

# ایجاد برنامه‌های ویندوزی با واکنش نسبت به رویدادها

در فصل ۵ با برنامه‌های ویندوزی و کنترل‌های فرم، برجسب و جعبه تصویر آشنا شدید. در محیط VS برنامه‌های ویندوزی ایجاد کرده، فرم ساده طراحی کردید و بدون نوشتن حتی یک خط برنامه، آن را اجرا نمودید؛ در این فصل علاوه بر آشنایی با کنترل‌های جدید، برای رویدادها و عملیاتی که کاربر انجام می‌دهد برنامه نوشته می‌شود.

### پس از پایان این فصل انتظار می‌رود که فراگیر بتواند:

- ۱- در برنامه‌های ویندوزی از کنترل دکمه استفاده نماید و ویژگی‌های مورد نیاز را تنظیم نماید.
- ۲- مفهوم رویداد را توضیح دهد و در رویداد کلیک ماوس، برنامه بنویسد.
- ۳- برای واکنش به رویدادها Event Handler بنویسد.
- ۴- به برنامه‌های خود کنترل جعبه متن اضافه کند و ویژگی‌های آن را تعیین کند.
- ۵- با استفاده از کنترل زمان سنج و تنظیم ویژگی‌های آن، برنامه‌های کاربردی بنویسد.
- ۶- با کنترل‌های Dialogs برای تغییر رنگ و قلم و تنظیمات پرونده، برنامه بنویسد.

## کار در کارگاه ۱: آشنایی با کنترل دکمه

### مثال ۱-۶: برنامه ویندوزی ساده با واکنش به رویداد کلیک

۱- یک پروژه جدید ویندوزی بسازید.

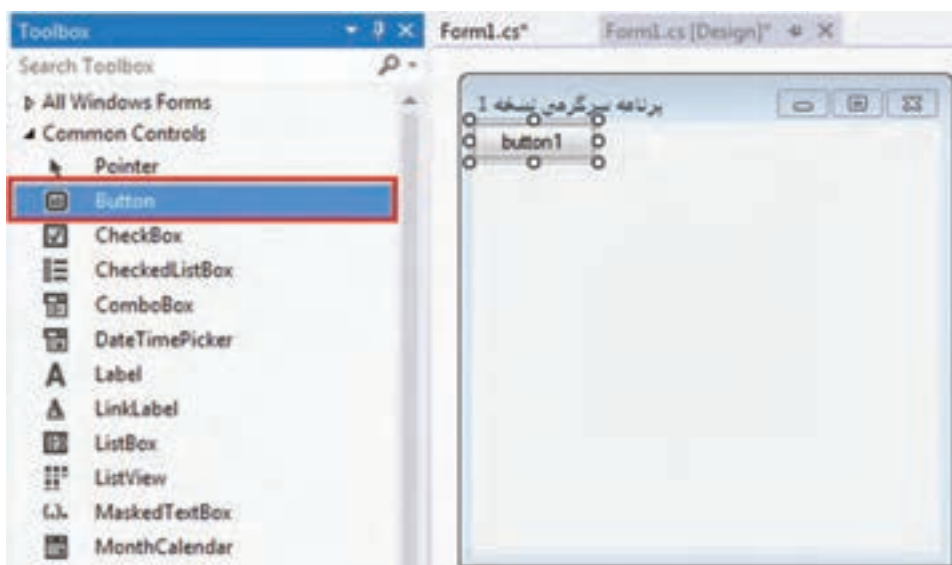
۲- ویژگی‌های فرم را مانند جدول ۱-۶ تنظیم کنید.

جدول ۱-۶- ویژگی‌های فرم

Form	
ویژگی	مقدار
Name	form1
Text	برنامه سرگرمی نسخه ۱
RightToLeft	Yes

۳- به جعبه ابزار مراجعه کرده و کنترل Button را پیدا کرده و آن را روی فرم قرار دهید. شکل

۱-۶ را مشاهده کنید.



شکل ۱-۶- قرار دادن کنترل Button روی فرم

۴- ویژگی‌های دکمه را مطابق جدول زیر تنظیم کنید.

جدول ۶-۲- ویژگی‌های Button

Button	
ویژگی	مقدار
Name	button1
Text	لبخند بزنید
RightToLeft	Yes
Font	Arial Size : 16
AutoSize	true
Location	Center in Form   Horizontally Center in Form   Vertically

۵- اکنون برنامه را اجرا کنید باید فرم ایجاد شده مانند شکل ۶-۲ باشد.



شکل ۶-۲

۶- با اجرای برنامه و کلیک کردن بر روی دکمه ، هیچ اتفاقی نمی افتد. نه تنها لبخندی در کار نیست شاید کاربر کمی هم دلگیر شود که چرا برنامه، کار خاصی انجام نمی دهد. در برنامه بعدی با روش ایجاد عکس العمل و واکنش نسبت به رویداد کلیک آشنا می شوید.

۷- گاهی اوقات علامت ها و نمادها می توانند خیلی بهتر از نوشته ها پیام خود را برسانند. مثلاً یک کنترل تلویزیون را در نظر بگیرید، در روی بعضی از دکمه های آن، علامت هایی حک شده است که به سرعت عملکرد کلید را نشان می دهد. در این مرحله می خواهیم به جای عبارت «لبخند بزن» یک نماد مثلاً 😊 را قرار دهیم.

۸- دو ویژگی Font و Text دکمه را که در جدول ۳-۶ با رنگ قرمز نشان داده شده است تغییر دهید. توجه داشته باشید که در ویژگی Text، حرف J بزرگ است.

جدول ۳-۶- ویژگی های Button

Button	
ویژگی	مقدار
Name	button1
Text	J
RightToLeft	No
Font	Wingdings Size : 28
AutoSize	True
Location	Center in Form   Horizontally  Center in Form   Vertically

۹- اکنون برنامه را اجرا کنید. نتیجه کار باید شبیه شکل ۳-۶ باشد

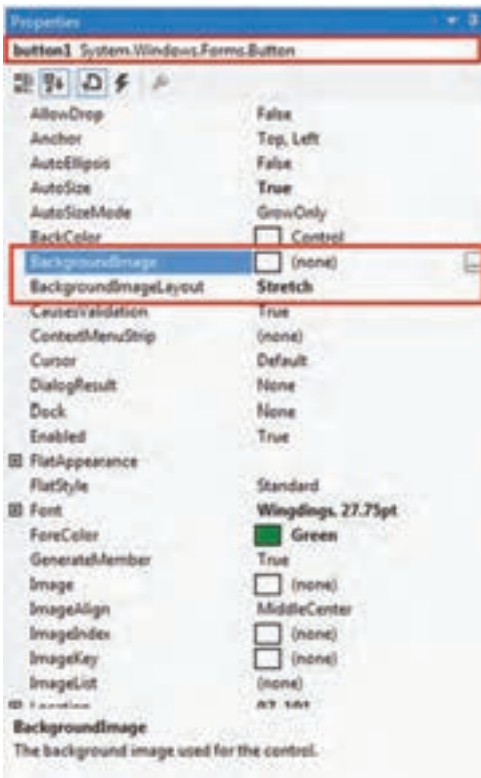
**سؤال:** چگونه می‌توانید رنگ نماد  
لیخند، را مانند شکل ۳-۶ به رنگ سبز درآورید؟



شکل ۳-۶

۱۰- ویژگی Text دکمه را به حرف K (بزرگ) تغییر دهید و برنامه را اجرا کنید. اگر مایل هستید یک بار هم با حرف L (بزرگ) امتحان کنید. اگر به استفاده از این نمادها علاقمند

هستید، وارد برنامه Word شوید و فونت را Wingdings قرار داده و با زدن کلیدهای صفحه کلید، نمادهای مختلفی را شناسایی کرده و از آنها در برنامه خود استفاده نمایید.



۱۱- علاوه بر نوشته یا نماد، می‌توانید برای روشن شدن کاربرد یک دکمه، تصویری را نیز به عنوان زمینه دکمه با استفاده از ویژگی BackgroundImage معین کنید. اگر اندازه تصویر مورد نظر با اندازه دکمه متفاوت است، مقدار ویژگی BackgroundImageLayout را برابر با Stretch قرار داده تا کل تصویر در روی دکمه جای داده شود. توجه داشته باشید که اگر متنی را نیز در ویژگی Text نوشته باشید، متن روی عکس زمینه قابل مشاهده است. شکل ۴-۶ را مشاهده کنید.

شکل ۴-۶ ویژگی‌های کنترل Button

۱۲- یکی دیگر از ویژگی‌هایی که در شکل ۴-۶ مشاهده می‌کنید، ویژگی Image است که برای انتخاب یک تصویر بر روی دکمه استفاده می‌شود. در صورت استفاده از چنین ویژگی، نوشته موجود در ویژگی Text دیده نمی‌شود چون تصویر روی آن را می‌پوشاند.

## کار در کارگاه ۲: کار با دکمه

در این بخش خواهید دید که VS، در تولید برنامه‌هایی که نسبت به عملیات کاربر، واکنش نشان می‌دهد، بسیاری از دستورات و عملیات را در پشت صحنه انجام می‌دهد و برنامه نویس تنها با چند کلیک ماوس و نوشتن دستورات متد مربوطه، می‌تواند برنامه‌های واکنش دار را بسازد.

**مثال ۲-۶:** ایجاد یک فرم شامل یک دکمه و واکنش نسبت به رویداد کلیک ماوس  
در این مثال می‌خواهیم با اجرای برنامه، فرمی مطابق با شکل ۵-۶ نشان داده شود که ظاهراً فقط شامل یک دکمه است. برنامه در وضعیت انتظار باقی می‌ماند تا کاربر، عملی را انجام دهد. هنگامی که کاربر بر روی دکمه «بگو سلام» کلیک کرد، پیامی مطابق با شکل ۶-۶ ظاهر می‌گردد.



شکل ۶-۶ - نمایش پیام به محض کلیک بر روی دکمه



شکل ۵-۶ - برنامه در حالت انتظار

مراحل اجرای این مثال را قدم به قدم به صورت زیر انجام می‌دهیم:

۱- ایجاد پروژه: وارد برنامه VS شوید و یک پروژه جدید از نوع Windows Forms Application با نام پروژه EventDemo و در مسیر مشخصی بسازید.

۲- تعیین ویژگی های فرم : ویژگی های فرم را به صورت جدول ۴-۶ تنظیم کنید.

جدول ۴-۶ - ویژگی های فرم

Form		
ویژگی	مقدار	
Name	Form1	
Text	واکنش به رویداد کلیک	
Size	Width	300
	Height	300

۳- اضافه کردن یک متن نمایشی بر روی فرم : یک کنترل برجسب بر روی فرم قرار دهید و ویژگی های آن را مانند جدول ۵-۶ تعیین کنید.

جدول ۵-۶ - ویژگی های برجسب

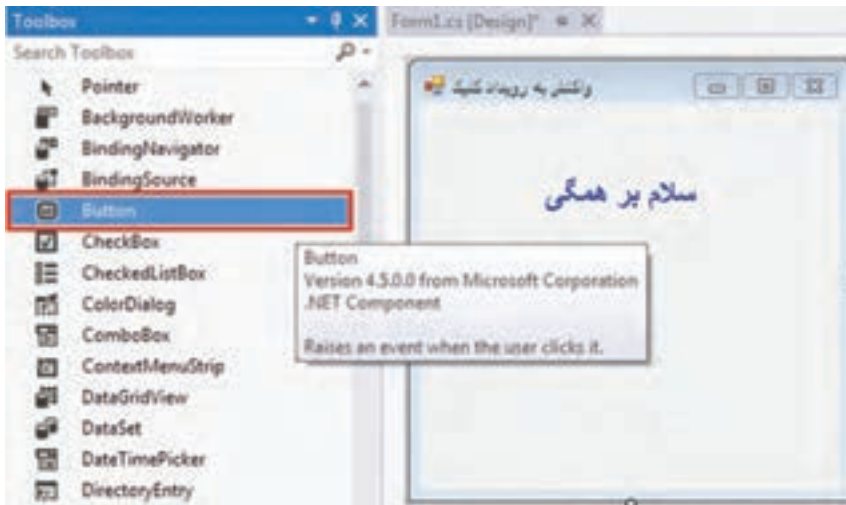
Label		
ویژگی	مقدار	
Name	Label1	
Text	سلام بر همگی	
RightToLeft	True	
Font	Arial Size : 16 Bold	
ForeColor	Blue	
Location	X	Center in Form   Horizontally
	Y	50
Visible	false	

اگر برنامه را در این مرحله اجرا کنید، مشاهده خواهید کرد که فقط یک فرم خالی دیده می‌شود. شاید تعجب کنید که، با وجود اینکه در روی فرم، یک کنترل برچسب قرار دادید، اما در هنگام اجرای برنامه دیده نمی‌شود. فکر می‌کنید علت چیست؟

اگر به ردیف آخر جدول ۵-۶ دقت کنید، متوجه می‌شوید که یک ویژگی جدید به نام Visible استفاده شده است که مقدار آن را false قرار داده‌ایم. با توجه به معنی کلمه Visible (قابل مشاهده) و مقداری که برای این ویژگی تعیین کردیم، باید انتظار داشت که کنترل برچسب نشان داده نشود. هرگاه مقدار true در این ویژگی قرار دهیم کنترل برچسب نشان داده می‌شود.

حال طراحی برنامه را ادامه می‌دهیم و در این مرحله می‌خواهیم کاری کنیم تا کاربر با کلیک بر روی ماوس، برچسب را نمایان کند.

**۴- اضافه کردن دکمه بر روی فرم :** در این مرحله یک دکمه به فرم اضافه می‌کنیم برای این منظور به جعبه ابزار مراجعه و کنترل دکمه را مطابق شکل ۷-۶ پیدا کرده و به فرم اضافه کنید.

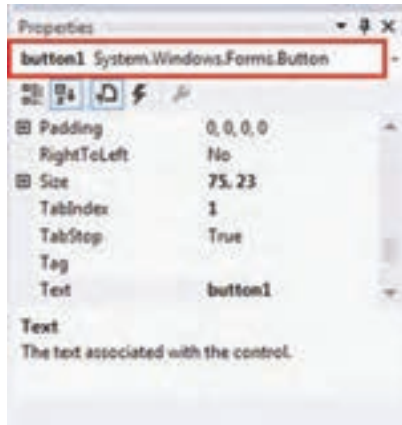


شکل ۷-۶- انتخاب کنترل دکمه از جعبه ابزار

به محض قرارگیری کنترل دکمه بر روی فرم، پنجره Properties به طور خودکار، ویژگی‌های این کنترل را نشان می‌دهد. نام شیء دکمه، button1 است که به وسیلهٔ VS انتخاب می‌شود. این شیء از کلاس Button ساخته شده است.

حال ویژگی‌های دکمه را مانند جدول ۶-۶ تنظیم کنید.





شکل ۸-۶ پنجره Properties برای نمایش ویژگی های دکمه

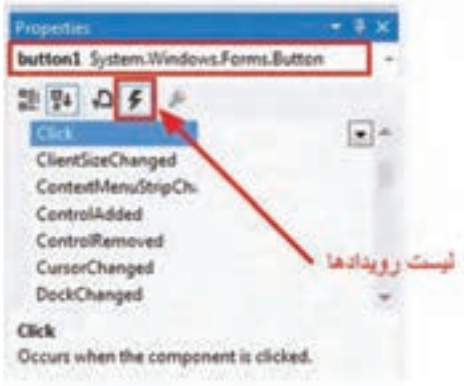
جدول ۶-۶- ویژگی های Button

دکمه Button		
ویژگی	مقدار	
Name	button1	
Text	بگو سلام	
Location	X	Center in Form   Horizontally
	Y	200

حال برنامه را اجرا و روی دکمه «بگو سلام» کلیک کنید، مشاهده می شود که هیچ اتفاقی نمی افتد. طبیعی است که برنامه عکس العملی نسبت به رویداد کلیک نداشته باشد چون هیچ متدی برای رویداد کلیک بر روی دکمه ایجاد نکردیم و همچنین هیچ دستوری برای نمایش پیام و تغییر ویژگی Visible برچسب نوشته نشده است. حال از برنامه خارج شده و به محیط طراحی فرم باز می گردیم.

در پنجره Properties علاوه بر لیست ویژگی ها، می توانید لیست رویدادهای مربوط به دکمه را نیز مشاهده کنید. با کلیک بر روی آیکن رعد و برق، مانند شکل ۹-۶ لیست رویدادها مشاهده خواهد

شد. مقدار تمام رویدادها خالی است، یعنی هیچ واکنشی نسبت به رویدادها ثبت نشده است.



شکل ۹-۶ پنجره Properties برای نمایش رویدادهای دکمه

در ردیف اول لیست رویدادها، رویداد Click را مشاهده می‌کنید. این رویداد مربوط به کلیک ماوس است. در مرحله بعدی خواهید دید که چگونه واکنشی نسبت به این رویداد ثبت می‌کنیم.

**۵-** واکنش برنامه نسبت به رویداد کلیک ماوس: هنگامی که کاربر بر روی دکمه «بگو سلام»، کلیک کرد، باید پیام «سلام بر همگی» دیده شود. برای انجام این کار، باید دستوری اجرا گردد تا ویژگی Visible برچسب برابر true شود.

```
label1.Visible = true;
```

**سؤال؟** این دستور کجا باید نوشته شود؟

برای نوشتن برنامه‌هایی که باید به عمل کاربر واکنش نشان دهند، لازم است دو مرحله زیر انجام شود:

۱- نوشتن یک متد (EH)

۲- ثبت کردن رویداد ماوس و برقرار کردن ارتباط رویداد و متد

خوشبختانه VS در انجام دو عمل فوق، شما را تنها نمی‌گذارد و بیشتر عملیات را خودش انجام می‌دهد فقط بایستی دستورات داخل متد EH را بنویسیم.

**مرحله اول: نوشتن متد EH:** برای اینکه بتوانید متد را بنویسید در حالی که در صفحه طراحی فرم هستید، روی دکمه «بگو سلام» دوبار کلیک کنید. در این صورت پنجره کد نویسی مانند شکل ۱۰-۶ باز می‌شود که در آن دستورات مربوط به خط عنوان متد به طور خودکار نوشته شده است.

```

namespace EventDemo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}

```

شکل ۱۰-۶- پنجره کد نویسی برای رویداد کلیک دکمه

نام متد، button1\_Click است که به وسیلهٔ VS انتخاب شده است. فقط بایستی در داخل متد، دستورات مورد نظر را بنویسیم. بنابراین دستور را مانند برنامه ۱-۶ می‌نویسیم:

```

using System. Collections. Generic;
using System. ComponentMode 1;
using System. Drawing;
using System. Text;
using System. Windows. Forms;

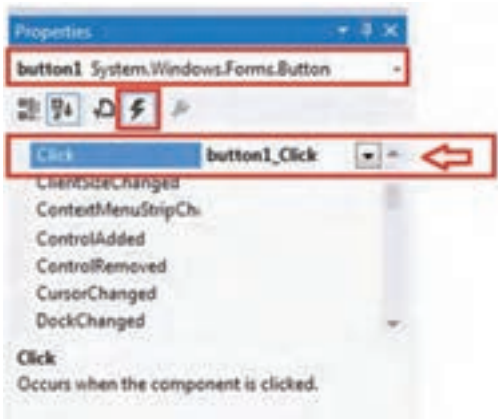
```

```

namespace EventDemo
{
    public partial class Form1: Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click (object sender, EventArgs e)
        {
            label1. Visible = true;
        }
    }
}

```



شکل ۱۱-۶ - تعیین متد EH مربوط به رویداد کلیک دکمه

### مرحله دوم : ثبت متد EH

برای رویداد کلیک : باید متد EH را برای یک رویداد ثبت کنیم. خوشبختانه این کار را VS به طور خودکار انجام می دهد و شما هیچ کاری لازم نیست انجام دهید. در این برنامه، VS به طور خودکار متد `button1_Click()` را برای رویداد Click دکمه ثبت می کند. برای اینکه مطمئن شوید که این کار صورت گرفته، کافی است به صفحه طراحی فرم برگشته و

در پنجره Properties بر روی آیکن رعد و برق کلیک نمایید تا لیست رویدادها ظاهر گردد. در لیست به رویداد Click توجه کنید. در جلوی رویداد Click، نام متدی نوشته شده است که در آن متد، دستور ظاهر کردن قرار دارد. شکل ۱۱-۶ را مشاهده کنید.

انکون برنامه را اجرا کنید و رویداد کلیک را آزمایش نمایید. از برنامه خارج شده و آن را مجدداً اجرا نمایید.

**سؤال:** چگونه از طریق پنجره Properties، می توانید واکنش نسبت به رویداد Click دکمه را از کار بیاندازید؟

تبریک می گوئیم که توانستید اولین برنامه ویندوزی با واکنش نسبت به کلیک ماوس را با همیاری VS بنویسید. در این مثال فقط یک خط دستور نوشتید و بقیه عملیات را با ماوس انجام دادید.

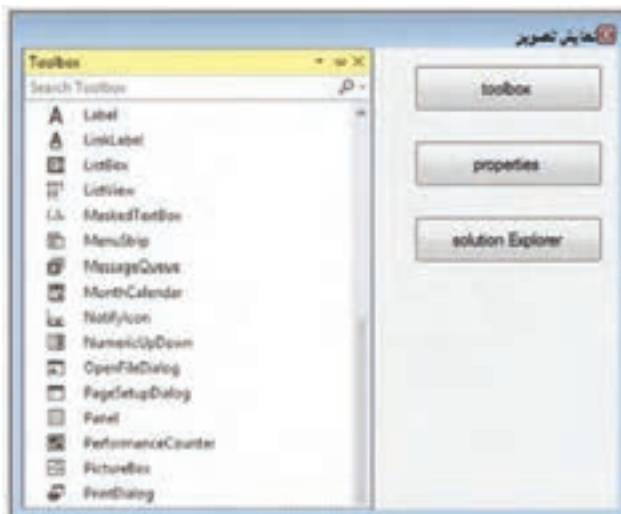
**توسعه و بهبود برنامه:** یک جعبه تصویری به فرم قبل اضافه کنید به طوری که بین پیام و دکمه مانند شکل ۱۱-۶ قرار گیرد. تصویر گل لاله<sup>۱</sup> را در آن قرار دهید و اگر لازم است موقعیت برچسب و دکمه را کمی تغییر دهید تا تصویر بهتر در روی فرم جای گیرد. با اجرای برنامه، یک فرم خالی نمایش داده شود (تصویر و پیام هر دو دیده نشوند) اما هنگامی که کاربر بر روی دکمه «بگو سلام» کلیک کرد هر دو نشان داده شوند.



شکل ۱۲-۶ - اضافه کردن یک جعبه تصویری

### کار در کارگاه ۳: کار با جعبه تصویر

**مثال ۳-۶:** ایجاد فرمی با سه دکمه و یک جعبه تصویر برای نمایش سه تصویر در این مثال می‌خواهیم فرمی مانند شکل ۱۳-۶ با سه دکمه ایجاد کنیم و سه تصویر مختلف را در PictureBox نمایش دهیم به صورتی که با کلیک روی هر دکمه، تصویر متناظر با آن دکمه در PictureBox قرار گیرد. ما از تصویر پنجره‌های Toolbox، Properties و Solution محیط IDE ویژوال استودیو استفاده می‌کنیم.



شکل ۱۳-۶ - فرمی با سه دکمه و PictureBox برای نمایش تصویر

۱- ایجاد پروژه: وارد برنامه VS شوید و پروژه جدیدی از نوع Windows Forms Application با نام Picture Show در مسیر مشخص بسازید.

۲- تعیین ویژگی های فرم

جدول ۶-۷- ویژگی های فرم

Form		
ویژگی		مقدار
Name		Form1
Text		نمایش تصویر
FormBorderStyle		FixedToolWindow
RightToLeft		Yes
Size	Width	440
	Height	362

۳- تعیین ویژگی دکمه ها و جعبه تصویر

جدول ۶-۹- ویژگی های دکمه دوم

Button2		
ویژگی		مقدار
Name		toolboxBtn
Text		toolbox
Size	Width	134
	Height	33

جدول ۶-۸- ویژگی های دکمه اول

Button1		
ویژگی		مقدار
Name		propertiesBtn
Text		properties
Size	Width	134
	Height	33

جدول ۱۰-۶ ویژگی های دکمه سوم

Button3		مقدار	ویژگی
Name		pic	
Text		solution Explorer	
Size	Width	134	
	Height	33	

جدول ۱۱-۶ ویژگی های جعبه تصویر

PictureBox			مقدار	ویژگی
Name			pic	
Dock			Left	
SizeMode			StretchImage	
Size	Width	250		
	Height	328		

ویژگی Image کنترل PictureBox را در پنجره Properties پیدا کرده روی علامت ... کلیک کنید تا جعبه محاوره ای انتخاب منبع (SelectResource) باز شود با استفاده از دکمه Import سه تصویر را به لیست اضافه کنید و در پایان یکی از تصاویر را از لیست انتخاب کرده و روی دکمه Ok کلیک کنید.

تصاویر اضافه شده در پنجره Solution Explorer در پوشه Resources قرار می گیرد.

**۴-** ایجاد واکنش نسبت به رویداد کلیک ماوس : با کلیک هر یک از دکمه ها تصویر PictureBox باید تغییر کند بنابراین باید ویژگی Image آن را تغییر دهید. همانطور که قبلاً مشاهده کردید در هنگام انتخاب عکس برای کنترل pictureBox در جعبه محاوره ای انتخاب منبع، گزینه Project resource file را انتخاب کردید. به این ترتیب با وارد کردن یک یا چند تصویر، یکی را به عنوان تصویر مورد نظر خود انتخاب نمودید. در واقع با این کار، منابعی از جنس تصویر برای نرم افزار ایجاد کردید و از آن استفاده نمودید. این منابع در داخل نرم افزار شما جایگذاری می شوند. اگر بخواهید از منابع معرفی شده در نرم افزار خود در زمان اجرا (در کد) استفاده کنید باید از کلاس Resources که در فضای نام Properties است، به صورت زیر استفاده نمایید.

نام منبع . Resources . Properties

در اینجا منابع ما از جنس تصویر (image) هستند و برای این مثال، چون نام تصویر جعبه ابزار toolbox بوده است به همین نام در منابع موجود است و از آن می‌توانیم استفاده نماییم:

Properties.Resources.toolbox

در فضای طراحی فرم روی دکمه‌ها دابل کلیک کنید تا الگوی رویداد کلیک آنها نوشته شود، سپس کد را مانند زیر تکمیل نمایید.

```
private void propertiesBtn_Click(object sender, EventArgs e)
{
    pic.Image = Properties.Resources.properties;
}
```

```
private void solutionBtn_Click(object sender, EventArgs e)
{
    pic.Image = Properties.Resources.solution;
}
```

```
private void toolboxBtn_Click(object sender, EventArgs e)
{
    pic.Image = Properties.Resources.toolbox;
}
```

برنامه ۶-۲

**توسعه و بهبود برنامه:** در برنامه فوق یک کنترل برچسب اضافه کنید که نام تصویری را نشان دهد که در PictureBox قرار دارد؟

## کار در کارگاه ۴: بزرگ کردن تصویر

**مثال ۴-۶:** در این مثال می‌خواهیم فرمی مانند شکل ۶-۱۴ با سه دکمه ایجاد کنیم و سه تصویر مختلف را روی دکمه‌ها قرار دهیم و با کلیک روی هر دکمه، تصویر آن دکمه در PictureBox قرار گیرد. در این مثال به جای متن، روی دکمه‌ها تصویر قرار می‌دهیم.





شکل ۱۴-۶- فرمی با سه دکمه و جعبه تصویر برای نمایش تصویر

- ۱- ایجاد پروژه: وارد برنامه VS شوید و پروژه جدیدی از نوع `WindowsFormsApplication` با نام `PictureZoom` در مسیر مشخص بسازید.
- ۲- تعیین ویژگی های فرم

جدول ۱۲-۶- ویژگی های فرم

Form		مقدار
ویژگی		
Name		Form1
Text		بزرگ کردن تصویر
RightToLeft		Yes
Size	Width	440
	Height	362

### ۳- تعیین ویژگی های دکمه ها و جعبه تصویر

جدول ۱۳-۶ ویژگی های دکمه اول

Button1		
ویژگی		مقدار
Name		pic1Btn
Text		
BackgroundImage		تصویر دلخواه ۱
BackgroundImageLayout		Stretch
Size	Width	126
	Height	90

جدول ۱۴-۶ ویژگی های دکمه دوم

Button2		
ویژگی		مقدار
Name		pic2Btn
Text		
BackgroundImage		تصویر دلخواه ۲
BackgroundImageLayout		Stretch
Size	Width	126
	Height	90

جدول ۱۵-۶ ویژگی های دکمه سوم

Button3		
ویژگی		مقدار
Name		pic3Btn
Text		
BackgroundImage		تصویر دلخواه ۳
BackgroundImageLayout		Stretch
Size	Width	126
	Height	90

جدول ۱۶-۶ ویژگی های جعبه تصویر

PictureBox		
ویژگی		مقدار
Name		piclarge
SizeMode		StretchImage
Size	Width	250
	Height	328

برای قرار دادن تصویر روی دکمه ها، در فضای طراحی روی دکمه کلیک کنید تا در پنجره Properties ویژگی های آن نمایش داده شود. ویژگی BackgroundImage دکمه را پیدا کرده و روی علامت ... کلیک کنید تا جعبه محاوره ای انتخاب منبع باز شود و سپس با استفاده از دکمه Import

سه تصویر را به لیست اضافه کنید. تصاویر اضافه شده در پنجره Solution Explorer در پوشه Resources قرار می‌گیرد. در پایان یکی از تصاویر اضافه شده به لیست را انتخاب کرده و روی دکمه Ok کلیک کنید.

برای اینکه تصویر روی دکمه‌ها و PictureBox کامل دیده شود باید ویژگی BackgroundImageLayout دکمه‌ها و PictureBox SizeMode کنترل PictureBox را مساوی Stretch قرار دهید.

مقدار ویژگی Text دکمه‌ها را پاک کنید تا روی تصاویر متنی نباشد و فقط تصویر زمینه دیده شود.

**۴-** ایجاد و اکنش نسبت به رویداد کلیک ماوس: با کلیک هر یک از دکمه‌ها، تصویر دکمه باید در PictureBox نمایش داده شود. بنابراین باید ویژگی Image آن را مساوی BackgroundImage دکمه قرار دهیم.

درفضای طراحی فرم روی دکمه‌ها دابل کلیک کنید تا الگوی رویداد کلیک آنها نوشته شود، سپس کد را مانند زیر تکمیل نمایید.

```
private void pic1Btn_Click(object sender, EventArgs e)
{
    piclarge.Image = pic1Btn.BackgroundImage;
}

private void pic2Btn_Click(object sender, EventArgs e)
{
    piclarge.Image = pic2Btn.BackgroundImage;
}

private void pic3Btn_Click(object sender, EventArgs e)
{
    piclarge.Image = pic3Btn.BackgroundImage;
}
```

برنامه ۳-۶

### مثال ۵-۶: حرکت دادن برجسب با کمک دکمه‌ها

در این مثال می‌خواهیم فرمی مانند شکل ۱۵-۶ طراحی کنیم که کاربر بتواند یک برجسب را در چهار جهت اصلی بالا، پایین، چپ و راست حرکت دهد و از جهت آخرین حرکت برجسب با اطلاع باشد.

الگوریتم و روش انجام کار: برای تعیین جهت حرکت از دکمه استفاده می‌کنیم. با کلیک روی هر دکمه فرمان، برجسب در جهت متناظر با آن دکمه فرمان، ۱۰ واحد جابه‌جا می‌شود و جهت حرکت نیز روی برجسب نمایش داده می‌شود. بنابراین به چهار دکمه فرمان و یک کنترل برجسب نیاز داریم.



شکل ۱۵-۶ - فرم جابه‌جایی برجسب

۱- ایجاد پروژه: وارد برنامه VS شوید و پروژه جدیدی از نوع `WindowsFormsApplication` به نام `MoveLabelProj` ایجاد نمایید.

۲- درج کنترل‌ها: روی فرم چهار کنترل دکمه و یک برجسب قرار دهید.

۳- تنظیم ویژگی‌های برجسب و فرم: ویژگی‌های فرم و برجسب را طبق جداول زیر تنظیم کنید.

جدول ۱۷-۶- ویژگی های برجسب

Label		ویژگی	مقدار
Name		flowLbl	
Text			
Backcolor		رنگ دلخواه	
Location	X	180	
	Y	9	

جدول ۱۸-۶- ویژگی های فرم

Form		ویژگی	مقدار
Name		Form1	
Text		جابه جایی برجسب	
Size	Width	452	
	Height	362	

۴- تنظیم ویژگی های دکمه ها : چهار کنترل دکمه را به فرم اضافه کرده و ویژگی های آنها را طبق جداول زیر تنظیم کنید.

جدول ۱۹-۶- ویژگی های دکمه left

Button		ویژگی	مقدار
Name		right	
Width		43	
Height		28	
Font		Wingdings	
Text		←	

جدول ۲۰-۶- ویژگی های دکمه right

Button		ویژگی	مقدار
Name		left	
Width		43	
Height		28	
Font		Wingdings	
Text		→	

جدول ۲۱-۶- ویژگی های دکمه down

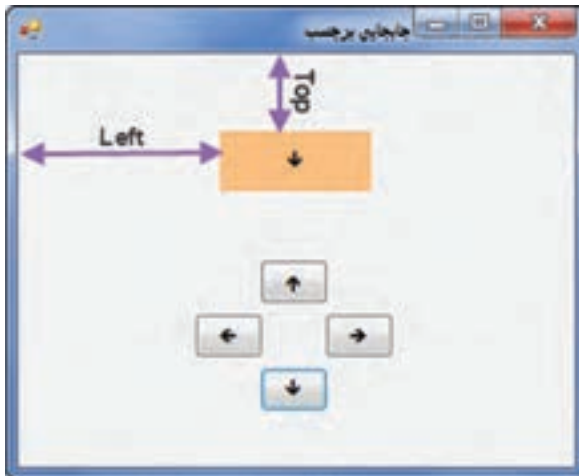
Button		ویژگی	مقدار
Name		down	
Width		43	
Height		28	
Font		Wingdings	
Text		↓	

جدول ۲۲-۶- ویژگی های دکمه up

Button		ویژگی	مقدار
Name		up	
Width		43	
Height		28	
Font		Wingdings	
Text		↑	

برای قرار دادن علامت جهت روی دکمه‌ها می‌توان از تصویر استفاده کرد و ویژگی BackgroundImage دکمه را تغییر داد و یا از نمادها و ویژگی Text دکمه استفاده کرد. در اینجا ما از روش دوم استفاده می‌کنیم. ویژگی Font دکمه‌ها را مساوی Wingdings قرار می‌دهیم. در این فونت نمادها، دارای کد اسکی ۲۳۱، ۲۳۲، ۲۳۳ و ۲۳۴ هستند. برای قرار دادن این نمادها در ویژگی Text هر دکمه، کلید Alt را نگه داشته و کد اسکی کلید را با کلیدهای ماشین حساب صفحه کلید وارد کنید.

**۵- نوشتن رویداد click برای دکمه‌ها :** در زمان طراحی فرم موقعیت هر کنترل نسبت به فرمی که روی آن قرار دارد با ویژگی Location که دارای دو مشخصه X و Y است تعیین می‌شود ولی برای تغییر موقعیت کنترل در زمان اجرا از دو ویژگی Left و Top که به ترتیب معادل X و Y هستند استفاده می‌کنیم. بنابراین مشخصه Left، فاصله کنترل از سمت چپ فرم و مشخصه Top، فاصله کنترل از بالای فرم است (شکل ۱۶-۶). بنابراین هر چه به مقدار مشخصه Left اضافه شود، فاصله کنترل از سمت چپ فرم بیشتر می‌شود و کنترل به سمت راست جابه‌جا می‌شود. هر چه از مقدار مشخصه Left کم شود، فاصله کنترل از سمت چپ فرم کمتر می‌شود و کنترل به سمت چپ جابه‌جا می‌شود. به همین ترتیب هر چه به مقدار مشخصه Top اضافه شود، فاصله کنترل از بالای فرم بیشتر می‌شود و کنترل به سمت پایین جابه‌جا می‌شود. هر چه از مقدار مشخصه Top کم شود، فاصله بالای کنترل از بالای فرم بیشتر می‌شود و کنترل به سمت بالا جابه‌جا می‌شود.



شکل ۱۶-۶- ویژگی Top و Left کنترل

روی هر یک از دکمه‌ها دابل کلیک کرده، تا الگوی رویداد کلیک آن نوشته شود، سپس رویداد click دکمه‌ها را به صورت زیر بنویسید :

```
private void right_Click(object sender, EventArgs e)
{
    flowLbl.Left += 10;
    flowLbl.Text = right.Text;
}
```

```
private void up_Click(object sender, EventArgs e)
{
    flowLbl.Top -= 10;
    flowLbl.Text = up.Text;
}
```

```
private void down_Click(object sender, EventArgs e)
{
    flowLbl.Top += 10;
    flowLbl.Text = down.Text;
}
```

```
private void left_Click(object sender, EventArgs e)
{
    flowLbl.Left -= 10;
    flowLbl.Text = left.Text;
}
```

برنامه ۴-۶

در کد نوشته شده، برچسب به هر جهتی حرکت کند آن جهت روی برچسب نشان داده می‌شود. به این منظور با کلیک هر دکمه مقدار ویژگی Text آن دکمه را در مشخصه Text برچسب قرار داده‌ایم.

**توسعه و بهبود برنامه :** با اجرای برنامه متوجه خواهید شد که با کد نوشته شده برچسب می‌تواند

از محدوده فرم خارج می‌شود. حداکثر مقداری که می‌توان در دو ویژگی Top و Left برچسب قرار داد تا از فرم خارج نشود چقدر است؟ کد برنامه را تغییر دهید تا با تغییر مقدار این دو مشخصه، برچسب از محدوده فرم خارج نشود.

## کار در کارگاه ۶: مدس تصویر

### مثال ۶-۶: کنترل جعبه متن<sup>۱</sup>

در این مثال با ایجاد فرمی مانند شکل ۶-۱۷ یک بازی می‌سازیم که در آن کاربر یک تصویر پنهان را حدس بزند! بازیکن برای اینکه بتواند تصویر مورد نظر را حدس بزند باید روی بخش‌هایی از تصویر کلیک کند تا آن بخش نمایش داده شود. هر زمان که کاربر تصویر را تشخیص داد، باید تشخیص خود را در جعبه متن بنویسد. در صورت درست بودن تشخیص کاربر، امتیاز او نمایش داده می‌شود.



شکل ۶-۱۷- فرم بازی حدس تصویر

**الگوریتم یا روش انجام کار:** برای اینکه عملیات نمایش قسمتی از تصویر را بتوانیم شبیه‌سازی کنیم در این برنامه از یک روش ساده اما کارا استفاده می‌کنیم. سطح تصویر را با چندین Label می‌پوشانیم. با کلیک هر برچسب، آن برچسب پنهان خواهد شد و قسمتی از تصویر که زیر آن قرار دارد نمایش داده می‌شود.

روش امتیاز دهی را نیز به این صورت پیاده‌سازی می‌کنیم که ۹ برچسب برای پنهان کردن

<sup>۱</sup> - Text Box



تصویر قرار می‌دهیم و به صورت زیر برای هر یک امتیازی مشخص می‌کنیم.

۵ Label1	۱۰ Label2	۵ Label3
۱۰ Label4	۱۰ Label5	۱۰ Label6
۵ Label7	۱۰ Label8	۵ Label9

شکل ۱۸-۶

دلیل آنکه به گوشه‌ها امتیاز کمتری داده‌ایم این است که معمولاً در گوشه‌های تصویر جزئیات کمتری برای شناخت آن وجود دارد. با پنهان شدن هر برجسب امتیاز آن از مجموع امتیاز (۷۰) کم می‌شود. با وارد کردن حدس داخل جعبه متن و زدن دکمه، حدس مورد نظر با جواب مقایسه می‌شود و پیام مناسب نمایش داده می‌شود.

۱- ایجاد پروژه: وارد برنامه VS شوید و پروژه جدیدی از نوع WindowsFormsApplication به نام GussePictureProj ایجاد نمایید.

۲- درج کنترل‌ها: در قسمت پایین فرم یک کنترل جعبه متن (TextBox)، یک دکمه و یک برجسب برای حدس و نشان دادن امتیاز کاربر قرار دهید. در بخش میانی یک جعبه تصویر (PictureBox) و ۹ برجسب برای پنهان کردن تصویر قرار دهید.

۳- تنظیم ویژگی‌های برجسب‌ها، جعبه متنی و دکمه: ویژگی‌های جعبه متن، دکمه و برجسب‌ها را طبق جداول زیر تنظیم کنید.

جدول ۲۳-۶ - ویژگی‌های جعبه متن

TextBox	
ویژگی	مقدار
Name	guess
Text	

جدول ۲۴-۶ - ویژگی های دکمه

Button	
ویژگی	مقدار
Name	check
Text	درست است؟

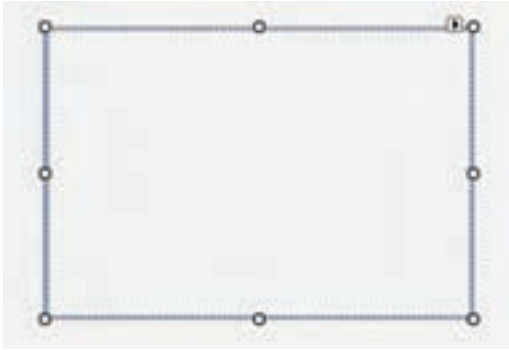
جدول ۲۵-۶ - ویژگی های برجسب امتیاز

Label (نمایش امتیاز)	
ویژگی	مقدار
Name	result
Text	حذف مقدار

جدول ۲۶-۶ - ویژگی های برجسب های پنهان کننده تصویر

Label2, Label3, ....., Label10 (برجسب هایی برای پنهان کردن تصویر)		
ویژگی	مقدار	توضیح
AutoSize	false	غیر فعال کردن اندازه خودکار
BackColor	#93aae1	تغییر رنگ پس زمینه
Text		متن داخل برجسب را خالی قرار می دهد.
Width	91	عرض کنترل را بر حسب پیکسل مشخص می کند.
Height	60	ارتفاع کنترل را بر حسب پیکسل مشخص می کند.

بعد از قراردادن یکی از برجسب هایی که برای پنهان کردن تصویر لازم است روی فرم در محل مناسب ، و مقداردهی ویژگی های آن براساس جدول ۲۶-۶، با نسخه برداری (copy) از آن، برجسب ها را به ۹ عدد می رسانیم. سپس این برجسب ها را به صورت دقیقی کنار هم قرار می دهیم تا در کنار هم، شکل مستطیلی را ایجاد کنند.



#### ۴- تنظیم ویژگی‌های جعبه

تصویر: یک کنترل pictureBox به فرم اضافه می‌کنیم. این عمل را توسط کشیدن ماوس روی فرم، طوری انجام می‌دهیم که اندازه آن برابر با اندازه مستطیل ساخته شده توسط برجسب‌ها باشد.

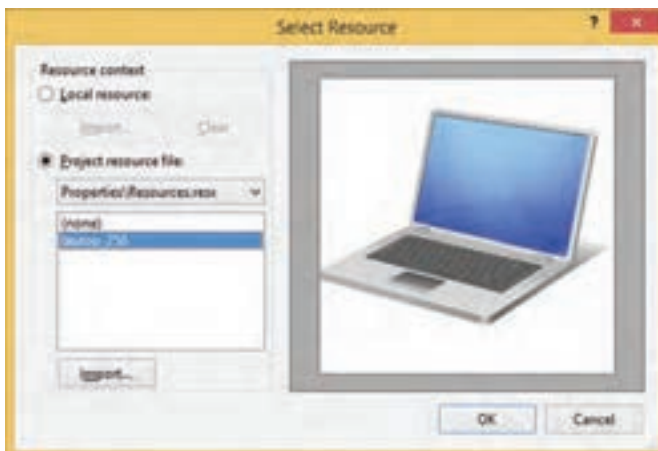
شکل ۱۸-۶ هم اندازه کردن کنترل کادر تصویر با مستطیل شامل برجسب‌ها

ویژگی‌های جعبه تصویر را مطابق جدول زیر تنظیم می‌نماییم:

جدول ۲۷-۶ ویژگی‌های جعبه تصویر

PictureBox	
ویژگی	مقدار
Name	Pic
SizeMode	StretchImage

همچنین توسط ویژگی Image، تصویر آن را مشخص می‌کنیم.



شکل ۱۹-۶ تنظیم ویژگی Image کنترل کادر تصویر

همان طور که ملاحظه می‌نمایید بعد از انتخاب عکس، به دلیل آنکه کنترل PictureBox بعد از Label ها ایجاد شده است روی آنها قرار می‌گیرد.



شکل ۲۰-۶- قرار گرفتن کادر تصویر روی برجسب‌ها

برای آنکه کنترل PictureBox پشت برجسب‌ها قرار گیرد، روی آن راست کلیک کرده، گزینه Send To Back را انتخاب کنید.



شکل ۲۱-۶- انتقال کنترل به پشت کنترل‌های دیگر

به این ترتیب تصویر توسط برجسب‌ها پنهان خواهد شد.

**۵- تعریف متغیر برای نگهداری امتیاز:** در کلاس Form روی فرم دابل کلیک کنید تا پنجره کد باز شود.

در کلاس فرم متغیر score را تعریف می‌کنیم تا بتوانیم در تمام رویدادهای فرم از آن استفاده کنیم.

```
public partial class Form1 : Form
{
    byte score = 70;
    ....
}
```

این متغیر را از نوع byte در نظر گرفته ایم و آن را برابر با ۷۰ قرار داده ایم. زیرا مجموع امتیازها ۷۰ است.

**سؤال:** چرا برای نگهداری امتیاز کاربر متغیری از نوع Byte تعریف کردیم؟

**۶-** نوشتن رویداد click برای برچسب‌های پنهان‌کننده تصویر: با کلیک روی برچسب، آن برچسب پنهان می‌شود. بنابراین در رویداد کلیک هر برچسب باید آن برچسب را پنهان کرد. همان‌طور که در کارگاه ۲ دیدید، پیدا و پنهان بودن یک کنترل به یک ویژگی کنترل بستگی دارد و آن ویژگی، Visible می‌باشد. اگر مقدار این ویژگی true باشد آن کنترل نمایش داده می‌شود و اگر مقدار آن false باشد، پنهان می‌شود.

برای ایجاد رویداد کلیک برچسب، همان‌طور که قبلاً گفته شد، می‌توانید روی آن دوبار کلیک کنید و یا از لیست رویدادها استفاده کنید. بعد از ایجاد کد رویداد برای برچسب label1 خواهیم داشت:

```
private void label1_Click(object sender, EventArgs e)
{
}
```

با کلیک روی این برچسب باید این کنترل پنهان شود و ۵ امتیاز از امتیاز کل کم شود؛ زیرا همان‌طور که در طرح امتیاز گفته شد، برچسب‌های گوشه ۵ امتیاز خواهند داشت.

```
private void label1_Click(object sender, EventArgs e)
{
    score -= 5;
    label1.Visible = false;
}
```

برای سایر برچسب‌هایی که تصویر را می‌پوشانند نیز همین عملیات را انجام می‌دهیم. اما توجه داشته باشید که در رویداد کلیک هر برچسب باید همان برچسب پنهان شود. همچنین برای برچسب‌هایی که در گوشه قرار نمی‌گیرند باید ۱۰ امتیاز کسر شود. برای برچسب‌های ۳، ۷ و ۹

مانند برچسب شماره ۱، ۵ امتیاز و برای سایر برچسب‌ها ۱۰ امتیاز کم می‌شود. به عنوان مثال برای label2 داریم:

```
private void label2_Click(object sender, EventArgs e)
{
    score -= 10;
    label2.Visible = false;
}
```

کدهای مربوط به برچسب‌ها در برنامه ۵\_۶ آورده شده است.

```
private void label1_Click(object sender, EventArgs e)
{
    score -= 5;
    label1.Visible = false;
}
private void label2_Click(object sender, EventArgs e)
{
    score -= 10;
    label2.Visible = false;
}
private void label3_Click(object sender, EventArgs e)
{
    score -= 5;
    label3.Visible = false;
}
private void label4_Click(object sender, EventArgs e)
{
    score -= 10;
    label4.Visible = false;
}
private void label5_Click(object sender, EventArgs e)
{
    score -= 10;
```

```

        label5.Visible = false;
    }
    private void label6_Click(object sender, EventArgs e)
    {
        score -= 10;
        label6.Visible = false;
    }
    private void label7_Click(object sender, EventArgs e)
    {
        score -= 5;
        label7.Visible = false;
    }
    private void label8_Click(object sender, EventArgs e)
    {
        score -= 10;
        label8.Visible = false;
    }
    private void label9_Click(object sender, EventArgs e)
    {
        score -= 5;
        label9.Visible = false;
    }
}

```

برنامه ۵-۶- کد مربوط به برجسب‌ها

**۷- نوشتن رویداد click برای دکمه:** کاربر پس از وارد کردن حدس خود در جعبه متن باید روی دکمه کلیک کند. برای بررسی اینکه کاربر درست است یا خیر، رویداد کلیک دکمه را می‌نویسیم. توسط شرط (if) بررسی می‌کنیم که آیا متن درون جعبه متنی (guess) برابر متن «لپ‌تاپ» می‌باشد یا خیر. اگر شرط درست بود پیام حدس درست به همراه امتیاز در برجسب result قرار می‌گیرد و در غیر این صورت پیام حدس اشتباه در برجسب result جای می‌گیرد. لازم به یادآوری است که متد ToString مقدار متغیر score را به متن تبدیل می‌کند.

```

private void check_Click(object sender, EventArgs e)
{
    if (guess.Text == «لپ تاپ»)
        result.Text = «آفرین درست حدس زدی.. امتیاز» + score.ToString();
    else
        result.Text = «حدست درست نبود!»;
}

```

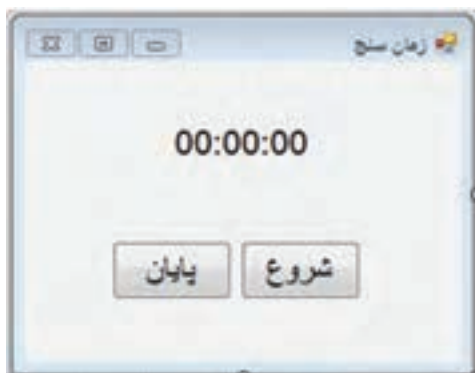
برنامه ۶-۶

**توسعه و بهبود برنامه:** به کدهایی که نوشته اید فکر کنید و ببینید آیا می‌توانید ایده‌های دیگری برای این بازی بدهید؟ به عنوان نمونه بیاندیشید چگونه می‌توان برای تعداد پاسخ‌های اشتباه محدودیت گذاشت. به عنوان مثال با دادن ۲ پاسخ اشتباه، پیام اتمام بازی داده شود.

## کار در کارگاه ۷: طراحی زمان‌سنج دیجیتال

### مثال ۷-۶: کنترل زمان‌سنج

در این مثال می‌خواهیم با طراحی فرم شکل ۶-۲۲ برنامه‌ای بنویسیم که کار زمان‌سنج را انجام دهد. الگوریتم یا روش انجام کار: این برنامه با کلیک دکمه شروع، زمان‌سنج را راه‌اندازی کرده و با کلیک دکمه پایان، زمان‌سنج را متوقف می‌کند. بنابراین به دو دکمه فرمان، یک کنترل برچسب و یک کنترل زمان‌سنج نیاز داریم.



شکل ۶-۲۲- فرم زمان‌سنج دیجیتالی



۱- ایجاد پروژه: وارد برنامه VS شوید و پروژه جدیدی از نوع WindowsFormsApplication به نام TimerProj ایجاد نمایید.

۲- درج کنترل‌ها: روی فرم دو کنترل دکمه و یک برچسب قرار دهید.

۳- تنظیم ویژگی‌های دکمه‌ها، برچسب و فرم: ویژگی‌های فرم و کنترل‌ها را طبق جدول زیر تنظیم کنید.

جدول ۶-۲۸ ویژگی‌های دکمه شروع    جدول ۶-۲۹ ویژگی‌های دکمه پایان    جدول ۶-۳۰ ویژگی‌های برچسب

Label	
ویژگی	مقدار
Name	timer Lbl
Text	00: 00: 00

Button	
ویژگی	مقدار
Name	stopBut
Text	پایان

Button	
ویژگی	مقدار
Name	startBut
Text	شروع

جدول ۶-۳۱ ویژگی‌های فرم

Form		
	ویژگی	مقدار
	Name	Form1
	Text	زمان سنج
	RightToLeft	yes
	RightToLeftLayout	True
Size	Width	292
	Height	228

جدول ۶-۳۲ ویژگی‌های زمان سنج

Timer	
ویژگی	مقدار
Name	Timer1
interval	1000
Enabled	false

۴- تنظیم ویژگی‌های زمان سنج: کنترل Timer را به فرم اضافه کرده و ویژگی‌های آن را طبق جدول ۶-۳۲ تنظیم کنید.

کنترل زمان سنج برای اطلاع از سپری شدن فاصله‌های زمانی یکسان به کار می‌رود. برای مثال اگر با رایانه کار می‌کنید، برای حفظ سلامتی لازم است هر نیم ساعت چند نرمش مناسب انجام دهید. ساعت را تنظیم می‌کنید که هر نیم ساعت زنگ بزند. زنگ ساعت به شما اطلاع می‌دهد که زمان نرمش رسیده است. کنترل زمان سنج این کار را برای برنامه انجام می‌دهد. فاصله زمانی در ویژگی Interval تایمر مشخص می‌شود. کنترل زمان سنج دارای رویداد Tick می‌باشد که مشابه زنگ ساعت است. این رویداد در فواصل زمانی یکسان که در ویژگی Interval کنترل مشخص می‌شود رخ می‌دهد. واحد زمانی ویژگی Interval میلی ثانیه است بنابراین اگر بخواهیم هرثانیه یکبار این رویداد رخ دهد، باید ویژگی Interval را مساوی ۱۰۰۰ قرار دهیم.

هرگاه کار شما با رایانه تمام شد، دیگر نیازی نیست که ساعت زنگ بزند؛ بنابراین زنگ ساعت را قطع می‌کنید. چگونه زنگ زمان سنج را قطع کنیم (رویداد Tick را متوقف کنیم)؟ کنترل‌ها دارای ویژگی Enabled هستند که از نوع Boolean است و هرگاه مقدار آن false شود، کنترل غیرفعال شده و به هیچ رویدادی واکنش نشان نمی‌دهد.

**۵- تعریف متغیرهایی برای نگهداری ساعت، ثانیه و دقیقه:** زمان سپری شده را به صورت ساعت، دقیقه و ثانیه باید نگهداری کنیم؛ بنابراین نیاز به متغیر داریم. متغیرها باید در تمام فرم قابل دسترسی باشند؛ بنابراین در کلاس Form آنها را تعریف کرده، با صفر مقداردهی می‌کنیم. روی فرم دابل کلیک کنید تا پنجره کد باز شود.

در کلاس فرم برای ثانیه، دقیقه و ساعت متغیرهایی با نام second، minute و hour و از نوع byte تعریف می‌کنیم.

```

Namespace TimerProject
{
    public partial class Form1 : Form
    {
        byte hour, second, minutes;
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

**۶-** نوشتن رویداد **click** برای دکمه‌ها : در رویداد **Click** دکمه شروع، کنترل زمان‌سنج را فعال می‌کنیم. برای این منظور ویژگی **Enabled** زمان‌سنج را با **true** مقداردهی می‌کنیم و متغیرهایی که مقدار ثانیه، دقیقه و ساعت را نگهداری می‌کنند، صفر می‌کنیم.

```
private void startBut_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
    hour = minute = second = 0;
}
```

برنامه ۸-۶

در رویداد **Click** دکمه پایان، کنترل زمان‌سنج را غیرفعال می‌کنیم.

```
private void stopBut_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
}
```

برنامه ۹-۶

**۷-** نوشتن رویداد **Tick** برای کنترل زمان‌سنج : از آنجا که فاصله زمانی رخ داده رویداد **Tick** را روی ۱ ثانیه ( $Interval=1000$ ) تنظیم کردیم، باید در رویداد **Tick** به متغیر ثانیه یکی اضافه کنیم و اگر مقدار آن به ۶۰ رسید آن را صفر کرده و به متغیر دقیقه یکی اضافه کنیم و اگر مقدار متغیر دقیقه به ۶۰ رسید مقدار متغیر دقیقه را صفر کرده و به متغیر ساعت یکی اضافه می‌کنیم. در انتها ویژگی **text** کنترل برچسب **timerLbl** را تغییر می‌دهیم که زمان را نشان دهد.

```
private void timer1_Tick(object sender, EventArgs e)
{
    second+=1;
    if(second==60)
    {
        second=0;
        minute +=1;
    }
}
```

```

if (minute == 60)
{
    minute = 0;
    hour += 1;
}
timerLbl.Text=string.Format("{0:00}: {1:00}: {2:00}",
hour,minutes,second);
}

```

برنامه ۱۰-۶

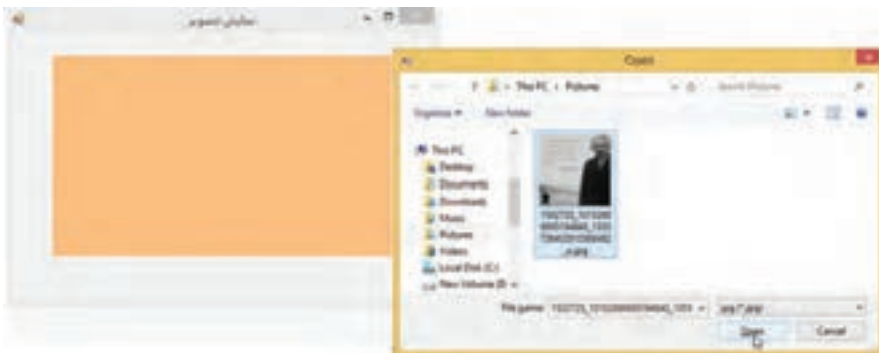
متد Format نوع داده string، رشته را به الگوی خاص تبدیل می‌کند. در اینجا می‌خواهیم ساعت به صورت 00:00:00 نمایش داده شود. بنابراین جای مقادیر ساعت، دقیقه و ثانیه را با پارامترهای {0}، {1} و {2} در رشته مشخص کرده و قرار دادن 00: در پارامتر ساعت، دقیقه و ثانیه سبب می‌شود که به صورت دو رقمی نمایش داده شوند.

**توسعه و بهبود برنامه:** در برنامه دکمه سومی برای توقف و ادامه (pause و resume) قرار دهید که در صورت کلیک آن زمان سنج به صورت موقت متوقف شود و با کلیک مجدد آن به کار خود ادامه دهد.

## کار در کارگاه ۸: تغییر عکس روی فرم در زمان اجرا

### مثال ۸-۶: کنترل کادر محاوره‌ای انتخاب فایل

در این مثال می‌خواهیم با ایجاد فرم شکل ۲۳-۶ برنامه‌ای بنویسیم که کاربر بتواند یک تصویر را به وسیله کادر محاوره‌ای، انتخاب کرده و نمایش دهد.



شکل ۲۳-۶ فرم انتخاب تصویر از رایانه

الگوریتم یا روش انجام کار : می‌خواهیم برنامه‌ای بنویسیم که تصویر انتخاب شده توسط کاربر روی فرم نمایش داده شود. این برنامه با کلیک جعبه تصویر، یک جعبه انتخاب فایل را نمایش می‌دهد و با انتخاب فایل تصویر از کادر انتخاب فایل، تصویر آن در جعبه تصویر نمایش داده می‌شود.

۱- ایجاد پروژه : وارد برنامه VS شوید و پروژه جدیدی از نوع `WindowsFormsApplication` به نام `ShowPictureProj` ایجاد نمایید.

۲- درج کنترل‌ها : روی فرم یک کنترل `PictureBox` و یک `OpenFileDialog` قرار دهید.

۳- تنظیم ویژگی‌های جعبه تصویر : ویژگی‌های فرم و جعبه تصویر را طبق جدول زیر تنظیم کنید.

جدول ۳۳-۶- ویژگی‌های جعبه تصویر

PictureBox	
ویژگی	مقدار
Name	pictureBox1
BackColor	#ffc080
SizeMode	StretchImage
Width	430
Height	253

۴- تنظیم ویژگی‌های کادر محاوره‌ای انتخاب فایل : برای باز کردن فایل جدید در برنامه‌هایی مانند `word` با انتخاب گزینه `open` از منوی `file` کادر محاوره‌ای `open` باز می‌شود و به وسیله آن می‌توان فایل موردنظر را انتخاب کرد. کنترل `OpenFileDialog` کادر محاوره‌ای `open` است. این کنترل پایین فرم قرار می‌گیرد و در زمان اجرا دیده نمی‌شود. در این مثال برای انتخاب فایل تصویر به این کنترل نیاز داریم. آن را به فرم اضافه کرده و ویژگی‌های آن را طبق جدول ۳۴-۶ تنظیم کنید.

## جدول ۳۴-۶- ویژگی های کادر محاوره ای OpenFileDialog

OpenFileDialog		
ویژگی	مقدار	توضیحات
Name	OpenFileDialog1	تعیین نام
Filter	Jpg File *.jpg jpeg files *.jpeg	تعیین فرمت فایل های قابل انتخاب
FileName		خالی کردن کادر نام فایل
Width	430	تعیین عرض
Height	253	تعیین کنترل

توجه کنید که می توانید توسط ویژگی Filter نوع فایل هایی که کادر محاوره ای نشان می دهد را مشخص کنید. مقداردهی این ویژگی به صورت زیر است :

.....|نوع ۲ | شرح نوع ۲ | نوع ۱ | شرح نوع ۱

مثال : Jpg Files|\*.jpg|jpeg Files|\*.jpeg

شکل ۲۴-۶

در این نمونه ما می خواهیم که یک تصویر را انتخاب کنیم، بنابراین نوع فایل ها را jpg و jpeg مشخص کرده ایم. در کادر محاوره ای OpenFileDialog لیست نوع فایل ها نشان داده شده به صورت شکل ۲۵-۶ است.



شکل ۲۵-۶- انتخاب نوع فایل های تعیین شده در ویژگی Filter

**۵-** نوشتن رویداد click برای جعبه تصویر : در رویداد کلیک جعبه تصویر، کادر محاوره ای انتخاب فایل را باز می کنیم.

```
private void pictureBox1_Click(object sender, EventArgs e)
{
    openFileDialog1.ShowDialog();
}
```

#### برنامه ۱۱-۶

کادر محاوره‌ای OpenFileDialog دارای متد ShowDialog برای باز کردن کادر محاوره‌ای است.

**۶-** نوشتن رویداد FileOk برای کنترل کادر محاوره‌ای انتخاب فایل : می‌خواهیم در صورت انتخاب فایل از کادر محاوره‌ای (دکمه open زده شد یا فایل دوبار کلیک شد) تصویر مورد نظر در pictureBox بارگذاری شود. با انتخاب فایل در کادر OpenFileDialog نام و مسیر فایل انتخاب شده در ویژگی FileName کنترل قرار می‌گیرد و رویداد FileOk برای کنترل رخ می‌دهد. برای نوشتن رویداد FileOk، روی کنترل OpenFileDialog در محیط طراحی دوبار کلیک کنید یا با زدن دکمه Eventes در پنجره Properties (با علامت رعد و برق) از لیست رویدادهای کنترل استفاده کنید.

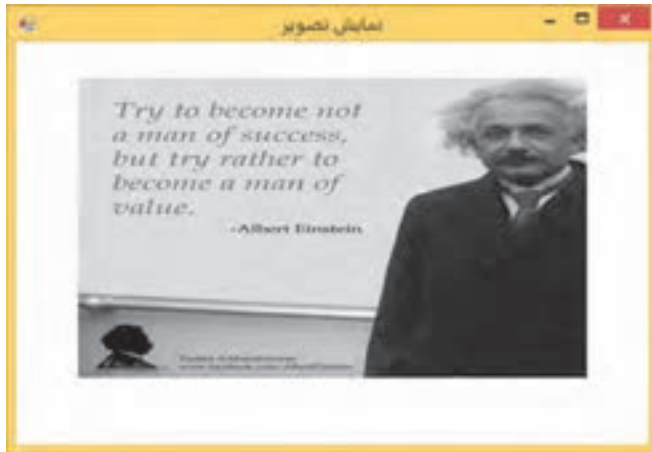
رویداد FileOk را مطابق زیر بنویسید.

```
private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
    pictureBox1.ImageLocation = openFileDialog1.FileName;
}
```

#### برنامه ۱۲-۶

در این رویداد تصویر pictureBox1 بارگذاری می‌شود. بارگذاری از طریق آدرسی است که در ویژگی FileName کنترل openFileDialog1 قرار دارد. این آدرس به ویژگی ImageLocation کادر تصویر داده می‌شود. ویژگی ImageLocation آدرس یک تصویر را می‌گیرد و آن را در کادر تصویر بارگذاری می‌نماید.

**توسعه و بهبود برنامه :** روی فرم دکمه‌ای با عنوان تغییر تصویر قرار دهید و به جای کلیک جعبه تصویر با کلیک دکمه، تصویر جعبه تصویر را تغییر دهید.



شکل ۲۶-۶- نمایش تصویر انتخاب شده در کادر تصویر

## کار در کارگاه ۹: کار با فونت و رنگ

### مثال ۹-۶: کنترل کادر محاوره‌ای رنگ<sup>۱</sup> و کادر محاوره‌ای فونت<sup>۲</sup>

در این مثال می‌خواهیم یک ویرایشگر متنی ساده بسازیم که در آن بتوان رنگ پس زمینه و تنظیم‌های قلم (فونت) را مشخص کرد (شکل ۲۷-۶).

برای تغییر فونت از کادر محاوره‌ای فونت و برای تغییر رنگ از کادر محاوره‌ای رنگ استفاده

می‌کنیم، بنابراین در این مثال به کنترل‌های ColorDialog و FontDialog نیاز داریم.

الگوریتم یا روش انجام کار: این برنامه دارای یک جعبه متن، دو دکمه یکی برای تغییر

فونت و دیگری برای تغییر رنگ قلم، یک ColorDialog و یک FontDialog است. با زدن دکمه



شکل ۲۷-۶- فرم ویرایشگر متنی ساده



قلم، کادر محاوره ای FontDialog باز می شود، فونت به وسیله کاربر انتخاب شده، روی جعبه متن اعمال می شود. با زدن دکمه «رنگ پس زمینه»، کادر محاوره ای ColorDialog باز می شود و رنگ به وسیله کاربر انتخاب شده، ویژگی رنگ پس زمینه جعبه متن تنظیم می شود.

۱- ایجاد پروژه: وارد برنامه VS شوید و پروژه جدیدی از نوع WindowsFormsApplication به نام EditorProj ایجاد نمایید.

۲- درج کنترل ها: روی فرم دو کنترل Button و یک TextBox قرار دهید.

۳- تنظیم ویژگی های دکمه ها و جعبه متن: ویژگی های جعبه متن و کنترل ها را طبق جداول زیر تنظیم کنید.

جدول ۳۷-۶- ویژگی های دکمه رنگ زمینه

Button	
ویژگی	مقدار
Name	colorBtn

جدول ۳۶-۶- ویژگی های دکمه قلم

Button	
ویژگی	مقدار
Name	fontBtn

جدول ۳۵-۶- ویژگی های جعبه متن

TextBox	
ویژگی	مقدار
Name	editor
Text	
RightToLeft	Yes
Multiline	True

در کنترل TextBox در حالت عادی تنها می توان یک خط نوشت. اما این کنترل دارای ویژگی MultiLine است که از نوع Boolean می باشد و در صورتی که مقدار آن مساوی true باشد، جعبه متن چند خطی ایجاد می کند.

۴- تنظیم ویژگی های کادرهای محاوره ای: دو کنترل FontDialog و ColorDialog را به فرم اضافه کرده و ویژگی آنها را طبق جداول زیر تنظیم کنید.

جدول ۳۸-۶- ویژگی های کادر محاوره ای رنگ

ColorDialog	
ویژگی	مقدار
Name	colorDialog1

جدول ۳۹-۶- ویژگی های کادر محاوره ای فونت

FontDialog	
ویژگی	مقدار
Name	fontDialog1
ShowColor	True

ویژگی ShowColor در کادر محاوره‌ای فونت امکان انتخاب رنگ قلم را نیز می‌سازد.  
**۵- نوشتن رویداد click برای دکمه‌ها:** در رویداد کلیک دکمه قلم، کادر محاوره‌ای تنظیمات قلم را نمایش می‌دهیم. برای این منظور متد ShowDialog() شیء fontDialog1 را فراخوانی می‌کنیم. سپس ویژگی‌های Font و ForeColor جعبه متن را با فونت و رنگ انتخاب شده در کادر محاوره‌ای فونت مقداردهی می‌کنیم.

```
private void fontBtn_Click(object sender, EventArgs e)
{
    fontDialog1.ShowDialog();
    editor.Font = fontDialog1.Font;
    editor.ForeColor = fontDialog1.Color;
}
```

برنامه ۶-۱۳



شکل ۶-۲۸- کادر محاوره‌ای فونت

در رویداد کلیک دکمه رنگ پس زمینه (colorBtn)، کادر محاوره‌ای انتخاب رنگ را نمایش می‌دهیم، بنابراین متد ShowDialog() شیء colorDialog1 را فراخوانی می‌کنیم، سپس رنگ پس‌زمینه جعبه متنی editor را با رنگ انتخاب شده مقداردهی می‌کنیم.

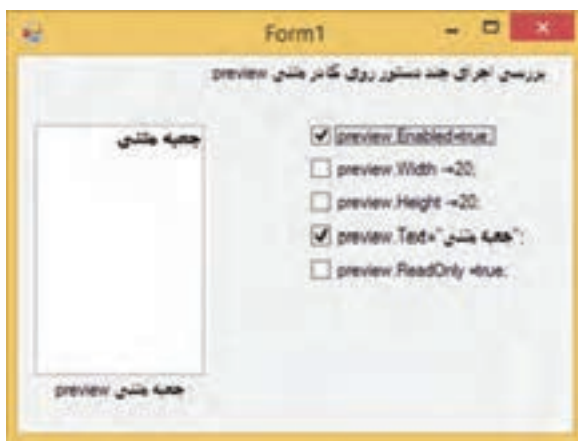
```
private void colorBtn_Click(object sender, EventArgs e)
{
    colorDialog1.ShowDialog();
    editor.BackColor = colorDialog1.Color;
}
```

برنامه ۶-۱۴

### مثال ۱۰-۶: کنترل کادر علامت

در این کارگاه می‌خواهیم با نوشتن برنامه‌ای به کاربر اجازه دهیم که تغییر برخی از ویژگی‌های کنترل جعبه متن را در زمان اجرا مشاهده کند. برای دادن امکان انتخاب ویژگی‌ها به کاربر از کنترل CheckBox استفاده می‌کنیم. این کنترل در مواردی که دو حالت داریم مثل true/false یا بله / خیر کاربرد دارد. در کنار این کنترل مربع کوچکی قرار دارد که در زمان اجرا با کلیک روی آن می‌توان علامت در آن قرار داد و یا علامت آن را حذف کرد.

برنامه‌ای بنویسید که حاصل اجرای چند دستور روی یک جعبه متنی را نمایش دهد. این دستورات به صورت CheckBox قابل انتخاب باشند و حاصل اجرای آنها هم‌زمان، نمایش داده شود (شکل ۶-۲۹).



شکل ۶-۲۹- فرم اجرای هم‌زمان چند دستور روی جعبه متنی

الگوریتم یا روش انجام کار: این فرم دارای پنج CheckBox، دو Label و یک TextBox است. با توجه به CheckBox‌هایی که دارای علامت هستند، گزینه دستوری مطابق با متن آنها، روی جعبه متنی اجرا می‌شود. اگر یک گزینه علامت‌دار باشد دستور آن اجرا می‌شود و در غیر این صورت عکس آن عمل اجرا می‌شود (چرا؟).

۱- ایجاد پروژه: وارد برنامه VS شوید و پروژه جدیدی از نوع WindowsFormsApplication

به نام CheckBoxProj ایجاد نمایید.

۲- درج کنترل‌ها: روی فرم پنج کنترل CheckBox، دو Label و یک TextBox قرار دهید.

۳- تنظیم ویژگی‌های کادرهای علامت، برچسب‌ها و جعبه متنی: ویژگی‌های کادرهای علامت، برچسب‌ها و جعبه متنی را طبق جداول زیر تنظیم کنید.

جدول ۴۱-۶- ویژگی‌های کادر علامت ۲

CheckBox1	
ویژگی	مقدار
Name	setEnabled
Text	preview.Enabled=true;
RightToLeft	No
Checked	True

جدول ۴۰-۶- ویژگی‌های کادر علامت ۱

checkBox2	
ویژگی	مقدار
Name	setWidth
Text	preview.Width -= 20;
RightToLeft	No
Checked	False

کنترل CheckBox دارای ویژگی Checked است که از نوع Boolean می‌باشد و اگر مقدار آن مساوی true باشد، کنترل دارای علامت است. برای آنکه با علامت‌دار شدن CheckBoxها عملیات مورد نظر روی جعبه متن انجام شود، در ویژگی Text کنترل‌ها دستور مورد نظر قرار داده‌ایم.

جدول ۴۳-۶- ویژگی‌های کادر علامت ۳

checkBox3	
ویژگی	مقدار
Name	setHeight
Text	preview.Height -= 20;
RightToLeft	No
Checked	False

جدول ۴۲-۶- ویژگی‌های کادر علامت ۴

checkBox4	
ویژگی	مقدار
Name	setReadOnly
Text	preview.ReadOnly= true;
RightToLeft	No
Checked	False

جدول ۴۴-۶- ویژگی های جعبه متن

TextBox 1	
ویژگی	مقدار
Name	preview
Text	جعبه متنی
RightToLeft	Yes
Multiline	True

جدول ۴۵-۶- ویژگی های کادر علامت ۵

checkBox5	
ویژگی	مقدار
Name	setText
Text	preview.Text = "جعبه متنی"
RightToLeft	No
Checked	True

دو برچسب روی فرم داریم که یکی عنوان پروژه را نشان می دهد و دیگری عنوانی برای جعبه متنی مشخص می کند. ویژگی این برچسب ها مطابق جداول زیر است.

جدول ۴۶-۶- ویژگی برچسب ۲

label 2	
ویژگی	مقدار
Name	label2
Text	جعبه متنی preview

جدول ۴۷-۶- ویژگی برچسب ۱

label 1	
ویژگی	مقدار
Name	label1
Text	بررسی اجرای چند دستور روی کادر متنی preview

**۴- نوشتن رویداد CheckedChanged برای کادرهای علامت:** برای آنکه با علامت دار شدن یک CheckBox عملیات مورد نظر روی جعبه متن انجام شود، در ویژگی Text کنترل ها دستور مورد نظر را قرار داده ایم، بنابراین کافی است همان دستور را در کد بنویسیم. در چه رویدادی؟ با کلیک کنترل CheckBox، کنترل علامت دار شده یا علامت آن حذف می شود و برای کنترل رویداد CheckedChanged رخ می دهد؛ بنابراین در هنگام رخ دادن این رویداد، ابتدا باید بررسی کنیم که اگر CheckBox مورد نظر انتخاب شده (علامت دار) دستور ذکر شده در متنش را انجام دهیم و در غیر این صورت باید عملی عکس آن انجام شود تا جعبه متنی به حالت قبلیش برگردد. پس از آنکه رویداد CheckedChanged را برای همه CheckBox ها ایجاد کردیم کد آنها را به صورت برنامه ۱۵-۶ تکمیل کنید.

```
private void setWidth_CheckedChanged(object sender, EventArgs e)
{
    if (setWidth.Checked)
        preview.Width -= 20;
    else
        preview.Width += 20;
}
```

برنامه ۱۵-۶

در این رویداد با کمک ویژگی Checked کنترل، بررسی می‌شود که کادر علامت setWidth انتخاب شده است یا خیر؟ اگر جواب درست بود از عرض جعبه متنی ۲۰ پیکسل کم می‌شود و در غیراین صورت عکس این عمل انجام می‌شود. یعنی ۲۰ واحد به عرض جعبه متنی اضافه می‌گردد. (ویژگی Width پهنای کنترل و ویژگی Height ارتفاع کنترل‌ها را برحسب پیکسل مشخص می‌کند.)

```
private void setHeight_CheckedChanged(object sender, EventArgs e)
{
    if (setHeight.Checked)
        preview.Height -= 20;
    else
        preview.Height += 20;
}
```

برنامه ۱۶-۶

در این رویداد با کمک ویژگی Checked کنترل، بررسی می‌شود که کادر علامت setHeight انتخاب شده است یا خیر؟ اگر جواب درست بود از ارتفاع جعبه متنی ۲۰ پیکسل کم می‌شود و در غیر این صورت عکس این عمل انجام می‌شود. یعنی ۲۰ واحد به ارتفاع جعبه متنی اضافه می‌گردد.

```
private void setEnabled_CheckedChanged(object sender, EventArgs e)
{
    preview.Enabled = setEnabled.Checked;
}
```

برنامه ۱۷-۶

ویژگی Enabled کنترل جعبه متن مانند ویژگی Checked کنترل کادر علامت از نوع Boolean است؛ بنابراین در صورت انتخاب کنترل `setEnabled` باید مقدار `Enabled` مساوی `true` و در غیر این صورت `false` شود، پس در این برنامه مقدار این دو ویژگی مساوی یکدیگر است.

```
private void setText_CheckedChanged(object sender, EventArgs e)
{
    if (setText.Checked)
        preview.Text = "جعبه متنی";
    else
        preview.Text = "";
}
```

برنامه ۶-۱۸

در این رویداد نیز با بررسی وضعیت انتخاب کادر علامت `setText`، متن جعبه متنی را برابر با «جعبه متنی» می‌کند یا آن را خالی می‌نماید.

ویژگی `ReadOnly` کنترل جعبه متن مانند ویژگی `Enabled` از نوع Boolean است و اگر مساوی `true` باشد جعبه متن برای کاربر مانند کنترل برچسب می‌شود و کاربر نمی‌تواند چیزی در آن بنویسد.

```
private void setReadOnly_CheckedChanged(object sender, EventArgs e)
{
    preview.ReadOnly = setReadOnly.Checked;
}
```

برنامه ۶-۱۹

## خودآزمایی فصل ششم

۱- هر ویژگی از سمت راست را با توضیح سمت چپ مطابقت دهید(یک توضیح اضافی است).	
۱- تعیین نوع و اندازه قلم متن یک شیء	<b>Visible</b>
۲- رنگ متن روی شیء	<b>Text</b>
۳- متن نمایش داده شده روی شیء	<b>Name</b>
۴- موقعیت شیء نسبت به گوشه بالا و سمت چپ فرم	<b>RightToLeft</b>
۵- تثبیت فاصله لبه‌های شیء از لبه‌های فرم	<b>Font</b>
۶- پشتیبانی شیء از زبان‌های راست به چپ	<b>Anchor</b>
۷- پنهان کردن شیء	<b>ForeColor</b>
۸- نام شیء برای دسترسی به ویژگی‌ها و متدهایش	<b>location</b>
۹- فعال کردن شیء	

۲- در پنجره Properties، چطور متوجه می‌شویم که کدام شیء یا کنترل در روی فرم، انتخاب شده است؟

۳- لیستی از کنترل‌هایی که تاکنون با آنها آشنا شدید را روی یک طرف کاغذی به ابعاد ۱۰×۵ سانتی‌متر بنویسید و در طرف دیگر کاربرد آن را ذکر کنید. سپس فیش‌های نوشته شده را با هم کلاسی خود به اشتراک بگذارید.

۴- فیش نویسی سؤال ۳ را برای کنترل‌ها و ویژگی‌های آن انجام دهید. در یک طرف کاغذ نام کنترل و طرف دیگر، سه تا چهار مورد ویژگی‌هایی که کاربردی هستند را بنویسید. می‌توانید فیش‌ها را در کلاس به کمک معلم خود، در یک کارگروهی و مسابقه تمرین کنید.



## تمرینات برنامه نویسی فصل ششم

۱- یک فرم مطابق با یک صفحه کلید بدون واکنش به رویدادها طراحی کنید.



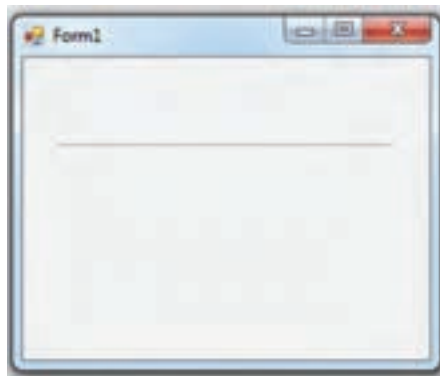
۲- یک فرم مطابق با کنترل تلویزیون بدون واکنش به رویدادهای کلیک ماوس بسازید. احتمالاً این کنترل باطری ندارد!



۳- یک فرم شامل چهار تصویر که هر یک مربوط به یک فصل باشد، بسازید. در زیر هر یک از تصاویر، یک برجسب قرار داده و نام فصل مربوطه را بنویسید. طراحی صفحه را با دقت و با رعایت فاصله یکسان و مناسب تصاویر با یکدیگر و همچنین از لبه‌های فرم انجام دهید. از ابزارهای منوی

- Format و یا در نوار ابزار، برای تنظیم موقعیت تصاویر و برچسب‌ها استفاده کنید.
- ۴- شبیه تمرین قبل را با موضوع تیم‌های ورزشی مورد علاقه خود بسازید. حداقل دو تیم ورزشی را در فرم معرفی کنید و لوگو تیم نشان داده شود.
- ۵- یک فرم شامل یک تصویر، یک جعبه متن و سه کادر علامت برای معرفی یکی از جاذبه‌های گردشگری شهر خود طراحی کنید. توضیحات آن جاذبه را در جعبه متن قرار دهید. سپس برنامه‌ای بنویسید که کاربر بتواند از طریق انتخاب یا عدم انتخاب کادرهای علامت، توضیحات را به زبان انگلیسی یا فارسی نمایش دهد و رنگ قلم و زمینه جعبه متن را تغییر دهد.
- ۶- تمرین زیر که به زبان انگلیسی ارائه شده است را بخوانید و سپس انجام دهید.

This example shows how to create a bevel line in Windows Forms. This line can be used as a visual separator of controls on a form.



To simulate the line in Windows Forms use a Label control. Set its Height to 2 pixels and BorderStyle to Fixed3D. That's all.

Now, create a new Windows Forms Application project and add a Label to the Form. Set its properties as shown in below table. Now, run the program.

Label Control	
Property	Value
AutoSize	False
BorderStyle	Fixed3D
Height	2
Width	250

The **OpenFileDialog** component allows users to browse the folders of their computer or any computer on the network and select one or more files to open. The dialog box returns the path and name of the file the user selected in the dialog box.

Use the **ShowDialog** method to display the dialog box and the OpenFile method to open the file.

## واژگان و اصطلاحات انگلیسی فصل ششم

ردیف	واژه انگلیسی	معنی واژه به فارسی
۱	Anchor	
۲	Button	
۳	Button Control	
۴	CheckBox	
۵	Double Click	
۶	Interval	
۷	ReadOnly	
۸	Tulip	



## تبدیل ضمنی انواع داده به یکدیگر در C#

گاهی هنگام استفاده از متغیرها و مقداردهی به آنها، مقداری که به متغیر منتسب می‌کنیم با نوع متغیر یکسان نیست اما سی شارپ پیام خطا نمی‌دهد. چرا؟  
 زیرا سی شارپ در صورتی که امکانش وجود داشته باشد به صورت ضمنی، نوع داده آن مقدار را به نوع داده متغیر تبدیل می‌کند. در جدول زیر مشخص شده که سی شارپ هر نوع داده را به کدام یک از انواع داده می‌تواند تبدیل کند.

ردیف	نوع داده	انواع داده قابل تبدیل
۱	<u>sbyte</u>	short, int, long, float, double, or decimal
۲	<u>byte</u>	short, ushort, int, uint, long, ulong, float, double, or decimal
۳	<u>short</u>	int, long, float, double, or decimal
۴	<u>ushort</u>	int, uint, long, ulong, float, double, or decimal
۵	<u>int</u>	long, float, double, or decimal
۶	<u>uint</u>	long, ulong, float, double, or decimal
۷	<u>long</u>	float, double, or decimal
۸	<u>char</u>	ushort, int, uint, long, ulong, float, double, or decimal
۹	<u>float</u>	double
۱۰	<u>ulong</u>	float, double, or decimal

## تبدیل صریح (casting) انواع داده به یکدیگر در C#

گاهی هنگام انتساب یک عبارت یا مقدار به یک متغیر سی شارپ خطایی مبنی بر اینکه نمی تواند نوع داده مقدار یا عبارت را به نوع داده متغیر تبدیل کند می دهد. برای حل این مشکل می توان از تبدیل صریح نوع داده استفاده کرد. در جدول زیر مشخص شده که با روش تبدیل صریح هر نوع داده را به کدام یک از انواع دیگر داده می توان تبدیل کرد.

جدول تبدیل صریح (casting) انواع داده به یکدیگر

ردیف	نوع داده	انواع داده قابل تبدیل
1	sbyte	byte , ushort, uint, ulong, or char
2	byte	sbyte or char
3	short	sbyte , byte, ushort, uint, ulong, or char
4	ushort	sbyte , byte, short, or char
5	int	sbyte , byte, short, ushort, uint, ulong, or char
6	uint	sbyte , byte, short, ushort, int, or char
7	long	sbyte , byte, short, ushort, int, uint, ulong, or char
8	char	sbyte , byte, or short
9	float	sbyte , byte, short, ushort, int, uint, long, ulong, char, or decimal
10	ulong	sbyte , byte, short, ushort, int, uint, long, or char
	double	sbyte , byte, short, ushort, int, uint, long, ulong, char, float, or decimal
	decimal	sbyte , byte, short, ushort, int, uint, long, ulong, char, float, or double

### الگوی نمایش اطلاعات

در جدول زیر لیستی از تعیین کننده‌های فرمت (format specifiers) که در الگوی نمایش {تعیین کننده فرمت : عدد} یا الگوی نمایش {تعیین کننده فرمت : عدد تراز، عدد} برای قالب بندی اعداد به کار می‌روند، آمده است :

Format	Specifier
مقدار پولی خاص یک محل	C
انواع صحیح (integer)	D
نماد علمی	E
نقطه اعشار ثابت	F
اعداد عمومی	G
نقطه اعشار ثابت با جدا کننده کاما	N
اعداد دارای درصد	P
هگزادسیمال	X

تأثیر دقت یک قالب خاص بستگی به تنظیمات منطقه‌ای دارد. به عنوان مثال قالب پولی C، به طور خودکار قالب پولی منطقه انتخاب شده را نشان می‌دهد. برای اکثر کاربران این اطلاعات به طور پیش فرض مربوط به منطقه و زبان آنها می‌باشد.

### قالب بندی سفارشی اعداد

سی‌شارپ به شما اجازه قالب بندی اعداد به صورت سفارشی را نیز می‌دهد. برای این کار کاراکترهای خاصی در اختیار شما می‌گذارد که در جدول صفحه بعد لیست آنها را مشاهده می‌کنید :

معنی	کاراکتر
عدد	#
اعشار	.
جدا کننده سه رقمی	,
درصد	%
اضافه کردن صفر	0
اعداد مثبت، منفی و مقادیر صفر را با قالب‌های خاصی نمایش می‌دهد.	;
نماد علمی	E0 E+0 E - 0 e0 e+0 e - 0

نقطه مکان اعشار را مشخص می‌کند. علامت # می‌تواند در برگیرنده عددی باشد که قرار است در سمت چپ یا راست آن اعشار قرار بگیرد. اگر این علامت در سمت راست اعشار قرار بگیرد دقت اعشار را مشخص می‌کند و ممکن است در صورت لزوم عدد را رند کند و اگر در سمت چپ ممیز باشد نماینده قسمت صحیح عدد می‌باشد. اگر طول عدد از تعداد علامت‌های # که در سمت چپ علامت اعشار قرار دارند بیشتر باشد، کل عدد نشان داده می‌شود.

کاراکتر E و e برای نمایش اعداد به صورت نماد علمی به کار می‌روند. حداقل یک صفر و در صورت لزوم تعداد بیشتری صفر بعد از E و e می‌آیند. صفرها تعداد ارقام اعشار را نشان می‌دهند. استفاده از حرف E و یا e ممکن است باعث چاپ این حروف در خروجی شوند. در نتیجه برای تعریف توان مثبت و منفی می‌توان از علامت - و + بعد از این دو کاراکتر به صورت E+, E-، e+، e- استفاده کرد. علامت ";" هم شما را قادر می‌سازد که اعداد مثبت، منفی و مقادیر صفر را با قالب‌های خاصی جدا سازید.



## پیوست ۳

### متدهای کلاس String

نوع برگشتی	شرح کار متد	متد
string	ایجاد یک کپی از رشته	Clone()
bool	مقایسه دو رشته	CompareTo()
bool	جستجوی یک کاراکتر یا یک رشته	Contains()
bool	آیا رشته به کاراکتر مورد نظر ختم می‌شود؟	EndsWith()
bool	مقایسه دو رشته	Equals()
int	موقعیت وجود یک کاراکتر در رشته را بر می‌گرداند.	IndexOf()
string	تبدیل تمام کاراکترهای یک رشته به حروف کوچک	ToLower()
string	تبدیل تمام کاراکترهای یک رشته به حروف بزرگ	ToUpper()
string	درج یک کاراکتر یا یک رشته در درون یک رشته دیگر	Insert()
int	موقعیت آخرین تکرار یک کاراکتر در درون یک رشته	LastIndexOf()
int	تعداد کاراکترهای یک رشته را بر می‌گرداند.	Length
string	حذف کاراکترهای یک رشته تا یک مکان مشخص	Remove()
string	یک کاراکتر در یک رشته را با کاراکتر یا یک رشته دیگری عوض می‌کند.	Replace()
String[]	یک رشته را به چند قسمت تقسیم می‌کند.	Split()
bool	آیا رشته با یک کاراکتر خاص شروع می‌شود؟	StartsWith()
string	یک قسمت از یک رشته را جدا می‌کند و در اختیار قرار می‌دهد.	Substring()
string	فاصله و کاراکترهای Tab را از ابتدا و انتهای رشته حذف می‌کند.	Trim()