

به نام خدا

با سلام خدمت هنرجویان عزیز

Pdf همراه مربوط به پودمان ۴ کتاب توسعه برنامه سازی می باشد که
مباحث موس و کیبرد توضیح داده شده است.

پس از مطالعه برنامه های نوشته شده را در کامپیوترتان وارد کرده و با
نام خودتان به ترتیب کارگاه ذخیره کنید و بعد از بازگشایی مدارس
بعنوان پروژه تحویل دهید.

با آرزوی سلامتی و موفقیت

عملکرد ماوس در ویندوز و برنامه‌های کاربردی

ماوس یکی دیگر از دستگاه‌های ورودی است که توسط دست کنترل می‌شود و قادر است حرکت دوبعدی را نسبت به یک سطح تشخیص دهد. این حرکت‌ها باعث جابجایی نشانگر موس بر روی صفحه‌نمایش می‌شود.

به طور معمول ماوس دستگاه کوچکی است با یک یا چند دکمه که توسط یک دست قابل کنترل است و برای کار با اشیای موجود بر روی صفحات نمایش گرافیکی کاربرد دارد.

ماوس‌ها در گذشته بر اساس حرکت چرخ (گوی متحرک) و در حال حاضر بر اساس سیستم تابش نور، اقدام به تشخیص حرکت در فضای دوبعدی می‌کنند. ماوس‌ها بر اساس حرکت دست کاربر، موقعیت نشانگر ماوس را بر روی صفحه‌نمایش گرافیکی کنترل می‌کنند. حرکت نشانگر در یک فضای دوبعدی در صفحه‌نمایش انجام می‌شود، بنابراین ماوس باید قادر باشد مختصات X, Y را نسبت به حرکت دست کاربر تعیین کند. ماوس علاوه بر امکان تشخیص حرکت، دارای دکمه‌هایی برای انجام امور قابل برنامه‌ریزی است. در حال حاضر ماوس‌ها عمدتاً دارای ۳ دکمه می‌باشند که عبارت‌اند از دکمه سمت چپ، دکمه سمت راست و دکمه وسط (دکمه اسکرو). ماوس قادر است همزمان با تشخیص حرکت، فشردن هر یک از این دکمه‌ها را نیز تشخیص داده و به سیستم‌عامل گزارش کند.

ما در برنامه‌نویسی قادر هستیم اجرای یک سری از امور و فعالیت‌ها را متناسب با تحقق هر یک از رویدادهای ماوس برنامه‌ریزی کنیم که از آن جمله می‌توان به برنامه‌ریزی اجرای یک فعالیت در صورت تحقق هر یک از رویدادهای زیر اشاره کرد.

- در صورت تحقق رویداد فشردن /رها کردن دکمه‌های ماوس
- در صورت تحقق رویداد حرکت ماوس
- در صورت تحقق ترکیبی از رویدادهای دو حالت قبلی

برنامه ریزی اجرای یک فعالیت در صورت تحقق رویداد فشردن / رها کردن دکمه های ماوس

رویدادهای فشردن و رها کردن دکمه های ماوس می تواند شامل موارد زیر باشد:

- فشردن دکمه موس
- رها کردن دکمه موس
- فشردن و رها کردن دکمه سمت چپ (یک بار : کلیک، دو بار : دابل کلیک)

برنامه ریزی اجرای یک فعالیت در صورت تحقق رویداد حرکت ماوس

حرکت ماوس می تواند شامل موارد زیر باشد:

- ورود نشانگر ماوس به محدوده یک شی
- حرکت نشانگر ماوس در محدوده یک شی
- خروج نشانگر ماوس از محدوده یک شی

برنامه ریزی اجرای یک فعالیت در صورت تحقق ترکیبی از رویدادهای دو حالت قبلی

ترکیب حرکت همزمان نشانگر ماوس درحالی که دکمه های آن نیز فشرده شده باشد، می تواند شامل موارد زیر باشد:

- فشردن ماوس بر روی یک شی و سپس حرکت ماوس و درنهایت رها کردن دکمه ماوس
- فشردن ماوس بر روی یک شی و سپس حرکت ماوس

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

کاربرد ماوس در برنامه‌های کاربردی	کاربرد ماوس در ویندوز
انتخاب، جابجایی، تغییر شکل و اندازه اشیا، کلیک کردن دکمه‌ها، انتخاب گزینه‌های منو	اجرای برنامه‌ها، کپی پرونده یا پوشه، جابجایی آیکون‌ها، فعال کردن منوهای کمکی با کلیک راست روی آیکون‌ها
کاربرد ماوس در بازی‌ها	کاربرد ماوس در نرم‌افزار MS-Word
انتخاب، جابجایی، تغییر شکل و اندازه اشیا، به‌کارگیری ماوس به جای دسته بازی	انتخاب بخشی از متن با ماوس، انتخاب واژه، جمله و پاره متن با یک یا چند بار کلیک روی متن، ترسیم اشکال

شرح رویدادهای جدول 1 را با توجه به شرح آن‌ها در VS بنویسید.

شرح رویداد در VS	شرح رویداد	نام رویداد
Occurs when the control is clicked by the mouse.	زمانی رخ می‌دهد که کنترل با ماوس کلیک شود.	MouseDown
Occurs when the control is double clicked by the mouse.	زمانی رخ می‌دهد که کنترل با ماوس دابل کلیک شود.	MouseDoubleClick
Occurs when the mouse pointer is over the control and a mouse button is pressed.	زمانی رخ می‌دهد که نشانگر موس بر روی کنترل باشد و دکمه ماوس فشرده شود.	MouseDown
Occurs when the mouse pointer is over the control and a mouse button is released.	زمانی رخ می‌دهد که نشانگر موس بر روی کنترل باشد و دکمه ماوس رها شود.	MouseUp

با انجام عملیات زیر روی ماوس عملیات چه رویدادی رخ می‌دهد؟

رویداد	عملیات
MouseDown	نگه‌داشتن دکمه راست یا چپ
MouseUp	رها کردن دکمه راست یا چپ

پودمان ۴:

کارگاه ۱ - واکنش به رویدادهای ماوس

ما در برنامه‌نویسی قادر هستیم اجرای یک سری از امور و فعالیت‌ها را متناسب با تحقق هر یک از رویدادهای ماوس برنامه‌ریزی کنیم. هدف از این کارگاه آموزش نحوه کار با رویدادهای وابسته به کلید در ماوس است. در این کارگاه هنرجو با برنامه‌ریزی رویدادهای `MouseDown`, `MouseClick`, `MouseUp` آشنا می‌شود و برنامه‌ریزی اجرای یک فعالیت در صورت تحقق رویداد فشردن/رها کردن دکمه‌های ماوس را فرامی‌گیرد.

کارگاه ۱ : واکنش برنامه به رویدادهای ماوس صفحه ۱۵۶



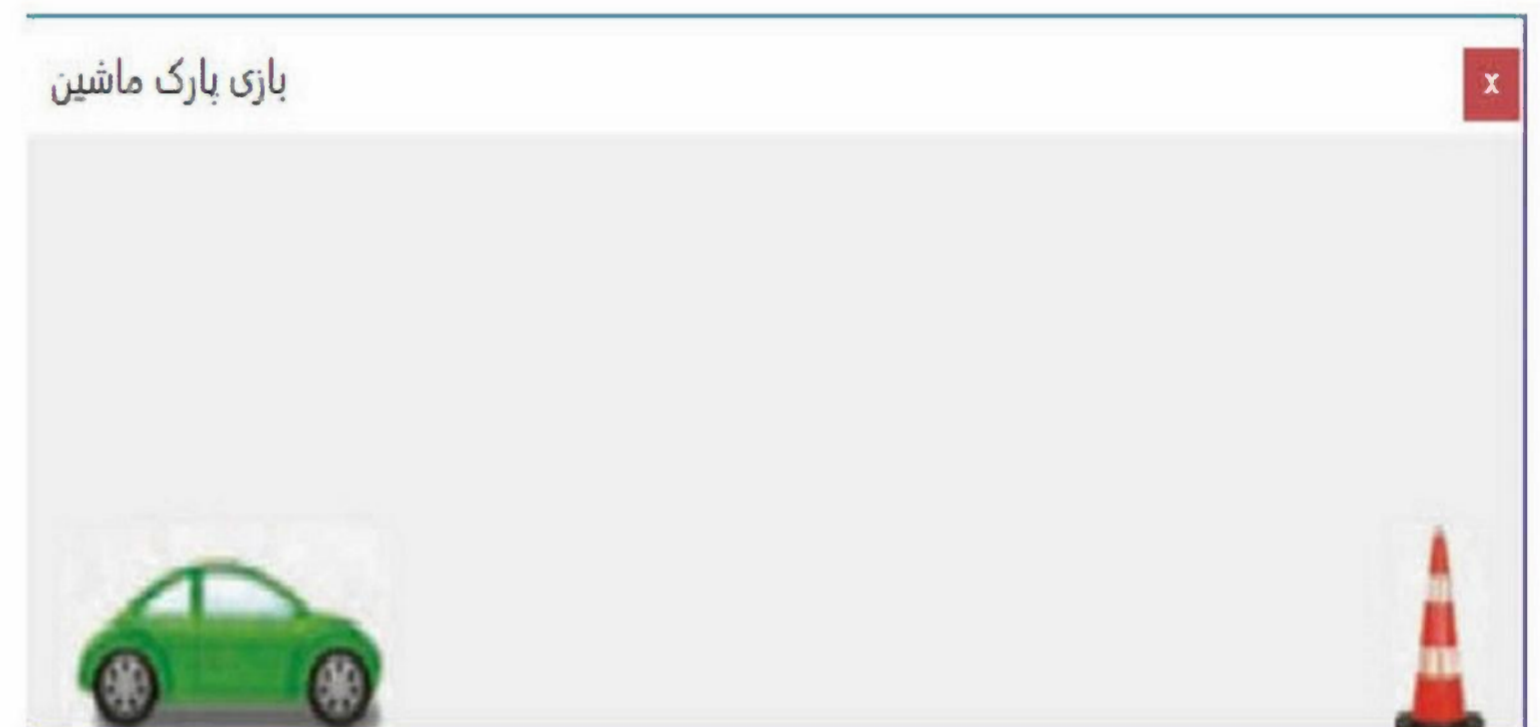
```
private void btnRun_Click(object sender, EventArgs e)
{
    MessageBox.Show("آمدید خوش شارپ سی به");
}

private void btnRun_MouseDown(object sender, MouseEventArgs e)
{
    btnRun.BackColor = Color.Red;
}

private void btnRun_MouseUp(object sender, MouseEventArgs e)
{
    btnRun.BackColor = Color.Blue;
}
```

کارگاه 2 - رویدادهای MouseUp و MouseDown

هدف از این کارگاه آموزش نحوه کار با رویدادهای وابسته به کلید در ماوس است. در این کارگاه هنرجو با برنامه‌ریزی رویدادهای MouseDown, MouseUp آشنا می‌شود و برنامه‌ریزی اجرای یک فعالیت در صورت تحقق رویداد فشردن/رها کردن دکمه‌های ماوس را فرامی‌گیرد. علاوه بر این موارد در این کارگاه یک زمان‌سنج نیز در نظر گرفته شده است که در زمان‌های مشخصی رویداد تیک آن فعال می‌شود و هنرجو قادر است با برنامه‌ریزی انجام یک فعالیت در زمان تحقق این رویداد، حرکت منظمی را برای تصویر ماشین بر روی صفحه‌نمایش شبیه‌سازی کند.



```
private void timer1_Tick(object sender, EventArgs e)
{
    picCar.Left += 5;
}

private void frm_page_158_MouseDown(object sender, MouseEventArgs e)
{
    timer1.Enabled = true;
}

private void frm_page_158_MouseUp(object sender, MouseEventArgs e)
{
    timer1.Enabled = false;
    if ((picCar.Left + picCar.Width + 10 > picBarrier.Left) &&
        (picCar.Left + picCar.Width < picBarrier.Left))
        MessageBox.Show("آفرین ، شما برنده شدید");
    else
        MessageBox.Show("شدید بازنده شما متاسفانه");
}
```

کارگاه ۳: رویدادهای mouse enter , mouse leave صفحه ۱۶۰

هدف از این کارگاه آموزش نحوه کار با رویدادهای مستقل از کلید در ماوس است. در این کارگاه هنجو با برنامه‌ریزی رویدادهای MouseEnter و MouseLeave آشنا می‌شود و برنامه‌ریزی اجرای یک فعالیت در صورت تحقق رویدادهای مرتبط با حرکت ماوس را فرامی‌گیرد.



```
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)
{
    lblBook.ForeColor = Color.Black;
    picBook.Visible = false;
    lblBook.Cursor = Cursors.Default;
}

private void lblBook_MouseHover(object sender, EventArgs e)
{
    ToolTip t = new ToolTip();
    t.SetToolTip(lblBook, "کتاب");
}

private void lblPen_MouseEnter(object sender, EventArgs e)
{
    lblPen.ForeColor = Color.Red;
    picPen.Visible = true;
    lblPen.Cursor = Cursors.Hand;
}

private void lblPen_MouseLeave(object sender, EventArgs e)
{
    lblPen.ForeColor = Color.Black;
    picPen.Visible = false;
    lblPen.Cursor = Cursors.Default;
}

private void lblPen_MouseHover(object sender, EventArgs e)
{
    ToolTip t = new ToolTip();
    t.SetToolTip(lblPen, "خونکاز");
}
}
```

کد فوق برای کتاب و خودکار نوشته شده است و مابقی نیز به همین صورت نوشته می‌شود.

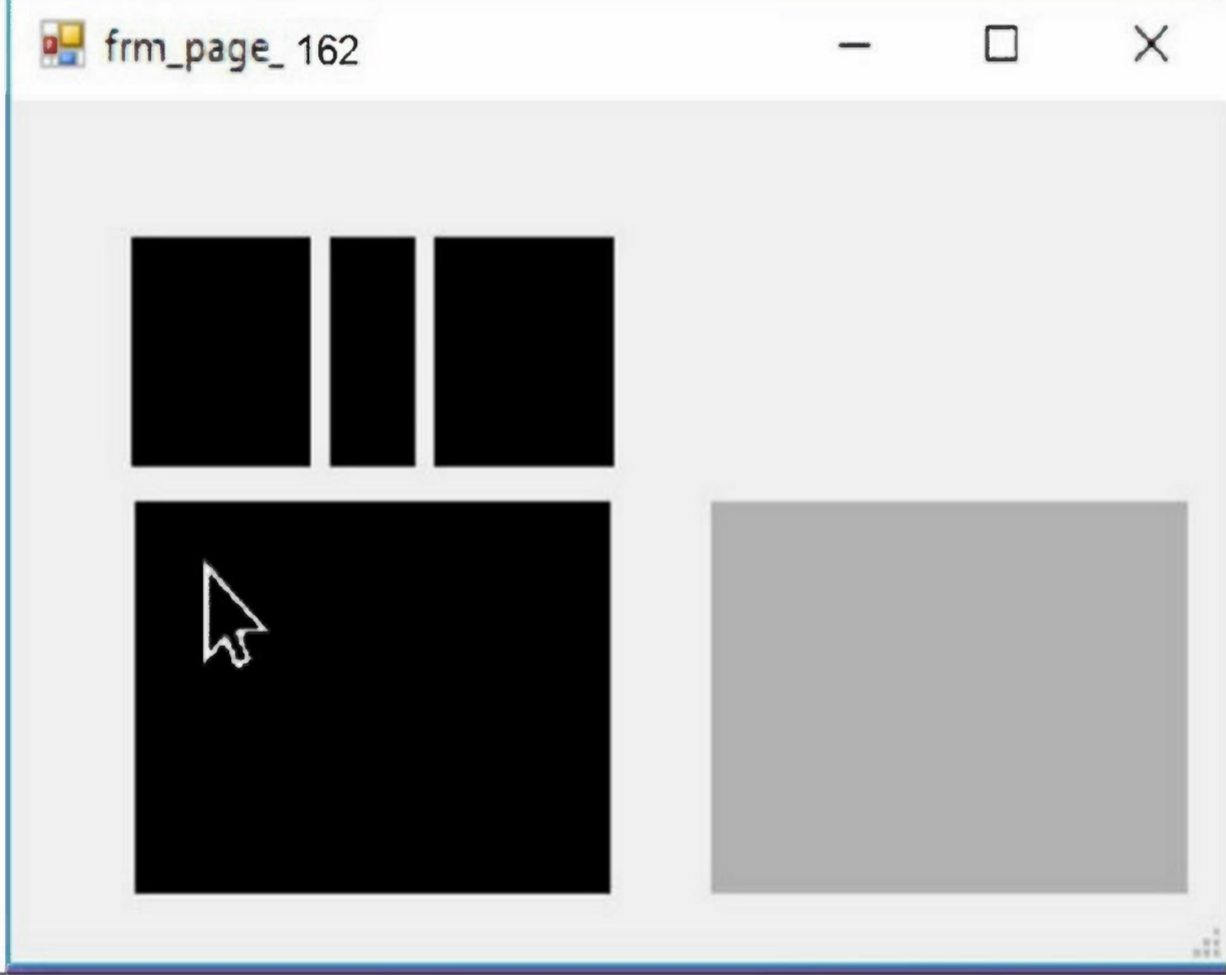
هدف از این کارگاه آموزش نحوه کار با رویدادهای مستقل از کلید و وابسته به کلید در ماوس است. به طور خلاصه در زیر ۷ رویداد معمول ماوس براساس اولویت گزارش در زمان وقوع و دسته بندی براساس استقلال از کلید و یا وابسته به کلید آورده شده است. رویدادهای مستقل از کلید شامل و

- 1.MouseEnter
- 2.MouseMove
5. MouseLeave

و رویدادهای وابسته به کلید شامل

- 3.MouseDown
 - 3.1.Click
 - 3.2.MouseClick
- 4.MouseUp

هستند که در بالا براساس ترتیب وقوع شماره گذاری شده اند. در سه کارگاه قبلی هنرجو با برنامه ریزی اجرای یک فعالیت در صورت تحقق رویداد فشردن / رها کردن دکمه های ماوس و نیز برنامه ریزی اجرای یک فعالیت در صورت تحقق رویداد حرکت ماوس آشنا شده است. در این کارگاه هنرجو با برنامه ریزی اجرای یک فعالیت در صورت تحقق ترکیبی از رویدادهای دو حالت قبلی آشنا می شود و برنامه ریزی اجرای یک فعالیت در صورت تحقق رویدادهای مرتبط با حرکت ماوس را فرامی گیرد.



```

private void MousePanel_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButton.Left)
        picLeft.BackColor = Color.White;
    else if (e.Button == MouseButton.Middle)
        picMiddel.BackColor = Color.White;

    else if (e.Button == MouseButton.Right)
        picRight.BackColor = Color.White;
}

private void MousePanel_MouseUp(object sender, MouseEventArgs e)
{
    picLeft.BackColor = Color.Black;
    picMiddel.BackColor = Color.Black;
    picRight.BackColor = Color.Black;
}

private void MousePanel_MouseMove(object sender, MouseEventArgs e)
{
    picCursor.Location = e.Location;
}

```


در VS با قرار دادن اشاره گر ماوس روی هر ویژگی، نوع و عملکرد آن به صورت راهنما در اختیار برنامه نویس قرار می گیرد. با توجه به این راهنما جدول 3 را تکمیل کنید.

نام ویژگی	نوع داده	شرح	شرح در VS
Button	داده شمارشی MouseButtons	مشخص می کند که کدام دکمه ماوس فشار داده شده است. (راست-چپ-وسط)	Gets which mouse button was pressed.
Location	کلاس Point	موقعیت نشان گر ماوس را روی کنترل مشخص می کند.	Gets the location of the mouse during the generating mouse event.
X	Int	موقعیت افقی نشانگر ماوس را در زمان ایجاد رویداد ماوس مشخص می کند	Gets the x-coordinate of the mouse during the generating mouse event.
Y	Int	موقعیت عمودی نشانگر ماوس را در زمان ایجاد رویداد ماوس مشخص می کند	Gets the y-coordinate of the mouse during the generating mouse event.
Clicks	Int	تعداد دفعات فشردن و رها شدن دکمه ماوس را مشخص می کند	Gets the number of times the button was pressed and released.

برنامه ای بنویسید که با حرکت ماوس در فضای فرم واژه Computer به همراه نشان گر ماوس نمایش داده شود.

یک کنترل برچسب بانام lblSample به فرم اضافه کنید و مقدار ویژگی Text آن Computer را قرار دهید. و برای حرکت از رویداد Form1_MouseMove استفاده کنید.

```
private void Form1_MouseMove(object sender,
MouseEventArgs e)
{
    lblSample.Location = e.Location;
}
```

برنامه بالا را به گونه ای ویرایش کنید که بجای واژه Computer مختصات ماوس به صورت (X,Y) به همراه نشان گر ماوس نمایش داده شود.

```
private void Form1_MouseMove(object sender,
MouseEventArgs e)
{
    lblSample.Location = e.Location;
    lblSample.Text = "(" + e.X + ", " + e.Y + ")";
}
```

کارگاه ۵ – رویدادهای ماوس و زمان سنج

هدف از این کارگاه آموزش نحوه کار با رویدادهای مربوط به زمان سنج و نیز رویدادهای مستقل از کلید و وابسته به کلید در ماوس است. در ۴ کارگاه قبلی هنرجو با برنامه ریزی اجرای یک فعالیت در صورت تحقق رویداد فشردن/رها کردن دکمه های ماوس و نیز برنامه ریزی اجرای یک فعالیت در صورت تحقق رویداد حرکت ماوس آشنا شده است. در این کارگاه هنرجو با برنامه ریزی اجرای یک فعالیت در صورت تحقق ترکیبی از رویدادهای دو حالت قبلی با اعمال محدودیت زمانی آشنا می شود و برنامه ریزی اجرای یک فعالیت در صورت تحقق رویدادهای مرتبط با حرکت ماوس در یک بازه زمانی معین را فرامی گیرد.

بازی پینگ پنگ با ماوس

یک کنترل برجسب بانام lblRacket برای راکت بازی و یک برجسب lblScore را برای امتیاز به فرم اضافه کنید و کدهای زیر به رویداد بالا اضافه کنید. با برخورد توپ به راکت سه امتیاز به کاربر اضافه می شود. اگر برخورد نکند یک امتیاز کسر و timer1 غیرفعال می شود و توپ در مکان اولیه متوقف می شود. یک کنترل کادر تصویر بانام picBall و یک کنترل زمان سنج به فرم اضافه می کنیم. برای حرکت خودکار توپ در فضای فرم از دو متغیر بولی up و right استفاده می کنیم.

```
public partial class Form1 : Form
{
    bool up = true, right = true; lblScore.Text="0";
    را با دستورات زیر می نویسیم. timer1_Tick متد رویداد
    private void timer1_Tick(object sender, EventArgs e)
    {
        picBall.Top += (up)?-5:5;
        picBall.Left +=(right)? 5:-5;
        right =(picBall.Left <= 0);
        int Score= int.Parse(lblScore.Text);
        up= (picBall.Top > 0)
            && (picBall.Top + picBall.Height+30>=
this.Height);
        if(picBall.Left + picBall.Width >= lblRacket.Left)
            if ((picBall.Top + picBall.Height >= lblRacket.Top)
&&
                (picBall.Top <= lblRacket.Top +
lblRacket.Height))
            {
                right = false;
                Score +=-3;
            }
            else
            {
                timer1.Enabled = false;
            }
    }
}
```

```

        Score += -1;
        picBall.Location = new Point(60, 190);
    }
    lblScore.Text=Score.ToString();
}

```

برای حرکت راکت lblRacket با ماوس در جهت عمودی از رویداد Form1_MouseMove استفاده می‌کنیم

```

private void Form1_MouseMove(object sender,
MouseEventArgs e)
{
    lblRacket.Top = e.Y;
}

```

برای محدودیت زمان ۶۰ ثانیه‌ای برنامه از کنترل برچسب lblTime و زمان‌سنج timer2 استفاده می‌کنیم و با دوبار کلیک روی فرم، توپ در فضای فرم به صورت خودکار به حرکت درمی‌آید. (timer1 فعال می‌شود)

```

private void timer2_Tick(object sender, EventArgs e)
{
    lblTime.Text =
        (byte.Parse(lblTime.Text) -1).ToString();
    if (lblTime.Text=="0")
    {
        timer2.Enabled = false;
        timer1.Enabled = false;
        if (int.Parse(lblScore.Text) >= 50)
            MessageBox.Show("برنده شدی");
        else
            MessageBox.Show("بازنده شدی");
    }
}
private void Form1_MouseDoubleClick(object sender,
MouseEventArgs e){
    timer1.Enabled = true;
    timer2.Enabled = true;
}

```

عملکرد صفحه کلید در ویندوز

صفحه کلید یکی از دستگاه‌های ورودی است که کاربر را قادر می‌سازد تا داده‌های موردنیاز را در اختیار نرم‌افزار قرار دهد، برای این منظور در صفحه کلید تعدادی کلید با عملکردهای متفاوتی وجود دارد، که می‌توان به کلیدهای الفبایی/ الفبا-عددی، کلیدهای نشان‌گذاری (کاما و نقطه و ...)، کلیدهای ناوبری و کلیدهای خاص (مثل Enter و Del و ESC و ...) اشاره کرد. در صفحه کلید ما با سه دسته رویداد روبرو هستیم که عبارت‌اند از: پایین رفتن کلید (KeyDown)، رها شدن/بالا آمدن کلید (KeyUp)، فشردن کلید (KeyPress). نکته مهم در خصوص رویدادهای صفحه کلید این است که رویدادهای پایین رفتن و رها شدن کلید برای همه کلیدها گزارش می‌شود، ولی رویداد فشردن کلید مخصوص کلیدهای الفبایی/ الفبا-عددی، کلیدهای نشان‌گذاری (کاما و نقطه و ...) و کلیدهای خاص (مثل Enter و Del و ESC و ...) است. در این دسته از کلیدها رویداد پایین رفتن کلید (KeyDown) در شروع سیکل فشردن شدن یک دکمه، و رویداد رها شدن کلید (KeyUp) در انتهای سیکل گزارش می‌شود و مادامی که کاربر کلید موردنظر را تحت فشار نگاه می‌دارد، به صورت پیاپی رویداد فشردن (KeyPress) گزارش می‌شود.

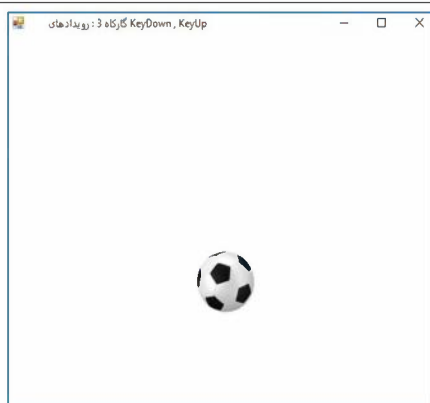
مثال: یکی از کلیدهای الفبایی را فشرده و برای چند لحظه رها نکنید، در این صورت سیستم‌عامل گزارشی به صورت زیر به برنامه ارسال می‌کند (از چپ به راست بخوانید): KeyDown, KeyPress, KeyPress, ..., KeyPress, KeyUp

رویدادهای صفحه کلید برای فرم

نام رویداد	شرح رویداد	شرح رویداد در VS
KeyDown	زمانی رخ می‌دهد که کلیدی فشرده شود.	Occurs when a key is first pressed.
KeyPress	زمانی رخ می‌دهد که فوکوس روی کنترل باشد و کاربر کلیدی را فشار دهد.	Occurs when the control has focus and the user presses and release a key.
KeyUp	زمانی رخ می‌دهد که کلید فشرده شده رها شود.	Occurs when a key is released.

گزاره ۳: رویدادهای KeyDown , KeyUp

می خواهیم برنامه ای بنویسیم که با نگه داشتن هر کلید دلخواه، توپ به سمت بالا پرتاب شود و با رها کردن کلید، توپ به سمت پایین سقوط کند.



```
bool isUp = false;
1 reference
public frm_page_184()
{
    InitializeComponent();
}
1 reference
private void frm_page_184_KeyDown(object sender, KeyEventArgs e)
{
    isUp = true;
    timer1.Enabled = true;
}
1 reference
private void frm_page_184_KeyUp(object sender, KeyEventArgs e)
{
    isUp = false;
}
1 reference
private void timer1_Tick(object sender, EventArgs e)
{
    if (isUp == true)
        picBall.Top -= 5;
    else
        picBall.Top += 5;
}
```

واکنش برنامه به فشردن کلیدهای ترکیبی

اگر کاربر بخواهد از کلیدهای ترکیبی مثل `ctrl+A` در برنامه استفاده کند باید از رویدادهای `KeyUp` و `KeyDown` استفاده کند.

می خواهیم برنامه ای بنویسیم که با فشردن کلید ترکیبی `ctrl` و جمع، تصویر بزرگ تر و با فشردن کلید ترکیبی `ctrl` و تفریق، تصویر کوچک تر شود.



1 reference

```
private void frm_page_189_KeyDown(object sender, KeyEventArgs e)
{
    double ratio = (double)picSample.Width / picSample.Height;
    if (e.Control == true)
    {
        if (e.KeyValue == 187)
            picSample.Height += 5;
        else if (e.KeyValue == 189)
            picSample.Height -= 5;

        picSample.Width = (int)(ratio * picSample.Height);
    }
}
```

آرگومان e در رویداد KeyDown و KeyPress

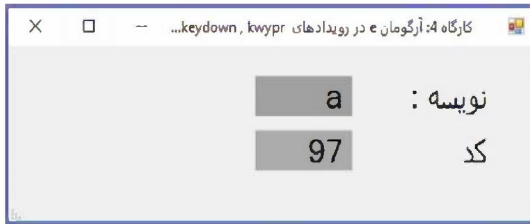
متد رویدادهای صفحه کلید دارای دو آرگومان است. تفاوت عملکرد رویدادها با آرگومان e مشخص می‌شود.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
```

آیا نوع آرگومان e در رویداد KeyDown و KeyPress یکسان است و ویژگی‌های یکسانی دارند؟
پاسخ: خیر، در رویداد KeyDown آرگومان e از نوع KeyEventArgs و در رویداد KeyPress از نوع KeyPressEventArgs است.
کد اسکی (ASCII code) چیست؟

پاسخ: یکی از شیوه‌های رایج کدگذاری حروف، ارقام و علامت‌ها استفاده از کدهای اسکی (ASCII) است. واژه اسکی (ASCII) مخفف و کوتاه شده عبارت "American Standard Code for Information Interchange" است که به معنای "استاندارد کدگذاری آمریکایی برای تبادل اطلاعات" است.

کارگاه ۴: آرگومان e در رویدادهای keydown , kwypress صفحه 185



```
1 reference
private void frm_page_185_KeyDown(object sender, KeyEventArgs e)
{
    // lblCharacter.Text = e.KeyCode.ToString();
    // lblCode.Text = e.KeyValue.ToString();
}

1 reference
private void frm_page_185_KeyPress(object sender, KeyPressEventArgs e)
{
    lblCharacter.Text = e.KeyChar.ToString();
    lblCode.Text = ((int)e.KeyChar).ToString();
}
```

رویدادهای صفحه کلید برای کنترل‌های دیگر

اگر به پنجره ویژگی‌های برخی از کنترل‌ها نظیر کادرمتن (TextBox) نگاه کنید، رویدادهای صفحه کلید برای این کنترل‌ها هم وجود دارد. زمانی یک کنترل به رویدادهای صفحه کلید واکنش نشان می‌دهد که فوکوس روی کنترل باشد.

فوکوس (focus)

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


کارگاه 7 استفاده از نوع داده شمارشی Keys

در ویژگی‌های آرگومان e جدول 6، ویژگی KeyCode از نوع داده شمارشی Keys است. نوع داده شمارشی Keys، مجموعه‌ای شامل همه نویسه‌های صفحه کلید یعنی کلیدهای حروف الفبا، کلیدهای عددی، کلیدهای کنترلی، کلیدهای تابعی، کلیدهای هدایت و صفحه کلید عددی (Numeric Keypad) است.

می‌خواهیم برنامه‌ای بنویسیم که با کلیدهای جهت‌دار (Arrow Keys)، یک سفینه را در فضای فرم در جهت‌های چهارگانه حرکت دهد



```
private void frm_page_191_KeyDown(object sender, KeyEventArgs e)
{
    switch (e.KeyCode)
    {
        case Keys.Left:
            picSpacecraft.Left -= 5 ;
            break;
        case Keys.Right:
            picSpacecraft.Left += 5;
            break;
        case Keys.Up:
            picSpacecraft.Top -= 5 ;
            break;
        case Keys.Down:
            picSpacecraft.Top += 5;
            break;
    }
}
```

کارگاه 8 رویداد KeyDown کنترل کادرمتن

می‌خواهیم برنامه‌ای بنویسیم که با فشردن کلید Enter، فوکوس از یک کادرمتن به کنترل بعدی منتقل شود. برخی از برنامه‌ها، دارای فرم ورود داده هستند که شامل چندین کنترل کادرمتن هست و کاربر برای ورود داده از صفحه کلید استفاده می‌کند و برای فوکوس روی کنترل‌ها می‌تواند از کلیک کردن یا کلید Tab استفاده کند. اما در برخی از برنامه‌ها، کاربر می‌تواند با فشردن کلید Enter به کادرمتن بعدی برود.

رویداد keyDown کنترل کادر متن

```
private void txtName_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        txtFamily.Focus();
}

1 reference
private void txtFamily_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        txtFatherName.Focus();
}

1 reference
private void txtFatherName_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        txtMelliCode.Focus();
}

1 reference
private void txtMelliCode_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        btnSave.Focus();
}
```

نوع داده نویسه (char) و متدهای آن

متغیر نوع char می تواند فقط شامل یک نویسه باشد و به صورت زیر اعلان می شود:

```
char ch = 'A';
```

نوع داده char در سی شارپ دارای متدهایی است که نوع نویسه شامل رقم بودن، حرف کوچک بودن، حرف بزرگ بودن و ... را مشخص می کند.

کارگاه ۹ استفاده از متدهای نوع داده char

می خواهیم برنامه ای بنویسیم که هنگام ورود داده در کادر متن، فقط کلیدهای ویژه مثلاً عددی عمل کند. برای نمونه در شکل ۱۷ در کادر متن مربوط به شماره دانش آموزی، فقط نویسه های رقمی پذیرفته شود.

```
private void txtStudentNO_KeyDown(object sender, KeyEventArgs e)
{
    if (!char.IsDigit((char)e.KeyCode))
        e.SuppressKeyPress = true;
}

private void txtFirstName_KeyDown(object sender, KeyEventArgs e)
{
    if ((char.IsLetter((char)e.KeyCode) || (e.KeyCode == Keys.Back)))
        e.SuppressKeyPress = true;
}

private void txtFirstName_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(char.IsLower(e.KeyChar)))
        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);
    }
}
```