متد حذف در جدول و فراخوانی آن در برنامه و آشنایی با دستور DELETE است. از هنرجو بخواهید ساختار دستور DELETE را در پودمان اول فراگرفته توضیح دهد. سپس از او بخواهید متدی را در TableAdaptor جدول کالا در پرونده xsd ایجاد کند که دستور حذف رکورد را انجام دهد.

### پاسخ به فعالیت‌ها

تکمیل کارگاه  
ص ۲۶۹

- ۱) دستور را تغییر دهید تا حذف بر اساس نام کالا باشد.  
DELETE FROM Goods WHERE (GoodsName = ?)
- ۲) دستور را برای حذف تمامی رکوردها تغییر دهید.  
DELETE FROM Goods

فعالیت منزل  
ص ۲۶۹

برای جدول مشتری برنامه حذف رکورد را بنویسید. (فیلم)  
برای جدول مخاطبان برنامه حذف مخاطب را بنویسید.  
پاسخ: با دابل کلیک روی PhoneNotebookDataSet.xsd در بخش Solution Explorer یک متد برای حذف کردن مخاطب ایجاد کنید. روی جدول Contacts راست کلیک کنید. گزینه Query از منوی Add را انتخاب کنید و سپس گزینه Use SQL statement را انتخاب و دکمه Next را کلیک کنید، سپس گزینه Delete را

انتخاب و دکمه Next را کلیک کنید. سپس پرس‌وجوی زیر را با Query Builder ایجاد کنید.

```
DELETE FROM Contacts WHERE (MobileNumber = ?)
AND (CountryCode = ?)
```

در فرم مدیریت مخاطب یک دکمه به نام btnDeleteContact اضافه کنید و کد زیر را بنویسید.

```
private void btnDeleteContact_Click(object sender, EventArgs e)
{
    if (txtCountryCode.TextLength > 0 &&
        txtMobileNO.TextLength>0)
        contactsTableAdapter.DeleteQuery(txtMobileNO.Text,
            txtCountryCode.Text);
}
```

## کارگاه ۱۰ – ویرایش یک رکورد در جدول

هدف از این کارگاه نحوه ایجاد متد ویرایش در جدول و فراخوانی آن در برنامه و آشنایی با دستور UPDATE است. از هنجرو بخواهید ساختار دستور UPDATE را در پودمان اول فراگرفته توضیح دهد. سپس از او بخواهید متدی را در TableAdaptor جدول کالا در پرونده xsd ایجاد کند که دستور ویرایش رکورد را انجام دهد.

پاسخ به فعالیت‌ها

برنامه درج کالای جدید را طوری تغییر دهید که اگر کالای جدید در جدول موجود بود، فقط موجودی کالا را ویرایش کند(موجودی کالا را افزایش دهد).

پاسخ: یک متد برای ویرایش موجودی کالا با نام UpdateGoodsStock با پرس‌وجو زیر ایجاد کنید:

```
UPDATE Goods SET Stock = ? WHERE (GoodsCode = ?)
```

ابتدای متد رویداد کلیک دکمه ثبت کالا، کدهای زیر را اضافه کنید.

```
goodsTableAdapter.FillByGoodsCode(myShopDataSet.Goods,
    int.Parse(txtGoodsCode.Text));
if(myShopDataSet.Goods.Rows.Count>0)
{
    int oldStock =
        int.Parse(myShopDataSet.Goods.Rows[0]["Stock"].ToString());
    int newStock = oldStock + int.Parse(txtStock.Text);
    goodsTableAdapter.UpdateGoodsStock(newStock,
```

فعالیت منزل  
ص ۲۷۰

```

int.Parse(txtGoodsCode.Text));
goodsTableAdapter.Fill(myShopDataSet.Goods);
MessageBox.Show("با موفقیت ثبت شد");
btnClearGoods_Click(sender, e);
return;
}

```

### فعالیت منزل ص ۲۷۱

برای جدول مشتری برنامه ویرایش رکورد را بنویسید. (فیلم را ملاحظه کنید).  
برای جدول مخاطبان برنامه ویرایش مخاطب را بنویسید.  
پاسخ: با دابل کلیک روی PhoneNotebookDataSet.xsd در بخش Solution Explorer یک متد برای ویرایش مخاطب ایجاد کنید. روی جدول Contacts راست کلیک کنید. گزینه Query از منوی Add را انتخاب کنید و سپس گزینه Use SQL statement را انتخاب و دکمه Next را کلیک کنید، سپس گزینه Update را انتخاب و دکمه Next را کلیک کنید. سپس پرس و جوی زیر را با Query Builder ایجاد کنید.

```

UPDATE Contacts
SET      MobileNumber = ?, CountryCode = ?, ContactFName
= ?, ContactLname = ?, Oragnization = ?, Email = ?,
ContactPicture = ?

```

```

WHERE   (MobileNumber = ?) AND (CountryCode = ?)

```

در فرم مدیریت مخاطب یک دکمه به نام btnUpdateContact اضافه کنید و کد زیر را بنویسید.

```

private void btnUpdateContact_Click(object sender, EventArgs e)
{
    if (IsValidMobileNO(txtMobileNO.Text) && txtFname.TextLength > 0 &&
        txtLname.TextLength > 0)
    {
        contactsTableAdapter.FillByMobileNoAndCountryCode(
            phoneNotebookDataSet.Contacts, txtMobileNO.Text,
            txtCountyCode.Text);
        if (fileName.Trim().Length == 0)
            fileName = phoneNotebookDataSet.Contacts.Rows[0]
                ["ContactPicture"].ToString();
        contactsTableAdapter.UpdateQuery(txtMobileNO.Text,
            txtCountyCode.Text, txtFname.Text, txtLname.Text, txtOrg.Text,
            txtEmail.Text, fileName, mobileNO, countryCode);
    }
}

```

دو متغیر mobileNO و countryCode در بخش کلاس فرم تعریف می شوند و مقدار خود را از فرم اصلی دریافت می کنند.

## کارگاه ۱۱ – درج و حذف سطر از DataGridView

هدف از این کارگاه نحوه اضافه و حذف کردن سطر از ی ک کنترل DataGridView نامقید است. با استفاده از متد Add و مفهوم Row و Cell کنترل DataGridView نحوه افزودن سطر جدید را به هنرجو آموزش دهید. متد RemoveAt یکی از متدهای مشترک بین کادرفهرست و کنترل DataGridView است که می‌توانید از این وجه تشابه، متد RemoveAt را آموزش دهید. برای پیمایش سطر به سطر کنترل DataGridView می‌توان از حلقه for استفاده کرد.

### پاسخ به فعالیت‌ها

#### تکمیل کارگاه

ص ۲۷۱

مرحله ۲: ویژگی‌های ImageLayout و FillWeight را به دلخواه تغییر دهید. کاربرد این ویژگی‌ها چیست؟

ویژگی ImageLayout نحوه قرارگیری تصویر در ستونی است که کنترل کادرتصویر قرارداد که در این کارگاه Stretch قرار داده می‌شود.

ویژگی FillWeight درصد قرارگیری عرض ستون است هنگامی که ویژگی AutoSizeMode برابر با Fill قرار داده می‌شود.

۸- متد رویداد کلیک دکمه جستجوی مشتری (btnCustomerTab) در سربرگ برگه خرید را بنویسید.

فعال کردن سربرگ مشتری // `tabControl1.SelectedTab = tabPage3;`

`txtSCustomerFamily.Focus();`

۹- متد رویداد کلیک دکمه «درج در برگه خرید» در سربرگ مشتری را بنویسید.

```
private void btnCustomerInsertFactor_Click(object sender, EventArgs e)
{
    lblCustomerName.Text = txtCustomerName.Text + " " +
    txtCustomerFamily.Text;
    txtFCustomerCode.Text = txtFCustomerCode.Text;
    tabControl1.SelectedTab = tabPage1; // فعال کردن سربرگ برگه خرید
    txtFCustomerCode.Text = txtCustomerCode.Text;
    lblCustomerName.Text = txtCustomerName.Text + " " +
    txtCustomerFamily.Text;
    txtFCustomerCode.Focus();
}
```

مرحله ۱۰:

■ چگونه می‌توان تعداد سطرها را یک DataGridView تعیین کرد؟

با ویژگی `FactorDataGridView.Rows.Count` می‌توان تعداد سطرها را مشخص کرد. چون یک سطر خالی دارد از یک کم می‌شود.

■ عملکرد متد Add کنترل FactorDataGridView چیست؟ یک سطر خالی ایجاد می‌کند.

■ قیمت کل در کدام خانه FactorDataGridView قرار می‌گیرد و چگونه محاسبه می‌شود؟ ستون قیمت کل حاصل ضرب تعداد کالا در قیمت کالا است که در ستون چهارم قرار می‌گیرد.

۱۱- کد مرحله ۶ را تغییر دهید تا در صورت خالی بودن کادر متن کد کالا خطا رخ ندهد.

قبل از اضافه کردن به سبد خرید دستورات زیر را بنویسید.

```
if (txtFGoodsCode.TextLength == 0 || txtFAmount.TextLength==0)
{
    MessageBox.Show("کنید وارد را تعداد یا کالا کد");
    txtFGoodsCode.Focus();
    return;
}
```

۱۲- به متد رویداد کلیک دکمه «درج در سبد خرید» محاسبه مبلغ قابل پرداخت را اضافه کنید.

```
int sum = 0;
for (int i = 0; i < n+1; i++)
{
    int price=(int)FactorDataGridView.Rows[i].Cells[4].Value;
    sum += price;
}
lblAllCost.Text = sum.ToString();
```

## کارگاه ۱۲ - کنترل خطا در برنامه پایگاه داده

هدف از این کارگاه آموزش کنترل خطا در برنامه‌های پایگاه داده با استفاده بلاک try-catch-finally است. ابتدا از هنرجو بخواهید در مورد این بلاک توضیحی ارائه کند و یک برنامه نمونه داده‌ای را وارد کند که خطای زمان اجرا رخ دهد. برای فهرست خطاهای احتمالی هر کدام یک مثال در برنامه پایگاه داده بزنید.

پاسخ به فعالیت‌ها

مرحله ۲: با درج کد تکراری چه اتفاقی رخ می‌دهد؟ خطا رخ می‌دهد.

تکمیل کارگاه  
ص ۲۷۵

نام عمل	فهرست برخی از خطاهای ممکن
درج	خطای ناشی از ساختار نادرست دستور درج- خطای ناشی از خالی ماندن فیلدی که نباید Null باشد- خطای ناشی از ورودی غیرهم نوع-خطای



ناشی از ورود رکورد تکراری	
حذف	خطای ناشی از ساختار نادرست دستور حذف- خطای ناشی از حذف رکوردی در جدول پدر که در ارتباط با جدول فرزند است.
ویرایش	خطای ناشی از ساختار نادرست دستور ویرایش- خطای ناشی از خالی ماندن فیلدی که نباید Null باشد- خطای ناشی از ورودی غیرهم نوع- خطای ناشی از ورود رکوردی که کلید اصلی تکراری دارد.

مرحله ۳: کالایی را ثبت کنید که کد کالای آن تکراری باشد، چه اتفاقی رخ می‌دهد؟ پیام «خطا در ثبت کالا» نمایش داده می‌شود.

مرحله ۴: آیا دستور if در برنامه عمل کنترل خطا را انجام می‌دهد؟ بله اگر کد کالا غیر عددی نوشته شود، آیا خطایی رخ می‌دهد؟ پیام «خطا در ثبت کالا» نمایش داده می‌شود.

## کارگاه ۱۳ - توسعه پروژه فروشگاه

هدف از این کارگاه ذخیره برگه خرید در برنامه فروشگاه است. ابتدا از هنرجو بخواهید توضیح دهد با ذخیره برگه خرید چه اتفاقی در پایگاه داده MyShop رخ می‌دهد. سپس با نمایش فیلم «ذخیره برگه خرید» این برنامه را در کلاس تکمیل کنید.

### نکات تکمیلی پروژه فروشگاه:

اگر موجودی کالایی برای درج در سبد خرید کافی نباشد برنامه به کاربر اعلام کند. یک کنترل برچسب به نام lblStock به فرم برگه خرید اضافه کنید و هم زمان با مقدار دهی برچسب نام کالا و قیمت کالا این برچسب را مقداردهی کنید. کد زیر در txtFGoodsCode\_KeyDown اضافه کنید.

```
lblStock.Text = myShopDataSet.Goods.Rows[0]["Stock"].ToString();
و کد زیر را در «درج در برگه خرید» سربرگ کالا اضافه کنید.
lblStock.Text = txtStock.Text;
و کدهای زیر را در دکمه «درج در سبد خرید» اضافه کنید.
if (txtFGoodsCode.TextLength == 0 || txtFAmount.TextLength==0)
{
    MessageBox.Show("کد کالا یا تعداد کالا را وارد کنید");
    txtFGoodsCode.Focus();
    return;
}
if (int.Parse(txtFAmount.Text) > int.Parse(lblStock.Text))
{
    MessageBox.Show("موجودی کافی نیست");
    txtFAmount.Focus();
}
```

```
return;
}
```

بعد از ثبت برگه خرید باید موجودی کالاهایی که خریداری شده نیز کم شود و اطلاعات برگه خرید جاری حذف شود و شماره برگه جدید ارائه شود. متد SetFactorNumber یک واحد به شماره برگه خرید اضافه می‌کند.

```
private void SetFactorNumber()
{
    factorTableAdapter.Fill(myShopDataSet.Factor);
    lblFactorNO.Text = (myShopDataSet.Factor.Rows.Count +
1).ToString();
}
```

### برنامه کامل عملیات «ثبت برگه خرید»:

با کلیک روی دکمه «ثبت برگه خرید» عملیات زیر انجام می‌شود.

```
private void btnSaveFactor_Click(object sender, EventArgs e)
{
    if (FactorDataGridView.Rows.Count-1 > 0)
    {
        int numofRecord = FactorDataGridView.Rows.Count;
        int factorNo =int.Parse(lblFactorNO.Text);
        DateTime fDate = DateTime.Now;
        int customerCode=int.Parse(txtFCustomerCode.Text);
        try
        {
            factorTableAdapter.InsertQuery(factorNo, customerCode,fDate);
            for (int i = 0; i < numofRecord - 1; i++)
            {
                int goodsCode =
                int.Parse(FactorDataGridView.Rows[i].Cells[0].Value.
ToString());
                int amount=
                int.Parse(FactorDataGridView.Rows[i].Cells[2].Value.
ToString());
                registerFactorTableAdapter.InsertQuery(factorNo, goodsCode,
amount);
                goodsTableAdapter.FillByGoodsCode(myShopDataSet.Goods,goodsCode);
                int oldStock =
                int.Parse(myShopDataSet.Goods.Rows[0]["Stock"].ToString());
                int newStock = oldStock - amount;
                //بروز رسانی موجودی کالا/
                goodsTableAdapter.UpdateGoodsStock(newStock, goodsCode);
            }
            FactorDataGridView.Rows.Clear();
            MessageBox.Show("برگه خرید ثبت شد");
            SetFactorNumber(); // شماره جدید برگه خرید
        }
        catch(Exception ex)
```

```

{
    MessageBox.Show("خطا در ثبت برگه خرید: "+ex.Message);
}
}
}

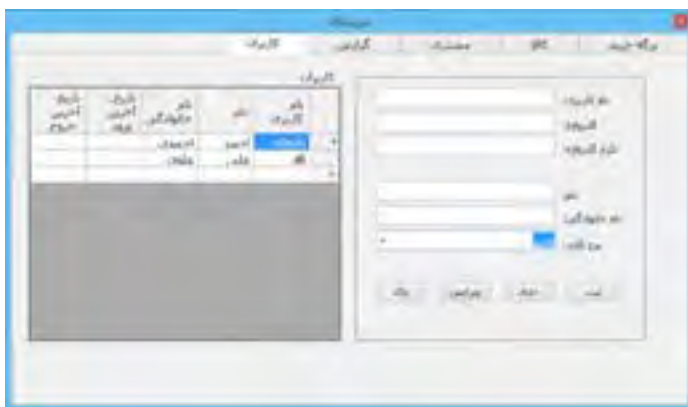
```

نکته : برای عملکرد بهتر برنامه باید کنترل خطا برای همه عملیات برنامه فروشگاه انجام شود.

### پاسخ به فعالیت‌ها

سربرگ جدیدی به برنامه فروشگاه به نام کاربران اضافه کنید که فقط مدیر فروشگاه دسترسی به آن داشته باشد و عملیات ایجاد، حذف و ویرایش کاربر را انجام دهد.

پاسخ: فیلد `typeofAccess` در جدول `Users` در پایگاه داده `MyShop` نوع دسترسی کاربران را مشخص می‌کند اگر کاربر مدیر فروشگاه باشد مقدار این فیلد ۱ یک در نظر گرفته می‌شود. یک سربرگ جدید «کاربران» به نام `tabPageUsers` با طراحی زیر ایجاد کنید.



متغیر **access** را بخش کلاس فرم `MainForm` به صورت عمومی (`Public`) اعلان کنید و قبل از نمایش فرم اصلی فروشگاه (`MainForm`) به صورت زیر مقداردهی کنید.

```

MainForm frm = new MainForm();
frm.access =
byte.Parse(myShopDataSet.Users.Rows[0]["typeofAccess"].ToString())
);
frm.Show();

```

مقدار `access` برابر نوع دسترسی کاربری است که وارد فرم فروشگاه می‌شود. کد زیر را برای نمایش یا عدم نمایش سربرگ «کاربران» در انتهای رویداد `MainForm_Load` بنویسید.

```

if (access != 1)

```

فعالیت منزل  
ص ۲۷۶

```
tabControl11.TabPages.Remove(tabPageUsers);
```

برنامه انتخابات شورای دانش‌آموزی را بنویسید. معاون پرورشی مدرسه داده‌های مربوط به دانش‌آموزان و نامزدها را وارد می‌کند. هر دانش‌آموز با نام کاربری و گذرواژه وارد فرم اخذ رأی می‌شود و به نامزدهای مورد علاقه خود رأی می‌دهد، سپس حق رأی از او گرفته می‌شود. در پایان انتخابات معاون پرورشی می‌تواند نتایج انتخابات را به ترتیب تعداد رأی مشاهده کند.

اگر بخواهیم طراحی ساده‌ای از پایگاه داده داشته باشیم سه جدول «مدرسه» و «دانش‌آموزان» و «نامزدها» را در پایگاه داده StudentsElection ایجاد می‌کنیم.

Field Name	Data Type	
SchoolName	Short Text	نام مدرسه
numOfMembers	Number	تعداد اعضای اصلی شورای دانش‌آموزی
ManagerUsername	Short Text	نام کاربری مدیر
ManagerPass	Short Text	گذرواژه مدیر

Field Name	Data Type	
studentUsername	AutoNumber	کد کاربری دانش‌آموز
Password	Number	گذرواژه
studentFname	Short Text	نام دانش‌آموز
studentLname	Short Text	نام خانوادگی
isElect	Yes/No	آیا دانش‌آموز برای داده یا خیر

Field Name	Data Type	
CandidCode	Number	کد نامزد
CandidFullName	Short Text	نام نامزد
NumOfElect	Number	تعداد رأی

یک پروژه ویندوزی به نام StudentsElection ایجاد کنید. همانند برنامه فروشگاه یک فرم ورود کاربر و یک فرم اصلی با کنترل TabControl ایجاد کنید و عملیات درج، حذف و ویرایش و جستجو را انجام دهید. یک فرم را برای اخذ رأی ایجاد کنید.

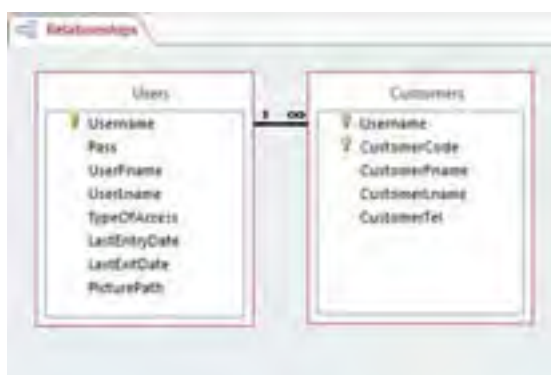
مدیر با ورود به برنامه می‌تواند فرم اصلی برنامه را ببیند و اطلاعات مدرسه را تکمیل

کند و داده‌های مربوط به دانش‌آموزان و نامزدها را وارد کند. دانش‌آموز با ورود به برنامه می‌تواند فرم اخذ رأی را ببیند و رأی خود را با انتخاب نامزدهای مورد علاقه ثبت کند. پس از رأی دادن حق رأی از دانش‌آموز گرفته می‌شود.



عملیات کامل فرم اخذ رأی در فیلم ارائه شده است. برنامه «کتابخانه هنرستان جابرین حیان» را همانند برنامه فروشگاه فرهنگیان انجام دهید. در یک سامانه کاربرانی وجود دارند که هر کدام به مشخصات مشتریان خاص خود دسترسی دارند و می‌توانند با آن‌ها تماس بگیرند. برنامه‌ای بنویسید تا کار ذخیره و بازیابی کاربران به همراه مشتریان را انجام دهد.

پاسخ: این پایگاه‌داده (ManageCustomers) دارای دو جدول Users و Customers است که نمودار ارتباطی آن به صورت زیر است.

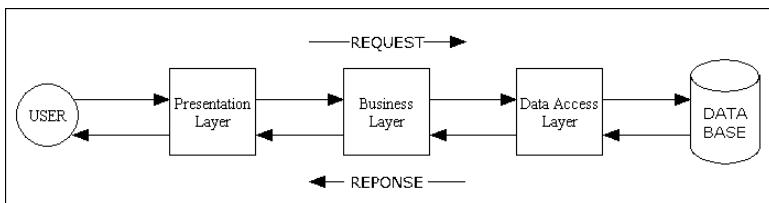


## مطالب تکمیلی

قصد داریم پروژه مدیریت مشتریان (ManageCustomers) را با برنامه‌نویسی سه لایه‌ای انجام دهیم. قبل از انجام این کار مدل سه لایه در معماری نرم‌افزار و تکنولوژی ADO.NET شرح داده می‌شود.

### مدل سه لایه در معماری نرم‌افزار

در مهندسی نرم‌افزار، برای کاهش پیچیدگی و ساده‌تر شدن مراحل طراحی، پیاده‌سازی و نگهداری نرم‌افزارها، سیستم‌های نرم‌افزاری را به چند زیر سیستم تقسیم کرده و قسمت‌های مختلف سیستم را به صورت لایه‌های جداگانه و مستقل از هم طراحی و پیاده‌سازی می‌کنند. این لایه‌ها با یکدیگر در ارتباط هستند و بین آن‌ها انتقال اطلاعات انجام می‌گیرد. هر کدام از این لایه‌ها وظیفه خاص خود را دارند و به لایه‌های بالایی و پایینی خود سرویس داده و از آن‌ها سرویس می‌گیرند. کاربر نهایی فقط با لایه بیرونی در ارتباط است و کاری با لایه‌های دیگر ندارد. به این روش طراحی سیستم‌های نرم‌افزاری، معماری چندلایه یا N-Tier گفته می‌شود.



معماری سه لایه (3-Tier) حالت خاصی از معماری چند لایه است که در آن سیستم‌ها به سه لایه جداگانه تقسیم می‌شوند. این لایه‌ها عبارتند از:

- ۱) **لایه نمایش (Presentation Layer):** این لایه که به آن لایه رابط کاربر (User Interface) نیز گفته می‌شود، عناصری مانند فرم‌ها، کنترل‌های روی فرم‌ها، منوهای برنامه و هر چیزی را دربرمی‌گیرد که کاربر نهایی از سیستم مشاهده می‌کند. کاربر سیستم به طور مستقیم فقط با این لایه در ارتباط است و در واقع درخواست خود را از طریق این لایه به لایه‌های زیرین انتقال می‌دهد. لایه نمایش اطلاعات لازم را از کاربر گرفته، در صورت لزوم برخی اعتبارسنجی‌ها (Validation) مانند کنترل طول فیلدها یا کنترل اجباری بودن بعضی فیلدها را انجام می‌دهد و سپس این اطلاعات را برای هرگونه پردازش دیگری به لایه بعدی ارسال می‌کند.
- ۲) **لایه منطق تجاری (Business Logic Layer):** وظیفه اصلی این لایه که به آن لایه میانی (Middle Tier) نیز گفته می‌شود، اعمال منطق اصلی برنامه روی درخواست کاربران و نیز برقراری ارتباط بین لایه نمایش و لایه داده است.

کلیه درخواست‌هایی که در اثر تعامل کاربر با لایه نمایش ایجاد شده است به این لایه منتقل می‌شود و تمام پردازش‌های لازم بر اساس منطق اصلی برنامه در این لایه انجام می‌شود. نتیجه این پردازش به لایه نمایش منتقل شده، برای کاربر به نمایش درمی‌آید. گاهی اوقات درخواست کاربر به گونه‌ای است که لایه منطق تجاری برای انجام آن نیاز دارد که با لایه داده یعنی لایه زیرین خود نیز ارتباط داشته باشد.

۳) لایه دسترسی به داده (Data Access Layer): این لایه که به آن لایه بانک اطلاعاتی نیز گفته می‌شود، وظیفه مدیریت اطلاعات موجود در بانک اطلاعاتی را بر عهده دارد. این لایه بر اساس درخواست‌هایی که از لایه بالایی خود دریافت می‌کند عملیاتی نظیر حذف، اضافه، اصلاح و خواندن اطلاعات را بر روی بانک اطلاعاتی انجام داده، نتیجه عمل را به لایه بالایی خود ارسال می‌کند. باید توجه داشت که ارتباط با بانک اطلاعاتی فقط از طریق لایه داده انجام می‌گیرد.

از مزایای معماری چند لایه می‌توان به عدم وابستگی لایه‌ها به همدیگر اشاره کرد. برای نمونه لایه داده مستقل از لایه‌های دیگر عمل کرده، در صورت لزوم می‌توان با سرعت بالا و هزینه پایین این لایه را تغییر داد. به عنوان مثال در برنامه‌ای که از بانک اطلاعاتی SQL Server استفاده می‌شود به راحتی می‌توان بانک اطلاعاتی را به Access و یا هر بانک اطلاعاتی دیگر تغییر داد و یا با کمترین هزینه می‌توان منطق اصلی برنامه را با تغییر لایه منطق تجاری تغییر داد. اما معماری سه لایه یا چند لایه همیشه راه حل مناسبی برای طراحی سیستم‌ها نیست. به عبارت دیگر برای طراحی هر سیستمی نمی‌توان از این معماری استفاده کرد.

برای مدیریت پایگاه داده ManageCustomers در برنامه‌نویسی سه لایه به صورت زیر اقدام کنید.

بعد از ایجاد پروژه ویندوزی ManageCustomers از منوی Project، با استفاده از گزینه Add Class یک کلاس ایجاد کنید و نام کلاس را DataAccessLayer قرار دهید. در این کلاس متدهایی برای اتصال و قطع ارتباط با پایگاه داده و انجام همه پرس‌وجوهای انتخابی و عملیاتی و تجمعی تعریف می‌شود.

کلاسی با نام جدول‌ها (بطور مثال برای جدول کاربران کلاس Users و برای جدول Customers کلاس Customers) ایجاد کنید و عملیات (متدهای) جستجو و درج، حذف و ویرایش و... روی جدول مورد نظر را به کمک متدهای کلاس DataAccessLayer تعریف کنید. (Business Logic Layer)

در بخش کلاس فرم نیز با توجه به نیازهای برنامه نمونه‌ای از کلاس‌های موجود را تعریف کنید و عملیات را در فرم ورود و کاربران و مشتریان انجام دهید.

نحوه تعریف و استفاده از کلاس DataAccessLayer و Users و Customers را در فیلم همراه کتاب ببینید.

## تکنولوژی ADO.Net :

در .NET Framework، دستیابی داده با استفاده از ADO.NET انجام می‌شود. ADO.NET در حقیقت یک رابط برنامه‌نویسی است که مجموعه امکانات لازم به منظور برقراری اتصال با پایگاه‌داده را در اختیار پیاده‌کنندگان برنامه‌های ویندوزی یا تحت وب قرار می‌دهد. ADO.NET برای دسترسی به داده‌ها از معماری غیر متصل استفاده می‌کند. در معماری غیر متصل ابتدا برنامه به موتور پایگاه‌داده مورد نظر متصل می‌شود و داده‌های مورد نیاز خود را از پایگاه‌داده می‌خواند و در حافظه کامپیوتر ذخیره می‌کند، سپس اتصال از پایگاه‌داده قطع می‌شود و تغییرات مورد نظر خود را در داده‌های ذخیره شده در حافظه انجام می‌دهد. هر زمان که نیاز باشد تغییرات ایجاد شده در پایگاه‌داده ذخیره شود، برنامه یک اتصال جدید را به پایگاه‌داده ایجاد کرده، از طریق این اتصال، تغییراتی را که در داده‌ها اعمال کرده بود در جدول اصلی ایجاد می‌کند. ADO.Net امکان دسترسی به منابع داده‌ای مختلف مثل پایگاه‌داده SQL Server، فایل XML، پایگاه‌های ODBC و... را فراهم می‌سازد. ADO.Net دسترسی به داده و دستکاری داده را از هم جدا کرده است، که هرکدام را به تنهایی می‌توانید به کار برده یا باهم استفاده کنید. ADO.Net شامل فراهم‌کننده‌های داده (NET Framework Providers) برای اتصال به منابع داده‌ای، اجرای دستورات و بازگرداندن نتایج به درخواست‌کننده است. جدول زیر فهرستی از فراهم‌کننده‌های داده‌ای را نمایش می‌دهد.

نام فراهم کننده داده ای	نشانه پیشوندی	توصیف Data Source	نام فضای نام مورد استفاده
<b>ODBC Data Provider</b>	Odbc	برای منابع داده‌ای به همراه یک واسط ODBC	System.Data.Odbc
<b>OleDb Data Provider</b>	OleDb	منابع داده‌ای که یک واسط OleDb مثل Access یا Excel را ارایه می‌کند	System.Data.OleDb
<b>Oracle Data Provider</b>	Oracle	برای پایگاه‌داده Oracle	System.Data.OracleClient
<b>SQL Server Data Provider</b>	Sql	برای ارتباط با Microsoft Sql Server	System.Data.SqlClient



فراهم‌کننده‌های داده‌ای NET. اجزایی هستند که به صورت خاص برای دستکاری و نقل و انتقال و خواندن اطلاعات طراحی شده‌اند. اشیاء اصلی این فراهم‌کننده‌ها عبارتند از :

■ شیء Connection: شرایط و امکانات لازم برای اتصال به منبع داده‌ای را فراهم می‌کند.

■ شیء Command: این شیء امکان دسترسی به دستورات لازم برای دریافت، ویرایش، درج اطلاعات و... و همچنین اجرای توابع ( Stored Procedure ) SQL و ارسال پارامترها را فراهم می‌کند.

■ شیء SqlDataReader: یک شیء کامل برای خواندن حجم گسترده‌ای از اطلاعات از منابع داده‌ای مختلف.

■ شیء DataAdapter: این شیء یک پل رابط بین شیء DataSet و منبع داده‌ای است.

شیء DataAdapter از شیء Command برای اجرای دستورات مختلف SQL برای دو هدف استفاده می‌کند:

(۱) دریافت اطلاعات مورد نظر از پایگاه‌داده و قراردادن آن‌ها در شیء DataSet.

(۲) دریافت تغییراتی که در اطلاعات شیء DataSet رخ داده و اعمال و بروزرسانی آن‌ها در منبع داده‌ای.

کلاس‌های کار با پایگاه‌داده Access در فضای نام System.Data.OleDb قرار دارند. نام کلاس‌ها عبارتند از:

OleDbConnection و OleDbCommand و OleDbDataAdapter و...

کلاس‌های کار با پایگاه‌داده MS SQL Server در فضای نام System.Data.SqlClient قرار دارند. نام کلاس‌ها عبارتند از:

SqlConnection و SqlCommand و SqlDataAdapter و... که شرح این کلاس‌ها در بالا آمده است.

- ۱- برنامه درسی رشته شبکه و نرم افزار رایانه. دفتر تالیف کتاب های درسی فنی و حرفه ای و کاردانش (۱۳۹۴)
- ۲- استاندارد ارزشیابی حرفه فناوری اطلاعات و ارتباطات. دفتر تالیف کتاب های درسی فنی و حرفه ای و کاردانش (۱۳۹۳)
- ۳- استاندارد شایستگی حرفه فناوری اطلاعات و ارتباطات. دفتر تالیف کتاب های درسی فنی و حرفه ای و کاردانش (۱۳۹۲)
- ۴- شیوه نامه نحوه ارزشیابی دروس شایستگی های فنی و غیر فنی شاخه های فنی و حرفه ای و کاردانش شماره ۴۰۰/۲۱۱۴۸۲ مورخ ۹۵/۱۱/۳۰
- ۵- کربلایی، مجید. (۱۳۹۵). برنامه سازی ۳ کد ۳-۴۵۱/۵، تهران: شرکت چاپ و نشر کتاب های درسی ایران
- ۶- جعفرنژاد قمی، عین الله. (۱۳۹۳). آموزش گام به گام برنامه سازی به زبان C#، بابل: انتشارات علوم رایانه
- ۷- جعفری امین. (۱۳۹۲). آموزش گام به گام C#.NET 2010، تهران: انتشارات صفار
- ۸- احمدزاده، اسلام. کتاب الکترونیکی آموزش جامع برنامه نویسی C #
- ۹- آریانیان، احسان، احمدی، محمدرضا. ملکی، داوود. (۱۳۹۲). اصول مجازی سازی و رایانش ابری، تهران: انتشارات نیاز دانش
- 10- Agarwal, V. V. (2012). Beginning C# 5.0 databases. 2nd ed. (NY): Apress.
- 11- Stellman, A., & Greene, J. (2010). Head First C#. 2nd ed. (NY): O'Reilly.
- 12- Ray, E. (2003). Learning XML. Beijing Cambridge, Mass: O'Reilly
- 13- C# Programming Guide. <https://msdn.microsoft.com/>
- 14- All C# Language Topics. <https://stackoverflow.com/documentation/c/topics>