
حل مسائل C#

(مرجع کامل)

تالیف

مهندس رمضان عباس نژادورزی



فن آوری نوین

سرشناسه	: عباس نژاد، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	: حل مسائل C# (مرجع کامل) / رمضان عباس نژادورزی.
مشخصات نشر	: بابل: فناوری نوین، ۱۳۸۸.
مشخصات ظاهری	: ۲۰۸ص: مصور، جدول.
شابک	: ۶۲۰۰۰ ریال: ۹۷۸۶۰۰۹۱۴۱۳۴۰
وضعیت فهرست نویسی	: فیپا
موضوع	: سی شارپ (زبان برنامه‌نویسی کامپیوتر)
موضوع	: سی شارپ (زبان برنامه‌نویسی کامپیوتر) -- مسائل، تمرین‌ها و غیره
رده بندی کنگره	: ۱۳۸۸ س ۹۵ع ۲۳/۷۶/۷۳ QA۷۳
رده بندی دیویی	: ۱۳۳/۰۰۵
شماره کتابشناسی ملی	: ۱۹۵۸۱۹۶



فن آوری نوین

www.fanavarinovin.net

بابل، کدپستی ۷۳۴۴۸-۴۷۱۶۷

تلفن: ۰۱۱۱-۲۲۵۶۶۸۷

حل مسائل C# (مرجع کامل)

تألیف: مهندس رمضان عباس نژادورزی

ناشر: فن آوری نوین

چاپ اول: زمستان ۱۳۸۸

تیراژ: ۱۰۰۰ جلد

طراح جلد: احمد فرجی

شابک: ۰-۴-۹۱۴۱۳-۶۰۰-۹۷۸

حروفچینی و صفحه‌آرایی: فن آوری نوین

قیمت: ۶۲۰۰ تومان

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

فهرست مطالب

۷	فصل اول: آشنایی با زبان C#
۱۵	فصل دوم: ساختار تصمیم و حلقه تکرار
۵۲	فصل سوم: آرایه‌ها و رشته‌ها
۹۱	فصل چهارم: کلاس‌ها و اشیا
۱۴۵	فصل پنجم: مباحث پیشرفته در C#
۲۰۵	پیوست: پروژه‌های حل شده در سایت
۲۰۷	منابع

مقدمه

زبان C# در فناوری دات نت (NET)، توسط شرکت مایکروسافت ارائه شده است که کاملاً شیء‌گرا است. امروزه، اکثر دانشجویان رشته کامپیوتر و فناوری اطلاعات با این زبان آشنایی دارند. زیرا، این زبان از محبوب‌ترین زبان‌های خانواده C می‌باشد. از طرف دیگر، زبان C# به عنوان یکی از مهم‌ترین زبان‌های رشته کامپیوتر، فناوری اطلاعات، ICT، علوم کامپیوتر و رشته‌های دیگر تحت عنوان درس زبان برنامه‌نویسی پیشرفته تدریس می‌شود. یکی از روش‌های موفق یادگیری زبان‌های برنامه‌نویسی، حل مسائل گوناگون می‌باشد.

در حال حاضر کتاب‌های متعددی برای زبان برنامه‌نویسی C# ارائه شده است که جای تقدیر و تشکر دارد. اما، این کتاب با بهره‌گیری از سال‌ها تجربه در زمینه تدریس، تالیف، برنامه‌نویسی و مدیریت پروژه‌های نرم‌افزاری تدوین شده است. از ویژگی‌های بارز این کتاب، بیان مثال‌های کاربردی و حل گام به گام به همراه توضیحات دقیق آن‌ها است.

مثال‌هایی که در کتاب آورده شده‌اند، همگی دارای کاربرد و هدف خاصی در برنامه‌نویسی هستند و برخی از آن‌ها را می‌توانید در پروژه‌های کاربردی و بزرگ استفاده کنید.

در این کتاب به آموزش بانک اطلاعاتی نپرداختیم. زیرا، مطالب آن گسترده است و بررسی برنامه‌نویسی بانک اطلاعاتی در این کتاب نمی‌گنجد. بنابراین، اگر می‌خواهید از زبان C# برای برنامه‌نویسی بانک اطلاعاتی استفاده کنید، به کتاب آموزش گام به گام برنامه‌نویسی بانک اطلاعاتی با C# نوشته همین مولف از انتشارات فن‌آوری نوین مراجعه نمایید.

امیدوارم این کتاب مورد قبول اساتید، دانشجویان و خوانندگان محترم واقع شود.

کتاب شامل برنامه‌هایی است که کد آن‌ها را می‌توانید از سایت انتشارات فن‌آوری نوین به آدرس www.fanavarinovin.net بگیرید.

رمضان عباس نژاد ورزی

fanavarienovin@yahoo.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 مکان‌نمای ماوس از محدوده فرم خارج شود، رنگ زمینه فرم آبی می‌گردد و اگر مکان‌نما 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.Maximized 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 می‌دهند.

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 می‌کنند.

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 می‌کنند.

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 می‌دهند.

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 می‌برید، به تغییر رنگ زمینه فرم دقت کنید.

مثال ۲ - ۱ برنامه‌ای که اعمال زیر را انجام می‌دهد:

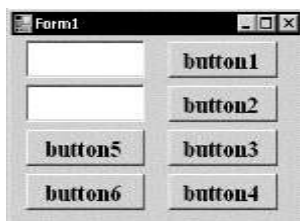
- ❖ عبارت " فرم اصلی " را در سمت راست عنوان فرم نمایش می‌دهد.
- ❖ با کلیک روی فرم، عنوان فرم به سمت چپ برده و با کلیک دیگر عنوان به سمت راست برود و این روند ادامه یابد.

- ❖ دارای دو کنترل TextBox و شش کنترل Button باشد.
- ❖ با کلیک button1، اطلاعات textBox2 به textBox1 اضافه می‌شود.
- ❖ با کلیک دکمه button2، اطلاعات textBox1 به textBox2 اضافه

می‌گردد.

- ❖ با کلیک دکمه button3، اطلاعات انتخاب شده کنترل textBox2 حذف می‌گردد.

- ❖ با کلیک دکمه button4، رنگ زمینه فرم به سبز تغییر یابد.



۹ آشنایی با زبان C#

❖ 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 می‌دهند.

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 می‌دهند.

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 می‌کنند. برای این منظور از متد FromFile کلاس Image استفاده می‌کند.

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 می‌دهند.

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# ۱۱

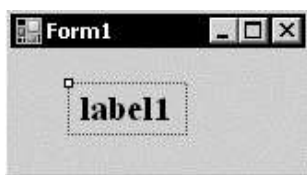
در `textBox2` مقدار `alu` را انتخاب کرده و به ترتیب دکمه‌های `button3` و `button6` را کلیک کنید و سپس بر روی فرم کلیک نمایید تا خروجی زیر ظاهر شود:



همان طور که در این خروجی می‌بینید، مقدار `alu` در `textBox2` حذف گردید، در زمینه 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 مکان‌نما فرم را ترک کند، رنگ زمینه `label` سفید و رنگ زمینه فرم آبی شود.
۴. اگر مکان‌نما وارد `label` شود، رنگ زمینه فرم قرمز و رنگ نوشته `label` 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 مکان‌نما 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 مکان‌نما 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.

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

آشنایی با زبان C# ۱۳

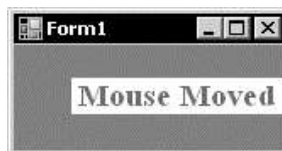
```
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    this.BackColor = Color.Green;
    label1.Text = "Mouse Moved";
}
```

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

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. مکان‌نما را روی فرم حرکت دهید تا شکل زیر را ببینید:



اندازه فرم را تغییر دهید تا خروجی به شکل مقابل در آید.

مثال ۴ - ۱ برنامه‌ای که اعمال زیر را انجام می‌دهد:

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

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:

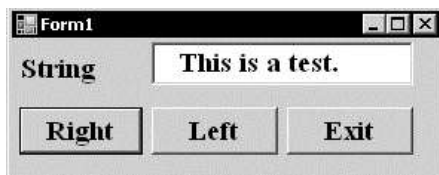
```
private void button2_Click(object sender, EventArgs e)
{
    textBox1.Text = (textBox1.Text.Length > 0 )?
        textBox1.Text.Substring(1, textBox1.Text.Length -
        1):textBox1.Text ;
}
```

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

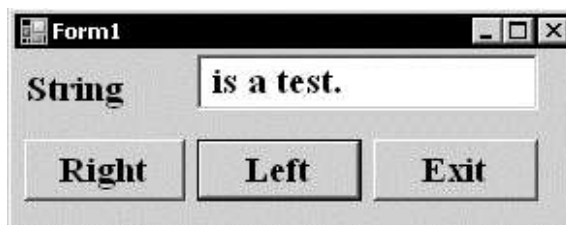


۷. پروژه را ذخیره و اجرا کنید. عبارت This is a test.

را تایپ کرده سه بار دکمه Right را کلیک کنید تا شکل

مقابل ظاهر شود.

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



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:

مالیات - بیمه - حقوق ناخالص = حقوق خالص

- ❖ 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 انتخاب نگردد 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 حقوق‌اش بیش از ۴۵۰۰۰۰۰ 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 کارکترهای 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:

ساختار تصمیم و حلقه تکرار ۱۷

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.

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 روبه‌رو را ببینید.

مثال ۳ - ۲ برنامه‌ای که حقوق ناخالص را گرفته و براساس قوانین زیر، بیمه و مالیات را کسر کرده،

حق تاهل به او تخصیص دهد:

- ❖ اگر نوع بیمه تامین اجتماعی باشد، ۷ درصد حقوق ناخالص بیمه کسر گردد.
- ❖ اگر نوع بیمه خدمات درمانی باشد، ۹ درصد حقوق ناخالص بیمه باشد.
- ❖ اگر نوع بیمه نیروی انتظامی باشد، ۱۰ درصد حقوق ناخالص بیمه می‌باشد.
- ❖ اگر کارمند جانباز باشد، از مالیات معاف است.
- ❖ اگر کارمند منطقه نیمه توسعه یافته باشد، ۵ درصد مازاد بر ۴۶۰۰۰۰۰ ریال مالیات کسر می‌شود.

ساختار تصمیم و حلقه تکرار ۱۹

❖ اگر کارمند عادی باشد، ۱۰ درصد مازاد بر ۴۶۰۰۰۰۰ ریال مالیات کسر می‌شود.

❖ اگر کارمند متاهل باشد، ۴۶۰۰۰۰۰ ریال حق تاهل به او تعلق می‌گیرد.

سپس با استفاده از فرمول زیر حقوق خالص را حساب کرده، نمایش می‌دهد:

مالیات - بیمه - حق تاهل + 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 متاهل است یا خیر.

مراحل طراحی و اجرا

۱. پروژه جدیدی به نام **Pay** ایجاد کنید. خاصیت **RightToLeft** فرم را **Yes** تعیین کنید.

۲. دو کنترل **Label**، دو کنترل **Button** و یک کنترل **TextBox** به فرم اضافه کرده، خاصیت **Text** کنترل‌های **Label** را به ترتیب **حقوق ناخالص** و **خالص** تعیین کنید. خاصیت **Text** 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 **متاهل** تعیین کنید.

۵. سه کنترل **RadioButton** به کنترل **groupBox1** (نوع بیمه) اضافه کرده، خاصیت **Text** آن‌ها را به ترتیب **تامین اجتماعی**، **خدمات درمانی** و **نیروی انتظامی** تعیین کنید.

۶. سه کنترل **RadioButton** دیگر به کنترل **نوع مالیات** اضافه کرده، خاصیت **Text** آن‌ها را به ترتیب **جانباز**، **نیمه توسعه یافته** و **عادی** تعیین کنید.



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 " + " " + " " + " بیمه " +
        tax.ToString() + "\n";
    label2.Text += " تاهل " + wife.ToString() + " " + " " + " پرداختی "
        + pay.ToString() ;
    }
    
```

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

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، گزینه‌های متاهل، تامین 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 \uparrow ، \downarrow ، End، Home 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.