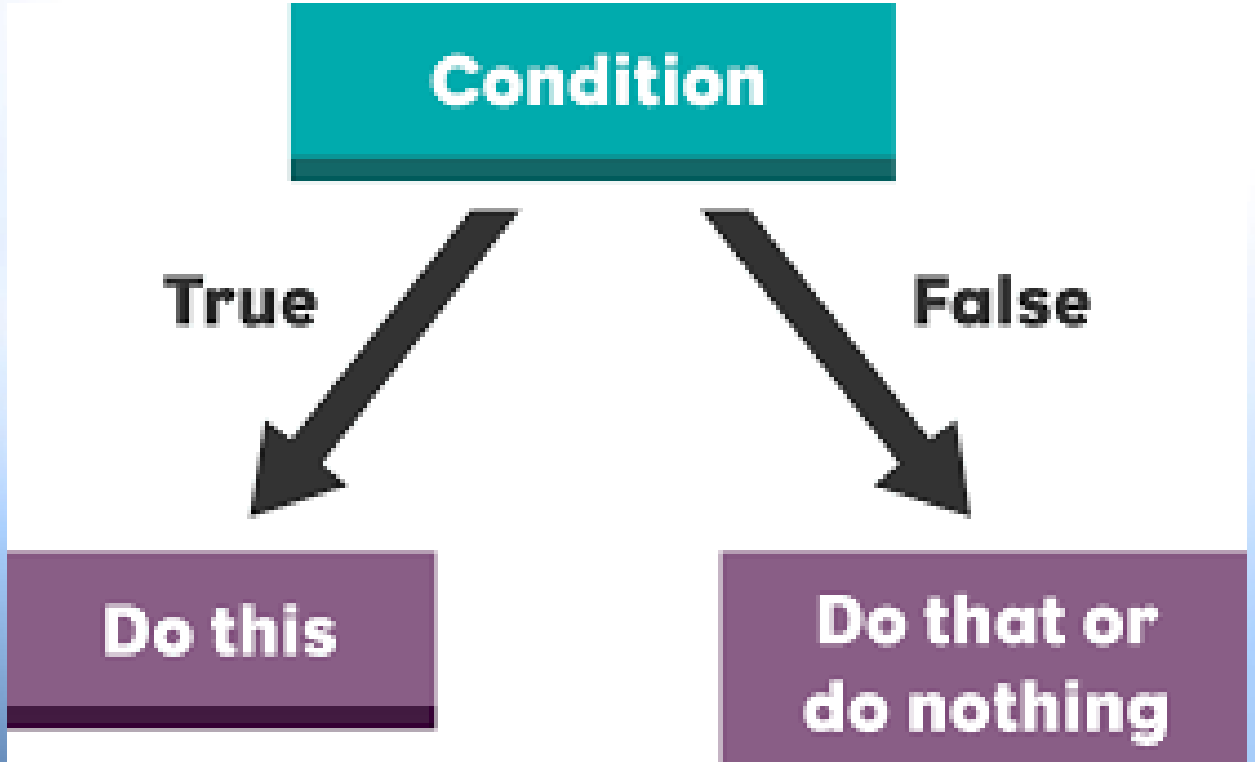
فصل دوم:
عبارت ها، عملگرها و دستورات شرطی



اصول برنامه نویسی با C#

برای برنامه نویسی سی شارپ باید بدانید که :

۱- بین حروف کوچک وبزرگ تفاوت وجود دارد .

۲- بعد از هر دستور العمل علامت ; (سمیکالن) قرار می گیرد .

۳- هر بلاک از دستورات داخل یک جفت آکولاد {}نوشته می شود .

۴- همه ی دستورات داخل توابع نوشته می شود .

۵- سی شارپ شی گراست و همه توابع داخل کلاس ها نوشته می شوند.

۶- متغیر ها قبل از استفاده باید تعریف شده و مقدار دهی شوند.

متغیرها و شیوه تعریف و استفاده از آنها

{data-type} {variable-name} = {variable-value};

data-typeنوع متغیری که میخواهیم تعریف کنیم را مشخص می کند. متغیرها می توانند رشته ای، عددی یا ... باشند.

variable-nameنام متغیر را مشخص می کند، نام متغیرها می توانند ترکیبی از حروف و عدد و کاراکتر «__» باشند، و نام متغیرها نمی توانند با عدد شروع شوند. برای مثال نام های number1 و first__name و a__12برای نامگذاری صحیح بوده.

variable-valueمقدار اولیه ای که داخل متغیر قرار خواهد گرفت را مشخص می کند

string website =" Tadris";

دقت کنید که کلیه رشته باید بین علامت «"» قرار بگیرند

نوشتن توضیحات یا Comment برای کدها

یادداشت قطعه ای از کد است که توسط کامپایلر، کامپایل نمی شود و با این ترتیب اجرا نیز نخواهد شد.
توضیحات بعد از علامت // یا بین علامت های /* و */ قرار می گیرند.

```
//یادداشت تک خطی

/*
یادداشت چند خطی
*/
```

انواع داده عددی

در زبان سی شارپ، چندین نوع داده عددی داریم:

نوع داده byte: در این نوع داده می توان از بازه ۰ تا ۲۵۵ را ذخیره کرد.

نوع داده sbyte: در این نوع داده می توان از بازه ۱۲۸- تا ۱۲۷ را ذخیره کرد.

نوع داده short: در این نوع داده می توان از بازه ۳۲،۷۶۸- تا ۳۲،۷۶۷ را ذخیره کرد.

نوع داده ushort: در این نوع داده می توان از بازه ۰ تا ۶۵،۵۳۵ را ذخیره کرد.

نوع داده int: در این نوع داده می توان از بازه ۲،۱۴۷،۴۸۳،۶۴۸- تا ۲،۱۴۷،۴۸۳،۶۴۷ را ذخیره کرد.

نوع داده uint: در این نوع داده می توان از بازه ۰ تا ۴،۲۹۴،۹۶۷،۲۹۵ را ذخیره کرد.

نوع داده long: در این نوع داده می توان از بازه ۹،۲۲۳،۳۷۲،۰۳۶،۸۵۴،۷۷۵،۸۰۸- تا ۹،۲۲۳،۳۷۲،۰۳۶،۸۵۴،۷۷۵،۸۰۷ را ذخیره کرد.

نوع داده ulong: در این نوع داده می توان از بازه ۰ تا ۱۸،۴۴۶،۷۴۴،۰۷۳،۷۰۹،۵۵۱،۶۱۵ را ذخیره کرد.

نکته :علامت s در کنار byte یعنی نوع دادهsbyte ، مخفف signed یا دارای علامت منفی است که نشان دهنده بازه اعداد منفی می باشد.

نکته :علامت u در کنار نوع هایی مانند short و int وlong ، مخفف unsigned یا بدون علامت منفی است که این نوع های داده اعداد بزرگتر از صفر را قبول می کنند.

float این نوع داده‌ی عددی فضایی معادل ۳۲ بیت را اشغال می کند.

double این نوع داده‌ی عددی فضایی معادل ۶۴ بیت را اشغال می کند.

decimal این نوع داده‌ی عددی فضایی معادل ۱۲۸ بیت را اشغال می کند.

توجه: داده‌های عددی float و double معمولا برای اندازه گیری مقادیری که دقت در آنها معیار نیست مورد استفاده قرار می‌گیرند. مثلا فاصله، مسافت و ...

اما داده‌ی عددی decimal برای حالتی که دقت عددی مدنظر می‌باشد بکار گرفته خواهد شد مثل واحد پول، محاسبات حسابداری و ...

نوع داده رشته ای

نوع داده رشته ای برای ذخیره یک رشته یا متن مورد استفاده قرار می گیرد:

```
string firstName = "Hossein";
```

نوع داده کاراکتر

برای ذخیره یک کاراکتر در زبان سی شارپ از نوع داده char استفاده می کنیم. یک کاراکتر در بین «'»قرار می گیرد.

```
char chr1 = 'A';
```

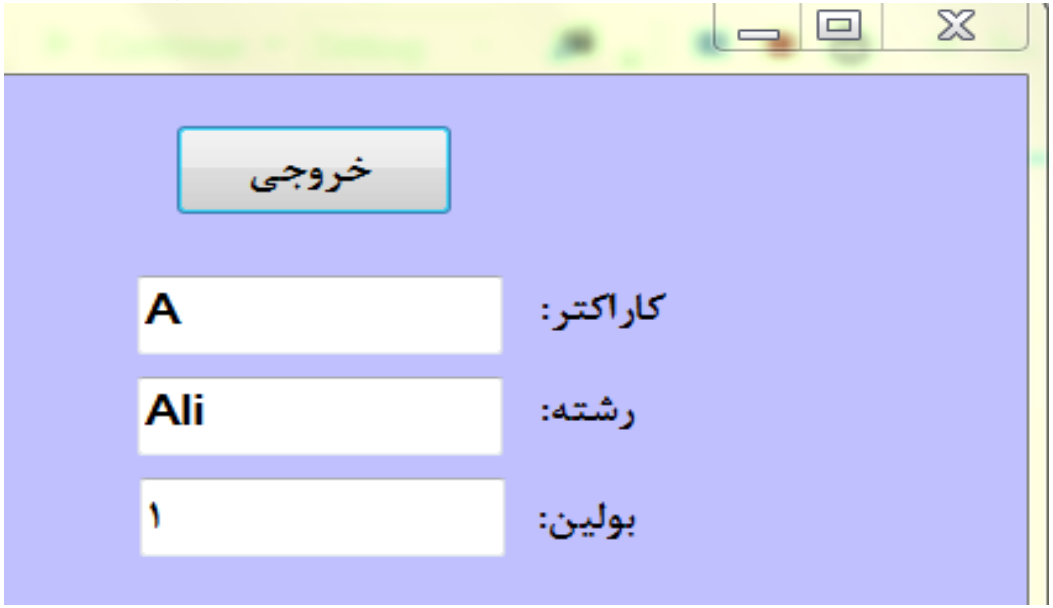
نوع داده منطقی

```
bool var2 = true;
```

```
private void button1_Click(object sender, EventArgs e)
    {
        char ch = 'A';
        textBox1.Text = ch.ToString();

        string str = "Ali";
        textBox2.Text = str;

        bool f = false;
        if (f)
        {
            textBox3.Text = 0.ToString();}
            else
        {
            textBox3.Text=1.ToString();
```



Object

نوع داده ایست که در آن هر نوع مقداری چه رشته ای، چه عددی و چه منطقی قابل ذخیره است:

```
object intObj = 123;
object stringObj = “ahemmati.ir";
```

برای تبدیل رشته به int می توانیم از متد int.Parse استفاده کنیم. این متد یک ورودی از نوع رشته گرفته و آن را تبدیل به عدد می کند. مثال:

```
int num1 = int.Parse("12");
```

var

بعضی وقت ها انتخاب نوع داده را بر اساس مقدار می خواهیم بر عهده زبان کامپایلر زبان سی شارپ بگذاریم. برای این کار، از کلمه کلیدی var استفاده می کنیم
مثال:

```
var num = 1223;
```

متغیرهایی که با کلمه کلیدی var تعریف می شوند حتماً باید مقدار اولیه داشته باشند.

بعد از تعریف متغیری با کلمه کلیدیvar ، نمی توان مقداری غیر از نوع اولیه در آن ریخت.

کلمه کلیدیConst

زمانی که متغیری را تعریف می کنید، در هر قسمت برنامه که به آن متغیر دسترسی دارید، می توانید مقدار آن را تغییر دهید.

اما فرض کنید می خواهید این مقدار ثابت بوده و قابل تغییر نباشد. در اینجا باید از کلمه کلیدی const که مخفف constant یا ثابت می باشد استفاده کنید. زمانی که متغیری با این کلمه کلیدی مشخص می شود مقدار آن تنها زمان تعریف متغیر قابل تعیین خواهد بود و در سایر قسمت ها امکان تغییر مقدار متغیر را نخواهید داشت

```
const int  const1= 12;
```

انواع عملگرها در C#

عملگرها در سی شارپ به دسته‌های مختلفی تقسیم می‌شوند:

۱) عملگرهای محاسباتی
۲) عملگرهای مقایسه‌ای
۳) عملگرهای منطقی
۴) عملگرهای انتسابی
۵) عملگرهای بیتی

باید توجه داشت که عملگرها در سی شارپ سه نوع هستند:

- عملگرهایی که به یک عملوند نیاز دارند (Unary).
- عملگرهایی که به دو عملوند نیاز دارند (Binary).
- عملگرهایی که به سه عملوند نیاز دارند (Ternary).

۱- عملگرهای محاسباتی

مثال	نوع	عملگر
Result = var1 + var2	Binary	+
Result = var1 - var2	Binary	-
Result = var1 * var2	Binary	*
Result = var1 / var2	Binary	/
Result = var1 % var2	Binary	%
Result = +var1	Unary	+
Result = -var1	Unary	-
Result = var1++	Unary	++
Result = var1--	Unary	--

مثال- عملگرهای محاسباتی

```
private void button2_Click(object sender, EventArgs e)
        {
            int a = 20, b = 10, c, d;
            c = a + b;
            textBox1.Text = c.ToString();

            d=a/b;
            textBox2.Text = d.ToString();

            a++;
            textBox3.Text =a.ToString();
             b--;
            textBox4.Text = b.ToString();
```

تفاوت عملکرد a++ و ++a

```
private void Form1_Load(object sender, EventArgs e)
        {
          int a = 10, b=10;
          textBox1.Text = (a++).ToString();


          textBox2.Text = (++b).ToString();
}
```

۲- عملگرهای مقایسه‌ای

از این دسته از عملگرها در دستورات شرطی بیشتر استفاده می‌شود. حاصل همه‌ی عبارت‌ها از نوع بولی است.

عملگرها	نام	مثال
>	بزرگتر	x>y
>=	بزرگتریا مساوی	x>=y
<	کوچکتر	X<y
<=	کوچکتر یا مساوی	X<=y
= =	متساوی	X= =y
!=	نا مساوی	X!=y

۳- عملگرهای منطقی

عملگر	نام	مثال
!	نقیض(not)	!x
&&	و(and)	x>y && m<p
||	یا(or)	x>y || m<p

۴- عملگرهای انتسابی

این نوع از عملگرها مقدار متغیر سمت راست خود را در متغیر سمت چپ قرار می‌دهند.

عملگر	نام	مثال
+=	انتسا ب جمع	X+=y
-=	انتسا ب تفریق	x-=y
*=	انتساب ضرب	X*=y
/=	انتساب تقسیم	x/=y
%=	انتساب باقیمانده	X%=y

عملگرهای بیتی

این عملگرها به ما اجازه‌ی دستکاری داده متغیرها در سطح بیت را می‌دهند.

عملگر ~: کار مکمل یک در سیستم باینری را انجام می دهد.

عملگر >>: عملیات شیفت به چپ را انجام می دهد.

عملگر <<: عملیات شیفت به راست را انجام می دهد.

عملگر ^: نام دیگر این عملگر XORمی باشد و بیت خروجی در صورتی ۱ خواهد بود که بیت های معادل با یکدیگر تفاوت داشته باشند:

```
int num1 = 2;
int num2 = 6;
int result = num1 ^ num2;
```

```
2 = 0 1 0
6 = 1 1 0
^ = 1 0 0
```

عملگرهای بیتی

عملگر |: نام دیگر این عملگر ORمی باشد و خروجی آن، در صورتی که یکی از بیت ها معادل ۱ باشد، ۱ خواهد بود، در غیر اینصورت ۰ خواهد شد.

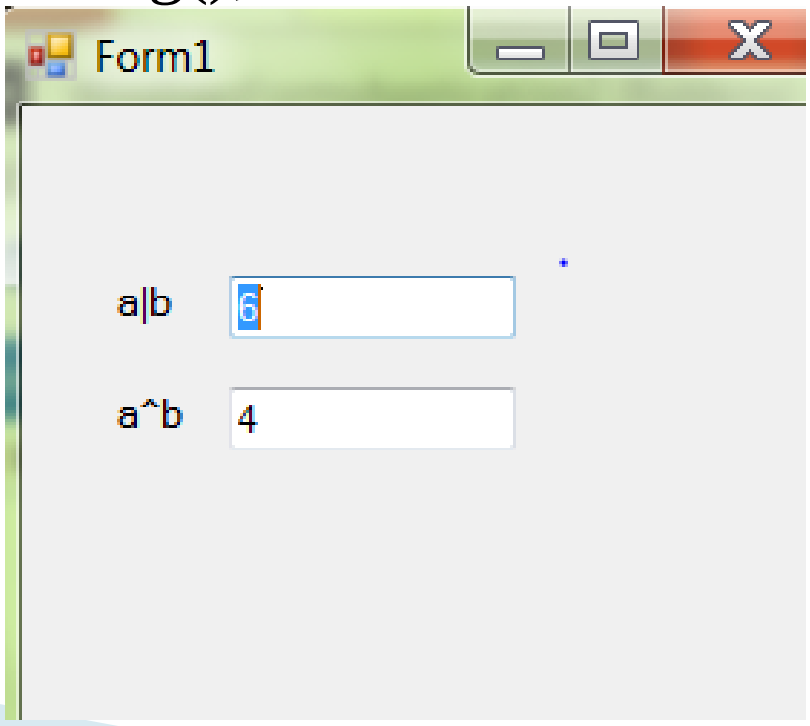
```
int num1 = 2;                2 = 0 1 0
int num2 = 6;                6 = 1 1 0
int result = num1 | num2;    | = 1 1 0
```

عملگر &: در صورتی که هر دو بیت ۱ باشند، خروجی ۱ خواهد بود. در غیر اینصورت خروجی ۰ خواهد شد. نام این عملگر AND می باشد.

```
int num1 = 2;                    2 = 0 1 0
int num2 = 6;                    6 = 1 1 0
int result = num1 &amp; num2;    &amp; = 0 1 0
```

مثال- عملگرهای بیتی

```
private void Form1_Load(object sender, EventArgs e)
        {
            int a = 2, b=6;
            textBox1.Text = (a | b).ToString();
            textBox2.Text =( a ^ b).ToString();
          }
```



تقدم عملگرها

طبق شکل زیر حق تقدم عملگرها از بالا به پایین در هر سطر مشخص شده و در محاسباتی که بیش از دو عملوند موجود هست، حق تقدم عملگرها مشخص می‌کند ابتدا کدام عملگر اثر کند.

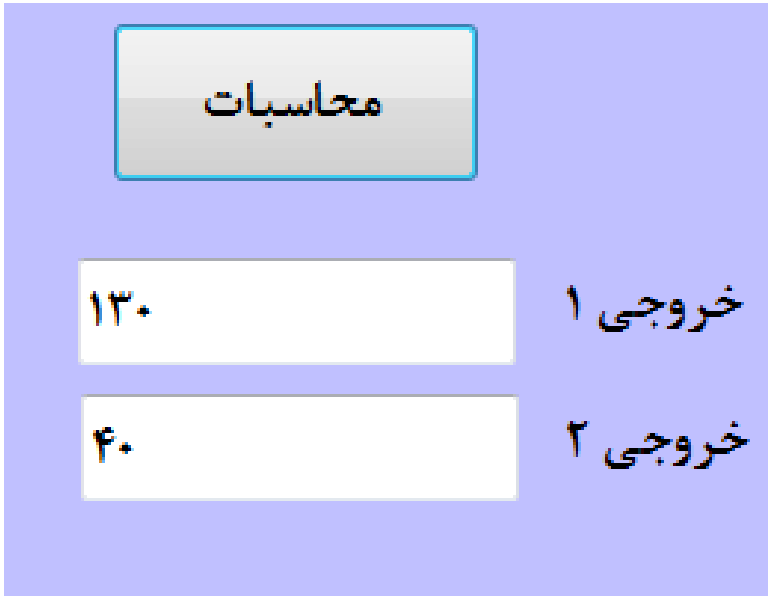
عملگرها
++ , -- (قبل از متغیر)
*, / , %
+ , -
>> , <<
<,> , <= ,>=
== , !=
&
^
|
&&
||
= , *= , /= , %= , += , -=
++ ,-- (بعد از متغیر)

نکته:

در یک عبارت ،عملگر داخل پرانتر بالاترین الویت نسبت به بقیه عملگرها دارد.

مثال – تقدم عملگرها

```
private void button1_Click(object sender, EventArgs e)
    {
        int a = 10, b = 3, c, v;
        c = a + b * 10;
        v = (a + b) * 10;
        textBox2.Text = c.ToString();
        textBox1.Text = v.ToString();
    }
```



دستورات کنترلی -دستور if

خیلی از مواقع، زمانی که برنامه ای را می نویسیم نیاز داریم تا بر اساس شرایط خاص و یا ورودی های کاربر، روند اجرای برنامه را تغییر دهیم.

بوسیله دستور if قادر خواهیم بود بر اساس شرط های مختلف کدها را اجرا کنیم.

```
if(condition)
{
   // statements
}
```

دستورات کنترلی -دستور if- else

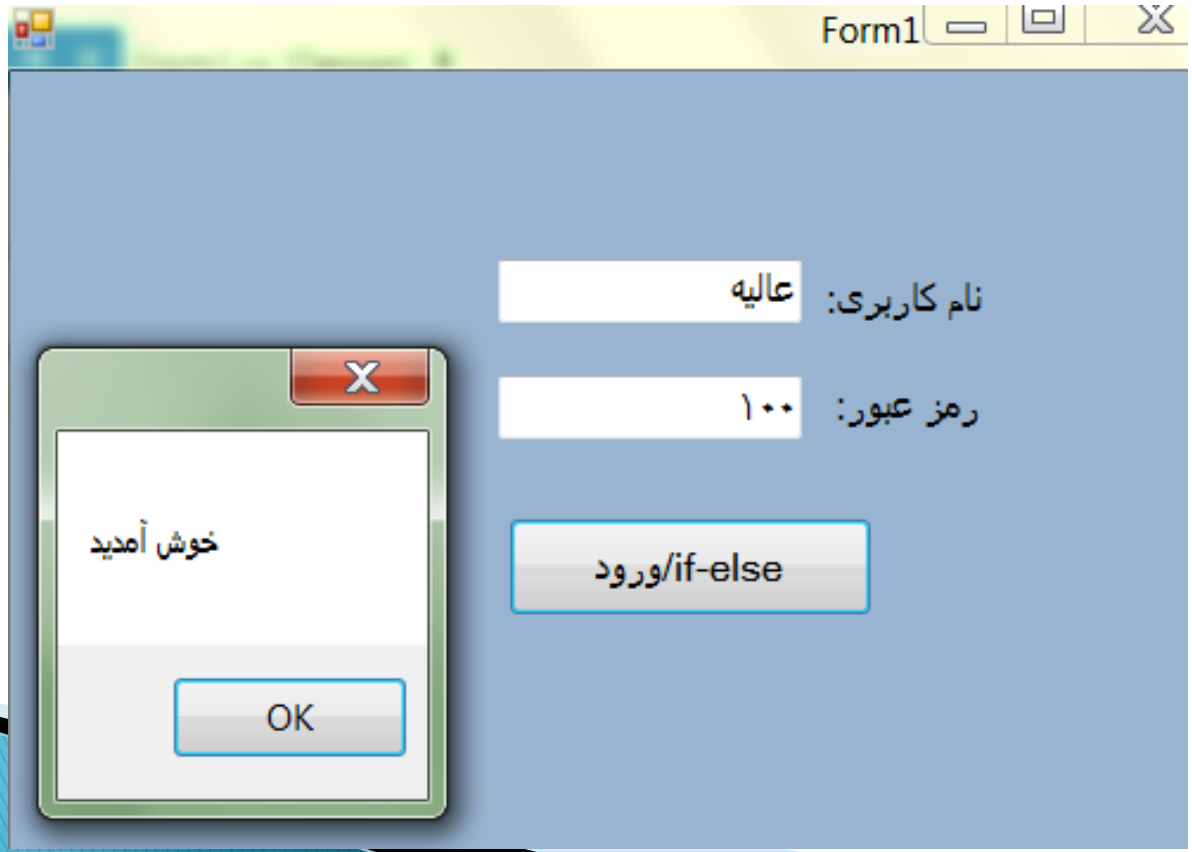
فرض کنید شرطی را مشخص کردیم، می خواهیم در صورت برقرار نبودن شرط، قطعه کد دیگری اجرا شود. در اینجا دستور else به کمک ما می آید. دستور else کدی که در صورت برقرار نبودن شرط باید اجرا شود را مشخص می کند. ساختار کلی else به صورت زیر است:

```
if(condition)
{
        // if body
}
else
{
        //else body
}
```

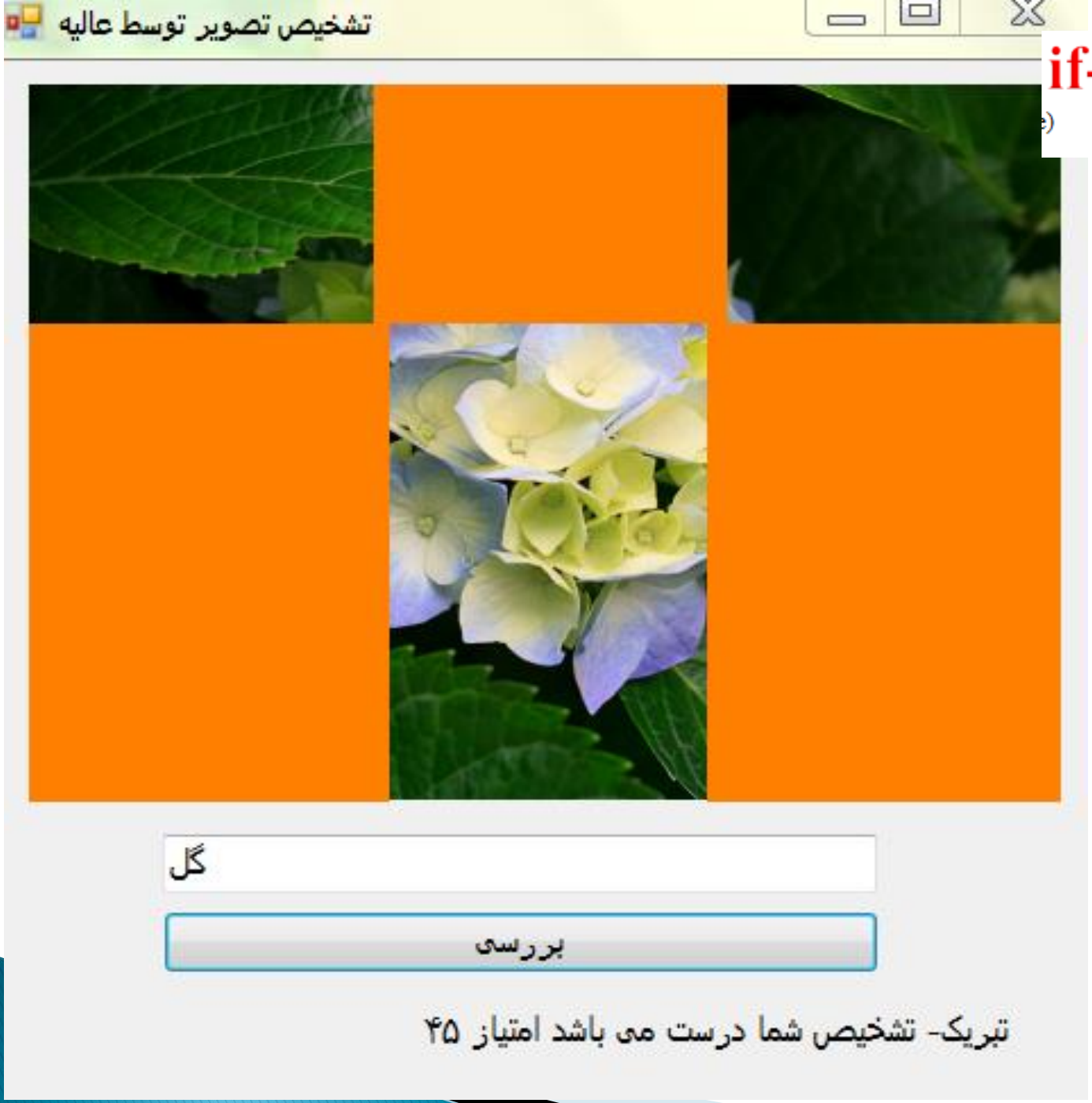
مثال۱ -دستور if-else

قصد داریم برنامه ای بنویسیم که از کاربر یک کلمه عبور و یک نام کاربری را دریافت کرده و در صورت صحیح بودن اطلاعات وارد شده، پیغام مناسبی را به کاربر نمایش دهد. در صورت صحیح نبودن نام کاربری و کلمه عبور نیز پیغام دیگری نمایش داده خواهد شد.

```
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "عالیه" && textBox2.Text == "100")
    {
        MessageBox.Show("خوش آمدید");
    }
    else
        MessageBox.Show("نام کاربری یا رمز عبور نادرست می باشد");
}
```



مثال ۲-دستور if-else
برنامه تشخیص تصویر



1. Picturebox:select resource/size mode
2. 9Lable:text/Autosize:false/back color
3. Textbox
4. Button: text: بررسی
5. lable

مثال ۲–دستور if-else
برنامه تشخیص تصویر

```
byte score = 70;
    private void button1_Click(object sender, EventArgs e)
    {
       if (textBox1.Text == "(" گل
       {
          label10.Text = " + " امتیاز " + " " + "تبریک- تشخیص شما درست می باشد"score.ToString();
       }
       else
          label10.Text = "  + " امتیاز " + " " + "تشخیص شما نادرست می باشد"score.ToString();

    }

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

    private void label7_Click(object sender, EventArgs e)
    {
       score -= 5;
       label7.Visible = false;
    }
```

برای بقیه lable ها هم به همین صورت کد نویسی می شود

دستور if - else if

بوسیله دستور else if قابلیت تعیین چند شرط متوالی را خواهیم داشت.
اگر بخواهیم این دستور را به زبان عامیانه ترجمه کنیم میگوییم:

اگر شرط ۱ برقرار بود => دستورات اول

در غیر اینصورت اگر شرط ۲ برقرار بود => دستورات دوم

در غیر اینصورت اگر شرط n برقرار بود => دستورات n ام

در غیر این صورت => سایر دستورات

ساختار کلی این دستور به صورت زیر است:

```
if(condition1)
{
   // if body
}
else if(condition2)
{
   // if body
```

```
}
else if(conditionN)
{
    // if body
}
else
{
    // if body
 }
```

دستور if - else if

```
private void button1_Click(object sender, EventArgs e)
    {
        int a = Convert.ToInt32(textBox1.Text);
        if (a == 0)
            MessageBox.Show("0");
        else if (a>0)
            MessageBox.Show("+");
        else if(a<0)
            MessageBox.Show("-");
        else
            MessageBox.Show(";("ورودی نادرست می باشد
    }
```

دستور switch

دستور switch بر اساس یک مقدار حالت های مختلف را بررسی کرده و دستورات قسمت مربوطه را اجرا خواهد کرد.

```
private void button2_Click(object sender, EventArgs e)
  { string name = textBox2.Text;
          switch (name)
    {   case "Aalia":
MessageBox.Show(" ;("خوش آمدید
break;
case "ali":
MessageBox.Show(";("سلام آقای حسینی
break;
 default :
          MessageBox.Show(";("اطلاعات ورودی نادرست می باشد
          break;
```

فعالیت کلاسی۱

۱-برنامه ای مشابه شکل زیر طراحی کنید که دو عدد از کاربر گرفته و مجموع و تفاضل وخارج قسمت و باقیمانده آنها را محاسبه و نمایش دهد.

۲- یک ویرایشگر متنی ساده با امکانات زیر
الف- بازکردن فایل
ب- ذخیره کردن با نام جدید
ج- کپی و ذخیره(از متدهای آماده text box استفاده شود)
د- تغییر رنگ زمینه
طراحی کنید.

۳- برنامه بازی تشخیص مسیر صحیح را طراحی کنید.
۴- فرمی جهت loading (بارگذاری) نرم افزارطراحی کنید.