با ایران توشه
یک قدم جلوتر
باش!

ایران توشه
توشه ای برای موفقیت



ایران توشه، مرجع دانلود فایل های کمک آموزشی دانش آموزان و کنکوری ها

همه چیز
برای موفقیت
تو!

- دانلود رایگان آزمون های آزمایشی قلمچی، خیلی سبز، گاج و ...
- دانلود رایگان گام به گام تمامی دروس
- دانلود رایگان نمونه سوالات امتحانی (نوبت اول و دوم) و امتحانات نهایی
- دانلود رایگان جزوه های کمک آموزشی و شب امتحانی

کافیست هر فایلی که نیاز دارید را در گوگل به این صورت جستجو کنید:

فایل مورد نیاز + ایران توشه

مثال:

دانلود گام به گام دوازدهم تجربی ایران توشه



irantooshe.ir

t.me/irantooshe

گام به گام توسعه برنامه‌سازی و پایگاه داده پایه یازدهم
پودمان پنجم

عنوان فصل: مدیریت پایگاه داده

از صفحه‌ی ۲۰۵ تا صفحه‌ی ۲۴۸ کتاب

در این فایل پاسخ فعالیت‌ها، کارهای کلاسی، تمرین‌ها و پرسش‌های فصل مدیریت پایگاه داده آمده است.

مباحث فصل: پرونده و پوشه، خواندن و نوشتن در پرونده‌ی متنی، مدیریت خطا با try-catch، کلاس Directory و Path، اتصال به پایگاه داده، مقیدسازی، درج، حذف، ویرایش و جست‌وجوی رکورد

تهیه‌شده در ایران‌توشه، توشه‌ای برای موفقیت

irantooshe.ir

فهرست پرسش‌ها

۱ صفحه ۲۰۷ · فعالیت گروهی — پرونده و انواع آن ۶

۲ صفحه ۲۰۹ · کارگاه ۱ — ایجاد پرونده و نوشتن در آن (WriteAllText و AppendAllText) ۶

۳ صفحه ۲۱۰ · کارگاه ۲ — پروژه Notepad: ذخیره (Save As / Save / New) ۷

۴ صفحه ۲۱۳ · کارگاه ۳ — خواندن پرونده‌ی متنی با ReadAllText ۱۰

۵ صفحه ۲۱۴ · کارگاه ۴ — خواندن پرونده‌ی متنی با ReadAllLines ۱۱

۶ صفحه ۲۱۶ · کارگاه ۵ — مدیریت خطا با دستور try-catch ۱۲

۷ صفحه ۲۱۷ · فعالیت کارگاهی — جدول ۱: متدهای کلاس Directory ۱۳

۸ صفحه ۲۱۸ · فعالیت کارگاهی — جدول ۲: متدهای کلاس Path ۱۳

ایران توشه
توشه ای برای موفقیت

۹ صفحه ۲۱۹ · کارگاه ۶ — کاربرد کلاس‌های Directory و Path ۱۴
۱۰ صفحه ۲۲۱ · فعالیت گروهی — کاربرد پایگاه داده (دفترچه مخاطبان) ۱۵
۱۱ صفحه ۲۲۲ · فعالیت کارگاهی — ایجاد پایگاه داده (فروشگاه فرهنگیان) ۱۶
۱۲ صفحه ۲۲۳ · فعالیت کارگاهی — طراحی پایگاه داده‌ی بازی و ساخت MyShop ۱۷
۱۳ صفحه ۲۲۴ · کارگاه ۷ — ارتباط با پایگاه داده از طریق wizard (واژه‌نامه) ۱۸
۱۴ صفحه ۲۲۶ · کارگاه ۸ — سفارشی کردن DataGridView و جدول ۳ ۱۹
۱۵ صفحه ۲۲۷ · کارگاه ۹ — مقیدسازی کنترل‌ها (Binding) ۲۱
۱۶ صفحه ۲۲۹ · کارگاه ۱۰ — بارگذاری داده و پیمایش رکوردها ۲۲
۱۷ صفحه ۲۳۰ · کارگاه ۱۱ — دسترسی به خانه‌های DataGridView ۲۳
۱۸ صفحه ۲۳۲ · کارگاه ۱۲ — جستجو در جدول (SplitContainer و پرسوجوی WHERE) ۲۴

ایران توشه
توشه ای برای موفقیت

۱۹ صفحه ۲۳۵ · کارگاه ۱۳ — ورود کاربر (Login) به برنامه‌ی فروشگاه ۲۵
۲۰ صفحه ۲۳۷ · کارگاه ۱۴ — درج یک رکورد در جدول (INSERT) ۲۷
۲۱ صفحه ۲۳۹ · کارگاه ۱۵ — حذف رکورد از جدول (DELETE) ۲۹
۲۲ صفحه ۲۴۰ · کارگاه ۱۶ — ویرایش رکورد در جدول (UPDATE) ۳۰
۲۳ صفحه ۲۴۱ · کارگاه ۱۷ — درج و حذف سطر در DataGridView (برگه خرید) ۳۱
۲۴ صفحه ۲۴۵ · کارگاه ۱۸ — مدیریت خطا در برنامه‌ی پایگاه داده ۳۳
۲۵ صفحه ۲۴۶ · کارگاه ۱۹ — توسعه‌ی پروژه‌ی فروشگاه (TryParse، برگه خرید، کاربران) ۳۵
۲۶ صفحه ۲۴۷ · پروژه فاز ۵ — اتصال پروژه‌ی کتابخانه به پایگاه داده ۳۶

ایران توشه
توشه ای برای موفقیت

۱ صفحه ۲۰۷ فعالیت گروهی

در جدول زیر با کمک هم‌گروهی 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؟
(Adobe pdf، html Document، Microsoft Word Document، MP3 Audio File، C# source code)

نوع پرونده‌ها

نوع پرونده	پسوند	دودویی / متنی
Adobe pdf	.pdf	دودویی
html Document	.html	متنی
Microsoft Word Document	.docx	دودویی
MP3 Audio File	.mp3	دودویی
C# source code	.cs	متنی

پرونده‌های متنی (html و cs) در Notepad به‌صورت نویسه‌های خوانا (کد و متن) نمایش داده می‌شوند و می‌توان آنها را ویرایش کرد. پرونده‌های دودویی (pdf، docx، mp3) در Notepad فقط مجموعه‌ای از نویسه‌های نامفهوم و علامت‌های عجیب نشان می‌دهند، زیرا داده‌ها به همان شکل حافظه (بایت) ذخیره شده‌اند و باید با برنامه‌ی مخصوص خودشان باز شوند.

۲ صفحه ۲۰۹ کارگاه ۱

۴ ـ پرونده‌ی Info.txt را باز کنید. پوشه‌ی Debug پروژه را باز کنید. آیا پرونده‌ی متنی Info.txt در آن پوشه قرار دارد؟ محتوای این پرونده‌ی متنی چیست؟

۵ ـ بعد از اجرای مجدد برنامه، محتوای پرونده‌ی متنی Info.txt را بررسی کنید. نام «حسن» را وارد کرده، دکمه «ثبت» را کلیک کنید. آیا نام «علی» در پرونده‌ی Info.txt وجود دارد؟

۶ ـ دستور نوشتن در پرونده را به‌صورت AppendAllText تغییر دهید. نام‌های «حسن» و «علی» را وارد کنید. آیا هر دو نام در پرونده ذخیره شده است؟ متدهای WriteAllText و AppendAllText هر دو رشته‌ی data را در پرونده می‌نویسند. این دو متد چه تفاوتی با هم دارند؟

۸ ـ کدهای برنامه را به‌صورت زیر تغییر دهید. پرونده‌ی Info.txt را حذف کنید. چند نام دلخواه وارد و ثبت کنید. چه داده‌هایی در پرونده ذخیره می‌شوند؟

۹ ـ برنامه را طوری تغییر دهید که نام پدر و کد ملی نیز در پرونده ذخیره شود.

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

۴ ـ بله. چون نام پرونده بدون مسیر نوشته شده، پرونده در مسیر جاری برنامه، یعنی پوشه‌ی bin\Debug پروژه ساخته می‌شود. محتوای آن همان نام واردشده است: علی.

۵ ـ خیر. WriteAllText اگر پرونده از قبل وجود داشته باشد، محتوای قبلی را پاک می‌کند و متن جدید را جایگزین می‌کند؛ بنابراین فقط «حسن» در پرونده می‌ماند.

۶ ـ بله، هر دو نام ذخیره می‌شود. تفاوت: WriteAllText محتوای قبلی پرونده را پاک کرده و متن جدید را می‌نویسد، ولی AppendAllText متن جدید را به انتهای محتوای قبلی اضافه می‌کند و اطلاعات قبلی حفظ می‌شود. اگر پرونده وجود نداشته باشد، هر دو آن را می‌سازند.

```
private void btnSave_Click(object sender, EventArgs e)
{
    string data = String.Format("{0}\t{1}\r\n",
                  txtFirstName.Text, txtLastName.Text);
    File.AppendAllText("Info.txt", data);
}
```

۸ ـ در هر بار ثبت، یک خط جدید به پرونده افزوده می‌شود که شامل نام، یک فاصله‌ی tab و نام خانوادگی است؛ مثلاً علی رضایی. دنباله‌ی \t نویسه‌ی Tab و \r\n رفتن به خط بعد است.

۹ ـ کافی است قالب رشته را تغییر دهیم:

```
string data = String.Format("{0}\t{1}\t{2}\t{3}\r\n",
    txtFirstName.Text, txtLastName.Text,
    txtFatherName.Text, txtNationalCode.Text);
File.AppendAllText("Info.txt", data);
```

کنجکاوی: نام پرونده را همراه مسیر کامل می‌نویسیم، مثلاً:

```
File.AppendAllText(@"D:\Data\Info.txt", data);
```

پوشه‌ی مقصد باید از قبل وجود داشته باشد، وگرنه خطای DirectoryNotFoundException رخ می‌دهد (می‌توان از Directory.CreateDirectory برای ساخت آن استفاده کرد).

۳ صفحه ۲۱۰ **کارگاه ۲**

۴ ـ ویژگی Filter کادر محاوره‌ای SaveFileDialog را طوری تنظیم کنید که پرونده‌های متنی را نمایش دهد.

۵ ـ متد رویداد کلیک گزینه Save As منوی File را بنویسید. برنامه را اجرا کنید؛ متن دلخواهی بنویسید و در یک پرونده‌ی متنی جدید ذخیره کنید.

۶ ـ برای گزینه‌های Save و New منوی File کدنویسی کنید.

ایران توشه
توشه ای برای موفقیت

۷ ـ برنامه را توسعه دهید. برنامه‌ای بنویسید که قابلیت ذخیره‌ی مخاطبان در دفترچه تلفن را داشته باشد (شکل ۴). به کمک متد AppendAllText همه‌ی اطلاعات یک مخاطب را در یک خط به پرونده‌ی متنی ContactList.txt اضافه کنید.

برنامه را توسعه دهید که هنگام ایجاد پرونده‌ی جدید و یا خروج از برنامه، در صورت نیاز به ذخیره‌ی پرونده‌ی جاری، تأیید کاربر را با نمایش پیام مناسب دریافت کند.

بازی اعصاب‌سنج در فصل ۴ را توسعه دهید تا پس از پایان بازی نام کاربر و امتیازش را در یک پرونده‌ی متنی ذخیره کند.

۴ ـ در پنجره‌ی Properties مقدار Filter را روی txt.*|پرونده متنی تنظیم می‌کنیم (نیمه‌ی اول عنوان نمایشی و پس از خط عمودی الگوی پرونده‌هاست).

```
saveFileDialog1.Filter = "Text files|*.txt";
```

۵ ـ کد گزینه‌ی Save As (برای نگهداری مسیر، متغیر documentFilePath در سطح کلاس تعریف شده است):

```
private void saveAsToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        documentFilePath = saveFileDialog1.FileName;
        File.WriteAllText(documentFilePath, txtMainDocument.Text);
    }
}
```

۶ ـ گزینه‌ی Save: اگر پرونده قبلاً ذخیره شده باشد (مسیر خالی نیست) مستقیم در همان مسیر ذخیره می‌شود، وگرنه مانند Save As عمل می‌کند. گزینه‌ی New: متن را پاک کرده و مسیر را خالی می‌کند.

```
private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (documentFilePath == "")
        saveAsToolStripMenuItem_Click(sender, e);
    else
        File.WriteAllText(documentFilePath, txtMainDocument.Text);
}
private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    txtMainDocument.Clear();
    documentFilePath = "";
}
```

۷ ـ دفترچه تلفن: فرم شامل کادرمتن‌های txtFirstName، txtLastName، txtPhone، txtAddress، دکمه‌ی «افزودن به فهرست» (btnAdd) و کادرفهرست lstContacts است. هر مخاطب در یک خط ذخیره می‌شود:

```
private void btnAdd_Click(object sender, EventArgs e)
{
    string contact = String.Format("{0}\t{1}\t{2}\t{3}\r\n",
        txtFirstName.Text, txtLastName.Text,
        txtPhone.Text, txtAddress.Text);
    File.AppendAllText("ContactList.txt", contact);
    lstContacts.Items.Add(contact);
}
```

توسعه‌ی ۱: برای تأیید کاربر، یک متغیر منطقی isChanged در رویداد TextChanged کادرمتن true می‌شود و پیش از New یا خروج، پرسیده می‌شود:

```
private bool isChanged = false;
private void txtMainDocument_TextChanged(object sender, EventArgs e)
{
    isChanged = true;
}
private bool ConfirmSave()
{
    if (!isChanged) return true;
    DialogResult r = MessageBox.Show("Save changes?", "Notepad",
        MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
    if (r == DialogResult.Cancel) return false;
    if (r == DialogResult.Yes) saveToolStripMenuItem_Click(null, null);
    return true;
}
private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (!ConfirmSave()) return;
    txtMainDocument.Clear();
    documentFilePath = "";
    isChanged = false;
}
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (ConfirmSave()) Application.Exit();
}
```

توسعه‌ی ۲ (بازی اعصاب‌سنج): در پایان بازی، نام کاربر (از یک کادرمتن یا InputBox/فرم کوچک) و امتیاز را در پرونده می‌نویسیم:

```
string line = String.Format("{0}\t{1}\r\n", txtPlayer.Text, score);
File.AppendAllText("Scores.txt", line);
```

این جواب ممکن است دقیق نباشد. (طراحی دقیق فرم دفترچه تلفن و بازی اعصاب‌سنج به شکل‌های کتاب بستگی دارد؛ کد بالا نمونه‌ی معقولی از آن است.)

این فایل توسط سایت ایران‌توشه تهیه شده است.

irantooshe.ir

۳ ـ متد رویداد کلیک گزینه Open را بنویسید. برنامه را اجرا کرده، یک پرونده‌ی متنی دلخواه را باز کنید.

عملکرد متد رویداد کلیک گزینه Open چیست؟

۴ ـ برنامه را توسعه دهید. کد مرحله‌ی ۳ را طوری تغییر دهید تا قبل از بازکردن پرونده‌ی جدید، کادر محاوره‌ای ذخیره‌ی پرونده‌ی بازشده‌ی قبلی نمایش داده شود.

فعالیت منزل: با تنظیم ویژگی راست‌به‌چپ فرم و ترجمه‌ی گزینه‌های منو به زبان فارسی، برنامه Notepad را فارسی کنید. برای گزینه‌های منوی Edit برنامه Notepad کدنویسی کنید. (راهنمایی: از ویژگی‌های SelectionStart، SelectedText و SelectionLength استفاده کنید.)

عملکرد: با کلیک روی Open کادر محاوره‌ای انتخاب پرونده نمایش داده می‌شود؛ اگر کاربر پرونده‌ای را انتخاب و OK کند، مسیر آن در documentFilePath ذخیره شده و کل محتوای پرونده با File.ReadAllText خوانده و در کادرمتن txtMainDocument نمایش داده می‌شود.

```
private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (isChanged)
    {
        DialogResult r = MessageBox.Show("Save changes?", "Notepad",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (r == DialogResult.Yes)
            saveToolStripMenuItem_Click(sender, e);
    }
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        documentFilePath = openFileDialog1.FileName;
        txtMainDocument.Text = File.ReadAllText(documentFilePath);
        isChanged = false;
    }
}
```

فعالیت منزل: برای فارسی‌کردن، ویژگی‌های RightToLeft = Yes و RightToLeftLayout = True فرم را تنظیم کرده و Text گزینه‌ها را فارسی می‌کنیم (پرونده، ویرایش، جدید، باز کردن، ذخیره، ذخیره به عنوان، خروج، برش، کپی، چسباندن، انتخاب همه). کد منوی Edit:

ایران توشه
توشه‌ای برای موفقیت

```
private void cutToolStripMenuItem_Click(object sender, EventArgs e)
{ txtMainDocument.Cut(); }
private void copyToolStripMenuItem_Click(object sender, EventArgs e)
{ txtMainDocument.Copy(); }
private void pasteToolStripMenuItem_Click(object sender, EventArgs e)
{ txtMainDocument.Paste(); }
private void selectAllToolStripMenuItem_Click(object sender, EventArgs e)
{ txtMainDocument.SelectAll(); }
private void undoToolStripMenuItem_Click(object sender, EventArgs e)
{ txtMainDocument.Undo(); }
private void deleteToolStripMenuItem_Click(object sender, EventArgs e)
{
    int start = txtMainDocument.SelectionStart;
    int len = txtMainDocument.SelectionLength;
    if (len > 0)
        txtMainDocument.Text = txtMainDocument.Text.Remove(start, len);
    txtMainDocument.SelectionStart = start;
}
```

SelectedText متن انتخاب‌شده، SelectionStart شماره‌ی نویسه‌ی ابتدای انتخاب و SelectionLength طول آن است.

۵ صفحه ۲۱۴ کارگاه ۴

۲ ـ متد رویداد Form1_Load را بنویسید. برنامه را اجرا کنید. چه اطلاعاتی را در فهرست مشاهده می‌کنید؟ با تعریف یک نقطه‌ی توقف روی دومین خط متد رویداد Load فرم، برنامه را با کلید F11 گام‌به‌گام اجرا کنید. رشته‌ی data حاوی چه مقادیری است؟ عملکرد متد ReadAllLines چیست؟ عملکرد متد AddRange چیست؟

۳ ـ کد مرحله‌ی ۲ و ۳ را با هم مقایسه کنید. برنامه را اجرا کنید. تفاوت متد ReadAllLines با متد ReadAllText در چیست؟

۴ ـ پرونده‌ی ContactList.txt را از پوشه‌ی Debug برنامه حذف کنید. برنامه را اجرا کنید. چه خطایی رخ می‌دهد؟ چرا؟

۵ ـ کد با دستور if و File.Exists را بنویسید و اجرا کنید. نوع خروجی متد Exists چیست؟

۲ ـ در فهرست lstContacts، هر خط پرونده یک عنصر جداگانه است؛ یعنی هر مخاطب (نام، نام خانوادگی، تلفن و نشانی) در یک سطر نمایش داده می‌شود. آرایه‌ی data به تعداد خطوط پرونده عنصر دارد و هر عنصر یک خط از پرونده است. ReadAllLines پرونده را باز می‌کند، خط‌به‌خط می‌خواند، می‌بندد و خطوط را در یک آرایه‌ی رشته‌ای برمی‌گرداند. AddRange همه‌ی عناصر آرایه را یک‌جا به مجموعه‌ی Items کادرفهرست اضافه می‌کند.

ایران توشه
توشه ای برای موفقیت

۳ ـ ReadAllText کل پرونده را در یک رشته‌ی واحد برمی‌گرداند و در فهرست همه‌ی مخاطبان به‌صورت یک عنصر (در یک سطر) نمایش داده می‌شوند؛ اما ReadAllLines پرونده را به آرایه‌ای از خطوط تقسیم می‌کند و هر خط یک عنصر فهرست می‌شود.

۴ ـ خطای FileNotFoundException (پرونده پیدا نشد) رخ می‌دهد و برنامه متوقف می‌شود، زیرا پرونده‌ای با این نام در مسیر جاری وجود ندارد که خوانده شود.

```
private void Form1_Load(object sender, EventArgs e)
{
    if (File.Exists("ContactList.txt"))
        lstContacts.Items.AddRange(File.ReadAllLines("ContactList.txt"));
    else
        MessageBox.Show("Contact file not found", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
```

۵ ـ متد File.Exists مسیر پرونده را می‌گیرد و خروجی آن از نوع bool است: اگر پرونده موجود باشد true و در غیر این صورت false برمی‌گرداند. با این کار پیش از خواندن، از وجود پرونده مطمئن می‌شویم و برنامه دچار خطا نمی‌شود.

۶ صفحه ۲۱۶ کارگاه ۵

۲ ـ کد متد رویداد کلیک دکمه‌ی «افزودن به فهرست» را به‌صورت زیر تغییر دهید (ذخیره در a:\ContactList.txt). برنامه را اجرا کنید. چه خطایی مشاهده می‌کنید؟

۳ ـ با کمک هنرآموز خود پیام نشان داده شده در پنجره‌ی خطا را به فارسی معنی کنید.

۴ ـ با استفاده از ساختار try-catch خطای رخ‌داده را مدیریت کنید. برنامه را اجرا کنید. اجرای برنامه در این حالت چه تفاوتی با حالت قبل دارد؟

پیش از آن (فیلم): رضا هر روز با دوچرخه به محل کار می‌رود. اگر دوچرخه‌ی او دچار مشکل فنی شود، آیا بروز این مشکل اجرای برنامه‌ی روزانه‌ی او را متوقف می‌کند؟

ایران توشه
توشه‌ای برای موفقیت

پیش‌پرسش: خیر؛ او راه‌حل جایگزین دارد (اتوبوس یا مترو). در برنامه نیز با try-catch برای حالت خطا راه‌حل جایگزین تعریف می‌کنیم.

۲ ـ اگر درایو A: (فلاپی/فلش) وجود نداشته باشد، برنامه با خطای زمان اجرا از نوع DirectoryNotFoundException یا IOException متوقف می‌شود. نمونه‌ی پیام: Could not find a part of the path 'a:\ContactList.txt'.

۳ ـ معنی فارسی: «بخشی از مسیر 'a:\ContactList.txt' پیدا نشد» (یعنی درایو یا پوشه‌ی موردنظر وجود ندارد).

```
string contact = string.Format("{0}\t{1}\t{2}\t{3}\t",
    txtFirstName.Text, txtLastName.Text,
    txtPhone.Text, txtAddress.Text);
try
{
    File.AppendAllText("a:\\ContactList.txt", contact);
}
catch
{
    MessageBox.Show("Operation failed. Please try again.", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
```

۴ ـ در حالت قبل برنامه با پنجره‌ی خطا و توقف اجرا مواجه می‌شد؛ اکنون به‌جای توقف، کنترل برنامه به بخش catch می‌رود، پیام «اجرای عملیات امکان‌پذیر نیست. لطفاً یک بار دیگر سعی کنید» نمایش داده می‌شود و برنامه به کار خود ادامه می‌دهد.

۷ صفحه ۲۱۷ فعالیت کارگاهی

با استفاده از راهنمای مایکروسافت جدول ۱ را تکمیل کنید. (متدهای GetDirectories، GetFiles، Exists)

جدول ۱ـ عملکرد برخی از متدهای کلاس Directory

نام متد	نوع ورودی	نوع خروجی	عملکرد
GetDirectories	رشته (مسیر)	آرایه‌ی رشته‌ای	نام و مسیر همه‌ی زیرپوشه‌های مسیر ورودی را در یک آرایه برمی‌گرداند.
GetFiles	رشته (مسیر)	آرایه‌ی رشته‌ای	نام و مسیر همه‌ی پرونده‌های مسیر ورودی را در یک آرایه برمی‌گرداند.
Exists	رشته (مسیر)	bool	بررسی می‌کند پوشه‌ای با این مسیر وجود دارد یا نه (true یا false).

۸ صفحه ۲۱۸ فعالیت کارگاهی

با استفاده از راهنمای مایکروسافت جدول ۲ را تکمیل کنید. (GetFileName، Combine)

جدول ۲ـ عملکرد برخی از متدهای کلاس Path

ایران توشه
توشه‌ای برای موفقیت

نام متد	نوع ورودی	نوع خروجی	عملکرد
GetFileName	رشته (مسیر)	رشته	نام و پسوند پرونده‌ی ورودی را برمی‌گرداند.
Combine	دو رشته (مسیر) یا بیشتر	رشته	رشته‌های ورودی را با جداکننده‌ی مسیر (\) به هم می‌چسباند و یک مسیر کامل می‌سازد.

نمونه (اجراشده با mono): Path.GetFileName(@"C:\Users\Ali\Info.txt") مقدار Info.txt، و Path.Combine(@"C:\Users", "Ali", "a.txt") مقدار C:\Users\Ali\a.txt را برمی‌گرداند. متدهای GetExtension (مثلاً .txt) و GetFileNameWithoutExtension (مثلاً Info) نیز در این کلاس هستند.

۹ صفحه ۲۱۹ کارگاه ۶

۴ ـ متد رویداد کلیک دکمه‌ی «فهرست پوشه‌ها» را بنویسید. برنامه را اجرا کنید و روی دکمه «فهرست پوشه‌ها» کلیک کنید. چه عناصری در فهرست مشاهده می‌کنید؟

۵ ـ کد مرحله‌ی ۴ را تغییر دهید تا مسیر directory را از کادرمتن دریافت کند.

۶ و ۷ ـ متد رویداد کلیک دکمه‌ی «فهرست پرونده‌ها» را بنویسید. چه عناصری در فهرست مشاهده می‌کنید؟ سپس مسیر را از کادرمتن دریافت کند.

۸ ـ کد مرحله‌ی ۴ و ۸ را با هم مقایسه کنید. ۹ ـ متد دکمه‌ی «فهرست پرونده‌ها» را ویرایش کنید تا نام پرونده‌ها بدون مسیر نمایش داده شود.

۴ ـ نام و مسیر کامل همه‌ی پوشه‌های درون درایو C:\ (مثل C:\Windows، C:\Users، C:\Program Files و ...) در فهرست نمایش داده می‌شود. به‌همین ترتیب در مرحله‌ی ۶ نام و مسیر پرونده‌های موجود مستقیماً در C:\ (مثل pagefile.sys) نمایش داده می‌شود.

۵ و ۷ ـ برای دریافت مسیر از کادرمتن txtPath:

```
private void btnShowDirectories_Click(object sender, EventArgs e)
{
    string directory = txtPath.Text;
    lstBoard.Items.Clear();
    if (Directory.Exists(directory))
        lstBoard.Items.AddRange(Directory.GetDirectories(directory));
}
private void btnShowFiles_Click(object sender, EventArgs e)
{
    string directory = txtPath.Text;
    lstBoard.Items.Clear();
    if (Directory.Exists(directory))
        lstBoard.Items.AddRange(Directory.GetFiles(directory));
}
```

ایران توشه
توشه‌ای برای موفقیت

۸ ـ در کد مرحله‌ی ۴ نام کامل (به‌همراه مسیر، مثل C:\Windows) نمایش داده می‌شود، ولی در کد مرحله‌ی ۸ با Path.GetFileName فقط نام زیرپوشه (مثل Windows) در فهرست قرار می‌گیرد.

۹ ـ (کد کتاب)

```
private void btnShowFiles_Click(object sender, EventArgs e)
{
    string directory = txtPath.Text;
    lstBoard.Items.Clear();
    string[] files = Directory.GetFiles(directory);
    foreach (string file in files)
        lstBoard.Items.Add(Path.GetFileName(file));
}
```

برای نمونه اجرای واقعی همین متدها روی پوشه‌ای با دو زیرپوشه و دو پرونده: زیرپوشه‌ها d1/sub1 و d1/sub2، پرونده‌ها (با GetFileName) a.txt و b.txt.

۱۰ صفحه ۲۲۱ فعالیت گروهی

به کمک هم‌گروهی خود یک پایگاه داده برای ذخیره‌ی مخاطبان ایجاد کنید و به سؤالات زیر پاسخ دهید: چگونه پایگاه داده را در Access ایجاد می‌کنید؟ چگونه در پایگاه داده جدول مخاطبان را ایجاد می‌کنید؟ چه فیلدهایی را برای جدول مخاطبان تعریف می‌کنید؟ نوع داده‌ی این فیلدها چیست؟ چگونه رکورد جدید در جدول مخاطبان وارد می‌کنید؟ چگونه یک رکورد را حذف می‌کنید؟

چند نرم‌افزار پیرامون خود را نام ببرید که از پایگاه داده برای ذخیره و بازیابی اطلاعات استفاده می‌کنند. نام بازی‌هایی را بنویسید که از پایگاه داده برای ذخیره‌ی اطلاعات بازی استفاده می‌کنند. چند نرم‌افزار فارسی تلفن هوشمند را نام ببرید که از پایگاه داده استفاده می‌کند.

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

کنجکاوی: یک بازی برخط (online) را بررسی کنید که مشخصات، سطح و امتیاز کاربران در آن ذخیره می‌شود. فکر می‌کنید پایگاه داده‌ی این بازی از چه جدول‌هایی تشکیل شده است؟

1. اجرای Access ← Blank database ← تعیین نام (مثلاً Contacts) و پوشه ← Create.
2. ساخت جدول: Create ← Table Design؛ تعریف فیلدها و نوع داده؛ تعیین کلید اصلی؛ ذخیره با نام Contacts.
3. ورود رکورد: باز کردن جدول در Datasheet View و تایپ مقادیر در سطر خالی انتهای جدول.
4. حذف رکورد: انتخاب سطر ← راست‌کلیک ← Delete Record (یا کلید Delete) و تأیید.

فیلدهای پیشنهادی جدول مخاطبان

نام فیلد	نوع داده	توضیح
ContactID	AutoNumber	کلید اصلی
FirstName	Short Text	نام
LastName	Short Text	نام خانوادگی
Phone	Short Text	شماره‌ی تلفن (به‌دلیل صفر ابتدایی، متنی)
Address	Long Text	نشانی

نرم‌افزارهایی که از پایگاه داده استفاده می‌کنند: برنامه‌های فروشگاهی و حسابداری، نرم‌افزار بانکی و خودپرداز، سامانه‌ی کتابخانه، سامانه‌ی ثبت‌نام و نمره‌ی مدرسه، دفترچه‌ی مخاطبان گوشی. بازی‌ها: بازی‌های برخط (مانند Clash of Clans و PUBG که اطلاعات بازیکن، سطح و امتیاز را نگه می‌دارند)، بازی‌های موبایلی دارای جدول امتیازات. نرم‌افزارهای فارسی تلفن همراه: مثل دیوار، بله، ایتا، اسنپ، تپسی، بازار، دیجی‌کالا، دفترچه‌ی مخاطبان.

امکانات Access کافی نیست، زیرا Access فقط ساخت و مدیریت داده را به کاربر آشنا ارائه می‌کند؛ ولی کاربران عادی (فروشنده، کتابدار) نیاز به یک واسط گرافیکی ساده و اختصاصی، منطق برنامه (محاسبه‌ی مبلغ، کنترل ورود کاربر، چاپ و ...) و ارتباط با سایر اجزای برنامه دارند که با زبان برنامه‌نویسی ساخته می‌شود.

کنجکاوی: جدول‌های پیشنهادی بازی برخط: Players (کد بازیکن، نام کاربری، گذرواژه، تاریخ ثبت‌نام)، Levels (کد مرحله، عنوان، حداقل امتیاز لازم)، PlayerLevel (کد بازیکن، کد مرحله، امتیاز، تعداد ستاره)، و در صورت نیاز Items برای اشیای بازی.

۱۱ صفحه ۲۲۲ فعالیت کارگاهی

جدول بالا را در Access پیاده‌سازی کرده، 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؟ مشکل را چگونه حل می‌کنید؟

ـ برگه‌ی خرید شامل چه داده‌هایی می‌شود؟

ـ جدول «کالا» شامل چه فیلدهایی می‌شود؟ ـ جدول «مشتری» شامل چه فیلدهایی می‌شود؟

ـ جدول «برگه خرید» شامل چه فیلدهایی می‌شود؟ ـ جدول «ثبت برگه خرید» شامل چه فیلدهایی می‌شود؟

جدول اولیه (Access)

نام	نام خانوادگی	تاریخ	مبلغ خرید (تومان)
محمد	محمدی	1396/2/1	52000
علی	غلامی	1396/2/1	120000
فرهاد	سهرابی	1396/2/2	30000
محمد	محمدی	1396/2/3	45000

بله، مشکل پیش می‌آید؛ چون دو نفر با نام و نام خانوادگی یکسان از روی نام قابل تشخیص نیستند و خریدهای آنها با هم جمع می‌شود. راه‌حل: در جدول مشتری یک فیلد یکتا مانند CustomerCode (کد مشتری، یا کد ملی/کد پرسنلی) به‌عنوان کلید اصلی تعریف می‌کنیم و خریدها را با آن کد ثبت می‌کنیم.

برگه‌ی خرید شامل: شماره‌ی برگه، تاریخ، مشخصات مشتری (نام و نام خانوادگی)، فهرست اقلام خرید (نام کالا، تعداد، قیمت واحد، قیمت کل هر قلم) و مبلغ کل قابل پرداخت است.

ساختار جدول‌های پایگاه داده‌ی MyShop

جدول	فیلدها
Goods (کالا)	GoodsCode (کلید)، GoodsName، GoodsUnitPrice، GoodsStock
Customers (مشتری)	CustomerCode (کلید)، CustomerFirstName، CustomerLastName، CustomerPhone، CustomerAddress
Factors (برگه خرید)	FactorCode (کلید)، CustomerCode (کلید خارجی)، FactorDate
FactorItems (ثبت برگه خرید)	FactorCode و GoodsCode (کلید ترکیبی/خارجی)، Amount (تعداد)، UnitPrice

جدول «ثبت برگه خرید» لازم است زیرا هر مشتری در هر بار مراجعه ممکن است چند قلم کالا بخرد (رابطه‌ی یک به چند میان برگه و اقلام).

ایران توشه
توشه ای برای موفقیت

۱۲ صفحه ۲۲۳ فعالیت کارگاهی

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 داده‌ی مناسب را برای این بازی طراحی کنید تا نشان دهد که هر کاربر در هر سطح چند امتیاز و چند ستاره دارد.

پایگاه داده‌ی MyShop را با توجه به شکل ۶ در Access ایجاد کرده، داده‌ی مناسب را در جدول‌ها وارد کنید.

فعالیت منزل: 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 پایگاه داده‌ی فروشگاه صورت گیرد؟

طراحی پایگاه داده‌ی بازی

جدول	فیلدها
Users	UserID (کلید)، UserName، Password
Levels	LevelID (کلید)، LevelName، MinScore (حداقل امتیاز برای ورود)
UserLevels	UserID، LevelID (کلید ترکیبی)، Stars، Score

هر ردیف جدول UserLevels نشان می‌دهد هر کاربر در هر سطح چند امتیاز و چند ستاره دارد. Users و Levels از طریق کلید خارجی به آن وصل می‌شوند (رابطه‌ی چند به چند میان کاربر و سطح).

MyShop: چهار جدول Goods، Customers، Factors و FactorItems را با فیلدهای جدول صفحه‌ی ۲۲۲ می‌سازیم (کلید اصلی GoodsCode در Goods، CustomerCode در Customers، FactorCode در Factors) و در پنجره‌ی Relationships روابط را برقرار کرده، چند رکورد نمونه وارد می‌کنیم؛ مثلاً (۱، خودکار، ۵۰۰۰، ۱۰۰)، (۲، دفتر، ۱۲۰۰۰، ۵۰).

فعالیت منزل: یک جدول جدید Sellers (SellerCode به‌عنوان کلید، SellerName، ...) ایجاد می‌کنیم و یک فیلد SellerCode به‌عنوان کلید خارجی به جدول Factors (برگه‌ی خرید) اضافه می‌کنیم تا در هر برگه مشخص باشد کدام فروشنده آن را صادر کرده است.

این فایل توسط سایت ایران‌توشه تهیه شده است.
irantooshe.ir

۱۳ صفحه ۲۲۴ کارگاه ۷

۲ ـ جدول Computer را با فیلدهای زیر ایجاد کنید. فیلد WordName باید یکتا باشد. چگونه از تکراری بودن آن جلوگیری می‌کنید؟

۶ ـ کنترل DataGridView را به منبع اطلاعات متصل کنید. پس از مشاهده‌ی فیلم: چه کنترل‌هایی به فرم برنامه اضافه شده است؟ در بخش Solution Explorer چه تغییری رخ داده است؟ چه پرونده‌ای به این بخش اضافه شده است؟ کلید F7 را فشار دهید. آیا کدی به برنامه اضافه شده است؟

ساختار جدول Computer

نام فیلد	نوع داده	توضیح
WordCode	AutoNumber	کلید اصلی
WordName	Short Text	واژه (یکتا)
WordMeans	Long Text	معنی واژه

برای جلوگیری از تکراری بودن WordName در طراحی جدول، ویژگی Indexed فیلد را روی Yes (No Duplicates) می‌گذاریم (یا آن را کلید اصلی/یکتا قرار می‌دهیم).

۶ ـ کنترل‌های اضافه‌شده به فرم (در سینی کنترل‌های غیرمرئی): dictionaryDataSet، computerBindingSource و computerTableAdapter (و برای راه‌اندازی خودکار، tableAdapterManager و bindingNavigator در صورت انتخاب). در Solution Explorer پرونده‌ی dictionaryDataSet.xsd (و پایگاه داده‌ی dictionary.mdb) به پروژه افزوده می‌شود. با فشردن F7 کد فرم باز می‌شود: wizard در رویداد Load فرم خودکار این خط را می‌نویسد:

```
this.computerTableAdapter.Fill(this.dictionaryDataSet.Computer);
```

یعنی بله، کدی به برنامه اضافه شده است (کد تولیدی طراح در Form1.Designer.cs و خط Fill بالا در Form1.cs).

۱۴ صفحه ۲۲۶ کارگاه ۸

۲ ـ فیلدهایی را که می‌خواهید نمایش داده نشود، حذف کنید. کدام فیلد نباید در واژه‌نامه نمایش داده شود؟ ۳ ـ HeaderText را فارسی کنید. در DataGridView چه تغییری اتفاق می‌افتد؟ ۴ ـ ویژگی Dock را به Fill تغییر دهید. چه تغییری مشاهده می‌کنید؟ ۵ ـ ویژگی AutoSizeColumnsMode را به Fill تغییر دهید. چه تغییری مشاهده می‌کنید؟

۶ ـ مقدار DataSource را None کنید. چه تغییری مشاهده می‌کنید؟ مقدار قبلی را بگذارید. چه تغییری مشاهده می‌کنید؟

۷ ـ رنگ قلم و زمینه‌ی کنترل را تغییر دهید. برای قالب‌بندی DataGridView چگونه ویژگی DefaultCellStyle را تغییر می‌دهید؟ برنامه را اجرا کنید و روی سرستون «واژه» کلیک کنید. چه تغییری رخ می‌دهد؟ آیا می‌توانید از طریق این کنترل رکوردی را حذف یا اضافه کنید؟ چه ویژگی را باید تغییر دهید؟

جدول ۳ را تکمیل کنید.

فعالیت منزل: جدول مخاطبان تلفن همراه را روی فرم نمایش دهید. از پایگاه داده‌ی فروشگاه، جدول کالا را روی فرم نمایش دهید. پژوهش: در مورد ویژگی‌های دیگر کنترل DataGridView تحقیق کنید.

۲ ـ فیلد WordCode (کد واژه، که فقط کلید اصلی خودکار است) نباید نمایش داده شود؛ با انتخاب آن در Edit Columns و کلیک Remove حذف می‌شود.

ایران توشه
توشه‌ای برای موفقیت

۳ ـ سرستون‌ها به‌جای نام فیلد انگلیسی، عنوان فارسی (مثلاً «واژه» و «معنی») نشان می‌دهند. ۴ ـ با Dock = Fill جدول تمام فضای فرم/ظرف را پر می‌کند و با تغییر اندازه‌ی فرم کش می‌آید. ۵ ـ با AutoSizeColumnsMode = Fill پهنای ستون‌ها طوری تنظیم می‌شود که کل عرض کنترل را پر کنند و نوار پیمایش افقی لازم نباشد.

۶ ـ با DataSource = None جدول خالی می‌شود و هیچ داده‌ای (و ستونی) نمایش داده نمی‌شود، چون کنترل به منبع داده وصل نیست. با برگرداندن مقدار قبلی (computerBindingSource) دوباره داده‌ها نمایش داده می‌شوند.

۷ ـ در Properties، DefaultCellStyle را باز می‌کنیم (دکمه‌ی ...) و ForeColor، BackColor، Font و Alignment را تنظیم می‌کنیم. با کلیک روی سرستون، داده‌ها بر اساس آن ستون مرتب (صعودی/نزولی) می‌شوند. از طریق کنترل می‌توان رکورد حذف یا اضافه کرد اگر ویژگی‌های AllowUserToAddRows و AllowUserToDeleteRows برابر True باشند (و منبع داده قابل ویرایش باشد).

جدول ۳ـ ویژگی‌های DataGridView

نام ویژگی	شرح
Dock	مشخص می‌کند که کدام لبه از کنترل به فرم (ظرف) محدود شود.
AutoSizeColumnsMode	وضعیت اندازه‌ی خودکار برای ستون را مشخص می‌کند.
DefaultCellStyle	قالب (رنگ، قلم، تراز و ...) پیش‌فرض سلول‌های داده را مشخص می‌کند.
DataSource	منبع داده‌ی کنترل (مثلاً BindingSource) را مشخص می‌کند.
HeaderText	عنوان نمایش‌داده‌شده‌ی سرستون را مشخص می‌کند.
ColumnHeadersDefaultCellStyle	قالب پیش‌فرض سلول‌های سرستون را مشخص می‌کند.
EnableHeadersVisualStyles	اگر False باشد، قالب سفارشی سرستون (رنگ و ...) اعمال می‌شود و سبک ظاهری ویندوز نادیده گرفته می‌شود.

پژوهش: ویژگی‌های دیگر مانند ReadOnly (فقط‌خواندنی)، AllowUserToAddRows، AllowUserToDeleteRows، SelectionMode (انتخاب کل سطر)، MultiSelect، AlternatingRowsDefaultCellStyle (رنگ یک‌درمیان سطرها)، RowHeadersVisible، ColumnCount، Columns و Rows.

فعالیت منزل: همانند کارگاه ۷ برای جدول Contacts و جدول Goods با wizard یک DataGridView به منبع داده متصل می‌کنیم؛ مراحل: DataSet ← Database ← Add New Data Source ← انتخاب پرونده ← انتخاب جدول ← کشیدن جدول به فرم.

ایران توشه
توشه‌ای برای موفقیت

۲ ـ برای فراهم کردن امکان نمایش چند خط در کادرمتن txtWordMeans، کدام ویژگی کادرمتن را باید تغییر داد؟

۳ ـ کادرهای متن را به فیلد موردنظر مقید کنید. برنامه را اجرا کنید و روی یک سطر دلخواه در DataGridView کلیک کنید. چه مقداری در txtWordName قرار می‌گیرد؟

کنجکاوی: چه کنترل‌هایی را می‌توان مقید کرد؟ چند کنترل را نام ببرید. چگونه با کدنویسی می‌توان یک کنترل را نامقید کرد؟

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

۵ ـ ComboBox را به فیلد WordName مقید کنید. ۶ ـ عملکرد کنترل BindingNavigator را بررسی کنید.

پس از طراحی پایگاه داده‌ی فروشگاه و ورود داده‌های مناسب در جدول‌ها، فرم برنامه‌ی فروشگاه را به‌صورت شکل ۸ طراحی کنید.

---

۲ ـ ویژگی MultiLine را True می‌کنیم (برای نوار پیمایش هم ScrollBars = Vertical).

۳ ـ در DataBindings ← Advanced ← Text، منبع computerBindingSource و فیلد WordName را انتخاب می‌کنیم. با کلیک روی هر سطر DataGridView، مقدار فیلد WordName همان رکورد (یعنی واژه‌ی همان سطر) در txtWordName قرار می‌گیرد؛ txtWordMeans هم به همین شکل به WordMeans مقید می‌شود.

کنجکاوی: کنترل‌هایی مانند TextBox، Label، ListBox، ComboBox، CheckBox، DateTimePicker و PictureBox قابل مقید شدن هستند. برای نامقید کردن با کد، مقیدشدن را پاک می‌کنیم:

```
txtWordName.DataBindings.Clear();   // removes all bindings of the control
```

۴ و ۵ ـ مقیدسازی کادرفهرست و کادر ترکیبی با کد (یا از Properties):

```
listBox1.DataSource = computerBindingSource;
listBox1.DisplayMember = "WordName";
comboBox1.DataSource = computerBindingSource;
comboBox1.DisplayMember = "WordName";
// binding a textbox by code:
txtWordName.DataBindings.Add("Text", computerBindingSource, "WordName");
```

۶ ـ BindingNavigator یک نوار ابزار برای پیمایش و مدیریت رکوردهاست (اولین، قبلی، شماره‌ی رکورد جاری، بعدی، آخرین، افزودن، حذف و ذخیره)، که با تنظیم BindingSource = computerBindingSource بدون کدنویسی کار می‌کند.

ایران توشه
توشه ای برای موفقیت

فرم فروشگاه (شکل ۸): یک TabControl با نام tabShop و سربرگ‌های «کالا» (tabPageGoods)، «مشتری»، «برگه خرید» (tabPageFactor) و ... ساخته می‌شود؛ در سربرگ کالا یک DataGridView، چهار کادرمتن برای مقادیر فیلدها، دکمه‌های پیمایش و دکمه‌های «ثبت»، «حذف» و «ویرایش» قرار می‌گیرد.

این جواب ممکن است دقیق نباشد. (چیدمان دقیق شکل ۸ در متن موجود نیست؛ طرح بالا بر اساس سربرگ‌های ذکرشده در کتاب است.)

---

۱۶ صفحه ۲۲۹ کارگاه ۱۰

کنجکاوی: myShopDataSet شامل چه جدول‌های دیگری است؟

۳ ـ متد Load فرم را بنویسید. برنامه را اجرا کنید. آیا می‌توان برای مقیدسازی کنترل dgvGoods دستور دیگری نوشت؟

۴ ـ برای رفتن به اولین رکورد کد زیر را بنویسید. برای رفتن به آخرین رکورد چه دستوری می‌نویسید؟

۵ ـ برای رفتن به رکورد قبلی کد بنویسید. برای رفتن به رکورد بعدی چه کدی می‌نویسید؟ (جدول نام و عملکرد دکمه‌ها)

کنجکاوی: myShopDataSet علاوه بر Goods، شامل جدول‌های Customers، Factors، FactorItems و Users است.

نام کنترل‌های پیمایش

کنترل	btnFirst	btnPrev	btnNext	btnLast
Text	>>	>	<	<<
عملکرد	رفتن به اولین رکورد	رفتن به رکورد قبلی	رفتن به رکورد بعدی	رفتن به آخرین رکورد

(در متن کتاب ترتیب نام‌ها و عملکردها در جدول به‌هم‌ریخته است؛ جدول بالا صورت درست است.)

```
private void MainForm_Load(object sender, EventArgs e)
{
    this.goodsTableAdapter.Fill(this.myShopDataSet.Goods);
    dgvGoods.DataSource = goodsBindingSource;
}
private void btnFirst_Click(object sender, EventArgs e)
{ goodsBindingSource.MoveFirst(); }
private void btnPrev_Click(object sender, EventArgs e)
{ goodsBindingSource.MovePrevious(); }
private void btnNext_Click(object sender, EventArgs e)
{ goodsBindingSource.MoveNext(); }
private void btnLast_Click(object sender, EventArgs e)
{ goodsBindingSource.MoveLast(); }
```

ایران توشه
توشه‌ای برای موفقیت

دستور Fill جدول Goods را از پایگاه داده می‌خواند و داده‌ها را در DataSet می‌ریزد. برای مقیدسازی dgvGoods دستور دیگر: dgvGoods.DataSource = myShopDataSet.Goods; یا dgvGoods.DataSource = myShopDataSet; dgvGoods.DataMember = "Goods";. برای رفتن به آخرین رکورد ()MoveLast و برای رکورد بعدی ()MoveNext نوشته می‌شود.

۱۷ صفحه ۲۳۰ کارگاه ۱۱

کنجکاوی: مقدار اندیس (شماره‌ی) اولین رکورد چیست؟

۳ ـ متد رویداد CellClick کنترل dgvGoods را بنویسید و برنامه را اجرا کنید و با کلیک روی یک سطر دلخواه نتیجه‌ی کار را ببینید.

فعالیت منزل: در سربرگ مشتری، فرمی همانند فرم کالا ایجاد کنید و از طریق کدنویسی بارگذاری داده در DataGridView را انجام دهید و دکمه‌های پیمایش رکوردها را ایجاد کنید. (فعالیت کارگاهی: برای سربرگ مشتری برنامه‌ی فروشگاه، عملیات بالا را انجام دهید.)

اندیس اولین رکورد (سطر) صفر است.

```
private void dgvGoods_CellClick(object sender, DataGridViewCellEventArgs e)
{
    int index = dgvGoods.CurrentCell.RowIndex;
    txtGoodsCode.Text = dgvGoods.Rows[index].Cells[0].Value.ToString();
    txtGoodsName.Text = dgvGoods.Rows[index].Cells[1].Value.ToString();
    txtUnitPrice.Text = dgvGoods.Rows[index].Cells[2].Value.ToString();
    txtStock.Text     = dgvGoods.Rows[index].Cells[3].Value.ToString();
}
```

با کلیک روی هر سطر، مقدار چهار خانه‌ی همان سطر (کد، نام، قیمت واحد و موجودی) در چهار کادرمتن نشان داده می‌شود.

سربرگ مشتری: همان کارها را با customersTableAdapter، customersBindingSource، dgvCustomers و کادرمتن‌های مشتری انجام می‌دهیم:

```
this.customersTableAdapter.Fill(this.myShopDataSet.Customers);
dgvCustomers.DataSource = customersBindingSource;

private void btnCFirst_Click(object sender, EventArgs e)
{ customersBindingSource.MoveFirst(); }
private void btnCLast_Click(object sender, EventArgs e)
{ customersBindingSource.MoveLast(); }
private void dgvCustomers_CellClick(object sender, DataGridViewCellEventArgs e)
{
    int i = dgvCustomers.CurrentCell.RowIndex;
    txtCustomerCode.Text = dgvCustomers.Rows[i].Cells[0].Value.ToString();
    txtCustomerName.Text = dgvCustomers.Rows[i].Cells[1].Value.ToString();
}
```

ایران توشه
توشه‌ای برای موفقیت

۳ ـ پنل سمت راست را به‌صورت افقی به دو بخش تقسیم کنید. ویژگی Orientation آن را Horizontal قرار دهید. در این حالت چه تغییری مشاهده می‌کنید؟

۴ ـ کنترل کادرفهرست را به پنل پایین در سمت راست فرم اضافه کنید و Dock آن را Fill قرار دهید. در این حالت چه تغییری مشاهده می‌کنید؟

کنجکاوی: روی چه کنترل‌هایی می‌توان کنترل SplitContainer را قرار داد؟

۸ ـ برای جستجوی واژه، متد رویداد TextChanged کادرمتن جستجو (txtSearchWord) را بنویسید. عملکرد دستور if چیست؟ عملکرد دستور SELECT چیست؟ علامت «سؤال» در پرسوجو چه چیزی است؟

۹ ـ برای استفاده از متد GetDataByWordName دستور if را ویرایش کنید. این کد را با کد مرحله‌ی ۸ مقایسه کنید و تفاوت دو متد GetDataByWordName و FillByWordName را بنویسید. نوع ویژگی DataSource کنترل computerBindingSource چیست؟

۱۰ ـ پرسوجو را برای جستجوی کلمات مشابه با LIKE ویرایش کنید. عملکرد دستور Like چیست؟

فعالیت منزل: قابلیت جستجو بر اساس فیلد نام و کدکالا را برای سربرگ کالا ایجاد کنید. قابلیت جستجو بر اساس نام و شماره‌ی همراه را برای جدول مخاطبان ایجاد کنید.

---

۳ ـ پنل سمت راست به دو بخش بالایی و پایینی (به‌جای چپ و راست) تقسیم می‌شود و بین آنها یک نوار افقی قرار می‌گیرد. ۴ ـ کادرفهرست تمام فضای پنل پایین را پر می‌کند.

کنجکاوی: SplitContainer را می‌توان روی Form، Panel، GroupBox، TabPage و حتی یک SplitContainer دیگر (پنل‌های آن) قرار داد.

۷ ـ پرسوجوی ساخته‌شده در wizard:

```
SELECT WordCode, WordName, WordMeans
FROM Computer
WHERE (WordName = ?)
```

SELECT برای انتخاب (بازیابی) داده‌ها از جدول است و ستون‌های ذکرشده را از رکوردهای جدول Computer که شرط WHERE را دارند برمی‌گرداند. علامت ? جای‌نگهدار ورودی (پارامتر) متد است که هنگام فراخوانی، مقدار آن (متن کادر جستجو) جایگزین می‌شود.

```
private void txtSearchWord_TextChanged(object sender, EventArgs e)
{
    if (txtSearchWord.TextLength > 0)
        computerTableAdapter.FillByWordName(dictionaryDataSet.Computer,
                                            txtSearchWord.Text);
    else
        computerTableAdapter.Fill(dictionaryDataSet.Computer);
}
```

ایران توشه
توشه‌ای برای موفقیت

دستور if: اگر کادر جستجو خالی نباشد، فقط رکورد(های) مطابق متن در جدول جستجو و نمایش داده می‌شوند، وگرنه همه‌ی رکوردها دوباره بارگذاری می‌شوند. متد FillByWordName مستقیماً DataTable را پر می‌کند و خروجی ندارد؛ اما GetDataByWordName یک DataTable جدید برمی‌گرداند که باید به DataSource داده شود:

```
if (txtSearchWord.TextLength > 0)
    computerBindingSource.DataSource =
        computerTableAdapter.GetDataByWordName(txtSearchWord.Text);
else
    computerTableAdapter.Fill(dictionaryDataSet.Computer);
```

نوع DataSource در BindingSource از نوع object است (می‌پذیرد DataTable ،DataSet، فهرست و ...).

برای جستجوی کلمات مشابه:

```
SELECT WordCode, WordName, WordMeans
FROM Computer
WHERE (WordName Like ? + '%')
```

Like برای مقایسه‌ی الگو است؛ با الگوی '%' + ? همه‌ی واژه‌هایی را که با متن تایپ‌شده شروع می‌شوند برمی‌گرداند (مثلاً با تایپ «Co» همه‌ی واژه‌های شروع‌شونده با Co). این رفتار با sqlite در پایتون نیز آزمایش شد: فیلتر '%' || 'N' Like فقط رکورد «Note» را برگرداند.

فعالیت منزل: برای جدول کالا یک متد FillByGoodsNameOrCode با پرسوجوی زیر می‌سازیم و در رویداد TextChanged کادر جستجو فراخوانی می‌کنیم:

```
SELECT GoodsCode, GoodsName, GoodsUnitPrice, GoodsStock
FROM Goods
WHERE (GoodsName Like ? + '%') OR (GoodsCode = ?)
```

برای جدول مخاطبان: WHERE (FirstName Like ? + '%' OR LastName Like ? + '%') AND (Phone Like ? + '%').

این جواب ممکن است دقیق نباشد. (در Access برای Like معمولاً از علامت * یا % بسته به ADO استفاده می‌شود؛ در پرسوجوهای TableAdapter با OleDb علامت % کار می‌کند.)

۱۹ صفحه ۲۳۵ کارگاه ۱۳

۴ ـ یک متد جدید با نام FillByUserNameAndPass برای usersTableAdapter ایجاد کنید.

۵ ـ در LoginForm فیلدی برای اعتبارسنجی کاربر به نام succeeded تعریف کنید. ۶ ـ متد رویداد کلیک دکمه‌ی btnLogin را بنویسید. مفهوم شرط دستور if چیست؟ عملکرد متد btnLogin_Click چیست؟

۷ ـ متد رویداد Load فرم اصلی را بنویسید.

فعالیت منزل: برنامه‌ی 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.

پژوهش: با بازکردن جدول کاربران در پایگاه داده، هر فردی می‌تواند نام کاربری و گذرواژه‌ی همه‌ی کاربران را ببیند و به برنامه‌ی فروشگاه وارد شود. برای امنیت بیشتر برنامه چه کار باید کرد؟

۴ ـ پرسوجوی FillByUserNameAndPass:

```
SELECT UserName, Password, UserFirstName, UserLastName,
       TypeOfAccess, LastEntryDate, LastExitDate
FROM Users
WHERE (UserName = ?) AND (Password = ?)
```

```
public partial class LoginForm : Form
{
    public bool succeeded;     // true if the user is valid
    ...
    private void btnLogin_Click(object sender, EventArgs e)
    {
        usersTableAdapter.FillByUserNameAndPass(myShopDataSet.Users,
            txtUserName.Text, txtPassword.Text);
        if (myShopDataSet.Users.Rows.Count > 0)
        {
            succeeded = true;
            this.Close();
        }
        else
            MessageBox.Show("Invalid user name or password", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

```
private void MainForm_Load(object sender, EventArgs e)
{
    LoginForm frmLogin = new LoginForm();
    frmLogin.ShowDialog();
    if (!frmLogin.succeeded)
        this.Close();
}
```

شرط if بررسی می‌کند آیا پرسوجو حداقل یک رکورد (کاربری با این نام کاربری و گذرواژه) پیدا کرده است (Rows.Count > 0)؛ اگر بله، کاربر معتبر است. btnLogin_Click با کلیک روی دکمه، کاربر را با نام کاربری و گذرواژه جستجو می‌کند؛ اگر یافت شد فرم ورود بسته می‌شود و succeeded برابر true می‌شود، وگرنه پیام خطا نشان می‌دهد. در MainForm_Load فرم ورود به‌صورت ShowDialog (مودال) نمایش داده می‌شود و اگر ورود موفق نبود، فرم اصلی بسته می‌شود.

فعالیت منزل: یک CheckBox به نام chkRemember اضافه می‌کنیم. با تیک‌خوردن، نام کاربری در یک پرونده‌ی متنی (یا Properties.Settings) ذخیره می‌شود و در Load فرم ورود خوانده می‌شود؛ تصویر کاربر هم با فیلد مسیر تصویر در جدول Users و یک PictureBox نمایش داده می‌شود:

ایران توشه
توشه‌ای برای موفقیت

```
// on successful login
if (chkRemember.Checked)
    File.WriteAllText("user.txt", txtUserName.Text);
// in LoginForm_Load
if (File.Exists("user.txt"))
{
    txtUserName.Text = File.ReadAllText("user.txt");
    chkRemember.Checked = true;
}
// picture (UserPicture holds the image path)
if (File.Exists(path)) picUser.Image = Image.FromFile(path);
```

پژوهش: گذرواژه‌ها را به‌صورت متن ساده ذخیره نکنیم؛ آنها را با الگوریتم درهم‌ساز (Hash) مانند SHA-256 همراه با Salt ذخیره و هنگام ورود، هش گذرواژه‌ی واردشده را با هش ذخیره‌شده مقایسه کنیم؛ برای پایگاه داده رمز عبور بگذاریم، دسترسی کاربران به پرونده‌ی پایگاه داده را محدود کنیم، از پرسوجوی پارامتری (نه چسباندن رشته) استفاده کنیم و کادر گذرواژه را با UseSystemPasswordChar = True بپوشانیم.

این جواب ممکن است دقیق نباشد. (فیلم «اتصال به پایگاه داده» در دسترس نبود؛ مراحل آن بر اساس همان کارگاه ۷ نوشته شد.)

این فایل توسط سایت ایران‌توشه تهیه شده است.

irantooshe.ir

۲۰ صفحه ۲۳۷ کارگاه ۱۴

۲ ـ متد «اضافه‌کردن کالا» را ایجاد کنید. برای ایجاد این متد روی فایل myShopDataSet.xsd دابل‌کلیک کنید. چه جدول‌هایی مشاهده می‌کنید؟

کنجکاوی: ساختار دستور INSERT INTO در زبان SQL چگونه است؟

۳ ـ متد رویداد کلیک دکمه‌ی «ثبت» را بنویسید. متد InsertQuery دارای چند آرگومان ورودی است؟ چرا برخی از ورودی‌ها عددی است؟ چرا از دستور if در برنامه استفاده شده است؟ عملکرد متد btnAddGoods_Click و دستور goodsTableAdapter.InsertQuery چیست؟ برنامه را اجرا کنید و یک رکورد جدید را وارد کنید و «ثبت» را کلیک کنید. آیا رکورد جدید در dgvGoods نمایش داده می‌شود؟ چرا؟ برنامه را ببندید و دوباره اجرا کنید. آیا رکورد اضافه‌شده در dgvGoods وجود دارد؟ جدول Goods پایگاه داده‌ی MyShop را باز کنید. آیا رکورد اضافه‌شده در جدول وجود دارد؟

۴ ـ ویژگی Copy to Output Directory پایگاه داده را تنظیم کنید. دوباره برنامه را اجرا کنید و یک رکورد جدید را وارد کرده، «ثبت» را کلیک کنید و برنامه را ببندید. دوباره اجرا کنید. آیا در جدول Goods این کالا اضافه شده است؟

فعالیت منزل: برای جدول مشتری برنامه‌ی درج رکورد را بنویسید. برای جدول مخاطبان برنامه‌ی ثبت مخاطب جدید را بنویسید.

۲ ـ جدول‌های Goods، Customers، Factors، FactorItems و Users (با GoodsTableAdapter و ... پایین هر جدول) در طراح مشاهده می‌شود.

ساختار کلی: INSERT INTO نام_جدول (فیلد۱, فیلد۲, ...) VALUES (... ,مقدار۱, مقدار۲). پرسوجوی ساخته‌شده:

```
INSERT INTO Goods (GoodsCode, GoodsName, GoodsUnitPrice, GoodsStock)
VALUES (?, ?, ?, ?)
```

```
private void btnAddGoods_Click(object sender, EventArgs e)
{
    if (txtGoodsCode.TextLength > 0 && txtGoodsName.TextLength > 0
        && txtUnitPrice.TextLength > 0)
    {
        goodsTableAdapter.InsertQuery(int.Parse(txtGoodsCode.Text),
            txtGoodsName.Text, int.Parse(txtUnitPrice.Text),
            int.Parse(txtStock.Text));
        goodsTableAdapter.Fill(myShopDataSet.Goods);
        MessageBox.Show("Saved successfully");
    }
}
```

متد InsertQuery چهار آرگومان دارد (به تعداد علامت‌های «؟»)؛ کد، قیمت و موجودی در جدول عددی‌اند، بنابراین با int.Parse از متن به عدد تبدیل می‌شوند. if بررسی می‌کند فیلدهای ضروری خالی نباشند تا خطای تبدیل/درج رخ ندهد. btnAddGoods_Click رکورد جدید را در جدول Goods درج کرده و جدول نمایش‌داده‌شده را دوباره بارگذاری (Fill) می‌کند؛ goodsTableAdapter.InsertQuery دستور INSERT را روی پایگاه داده اجرا می‌کند.

اجرا: بله، رکورد جدید در dgvGoods نمایش داده می‌شود (چون بعد از درج، Fill اجرا شده). اما با بستن و اجرای دوباره، رکورد وجود ندارد و در پرونده‌ی پایگاه داده‌ی اصلی هم نیست؛ زیرا هر بار اجرا، Visual Studio نسخه‌ی پایگاه داده‌ی پروژه را روی نسخه‌ی پوشه‌ی bin\Debug کپی (بازنویسی) می‌کند و تغییرات در همان نسخه‌ی خروجی بوده است.

۴ ـ با تغییر Copy to Output Directory از Copy always به Copy if newer، پایگاه داده فقط وقتی کپی می‌شود که نسخه‌ی پروژه جدیدتر باشد؛ پس تغییرات در نسخه‌ی خروجی حفظ می‌شود و در اجرای بعدی رکورد اضافه‌شده موجود است.

فعالیت منزل: برای جدول مشتری یک پرسوجوی InsertQuery مشابه بسازید و کد زیر را بنویسید (نام کنترل‌ها نمونه است):

```
private void btnAddCustomer_Click(object sender, EventArgs e)
{
    if (txtCustomerCode.TextLength > 0 && txtCustomerName.TextLength > 0)
    {
        customersTableAdapter.InsertQuery(int.Parse(txtCustomerCode.Text),
            txtCustomerName.Text, txtPhone.Text);
        customersTableAdapter.Fill(myShopDataSet.Customers);
        MessageBox.Show("Saved successfully");
    }
}
// Contacts table: same idea
contactsTableAdapter.InsertQuery(txtFirstName.Text, txtLastName.Text,
                                 txtPhone.Text, txtAddress.Text);
contactsTableAdapter.Fill(dictionaryDataSet.Contacts);
```

۲۱ صفحه ۲۳۹ کارگاه ۱۵

۳ ـ متد رویداد کلیک دکمه‌ی «حذف» را بنویسید. برنامه را اجرا کرده، یک رکورد دلخواه را حذف کنید.

۴ ـ دستور را تغییر دهید تا حذف بر اساس نام کالا باشد. ۵ ـ دستور را برای حذف تمامی رکوردها تغییر دهید.

فعالیت منزل: برای جدول مشتری برنامه‌ی حذف رکورد را بنویسید. برای جدول مخاطبان برنامه‌ی حذف مخاطب را بنویسید.

```
DELETE FROM Goods WHERE (GoodsCode = ?)
```

```
private void btnDeleteGoods_Click(object sender, EventArgs e)
{
    if (txtGoodsCode.TextLength > 0)
    {
        goodsTableAdapter.DeleteQuery(int.Parse(txtGoodsCode.Text));
        goodsTableAdapter.Fill(myShopDataSet.Goods);
        MessageBox.Show("Deleted successfully");
    }
}
```

۴ ـ حذف بر اساس نام: شرط پرسوجو را به WHERE (GoodsName = ?) تغییر می‌دهیم (متد DeleteQuery یک ورودی رشته‌ای می‌گیرد) و فراخوانی می‌شود:

goodsTableAdapter.DeleteQuery(txtGoodsName.Text);.

۵ ـ حذف همه‌ی رکوردها: دستور بدون شرط نوشته می‌شود و متد بدون ورودی خواهد بود:

```
DELETE FROM Goods
```

(این دستور خطرناک است؛ بهتر است پیش از اجرا از کاربر تأیید گرفته شود.) آزمایش با sqlite: پس از DELETE FROM Goods WHERE GoodsCode = 1 از سه رکورد، دو رکورد باقی ماند.

ایران توشه
توشه‌ای برای موفقیت

```
private void btnDeleteCustomer_Click(object sender, EventArgs e)
{
    if (txtCustomerCode.TextLength > 0)
    {
        customersTableAdapter.DeleteQuery(int.Parse(txtCustomerCode.Text));
        customersTableAdapter.Fill(myShopDataSet.Customers);
    }
}
// Contacts
contactsTableAdapter.DeleteQuery(int.Parse(txtContactID.Text));
contactsTableAdapter.Fill(dictionaryDataSet.Contacts);
```

۲۲ صفحه ۲۴۰ کارگاه ۱۶

با استفاده از دکمه‌ی Query Builder دستور SQL ایجادشده را طوری تغییر دهید که ویرایش رکورد بر اساس کد کالا انجام شود. ۳ ـ متد رویداد کلیک دکمه‌ی «ویرایش» را بنویسید. برنامه را اجرا کنید و با انتخاب یک رکورد عمل ویرایش را انجام دهید.

فعالیت منزل: برنامه‌ی درج کالای جدید را طوری تغییر دهید که اگر کالا در جدول موجود بود، فقط موجودی کالا را افزایش دهد. برای جدول مشتری برنامه‌ی ویرایش رکورد را بنویسید. برای جدول مخاطبان برنامه‌ی ویرایش مخاطب را بنویسید.

```
UPDATE Goods
SET GoodsName = ?, GoodsUnitPrice = ?, GoodsStock = ?
WHERE (GoodsCode = ?)
```

```
private void btnUpdateGoods_Click(object sender, EventArgs e)
{
    if (txtGoodsCode.TextLength > 0 && txtGoodsName.TextLength > 0 &&
        txtUnitPrice.TextLength > 0)
    {
        goodsTableAdapter.UpdateQuery(txtGoodsName.Text,
            int.Parse(txtUnitPrice.Text), int.Parse(txtStock.Text),
            int.Parse(txtGoodsCode.Text));
        goodsTableAdapter.Fill(myShopDataSet.Goods);
        MessageBox.Show("Updated successfully");
    }
}
```

نکته: ترتیب آرگومان‌های UpdateQuery باید با ترتیب علامت‌های «؟» در پرسوجو یکی باشد (نام، قیمت، موجودی و در پایان کد کالا در شرط WHERE). آزمایش sqlite: دستور UPDATE با شرط کد ۱ رکورد را به (Pencil، ۴۰۰۰، ۲۰) تغییر داد. توجه: ترتیب ورودی‌ها در متن چاپی کتاب (کد ابتدا) به‌هم‌ریخته است؛ در پرسوجوی بالا کد در انتها آمده است.

فعالیت منزل ۱ ـ اگر کالا موجود بود فقط موجودی افزایش یابد:

```
int code = int.Parse(txtGoodsCode.Text);
goodsTableAdapter.FillByGoodsCode(myShopDataSet.Goods, code);
if (myShopDataSet.Goods.Rows.Count > 0)
{
    int stock = int.Parse(myShopDataSet.Goods.Rows[0]["GoodsStock"].ToString());
    goodsTableAdapter.UpdateStockQuery(stock + int.Parse(txtStock.Text), code);
}
else
    goodsTableAdapter.InsertQuery(code, txtGoodsName.Text,
        int.Parse(txtUnitPrice.Text), int.Parse(txtStock.Text));
goodsTableAdapter.Fill(myShopDataSet.Goods);
```

که UpdateStockQuery برابر UPDATE Goods SET GoodsStock = ? WHERE GoodsCode = ? است. برای مشتری و مخاطبان: همین الگو با پرسوجوی UPDATE Customers SET ... WHERE CustomerCode = ? و UPDATE Contacts SET ... WHERE ContactID = ? و فراخوانی UpdateQuery سپس Fill.

این جواب ممکن است دقیق نباشد. (نام کنترل‌ها و فیلدهای جدول‌های مشتری و مخاطبان در کتاب مشخص نشده و پیشنهادی است.)

۲۳ صفحه ۲۴۱ کارگاه ۱۷

۲ ـ ... کاربرد ویژگی‌های ImageLayout و FillWeight چیست؟

۴ ـ متد FillByGoodsCode را برای پرسوجو بر اساس کد کالا ایجاد کنید. ۵ ـ متد رویداد KeyDown_txtFGoodsCode را بنویسید.

۶ ـ متد رویداد کلیک دکمه‌ی انتقال به سربرگ کالا را بنویسید. ۷ ـ متد رویداد کلیک دکمه‌ی «درج در برگه خرید» سربرگ کالا را بنویسید.

۸ ـ متد رویداد کلیک دکمه‌ی جستجوی مشتری (btnCustomerTab) در سربرگ «برگه خرید» را بنویسید. ۹ ـ متد رویداد کلیک دکمه‌ی «درج در برگه خرید» در سربرگ مشتری را بنویسید.

۱۰ ـ متد رویداد کلیک دکمه‌ی «درج در سبد خرید» را بنویسید. چگونه می‌توان تعداد سطرهای یک DataGridView را تعیین کرد؟ عملکرد متد Add کنترل dgvFactors چیست؟ قیمت کل در کدام خانه‌ی dgvFactors قرار می‌گیرد و چگونه محاسبه می‌شود؟

۱۱ ـ کد مرحله‌ی ۱۰ را تغییر دهید تا در صورت خالی بودن کادرمتن کد کالا، خطا رخ ندهد. ۱۲ ـ محاسبه‌ی مبلغ قابل پرداخت را اضافه کنید و در lblAllCost قرار دهید.

۱۳ ـ متد حذف سطر را برای کلیک روی علامت حذف سطر بنویسید.

---

۲ ـ ImageLayout نحوه‌ی نمایش تصویر در سلول را مشخص می‌کند (None، Tile، Center، Stretch، Zoom)؛ FillWeight وزن (سهم) پهنای ستون نسبت به ستون‌های دیگر را در حالت AutoSizeColumnsMode = Fill تعیین می‌کند.

```
SELECT GoodsCode, GoodsName, GoodsUnitPrice, GoodsStock
FROM Goods
WHERE (GoodsCode = ?)       -- FillByGoodsCode
```

```
private void txtFGoodsCode_KeyDown(object sender, KeyEventArgs e)
{
    lblPrice.Text = " ";
    lblGoodsName.Text = " ";
    if (e.KeyCode == Keys.Enter)
        if (txtFGoodsCode.Text.Length > 0)
        {
            goodsTableAdapter.FillByGoodsCode(myShopDataSet.Goods,
                int.Parse(txtFGoodsCode.Text));
            if (myShopDataSet.Goods.Rows.Count > 0)
            {
                lblGoodsName.Text = myShopDataSet.Goods.Rows[0]["GoodsName"].ToString();
                lblPrice.Text     = myShopDataSet.Goods.Rows[0]
["GoodsUnitPrice"].ToString();
                lblStock.Text     = myShopDataSet.Goods.Rows[0]
["GoodsStock"].ToString();
            }
        }
}
```

```
private void btnGoodsTab_Click(object sender, EventArgs e)
{
    tabShop.SelectedTab = tabPageGoods;   // activate Goods tab
    txtSGoodsName.Focus();
}
private void btnInsertGoodsFactor_Click(object sender, EventArgs e)
{
    tabShop.SelectedTab = tabPageFactor;  // activate Factor tab
    txtFGoodsCode.Text = txtGoodsCode.Text;
    lblGoodsName.Text = txtGoodsName.Text;
    lblPrice.Text = txtUnitPrice.Text;
    txtFGoodsCode.Focus();
}
private void btnCustomerTab_Click(object sender, EventArgs e)
{
    tabShop.SelectedTab = tabPageCustomers;
}
private void btnInsertCustomerFactor_Click(object sender, EventArgs e)
{
    tabShop.SelectedTab = tabPageFactor;
    lblCustomerCode.Text = txtCustomerCode.Text;
    lblCustomerName.Text = txtCustomerName.Text;
}
```

ایران توشه
توشه ای برای موفقیت

```
private void btnInsertGoods_Click(object sender, EventArgs e)
{
    if (txtFGoodsCode.TextLength == 0 || txtFAmount.TextLength == 0)
        return;                          // step 11: avoid the error
    int n = dgvFactors.Rows.Count - 1;   // index of the new row
    dgvFactors.Rows.Add();
    dgvFactors.Rows[n].Cells[0].Value = txtFGoodsCode.Text;
    dgvFactors.Rows[n].Cells[1].Value = lblGoodsName.Text;
    dgvFactors.Rows[n].Cells[2].Value = txtFAmount.Text;
    dgvFactors.Rows[n].Cells[3].Value = lblPrice.Text;
    dgvFactors.Rows[n].Cells[4].Value =
        int.Parse(txtFAmount.Text) * int.Parse(lblPrice.Text);
    // step 12: payable amount
    int sum = 0;
    for (int i = 0; i < dgvFactors.Rows.Count; i++)
        if (dgvFactors.Rows[i].Cells[4].Value != null)
            sum += int.Parse(dgvFactors.Rows[i].Cells[4].Value.ToString());
    lblAllCost.Text = sum.ToString();
}
```

تعداد سطرها با dgvFactors.Rows.Count به‌دست می‌آید (در حالتی که سطر خالی جدید فعال است، یکی بیشتر از رکوردهای واقعی است؛ برای همین 1 – Count نوشته شده). متد Add یک سطر خالی جدید به انتهای جدول اضافه می‌کند. قیمت کل در خانه‌ی شماره‌ی ۴ (ستون پنجم) هر سطر قرار می‌گیرد و از حاصل‌ضرب تعداد (ستون ۲) در قیمت واحد (ستون ۳) محاسبه می‌شود.

```
private void dgvFactors_CellContentClick(object sender,
                                         DataGridViewCellEventArgs e)
{
    if (dgvFactors.Columns[e.ColumnIndex] is DataGridViewImageColumn
        && e.RowIndex >= 0)
        dgvFactors.Rows.RemoveAt(e.RowIndex);
}
```

اپراتور is بررسی می‌کند ستونِ کلیک‌شده از نوع ستون تصویر (علامت حذف) باشد؛ در این حالت سطر انتخاب‌شده با RemoveAt حذف می‌شود. توجه: برای استفاده از Rows.Add باید DataGridView مقید نباشد.

ایران توشه
توشه ای برای موفقیت

این جواب ممکن است دقیق نباشد. (مراحل ۸ و ۹ در کتاب فقط توضیح متنی داشتند؛ کد آنها بر اساس همان توضیحات و نام‌های پیشنهادی نوشته شد.)

---

۲۴ صفحه ۲۴۵ کارگاه ۱۸

۲ ـ در سربرگ کالا، کالایی را ثبت کنید که کد کالای آن قبلاً ثبت شده باشد. با درج کد تکراری چه اتفاقی رخ می‌دهد؟

برای اینکه بدانیم هنگام عملیات مبنایی چه خطاهایی ممکن است رخ دهد، جدول (نام عمل ـ فهرست خطاها) را کامل کنید.

۳ ـ روی دکمه‌ی ثبت در سربرگ «کالا» دابل‌کلیک کنید و کدها را با try-catch ویرایش کنید. برنامه را اجرا کنید و کالایی را ثبت کنید که کد کالای آن تکراری باشد، چه اتفاقی رخ می‌دهد؟

۴ ـ مدیریت خطا را برای عملیات دیگری در سربرگ کالا انجام دهید. آیا دستور if در برنامه عمل کنترل خطا را انجام می‌دهد؟ اگر کد کالا غیرعددی نوشته شود، آیا خطایی رخ می‌دهد؟

۲ ـ چون GoodsCode کلید اصلی است، پایگاه داده درج رکورد تکراری را نمی‌پذیرد و خطای زمان اجرا (OleDbException: تغییرات به‌دلیل ایجاد مقدار تکراری در کلید اصلی/اندیس انجام نشد) رخ می‌دهد و برنامه بدون مدیریت خطا متوقف می‌شود. (در آزمایش sqlite نیز خطای UNIQUE constraint failed: Goods.GoodsCode مشاهده شد.)

خطاهای محتمل در عملیات مبنایی

نام عمل	فهرست خطاها
درج	• خطای ناشی از ساختار نادرست دستور درج • درج مقدار تکراری برای کلید اصلی • خالی‌ماندن فیلدی که نباید Null باشد • ناسازگاری نوع داده (مثلاً متن به‌جای عدد)
حذف	• خطای ناشی از ساختار نادرست دستور حذف • حذف رکوردی که در جدول دیگر به آن ارجاع داده شده (نقض یکپارچگی ارجاعی) • نبود اتصال به پایگاه داده
ویرایش	• خطای ناشی از ساختار نادرست دستور ویرایش • خالی‌ماندن فیلدی که نباید Null باشد • تغییر کلید اصلی به مقدار تکراری

```
private void btnAddGoods_Click(object sender, EventArgs e)
{
    if (txtGoodsCode.TextLength > 0 && txtGoodsName.TextLength > 0 &&
        txtUnitPrice.TextLength > 0)
    {
        try
        {
            goodsTableAdapter.InsertQuery(int.Parse(txtGoodsCode.Text),
                txtGoodsName.Text, int.Parse(txtUnitPrice.Text),
                int.Parse(txtStock.Text));
            goodsTableAdapter.Fill(myShopDataSet.Goods);
            MessageBox.Show("Saved successfully");
        }
        catch
        {
            MessageBox.Show("Error in saving the goods");
        }
    }
}
```

اکنون با ثبت کد تکراری برنامه متوقف نمی‌شود، بلکه پیام «خطا در ثبت کالا» نمایش داده شده و برنامه ادامه می‌یابد؛ رکورد تکراری هم ثبت نمی‌شود. برای حذف و ویرایش نیز دستورهای UpdateQuery/DeleteQuery و Fill را داخل try می‌گذاریم و در catch پیام «خطا در حذف/ویرایش کالا» می‌دهیم.

دستور if فقط خالی‌نبودن کادرها را بررسی می‌کند و کنترل کامل خطا نیست. اگر کد کالا غیرعددی باشد، int.Parse با FormatException خطا می‌دهد؛ در حالت بدون try-catch برنامه متوقف می‌شود و با try-catch پیام خطا نمایش داده می‌شود.

۲۵ صفحه ۲۴۶ کارگاه ۱۹

فعالیت منزل: مدیریت خطا را برای کدهای سربرگ مشتری با استفاده از متد TryParse انجام دهید.

فعالیت کارگاهی (فیلم «ذخیره برگه خرید در برنامه فروشگاه»): پس از مشاهده‌ی فیلم، برنامه‌ی جستجوی مشتری و کالا و درج در سبد خرید و ثبت برگه خرید در سربرگ را بنویسید.

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

TryParse بدون ایجاد استثنا بررسی می‌کند متن عدد است یا نه (خروجی bool) و در صورت موفقیت عدد را در پارامتر out می‌گذارد:

```
private void btnAddCustomer_Click(object sender, EventArgs e)
{
    int code;
    if (int.TryParse(txtCustomerCode.Text, out code)
        && txtCustomerName.TextLength > 0)
    {
        try
        {
            customersTableAdapter.InsertQuery(code, txtCustomerName.Text,
                                              txtPhone.Text);
            customersTableAdapter.Fill(myShopDataSet.Customers);
        }
        catch
        {
            MessageBox.Show("Error in saving the customer");
        }
    }
    else
        MessageBox.Show("Customer code must be a number");
}
```

ثبت برگه خرید (طراحی خودمان به‌جای فیلم): ۱) با کد مشتری و Enter نام مشتری پر می‌شود؛ ۲) با کد کالا و Enter نام و قیمت کالا نشان داده می‌شود؛ ۳) با «درج در سبد خرید» سطر به dgvFactors افزوده و مبلغ کل محاسبه می‌شود؛ ۴) با «ثبت برگه خرید»، یک رکورد در Factors و برای هر سطر یک رکورد در FactorItems درج می‌شود و موجودی کالاها کم می‌شود:

```
private void btnSaveFactor_Click(object sender, EventArgs e)
{
    if (dgvFactors.Rows.Count <= 1) return;     // cart is empty
    int factorCode = int.Parse(txtFactorCode.Text);
    try
    {
        factorsTableAdapter.InsertQuery(factorCode,
            int.Parse(lblCustomerCode.Text), DateTime.Now.ToString("yyyy/MM/dd"));
        for (int i = 0; i < dgvFactors.Rows.Count - 1; i++)
        {
            int goods  = int.Parse(dgvFactors.Rows[i].Cells[0].Value.ToString());
            int amount = int.Parse(dgvFactors.Rows[i].Cells[2].Value.ToString());
            int price  = int.Parse(dgvFactors.Rows[i].Cells[3].Value.ToString());
            factorItemsTableAdapter.InsertQuery(factorCode, goods, amount, price);
            goodsTableAdapter.UpdateStockByCode(amount, goods); // stock = stock -
amount
        }
        dgvFactors.Rows.Clear();
        lblAllCost.Text = "0";
        MessageBox.Show("Invoice saved");
    }
    catch
    {
        MessageBox.Show("Error in saving the invoice");
    }
}
```

سربرگ کاربران: برای محدود کردن دسترسی، پس از ورود موفق، فیلد TypeOfAccess کاربر نگه‌داشته می‌شود و سربرگ فقط برای مدیر فعال می‌شود:

```
// LoginForm: public string access;  -> set from myShopDataSet.Users.Rows[0]
["TypeOfAccess"]
if (frmLogin.access != "Admin")
    tabShop.TabPages.Remove(tabPageUsers);   // hide Users tab
```

در این سربرگ یک DataGridView برای جدول Users و دکمه‌های «ثبت»، «حذف» و «ویرایش» مانند کارگاه‌های ۱۴ تا ۱۶ (با usersTableAdapter.InsertQuery، DeleteQuery، UpdateQuery) نوشته می‌شود.

این جواب ممکن است دقیق نباشد. (فیلم «ذخیره برگه خرید» در دسترس نبود؛ برنامه بر اساس ساختار جدول‌ها طراحی شد و نام متدهای TableAdapter پیشنهادی است.)

---

۲۶ صفحه ۲۴۷ پروژه فاز ۵

در این پودمان آرایه‌های ایجادشده برای نگهداری اطلاعات را حذف کرده و اطلاعات را از پایگاه داده دریافت کرده و در پایگاه داده ذخیره می‌کنیم. پروژه را به پایگاه داده‌ی کتابخانه که در پودمان ۱ ایجاد کردید، متصل کنید. موارد زیر را به پروژه اضافه کنید: ثبت زمان login کردن مدیر در یک پرونده به نام Log.txt همراه با مدیریت خطا؛ اضافه کردن دکمه «دریافت پرونده» و کادر متنی برای دریافت مسیر ذخیره‌سازی پرونده (در صورت کلیک روی دکمه «دریافت پرونده»، پرونده‌ی Log.txt را در مسیر تعیین‌شده در کادر متن ذخیره کند)؛

اضافه کردن دو سربرگ نمایش اعضا و نمایش کتاب‌ها که داده‌های جدول اعضا و کتاب‌ها در این سربرگ مشاهده شود؛ فراهم کردن امکان حذف اعضا و ویرایش مشخصات آنها؛ فراهم کردن امکان حذف کتاب و ویرایش عنوان آنها.

گام‌ها: ۱) با Add New Data Source پایگاه داده‌ی کتابخانه (جدول‌های Members و Books از پودمان ۱) به پروژه متصل می‌شود و libraryDataSet، membersBindingSource/membersTableAdapter و booksBindingSource/booksTableAdapter ساخته می‌شوند. ۲) آرایه‌های قبلی حذف و به‌جای آنها Fill استفاده می‌شود. ۳) پرسوجوهای DeleteQuery و UpdateQuery برای هر دو جدول با Query در طراح xsd ساخته می‌شوند.

ثبت زمان ورود مدیر در Log.txt (با مدیریت خطا) و دریافت پرونده:

```
// after a successful admin login
private void WriteLog()
{
    try
    {
        string line = String.Format("{0}\t{1}\r\n",
            txtUserName.Text, DateTime.Now);
        File.AppendAllText("Log.txt", line);
    }
    catch
    {
        MessageBox.Show("Could not write the log file");
    }
}

private void btnGetFile_Click(object sender, EventArgs e)
{
    try
    {
        string dest = Path.Combine(txtSavePath.Text, "Log.txt");
        File.Copy("Log.txt", dest, true);   // true = overwrite
        MessageBox.Show("File saved");
    }
    catch
    {
        MessageBox.Show("Invalid path or file not found");
    }
}
```

سربرگ‌های «نمایش اعضا» و «نمایش کتاب‌ها»: هر کدام یک DataGridView (dgvMembers و dgvBooks) دارند که با Fill پر می‌شوند:

ایران توشه
توشه ای برای موفقیت

```
private void MainForm_Load(object sender, EventArgs e)
{
    membersTableAdapter.Fill(libraryDataSet.Members);
    dgvMembers.DataSource = membersBindingSource;
    booksTableAdapter.Fill(libraryDataSet.Books);
    dgvBooks.DataSource = booksBindingSource;
}

private void btnDeleteMember_Click(object sender, EventArgs e)
{
    if (txtMemberCode.TextLength > 0)
        try
        {
            membersTableAdapter.DeleteQuery(int.Parse(txtMemberCode.Text));
            membersTableAdapter.Fill(libraryDataSet.Members);
        }
        catch { MessageBox.Show("Error in deleting the member"); }
}
private void btnUpdateMember_Click(object sender, EventArgs e)
{
    try
    {
        membersTableAdapter.UpdateQuery(txtMemberName.Text, txtMemberPhone.Text,
            int.Parse(txtMemberCode.Text));
        membersTableAdapter.Fill(libraryDataSet.Members);
    }
    catch { MessageBox.Show("Error in editing the member"); }
}
private void btnDeleteBook_Click(object sender, EventArgs e)
{
    try
    {
        booksTableAdapter.DeleteQuery(int.Parse(txtBookCode.Text));
        booksTableAdapter.Fill(libraryDataSet.Books);
    }
    catch { MessageBox.Show("Error in deleting the book"); }
}
private void btnUpdateBook_Click(object sender, EventArgs e)
{
    try
    {
        booksTableAdapter.UpdateTitleQuery(txtBookTitle.Text,
            int.Parse(txtBookCode.Text));
        booksTableAdapter.Fill(libraryDataSet.Books);
    }
    catch { MessageBox.Show("Error in editing the book"); }
}
```

پرسوجوها: DELETE و UPDATE روی Members و Books با شرط کلید اصلی (WHERE ... = ?). حذف عضو دارای امانت فعال در catch مدیریت می‌شود.

این جواب ممکن است دقیق نباشد. (نام جدول‌ها و فیلدهای پایگاه داده‌ی کتابخانه به طراحی هنرجو در پودمان ۱ بستگی دارد؛ نام‌ها نمونه‌اند.)

ایران توشه
توشه ای برای موفقیت