
گنجینه سوالات C#

(حل مسائل – مرجع کامل)

تالیف:

دکتر رمضان عباس نژادورزی
دکتر هادی صالحی



فن آوری نوین

سرشناسه : عباس نژاد ورزی، رمضان، ۱۳۴۸ -
 عنوان و نام پدید آور : گنجینه سوالات C# (حل مسایل - مرجع کامل) / تالیف رمضان عباس نژاد ورزی، هادی صالحی
 مشخصات نشر : بابل: فناوری نوین، ۱۳۹۷
 مشخصات ظاهری : ۳۶۸ ص.: مصور.
 شابک : ۳۹۰۰۰۰ ریال: ۰ - ۱۹ - ۷۲۷۲ - ۶۰۰ - ۹۷۸
 وضعیت فهرست نویسی : فیبا
 موضوع : راهنمای آموزشی (عالی) سی شارپ (زبان برنامه نویسی کامپیوتر) --
 موضوع : C # (Computer program language) -- Study and teaching (Higher)
 موضوع : سی شارپ (زبان برنامه نویسی کامپیوتر) -- مسائل، تمرین ها و غیره
 موضوع : language -- (Problems, exercises, etc
 شناس افزوده : ۱۳۶۳ - صالحی، هادی،
 رده بندی کنگره : ۱۳۹۷ ع ۲۳ / ۹۵ س / ۷۶ / ۷۳ QA
 رده بندی دیویی : ۱۳۳ / ۰۰۵
 شماره کتابشناسی ملی : ۵۲۰۷۲۶۲



www.fanavarinovin.net

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کد پستی ۷۳۴۴۸-۴۷۱۶۷

فن آوری نوین

گنجینه سوالات C#

تألیف: دکتر رمضان عباس نژاد ورزی، دکتر هادی صالحی

ناشر: فن آوری نوین

چاپ اول: تابستان ۱۳۹۷

تیراژ: ۱۰۰۰ جلد

شابک: ۰ - ۱۹ - ۷۲۷۲ - ۶۰۰ - ۹۷۸

حروفچینی و صفحه آرایی: فن آوری نوین

قیمت: ۳۹۰۰۰ تومان

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

فهرست مطالب

۵	فصل اول: آشنایی با زبان C#
۳۳	فصل دوم: ساختار تصمیم و حلقه تکرار
۹۴	فصل سوم: متدها و پیاده سازی آن
۱۲۰	فصل چهارم: آرایه‌ها و رشته‌ها
۱۸۳	فصل پنجم: کلاس‌ها و اشیا
۲۵۴	فصل ششم: مباحث پیشرفته در C#
۳۴۰	فصل هفتم: گرافیک در زبان C#
۳۶۸	منابع:

مقدمه

زبان C# (سی شارپ) یکی از محبوب‌ترین زبان‌های برنامه‌نویسی شی‌گرا در ایران است. این زبان در دانشگاه‌ها نیز به عنوان یکی از مهم‌ترین زبان‌های رشته‌های کامپیوتر، فناوری اطلاعات، ICT علوم کامپیوتر و رشته‌های دیگر تدریس می‌شود. علاوه بر این، اکثر برنامه‌های تجاری در ایران با زبان C# پیاده‌سازی شده و توسعه می‌یابند. به همین دلیل، یادگیری این زبان برای دانشجویان رشته‌های کامپیوتر، فناوری اطلاعات، ICT، علوم کامپیوتر و مهندسی کامپیوتر یک امر ضروری است. یکی از روش‌های موفق یادگیری هر زبان برنامه‌نویسی، حل مسائل متنوع و آشنایی با الگوریتم‌های مختلف است.

در این کتاب سعی شده است با ارائه مثال‌های ساده، روان، هدف‌دار و متنوع، مفاهیم اساسی برنامه‌نویسی C# را به صورت گام به گام آموزش دهد. به علاوه کتاب شامل مسائل متنوع و حل آن‌ها می‌باشد که در پروژه‌های بزرگ و پیچیده می‌توانید از آن‌ها استفاده کنید.

در زبان C# مباحثی مختلفی را می‌توان آموزش داد. این کتاب فقط بحث Form Application را آموزش می‌دهد. چنانچه بخواهید Console Application را بی‌آموزید می‌توانید به کتاب‌های سیستم‌های شی‌گرا با زبان C#، برنامه‌نویسی پیشرفته با زبان C# و حل ۶۰۰ مسئله C# از همین انتشارات مراجعه کنید. اما، چنانچه بخواهید با بانک اطلاعاتی (تکنولوژی ADO.NET) کار کنید می‌توانید به کتاب آموزش گام به گام بانک اطلاعاتی با زبان C# تالیف رمضان عباس نژادورزی مراجعه نمایید. همچنین اگر نیاز دارید تکنولوژی LINQ را بی‌آموزید، می‌توانید به کتاب آموزش گام به گام LINQ با زبان C# تالیف رمضان عباس نژادورزی مراجعه کنید.

این کتاب شامل یک پیوست ۲۰۰ صفحه‌ای می‌باشد که به صورت الکترونیکی در کتاب الکترونیکی ارائه شده است.

امیدوارم این کتاب مورد استقبال اساتید و دانشجویان رشته‌های مختلف که زبان برنامه‌نویسی C# را مطالعه می‌کنند، قرار گیرد.

کتاب حاوی برنامه‌های است که کد آن‌ها را می‌توانید به صورت رایگان از سایت انتشارات فناوری نوین به آدرس www.fanavarienovin.net بگیرید.

مولفین

fanavarienovin@gmail.com

آشنایی با زبان C#

مثال ۱ - ۱. برنامه‌ای که سه کنترل Button را به فرم اضافه کرده با کلیک دکمه button1، اندازه فرم حداکثر می‌گردد. با کلیک دکمه button2، اندازه فرم به حالت نرمال برمی‌گردد و با کلیک دکمه button3، اندازه فرم کمینه می‌شود. چنانچه مکان نمای ماوس از محدوده فرم خارج شود، رنگ زمینه فرم آبی می‌گردد و اگر مکان نما به محدوده فرم برگردد (وارد شود)، رنگ زمینه فرم قرمز می‌شود.

مراحل طراحی و اجرا:

۱. پروژه جدیدی به نام Events ایجاد کنید.

۲. سه کنترل Button به فرم اضافه کنید.



۳. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Form1_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Maximize";
    button2.Text = "Normal";
    button3.Text = "Minimize";
}
```

۴. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Maximized;
}
```

این دستورات، خاصیت FormWindowState را فرم را FormWindowState.Maximized تعیین

می‌کنند تا با کلیک این دکمه اندازه فرم بیشینه گردد.

۵. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Normal;
}
```

این دستورات، با تغییر خاصیت FormWindowState به مقدار FormWindowState.Normal اندازه فرم را به اندازه معمولی (نرمال) آن تغییر می دهند.

۶. دکمه button3 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button3_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Minimized;
}
```

این دستورات اندازه فرم را کمینه می کنند.

۷. بر روی فرم کلیک کنید تا انتخاب شود. اکنون، کلید F4 را بزنید تا پنجره Properties ظاهر شود. در این پنجره دکمه Events را کلیک کنید تا لیست رویدادهای فرم را ببینید. رویداد MouseLeave را پیدا کرده، نوار جلوی آن را کلیک مضاعف نمایید و دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_MouseLeave(object sender, EventArgs e)
{
    this.BackColor = Color.Blue;
}
```

این دستورات، با خروج مکان‌نمای ماوس از محدوده فرم یا انتقال بر روی کنترل‌های روی آن، رنگ زمینه فرم را آبی می کنند.

۸. همانند مرحله ۷، رویداد MouseEnter را برای فرم اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_MouseEnter(object sender, EventArgs e)
{
    this.BackColor = Color.Red;
}
```

این دستورات، با ورود مکان‌نمای ماوس در محدوده فرم رنگ فرم را به قرمز تغییر می دهند.

۹. پروژه را ذخیره و اجرا کنید. اکنون دکمه maximize را کلیک کرده تا اندازه فرم حداکثر گردد. دکمه Normal را کلیک نمایید تا فرم به اندازه واقعی‌اش برگردد و همان‌طور که مکان‌نما را به خارج یا داخل فرم می برید، به تغییر رنگ زمینه فرم دقت کنید.

مثال ۲-۱. برنامه‌ای که اعمال زیر را انجام می دهد:

❖ عبارت " فرم اصلی " را در سمت راست عنوان فرم نمایش می دهد.
❖ با کلیک روی فرم، عنوان فرم به سمت چپ برده و با کلیک دیگر عنوان به سمت راست برود و این روند ادامه یابد.

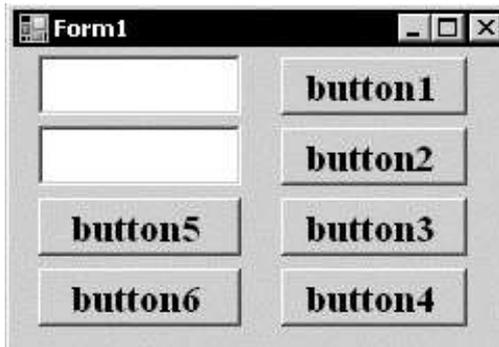
- ❖ دارای دو کنترل TextBox و شش کنترل Button باشد.
- ❖ با کلیک button1، اطلاعات textBox2 به textBox1 اضافه می شود.
- ❖ با کلیک دکمه button2، اطلاعات textBox1 به textBox2 اضافه می گردد.
- ❖ با کلیک دکمه button3، اطلاعات انتخاب شده کنترل textBox2 حذف می گردد.
- ❖ با کلیک دکمه button4، رنگ زمینه فرم به سبز تغییر یابد.
- ❖ با کلیک دکمه button5، رنگ زمینه فرم به خاکستری تبدیل شود.

❖ با کلیک دکمه button6، یک تصویر در زمینه فرم قرار می‌گیرد.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Event1 ایجاد کنید.

۲. دو کنترل TextBox و شش کنترل Button به فرم اضافه کنید.



۳. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Form1_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.RightToLeft = RightToLeft.Yes;
    this.Text = "فرم اصلی";
}
```

این دستورات، جهت نمایش اطلاعات روی فرم را از راست به چپ کرده، عنوان فرم را به فرم اصلی تغییر

می‌دهند.

۴. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text += textBox2.Text;
}
```

این دستورات، اطلاعات کنترل textBox2 را به انتهای کنترل textBox1 اضافه می‌کنند.

۵. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    textBox2.Text = textBox2.Text + textBox1.Text;
}
```

این دستورات، اطلاعات کنترل textBox1 را به انتهای کنترل textBox2 اضافه می‌کنند.

۶. دکمه button3 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button3_Click(object sender, EventArgs e)
{
    textBox2.SelectedText = "";
}
```

این دستورات، اطلاعات انتخاب شده کنترل textBox2 (مقدار خاصیت SelectedText) را پاک می‌کنند.

۷. دکمه button4 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button4_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Green;
}
```

این دستورات، رنگ زمینه فرم را به سبز تغییر می دهند.

۸. دکمه button5 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button5_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Gray;
}
```

این دستورات، رنگ زمینه فرم را به خاکستری تغییر می دهند.

۹. دکمه button6 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button6_Click(object sender, EventArgs e)
{
    this.BackgroundImage = Image.FromFile("c:\\windows\\Gone
    Fishing.bmp");
}
```

این دستورات، زمینه فرم را با یک تصویر پر می کنند. برای این منظور از متد FromFile کلاس Image

استفاده می کند.

۱۰. رویداد Click را برای فرم اضافه نمایید. برای این منظور، مراحل زیر را انجام دهید:

- ❖ ناحیه خالی فرم را کلیک کنید تا انتخاب شود.
- ❖ کلید F4 را فشار دهید تا پنجره Properties فرم ظاهر شود.
- ❖ در این پنجره دکمه Events را کلیک کنید تا رویدادهای فرم را مشاهده نمایید.
- ❖ رویداد Click را پیدا کرده نوار جلوی آن را کلیک مضاعف کنید و دستورات آن را به صورت زیر تغییر

دهید:

```
private void Form1_Click(object sender, EventArgs e)
{
    this.RightToLeft = (this.RightToLeft == RightToLeft.Yes) ?
    RightToLeft.No : RightToLeft.Yes;
}
```

این دستورات، با هر کلیک روی فرم جهت نمایش اطلاعات فرم را تغییر می دهند.

۱۱. برنامه را ذخیره و اجرا کنید. در textBox1 مقدار Value1 و در textBox2 مقدار Value2 را تایپ

کرده. دکمه button1 را کلیک کنید تا خروجی زیر را ببینید:

آشنایی با زبان C# ۹



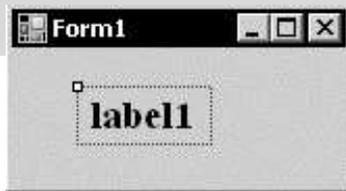
در `textBox2` مقدار `alu` را انتخاب کرده و به ترتیب دکمه‌های `button3` و `button6` را کلیک کنید و سپس بر روی فرم کلیک نمایید تا خروجی زیر ظاهر شود:



همان‌طور که در این خروجی می‌بینید، مقدار `alu` در `textBox2` حذف گردید، در زمینهٔ فرم یک تصویر قرار گرفت و جهت نمایش اطلاعات روی فرم از چپ به راست تغییر یافت.

مثال ۳ - ۱. برنامه‌ای که اعمال زیر را انجام می‌دهد:

۱. یک کنترل `Label` به فرم اضافه می‌کند.
۲. اگر فرم کلیک گردد، عبارت `Form Clicked` بر روی `label1` و عنوان فرم نوشته شود.
۳. اگر مکان نما فرم را ترک کند، رنگ زمینه `label` سفید و رنگ زمینه فرم آبی شود.
۴. اگر مکان نما وارد `label` شود، رنگ زمینه فرم قرمز و رنگ نوشته `dabel` سفید شود.
۵. اگر بر روی فرم کلیدی فشرده شود، بر روی عنوان فرم و `label` عبارت `Key Pressed` نوشته شود.
۶. اگر اندازه فرم تغییر یابد، در عنوان فرم و `label1` عبارت `Resizing` با رنگ سبز نوشته شود.
۷. وقتی مکان نما بر روی فرم حرکت می‌کند رنگ زمینه فرم سبز شده و در عنوان فرم و `label1` عبارت `Mouse Moved` نوشته می‌شود.
۸. اگر `label` کلیک گردید، برنامه خاتمه یابد.



مراحل طراحی و اجرا

۱. پروژه جدیدی به نام `Event2` ایجاد کنید.
۲. یک کنترل `Label` به فرم اضافه کنید.
۳. رویداد `MouseLeave` را برای فرم اضافه کنید. برای این منظور مراحل زیر را انجام دهید:
 - ❖ ناحیه خالی فرم را کلیک کرده تا انتخاب گردد.
 - ❖ کلید `F4` را بزنید تا پنجره خواص فرم را ببینید.

- ❖ در پنجره خواص، دکمه Event را کلیک نمایید تا لیست رویدادهای فرم را مشاهده کنید.
- ❖ رویداد MouseLeave را پیدا کرده، جلوی آن کلیک مضاعف کنید و دستورات آن را به صورت زیر

تغییر دهید:

```
private void Form1_MouseLeave(object sender, EventArgs e)
{
    label1.BackColor = Color.White;
    this.BackColor = Color.Blue;
}
```

دستور اول، رنگ زمینه label1 را سفید کرده و دستور دوم، رنگ زمینه فرم را به آبی تغییر می‌دهد.

۴. رویداد Click را مانند مرحله ۳ برای فرم اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_Click(object sender, EventArgs e)
{
    this.Text = "Form Clicked";
    label1.Text = "Form Clicked";
}
```

این دستورات با کلیک روی فرم، عنوان فرم و label1 را به عبارت Form Clicked تغییر می‌دهند.

۵. کنترل label1 را کلیک کنید تا انتخاب شود. اکنون مانند مرحله ۳ رویداد MouseEnter را برای کنترل

label1 اضافه کرده دستورات آن را به صورت زیر تغییر دهید:

```
private void label1_MouseEnter(object sender, EventArgs e)
{
    label1.ForeColor = Color.Green ;
    this.BackColor = Color.Red ;
}
```

این دستورات با ورود مکان نما به label1، رنگ زمینه فرم را قرمز و رنگ نوشته label1 را سبز می‌نمایند.

۶. ناحیه خالی فرم را کلیک کنید تا انتخاب شود. اکنون رویداد KeyPress را برای فرم (مانند مرحله ۳)

اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_KeyPress(object sender,KeyPressEventArgs e)
{
    this.Text = "Key Pressed";
    label1.Text = "Key Pressed";
}
```

این دستورات با فشردن کلیدی روی فرم عنوان فرم و label1 را به عبارت Key Pressed تغییر می‌دهند.

۷. رویداد ResizeBegin را برای فرم اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_ResizeBegin(object sender, EventArgs e)
{
    this.ForeColor = Color.Green;
    label1.ForeColor = Color.Green;
    this.Text = "Resizing";
    label1.Text = "Resizing";
}
```

دستورات اول و دوم، رنگ نوشته فرم و label1 را به آبی تغییر می‌دهند و دستورات سوم و چهارم، عنوان

فرم و label1 را عبارت Resizing تعیین می‌کنند.

۱۱ آشنایی با زبان C#

۸. رویداد MouseMove را برای فرم اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    this.BackColor = Color.Green;
    label1.Text = "Mouse Moved";
}
```

دستور اول، رنگ زمینه فرم را سبز کرده و دستور دوم، بر روی label1 عبارت Mouse Moved را نمایش

می دهد.

۹. بر روی label1 کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

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

۱۰. برنامه را ذخیره و اجرا کنید. مکان نما را روی فرم حرکت دهید تا شکل زیر را ببینید:



اندازه فرم را تغییر دهید تا خروجی به شکل زیر در آید:



مثال ۴ - ۱. برنامه ای که اعمال زیر را انجام می دهد:

- ❖ یک کنترل TextBox و سه کنترل Button به فرم اضافه می نماید.
- ❖ با کلیک دکمه button1، محتویات TextBox را یک واحد به سمت راست انتقال می دهد.
- ❖ با کلیک دکمه button2، محتویات TextBox را یک واحد به سمت چپ انتقال می دهد.
- ❖ با کلیک دکمه button3، برنامه خاتمه می یابد.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Move ایجاد کنید.

۲. یک کنترل Label، یک کنترل TextBox و سه کنترل Button به فرم اضافه کنید.



۳. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Form1_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Right";
    button2.Text = "Left";
    button3.Text = "Exit";
    label1.Text = "String";
}
```

این دستورات خواص کنترل‌های روی فرم را مقداردهی می‌کنند.

۴. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = " " + textBox1.Text;
}
```

این دستورات، اطلاعات textBox1 را یک واحد به سمت راست حرکت می‌دهند.

۵. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    textBox1.Text = (textBox1.Text.Length > 0 )?
    textBox1.Text.Substring(1, textBox1.Text.Length -
    1):textBox1.Text ;
}
```

این دستورات، با استفاده از عملگر؟، اگر طول محتویات textBox1 بیشتر از صفر باشد، آن را یک کاراکتر

به سمت چپ منتقل می‌کنند.

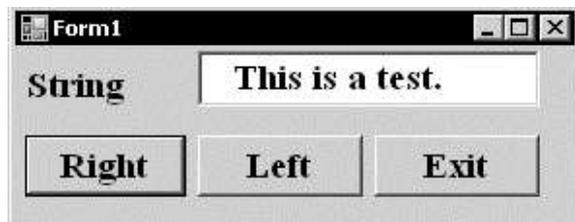
۶. دکمه button3 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تایپ کنید:

```
private void button3_Click(object sender, EventArgs e)
{
    Close();
}
```

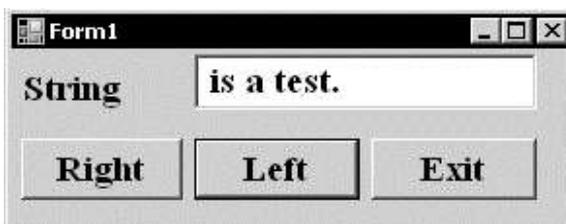
این دستورات، برنامه را خاتمه می‌دهند.

۷. پروژه را ذخیره و اجرا کنید. عبارت Tthis is a test. را تایپ کرده سه بار دکمه Right را کلیک کنید تا

شکل زیر ظاهر شود:



اکنون ۷ بار دکمه Left را کلیک کنید تا شکل زیر ظاهر گردد:



مثال ۵ - ۱. برنامه‌ای که اعمال زیر را انجام می‌دهد (هدف برنامه آشنایی با خواص فرم در زمان طراحی):

- + عنوان فرم را اولین برنامه ویندوز فرم تعیین می‌کند.
- + تصویری را در زمینه فرم قرار داده به طوری که تصویر به اندازه فرم تغییر یابد و کل فرم را بپوشاند.
- + دکمه بیشینه (Maximize) روی فرم را غیر فعال می‌کند.
- + آیکن روی عنوان فرم را مخفی می‌کند.
- + فرم را در هنگام اجرا در وسط صفحه نمایش قرار می‌دهد.

مراحل طراحی و اجرا



۱. پروژه جدیدی از نوع Windows Forms Application ایجاد کنید.

برای این منظور مراحل زیر را انجام دهید:

+ گزینه File/New/Project را اجرا کنید تا پنجره New

Project ظاهر شود.

+ در سمت چپ این پنجره، زبان Visual C# و سپس گزینه Windows را انتخاب کنید.

+ در وسط پنجره New Project، نوع پروژه را Windows Forms Application انتخاب کنید.

+ در جلوی نام پروژه را Ch2-1 وارد کنید.

+ در جلوی Location مسیر D:\booksharp\2 را انتخاب کنید. این پروژه در این مسیر قرار می‌گیرد (البته

شما می‌توانید مسیر دیگر را انتخاب کنید).

+ دکمه OK را کلیک کنید تا پنجره جدید ایجاد شود.

۲. کلید F4 را فشار دهید تا پنجره Properties فرم ظاهر شود. در این پنجره خاصیت Text را پیدا کرده

جلوی آن "اولین برنامه ویندوز فرم" را وارد کنید (مانند شکل زیر):

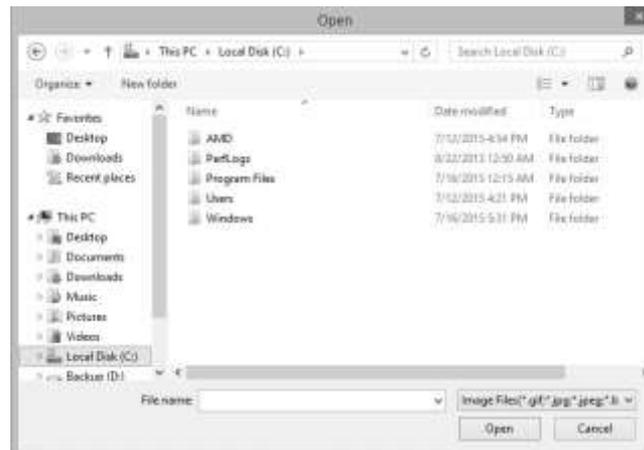


۳. در پنجره Properties خاصیت BackgroundImage را پیدا کرده، اعمال زیر را انجام دهید:
 + جلوی این خاصیت کلیک کنید تا دکمه [...] ظاهر شود.

+ دکمه [...] را کلیک کنید تا پنجره Select Resource ظاهر شود (شکل زیر).



+ در پنجره Select Resource، دکمه Import را کلیک کنید تا پنجره Open ظاهر شود (شکل زیر):



+ در بخش درایو، درایو موردنظر را انتخاب کنید (من درایو C را انتخاب کردم).

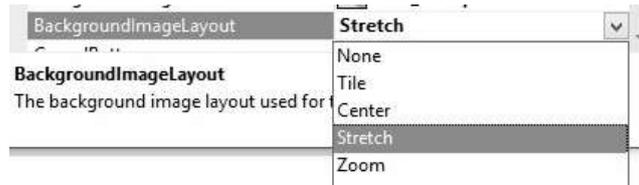
+ اگر مسیر تصویر را می‌دانید، آن تصویر را از مسیر موردنظر انتخاب کنید. وگرنه در بخش Search Local Disk(C:)، *.jpg را تایپ کرده و کلید Enter را فشار دهید تا لیست تصاویر درایو C نمایش داده شوند.

+ اکنون بر روی یک تصویر کلیک مضاعف کنید تا پنجره Select Resource مجدداً ظاهر شود و تصویر موردنظر به این پنجره اضافه گردد.

+ دکمه OK را در پنجره Select Resource کلیک کنید تا تصویر به فرم برنامه اضافه شود.

آشنایی با زبان C# ۱۵

۴. اکنون در پنجره Properties خاصیت BackgroundImageLayout را پیدا کرده، دکمه \blacktriangledown جلوی آن کلیک کنید تا مقادیر این خاصیت ظاهر شوند. اکنون خاصیت Stretch را انتخاب نمایید تا اندازه تصویر به اندازه فرم تغییر یابد (مانند شکل زیر انتخاب کنید):



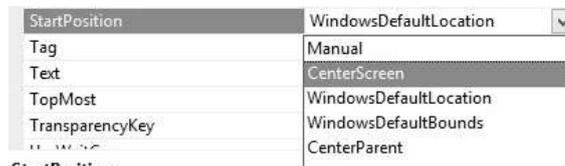
۵. در پنجره Properties، خاصیت MaximizeBox را پیدا کرده و مقدار خاصیت آن را False تعیین کنید تا دکمه بیشینه روی فرم غیرفعال شود (مانند شکل زیر):



۶. در پنجره Properties، خاصیت ShowIcon را پیدا کرده، مقدار آن را به False تغییر دهید تا آیکون روی عنوان فرم مخفی شود (مانند شکل زیر):



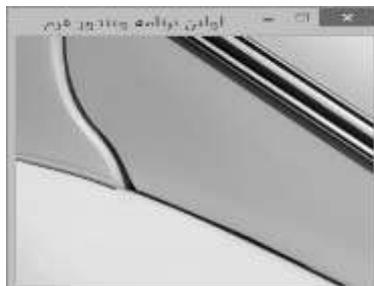
۷. در پنجره Properties، خاصیت StartPosition را انتخاب کرده و مقدار آن را مانند شکل زیر Center Screen انتخاب نمایید تا در هنگام اجرا فرم در وسط صفحه نمایش قرار گیرد:



StartPosition
Determines the position of a form when it first appears.

۸. با گزینه File / Save All (کلیدهای ترکیبی Ctrl + Shift + s) پروژه را ذخیره کنید.

۹. با دکمه  یا گزینه Debug / Strat Debugging (کلید F5) پروژه را اجرا کنید تا خروجی پروژه به شکل زیر ظاهر شود:



همان طور که در این شکل مشاهده می گردد، آیکون روی عنوان فرم، مخفی شده است، تصویری در زمینه فرم نمایش داده، فرم در وسط صفحه نمایش ظاهر گردید، عنوان فرم “اولین برنامه ویندوز فرم” می باشد و دکمه بیشینه نیز غیرفعال گردید.

مثال ۶-۱. برنامه‌ای که اعمال زیر را انجام می‌دهد (هدف برنامه آشنایی با خواص فرم در زمان طراحی):

➤ رنگ زمینه فرم را آبی تعیین می‌کند.

➤ دکمه‌های پیشینه، کمینه، بستن فرم و منوی سیستمی فرم را غیرفعال کنید.

➤ مکان نما را به شکل + تغییر می‌دهد.

➤ اندازه فرم را 150×150 انتخاب می‌کند.

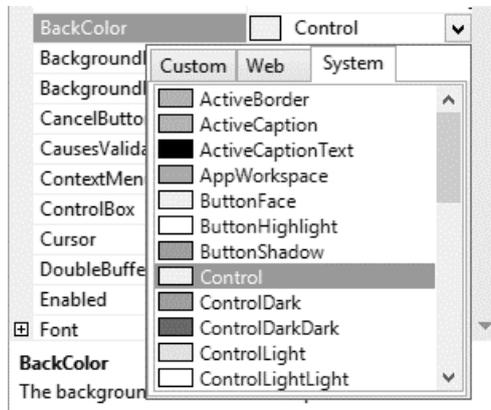
➤ شفافیت فرم را ۵۰ درصد انتخاب می‌کند.

مراحل طراحی و اجرا

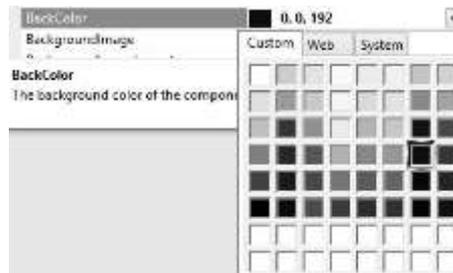


۱. پروژه جدیدی به نام Ch2-2 از نوع Windows Forms Application ایجاد کنید. برای این منظور گزینه File / New / Project را کلیک کنید تا پنجره New Project ظاهر شود. در این پنجره، زبان را Visual C# و نوع پروژه را Windows Forms انتخاب کرده، سپس پروژه را با نوع Windows Forms Application انتخاب نمایید، جلوی نام پروژه را Ch2-2 انتخاب کرده، دکمه OK را کلیک کنید.

۲. با کلید F4 پنجره Properties را فعال کنید. در پنجره Properties خاصیت BackColor را پیدا کرده، دکمه جلوی آن را کلیک کنید تا رنگ‌ها را ببینید (مانند شکل زیر):



اکنون می‌توانید رنگ را انتخاب کنید یا تب Custom را کلیک کنید تا شکل زیر ظاهر شود:



آشنایی با زبان C# ۱۷

در این شکل رنگ آبی را انتخاب کنید.

۳. در پنجره Properties خاصیت ControlBox را پیدا کرده، مقدار آن را به False تغییر دهید تا دکمه‌های پیشینه، کمینه، بستن و منوی سیستمی فرم غیرفعال شوند (این خاصیت را مانند شکل زیر تغییر دهید):



۴. در پنجره Properties، خاصیت Cursor را پیدا کرده و مقدار مکان‌نمای آن را  تعیین کنید.

۵. در پنجره Properties، خاصیت Size را پیدا کرده و مقدار خواص Width و Height آن را ۱۵۰ وارد کنید. برای این منظور، دکمه + جلوی خاصیت را کلیک کنید تا خواص داخلی Width و Height آن ظاهر شوند. اکنون مانند شکل زیر مقدار این خواص را ۱۵۰ انتخاب کنید:



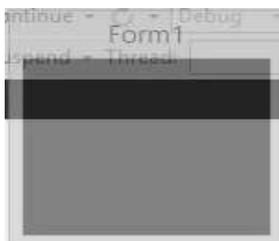
۶. در پنجره Properties، خاصیت Opacity را پیدا کرده و مقدار آن را ۵۰٪ وارد کنید (مانند شکل زیر):



۷. با گزینه File /Save All پروژه را ذخیره کنید.

۸. با گزینه Debug /Strat Debugging (کلید F5) پروژه را اجرا

کنید تا شکل زیر ظاهر شود.

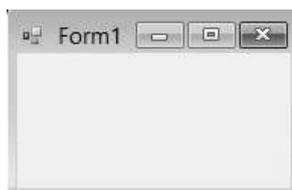


همان‌طور که در این شکل می‌بینید، اندازه فرم 150×150 انتخاب گردید، دکمه‌های منوی سیستمی فرم مخفی شدند، رنگ زمینه فرم آبی تعیین گردید و شفافیت فرم ۵۰ درصد انتخاب شده است. چون دکمه بستن فرم وجود ندارد، با کلیدهای ترکیبی Alt + F4 برنامه را خاتمه دهید.

مثال ۷-۱. برنامه‌ای که اعمال زیر را انجام می‌دهد (هدف برنامه اضافه کردن رویدادها):

- در هنگام بار کردن فرم، کورسور فرم را NO انتخاب می‌کند، شفافیت فرم را ۷۰ درصد انتخاب می‌نماید، دکمه کمینه (Minimize) فرم را غیرفعال می‌کند و رنگ زمینه فرم را زرد انتخاب می‌نماید.
- با کلیک روی فرم شفافیت فرم به ۱۰۰ درصد تغییر یابد، رنگ زمینه فرم به قرمز تغییر یابد.
- چنانچه مکان‌نما فرم را ترک کند، برنامه خاتمه یابد.

مراحل طراحی و اجرا



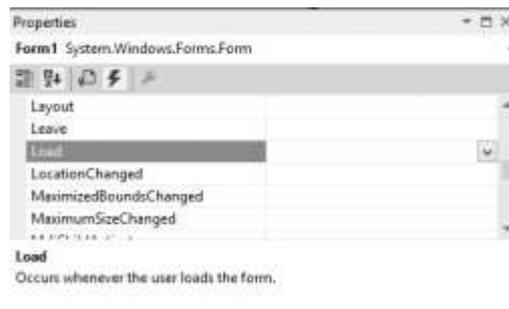
۱. پروژه جدیدی با نوع Windows Forms Application ایجاد کنید.

۲. رویداد Form1_Load را به برنامه اضافه کنید. برای این منظور

مراحل زیر را انجام دهید:

➤ کلید F4 را بزنید تا پنجره Properties ظاهر شود.

در پنجره Properties، دکمه  را کلیک کنید تا رویدادهای فرم ظاهر شوند. رویداد Load را پیدا کنید (البته چون این رویداد، رویداد پیش فرض فرم است، مکان نما روی این رویداد قرار دارد. رویدادهای پیش فرض را می توان با کلیک مضاعف برای کنترل (فرم) اضافه نمود). اکنون پنجره Properties به شکل زیر تغییر می یابد:



جلوی رویداد Load کلیک مضاعف کرده تا رویداد Form1_Load به برنامه اضافه شود (مانند شکل زیر):



همان طور که مشاهده می شود، این رویداد (Form1_Load) متدی است که دسترسی به آن از نوع خصوصی (private) است (یعنی، فقط در همین کلاس می توان این متد را فراخوانی کرد)، هیچ مقداری را بر نمی گرداند (از نوع void است) و دو پارامتر به نام های sender و e دارد که پارامتر sender از نوع object است و تعیین می کند که این رویداد بر روی چه شی ای اتفاق افتاده است. اما پارامتر e اطلاعاتی در مورد رویداد رخ داده را نگهداری می کند. در ادامه بیش تر با پارامترهای sender و e آشنا خواهیم شد.

دستورات رویداد Form1_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.Cursor = Cursors.No;
    Opacity = (double)70 / 100;
    MinimizeBox = false;
    BackColor = Color.Yellow;
}
```

دستور اول، کرسر (مکان نما ماوس) را No انتخاب می کند (در این دستور this به معنی همین کلاس (یعنی کلاس Form1) می باشد)، دستور دوم، خاصیت شفافیت فرم را ۷۰ درصد انتخاب می کند (دقت داشته باشید

آشنایی با زبان C# ۱۹

که (double) به معنی این است که عدد ۷۰ بر ۱۰۰ به صورت اعشاری تقسیم گردد و اگر (double) ذکر نشود، این تقسیم به صورت صحیح انجام می شود که نتیجه ۷۰ تقسیم بر ۱۰۰ به صورت صحیح صفر خواهد شد، دستور سوم، دکمه کمینه روی فرم را غیرفعال می کند و دستور چهارم، زمینه فرم را رنگ زرد تعیین می کند.

۳. گزینه Window/Form1.cs[Design] را کلیک کنید تا به محیط طراحی برگردید.

۴. رویداد Click را پیدا کرده، دو بار کلیک کنید تا رویداد Form1_Click اضافه شود. اکنون دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_Click(object sender, EventArgs e)
{
    Opacity = 1;
    BackColor = Color.Red;
}
```

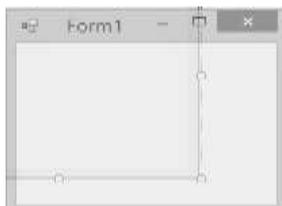
دستور اول، شفافیت فرم را به ۱۰۰ درصد تغییر می دهد و دستور دوم، رنگ زمینه فرم را به قرمز تغییر می دهد.

۵. تب Form1.cs [Design] بالای محیط کد نویسی را کلیک کنید تا به حالت طراحی برگردید.

۶. رویداد MouseLeave مربوط به Form1 را پیدا کرده، آن را کلیک مضاعف نمایید و دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_MouseLeave(object sender, EventArgs e)
{
    Close();
}
```

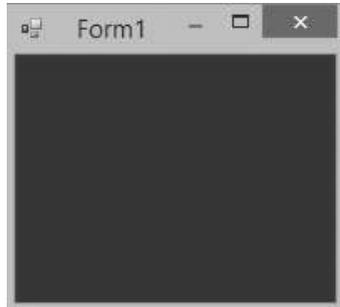
متد Close() موجب بسته شدن فرم فعلی می شود. چون این برنامه فقط یک فرم دارد، با اجرای این متد برنامه خاتمه می یابد.



۷. با گزینه File /Save All پروژه را ذخیره کنید.

۸. با فشار دادن کلید F5 پروژه را اجرا کنید تا خروجی برنامه به شکل مقابل ظاهر شود.

همان طور که در این شکل مشاهده می شود، شفافیت فرم ۷۰ درصد انتخاب شده است و رنگ زمینه فرم زرد می باشد. اکنون بر روی فرم کلیک کنید تا شکل زیر ظاهر شود. همان طور که در این شکل می بینید، رنگ زمینه فرم قرمز شده است و شفافیت فرم به ۱۰۰ درصد تغییر یافت. مکان نما را از فرم خارج کنید تا برنامه خاتمه یابد.



مثال ۸ - ۱. برنامه‌ای که اعمال زیر را انجام می‌دهد (هدف برنامه آشنایی با خواص، رویدادها و متدهای کنترل‌های Label، TextBox و Button است):

➤ با کلیک دکمه button1، اطلاعات textBox2 را به انتهای textBox1 اضافه می‌کند.

➤ با کلیک دکمه button2، اطلاعات انتخاب شده textBox1 را پاک می‌کند.

➤ با تغییر اندازه فرم، در عنوان فرم عبارت Resizing را نوشته و رنگ نوشته‌های کنترل روی فرم به رنگ سبز تغییر یابد.

➤ با کلیک دکمه button3 تصویر مشخصی بر روی فرم نمایش داده می‌شود.

➤ با فشردن کلید Enter، رویداد button1_Click اجرا گردد.

➤ با کلیک دکمه button4 برنامه خاتمه یابد.

➤ با فشردن کلید ESC رویداد button4_Click اجرا شود.

مراحل طراحی و اجرا

۱. پروژه جدیدی با نوع Windows Forms Application

ایجاد کنید.

۲. دو کنترل TextBox و چهار کنترل Button به فرم اضافه کنید.

۳. کلید F4 را فشار دهید تا پنجره Properties ظاهر شود. اکنون بر روی فرم کلیک کنید تا خواص فرم را مشاهده نمایید. خاصیت AcceptButton فرم را پیدا کرده و به button1 تغییر دهید (شکل زیر). اکنون با فشردن کلید Enter روی فرم، رویداد button1_Click اجرا می‌گردد.



۴. خاصیت CancelButton مربوط به فرم را پیدا کرده و مقدار آن را button4 تعیین کنید (شکل زیر). اکنون با فشردن کلید ESC روی فرم دستورات رویداد button4_Click اجرا می‌شود.



۵. دکمه button1 را کلیک مضاعف کرده تا رویداد button1_Click به پروژه اضافه شود. اکنون دستورات آن را به صورت زیر تغییر دهید:

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

۲۱ آشنایی با زبان C#

```
textBox1.Text += textBox2.Text;  
}
```

این دستورات، اطلاعات کنترل textBox2 را به انتهای کنترل textBox1 اضافه می‌کنند (عملگر += برای الحاق و اتصال دو رشته به کار می‌رود).

۶. به محیط طراحی برگردید.

۷. دکمه button2 را کلیک مضاعف کرده تا رویداد button2_Click به پروژه اضافه شود. اکنون، دستورات آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)  
{  
    textBox1.SelectedText = "";  
}
```

این دستورات، اطلاعات انتخاب شده کنترل textBox1 (مقدار خاصیت textBox1.SelectText) را حذف می‌کنند.

۸. به محیط طراحی برگردید (گزینه Window / Form1.Cs[Design] را کلیک کنید).

۹. دکمه button3 را کلیک مضاعف کرده تا رویداد button3_Click به پروژه اضافه شود. اکنون دستورات آن را به صورت زیر تغییر دهید:

```
private void button3_Click(object sender, EventArgs e)  
{  
    BackgroundImage=Image.FromFile("d:\\BookCSharp\\test.jpg");  
    BackgroundImageLayout = ImageLayout.Stretch;  
}
```

دستور اول تصویر test.jpg موجود در درایو D پوشه bookCSharp را در زمینه فرم قرار می‌دهد و دستور دوم اندازه تصویر را به اندازه فرم تغییر می‌دهد (در اینجا \\ به معنی کاراکتر \ است که کاراکتر اول \ کاراکتر کنترلی می‌باشد).

۱۰. به محیط طراحی برگردید.

۱۱. دکمه button4 را کلیک مضاعف کرده تا رویداد button4_Click به پروژه اضافه شود و دستورات آن را به صورت زیر تغییر دهید:

```
private void button4_Click(object sender, EventArgs e)  
{  
    Application.Exit();  
}
```

این دستورات، اجرای برنامه را خاتمه می‌دهند. متد Exit از کلاس Application موجب خاتمه اجرای برنامه می‌شود.

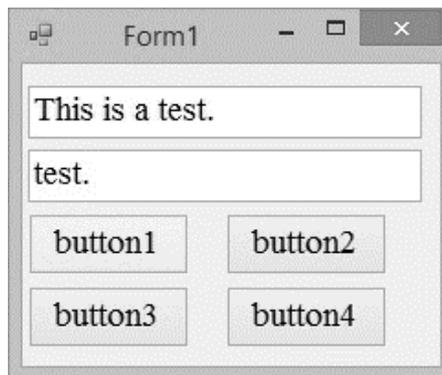
۱۲. به محیط طراحی برگردید.

۱۳. رویداد Resize را برای فرم به پروژه اضافه کنید. برای این منظور، فرم را کلیک کرده تا انتخاب شود. دکمه  را در پنجره Properties کلیک نمایید تا رویدادهای فرم ظاهر شوند. رویداد Resize را پیدا کرده و کلیک مضاعف کنید. اکنون دستورات آن را به صورت زیر تغییر دهید:

```
private void Form1_Resize(object sender, EventArgs e)
{
    this.ForeColor = Color.Green;
    Text = "Resizing";
}
```

دستور اول، رنگ نوشته‌های کنترل روی فرم را به رنگ سبز تغییر می‌دهد و دستور دوم، عنوان فرم را Resizing تعیین می‌کند.

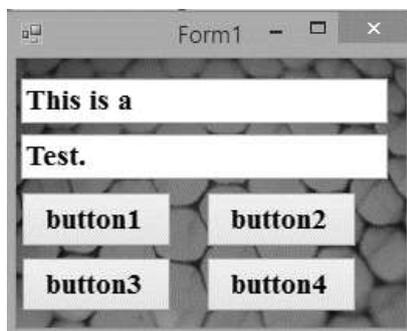
۱۴. پروژه را ذخیره و اجرا کنید. در textBox1، textBox2 به ترتیب This is a test. و test. را وارد کرده دکمه button1 را کلیک کنید تا شکل زیر ظاهر شود.



اکنون عبارت test. را در textBox1 انتخاب کرده، دکمه button2 را کلیک نمایید تا عبارت test. از textBox1 حذف شود (شکل زیر).



دکمه button3 را کلیک کرده تا تصویر موردنظر بر روی فرم ظاهر شود (شکل زیر).



اندازه فرم را تغییر دهید تا خروجی به شکل زیر ظاهر شود.



همان‌طور که در این شکل می‌بینید، عنوان فرم به Resizing تغییر یافته و نوشته‌های کنترل‌های روی فرم به رنگ سبز تغییر یافت. برای خروجی از برنامه دکمه button4 را کلیک کنید یا روی فرم کلید Enter را فشار دهید.

مثال ۹-۱. برنامه‌ای که سه عدد را دریافت کرده، کوچک‌ترین عدد را نمایش می‌دهد (هدف برنامه آشنایی با بیش‌تر با کنترل‌های Label، TextBox، Button و عملکرد ؟ است):

مراحل طراحی و اجرا

۱. پروژه جدیدی با نوع Windows Forms Application ایجاد کنید.

۲. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Form1_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    Text = "Find Minimum";
    label1.Text = "Enter a:";
    label2.Text = "Enter b:";
    label3.Text = "Enter c:";
    label4.ResetText();
}
```

دستور اول، خاصیت Text فرم را به عبارت Find Minimum تغییر می‌دهد، دستورات دوم تا چهارم خاصیت Text کنترل‌های label1 تا label3 را تعیین می‌کنند. دستور پنجم، عنوان label4 را پاک می‌کند.

۳. خاصیت Text کنترل button1 را عبارت &Min تعیین کنید. همان طور که مشاهده می شود، قبل از M عملگر & قرار گرفته است، در این صورت در هنگام اجرا با فشردن کلیدهای ترکیبی Alt + m می توان رویداد button1_Click را اجرا کرد.

۴. خاصیت Text کنترل button2 را عبارت E&xit تعیین کنید. اکنون با فشردن کلید ترکیبی Alt + x می توان رویداد button2_Click را اجرا کرد.

۵. دکمه Min را کلیک مضاعف کرده، دستورات رویداد button1_Click را به صورت زیر تغییر دهید:

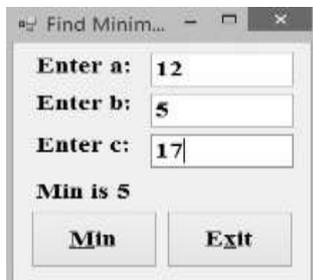
```
private void button1_Click(object sender, EventArgs e)
{
    int a = int.Parse(textBox1.Text);
    int b = Convert.ToInt32(textBox2.Text);
    int c = int.Parse(textBox3.Text);
    int min = a;
    min = (b < min) ? b : min;
    min = (c < min) ? c : min;
    label4.Text = "Min is " + min.ToString();
}
```

دستور اول، متغیر a را با نوع int تعریف کرده، محتوی textBox1 را از طریق متد Parse به عدد تبدیل می نماید و در a قرار می دهد. دستور دوم، متغیر b را با نوع int تعریف کرده، محتوی textBox2 را از طریق متد Convert به عدد تبدیل کرده، در b قرار می دهد، دستور سوم، متغیر c را با نوع int تعریف کرده، محتوی textBox3 را به عدد تبدیل می نماید و در c قرار می دهد، دستور چهارم، متغیر min را با نوع int تعریف کرده، a را در آن قرار می دهد، دستور پنجم، بررسی می کند، اگر b کوچک تر از min باشد، آن را در min قرار می دهد، در دستور ششم، چنانچه c کوچک تر از min باشد، c را در min قرار می دهد و دستور هفتم، مقدار min را به همراه عبارت Min is در label4 نمایش می نماید. چون خاصیت Text کنترل label4 از نوع رشته می باشد، min.ToString() مقدار عددی min را به رشته تبدیل کرده است.

۶. دکمه Exit را کلیک مضاعف کرده، دستورات رویداد button2_Click را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

۷. پروژه را ذخیره و اجرا کنید. اکنون اعداد ۱۲، ۵ و ۱۷ را جلوی Enter a:، Enter b: و Enter c: وارد کرده، کلیدهای ترکیبی Alt + m را فشار دهید تا خروجی را به شکل زیر ببینید:



همان طور که در خروجی می بینید، با فشردن کلیدهای Alt + m، رویداد button1_Click اجرا گردید و کوچک ترین عدد در label4 نمایش داده شد. برای خاتمه برنامه کلید ترکیبی Alt + x را فشار دهید یا دکمه Exit را کلیک کنید.

مثال ۱۰-۱. برنامه ای که روی برچسبی "فناوری نوین" را نمایش می دهد. این برنامه دکمه های برای انتقال برچسب "فناوری نوین" به سمت راست، چپ، بالا و پایین دارد (هدف برنامه آشنایی با خواص Top، Height، Left، Bottom فرم و کنترل ها است):



مراحل طراحی و اجرا
۱. پروژه جدیدی با نوع Windows Forms Application ایجاد کنید.

۲. یک کنترل Label و چهار کنترل Button به فرم اضافه کنید و خاصیت Text برچسب را "فناوری نوین" تعیین نمایید. خاصیت Text دکمه ها را به ترتیب Left + 5، Top + 5، Left - 5 و Top - 5 تعیین نمایید.

۳. دکمه Left + 5 را کلیک مضاعف کرده، دستورات رویداد button1_Click را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    label1.Left = (label1.Right + label1.Text.Length < Width) ? label1.Left + 5 : label1.Left;
}
```

این دستور ابتدا اگر $label1.Right + label1.Text.Length$ (مکان سمت چپ برچسب و طول متن برچسب را با هم جمع کرده)، کم تر از Width فرم (سمت راست فرم) باشد، به سمت چپ برچسب ۵ واحد اضافه می کند (یعنی، اگر برچسب از سمت راست فرم خارج نشود، به سمت چپ آن ۵ واحد اضافه می کند و آن را ۵ واحد به سمت راست انتقال می دهد).

۴. دکمه Top + 5 را کلیک مضاعف کرده، دستورات رویداد button2_Click را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    label1.Top = (label1.Bottom + label1.Height < Height) ? label1.Top + 5 : label1.Top;
}
```

این دستور، اگر `label1.Bottom` (موقعیت پایین برچسب ۱) به علاوه `label1.Height` (ارتفاع برچسب ۱) کم تر از `Heigh` فرم (ارتفاع فرم) باشد، به `label1.Top` ۵ واحد اضافه می کند (یعنی، در صورتی که `label1` از پایین فرم، پایین تر نرود، آن را ۵ واحد به سمت پایین منتقل می کند).

۵. دکمه `Left - 5` را کلیک مضاعف کرده، دستورات رویداد `button3_Click` را به صورت زیر تغییر دهید:

```
private void button3_Click(object sender, EventArgs e)
{
    label1.Left = (label1.Left - 5 > 0) ? label1.Left + - 5 : label1.Left;
}
```

این دستورات، چنانچه `label1` از سمت چپ فرم خارج نشود، از `label1.Left` ۵ واحد کم می کند (یعنی برچسب ۱ را ۵ واحد به سمت چپ منتقل می کند).

۶. دکمه `Top - 5` را کلیک مضاعف کرده، دستورات رویداد `button4_Click` را به صورت زیر تغییر دهید:

```
private void button4_Click(object sender, EventArgs e)
{
    label1.Top = (label1.Top - 5 > 0) ? label1.Top - 5 : label1.Top;
}
```

این دستور، در صورت خارج نشدن `label1` از بالای فرم، از `label1.Top` ۵ واحد کم می کند (یعنی، ۵ واحد `label1` را به سمت بالا منتقل می کند).

۷. پروژه را ذخیره و اجرا کنید. اکنون دکمه های `Left + 5` را کلیک کرده تا عبارت “فناوری نوین” ۵ واحد به سمت راست انتقال یابد. دکمه های دیگر را امتحان نمایید.

مثال ۱۱-۱. برنامه ای که فشردن کلیدهای ↓ (۵ واحد ارتفاع فرم را اضافه می کند)، ↑ (۵ واحد از ارتفاع فرم کم می کند)، → (۵ واحد عرض فرم را اضافه می کند) و ← (۵ واحد از عرض فرم کم می کند) را پیاده سازی می کند (هدف آشنایی رویداد `KeyDown` فرم است):

مراحل طراحی و اجرا

۱. پروژه جدیدی با نوع `Windows Forms Application` پیاده سازی کنید.



۲. کلید `F4` را فشار دهید تا پنجره خواص فرم ظاهر شود. در این پنجره دکمه  را کلیک کرده تا رویدادهای فرم را ببینید. رویداد `KeyDown` را پیدا کرده و کلیک مضاعف نمایید. اکنون دستورات رویداد `Form1_KeyDown` را به صورت زیر تغییر دهید:

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    Height = (e.KeyData == Keys.Down) ? Height + 5 : Height;
    Height = (e.KeyData == Keys.Up) ? Height - 5 : Height;
    Width = (e.KeyData == Keys.Right) ? Width + 5 : Width;
    Width = (e.KeyData == Keys.Left) ? Width - 5 : Width;
}
```

دستور اول، اگر کلید فشار داده شده (e.KeyData) کلید ↓ (Keys.Down) باشد، ۵ واحد به ارتفاع فرم اضافه می‌کند، دستور دوم، چنانچه کلید فشار داده شده کلید ↑ (Keys.Up) باشد، ۵ واحد از ارتفاع فرم کم می‌کند، دستور سوم، اگر کلید فشرده شده کلید → (Keys.Right) باشد، ۵ واحد به عرض فرم اضافه می‌کند و دستور چهارم، چنانچه کلید فشرده شده، کلید ← (Keys.Left) باشد، ۵ واحد از عرض فرم کم می‌کند. توجه کنید با استفاده از خاصیت KeyData پارامتر e رویداد KeyDown می‌توانید کلیدهای دیگر از قبیل D (Keys.D)، Insert (Keys.Insert)، Delete (Keys.Delete)، و غیره را چک کنید.

مثال ۱۲-۱. برنامه‌ای که با کلیک یک دکمه یا برچسب نرم‌افزار Paint اجرا می‌شود. در این برنامه کلیدهای H و h برای مخفی کردن label1، کلیدهای u و U، برای آشکار کردن Label1 مخفی شده، کلیدهای E و e برای فعال کردن label1 و کلیدهای N و n برای غیرفعال کردن label1 به کار می‌روند. اگر label1 غیرفعال باشد، به هیچ رویدادی پاسخ نمی‌دهد (هدف این برنامه استفاده از خواص Visible و Enabled، رویداد KeyPress فرم، فراخوانی متد PerformClick دکمه button1 و متد Start کلاس Process است).



مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Ch2_9 از نوع Windows Forms Application ایجاد کنید.

۲. یک کنترل Label و یک کنترل Button به فرم اضافه کرده، خاصیت Text هر دو را Paint تعیین کنید. با انجام این کار، خاصیت

Text هر دو کنترل label1 و button1 به Paint تغییر می‌یابد. خاصیت KeyPreview فرم را به True تغییر دهید تا در صورت فشردن هر کلید بر روی دکمه یا Label، رویداد KeyPress فرم اجرا شود.

۳. دکمه Paint را کلیک کرده، دستورات رویداد button1_Click را به صورت زیر تغییر دهید.

```
private void button1_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("mspaint");
}
```

این دستور، با فراخوانی متد Start از کلاس Process فایل اجرای msPaint.exe را اجرا می‌کند.

۴. به محیط طراحی برگردید.

۵. برچسب Paint را کلیک مضاعف کرده، دستورات رویداد label1_Click را به صورت زیر تغییر دهید:

```
private void label1_Click(object sender, EventArgs e)
{
    button1.PerformClick();
}
```

این دستور با فراخوانی متد PerformClick() روی button1، رویداد button1_Click() را اجرا می‌کند.

۱. به محیط طراحی برگردید (گزینه Window / Form1.Cs[Degin] را اجرا کنید).

۲. فرم را کلیک کنید تا انتخاب شود. اکنون کلید F4 را فشار دهید تا پنجره Properties ظاهر شود. در

پنجره Properties دکمه  را کلیک کرده تا رویدادهای فرم را مشاهده نمایید. رویداد KeyPress را

پیدا کرده، آن را دو بار کلیک کنید و دستورات رویداد Form1_KeyPress را به صورت زیر تغییر دهید:

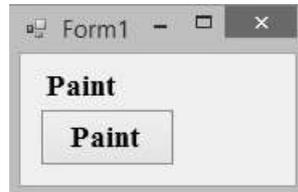
```
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    label1.Visible=(e.KeyChar=='U' || e.KeyChar=='u') ? true: label1.Visible;
    label1.Visible=(e.KeyChar=='H' || e.KeyChar=='h') ? false: label1.Visible;
    label1.Enabled=(e.KeyChar=='E' || e.KeyChar=='e') ? true: label1.Enabled;
    label1.Enabled=(e.KeyChar=='N' || e.KeyChar=='n') ? false: label1.Enabled;
}
```

دستور اول، چنانچه کلیدهای U یا u بر روی فرم فشار داده شوند، label1 را آشکار می‌کند، دستور دوم،

label1 را مخفی می‌نماید، دستور سوم، چنانچه کلیدهای E یا e بر روی فرم فشار داده شوند، label1 را فعال

می‌کند و دستور چهارم، اگر کلیدهای N یا n بر روی فرم فشار داده شوند، label1 را غیرفعال می‌کنند.

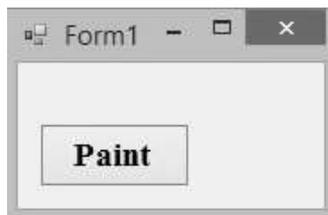
۳. پروژه را ذخیره و اجرا کنید تا (شکل زیر) را ببینید.



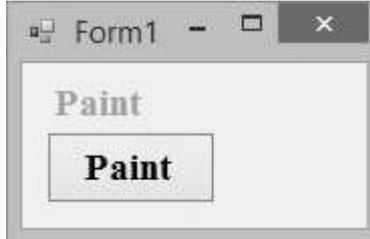
اکنون دکمه Paint را کلیک کرده تا پنجره نرم‌افزار Paint ظاهر شود (زیر).



نرم‌افزار Paint را ببندید و کلید h را فشار دهید تا برجسب Paint مخفی شود (شکل زیر).

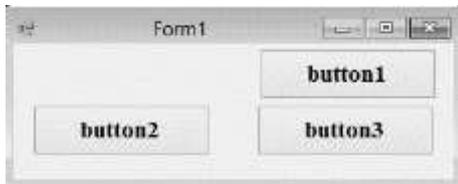


اکنون کلید u را فشار دهید تا برچسب Paint روی فرم ظاهر شود. برچسب Paint را کلیک کرده تا نرم افزار Paint اجرا شود. اکنون نرم افزار Paint را ببندید. کلید n را فشار داده تا فرم برنامه به شکل زیر تغییر یابد.



همان طور که در این شکل می بینید، برچسب Paint غیرفعال گردیده است (رنگ آن خاکستری گردید). اکنون برچسب Paint را کلیک کنید. همان طور که ملاحظه می نماید، نرم افزار Paint اجرا نمی شود، چون برچسب Paint غیرفعال است و به هیچ رویدادی پاسخ نمی دهد. اکنون کلید E را فشار دهید تا برچسب Paint فعال شود.

مثال ۱۳-۱. برنامه ای که برچسبی را در زمان اجرا ایجاد کرده، به فرم برنامه اضافه می کند. در این برنامه دکمه ای برای اضافه کردن رویداد Click و دکمه دیگری برای حذف کردن رویداد Click برچسب اضافه شده وجود دارد (هدف این برنامه آشنایی با چگونگی ایجاد کنترل در زمان اجرا، اضافه و حذف کردن رویدادها در زمان اجرا می باشد).



مراحل طراحی و اجرا
۱. پروژه جدیدی با نوع Windows Forms Application ایجاد کنید.

۲. سه کنترل Button به فرم اضافه کنید.

۳. ناحیه خالی فرم را کلیک کرده، دستورات رویداد Form1_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Create Label";
    button2.Text = "Add Click";
    button3.Text = "Remove Click";
    button2.Enabled = false;
    button3.Enabled = false;
}
```

دستور اول تا سوم، خاصیت Text کنترل های button1 تا button3 را به ترتیب Create Label، Add Click و Remove Click تعیین می کنند، دستور چهارم و پنجم، دکمه های button2 و button3 را غیرفعال می کنند تا label در زمان ایجاد گردد (برای labelی که ایجاد نشده است نمی توان رویداد Click اضافه نمود).

۴. به قبل از رویداد Form1_Load بروید و دستور زیر را تایپ کنید:

```
Label label1;
```

این دستور متغیر label1 را با نوع Label تعریف می‌کند، به طوری که می‌توان در تمام رویدادها از label1 استفاده کرد.

۵. به محیط طراحی برگردید (گزینه Window / Form1.Cs[Degin] را اجرا کنید).

۶. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد button1_Click را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    label1 = new Label();
    label1.Location = new System.Drawing.Point(20, 20);
    label1.Size = new System.Drawing.Size(52, 21);
    label1.AutoSize = true;
    label1.Text = "Created run time";
    Controls.Add(label1);
    button1.Enabled = false;
    button2.Enabled = true;
}
```

دستور اول، یک نمونه از نوع کنترل Label ایجاد کرده، آدرس آن را در label1 قرار می‌دهد، دستور دوم، مکان ایجاد label1 را نقطه (۲۰ و ۲۰) قرار می‌دهد، دستور سوم، اندازه label1 را به ارتفاع ۵۲ و عرض ۲۱ در نظر می‌گیرد. دستور چهارم، خاصیت AutoSize کنترل label1 را true تعیین می‌کند تا با تغییر محتوی label1، اندازه آن نیز تغییر یابد، دستور پنجم، خاصیت Text کنترل label1 را "Create run time" در نظر می‌گیرد، دستور ششم، کنترل label1 را با متد Add به کنترل‌های Form1 اضافه می‌کند، دستور هفتم، خاصیت Enabled کنترل button1 را false تعیین می‌کند تا کاربر مجدداً این دکمه را کلیک نکند و دستور هشتم، خاصیت Enabled، کنترل button2 را true تعیین می‌کند تا کاربر بتواند رویداد Click را برای کنترل label1 اضافه کند.

۷. به محیط طراحی برگردید.

۸. دکمه button2 را کلیک مضاعف کنید، دستورات رویداد button2_Click را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    label1.Click += new System.EventHandler(label1_Click);
    button3.Enabled = true;
    button2.Enabled = false;
}
```

دستور اول، با فراخوانی EventHandler، رویداد label1_Click را به لیست رویدادهای اضافه می‌کند، دستور دوم، خاصیت Enabled کنترل button3 را فعال می‌کند تا با کلیک این دکمه بتوان رویداد label1_Click را از لیست رویدادهای حذف کرد و دستور سوم، دکمه button2 را غیرفعال می‌کند.

۹. رویداد label1_Click را به صورت زیر به پروژه اضافه کنید:

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

۳۱ آشنایی با زبان C#

```
label1.Text = "Label1 Clicked";  
}
```

این دستورات متدی به نام label1_Click تعریف می کنند که با کلیک label1 اجرا شده و خاصیت Text کنترل label1 را به Label1 Click تغییر می دهند.

۱۰. به محیط طراحی برگردید.

۱۱. دکمه button3 را کلیک مضاعف کرده، دستورات رویداد button3_Click را به صورت زیر تغییر دهید:

```
private void button3_Click(object sender, EventArgs e)  
{  
    label1.Click -= new System.EventHandler(label1_Click);  
    label1.Text = "Created run time";  
    button2.Enabled = true;  
    button3.Enabled = false;  
}
```

دستور اول، رویداد label1_Click را از لیست رویدادها حذف می کند، دستور دوم، عنوان label1 را به عبارت "Created run time" تغییر دهد، دستور سوم، کنترل button2 را فعال می کند و دستور چهارم، کنترل button3 را غیر فعال می نماید.

۱۲. پروژه را ذخیره و اجرا کنید تا (شکل زیر) ظاهر شود.



در این شکل دکمه Create Label را کلیک کرده تا برچسب روی فرم ایجاد شود (شکل زیر).



اکنون برچسب Created run time را کلیک کنید (هیچ عملی انجام نمی شود، چون رویداد Click به آن اضافه نگردید). دکمه Add Click را کلیک کنید تا شکل زیر ظاهر شود:



برچسب Created run time را کلیک کنید تا شکل زیر ظاهر گردد.



همان طور که در شکل می بینید، عنوان برچسب به label1_Click تغییر یافته است (یعنی رویداد label1_Click اضافه شده اجرا گردیده، در ادامه دکمه Remove Click را کلیک نمایید تا رویداد label1_Click حذف شود. اکنون برچسب Created run time را کلیک نمایید. همان طور که ملاحظه می گردد، هیچ عملی انجام نمی شود، چون رویداد Label1_Click حذف گردید.

ساختار تصمیم و حلقه تکرار

مثال ۱-۲. برنامه‌ای که حقوق ناخالص و نوع بیمه را دریافت می‌کند (حقوق ناخالص عددی و نوع بیمه به صورت RadioButton باشد) و با استفاده از فرمول زیر بیمه، مالیات و حقوق خالص را محاسبه می‌کند:
حقوق خالص = مالیات - بیمه - حقوق ناخالص

- ❖ اگر نوع بیمه ۷٪ انتخاب شده باشد، ۷ درصد حقوق ناخالص بیمه خواهد بود.
- ❖ اگر نوع بیمه ۹٪ انتخاب شود، ۹ درصد حقوق ناخالص بیمه خواهد بود.
- ❖ اگر نوع بیمه ۱۰٪ انتخاب شود، ۱۰ درصد حقوق ناخالص بیمه خواهد بود.
- ❖ اگر کنترل CheckBox کسر مالیات انتخاب شده باشد و حقوق ناخالص بیش از ۴۵۰۰۰۰۰ ریال باشد، مازاد بر این مبلغ ۱۰٪ مالیات خواهد بود، وگرنه مالیات صفر منظور خواهد شد.
- ❖ در بخش حقوق از ورود داده‌های غیر عددی جلوگیری شود.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Salary ایجاد کنید.
۲. دو کنترل Label به فرم اضافه کنید. خاصیت Text اولین Label را حقوق تعیین کنید.
۳. سه کنترل RadioButton به فرم اضافه کرده، خاصیت Text آن‌ها را به ترتیب ۷ درصد، ۹ درصد و ۱۰ درصد تعیین کنید.
۴. یک کنترل CheckBox به فرم اضافه کرده، خاصیت Text آن را جانباز تعیین کنید.
۵. دو کنترل Button به فرم اضافه کرده، خاصیت Text آن‌ها را به ترتیب محاسبه و خروج تعیین نمایید.

۶. دکمه محاسبه را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    double salary = Convert.ToDouble(textBox1.Text);
    double tax= 0, insuranc = 0;
    if (radioButton1.Checked == true) insuranc = 0.07 * salary;
    else if (radioButton2.Checked == true) insuranc = 0.09*salary;
    else if (radioButton3.Checked == true) insuranc = 0.1*salary;
    if (checkBox1.Checked == false && salary > 4500000)
        tax = (salary - 4500000) * 0.1;
    double pay = salary - insuranc - tax;
    label2.Text = "";
}
```

```

label2.Text = label2.Text + tax.ToString() + " : "+"مالیات";
label2.Text=label2.Text+"\n"+insuranc.ToString()+" : " + "بیمه";
label2.Text = label2.Text+"\n"+pay.ToString() + " : " + "حقوق";
}

```

دستور اول، حقوق را خوانده در salary قرار می‌دهد. دستور دوم، بیمه و مالیات را تعریف کرده در آنها صفر قرار می‌دهد. دستور if تودرتو، با توجه به انتخاب کاربر، بیمه را محاسبه می‌کند. دستور if مربوط به checkBox1، اگر کارمند جانباز نباشد و حقوقش بیش از ۴۵۰۰۰۰۰۰ ریال باشد، ۱۰ درصد حقوق، مازاد بر ۴۵۰۰۰۰۰ مالیات کسر خواهد شد و دستورات بعدی مالیات، بیمه و حقوق را نمایش می‌دهند.

۷. دکمه خروج را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```

private void button2_Click(object sender, EventArgs e)
{
    Close();
}

```

۸. رویداد KeyPress را برای textBox1 اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar)))
        e.Handled = true;
}

```

این دستورات از ورود کاراکترهای غیر عددی در textBox1 جلوگیری می‌کنند.

۹. پروژه را ذخیره و اجرا کنید. جلوی حقوق،

۱۰۰۰۰۰۰۰ ریال را وارد کرده، گزینه ۹ درصد را

انتخاب و دکمه محاسبه را کلیک نمایید

تا خروجی مقابل را ببینید:

اکنون گزینه‌های جانباز و بیمه ۷ درصد را انتخاب کرده،

دکمه محاسبه را کلیک کنید تا خروجی زیر را مشاهده نمایید:

مثال ۲-۲. فروشگاه‌های را در نظر بگیرید که درصدی از فروش را به عنوان دستمزد به فروشنده می‌پردازد. میزان دستمزد با توجه به نوع کالا به صورت زیر محاسبه می‌شود:

کالاهای درجه ۱: اگر میزان فروش کمتر یا مساوی ۲۰۰۰۰۰۰۰ ریال باشد، ۵ درصد به فروشنده پرداخت می‌شود، اگر میزان فروش بین ۲۰۰۰۰۰۰۰ تا ۲۰۰۰۰۰۰۰۰ ریال باشد، ۷ درصد، وگرنه ۱۰ درصد به فروشنده پرداخت می‌شود.

کالاهای درجه ۲: اگر میزان فروش کمتر یا مساوی ۲۰۰۰۰۰۰۰ ریال باشد، ۴ درصد، وگرنه، ۶ درصد به فروشنده پرداخت می‌گردد.

۳۵ ساختار تصمیم و حلقه تکرار

کالاهای درجه ۳: ۴ درصد کل فروش به فروشنده پرداخت خواهد شد.
کالاهای درجه ۴: ۶ درصد کل فروش به فروشنده پرداخت می‌گردد.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Sell ایجاد کنید.
۲. دو کنترل Label به فرم اضافه کرده، خاصیت Text آن‌ها را به ترتیب "فروش" و "" تعیین کنید.
۳. دو کنترل Button به فرم اضافه کرده، خاصیت Text آن‌ها را محاسبه و خروج تعیین نمایید.
۴. چهار کنترل RadioButton به فرم اضافه کرده، خاصیت Text آن‌ها را به ترتیب درجه ۱، درجه ۲، درجه ۳ و درجه ۴ تعیین کنید.



۵. دکمه محاسبه را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    double s = Convert.ToDouble(textBox1.Text);
    double pay = 0;
    if (radioButton1.Checked == true)
    {
        if (s <= 20000000)
            pay = s * 5 / 100;
        else if (s <= 200000000)
            pay = s * 7 / 100;
        else
            pay = s * 10 / 100;
    }
    else if (radioButton2.Checked == true)
    {
        if (s <= 20000000)
            pay = s * 4 / 100;
        else
            pay = s * 6 / 100;
    }
    else if (radioButton3.Checked == true)
    {
        pay = s * 4 / 100;
    }
    else if (radioButton4.Checked == true)
    {
        pay = s * 6 / 100;
    }
}
```

این دستورات با توجه به انتخاب درجه کالا توسط کاربر (از طریق radioButtonها)، دستمزد فروشنده را محاسبه کرده نمایش می‌دهند.

۶. دکمه خروج را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
```

```
{
    Close();
}
```

۷. پروژه را ذخیره و اجرا کنید. نوع کالا را درجه ۲ و مبلغ فروش را ۲۰۰۰۰۰۰۰۰ ریال وارد کرده و دکمه محاسبه را کلیک نمایید تا خروجی زیر را ببینید:

مثال ۳ - ۲. برنامه‌ای که حقوق ناخالص را گرفته و بر اساس قوانین زیر، بیمه، مالیات را کسر کرده، حق تأهل به او تخصیص دهد:

- ❖ اگر بیمه تأمین اجتماعی باشد، ۷ درصد حقوق ناخالص بیمه کسر گردد.
 - ❖ اگر بیمه خدمات درمانی باشد، ۹ درصد حقوق ناخالص بیمه باشد.
 - ❖ اگر بیمه نیروی انتظامی باشد، ۱۰ درصد حقوق ناخالص بیمه می‌باشد.
 - ❖ اگر کارمند جانباز باشد، مالیات معاف است.
 - ❖ اگر کارمند منطقه نیمه توسعه یافته باشد، ۵ درصد مزاد بر ۴۶۰۰۰۰۰ ریال مالیات کسر می‌شود.
 - ❖ اگر کارمند عادی باشد، ۱۰ درصد مزاد بر ۴۶۰۰۰۰۰ ریال مالیات کسر می‌شود.
 - ❖ اگر کارمند متأهل باشد، ۴۶۰۰۰۰ ریال حق تأهل به او تعلق می‌گیرد.
- سپس با استفاده از فرمول زیر حقوق خالص مالیات را حساب کرده، نمایش می‌دهد:
- بیمه - حق تأهل + حقوق ناخالص = حقوق خالص**

توضیح: در این برنامه دو کنترل GroupBox در نظر گرفته شده است و در هر یک از کنترل‌های GroupBox سه کنترل RadioButton اضافه شده‌اند. گروه اول، نوع بیمه و گروه دوم نوع کسر مالیات را تعیین می‌کند و یک CheckBox تعیین می‌کند که آیا کارمند متأهل یا خیر.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Pay ایجاد کنید. خاصیت Right To Left فرم را Yes تعیین کنید.
۲. دو کنترل Label، دو کنترل Button و یک کنترل TextBox به فرم اضافه کرده، خاصیت کنترل‌های Label را به ترتیب حقوق ناخالص و خالی تعیین کنید. خاصیت کنترل‌های Button را به ترتیب محاسبه و خروج تعیین نمایید.
۳. دو کنترل GroupBox به فرم اضافه کنید. خاصیت Text آن‌ها را به ترتیب نوع بیمه و نوع مالیات تعیین کنید.
۴. یک کنترل CheckBox به فرم اضافه کرده، خاصیت Text آن را متأهل تعیین کنید.
۵. سه کنترل RadioButton به کنترل groupBox1 (نوع بیمه) اضافه کرده، خاصیت Text آن‌ها را به ترتیب تأمین اجتماعی، خدمات درمانی و نیروی انتظامی تعیین کنید.
۶. سه کنترل RedioButton دیگر به کنترل نوع مالیات اضافه کرده، خاصیت Text آن‌ها را به ترتیب جانباز، نیمه توسعه یافته و عادی تعیین کنید.

۳۷ ساختار تصمیم و حلقه تکرار

۷. دکمه محاسبه را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    double salary = Convert.ToDouble(textBox1.Text);
    double tax = 0, wife = 0, insuranc = 0;
    if (radioButton1.Checked == true)
        insuranc = salary * 7 / 100;
    else if (radioButton2.Checked == true)
        insuranc = salary * 9 / 100;
    else if (radioButton3.Checked == true)
        insuranc = salary * 10 / 100;
    if (radioButton4.Checked == true)    tax = 0;
    else if (radioButton5.Checked == true)
        tax = (salary - 4600000) * 5 / 100;
    else if (radioButton6.Checked == true)
        tax = (salary - 4600000) * 10 / 100;
    if (checkBox1.Checked == true)    wife = 460000;
    double pay = salary + wife - insuranc - tax;
    label2.Text = " مالیات " + insuranc.ToString() + " " + " بیمه "
        + tax.ToString() + "\n";
    label2.Text += " پرداختی " + wife.ToString() + " " + " تاهل "
        + pay.ToString() ;
}
```

این دستورات ابتدا، حقوق ناخالص را خوانده و متغیرهای بیمه (insuranc)، مالیات (tax) و حق تاهل (wife) را تعریف می کنند. سپس، با توجه به انتخاب کاربر و با استفاده از دو if متداخل، بیمه و مالیات را محاسبه می کنند. در پایان، اگر کارمند حق تاهل بگیر است، حق تاهل را برای او در نظر می گیرد و حقوق خالص را محاسبه کرده، بیمه، مالیات، حق تاهل و حقوق خالص را نمایش می دهد.

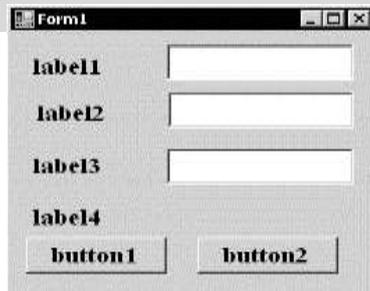
۸. دکمه خروج را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

۹. پروژه را ذخیره و اجرا کنید. جلوی حقوق ناخالص مقدار ۱۰۰۰۰۰۰۰ را وارد کرده، گزینه های متاهل، تأمین اجتماعی، نیمه توسعه یافته را انتخاب نمایید و دکمه محاسبه را کلیک کنید تا خروجی زیر را ببینید:



مثال ۴ - ۲. برنامه‌ای که سن‌تان را به سال، ماه و روز گرفته به دقیقه تبدیل کرده و نمایش می‌دهد (هر سال ۳۵۶ روز، هر ماه ۳۰ روز و هر روز ۲۴ ساعت است). در این برنامه کاربر می‌تواند از طریق کلیدهای ↑، ↓، ←، →، Home، End بین TextBox‌ها حرکت کند.



مراحل طراحی و اجرا

۱. پروژه جدیدی به نام KeyTest ایجاد کنید.
۲. چهار کنترل Label، سه کنترل TextBox و دو کنترل Button به فرم اضافه کنید.
۳. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Load آن را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.RightToLeft = RightToLeft.Yes;
    this.Text = "Key test";
    label1.Text = "Year";
    label2.Text = "Month";
    label3.Text = "Day";
    label4.ResetText();
    button1.Text = "Convert";
    button2.Text = "Exit";
    textBox1.Focus();
}
```

دستور اول، جهت نمایش عنوان فرم و اطلاعات روی آن را از راست به چپ تغییر می‌دهد. دستور دوم، عنوان فرم را به Key test تغییر خواهد داد. دستورات بعدی، عناوین کنترل‌های روی فرم را تغییر می‌دهند و دستور آخر مکان‌نما به کنترل textBox1 انتقال می‌دهد.

۴. خاصیت TabIndex کنترل‌های textBox1، textBox2، textBox3 را به ترتیب صفر، یک و دو انتخاب کنید تا این کنترل‌ها به صورت پشت سرهم قرار گیرند و با فشردن کلیدهای Tab، Enter، ↓ به کنترل بعدی و با فشردن کلید ↑ به کنترل قبلی انتقال یابند.

۵. متد nextControl را به قبل از رویداد Form1_Load به صورت زیر اضافه کنید:

```
private void nextControl(Control current, KeyEventArgs e, Control
first, Control last)
{
    switch (e.KeyCode)
    {
        case Keys.Down :
        case Keys.Enter:
```

۳۹ ساختار تصمیم و حلقه تکرار

```
Control next = this.GetNextControl(current, true);
if (next != null && next.GetType().ToString() ==
"System.Windows.Forms.TextBox")
    this.GetNextControl(current, true).Focus();
else
    first.Focus();
break;
case Keys.Up:
Control pre = this.GetNextControl(current, false);
if (pre != null && pre.GetType().ToString() ==
"System.Windows.Forms.TextBox")
    this.GetNextControl(current, false).Focus();
else last.Focus();
break;
case Keys.End: // End key
    last.Focus();
break;
case Keys.Home:
    first.Focus();
break;
}
}
```

این متد چهار پارامتر را می‌پذیرد. پارامتر اول، از نوع **Control** می‌باشد که کنترل فعلی را تعیین می‌کند. پارامتر دوم، از نوع **KeyEventArgs** به نام **e** است که کلید فشرده‌شده و اطلاعاتی درباره آن را نگهداری می‌کند. پارامترهای سوم و چهارم از نوع **Control** می‌باشند که اولین کنترل و آخرین کنترلی که مکان‌نما باید به آن‌ها منتقل شود را تعیین می‌کنند. در این متد، ابتدا **e.KeyCode** تست می‌شود و با توجه به مقادیر زیر، یکی از تصمیمات اتخاذ می‌گردد:

❖ مقدار **Keys.Down** یا **Keys.Enter**. اگر یکی از کلیدهای **Down** (↓) یا **Enter** باشد، کنترل بعدی با متد **GetNextControl** تعیین می‌شود. اگر کنترل بعدی **null** نباشد و نوع کنترل **TextBox** باشد، مکان‌نما به کنترل بعدی انتقال می‌یابد. وگرنه، مکان‌نما به اولین کنترل (**first.Focus()**) انتقال می‌یابد.

❖ مقدار **Keys.Up**. اگر کلید **↑** فشار داده شود، کنترل قبلی با فراخوانی متد **GetNextControl** (با پارامتر دوم **false**) تعیین گردیده در کنترل **Pre** قرار می‌گیرد. اگر این کنترل **null** نباشد و نوع آن **TextBox** باشد، مکان‌نما به این کنترل انتقال می‌یابد. وگرنه، مکان‌نما به آخرین کنترل (**last.Focus()**) انتقال می‌یابد.

❖ مقدار **Keys.End**. اگر کاربر کلید **End** را فشار دهد، مکان‌نما را به آخرین کنترل انتقال می‌دهد.

❖ مقدار **Keys.Home**. اگر کاربر کلید **Home** را فشار دهد، مکان‌نما به اولین کنترل انتقال می‌یابد.

۶. رویداد **KeyDown** را برای کنترل **textBox1** اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void textBox1_KeyDown(object sender, EventArgs e)
{
    nextControl(textBox1, e, textBox1, textBox3 )
}
```

این رویداد، متد **nextControl** را فراخوانی می‌کند تا با توجه به کلید فشرده‌شده، مکان‌نما به کنترل موردنظر انتقال یابد.

۷. رویداد **KeyDown** را برای کنترل **textBox2** اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void textBox2_KeyDown(object sender, KeyEventArgs e)
{
    nextControl(textBox2, e, textBox1, textBox3 );
}
```

۸. رویداد KeyDown را برای کنترل textBox3 اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void textBox3_KeyDown(object sender, KeyEventArgs e)
{
    nextControl(textBox3, e, textBox1, textBox3 );
}
```

۹. کلید button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

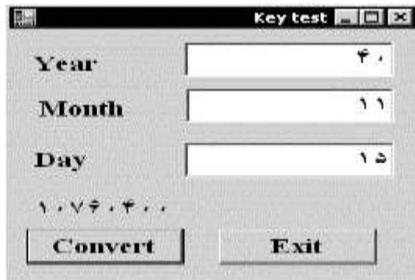
```
private void button1_Click(object sender, EventArgs e)
{
    int year = Convert.ToInt16(textBox1.Text);
    if (year < 0 )
    {
        MessageBox.Show("Enter year > 0");        return;
    }
    int month = Convert.ToInt16(textBox2.Text);
    if (month < 0 || month > 11)
    {
        MessageBox.Show("Enter month 0 to 11");    return;
    }
    int day = Convert.ToInt16(textBox3.Text);
    if (day < 0 || day > 31)
    {
        MessageBox.Show("Enter day 0 to 30");      return;
    }
    long minutes = (year * 365 + 30 * month + day) * 24 * 30;
    label4.Text = minutes.ToString();
}
```

دستور اول، محتوی textBox1 را به عدد تبدیل کرده، در متغیر year (سال) قرار می دهد. اگر year کوچک تر از ۰ باشد، پیام مناسب می دهد و از رویداد خارج می گردد. در مرحله بعدی، محتویات textBox2 را به عدد تبدیل کرده، در متغیر month قرار می دهد. اگر month خارج از بازه ۱ تا ۱۲ باشد، پیام مناسب نمایش داده از رویداد خارج می شود. در ادامه محتوی textBox3 را به عدد تبدیل کرده، در متغیر day قرار می دهد. اگر day خارج از بازه ۱ تا ۳۱ باشد، پیام مناسب نمایش می دهد و از برنامه خارج می گردد. در پایان، سال، ماه و روز وارد شده را به دقیقه (minutes) تبدیل کرده، نمایش می دهد.

۱۰. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

ساختار تصمیم و حلقه تکرار ۴۱



۱۱. پروژه را ذخیره و اجرا کنید. اعداد ۴۰، ۱۱ و ۱۵ را جلوی year، month و day وارد کنید با کلیدهای Enter، ↑، End و Home می‌توانید بین textboxes حرکت کنید. اکنون، دکمه Convert را کلیک نمایید تا خروجی روبرو را ببینید:

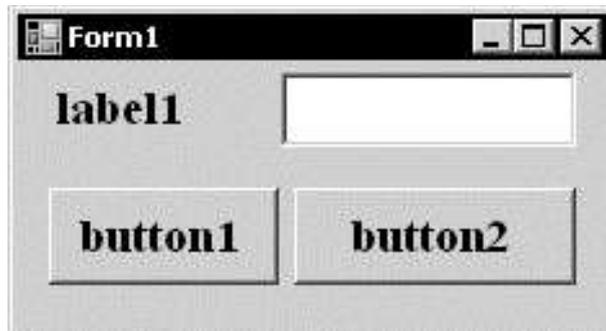
مثال ۵ - ۲. برنامه‌ای که به دانش‌آموزان ضرب را آموزش می‌دهد. در این برنامه با استفاده از متد next کلاس Random یک سری اعداد تصادفی مثبت بین ۰ تا ۹ تولید می‌کند و سپس با استفاده از این اعداد تصادفی یک سؤال به صورت زیر مطرح می‌نماید:
 $5 * 4 = ?$

اکنون، دانش‌آموزان، جواب سؤال را در یک TextBox وارد می‌کنند. اگر پاسخ صحیح باشد، پیغام "very Good!" را نمایش می‌دهد. وگرنه، پیغام "No.Please try again" را نمایش خواهد تا دانش‌آموزان جواب را دوباره وارد کند (برنامه متدی برای طرح سؤال دارد).

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Multiplylearning ایجاد کنید.

۲. یک کنترل Label، یک کنترل TextBox و دو کنترل Button به فرم اضافه کنید.



۳. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد آن را به صورت زیر تایپ کنید:

```
private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Create";
    button2.Text = "Answer";
}
```

۴. متغیرهای x و y را قبل از رویداد Form1_Load به صورت زیر تایپ کنید:

```
Int x, y;
```

این دستور، دو متغیر سراسری x و y را تعریف می‌کند.

۵. تابع random را پس از تعریف متغیرها به صورت زیر تایپ کنید:

```
private void random(out int x, out int y)
{
    Random r1 = new Random();
    Random r = new Random(r1.Next());
    x = r1.Next(9);
    y = r1.Next(9);
}
```

```
    }
```

این تابع دو عدد تصادفی بین ۰ تا ۹ تولید کرده، از طریق آرگومان‌های x و y برمی‌گرداند.

۶. دکمه `button1` را کلیک مضاعف کرده، دستورات رویداد `Click` آن را به صورت زیر تغییر دهید:

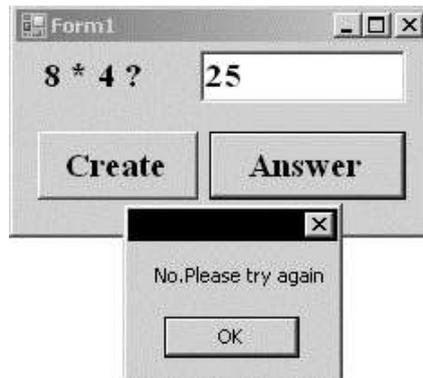
```
private void button1_Click(object sender, EventArgs e)
{
    int x, y;
    random( out x, out y);
    label1.Text = x.ToString() + " * " + y.ToString()+ "?";
}
```

این دستورات، دو عدد تصادفی را تولید کرده (با فراخوانی تابع `Random`) آن‌ها را در x و y قرار داده بر روی `Label` نمایش می‌دهد.

۷. دکمه `button2` را کلیک مضاعف کرده، دستورات رویداد `Click` آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    if (Convert.ToInt16(textBox1.Text) == x * y)
        MessageBox.Show("Very Good");
    else
        MessageBox.Show("No.Please try again");
}
```

۸. پروژه را ذخیره و اجرا کنید. دکمه `Create` را کلیک کرده عدد ۲۵ را وارد نمایید. و دکمه `Answer` را کلیک نمایید تا خروجی زیر ظاهر شود:



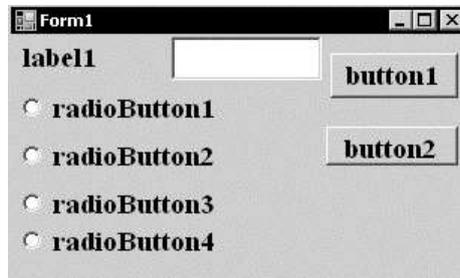
مثال ۶-۲. برنامه که اعتبار سنجی اطلاعات یک `TextBox` را انجام می‌دهد. یعنی کاربر می‌تواند تعیین کند چه اطلاعاتی را وارد کند. کاربر در این برنامه می‌تواند تعیین کند یکی از اطلاعات عددی، فقط کاراکتر، پست الکترونیک و اعداد اعشاری وارد شود.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام `Validation` ایجاد کنید.

۲. یک کنترل `Label`، یک کنترل `TextBox`، یک کنترل `Button` و چهار کنترل `RadioButton` به فرم اضافه کنید.

ساختار تصمیم و حلقه تکرار ۴۳



۳. فضای نام `System.Text.RegularExpressions` را با دستور زیر در بخش `using` اضافه کنید:

```
using System.Text.RegularExpressions;
```

۴. متد `number` را به صورت زیر تعریف کنید:

```
private void number(object sender, System.Windows.Forms
.KeyPressEventArgs e)
{
    if (!(char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar)))
    {
        e.Handled = true;
    }
}
```

این متد دو پارامتر به نام‌های `sender` (از نوع `object`) و `e` (از نوع `KeyPressEventArgs`) را گرفته از ورود به جز ارقام غیر عددی (۰ تا ۹) و کلیدهای کنترل جلوگیری می‌کند (برای این منظور، اگر کاربر کاراکتری غیر از رقم و کلیدهای کنترلی وارد کند، خاصیت `Handled` برابر `true` می‌شود و از ورود آن جلوگیری می‌کند).

۵. متد `floatNumber` را به صورت زیر تعریف کنید:

```
private void floatNumber(object
sender, System.Windows.Forms.KeyPressEventArgs e)
{
    if (!(char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar)
) || (e.KeyChar == (char)46))
    {
        e.Handled = true;
    }
}
```

این متد از ورود کاراکترهای به جز ارقام، نقطه اعشاری (کد ۶۵) و کاراکترهای کنترلی جلوگیری می‌کند.

۶. متد `letter` را به صورت زیر تعریف کنید:

```
private void letter(object
sender, System.Windows.Forms.KeyPressEventArgs e)
{
    if (!(char.IsLetter(e.KeyChar) || char.IsControl(e.KeyChar)))
    {
        e.Handled = true;
    }
}
```

این متد از ورود کاراکترهای غیر الفبایی و غیر کنترلی جلوگیری می‌کند. این متد تعیین می‌کند که آیا در `emailAddress` (پارامتر) کاراکترهای به جز کاراکترهای تعیین شده در رشته `patternstrict` وارد شده است یا خیر.

۷. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد `Form1_Load` را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = "Value";
    radioButton1.Text = "Number";
    radioButton2.Text = "Float";
    radioButton3.Text = "Letter";
    radioButton4.Text = "Email";
    button1.Text = "Email";
    button2.Text = "Exit";
}
```

این دستورات، خواص کنترل‌های روی فرم را مقداردهی می‌کنند.

۸. رویداد KeyPress را برای کنترل textBox1 اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (radioButton1.Checked == true)    number(sender, e);
    else if (radioButton2.Checked == true) floatNumber(sender, e);
    else if (radioButton3.Checked == true)    letter (sender, e);
}
```

این دستورات، اگر کنترل radioButton1 انتخاب شده باشد، متد number را فراخوانی می‌کند تا کاربر بتواند فقط ارقام را وارد نماید. وگرنه، اگر کاربر radioButton2 را انتخاب کرده باشد، متد floatNumber را فراخوانی می‌کند تا کاربر بتواند اعداد اعشاری را وارد کند. وگرنه، اگر کاربر radioButton3 را انتخاب کرده باشد، letter را فراخوانی می‌کند تا کاربر فقط بتواند حروف الفبا را وارد نماید.

۹. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

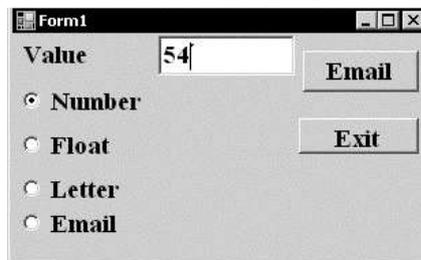
```
private void button1_Click(object sender, EventArgs e)
{
    if (TestEmailRegex(textBox1.Text) == true)
        MessageBox.Show("Email correct");
    else MessageBox.Show("Email not correct");
}
```

این دستورات با فراخوانی متد TextEmailRegex تست می‌کند، آیا پست الکترونیکی صحیح است یا خیر.

۱۰. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تایپ کنید:

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

۱۱. پروژه را ذخیره و اجرا کنید. گزینه Number را انتخاب کرده و مقدار را جلوی Value وارد نمایید. سعی کنید، کاراکترهای غیر عددی وارد کنید. همان طور که می‌بینید، از ورود مقادیر غیر عددی جلوگیری می‌کند. گزینه‌های دیگر را امتحان کنید (خروجی زیر):



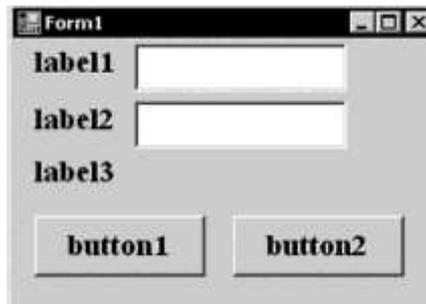
مثال ۷-۲. برنامه‌ای که x و n را خوانده، حاصل زیر را محاسبه نموده نمایش می‌دهد:

ساختار تصمیم و حلقه تکرار ۴۵

$$s = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \pm \frac{x^n}{n!}$$

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام Sum ایجاد کنید.
۲. سه کنترل Label، دو کنترل TextBox و دو کنترل Button به فرم اضافه کنید.



۳. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Load آن را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = "n?";
    label2.Text = "x?";
    label3.ResetText();
    button1.Text = "Sum";
    button2.Text = "Exit";
}
```

۴. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt16(textBox1.Text);
    double x = Convert.ToDouble(textBox2.Text);
    if (n >= 1)
    {
        double sum = 0;
        double fact = 1;
        double pow = 1.0;
        int sign = -1;
        for (int i = 3; i <= n; i += 2)
        {
            pow = pow * x * x;
            fact = fact * i * (i - 1);
            sum = sum + sign * pow / fact;
            sign = -sign;
        }
        label3.Text = sum.ToString();
    }
}
```

این دستورات، مجموع سری بیان شده را محاسبه کرده نمایش می دهند. در این برنامه sign، علامت عدد است که یکی در میان مثبت و منفی خواهد شد. Pow، توان x، fact، فاکتوریل I و I شمارنده حلقه تکرار که دو تا دو تا اضافه می شود، است.

۵. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{

```

```
Close();
}
```

۶. پروژه را ذخیره و اجرا کنید. جلوی n ? و x ? به ترتیب ۷ و ۹ را وارد کرده دکمه Sum را کلیک نمایید تا خروجی زیر را ببینید:

مثال ۸-۲. برنامه‌ای که n را از ورودی خوانده و مقدار سری زیر (π) را حداکثر تا n جمله نمایش دهد:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام PI ایجاد کنید.
۲. دو کنترل Label، به فرم اضافه کرده، خاصیت Text اولی را n تعیین کنید.
۳. یک کنترل TextBox و دو کنترل Button به فرم اضافه کرده، خاصیت Text دکمه‌ها را Pi و Exit تعیین نمایید.

۴. دکمه Pi را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt32(textBox1.Text);
    double pi = 4;
    int sign = -1;
    int j = 3;
    for (int i = 2; i <= n; i++)
    {
        pi = pi + sign * 4.0 / j;
        j += 2;
        sign = -sign;
    }
    label2.Text = "Pi is " + pi.ToString();
}
```

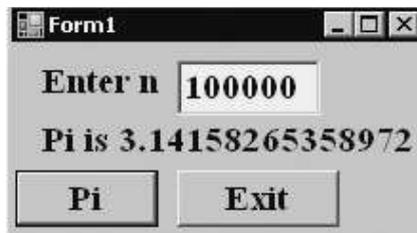
ساختار تصمیم و حلقه تکرار ۴۷

این دستورات، ابتدا n را خوانده در یک حلقه `for` سری را محاسبه می‌کند. چون π در هنگام تعریف برابر ۴ است، شمارنده حلقه `for` از ۲ شروع می‌شود (`sign`)، علامت است که یکی در میان مثبت و منفی می‌گردد).

۵. دکمه `Exit` را کلیک مضاعف کرده، دستورات رویداد `Click` آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

۶. پروژه را ذخیره و اجرا کنید. جلوی `n` `Enter` مقدار ۱۰۰۰۰۰ را وارد کرده، دکمه `Pi` را کلیک نمایید تا خروجی زیر را مشاهده کنید:



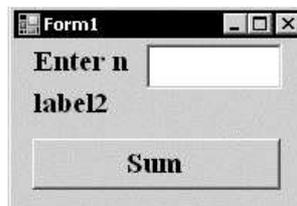
مثال ۹-۲. برنامه‌ای که مجموع سری زیر را نمایش می‌دهد (n را می‌خواند):

$$\frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام `Seri1` ایجاد کنید.

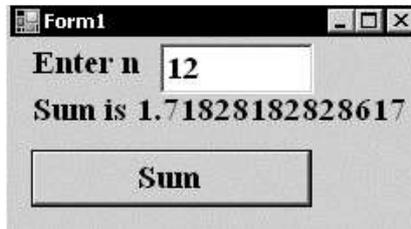
۲. دو کنترل `Label`، یک کنترل `TextBox` و یک کنترل `Button` به فرم اضافه کنید. خاصیت `Text` کنترل‌های `label1` و `button1` را به ترتیب عبارت `Enter n` و `Sum` تعیین نمایید.



۳. دکمه `Sum` را کلیک مضاعف کرده، دستورات رویداد `Click` آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt16(textBox1.Text);
    double fact = 1, sum = 0;
    for (int i = 1; i <= n; i++)
    {
        fact = fact * i;
        sum = sum + 1 / fact;
    }
    label2.Text = "Sum is " + sum.ToString();
}
```

۴. پروژه را ذخیره و اجرا کنید. جلوی `n` `Enter` عدد ۱۲ را وارد کرده، دکمه `Sum` را کلیک نمایید تا خروجی زیر را ببینید



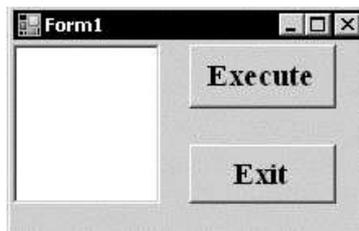
مثال ۱۰-۲. برنامه‌ای که کلیه اعداد سه‌رقمی را چاپ می‌کند که مجموع فاکتوریل ارقام آن برابر خودش باشد (مانند فرمول زیر):

$$n_3n_2n_1 = n_3! + n_2! + n_1!$$

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام SumFact ایجاد کنید.

۲. یک کنترل ListView و دو کنترل Button به فرم برنامه اضافه کنید. خاصیت Text کنترل‌های Button را به ترتیب Execute و Exit تعیین نمایید.



۳. متد fact را به صورت زیر تعریف کنید:

```
int fact(int n)
{
    if (n==0) return 1;
    else return(n * fact(n - 1));
}
```

این متد n را گرفته به طور بازگشتی فاکتوریل n را محاسبه می‌کند و برمی‌گرداند.

۴. دکمه Execute را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تایپ کنید:

```
private void button1_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();
    int n1,n2,n3,temp, i,sum;
    for (i = 100; i <= 999; i++)
    {
        n1 = i % 10;
        temp =( int) i / 10;
        n2 = temp % 10;
        temp /= 10;
        n3 = temp % 10;
        sum = fact(n1) + fact(n2) + fact(n3);
        if (sum == i) listView1.Items.Add(i.ToString());
    }
}
```

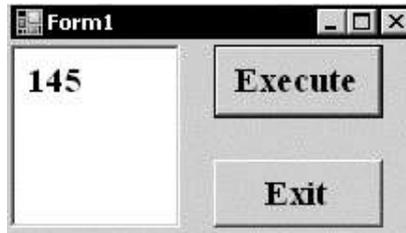
این دستورات، تمامی اعداد سه‌رقمی که مجموع فاکتوریل ارقام آن‌ها برابر خودشان باشد را نمایش می‌دهند.

۵. دکمه Exit را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تایپ کنید:

ساختار تصمیم و حلقه تکرار ۴۹

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

۶. پروژه را ذخیره و اجرا کنید. دکمه Execucte را کلیک نمایید تا خروجی زیر را مشاهده کنید:

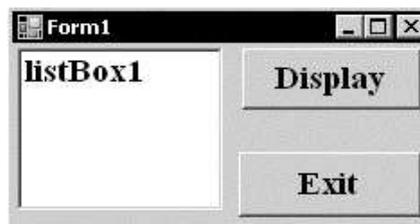


مثال ۱۱ - ۲. برنامه‌ای که کلیه اعداد چهاررقمی را چاپ می‌کند که مجموع رقم اول به توان ۱ و رقم چهارم به توان ۴ برابر با مجموع رقم دوم به توان ۲ و رقم سوم به توان ۳ است. به‌عنوان مثال، عدد $2141 = 2^4 + 1^1 = 1^3 + 4^2 = 17$ از جمله این اعداد می‌باشد.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام FourDigit ایجاد کنید.

۲. یک کنترل ListBox و دو کنترل Button به فرم اضافه کرده، خاصیت Text دکمه‌ها را به ترتیب Display و Exit تعیین کنید.



۳. متد pow را به‌صورت زیر تعریف کنید:

```
int pow(int x,int n)
{
    if (n==0) return 1;
    else return(x * pow(x,n-1));
}
```

این متد، x و n را گرفته به‌صورت بازگشتی x به توان n را برمی‌گرداند.

۴. دکمه Display را کلیک مضاعف کرده، دستورات رویداد Click آن را به‌صورت زیر تایپ کنید:

```
private void button1_Click(object sender, EventArgs e)
{
    int n1, n2, n3, n4, temp, i;
    for (i = 1000; i <= 9999; i++)
    {
        n1 = i % 10;
        temp = i / 10;
        n2 = temp % 10;
        temp /= 10;
        n3 = temp % 10;
        n4 = temp / 10;
        if( (n1 + pow(n4 , 4)) == (n2 * n2 + pow(n3, 3))) listBox1
        . Items.Add(i);
    }
}
```