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 چهارم

عنوان فصل: توسعه‌ی واسط گرافیکی

از صفحه‌ی ۱۵۳ تا صفحه‌ی ۲۰۴ کتاب

در این فایل پاسخ فعالیت‌ها، کارهای کلاسی، تمرین‌ها و پرسش‌های فصل توسعه‌ی واسط گرافیکی آمده است.

مباحث فصل: رویدادهای ماوس، منوی نواری و منوی زمینه، رویدادهای صفحه‌کلید، نوع داده‌ی char و کلاس String

تهیه‌شده در ایران‌توشه، توشه‌ای برای موفقیت

irantooshe.ir

فهرست پرسش‌ها

۱ صفحه ۱۵۵ · فعالیت گروهی — عملکرد ماوس در ویندوز و برنامه‌های کاربردی ۸

۲ صفحه ۱۵۶ · فعالیت گروهی — جدول ۱: رویدادهای ماوس وابسته به دکمه ۸

۳ صفحه ۱۵۶ · کارگاه ۱ — واکنش برنامه به رویدادهای ماوس (MouseEvent) ۹

۴ صفحه ۱۵۸ · کارگاه ۲ — MouseDown و MouseUp (ParkCar) ۱۰

۵ صفحه ۱۵۹ · فعالیت منزل — حرکت ماشین به نسبت زمان نگه‌داشتن دکمه ۱۱

۶ صفحه ۱۵۹ · کنجکاوی — تغییر شکل نمادها با حرکت ماوس ۱۲

۷ صفحه ۱۶۰ · کارگاه ۳ — MouseEnter و MouseLeave (EnglishLearning) ۱۲

۸ صفحه ۱۶۱ · فعالیت کارگاهی — جدول ۲: رویدادهای ماوس مستقل از دکمه ۱۳

۹ صفحه ۱۶۲ · فعالیت منزل — دکمه‌ی گرافیکی سه‌حالته و آموزش واژگان فارسی ۱۳

ایران توشه
توشه ای برای موفقیت

۱۰ صفحه ۱۶۳ · کارگاه ۴ — آرگومان‌های رویدادهای ماوس (MouseSimulation) ۱۴
۱۱ صفحه ۱۶۴ · فعالیت کارگاهی — نمایش واژه‌ی Computer و مختصات ماوس ۱۵
۱۲ صفحه ۱۶۴ · فعالیت منزل — شبیه‌ساز ماوس: راست‌دست و چپ‌دست ۱۶
۱۳ صفحه ۱۶۴ · فعالیت کارگاهی — جدول ۳: ویژگی‌های آرگومان e ۱۶
۱۴ صفحه ۱۶۵ · پژوهش — تخصیص متد رویداد با برنامه‌نویسی و دستور as ۱۷
۱۵ صفحه ۱۶۵ · فعالیت منزل — یک متد رویداد برای همه‌ی برچسب‌ها ۱۷
۱۶ صفحه ۱۶۵ · فیلم — بازی سرعت کلیک ۱۸
۱۷ صفحه ۱۶۶ · فیلم — بازی اعصاب‌سنج ۱۹
۱۸ صفحه ۱۶۷ · کنجکاوی — تفاوت منوها ۲۰
۱۹ صفحه ۱۶۸ · فعالیت گروهی — منوها و کلیدهای میان‌بر در ویندوز ۲۰
۲۰ صفحه ۱۶۹ · کارگاه ۶ — ایجاد منو با MenuStrip (MenuBar) ۲۱

ایران توشه
توشه ای برای موفقیت

۲۱ صفحه ۱۷۰ · کارگاه ۷ — سفارشی کردن منو ۲۲

۲۲ صفحه ۱۷۰ · فعالیت منزل — طراحی منوی Notepad ۲۳

۲۳ صفحه ۱۷۱ · فعالیت کارگاهی — جدول ۴: ویژگی‌های کنترل منو ۲۳

۲۴ صفحه ۱۷۱ · کنجکاوی — جهت نمایش متن گزینه‌های منو ۲۴

۲۵ صفحه ۱۷۲ · پژوهش — رویدادهای دیگر منو ۲۴

۲۶ صفحه ۱۷۲ · کارگاه ۸ — رویداد برای گزینه‌های منو (TextEditor) ۲۵

۲۷ صفحه ۱۷۳ · فعالیت کارگاهی — منوی استاندارد ۲۷

۲۸ صفحه ۱۷۴ · کارگاه ۹ — طراحی منوی زمینه (ContextMenuStrip) ۲۷

۲۹ صفحه ۱۷۵ · کارگاه ۱۰ — پروژه‌ی منودار چندرسانه‌ای (MyIran) ۲۸

۳۰ صفحه ۱۸۰ · فعالیت کارگاهی — عملکرد صفحه‌کلید در ویندوز ۳۰

۳۱ صفحه ۱۸۱ · کارگاه ۱ — رویدادهای صفحه‌کلید برای فرم (KeyBoardEvent1) ۳۱

ایران توشه
توشه ای برای موفقیت

۳۲ صفحه ۱۸۲ · کارگاه ۲ — ترتیب رخداد رویدادهای صفحه‌کلید (KeyboardEvent2) ۳۲

۳۳ صفحه ۱۸۴ · کارگاه ۳ — رویداد KeyDown و KeyUp (ShootBall) ۳۲

۳۴ صفحه ۱۸۵ · کارگاه ۴ — آرگومان e در KeyDown و KeyPress (AsciiCode) ۳۳

۳۵ صفحه ۱۸۶ · کارگاه ۴ — ویژگی KeyData و رویداد KeyPress ۳۴

۳۶ صفحه ۱۸۷ · کنجکاوی — جدول ۶ و ویژگی‌های آرگومان e در KeyPress ۳۵

۳۷ صفحه ۱۸۷ · فعالیت منزل — کد حروف فارسی ۳۶

۳۸ صفحه ۱۸۹ · کارگاه ۵ — کلیدهای ترکیبی (ZoomPicture) ۳۶

۳۹ صفحه ۱۹۰ · فعالیت منزل — برنامه‌ی کارگاه ۵ با KeyData ۳۷

۴۰ صفحه ۱۹۰ · کارگاه ۶ — فوکوس (Focus) و KeyPreview ۳۸

۴۱ صفحه ۱۹۱ · فعالیت منزل — فیلم: برنامه‌ی یافتن نویسه در صفحه‌کلید ۳۸

۴۲ صفحه ۱۹۲ · کارگاه ۷ — نوع داده‌ی شمارشی Keys (MoveSpacecraft) ۳۹

ایران توشه
توشه ای برای موفقیت

۴۳ صفحه ۱۹۳ · کارگاه ۸ — انتقال فوکوس با کلید Enter (FocusByEnter) ۴۰
۴۴ صفحه ۱۹۴ · پژوهش — تغییر رنگ زمینه‌ی کادر متن با فوکوس ۴۱
۴۵ صفحه ۱۹۵ · کارگاه ۹ — متدهای نوع داده‌ی char (FilterInputData) ۴۱
۴۶ صفحه ۱۹۶ · کارگاه ۹ — فیلتر حروف با KeyDown و KeyPress ۴۲
۴۷ صفحه ۱۹۷ · فعالیت کارگاهی — جدول ۷: متدهای نوع داده‌ی char ۴۳
۴۸ صفحه ۱۹۸ · کنجکاوی — کلاس String ۴۳
۴۹ صفحه ۱۹۹ · کارگاه ۱۰ — بررسی متدهای رشته‌ای (StringMethod) ۴۴
۵۰ صفحه ۲۰۰ · کارگاه ۱۱ — متدهای CompareTo و Trim (LoginApp) ۴۵
۵۱ صفحه ۲۰۱ · فعالیت کارگاهی — جدول ۸: شرح متدهای رشته‌ای ۴۶
۵۲ صفحه ۲۰۱ · فیلم — برنامه‌ی جست‌وجو و جایگزینی واژه (کارگاه ۱۲) ۴۸
۵۳ صفحه ۲۰۲ · پروژه فاز ۴ — منو برای برنامه‌ی کتابخانه ۴۹

ایران توشه
توشه ای برای موفقیت

۱ صفحه ۱۵۵ فعالیت گروهی

برای بررسی عملکرد ماوس در ویندوز و برنامه‌های کاربردی به پرسش‌های زیر پاسخ دهید:

برای اجرای یک برنامه در نوار وظیفه چگونه از ماوس استفاده می‌کنید؟

برای اجرای یک برنامه از روی میزکار چگونه از ماوس استفاده می‌کنید؟

برای ایجاد نسخه‌ی مشابه یک پرونده یا پوشه چگونه از ماوس استفاده می‌کنید؟

وقتی که اشاره‌گر ماوس روی نمادهای میزکار قرار می‌گیرد، آیا شکل نماد تغییر می‌کند؟

در برنامه‌ی نقاشی ویندوز برای ترسیم شکل چگونه از ماوس استفاده می‌کنید؟

به کمک هم‌گروهی‌های خود کاربردهای دیگر ماوس در ویندوز و برنامه‌های کاربردی را پیدا کنید و در جدول زیر بنویسید.

- نوار وظیفه: یک بار کلیک چپ روی نماد برنامه.
- میزکار: دوبار کلیک چپ (دابل‌کلیک) روی نماد برنامه.
- نسخه‌ی مشابه: نگه داشتن کلید Ctrl و کشیدن (Drag) پرونده یا پوشه با ماوس، یا راست‌کلیک ← Copy و سپس Paste.
- بله؛ با قرار گرفتن اشاره‌گر روی نماد، ظاهر نماد (پس‌زمینه‌ی آن) تغییر می‌کند و روشن یا برجسته می‌شود.
- در نقاشی: با نگه داشتن دکمه‌ی چپ و حرکت دادن ماوس (کشیدن) شکل ترسیم می‌شود.

کاربردهای ماوس

کاربرد ماوس در ویندوز	کاربرد ماوس در برنامه‌های کاربردی
انتخاب و اجرای نمادها و پنجره‌ها با کلیک و دابل‌کلیک	انتخاب ابزار و ترسیم شکل (کشیدن) در برنامه‌ی نقاشی
جابه‌جایی پنجره‌ها و تغییر اندازه‌ی آنها (کشیدن)	انتخاب متن در برنامه‌ی واژه‌پرداز (کشیدن یا دابل‌کلیک)
باز کردن منوی میان‌بر با راست‌کلیک	کلیک روی دکمه‌ها و گزینه‌های منو در نرم‌افزار
نمایش راهنمای ابزار (ToolTip) با نگه داشتن اشاره‌گر روی نماد	هدف‌گیری و شلیک در بازی‌های رایانه‌ای
تغییر شکل اشاره‌گر (دست، ساعت شنی، ...)	بزرگ‌نمایی و جابه‌جایی با چرخ ماوس

۲ صفحه ۱۵۶ فعالیت گروهی

شرح رویدادهای جدول ۱ را با توجه به شرح آنها در VS بنویسید.

ایران توشه
توشه ای برای موفقیت

با انجام عملیات زیر به‌وسیله‌ی ماوس چه رویدادی رخ می‌دهد؟ (نگه‌داشتن دکمه‌ی راست یا چپ؛ رها کردن دکمه‌ی راست یا چپ)

کدام رویداد ماوس وابسته به دکمه‌ی دیگری را قبلاً استفاده کرده‌اید؟

جدول ۱

نام رویداد	شرح رویداد
MouseClick	زمانی رخ می‌دهد که روی کنترل با ماوس کلیک شود.
MouseDoubleClick	زمانی رخ می‌دهد که روی کنترل با ماوس دابل‌کلیک (دو بار پشت سر هم کلیک) شود.
MouseDown	زمانی رخ می‌دهد که اشاره‌گر ماوس روی کنترل است و یکی از دکمه‌های ماوس فشرده می‌شود.
MouseUp	زمانی رخ می‌دهد که اشاره‌گر ماوس روی کنترل است و یکی از دکمه‌های ماوس رها می‌شود.

عملیات و رویداد

عملیات	رویداد
نگه‌داشتن (فشردن) دکمه‌ی راست یا چپ	MouseDown
رها کردن دکمه‌ی راست یا چپ	MouseUp

رویدادهای Click (برای دکمه‌ها) و MouseClick پیش‌تر استفاده شده‌اند.

این فایل توسط سایت ایران‌توشه تهیه شده است.

irantooshe.ir

۳ صفحه ۱۵۶ کارگاه ۱

۳ ـ دستوری در رویداد MouseClick بنویسید که رشته‌ی «به سی‌شارپ خوش‌آمدید» را در کادر پیام نمایش دهد. برنامه را اجرا کنید و روی دکمه کلیک کنید.

۴ ـ برای خروج از برنامه می‌توانید از متد Application.Exit() استفاده کنید. آیا روش دیگری می‌شناسید؟

۵ ـ در متد رویداد MouseDown دکمه‌ی btnRun کد زیر را بنویسید ... با نگه داشتن دکمه‌ی ماوس، رنگ دکمه‌ی btnRun چه تغییری می‌کند؟

۶ ـ در متد رویداد MouseUp دکمه‌ی btnRun رنگ دکمه را به حالت اول برگردانید.

```
private void btnRun_MouseClick(object sender, MouseEventArgs e)
{
    MessageBox.Show("به سی‌شارپ خوش‌آمدید");
}
private void btnRun_MouseDown(object sender, MouseEventArgs e)
{
    btnRun.BackColor = Color.Red;     // با نگه داشتن دکمه‌ی ماوس، رنگ دکمه قرمز می‌شود
}
private void btnRun_MouseUp(object sender, MouseEventArgs e)
{
    btnRun.BackColor = Color.Blue;    // بازگشت به رنگ زمینه‌ی اولیه (آبی)
}
private void Form1_DoubleClick(object sender, EventArgs e)
{
    Application.Exit();               // خروج با دابل‌کلیک روی فرم
}
```

روش دیگر خروج از برنامه استفاده از this.Close() (بستن فرم اصلی) است؛ Environment.Exit(0) هم برنامه را به پایان می‌رساند.

۴ صفحه ۱۵۸ کارگاه ۲

چرا در این برنامه نیاز به زمان‌سنج داریم؟

ویژگی Right چه مقداری را نگهداری می‌کند؟

برنامه‌ی پارک ماشین را اجرا کنید. با چه شرایطی، کاربر برنده می‌شود؟

پس از پایان یک بازی چگونه می‌توان بازی جدیدی را شروع کرد؟ برنامه‌ی آن را بنویسید.

حرکت ماشین باید تا زمانی که دکمه‌ی ماوس نگه داشته شده است به‌طور مداوم و تدریجی انجام شود. رویداد MouseDown فقط یک بار رخ می‌دهد؛ پس برای تکرار حرکت در فواصل زمانی کوتاه به زمان‌سنج نیاز است که با MouseDown فعال و با MouseUp غیرفعال می‌شود.

ایران توشه
توشه ای برای موفقیت

```
private void timer1_Tick(object sender, EventArgs e)
{
    picCar.Left += 5;
}
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    timer1.Enabled = true;
}
private void Form1_MouseUp(object sender, MouseEventArgs e)
{
    timer1.Enabled = false;
    if ((picCar.Right + 10 > picBarrier.Left) && (picCar.Right < picBarrier.Left))
        MessageBox.Show("آفرین، برنده شدی");
    else
        MessageBox.Show("متأسفانه، بازنده شدی");
}
```

ویژگی Right مختصات افقی لبه‌ی راست کنترل را (فاصله‌ی لبه‌ی راست کادر تصویر از سمت چپ فرم) نگه می‌دارد و برابر Left + Width است.

شرط برنده شدن: لبه‌ی راست ماشین باید کمتر از ۱۰ پیکسل با مانع فاصله داشته باشد و به مانع نرسیده باشد: picBarrier.Left – 10 < picCar.Right < picBarrier.Left.

برای شروع بازی جدید، مکان اولیه‌ی ماشین را بازمی‌گردانیم (مثلاً با یک دکمه‌ی «بازی جدید» یا پس از نمایش پیام):

```
private void btnNewGame_Click(object sender, EventArgs e)
{
    picCar.Left = 20;      // مکان اولیه‌ی ماشین
    timer1.Enabled = false;
}
```

۵ صفحه ۱۵۹ **فعالیت منزل**

بازی کارگاه ۲ را به‌صورت زیر تغییر دهید: بعد از رها کردن ماوس، ماشین به نسبت زمان نگه‌داشتن دکمه‌ی ماوس، حرکت کند. به‌طور مثال اگر کاربر یک ثانیه دکمه‌ی ماوس را نگه دارد، ماشین ۱۰۰ پیکسل حرکت کند.

با زمان‌سنج با Interval = 100، شمارنده‌ی ticks را در زمان نگه‌داشتن افزایش می‌دهیم (هر ثانیه ۱۰ تیک)؛ پس از رها کردن ماوس ماشین به اندازه‌ی ticks * 10 پیکسل (یک ثانیه ۱۰۰ پیکسل) حرکت می‌کند:

```
int ticks = 0;
private void timer1_Tick(object sender, EventArgs e)   // Interval = 100
{
    ticks++;
}
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    ticks = 0;
    timer1.Enabled = true;
}
private void Form1_MouseUp(object sender, MouseEventArgs e)
{
    timer1.Enabled = false;
    picCar.Left += ticks * 10;     // ۱۰ پیکسل = ۱۰۰ ثانیه = ۱ تیک
}
```

۶ صفحه ۱۵۹ کنجکاوی

با حرکت ماوس روی نمادهای برنامه در میزکار و نوار وظیفه چه تغییراتی در شکل نمادها مشاهده می‌کنید؟

لحظه‌ای ماوس را بدون آنکه کلیک کنید، روی یکی از نمادهای نوار وظیفه نگه دارید. چه مشاهده می‌کنید؟

با حرکت ماوس روی نماد، پس‌زمینه‌ی نماد برجسته (روشن یا رنگی) می‌شود. اگر ماوس چند لحظه روی نماد نوار وظیفه بماند، پیش‌نمایش پنجره‌ی برنامه یا متن راهنما (نام برنامه) نمایش داده می‌شود. اینها نمونه‌ی رویدادهای ماوس مستقل از دکمه (MouseEnter، MouseLeave و MouseHover) هستند.

۷ صفحه ۱۶۰ کارگاه ۳

چرا ویژگی Visible کادرتصویرها برابر با false است؟

۴ ـ در متد رویداد MouseLeave_lblBook دستوری بنویسید که picBook پنهان شود و رنگ lblBook مشکی شود.

۵ ـ متد رویداد MouseHover_lblBook را ایجاد کنید ... برنامه را اجرا کنید. عملکرد متد SetToolTip چیست؟

۶ ـ برای آموزش همه‌ی واژگان، برنامه را کامل کنید.

تصویر هر واژه باید در شروع برنامه دیده نشود و فقط وقتی اشاره‌گر ماوس روی واژه‌ی مربوط می‌آید نمایش داده شود؛ برای همین Visible = false است و در MouseEnter برابر true می‌شود.

ایران توشه
توشه ای برای موفقیت

```
private void lblBook_MouseEnter(object sender, EventArgs e)
{
    lblBook.ForeColor = Color.Red;
    picBook.Visible = true;
    lblBook.Cursor = Cursors.Hand;
}
private void lblBook_MouseLeave(object sender, EventArgs e)
{
    picBook.Visible = false;
    lblBook.ForeColor = Color.Black;
}
private void lblBook_MouseHover(object sender, EventArgs e)
{
    ToolTip tip = new ToolTip();
    tip.SetToolTip(this.lblBook, "کتاب");
}
// نیز سه متد مشابه (ساعت) lblClock و (قلم) lblPen، (ماشین) lblCar، (سیب) lblApple برای
نوشته می‌شود.
```

متد SetToolTip(control, text) متن راهنمای (ToolTip) را به کنترل مشخص‌شده نسبت می‌دهد تا وقتی ماوس چند لحظه روی آن ماند، متن در یک کادر مستطیلی کوچک (اینجا معنی فارسی واژه) نمایش داده شود.

۸ صفحه ۱۶۱ **فعالیت کارگاهی**

ستون شرح رویدادها در جدول ۲ را تکمیل کنید.

جدول ۲

نام رویداد	شرح رویداد
MouseEnter	زمانی رخ می‌دهد که ماوس به بخشی از کنترل (قسمت قابل‌مشاهده‌ی آن) وارد شود.
MouseLeave	زمانی رخ می‌دهد که اشاره‌گر ماوس از کنترل خارج شود.
MouseMove	زمانی رخ می‌دهد که اشاره‌گر ماوس روی کنترل حرکت داده شود.
MouseHover	زمانی رخ می‌دهد که اشاره‌گر ماوس برای چند لحظه روی کنترل ساکن بماند.

۹ صفحه ۱۶۲ **فعالیت منزل**

ـ در برنامه‌های کاربردی برخی از دکمه‌های گرافیکی سه حالت up، down و over دارند. به کمک کنترل PictureBox یک دکمه‌ی گرافیکی سه‌حالته طراحی کنید.

ـ برنامه‌ی آموزشی کارگاه ۳ را برای آموزش واژگان فارسی استفاده کنید. کاربر با ورود ماوس به روی تصویر، واژه‌ی فارسی معادل آن را ببیند.

سه تصویر هم‌اندازه (حالت عادی، روی ماوس، فشرده) به‌صورت فایل‌های up.png، over.png و down.png ساخته و در پوشه‌ی Debug قرار می‌دهیم. حالت‌ها را با رویدادهای ماوس تغییر می‌دهیم:

```
private void picButton_MouseEnter(object sender, EventArgs e)
{ picButton.ImageLocation = "over.png"; }
private void picButton_MouseLeave(object sender, EventArgs e)
{ picButton.ImageLocation = "up.png"; }
private void picButton_MouseDown(object sender, MouseEventArgs e)
{ picButton.ImageLocation = "down.png"; }
private void picButton_MouseUp(object sender, MouseEventArgs e)
{ picButton.ImageLocation = "over.png"; }
```

برای واژگان فارسی، رویداد MouseEnter هر کادر تصویر، برچسب فارسی همان واژه را نمایش می‌دهد و MouseLeave آن را پنهان می‌کند:

```
private void picBook_MouseEnter(object sender, EventArgs e)
{
    lblPersian.Text = "کتاب";
    lblPersian.Visible = true;
}
private void picBook_MouseLeave(object sender, EventArgs e)
{
    lblPersian.Visible = false;
}
```

۱۰ صفحه ۱۶۳ کارگاه ۴

۴ ـ قطعه کدی بنویسید که با رها کردن دکمه‌ی ماوس، رنگ دکمه‌ها مشکی شود.

ایران توشه
توشه ای برای موفقیت

۶ ـ کد مرحله‌ی ۵ را تغییر دهید تا به‌جای تصویر اشاره‌گر، شکل نمادین ماوس حرکت کند.

```
private void pnlMousePanel_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
        picLeft.BackColor = Color.White;
    else if (e.Button == MouseButtons.Middle)
        picMiddle.BackColor = Color.White;
    else if (e.Button == MouseButtons.Right)
        picRight.BackColor = Color.White;
}
private void pnlMousePanel_MouseUp(object sender, MouseEventArgs e)
{
    picLeft.BackColor = Color.Black;
    picMiddle.BackColor = Color.Black;
    picRight.BackColor = Color.Black;
}
private void pnlMousePanel_MouseMove(object sender, MouseEventArgs e)
{
    picCursor.Location = e.Location;    // حرکت تصویر اشاره‌گر
}
```

برای مرحله‌ی ۶ (حرکت «شکل نمادین ماوس» به‌جای تصویر اشاره‌گر) کافی است picCursor را به‌جای تصویر اشاره‌گر با تصویر کوچک شکل ماوس مقداردهی کنیم (ویژگی Image) و همان دستور picCursor.Location = e.Location را نگه داریم.

۱۱ صفحه ۱۶۴ **فعالیت کارگاهی**

ـ برنامه‌ای بنویسید که با حرکت ماوس در فضای فرم واژه‌ی Computer به همراه اشاره‌گر ماوس نمایش داده شود.

ـ برنامه را به‌گونه‌ای ویرایش کنید که به‌جای واژه‌ی Computer مختصات ماوس به‌صورت (x,y) به همراه اشاره‌گر ماوس نمایش داده شود.

ایران توشه
توشه ای برای موفقیت

```
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    lblText.Location = e.Location;        // lblText.Text = "Computer"
}

// حالت دوم: نمایش مختصات
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    lblText.Text = "(" + e.X + "," + e.Y + ")";
    lblText.Location = e.Location;
}
```

۱۲ صفحه ۱۶۴ فعالیت منزل

برنامه‌ی شبیه‌ساز را اجرا کنید. با راست‌کلیک، کلیک چپ، کلیک وسط، دابل‌کلیک و حرکت ماوس روی پنل ماوس، برنامه را آزمایش کنید.

به برنامه‌ی کارگاه ۴ امکان تعیین راست‌دست و چپ‌دست بودن کاربر را اضافه کرده، متناظر با آن دکمه‌های ماوس را جابه‌جا کنید.

با کلیک چپ، دکمه‌ی picLeft، با کلیک وسط picMiddle و با راست‌کلیک picRight سفید می‌شود و با رها کردن مشکی برمی‌گردد؛ با حرکت ماوس تصویر اشاره‌گر همراه ماوس حرکت می‌کند.

یک کادر علامت chkLeftHanded اضافه می‌کنیم. در حالت چپ‌دست، نقش دکمه‌ها جابه‌جا می‌شود:

```
private void pnlMousePanel_MouseDown(object sender, MouseEventArgs e)
{
    MouseButtons primary = chkLeftHanded.Checked ? MouseButtons.Right :
MouseButtons.Left;
    MouseButtons secondary = chkLeftHanded.Checked ? MouseButtons.Left :
MouseButtons.Right;
    if (e.Button == primary)
        picLeft.BackColor = Color.White;
    else if (e.Button == MouseButtons.Middle)
        picMiddle.BackColor = Color.White;
    else if (e.Button == secondary)
        picRight.BackColor = Color.White;
}
```

۱۳ صفحه ۱۶۴ فعالیت کارگاهی

در VS با قرار دادن اشاره‌گر ماوس روی هر ویژگی، نوع و عملکرد آن به‌صورت راهنما در اختیار برنامه‌نویس قرار می‌گیرد (شکل ۶). با توجه به این راهنما جدول ۳ را تکمیل کنید.

توشه ای برای موفقیت

جدول ۳ـ ویژگی‌های آرگومان e

نام ویژگی	نوع داده	شرح
Button	نوع داده‌ی شمارشی MouseButtons	مشخص می‌کند کدام دکمه‌ی ماوس فشار داده شده است (راست، چپ، وسط).
Location	ساختار Point	موقعیت اشاره‌گر ماوس روی کنترل را مشخص می‌کند.
X	int	مختصات افقی (x) اشاره‌گر ماوس هنگام رخداد رویداد را برمی‌گرداند.
Y	int	مختصات عمودی (y) اشاره‌گر ماوس هنگام رخداد رویداد را برمی‌گرداند.
Clicks	int	تعداد دفعاتی که دکمه فشرده و رها شده است (مثلاً ۲ برای دابل‌کلیک).

۱۴ صفحه ۱۶۵ پژوهش

ـ بررسی کنید چگونه می‌توان با برنامه‌نویسی، یک متد رویداد را برای یک کنترل تخصیص داد.

ـ کاربرد دستور روبه‌رو را بررسی کنید: Button btn = sender as Button;

با عملگر += می‌توان یک متد را به رویداد کنترل متصل کرد (معمولاً در سازنده‌ی فرم یا رویداد Load):

```
button1.Click += new EventHandler(MyClickMethod);   // یا ساده‌تر: button1.Click +=
MyClickMethod;
void MyClickMethod(object sender, EventArgs e) { MessageBox.Show("Clicked"); }
```

دستور Button btn = sender as Button; آرگومان sender (از نوع object) را به نوع Button تبدیل می‌کند تا بتوان به ویژگی‌های دکمه‌ای که رویداد برای آن رخ داده دسترسی داشت؛ تفاوت آن با (Button)sender این است که اگر تبدیل ممکن نباشد خطا نمی‌دهد و مقدار null برمی‌گرداند.

۱۵ صفحه ۱۶۵ فعالیت منزل

در برنامه‌ی کارگاه ۴ واحد یادگیری ۶ کد متد رویداد کلیک همه‌ی برچسب‌ها مشابه است. بنابراین می‌توانیم به‌جای ۹ متد رویداد کلیک برای برچسب‌ها، یک متد رویداد کلیک برای همه‌ی برچسب‌ها بنویسیم. ... کد زیر را در متد رویداد کلیک برچسب label1 وارد می‌کنیم.

در پنجره‌ی Properties برای رویداد Click برچسب‌های label2 تا label9 نام متد label1_Click را از فهرست کشویی انتخاب می‌کنیم؛ سپس:

ایران توشه
توشه‌ای برای موفقیت

```
private void label1_Click(object sender, EventArgs e)
{
    Label lbl = (Label)sender;      // برچسبی که روی آن کلیک شده
    lbl.Visible = false;            // همان برچسب پنهان می‌شود
}
```

هر بار که روی هر یک از ۹ برچسب کلیک شود، همین متد اجرا شده و فقط همان برچسب پنهان می‌شود؛ زیرا sender کنترلی است که رویداد برای آن رخ داده است.

۱۶ صفحه ۱۶۵ فیلم

پس از مشاهده‌ی فیلم، «بازی سرعت کلیک» را نوشته و اجرا کنید. (فیلم شماره‌ی ۱۱۰۶)

بازی سرعت کلیک: یک دکمه در مکان تصادفی فرم ظاهر می‌شود، کاربر باید با کلیک روی آن امتیاز بگیرد و زمان‌سنج بازی را پس از مدت معین پایان می‌دهد. (فیلم آموزشی در دسترس نبود؛ برنامه‌ی زیر طراحی شد.)

```
int score = 0, time = 20;
Random rnd = new Random();
private void btnStart_Click(object sender, EventArgs e)   // دکمه‌ی شروع
{
    score = 0; time = 20;
    lblScore.Text = "0"; lblTime.Text = "20";
    timer1.Interval = 1000; timer1.Enabled = true;
    btnTarget.Visible = true;
}
private void btnTarget_Click(object sender, EventArgs e)  // دکمه‌ی هدف
{
    score++;
    lblScore.Text = score.ToString();
    btnTarget.Left = rnd.Next(0, this.ClientSize.Width - btnTarget.Width);
    btnTarget.Top = rnd.Next(50, this.ClientSize.Height - btnTarget.Height);
}
private void timer1_Tick(object sender, EventArgs e)
{
    time--;
    lblTime.Text = time.ToString();
    if (time == 0)
    {
        timer1.Enabled = false;
        btnTarget.Visible = false;
        MessageBox.Show("پایان بازی! امتیاز شما: " + score);
    }
}
```

این جواب ممکن است دقیق نباشد.

ایران توشه
توشه ای برای موفقیت

فیلم را مشاهده کنید و پروژه‌ی مشابه ایجاد کرده، کدهای آن را بنویسید. (فیلم شماره‌ی ۱۱۰۷)

«بازی اعصاب‌سنج» را طراحی کنید. با سلیقه‌ی خودتان مسیرهای مشکل‌تری برای بازی ایجاد کنید. کادر تصویر لامپ را به پروژه اضافه کنید تا به هنگام خطای کاربر به‌صورت چشمک‌زن ظاهر شود. با خطای کاربر در بازی شکل اشاره‌گر ماوس تغییر کند.

ـ آیا می‌توان در بازی اعصاب‌سنج از رویدادهای MouseMove_Form یا MouseHover_Form برای بررسی خطای کاربر استفاده کرد؟

ـ اگر به‌جای رویداد MouseClick از رویداد Click استفاده کنیم، چه مشکلی پیش می‌آید؟ چه تفاوتی بین رویداد Click و MouseClick وجود دارد؟

طراحی: مسیر بازی را با چند برچسب lblWall1 تا lblWallN با رنگ زمینه‌ی سیاه (دیوارهای مسیر) روی فرم می‌کشیم؛ هدف (خط پایان) برچسب lblFinish و کادر تصویر لامپ picLamp با Visible = false است. اگر ماوس به یکی از دیوارها وارد شود (MouseEnter)، کاربر خطا کرده است.

```
private void Wall_MouseEnter(object sender, EventArgs e)   // برای همه‌ی دیوارها یک متد
{
    this.Cursor = Cursors.No;           // تغییر شکل اشاره‌گر
    picLamp.Visible = true;             // لامپ روشن می‌شود
    timerBlink.Enabled = true;          // Interval = 300 برای چشمک‌زدن
}
private void timerBlink_Tick(object sender, EventArgs e)
{
    picLamp.Visible = !picLamp.Visible;
}
private void lblFinish_MouseEnter(object sender, EventArgs e)
{
    MessageBox.Show("آفرین! به پایان رسیدید");
}
private void lblStart_MouseEnter(object sender, EventArgs e)
{
    this.Cursor = Cursors.Default;      // شروع دوباره
    timerBlink.Enabled = false;
    picLamp.Visible = false;
}
```

استفاده از MouseMove یا MouseHover فرم مناسب نیست: MouseMove روی کل فرم با هر حرکت رخ می‌دهد و تشخیص برخورد با دیوار ممکن نمی‌کند (و تشخیص باید برای هر دیوار جداگانه باشد)؛ MouseHover نیز فقط پس از ساکن ماندن ماوس رخ می‌دهد و برخورد لحظه‌ای را نمی‌بیند. MouseEnter روی هر دیوار بهترین گزینه است.

رویداد Click می‌تواند با صفحه‌کلید (مثلاً فشردن Enter یا Space روی دکمه‌ی دارای فوکوس) هم رخ دهد و آرگومان آن EventArgs است و اطلاعات ماوس (مثل Button و Location) را ندارد؛ ولی MouseClick فقط با کلیک ماوس رخ می‌دهد و آرگومان MouseEventArgs دارد که دکمه و مختصات ماوس را می‌دهد.

این جواب ممکن است دقیق نباشد.

۱۸ صفحه ۱۶۷ کنجکاوی

ـ چه تفاوتی بین منوهای بالا از لحاظ شکل و قالب و گروه‌بندی وجود دارد؟

ـ در کدام شکل، منو با راست‌کلیک فعال می‌شود؟

منوی نواری (Menu Bar) در بالای پنجره به‌صورت افقی چیده می‌شود و گزینه‌های اصلی (مثل پرونده، ویرایش، نما) هر یک زیرمنوی عمودی دارند که گزینه‌ها با خط جداکننده گروه‌بندی شده‌اند؛ منوی میان‌بر (Context Menu) در جای ماوس باز می‌شود و فقط شامل عملیات مرتبط با همان محل است و گزینه‌های آن به‌صورت یک فهرست عمودی بدون نوار اصلی هستند.

منوی میان‌بر (Context Menu) با راست‌کلیک فعال می‌شود؛ مثلاً منوی باز شده با راست‌کلیک روی میزکار یا یک پرونده.

۱۹ صفحه ۱۶۸ فعالیت گروهی

برای انجام چه عملیاتی در ویندوز از منوها استفاده می‌کنید؟ آیا این عملیات کلید میان‌بر دارند؟ آیا این منوها با کلید Alt فعال می‌شوند؟ یک پنجره باز کنید و کلید Alt را فشار دهید.

با راست‌کلیک در محیط‌های مختلف ویندوز، منوی یکسانی باز می‌شود؟

چند نمونه از این عملیات را در جدول زیر بنویسید و در صورت وجود کلید میان‌بر آن را مشخص کنید.

منوها برای باز کردن، ذخیره، چاپ، کپی، برش و چسباندن، جست‌وجو، حذف و تنظیمات استفاده می‌شوند. بیشتر عملیات کلید میان‌بر دارند. با فشردن Alt در بسیاری از پنجره‌ها (مانند Notepad یا File Explorer) منوی نواری فعال شده، حروف دسترسی (زیرخط‌دار) نمایان می‌شوند. منوی راست‌کلیک در محیط‌های مختلف ویندوز یکسان نیست؛ گزینه‌ها بر اساس شیء (میزکار، پوشه، پرونده، متن و ...) تغییر می‌کنند.

عملیات و کلید میان‌بر

نوع عملیات با منو	کلید میان‌بر
تهیه‌ی نسخه‌ی مشابه پرونده یا پوشه (Copy)	Ctrl+C
چسباندن (Paste)	Ctrl+V
برش (Cut)	Ctrl+X
چاپ (Print)	Ctrl+P
ذخیره (Save)	Ctrl+S
بازکردن پرونده (Open)	Ctrl+O
انتخاب همه (Select All)	Ctrl+A
واگرد (Undo)	Ctrl+Z

۲۰ صفحه ۱۶۹ کارگاه ۶

۲ ـ کنترل MenuStrip کجا قرار می‌گیرد؟

۳ ـ گزینه‌های دیگر منو را کجا می‌نویسید؟

۴ ـ با راست‌کلیک روی گزینه‌ها، چه گزینه‌های دیگری مشاهده می‌کنید؟

۵ ـ روی گزینه‌ی «ذخیره» راست‌کلیک کنید و از گزینه‌ی MenuItem، Insert را انتخاب کنید. چه تغییری در منو رخ می‌دهد؟ اگر بخواهید گزینه‌ای را حذف کنید، چگونه اقدام می‌کنید؟

۶ ـ مقدار ویژگی RightToLeft را Yes کنید. چه تغییری در منو رخ می‌دهد؟

کنترل MenuStrip در بالای فرم قرار می‌گیرد (به‌صورت نواری) و علاوه بر آن به‌عنوان یک component در سینی اجزای پایین پنجره‌ی طراحی نیز دیده می‌شود.

گزینه‌های بعدی منوی اصلی را در کادر Type Here کنار گزینه‌ی «پرونده» و گزینه‌های زیرمنو را در کادر Type Here زیر هر گزینه می‌نویسیم.

با راست‌کلیک روی گزینه‌ها گزینه‌هایی مانند Insert (با زیرگزینه‌های MenuItem، ComboBox، TextBox و Separator)، Delete، Set Image، Enabled، Checked، ShowShortcutKeys، Properties و ... دیده می‌شود.

با Insert ← MenuItem یک گزینه‌ی خالی جدید قبل از گزینه‌ی «ذخیره» وارد می‌شود تا متن آن را بنویسیم؛ برای حذف یک گزینه روی آن راست‌کلیک کرده و Delete را انتخاب می‌کنیم (یا آن را انتخاب و کلید Delete را می‌زنیم).

با RightToLeft = Yes چیدمان منو راست‌به‌چپ می‌شود: گزینه‌ی «پرونده» در سمت راست قرار می‌گیرد و متن گزینه‌ها راست‌چین می‌شود (مناسب منوهای فارسی).

۲۱ صفحه ۱۷۰ کارگاه ۷

۲ ـ ویژگی RightToLeft را برابر Yes قرار دهید. شکل منو چه تغییری می‌کند؟

۴ ـ ویژگی Enabled گزینه‌ی «چسباندن» را false کنید. شکل این گزینه چه تغییری می‌کند؟

۵ ـ ویژگی Visible گزینه‌ی «ذخیره» را false کنید. شکل این گزینه چه تغییری می‌کند؟ آیا پس از اجرا گزینه تغییر می‌کند؟

۶ ـ ویژگی Checked گزینه‌ی «قرمز» از «رنگ زمینه» را true کنید. شکل این گزینه چه تغییری می‌کند؟

۷ ـ از چه قالب‌های تصویری می‌توان برای تصویر گزینه‌ی منو استفاده کرد؟

۸ ـ چه ویژگی‌هایی باید تغییر کنند؟ (رنگ زمینه و متن گزینه‌های منو)

۱۰ ـ متن گزینه‌ی «پرونده» را به «&پرونده» تغییر دهید. شکل این گزینه چه تغییری می‌کند؟ برنامه را اجرا کنید. هم‌زمان کلید ترکیبی Alt و پ را بگیرید. چه مشاهده می‌کنید؟ کلید Alt را بگیرید. چه مشاهده می‌کنید؟

- RightToLeft = Yes: گزینه‌های منوی اصلی از راست شروع می‌شوند و زیرمنوها راست‌چین می‌شوند.
- Enabled = false: گزینه‌ی «چسباندن» کم‌رنگ (خاکستری) می‌شود و قابل انتخاب نیست.
- Visible = false: گزینه در زمان طراحی کم‌رنگ دیده می‌شود ولی پس از اجرا کاملاً حذف و ناپدید است و دیده نمی‌شود.
- Checked = true: علامت تیک (√) کنار گزینه‌ی «قرمز» ظاهر می‌شود.
- قالب‌های تصویری مناسب: png، jpg، bmp، gif، ico (قالب‌های پشتیبانی‌شده برای ویژگی Image).
- برای رنگ‌ها: BackColor و ForeColor گزینه‌های منو (و در صورت نیاز Font).
- ویژگی ShortcutKeys گزینه‌ی «خروج از برنامه» را Ctrl+E قرار می‌دهیم.
- با نوشتن «&پرونده» حرف «پ» زیرخط‌دار می‌شود (در حالت فعال شدن منو با Alt)؛ با گرفتن Alt + پ منوی «پرونده» باز می‌شود و با فقط گرفتن Alt حروف دسترسی زیرخط‌دار نمایان می‌شوند.

برای نام‌گذاری گزینه‌ها از پیشوند mnu استفاده می‌کنیم؛ مثلاً mnuNew برای «جدید» و mnuExit برای «خروج».

۲۲ صفحه ۱۷۰ فعالیت منزل

منوی نرم‌افزار Notepad ویندوز را در یک پروژه طراحی کنید.

کنترل MenuStrip را روی فرم گذاشته، ساختار زیر را با Insert → MenuItem و Separator می‌سازیم (نام‌گذاری با پیشوند mnu):

منوی Notepad

منوی اصلی	گزینه‌ها (کلید میان‌بر)
پرونده (File)	جدید (Ctrl+N)، باز کردن (Ctrl+O)، ذخیره (Ctrl+S)، ذخیره به‌عنوان، — جداکننده —، چاپ (Ctrl+P)، خروج
ویرایش (Edit)	واگرد (Ctrl+Z)، — جداکننده —، برش (Ctrl+X)، کپی (Ctrl+C)، چسباندن (Ctrl+V)، حذف (Del)، جست‌وجو (Ctrl+F)، انتخاب همه (Ctrl+A)، زمان/تاریخ (F5)
قالب (Format)	شکستن خودکار خط (Word Wrap)، قلم (Font...)
نما (View)	نوار وضعیت (Status Bar)
راهنما (Help)	مشاهده‌ی راهنما، درباره‌ی Notepad

۲۳ صفحه ۱۷۱ فعالیت کارگاهی

جدول ۴ شامل ویژگی‌های متداول گزینه‌ی منو است. با توجه به کارگاه ۷ این جدول را تکمیل کنید.

جدول ۴ـ ویژگی‌های کنترل منو

ایران توشه
توشه‌ای برای موفقیت

ویژگی	شرح
Name	نام منو یا گزینه‌های منو
Text	متن (عنوان) گزینه‌های منو
Checked	نمایش یا عدم نمایش علامت تیک کنار گزینه
Enabled	فعال یا غیرفعال بودن گزینه (در حالت غیرفعال کم‌رنگ می‌شود)
Visible	نمایش یا پنهان بودن گزینه
Image	تصویر کنار گزینه
RightToLeft	جهت نمایش (راست‌به‌چپ یا چپ‌به‌راست) متن و چیدمان گزینه‌ها
ShortcutKeys	کلید یا ترکیب کلید میان‌بر گزینه (مثل Ctrl+E)
ToolTipText	متن راهنما که با نگه داشتن ماوس روی گزینه نمایش داده می‌شود

۲۴ صفحه ۱۷۱ کنجکاوی

آیا می‌توان جهت نمایش متن گزینه‌های منو را تغییر داد؟

بله؛ با ویژگی RightToLeft منو (راست‌به‌چپ یا چپ‌به‌راست) می‌توان جهت چیدمان و نمایش گزینه‌ها را تغییر داد، و با ویژگی TextDirection گزینه‌ها (افقی، عمودی ...) جهت متن گزینه قابل تنظیم است.

۲۵ صفحه ۱۷۲ پژوهش

بررسی کنید که چه رویدادهای دیگری برای منو استفاده می‌شود و کاربرد آنها چیست؟

- DropDownOpening: هنگام باز شدن زیرمنو رخ می‌دهد؛ مناسب برای فعال/غیرفعال کردن گزینه‌ها قبل از نمایش (مثلاً فعال بودن «چسباندن»).
- DropDownClosed: هنگام بسته شدن زیرمنو رخ می‌دهد.
- MouseEnter و MouseHover: ورود ماوس به گزینه (مثلاً نمایش راهنما در نوار وضعیت).
- CheckedChanged: هنگام تغییر وضعیت علامت گزینه (تیک‌دار شدن).
- CheckStateChanged و Paint و TextChanged: به‌ترتیب هنگام تغییر حالت علامت، رسم گزینه و تغییر متن.

ایران توشه
توشه ای برای موفقیت

۲۶ صفحه ۱۷۲ کارگاه ۸

عملکرد گزینه‌ی «پاک» چیست؟

پس از اجرای برنامه، با کلیک روی گزینه‌ی «سفید» چه رخ می‌دهد؟ برای گزینه‌ی «زرد» از زیرمنوی «رنگ زمینه»، همانند مرحله‌ی قبل، کدنویسی کنید.

۶ ـ برای گزینه‌های «قرمز» و «آبی» از زیرمنوی «رنگ متن» نیز کد بنویسید.

۷ ـ متد رویداد کلیک گزینه‌ی «فونت» را بنویسید. کد تغییر قلم کادرمتن txtPaper را با استفاده از کادر محاوره‌ای fontDialog1 بنویسید.

۸ ـ برای گزینه‌ی «خروج از برنامه» کد مناسب بنویسید.

گزینه‌ی «پاک» با txtPaper.Clear() تمام متن کادر متن را پاک می‌کند. با کلیک روی «سفید» رنگ زمینه‌ی کادر متن سفید شده، کنار «سفید» علامت تیک ظاهر می‌شود و تیک گزینه‌ی «زرد» برداشته می‌شود.

ایران توشه
توشه ای برای موفقیت

```
private void mnuClear_Click(object sender, EventArgs e)
{
    txtPaper.Clear();
}
private void mnuWhite_Click(object sender, EventArgs e)
{
    txtPaper.BackColor = Color.White;
    mnuWhite.Checked = true;
    mnuYellow.Checked = false;
}
private void mnuYellow_Click(object sender, EventArgs e)
{
    txtPaper.BackColor = Color.Yellow;
    mnuYellow.Checked = true;
    mnuWhite.Checked = false;
}
private void mnuBlack_Click(object sender, EventArgs e)
{
    txtPaper.ForeColor = Color.Black;
    mnuBlack.Checked = true; mnuRed.Checked = false; mnuBlue.Checked = false;
}
private void mnuRed_Click(object sender, EventArgs e)
{
    txtPaper.ForeColor = Color.Red;
    mnuRed.Checked = true; mnuBlack.Checked = false; mnuBlue.Checked = false;
}
private void mnuBlue_Click(object sender, EventArgs e)
{
    txtPaper.ForeColor = Color.Blue;
    mnuBlue.Checked = true; mnuBlack.Checked = false; mnuRed.Checked = false;
}
private void mnuFont_Click(object sender, EventArgs e)
{
    if (fontDialog1.ShowDialog() == DialogResult.OK)
        txtPaper.Font = fontDialog1.Font;
}
private void mnuExit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

ایران توشه
توشه ای برای موفقیت

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

۲۷ صفحه ۱۷۳ فعالیت کارگاهی

بعد از افزودن کنترل MenuStrip به فرم، روی آن راست‌کلیک کنید و گزینه‌ی Insert Standard Items را انتخاب کنید. در فرم برنامه چه مشاهده می‌کنید؟ منوی ایجادشده را به‌دلخواه ویرایش و سفارشی کنید. آیا پس از اجرای برنامه و کلیک روی گزینه‌ها، عملیاتی انجام می‌شود؟

یک منوی استاندارد شامل منوهای File (New، Open، Save، Save As، Print، Print Preview، Exit)، Edit (Undo، Redo، Cut، Copy، Paste، Select All)، Tools (Customize، Options) و Help (Contents، Index، Search، About) به‌صورت خودکار در بالای فرم ساخته می‌شود. این گزینه‌ها فقط ظاهر منو را می‌سازند؛ پس از اجرا با کلیک روی آنها هیچ عملیاتی انجام نمی‌شود، زیرا برای هر گزینه باید متد رویداد Click نوشته شود.

۲۸ صفحه ۱۷۴ کارگاه ۹

برای تغییر Theme ویندوز چگونه عمل می‌کنید؟ برای مشاهده‌ی ویژگی‌های سیستم (System Properties) چگونه عمل می‌کنید؟

۳ ـ ... برنامه را اجرا کنید و روی کنترل کادر متن راست‌کلیک کنید. چه چیزی مشاهده می‌کنید؟ برای اینکه با راست‌کلیک روی فرم همین منو نمایش داده شود، چگونه عمل می‌کنید؟

۴ ـ برای رویداد کلیک گزینه‌ها، کد بنویسید.

چه تفاوتی بین کنترل MenuStrip و ContextMenuStrip وجود دارد؟

برای تغییر Theme: راست‌کلیک روی میزکار ← Personalize (شخصی‌سازی) ← Themes؛ برای ویژگی‌های سیستم: راست‌کلیک روی This PC (رایانه) ← Properties. هر دو با راست‌کلیک و استفاده از منوی میان‌بر انجام می‌شوند.

پس از قرار دادن ContextMenuStrip و تخصیص آن به ویژگی ContextMenuStrip کادر متن، با راست‌کلیک روی کادر متن همان منوی نوشته (رنگ زمینه، رنگ متن، قلم، پاک، خروج) باز می‌شود. برای نمایش همین منو با راست‌کلیک روی فرم نیز کافی است ویژگی ContextMenuStrip فرم را روی contextMenuStrip1 قرار دهیم.

برای رویداد کلیک گزینه‌ها، همان کدهای کارگاه ۸ نوشته می‌شود (می‌توان رویداد Click گزینه‌های منوی زمینه را همان متدهای قبلی (مثلاً mnuClear_Click) تعیین کرد):

```
private void ctxClear_Click(object sender, EventArgs e)
{
    txtPaper.Clear();
}
private void ctxRed_Click(object sender, EventArgs e)
{
    txtPaper.ForeColor = Color.Red;
}
private void ctxFont_Click(object sender, EventArgs e)
{
    if (fontDialog1.ShowDialog() == DialogResult.OK)
        txtPaper.Font = fontDialog1.Font;
}
```

تفاوت: MenuStrip منوی نواری ثابت در بالای فرم است و همیشه دیده می‌شود، ولی ContextMenuStrip منوی زمینه (میان‌بر) است که به یک کنترل یا فرم نسبت داده می‌شود و فقط با راست‌کلیک روی آن، در محل اشاره‌گر ظاهر می‌شود.

۲۹ صفحه ۱۷۵ کارگاه ۱۰

۳ ـ متد رویداد کلیک گزینه‌ی «شهدای دفاع مقدس» را بنویسید. قطعه کدی به برنامه اضافه کنید تا با کلیک روی هر تصویر اطلاعاتی مربوط به آن در ToolTip نمایش داده شود.

عملکرد علامت @ قبل از مسیر پرونده چیست؟

۴ ـ متد رویداد کلیک گزینه‌ی «سرود جمهوری اسلامی» را بنویسید.

ایران توشه
توشه ای برای موفقیت

۵ ـ برنامه‌ی نمایش تصاویر شهدای مدافع حرم را همانند شکل تغییر دهید. ۶ ـ با حرکت ماوس روی تصویر شهید، توضیحی از ویژگی‌های شهید در پایین تصویر نمایش داده شود. ۷ ـ قابلیتی به برنامه اضافه کنید تا با کلیک روی هر تصویر، تصویر در اندازه‌ی بزرگ‌تر نمایش داده شود.

```
private void mnuShohadaDefa_Click(object sender, EventArgs e)
{
    pic1.ImageLocation = @"images\hemmat.jpg";
    pic2.ImageLocation = @"images\aviny.jpg";
    pic3.ImageLocation = @"images\mahdavi.jpg";
    pic4.ImageLocation = @"images\hadi.jpg";
    pic5.ImageLocation = @"images\doran.jpg";
    pic6.ImageLocation = @"images\babaee.jpg";
    pic7.ImageLocation = @"images\chamran.jpg";
    pic8.ImageLocation = @"images\bronsi.jpg";
    pnlPictures.Visible = true;
    lblTitle.Text = "شهدای دفاع مقدس";
    lblDescription.Text = "آمار شهدای دفاع مقدس ۱۸۸ هزار و ۱۵ نفر است";
}
// ToolTip برای هر تصویر
private void pic1_Click(object sender, EventArgs e)
{
    ToolTip tip = new ToolTip();
    tip.Show("شهید همت؛ فرمانده لشکر ۲۷ محمد رسول‌الله", pic1, 3000);
}

private void mnuSorood1_Click(object sender, EventArgs e)
{
    System.Media.SoundPlayer player;
    player = new System.Media.SoundPlayer(@"sounds\Sorod.wav");
    player.Play();
    mnuSorood1.Checked = true;
    mnuSorood2.Checked = false;
    // ... mnuSorood3 تا mnuSorood6 نیز false
}
```

علامت @ قبل از رشته، آن را «رشته‌ی تحت‌اللفظی» (verbatim) می‌کند: نویسه‌ی \ در مسیر پرونده (images\hemmat.jpg) به‌عنوان نویسه‌ی فرار (escape) تفسیر نمی‌شود و نیازی به نوشتن \\ نیست. کلاس SoundPlayer فقط فایل صوتی با قالب wav را پخش می‌کند و متد Play آن صدا را پخش می‌کند.

مراحل ۵ تا ۷

```
// ۶: نمایش توضیح با حرکت ماوس روی تصویر
private void pic1_MouseEnter(object sender, EventArgs e)
{
    lblDescription.Text = "شهید حسین همدانی؛ از فرماندهان مدافع حرم";
}
// ۷: بزرگ‌نمایی با کلیک روی تصویر (یک متد برای همه‌ی تصویرها)
private void pic_Click(object sender, EventArgs e)
{
    PictureBox p = (PictureBox)sender;
    picBig.ImageLocation = p.ImageLocation;
    picBig.Visible = true;
}
```

این جواب ممکن است دقیق نباشد.

۳۰ صفحه ۱۸۰ **فعالیت کارگاهی**

در نرم‌افزار Word برای تایپ واژگان و قالب‌بندی آنها چگونه از صفحه‌کلید استفاده می‌کنید؟

اگر رایانه دستگاه ماوس نداشته باشد چگونه در سیستم‌عامل ویندوز یک پنجره را می‌بندید؟

چرا در بعضی از صفحه‌کلیدها، رنگ برخی از کلیدها متفاوت است؟

نرم‌افزار Notepad را اجرا کنید. با فشردن کلیدهای زیر چه چیزی نمایش داده می‌شود؟ (کلید جهت‌دار بالا، A، Ctrl، BackSpace)

با فشردن کلیدهای زیر چه نوع نویسه‌ای ایجاد می‌شود؟ (0، A، Enter، Space، Home، Alt، #)

- حروف و اعداد با کلیدهای نویسه‌ای تایپ می‌شوند و قالب‌بندی با کلیدهای میان‌بر انجام می‌شود (مثلاً Ctrl+B برای پررنگ، Ctrl+I برای مورب، Ctrl+U برای زیرخط) و کلید Shift برای حروف بزرگ.
- با ترکیب کلیدهای Alt+F4 پنجره‌ی فعال بسته می‌شود.
- کلیدهای رنگی (مثل کلیدهای کنترلی، تابعی، یا کلیدهای عددی) برای تفکیک گروه‌های کلیدها و پیدا کردن سریع‌تر آنها هستند.

ایران توشه
توشه ای برای موفقیت

نتیجه‌ی فشردن کلید در Notepad

کلید	نتیجه
کلید جهت‌دار بالا (UpArrow)	مکان‌نما یک سطر به بالا می‌رود (نویسه‌ای تولید نمی‌شود)
A	نویسه‌ی a (یا A با Shift یا Caps Lock) نوشته می‌شود
Ctrl	به‌تنهایی هیچ چیز نمایش داده نمی‌شود (کلید ترکیبی است)
BackSpace	نویسه‌ی قبل از مکان‌نما پاک می‌شود

نوع نویسه

کلید	نوع نویسه
Alt	غیر چاپ‌شدنی
Home	غیر چاپ‌شدنی
Space	چاپ‌شدنی (فاصله)
Enter	غیر چاپ‌شدنی (رفتن به سطر بعد)
A	چاپ‌شدنی
0	چاپ‌شدنی
#	چاپ‌شدنی

۳۱ صفحه ۱۸۱ کارگاه ۱

شرح رویدادهای صفحه‌کلید در جدول ۵ آمده است. جدول را تکمیل کنید.

۲ ـ ... برنامه را اجرا کنید. کلیدی را به دلخواه فشار دهید. چه اتفاقی می‌افتد؟

۳ ـ در متد رویداد KeyUp فرم کدی بنویسید که رنگ زمینه‌ی فرم را آبی کند. برنامه را اجرا کنید. کلیدی را به دلخواه فشار دهید. چه اتفاقی می‌افتد؟

جدول ۵ـ رویدادهای صفحه‌کلید

نام رویداد	شرح رویداد
KeyDown	زمانی رخ می‌دهد که کلیدی فشرده شود (اولین بار).
KeyPress	زمانی رخ می‌دهد که کنترل فوکوس داشته باشد و کاربر کلیدی را فشرده و رها کند (کلید نویسه‌ای).
KeyUp	زمانی رخ می‌دهد که کلیدی رها شود.

ایران توشه
توشه‌ای برای موفقیت

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    this.BackColor = Color.Red;
}
private void Form1_KeyUp(object sender, KeyEventArgs e)
{
    this.BackColor = Color.Blue;
}
```

با فشردن هر کلید رنگ فرم قرمز می‌شود و با رها کردن کلید رنگ آن آبی می‌شود.

۳۲ صفحه ۱۸۲ کارگاه ۲

۴ ـ برنامه را اجرا کرده، کلیدهای زیر را وارد کنید و نتیجه را در جدول زیر بنویسید (Alt، Home، Space، Enter، A، 0، #). نتیجه‌ی حاصل از ورود نویسه‌های چاپ‌شدنی و غیرچاپ‌شدنی چیست؟

با کمک هم‌گروهی خود، متن انگلیسی زیر را برای آشنایی بهتر با رویدادهای صفحه‌کلید، ترجمه کنید. The KeyPress event is not raised by non-character keys other than space and backspace; however, the non-character keys do raise the KeyDown and KeyUp events. Character keys include letters, numbers, punctuation, the Space bar, and the following non-printing characters: Tab, Enter, Return, Delete (or Backspace), Clear and Esc (Escape).

رویدادهای رخ‌داده

کلید	نتیجه (برچسب)
Alt	KeyDown-KeyUp- (KeyPress رخ نمی‌دهد)
Home	KeyDown-KeyUp-
Space	KeyDown-KeyPress-KeyUp-
Enter	KeyDown-KeyPress-KeyUp-
A	KeyDown-KeyPress-KeyUp-
0	KeyDown-KeyPress-KeyUp-
#	KeyDown-KeyPress-KeyUp- (با Shift؛ ابتدا KeyDown مربوط به Shift)

نتیجه: برای نویسه‌های چاپ‌شدنی (و Space، Enter، BackSpace، Esc، Tab) هر سه رویداد (KeyDown، KeyPress، KeyUp) به این ترتیب رخ می‌دهد، ولی برای نویسه‌های غیرچاپ‌شدنی (مانند Alt و Home) فقط KeyDown و KeyUp رخ می‌دهد.

ترجمه‌ی متن انگلیسی

رویداد KeyPress به‌وسیله‌ی کلیدهای غیرنویسه‌ای، جز کلیدهای فاصله (Space) و پاک‌کردن به عقب (Backspace)، ایجاد نمی‌شود؛ با این حال، کلیدهای غیرنویسه‌ای رویدادهای KeyDown و KeyUp را ایجاد می‌کنند. کلیدهای نویسه‌ای شامل حروف، اعداد، علائم نگارشی، کلید فاصله و نویسه‌های غیرچاپی زیر هستند: Tab، Enter، Return، Delete (یا Backspace)، Clear و Esc (Escape).

۳۳ صفحه ۱۸۴ کارگاه ۳

عملکرد متد رویداد Form1_KeyDown چیست؟

عملکرد timer1 در این برنامه چیست؟ برنامه را اجرا کنید. یک کلید دلخواه را نگه دارید. چه اتفاقی می‌افتد؟ کلید را رها کنید. چه اتفاقی می‌افتد؟

۸ ـ برنامه را طوری تغییر دهید که توپ پس از رها کردن کلید در جایگاه اولیه‌اش متوقف شود.

آیا نوع آرگومان e در رویداد KeyDown و KeyPress یکسان است و ویژگی‌های یکسانی دارد؟

متد Form1_KeyDown با فشردن هر کلید، متغیر isUp را true می‌کند (جهت حرکت توپ به بالا) و زمان‌سنج را فعال می‌کند. timer1 هر ۱۰ میلی‌ثانیه Tick را اجرا می‌کند و بسته به isUp توپ را ۵ پیکسل بالا یا پایین می‌برد. با نگه داشتن کلید توپ به بالا می‌رود (پرتاب) و با رها کردن کلید (KeyUp: isUp = false) توپ شروع به پایین آمدن (سقوط) می‌کند.

```
bool isUp;
int startTop;                                   // مکان اولیه‌ی توپ (در Form_Load:
startTop = picBall.Top)
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    isUp = true;
    timer1.Enabled = true;
}
private void Form1_KeyUp(object sender, KeyEventArgs e)
{
    isUp = false;
}
private void timer1_Tick(object sender, EventArgs e)
{
    if (isUp)
        picBall.Top -= 5;
    else
    {
        picBall.Top += 5;
        if (picBall.Top >= startTop)        // رسیدن به جایگاه اولیه
        {
            picBall.Top = startTop;
            timer1.Enabled = false;
        }
    }
}
```

خیر؛ آرگومان e در KeyDown و KeyUp از نوع KeyEventArgs (دارای KeyCode، KeyValue، KeyData، Alt، Shift، Control، SuppressKeyPress) و در KeyPress از نوع KeyPressEventArgs (دارای KeyChar و Handled) است.

۳۴ صفحه ۱۸۵ کارگاه ۴

کد اسکی (ASCII code) چیست؟

۴ ـ برنامه را اجرا کرده، کلیدهای زیر را فشار دهید و نتایج را در جدول بنویسید: B، b، + و –، + و – بخش عددی صفحه‌کلید، Shift.

آیا در این برنامه، نتیجه برای حروف کوچک و بزرگ یکسان است؟ آیا رویداد KeyDown و KeyUp برای تشخیص حروف الفبای کوچک و بزرگ مناسب است؟ کدام ویژگی آرگومان e کد اسکی نویسه را مشخص می‌کند؟

کد اسکی (ASCII: American Standard Code for Information Interchange) یک جدول استاندارد است که به هر نویسه‌ی انگلیسی، عدد و علامت یک کد عددی (۰ تا ۱۲۷) اختصاص می‌دهد؛ مثلاً A=۶۵، B=۶۶، a=۹۷، b=۹۸ و 0=۴۸.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    lblCharacter.Text = e.KeyCode.ToString();
    lblCode.Text = e.KeyValue.ToString();
}
```

جدول مرحله‌ی ۴ (KeyDown و KeyCode/KeyValue)

نام کلید	نویسه (KeyCode)	کد اسکی (KeyValue)
B	B	۶۶
b	B	۶۶
+ و - (ردیف بالای حروف)	Oemplus و OemMinus	۱۸۷ و ۱۸۹
+ و - بخش عددی صفحه‌کلید	Add و Subtract	۱۰۷ و ۱۰۹
Shift	ShiftKey	۱۶

بله؛ برای حروف کوچک و بزرگ نتیجه یکسان است (B و b هر دو KeyCode = B و KeyValue = 66)؛ زیرا KeyUp/KeyDown نویسه را نمی‌شناسند و فقط کلید فیزیکی را می‌شناسند. بنابراین این دو رویداد برای تشخیص حروف کوچک و بزرگ مناسب نیستند (مگر با بررسی e.Shift). ویژگی KeyValue کد کلید را برمی‌گرداند.

ایران توشه
توشه ای برای موفقیت

۳۵ صفحه ۱۸۶ کارگاه ۴

۵ ـ ویژگی KeyData را برای مرحله‌ی ۳ آزمایش کنید ... جدول مرحله‌ی ۴ و ۵ را با هم مقایسه کنید.

۶ ـ کد اسکی نویسه‌ها را با استفاده از رویداد KeyPress نمایش دهید.

۷ ـ جدول مرحله‌ی ۴ را با توجه به برنامه‌ی جدید، تکمیل کنید. آیا رویداد KeyPress برای تشخیص حروف الفبای کوچک و بزرگ مناسب است؟ آیا کد کلیدهای Shift و جهت‌دار نمایش داده می‌شود؟

مرحله‌ی ۵: نمایش KeyData

نام کلید	KeyData
B (بدون Shift)	B
B / b با Shift	B, Shift
+ (ردیف بالا، با Shift)	Oemplus, Shift
- (ردیف بالا)	OemMinus
+ و - بخش عددی	Add و Subtract
Shift	ShiftKey, Shift

تفاوت: KeyCode فقط نام کلید را می‌دهد ولی KeyData کلید به‌همراه کلیدهای ترکیبی (Shift، Control، Alt) را نشان می‌دهد؛ مثلاً Ctrl+A برای KeyData مقدار A, Control است.

```
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    lblCharacter.Text = e.KeyChar.ToString();
    int code = (int)e.KeyChar;
    lblCode.Text = code.ToString();
}
```

جدول مرحله‌ی ۷ (KeyPress)

نام کلید	نویسه	کد اسکی
B	B	۶۶
b	b	۹۸
+ (هر دو)	+	۴۳
- (هر دو)	-	۴۵
Shift	رویدادی رخ نمی‌دهد	—

بله، KeyPress برای تشخیص حروف کوچک و بزرگ مناسب است (کد اسکی ۶۶ و ۹۸ متفاوت است)؛ ولی کد کلیدهای Shift و جهت‌دار نمایش داده نمی‌شود، زیرا این کلیدها غیرنویسه‌ای هستند و KeyPress برای آنها رخ نمی‌دهد.

۳۶ صفحه ۱۸۷ کنجکاوی

برای شناسایی کلید فشرده‌شده در رویداد KeyPress از چه ویژگی استفاده کردیم؟ نوع داده‌ی آن چیست؟

ایران توشه
توشه ای برای موفقیت

مشابه جدول ۶ ویژگی‌های آرگومان e در رویداد KeyPress را بنویسید.

از ویژگی e.KeyChar از نوع char استفاده کردیم (و برای کد اسکی آن را به int تبدیل کردیم).

ویژگی‌های آرگومان e در رویداد KeyPress (کلاس KeyPressEventArgs)

نام ویژگی	نوع داده	شرح
KeyChar	char	نویسه‌ی مربوط به کلید فشرده‌شده را برمی‌گرداند یا تنظیم می‌کند.
Handled	bool	اگر true شود، رویداد پردازش‌شده تلقی می‌شود و نویسه وارد کنترل نمی‌شود (جلوگیری از ورود نویسه).

۳۷ صفحه ۱۸۷ **فعالیت منزل**

برنامه‌ای بنویسید که با فشردن کلیدهای فارسی، کد حروف فارسی نمایش داده شود.

با زبان فارسی صفحه‌کلید، رویداد KeyPress نویسه‌ی یونیکد حرف را در e.KeyChar می‌دهد و تبدیل آن به int کد یونیکد (مثلاً «ا» ← ۱۵۷۵ و «ب» ← ۱۵۷۶) است:

```
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    lblCharacter.Text = e.KeyChar.ToString();
    lblCode.Text = ((int)e.KeyChar).ToString();   // کد یونیکد حرف فارسی
}
```

۳۸ صفحه ۱۸۹ **کارگاه ۵**

ـ متغیر ratio چه مقداری می‌گیرد و علت استفاده از آن چیست؟

ـ اعداد ۱۸۷ و ۱۸۹ کد اسکی چه کلیدهایی هستند؟

ـ عملکرد متد رویداد KeyDown فرم چیست؟

۴ ـ برنامه را طوری تغییر دهید که با کلیدهای + و - در قسمت عددی صفحه‌کلید هم، تصویر بزرگ و کوچک شود.

۵ ـ برنامه را طوری تغییر دهید که با تغییر اندازه‌ی تصویر، کادرتصویر در وسط فرم قرار گیرد.

ایران توشه
توشه‌ای برای موفقیت

۶ ـ برنامه را طوری تغییر دهید که با تغییر اندازه‌ی تصویر، ارتفاع تصویر حداکثر ۳۰۰ و حداقل ۵۰ پیکسل شود.

متغیر ratio نسبت عرض به ارتفاع کادر تصویر (Width / Height) را می‌گیرد؛ با ضرب آن در ارتفاع جدید، عرض متناسب محاسبه می‌شود تا تصویر هنگام بزرگ و کوچک شدن تغییر شکل ندهد. اعداد ۱۸۷ و ۱۸۹ کد کلیدهای + (Oemplus) و – (OemMinus) ردیف بالای حروف هستند. متد رویداد با فشردن Ctrl و + ارتفاع را ۵ پیکسل زیاد، با Ctrl و – ۵ پیکسل کم می‌کند و عرض را متناسب با آن تنظیم می‌کند.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    double ratio = (double)picSample.Width / picSample.Height;
    if (e.Control)
    {
        // + (109 ,107) و بخش عددی (189 ,187) و - ردیف بالا
        if (e.KeyValue == 187 || e.KeyValue == 107)
        {
            if (picSample.Height < 300) picSample.Height += 5;
        }
        else if (e.KeyValue == 189 || e.KeyValue == 109)
        {
            if (picSample.Height > 50) picSample.Height -= 5;
        }
        picSample.Width = (int)(ratio * picSample.Height);
        // قرار دادن کادرتصویر در وسط فرم
        picSample.Left = (this.ClientSize.Width - picSample.Width) / 2;
        picSample.Top = (this.ClientSize.Height - picSample.Height) / 2;
    }
}
```

۳۹ صفحه ۱۹۰ **فعالیت منزل**

برنامه‌ی کارگاه ۵ را با استفاده از ویژگی KeyData بنویسید.

ایران توشه
توشه ای برای موفقیت

KeyData مقدار شمارشی Keys است که کلید و کلیدهای ترکیبی را با هم دارد؛ ترکیب Ctrl و + برابر Keys.Control | Keys.Oemplus است:

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    double ratio = (double)picSample.Width / picSample.Height;
    if (e.KeyData == (Keys.Control | Keys.Oemplus))
        picSample.Height += 5;
    else if (e.KeyData == (Keys.Control | Keys.OemMinus))
        picSample.Height -= 5;
    picSample.Width = (int)(ratio * picSample.Height);
}
```

۴۰ صفحه ۱۹۰ کارگاه ۶

۴ ـ برنامه را دوباره اجرا کنید. اگر کلیدی فشار دهید، آیا کد اسکی و نویسه نمایش داده می‌شوند؟ آیا رویداد KeyDown فرم رخ می‌دهد؟

ـ اگر کلید حرفی را تایپ کنید، آیا متنی در کادرمتن نوشته می‌شود؟ ـ کلید Tab را فشار دهید. اگر کلید حرفی تایپ کنید متنی در کادرمتن نوشته می‌شود؟ ـ اگر کلیدی فشار دهید، آیا کد اسکی و نویسه نمایش داده می‌شوند؟ آیا رویداد KeyDown فرم رخ می‌دهد؟

چند کنترل در جعبه‌ابزار بیابید که به رویدادهای صفحه‌کلید واکنش نشان می‌دهند.

ـ آیا در کارگاه ۴ باید مقدار ویژگی KeyPreview، true شود؟ چرا؟

ـ برای تغییر فوکوس در برنامه از چه کلیدی استفاده می‌کنید؟ در هنگام برنامه‌نویسی از چه کدی برای تغییر فوکوس استفاده می‌کنید؟

اگر فوکوس روی کادر متن باشد (KeyPreview = false)، رویدادهای صفحه‌کلید فرم رخ نمی‌دهند: کد اسکی و نویسه نمایش داده نمی‌شوند؛ و حرف تایپ‌شده در کادر متن نوشته می‌شود. پس از فشردن Tab فوکوس به دکمه می‌رود؛ تایپ حرف، متنی در کادر متن نمی‌نویسد. با KeyPreview = true فرم قبل از کنترل به رویدادهای صفحه‌کلید واکنش نشان می‌دهد و در هر دو حالت (فوکوس روی کادر متن یا دکمه) کد اسکی و نویسه نمایش داده می‌شوند و رویداد KeyDown فرم رخ می‌دهد.

کنترل‌هایی مانند TextBox، Button، ListBox، ComboBox، CheckBox، RadioButton و DataGridView رویدادهای صفحه‌کلید دارند.

در کارگاه ۴ فقط فرم دارای کنترلی نبود که فوکوس را بگیرد؛ پس نیازی نیست، ولی اگر در فرم کنترل‌هایی مانند کادر متن یا دکمه وجود داشته باشد باید KeyPreview = true شود. با کلید Tab فوکوس تغییر می‌کند و در برنامه‌نویسی با متد Focus() (مثلاً txtLastName.Focus();).

ایران توشه
توشه ای برای موفقیت

۴۱ صفحه ۱۹۱ فعالیت منزل

پس از مشاهده‌ی فیلم، «برنامه‌ی یافتن نویسه در صفحه‌کلید» را بنویسید. (فیلم شماره‌ی ۱۱۰۸)

برنامه نویسه‌ی تایپ‌شده را دریافت کرده و نمایش می‌دهد که روی کدام کلید است (و با رنگ کردن برچسب کلید مربوط، آن را «پیدا می‌کند»). فیلم در دسترس نبود؛ طراحی ساده: برای هر کلید یک برچسب هم‌نام (مثلاً lblA) وجود دارد.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    foreach (Control c in this.Controls)
        if (c is Label) c.BackColor = Color.White;
    Control[] found = this.Controls.Find("lbl" + e.KeyCode.ToString(), true);
    if (found.Length > 0)
        found[0].BackColor = Color.Yellow;     // کلید فشرده‌شده
}
```

این جواب ممکن است دقیق نباشد.

۴۲ صفحه ۱۹۲ کارگاه ۷

۴ ـ کد را تغییر دهید تا سفینه علاوه بر کلیدهای جهت‌دار با فشار کلیدهای A ـ W ـ S ـ D هم حرکت کند.

برنامه را اجرا کنید.

آیا ویژگی KeyPreview فرم، باید true شود؟ چرا؟

ـ در کارگاه ۷ برنامه را تغییر دهید تا با فشردن کلیدهای جهت‌دار، سفینه به‌صورت خودکار در همان جهت حرکت کند و با برخورد به دیواره‌ی فرم در جهت خلاف برگردد. از زمان‌سنج استفاده کنید.

ـ برنامه‌ای بنویسید که در آن پنجره‌ی برنامه، با فشردن کلید ESC بسته شود.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    switch (e.KeyCode)
    {
        case Keys.Left:
        case Keys.A:
            picSpacecraft.Left -= 5; break;
        case Keys.Right:
        case Keys.D:
            picSpacecraft.Left += 5; break;
        case Keys.Up:
        case Keys.W:
            picSpacecraft.Top -= 5; break;
        case Keys.Down:
        case Keys.S:
            picSpacecraft.Top += 5; break;
        case Keys.Escape:
            this.Close(); break;       // بستن پنجره با ESC
    }
}
```

اگر در فرم کنترلی (مثل دکمه) فوکوس را بگیرد، رویداد KeyDown فرم رخ نمی‌دهد؛ بنابراین در صورت وجود چنین کنترل‌هایی باید KeyPreview = true شود. در این برنامه که فقط کادر تصویر دارد (فرم فوکوس را دارد) لازم نیست، ولی گذاشتن آن بی‌ضرر است و توصیه می‌شود.

حرکت خودکار با زمان‌سنج و برگشت با برخورد به دیواره

ایران توشه
توشه ای برای موفقیت

```
int dx = 0, dy = 0;                        // جهت حرکت (timer1.Interval = 20, Enabled =
true)
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Left)  { dx = -5; dy = 0; }
    if (e.KeyCode == Keys.Right) { dx = 5;  dy = 0; }
    if (e.KeyCode == Keys.Up)    { dx = 0;  dy = -5; }
    if (e.KeyCode == Keys.Down)  { dx = 0;  dy = 5; }
}
private void timer1_Tick(object sender, EventArgs e)
{
    picSpacecraft.Left += dx;
    picSpacecraft.Top += dy;
    if (picSpacecraft.Left <= 0 || picSpacecraft.Right >= this.ClientSize.Width) dx = -
dx;
    if (picSpacecraft.Top <= 0 || picSpacecraft.Bottom >= this.ClientSize.Height) dy = -
dy;
}
```

۴۳ صفحه ۱۹۳ کارگاه ۸

فوکوس روی کدام کنترل قرار دارد؟

عملکرد متد رویداد KeyDown_txtFirstName چیست؟

۴ ـ متد رویداد KeyDown را برای کنترل‌های کادرمتن دیگر هم بنویسید.

آخرین کنترلی که فوکوس دریافت می‌کند، کدام است؟

هنگام اجرا فوکوس روی اولین کنترل به ترتیب TabIndex یعنی txtFirstName است. متد txtFirstName_KeyDown هنگام فشردن Enter در کادر نام، فوکوس را با متد ()Focus به کادر نام خانوادگی منتقل می‌کند.

ایران توشه
توشه ای برای موفقیت

```
private void txtFirstName_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        txtLastName.Focus();
}
private void txtLastName_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        txtFatherName.Focus();
}
private void txtFatherName_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        txtMelliCode.Focus();
}
private void txtMelliCode_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        btnOk.Focus();            // دکمه‌ی اول
}
```

آخرین کنترل، دکمه‌ای است که کادر کد ملی فوکوس را به آن می‌دهد (در اینجا دکمه‌ی اول، مثلاً btnOk).

۴۴ صفحه ۱۹۴ پژوهش

اگر بخواهیم با فوکوس روی هر کادرمتن رنگ زمینه‌ی آن تغییر کند و با از دست دادن فوکوس رنگ آن به حالت اول درآید، از چه رویدادهایی استفاده می‌کنیم؟

از رویدادهای Enter (یا GotFocus) برای دریافت فوکوس و Leave (یا LostFocus) برای از دست دادن فوکوس:

```
private void txtFirstName_Enter(object sender, EventArgs e)
{
    txtFirstName.BackColor = Color.LightYellow;
}
private void txtFirstName_Leave(object sender, EventArgs e)
{
    txtFirstName.BackColor = Color.White;
}
```

ایران توشه
توشه ای برای موفقیت

۴۵ صفحه ۱۹۵ کارگاه ۹

۳ ـ ... چه کلیدهایی در کادرمتن شماره‌ی دانش‌آموزی تایپ می‌شوند؟ عملکرد ویژگی SuppressKeyPress چیست؟

۴ ـ لزوم تبدیل e.KeyCode به char چیست؟ چه کلیدهایی در کادرمتن شماره‌ی دانش‌آموزی تایپ می‌شوند؟ آیا می‌توان داده‌ی درون کادرمتن را با کلید Delete یا Backspace حذف کرد؟ آیا می‌توان با کلیدهای جهت‌دار، داده را پیمایش کرد؟

۵ ـ کد مرحله‌ی ۴ را به‌گونه‌ای ویرایش کنید که بتوان عدد ورودی را حذف و پیمایش کرد.

با e.SuppressKeyPress = true هیچ کلیدی در کادر متن تایپ نمی‌شود؛ این ویژگی با true شدن از ورود داده از طریق صفحه‌کلید جلوگیری می‌کند.

e.KeyCode از نوع Keys است ولی متد char.IsDigit نویسه‌ی char می‌گیرد؛ پس باید (char)e.KeyCode نوشته شود. در این حالت فقط رقم‌های ردیف بالای صفحه‌کلید (کد ۴۸ تا ۵۷) تایپ می‌شوند؛ کلیدهای Backspace، Delete و جهت‌دار نیز مسدود می‌شوند (غیرقابل حذف و پیمایش) و ارقام بخش عددی هم پذیرفته نمی‌شوند.

```
private void txtStudentNO_KeyDown(object sender, KeyEventArgs e)
{
    // فقط رقم، به‌همراه Backspace، Delete و کلیدهای جهت‌دار
    if (!char.IsDigit((char)e.KeyCode)
        && e.KeyCode != Keys.Back && e.KeyCode != Keys.Delete
        && e.KeyCode != Keys.Left && e.KeyCode != Keys.Right
        && !(e.KeyCode >= Keys.NumPad0 && e.KeyCode <= Keys.NumPad9))
        e.SuppressKeyPress = true;
}
```

۴۶ صفحه ۱۹۶ کارگاه ۹

۶ ـ ... چه نویسه‌هایی در کادرمتن تایپ می‌شود؟

۷ ـ ... برنامه را اجرا کرده، نام AliReza را در کادرمتن وارد کنید. به حروف بزرگ و کوچک توجه کنید. عملکرد ویژگی Handled چیست؟

۸ ـ برنامه را طوری تغییر دهید که بتوان حروف تایپ‌شده را پاک و پیمایش کرد.

۹ ـ ... برنامه را اجرا کرده، نام خانوادگی Ahmadi را در کادرمتن وارد کنید. چه متنی در کادرمتن نوشته می‌شود؟ عملکرد متد رویداد KeyPress_txtLastName چیست؟

در متد KeyDown نام (txtFirstName) فقط حروف الفبا (char.IsLetter) و کلید Backspace پذیرفته می‌شود؛ ارقام و علائم تایپ نمی‌شوند.

در متد KeyPress فقط حروف کوچک (char.IsLower) پذیرفته می‌شود؛ ویژگی Handled اگر true شود نویسه پردازش‌شده تلقی شده و وارد کادر متن نمی‌شود. با ورود AliReza نویسه‌های بزرگ A و R نادیده گرفته می‌شوند و فقط lieza نوشته می‌شود.

```
private void txtFirstName_KeyPress(object sender, KeyPressEventArgs e)
{
    // حروف کوچک، Backspace (کد 8) و بقیه‌ی کلیدهای کنترلی (جهت‌دار) KeyPress ندارند)
    if (!(char.IsLower(e.KeyChar) || e.KeyChar == (char)Keys.Back))
        e.Handled = true;
}
private void txtLastName_KeyPress(object sender, KeyPressEventArgs e)
{
    if ((e.KeyChar >= 'a') && (e.KeyChar <= 'z'))
    {
        int Code = (int)e.KeyChar - 32;
        e.KeyChar = (char)Code;        // تبدیل حرف کوچک به بزرگ
    }
}
```

با ورود Ahmadi در کادر نام خانوادگی، متن AHMADI نوشته می‌شود؛ متد KeyPress_txtLastName هر حرف کوچک انگلیسی را با کم کردن ۳۲ از کد اسکی آن به حرف بزرگ تبدیل می‌کند (اختلاف کد حروف کوچک و بزرگ ۳۲ است).

۴۷ صفحه ۱۹۷ فعالیت کارگاهی

نتیجه‌ی خروجی مربوط به مثال‌های جدول ۷ را بنویسید.

جدول ۷ـ متدهای نوع داده‌ی char

نام متد	نوع خروجی	شرح	مثال ۱	مثال ۲
IsDigit	bool	آیا نویسه‌ی ورودی رقم است.	char.IsDigit('A') ← false	char.IsDigit('5') ← true
IsLetter	bool	آیا نویسه‌ی ورودی حرف است.	char.IsLetter('A') ← true	char.IsLetter('@') ← false
IsUpper	bool	آیا نویسه‌ی ورودی حرف بزرگ است.	char.IsUpper('d') ← false	char.IsUpper('M') ← true
IsLower	bool	آیا نویسه‌ی ورودی حرف کوچک است.	char.IsLower('d') ← true	char.IsLower('M') ← false

۴۸ صفحه ۱۹۸ کنجکاوی

متغیر از نوع رشته‌ای شامل چه مقادیری می‌شود؟

ثابت رشته‌ای چیست؟

ایران توشه
توشه ای برای موفقیت

نوع داده‌ی name[0] چیست؟ آیا می‌توان این نویسه را مقداردهی کرد؟

چه عملیاتی در واژه‌پردازها روی رشته‌ها انجام می‌شود؟ نرم‌افزارهایی را نام ببرید که داده‌های رشته‌ای در آن استفاده می‌شود.

به وبگاه سامانه‌ی دانش‌آموزی مراجعه کرده‌اید؟ چه داده‌هایی در سامانه، در مورد شما ثبت‌شده است؟

هنگامی که می‌خواهید یک رایانامه (Email) ایجاد کنید چه داده‌هایی باید در سرویس‌دهنده وارد کنید؟ کدام یک از این داده‌ها، رشته‌ای است؟

- متغیر رشته‌ای می‌تواند دنباله‌ای از نویسه‌ها (حروف، ارقام، علائم و فاصله) را نگه دارد؛ مثلاً "AliReza" یا "123" یا رشته‌ی خالی "".
- ثابت رشته‌ای مقدار رشته‌ای است که مستقیم در برنامه نوشته می‌شود و بین دو علامت نقل‌قول قرار دارد؛ مانند "Hello".
- نوع داده‌ی name[0] برابر char است (یک نویسه). نمی‌توان آن را مقداردهی کرد: رشته در سی‌شارپ تغییرناپذیر (immutable) و name[0] فقط‌خواندنی است؛ دستور name[0] = 'B'; خطای کامپایل می‌دهد.
- عملیات در واژه‌پردازها روی رشته‌ها: جست‌وجو و جایگزینی، حذف، درج، تبدیل حروف به بزرگ/کوچک، شمارش نویسه‌ها و واژه‌ها، مرتب‌سازی، مقایسه. نرم‌افزارها: Notepad، Word، مرورگرها، موتورهای جست‌وجو، پیام‌رسان‌ها، پایگاه‌های داده.
- داده‌های سامانه‌ی دانش‌آموزی: نام و نام خانوادگی، نام پدر، کد ملی، شماره‌ی دانش‌آموزی، نشانی، نمرات و پایه؛ داده‌های رایانامه: نام، نام خانوادگی، نام کاربری، گذرواژه، شماره‌ی تلفن؛ بیشتر آنها (نام، نام کاربری، گذرواژه، نشانی و حتی کد ملی) رشته‌ای هستند.

ایران توشه
توشه‌ای برای فردا

۴۹ صفحه ۱۹۹ کارگاه ۱۰

۳ ـ ... پس از کلیک روی دکمه چه مقداری در کادرمتن قرار می‌گیرد؟ عملکرد ویژگی Length چیست؟

۴ ـ ... چه مقداری در کادرمتن قرار می‌گیرد؟ عملکرد متد ToUpper چیست؟ عملکرد متد ToLower را بررسی کنید.

۵ ـ ... عملکرد متد IndexOf چیست؟ اگر به‌جای ۰ عدد ۵ قرار دهید، خروجی چه می‌شود؟ اگر در مرحله‌ی ۵ به‌جای Reza، واژه‌ی Ahmad را بنویسید، خروجی چه می‌شود؟

عملکرد متد LastIndexOf چیست؟

```
string name = "AliReza";
txtResult.Text = name.Length.ToString();    // 7
txtResult.Text = name.ToUpper();            // ALIREZA
txtResult.Text = name.ToLower();            // alireza

string name2 = "AliReza Rezaee";
txtResult.Text = name2.IndexOf("Reza", 0).ToString();   // 3
txtResult.Text = name2.IndexOf("Reza", 5).ToString();   // 8
txtResult.Text = name2.IndexOf("Ahmad", 0).ToString();  // -1
txtResult.Text = name2.LastIndexOf("Reza").ToString();  // 8
```

- Length: تعداد نویسه‌های رشته را برمی‌گرداند؛ برای AliReza مقدار 7.
- ToUpper(): همه‌ی حروف را بزرگ می‌کند (ALIREZA) و ToLower() همه‌ی حروف را کوچک می‌کند (alireza).
- IndexOf("Reza", 0): اندیس اولین رخداد واژه‌ی Reza را از اندیس ۰ به بعد برمی‌گرداند: 3. با شروع از ۵، رخداد اول نادیده گرفته می‌شود و رخداد دوم (Rezaee) در اندیس 8 پیدا می‌شود. اگر واژه پیدا نشود (مثل Ahmad) نتیجه 1- است.
- LastIndexOf: اندیس آخرین رخداد واژه را (جست‌وجو از انتها) برمی‌گرداند: برای Reza در AliReza Rezaee برابر 8.

۵۰ صفحه ۲۰۰ کارگاه ۱۱

اگر کاربر در کادرمتن نام کاربری ADMIN را وارد کند، چه پیامی دریافت می‌کند؟

۴ ـ برنامه را بهگونه‌ای ویرایش کنید که ورود حروف کوچک یا بزرگ در نام کاربری یا گذرواژه، تأثیری در اعتبارسنجی کاربر نداشته باشد.

برنامه را اجرا کنید. نام کاربری را با حروف کوچک و بزرگ وارد کنید. (CompareTo)

۵ ـ فاصله‌ی ابتدا و انتهای نام کاربری را حذف کنید.

با ورود ADMIN پیام «نام کاربری یا گذرواژه اشتباه است» نمایش داده می‌شود؛ زیرا مقایسه‌ی == بین رشته‌ها به حروف بزرگ و کوچک حساس است.

ایران توشه
توشه‌ای برای موفقیت

```
private void btnLogin_Click(object sender, EventArgs e)
{
    // بدون توجه به حروف بزرگ و کوچک، و حذف فاصله‌ی ابتدا و انتها
    string user = txtUserName.Text.Trim().ToLower();
    string pass = txtPassword.Text.Trim().ToLower();
    if (user == "admin" && pass == "123")
        MessageBox.Show("ورود شما با موفقیت انجام شد");
    else
    {
        string s = "نام کاربری یا گذرواژه اشتباه است";
        MessageBox.Show(s);
    }
}

// با CompareTo (مقدار 0 یعنی دو رشته برابرند):
if (txtUserName.Text.Trim().ToLower().CompareTo("admin") == 0 &&
    txtPassword.Text.Trim().ToLower().CompareTo("123") == 0)
    MessageBox.Show("ورود شما با موفقیت انجام شد");
```

متد CompareTo اگر رشته‌ها برابر باشند ۰، اگر رشته‌ی اول در ترتیب الفبایی بعد از دومی باشد مقدار مثبت (۱) و در غیر این صورت منفی (۱-) برمی‌گرداند؛ به حروف بزرگ و کوچک حساس است (پس با ToLower هم‌ترکیب شده است). متد Trim() فاصله‌های ابتدا و انتهای رشته را حذف می‌کند تا اعتبارسنجی با فاصله‌ی اضافی خراب نشود.

۵۱ صفحه ۲۰۱ **فعالیت کارگاهی**

ستون‌های «عملکرد» و «نوع داده‌ی خروجی» را تکمیل کنید. اگر "sample = "AliReza باشد، حاصل دستورات در ستون «مثال» را بنویسید.

جدول ۸ـ شرح متدهای رشته‌ای ("sample = "AliReza

ایران توشه
توشه ای برای موفقیت

نام متد	عملکرد	نوع خروجی	مثال	حاصل
ToUpper	تبدیل همه‌ی حروف به بزرگ	string	sample.ToUpper()	ALIREZA
ToLower	تبدیل همه‌ی حروف به کوچک	string	sample.ToLower()	alireza
IndexOf	اندیس اولین رخداد یک رشته از مکان مشخص (۱- اگر نباشد)	int	sample.IndexOf("Reza",0)	3
Replace	جایگزینی همه‌ی رخدادهای یک رشته با رشته‌ی دیگر	string	sample.Replace("Ali","Mohammad")	MohammadReza
Remove	حذف تعداد معینی نویسه از یک مکان	string	sample.Remove(2,3)	Alza
Insert	درج یک رشته در مکان مشخص	string	sample.Insert(4,"Mohammad")	AliRMohammadeza
Format	قالب‌بندی و ساخت رشته با جانگهدارها	string	string.Format("{0} is a student", sample)	AliReza is a student
CompareTo		int	sample.CompareTo("Ali")	

ایران توشه
توشه ای برای موفقیت

	مقایسه‌ی دو رشته؛ ۰ برابر، مثبت یا منفی بر اساس ترتیب			۱ (عدد مثبت؛ AliReza بزرگ‌تر از Ali)
Trim	حذف فاصله‌های ابتدا و انتهای رشته	string	" Ali ".Trim()	Ali

۵۲ صفحه ۲۰۱ فیلم

پس از مشاهده‌ی فیلم «برنامه‌ی جست‌وجو و جایگزینی واژه» را بنویسید. (فیلم شماره‌ی ۱۱۰۹) کارگاه ۱۲: به‌کارگیری متدهای IndexOf و Replace

برنامه: یک کادر متن چندخطی (txtText)، کادر «واژه‌ی مورد جست‌وجو» (txtFind)، کادر «واژه‌ی جایگزین» (txtReplace) و دکمه‌های btnFind و btnReplace (فیلم در دسترس نبود؛ طراحی ساده):

```
private void btnFind_Click(object sender, EventArgs e)
{
    int index = txtText.Text.IndexOf(txtFind.Text);
    if (index >= 0)
    {
        txtText.Focus();
        txtText.Select(index, txtFind.Text.Length);   // انتخاب واژه‌ی پیداشده
    }
    else
        MessageBox.Show("واژه پیدا نشد");
}
private void btnReplace_Click(object sender, EventArgs e)
{
    if (txtText.Text.IndexOf(txtFind.Text) >= 0)
        txtText.Text = txtText.Text.Replace(txtFind.Text, txtReplace.Text);
    else
        MessageBox.Show("واژه پیدا نشد");
}
```

این جواب ممکن است دقیق نباشد.

این فایل توسط سایت ایران‌توشه تهیه شده است.

ایران توشه
توشه ای برای موفقیت

ایران توشه
توشه ای برای موفقیت

۵۳ صفحه ۲۰۲ پروژه فاز ۴

به پروژه‌ی کتابخانه منو اضافه کنید: با انتخاب منوی «خروج» کادر پیام تأیید خروج نمایش داده شده، در صورت تأیید کاربر برنامه تمام شود؛ با کلیک روی منوی «راهنما» فرم «درباره‌ی ما» ظاهر شود؛ گزینه‌ی «ورود» منوی «مدیر» برای دریافت نام کاربری و گذرواژه‌ی مدیر است؛ منوی «ویرایش» برای انتخاب رنگ و قلم کادرهای متن در تمام سربرگ‌هاست. موارد زیر را برای تکمیل پروژه اضافه کنید: دسترسی به سربرگ‌های فعال با صفحه‌کلید؛ ComboBox برای محدود کردن سال چاپ کتاب؛ فقط نویسه‌های حرفی در نام مؤلف؛ حذف فضای خالی در کادر جست‌وجو؛ غیرفعال بودن دکمه‌ی Search تا زمانی که عبارتی وارد نشده است.

```
// منوی خروج
private void mnuExit_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("خروج" ,"آیا مایل به خروج هستید؟", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
        Application.Exit();
}
// منوی راهنما ← درباره‌ی ما
private void mnuAbout_Click(object sender, EventArgs e)
{
    AboutForm frm = new AboutForm();
    frm.ShowDialog();
}
// منوی مدیر ← ورود
private void mnuAdminLogin_Click(object sender, EventArgs e)
{
    LoginForm frm = new LoginForm();
    frm.ShowDialog();
    if (frm.succeeded)
        tabMember.Enabled = tabBook.Enabled = true;
}
// منوی ویرایش ← رنگ و قلم کادرهای متن (همه‌ی سربرگ‌ها)
private void mnuFont_Click(object sender, EventArgs e)
{
    if (fontDialog1.ShowDialog() == DialogResult.OK)
        foreach (TabPage tp in tabControl1.TabPages)
            foreach (Control c in tp.Controls)
                if (c is TextBox) c.Font = fontDialog1.Font;
}
private void mnuColor_Click(object sender, EventArgs e)
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
        foreach (TabPage tp in tabControl1.TabPages)
            foreach (Control c in tp.Controls)
                if (c is TextBox) c.BackColor = colorDialog1.Color;
}
```

ایران توشه
توشه‌ای برای موفقیت

```
// تکمیل پروژه
// ۱) دسترسی به سربرگ‌ها با صفحه‌کلید: Ctrl+1 / Ctrl+2 / Ctrl+3
private void MainForm_KeyDown(object sender, KeyEventArgs e)   // KeyPreview = true
{
    if (e.Control && e.KeyCode == Keys.D1 && tabMember.Enabled) tabControl1.SelectedTab
= tabMember;
    if (e.Control && e.KeyCode == Keys.D2 && tabBook.Enabled)   tabControl1.SelectedTab
= tabBook;
}
// ۲) سال چاپ با ComboBox (DropDownStyle = DropDownList؛ سال‌ها در Items)
// ۳) نام مؤلف فقط حروف
private void txtAuthor_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(char.IsLetter(e.KeyChar) || e.KeyChar == ' ' || e.KeyChar == (char)Keys.Back))
        e.Handled = true;
}
// ۴ و ۵) جست‌وجو
private void txtSearch_TextChanged(object sender, EventArgs e)
{
    btnSearch.Enabled = txtSearch.Text.Trim().Length > 0;   // فعال فقط وقتی عبارتی وارد
شده
}
private void btnSearch_Click(object sender, EventArgs e)
{
    string key = txtSearch.Text.Trim();                      // حذف فاصله‌ی ابتدا و
انتها
    // جست‌وجو در آرایه‌ی کتاب‌ها با key ...
}
```

طرح دقیق فرم کتابخانه در دسترس نبود؛ نام کنترل‌ها پیشنهادی هستند.

ایران توشه
توشه ای برای موفقیت