فصل چهارم

کار با متدها و کلاس‌های آماده

تاکنون با تعدادی از متدها و کلاس‌های سی شارپ کار کرده‌اید. این متدها و کلاس‌ها، از قبل آماده و درون ساختار سی شارپ تعبیه شده است که می‌توانیم از آنها در برنامه‌های خود استفاده کنیم و به این طریق برنامه خود را به راحتی و با قدرت بیشتر بنویسیم. کتابخانهNET Framework. به فراوانی دارای این کلاس‌ها و متدهاست. در این فصل برای نمونه به سه کلاس اشاره می‌کنیم که کاربردهای زیادی در برنامه‌های شما خواهد داشت.

پس از پایان این فصل انتظار می‌رود که فراگیر بتواند :

۱ــ اجزای خط عنوان یک متد را توضیح دهد.

۲ــ ورودی‌ها و خروجی‌های یک متد را نام ببرد.

۳ــ با کلاس math و متدهای پرکاربرد آن برنامه بنویسد.

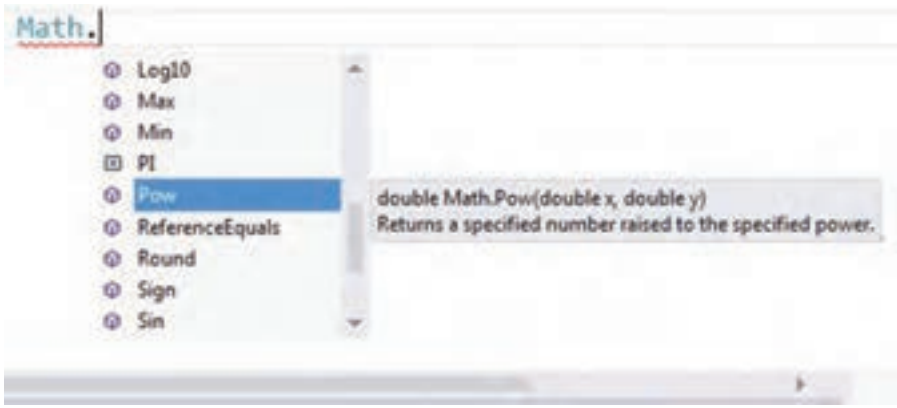
۴ــ از کلاس array و متدهای sort و search این کلاس در برنامه‌های خود استفاده کند.

۵ــ کاربرد کلاس string را بیان نماید و با استفاده از متدهای این کلاس برنامه بنویسد.

۱ــ۴ــ کلاس Math

برای انجام محاسبات ریاضی علاوه بر چهار عمل اصلی، نیاز به عملگرهای دیگری نظیر توان و یا عمل جذر و غیره می‌باشد. زبان C# از طریق کتابخانه Net.، انواع توابع ریاضی، مثلثاتی و لگاریتمی را برای برنامه‌نویس فراهم نموده است. این توابع به‌صورت متدهای استاتیک در کلاس Math تعریف شده‌اند. در این قسمت با تعدادی از آنها که ساده می‌باشند، آشنا می‌شویم. البته قرار نیست اینجا ریاضیات کار کنیم، اما خواهید دید به راحتی می‌توانید کار محاسبه را به برنامه خود واگذارید.

به محض نوشتن نام کلاس Math و علامت نقطه، لیستی از متدهای کلاس Math به وسیلهٔ IntelliSense نشان داده می‌شود.



شکل ۱ــ۴ــ لیست متدهای کلاس Math

با حرکت در لیست مربوطه، در مورد هر یک از متدها توضیحی ظاهر می‌شود. این توضیح شامل خط عنوان متد و عملکرد آن است. در واقع شناسنامه متد نشان داده می‌شود. در ادامه به‌توضیح برخی متدهای کلاس Math می‌پردازیم.

۱ــ۱ــ۴ــ متد ( ) Pow

ورودی‌های این متد، دو عدد x و y از نوع اعشاری با دقت بالا است و متد نیز یک عدد اعشاری دقت بالا که حاصل x به توان y است را تولید نموده و به برنامه اصلی باز می‌گرداند.

مثلاً برای محاسبه عبارت ۵ به توان ۲ کافی است اعداد ۵ و ۲ را به این متد ارسال کنیم و این متد نیز پس از اجرا عدد ۲۵ را بر می‌گرداند. از مقدار برگشتی می‌توان استفاده کرد. به عنوان مثال آن را داخل یک متغیر بریزیم و در یک عملیات دیگر مانند چاپ از آن بهره بگیریم :

```
double number = Math. Pow (5, 2);
Console. WriteLine (number);
```

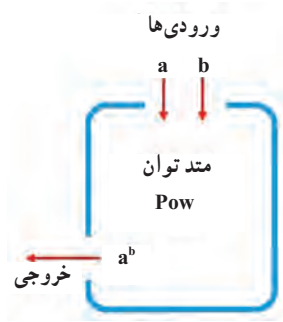
البته می‌توانیم به صورت مستقیم نیز مقدار برگشتی را چاپ کنیم :

```
Console. WriteLine (Math.Pow (5, 2);
```

و یا مقدار خروجی را استفاده نکرد و دور ریخت!

```
Math.Pow (5, 2);
```

عملکرد متد Pow را می‌توان به صورت شکل زیر نشان داد :



شکل ۲ــ۴ــ عملکرد متد Pow

مثال ۱ــ۴ : با اجرای برنامه زیر، عدد ۲ به توان اعداد مختلف (از توان صفر تا ۱۶) روی صفحه نمایش نشان داده می‌شوند.

```
class mathDemo
{
    static void Main ( )
    {
        for (int i = 0; i <= 16;  i++)
        {
            double number = Math. Pow (2, i);
            Console. WriteLine (number);
        }
    }
}
```

برنامه ۱ــ۴ــ نمایش توان‌های مختلف عدد ۲

نکته

کلاس Math در فضای نامی System تعریف شده است و برای استفاده از متدهای آن باید نام System قبل از نام کلاس نیز ذکر گردد و یا در ابتدای برنامه با دستور using، حوزه System به برنامه معرفی شود.

در جدول زیر لیستی از متدهای ریاضی نشان داده شده است. این متدها از کلاس Math هستند.

جدول ۱ــ۴ــ برخی متدهای ریاضی از کلاس Math

نام متد	کاربرد	شکل کلی	مثال
Cos	کسینوس یک زاویه برحسب رادیان را حساب می‌کند.	Math.Cos (عدد)	Math.Cos(10)
Log	لگاریتم عدد را در مبنای تعیین شده حساب می‌کند.	Math. Log (مبنا، عدد)	Math. Log(16,2)
Max	عدد بزرگ‌تر از بین دو عدد را پیدا می‌کند.	Math.Max (عدد، عدد)	Math.Max(10,20)
Min	عدد کوچک‌تر از بین دو عدد پیدا می‌کند.	Math.Min (عدد،عدد)	Math.Min(10,20)
Pow	حاصل عدد به توان تعیین شده را حساب می‌کند.	Math.Pow (توان، عدد)	Math.Pow(10,3)
Round	عدد را گرد می‌کند.	Math.Round (عدد)	Math.Round(10.7)
Sin	سینوس یک زاویه برحسب رادیان را حساب می‌کند.	Math.Sin (عدد)	Math.Sin(20)
Sqrt	جذر عدد را حساب می‌کند. همان رادیکال خودمان.	Math.Sqrt (عدد)	Math.Sqrt(100)
Tan	تانژانت یک زاویه برحسب رادیان را حساب می‌کند.	Math.Tan (عدد)	Math.Tan(20)
Truncate	بخش اعشار عدد را حذف می‌کند.	Math.Truncate (عدد)	Math.Truncate (10.5)

۲ــ۱ــ۴ــ عدد π : برای محاسبه مساحت دایره نیاز به عدد π است. این عدد گنگ از حاصل تقسیم محیط دایره بر اندازه قطر به‌دست می‌آید و برای تمامی‌دایره‌ها ثابت و در حدود ۳/۱۴ است[۱]. اگر نیاز به دقت بیشتری در محاسبات داشتید می‌توانید از ثابت PI که در کلاس Math تعریف شده است استفاده کنید.

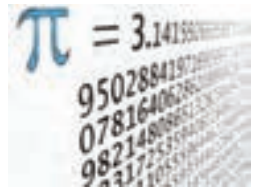
```
Math. PI
```

توجه داشته باشید که PI متد نیست بلکه یک ثابت است. می‌توانید مقدار آن را در یک متغیر اعشاری ذخیره کنید و یا به متد WriteLine() برای چاپ ارسال کنید :

```
Console. WriteLine (Math. PI);
```

فرض کنید : می‌خواهیم اندازه مساحت دایره‌ای را محاسبه و نمایش دهیم که کاربر اندازه شعاع آن را وارد می‌کند.

الگوریتم و روش انجام کار : برای محاسبه مساحت[۲] دایره از فرمول πr^2 استفاده می‌کنیم که منظور از r شعاع[۳] دایره است.




```
// Area of a Circle = πr2
double area = Math. PI * Math. Pow (radius, 2);
```

کنجکاوی

بررسی کنید متد Abs چه عملی انجام می‌دهد؟

۱ــ Pi= 3/14159265358979323846

۲ــ Area

۳ــ Radius

کار در کارگاه ۱

۱ــ برنامه زیر را در محیط VS تایپ کرده و نتیجه اجرای آن را مشاهده کنید :

```
using System;
class Program
{
    static void Main ( )
    {
        Console. WriteLine (Math.Max(100, 200) );   //200
        Console. WriteLine (Math.Max(−100, −200) );

        Console. WriteLine (Math.Min(100, 200) );   //100
        Console. WriteLine (Math.Min(−100, −200) );

        Console. WriteLine (Math.Pow (2, 3) );   //8
        Console. WriteLine (Math.Pow(−2, 3) );

        Console. WriteLine (Math.Round(18.5) );  //19
        Console. WriteLine (Math.Round(18.25) );

        Console. WriteLine (Math.Sqrt (225) );   //15
        Console. WriteLine (Math.Sqrt (200) );

        Console. WriteLine (Math.Truncate (1.23) );   //1
        Console. WriteLine (Math.Truncate (1.54) );
    }
}
```

برنامه ۲ــ۴ــ آزمایش متدهای کلاس Math

۲ــ خروجی هر خط برنامه را بنویسید. نتیجه خروجی بعضی از خطوط برنامه نوشته شده است.

مثال ۲ــ۴ : برنامه‌ای بنویسید که ۱۰ عدد را بگیرد و کمترین آنها را نمایش دهد.

الگوریتم و روش انجام کار : برنامهٔ محاسبهٔ کمترین عدد را سال گذشته نوشته‌اید. با توجه به اینکه در فصل ۲ آرایه را یاد گرفته‌اید، این برنامه را با آرایه می‌نویسیم. برای این منظور پس از تعریف متغیر آرایه‌ای، برای دریافت اعداد از حلقه for کمک می‌گیریم. سپس اولین عدد را به عنوان کمترین عدد لیست در متغیر min می‌ریزیم و از عدد دوم تا انتهای لیست، عمل مقایسه و ریختن در متغیر min را در صورت نیاز انجام می‌دهیم. متد Min کلاس math به ما کمک می‌کند تا عمل مقایسهٔ زیر را با دستور انتساب جایگزین کنیم (برنامه ۳ــ۴).

```
if (min < arr [i])
    min = arr[i];
```

سؤال: با توجه به برنامه زیر توضیح دهید کدام دستور جایگزین شرط بالاست.

```
using System;
namespace MinAndMax
{
    class Program
    {
        static void Main(string[ ] args)
        {
            int[] arr;
            arr = new int[10];
            for (int i = 0; i <=9; i++)
            {
                Console.Write("enter the {0}th number: ", i+1);
                arr[i] = int.Parse(Console.ReadLine());
            }
            int min = arr[0];
            for (int i = 1; i <= 9; i++)
                min = Math.Min(min, arr[i]);
```

```
            Console.WriteLine("minimum is {0}", min);
            Console.ReadKey();
        }
    }
}
```

برنامه ۳ــ۴ــ محاسبه کوچک‌ترین عدد

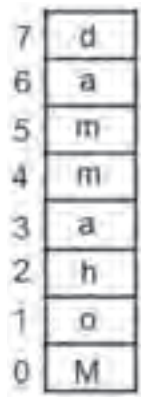
سؤال: اگر بخواهیم بیشترین عدد را هم در این برنامه نمایش دهیم چه تغییراتی باید انجام دهیم؟

۲ــ۴ــ کلاس string

همان‌طور که در برنامه می‌توان متغیری برای نگهداری داده‌های عددی تعریف کرد، برای داده‌های رشته‌ای نیز می‌توان متغیری از نوع string تعریف کرد. این نوع داده در واقع یک کلاس است و متغیری که از این نوع تعریف می‌شود اشاره به ردیفی از کاراکترها می‌کند.

مثال ۳ــ۴ : برای نگهداری کلمه Mohammad از یک متغیر رشته‌ای به صورت زیر استفاده می‌کنیم :

```
string name = "Mohammad" ;
```



شکل ۴ــ۴ــ ساختمان یک رشته

برای آنکه به حروف یک رشته دسترسی داشته باشیم می‌توانیم مانند آرایه‌ها، از اندیس حروف استفاده کنیم.

[شماره کاراکتر] نام رشته متنی

توجه داشته باشید اندیس از صفر شروع می‌شود. بنابراین name[5] کاراکتر ششم رشته name را در خود جای داده است.

نکته

تغییر ناپذیری یک رشته : پس از اینکه یک رشته در حافظه ایجاد شد، محتویات آن قابل تغییر نیست. به عبارت دیگر یک رشته پس از ایجاد، **تغییر ناپذیر**[1] است. شاید از خود بپرسید چرا می‌توانیم به یک رشته مقداری دیگر دهیم؟ پاسخ این است که هر بار که متغیر رشته‌ای مقدار دیگری می‌گیرد مانند این است که دوباره تعریف شده است! اما ما نمی‌توانیم مثلاً یک حرف از آن را تغییر دهیم.

اگر متغیری را از نوع string تعریف نماییم، این متغیر دارای متدهایی خواهد بود که می‌توانند عملیات سودمندی را انجام دهند. مثلاً متدی به نام ()ToLower قادر است حروف انگلیسی موجود در رشته را به حروف کوچک تبدیل کند و در مقابل آن، متد ()ToUpper می‌تواند حروف انگلیسی رشته را به حروف بزرگ تبدیل کند.

۱ ـ Immutable

جدول ۲ــ۴ــ برخی متدهای رشته‌ای از کلاس String[1]

متد	شرح کار متد	نوع برگشتی
CompareTo()	مقایسه دو رشته	bool
IndexOf()	موقعیت وجود یک کاراکتر در رشته را برمی‌گرداند.	int
ToLower ()	تبدیل تمام کاراکترهای یک رشته به حروف کوچک	string
ToUpper ()	تبدیل تمام کاراکترهای یک رشته به حروف بزرگ	string
Insert()	درج یک کاراکتر یا یک رشته در درون یک رشته دیگر	string
Length	تعداد کاراکترهای یک رشته را برمی‌گرداند.	int
Replace()	یک کاراکتر در یک رشته را با کاراکتر یا یک رشته دیگری عوض می‌کند.	string

نکته

این متدها روی خود رشته تأثیر نمی‌گذارند بلکه یک رشته دیگر بر می‌گردانند. بنابراین محتوای متغیر رشته‌ای را تغییر نمی‌دهند.

کار در کارگاه ۳

مثال ۴ــ۴ : کار با متدهای رشته‌ای :

۱ــ عملکرد متدهای مختلف کلاس string را با تایپ برنامه صفحه بعد مشاهده کنید :

۱ــ متدهای کامل در پیوست ۳ ببینید.

```
using System;
class Program
{
    static void Main(string[] args)
    {
      string s = "  This is a String!  ";
      Console.WriteLine ("the main string :  \"{0}\"", s);
      Console.WriteLine (" ToLower:  \"{0}\"", s.ToLower());
      Console.WriteLine (" ToUpper:  \"{0}\"", s.ToUpper());
      Console.WriteLine (" Insert:  \"{0}\"", s.Insert(0, "Note," ) );
      Console.WriteLine (" Length:  {0}", s.Length);
      Console.WriteLine (" index of 'i':  {0}", s.IndexOf('i'));
      Console.WriteLine (" Remove: \"{0}\"", s.Replace("This is", "we have" ) );
      Console.ReadKey();
    }
}
```

برنامه ۴ــ۴ــ آزمایش متدهای کلاس String

کار در کارگاه ۴

مثال ۵ ــ ۴ : برنامه‌ای بنویسید که یک نام کاربری و گذرواژه را بگیرد. اگر نام کاربری user و گذرواژه "pass" وارد شود پیام ورود موفقیت‌آمیز داده شود. همان‌طوری که در پست الکترونیکی و برنامه‌های مختلف دیده‌اید نام کاربری به کوچک یا بزرگی حروف حساس نیست ولی گذرواژه باید دقیقاً مطابقت داشته باشد.

الگوریتم و روش انجام کار : در این برنامه پس از دریافت نام کاربری و گذرواژه، باید بررسی شود که برابر با مقدار خواسته شده می‌باشد یا خیر. برای آنکه بررسی نام کاربری حساس به نوع حروف نباشد نامی که کاربر وارد می‌کند را به رشته‌ای با حروف کوچک تبدیل می‌نماییم. چرا؟

```
using System;
class Program
{
    static void Main(string[] args)
    {
        Console.Write("enter username: ");
        string username = Console.ReadLine();
        Console.Write("enter password: ");
        string password = Console.ReadLine();
        if (username.ToLower() == "user" && password == "pass")
            Console.WriteLine("your login was successful. Welcome!");
        else
            Console.WriteLine("username or password is wrong");
        Console.ReadKey();
    }
}
```

برنامه ۵ــ۴ــ بررسی نام کاربری حساس به کوچکی و بزرگی حروف

سؤال: اگر بخواهیم از ToUpper به‌جای ToLower استفاده کنیم کد چه تغییری می‌کند؟

سؤال: این برنامه را طوری تغییر دهید که تا واردنشدن نام و گذرواژه صحیح، با دادن پیام مناسب، عمل دریافت تکرار شود.

۳ــ۴ــ کلاس Array

در فصل ۲ با مبحث آرایه و کاربردهای آن آشنا شدید. برخی از برنامه‌هایی که با استفاده از آرایه‌ها نوشتیم دارای کدهای طولانی و وقت‌گیری است که با کلاس Array و متدهای آن قابل جایگزینی است. از ویژگی‌های این کلاس، وجود متدهای مختلف استاتیک، برای عملیات بر روی آرایه‌ها است. این متدها کاملاً آزمایش شده و مطمئن هستند. برای مثال عمل جستجوی خطی، دودویی و مرتب سازی آرایه با استفاده از متدهای کلاس Array به راحتی قابل انجام است. در جدول ۳ــ۴ لیست این متدها و توضیحات آنها را می‌بینید.

جدول ۳ــ۴ــ برخی متدهای کلاس Array

نام متد	کاربرد	شکل کلی	مثال
Sort	مرتب سازی لیست	Array.Sort (نام آرایه)	Array.Sort (names)
Reverse	وارونه کردن عناصر آرایه	Array.Reverse (نام آرایه)	Array.Sort (names)
IndexOf	جستجوی یک مقدار در آرایه و برگرداندن مکان اولین مورد پیدا شده	Array.IndexOf (نام آرایه، مقدار)	Array.IndexOf(names,"ali")
LastIndexOf	جستجوی یک مقدار در آرایه و برگرداندن مکان آخرین مورد پیدا شده	Array.LastIndexOf (نام آرایه،مقدار)	Array.LastIndexOf(names, "ali")
BinarySearch	جستجوی یک مقدار در آرایه مرتب شده صعودی و برگرداندن مکان اولین مورد پیدا شده	Array.BinarySearch (نام آرایه، مقدار)	Array.BinarySearch(names, name));
Copy	کپی کردن عناصر یک آرایه در آرایه دیگر از اولین عنصر تا تعداد تعیین شده	Array.Copy (تعداد، نام آرایه مبدأ عناصر، نام آرایه مقصد)	Array.BinarySearch(names, family,10);
Clear	پاک کردن و تنظیم مقدار عناصر تعیین شده آرایه یا مقدار پیش فرض نوع داده آرایه	Array.Clear (تعداد عناصر، شروع اندیس، نام آرایه)	Array.BinarySearch(names, 4,10);

کار در کارگاه ۵

مثال ۶ــ۴ : برنامه‌ای بنویسید که اسامی دانش‌آموزان یک کلاس را دریافت کرده و سپس آنها را براساس نام، مرتب کرده و نمایش دهد.

الگوریتم یا روش انجام کار : ابتدا تعداد دانش آموزان از کاربر دریافت و به همان میزان، آرایه‌ای ایجاد و اسامی از کاربر دریافت می‌شود. در مرحله بعد، اسامی وارد شده مرتب و در نهایت

اسامی چاپ می‌شود.

```
using System;
class Program
    {
    static void Main(string[] args)
        {
        Console.Write("enter number of students: ");
        byte count = byte.Parse(Console.ReadLine());
        string[] studentName = new string[count];
        for (int i = 0; i < count; i++)
        {
            Console.Write("Student Name {0}: ", i + 1);
            studentName[i] = Console.ReadLine();
        }
        Array.Sort(studentName);
        Console.WriteLine("students in name order: ");
        for (int i = 0; i < count; i++)
        Console.WriteLine("{0,3} ==> {1,-25}", i + 1, studentName[i]);
        Console.ReadKey();
    }
}
```

برنامه ۶ــ۴ــ مرتب‌سازی اسامی با استفاده از متد Sort

۲ــ آرایه را پس از مرتب شدن برعکس نمایید.

۳ــ در انتهای برنامه و قبل از دستور Console.ReadKey() قطعه برنامه زیر را که نتیجه جستجوی دودویی و خطی یک نام در آرایه است را اضافه کنید.

```
Console.Write(" \n enter a name for search in students: ");
string name = Console.ReadLine ( );
Console.WriteLine("Binary Search Result: " + Array.Binary Search
(studentName, name) );
Console.WriteLine("linear search Result: " + Array.Index Of (student Name,
name) );
```


شکل ۵ــ۴

۴ــ در اجرای برنامه، یکی از اسامی موجود در لیست دانش آموزان را برای جستجو وارد کنید. نتایج را مشاهده کنید. نتیجه جستجوی دودویی ۱- خواهد بود. این پاسخ به معنای یافت نشدن نام مورد نظر است. چرا نتیجه چنین است؟

۱ــ هر متد از سمت راست را با توضیح سمت چپ مطابقت دهید (یک توضیح اضافی است).

الف) Math.Sqrt	۱ــ گرد کردن عدد
ب) Math.Pow	۲ــ مرتب کردن آرایه
ج) Math.Round	۳ــ وارون نمودن آرایه
د) Array.BinarySearch	۴ــ توان رساندن عدد
و) Array.Sort	۵ــ جستجوی خطی کلید
هـ) Array.Reverse	۶ــ محاسبه جذر عدد
	۷ــ جستجوی دودویی در یک آرایه مرتب

۲ــ با متدهایی که تاکنون شناخته‌اید جدول زیر را تکمیل کنید.

کلاس math	کلاس string	کلاس array

۳ــ راجع به کاربرد متدهایی که در جدول بالا نوشته‌اید با هم کلاس خود بحث کنید.

۴ــ خروجی برنامه زیر چیست؟

```
using System;
class Convert
{
    static void Main(string[] args)
    {
        int a, b;
        double d = 0.75;
        a = (int) Math. Round (d);
        b = (int)d;
        Console. WriteLine (a);
        Console. WriteLine (b);
    }
}
```

۵ــ حاصل عبارت زیر چیست؟

```
Math.Truncate(Math.Round (Math.Pow(2,3)/5) +2.8)
```

۶ــ خروجی دستورات زیر چیست؟

```
static void Main(string[] args)
{
        int [] numbers = { 7, 42, 16, 8, 16, 14 };
        Array.Reverse(numbers);
        Array.Sort(numbers);
        for (int i = 0; i < 6; i++)
           Console.WriteLine(numbers[i]);
         Console.ReadKey();
}
```

۷ــ در سؤال قبل اگر به‌جای دستور for دستور زیر را می‌نوشتیم چه چیزی نمایش داده می‌شد؟

```
Array.IndexOf (numbers, 16);
```

۸ــ دستورات زیر چه کاری انجام می‌دهند؟

```
static void Main(string[] args)
{
    string exp = "i am a good student";
    for (int i = 0; i < strings.Length; i++)
       Console.WriteLine(strings[i]);
    Console.ReadKey();
}
```

۱ــ می‌خواهیم ۲۰ نفر از دانش آموزان را برای نشستن در کارگاه رایانه، به گروه‌های دو نفری تقسیم کنیم. برنامه ای بنویسید که نام این ۲۰ نفر را دریافت کند و با مرتب کردن نام آنها هر دو نفر را برای گروه‌های شماره بندی شده از ۱ تا ۱۰ نمایش دهد.

۲ــ برنامه‌ای بنویسید که یک رشته متنی را دریافت کند. کلمات آن را برعکس نمایش دهد. به عنوان مثال اگر رشته متنی "I am a good student" را به برنامه بدهیم متن "student good a am I" را نمایش دهد.

۳ــ می‌خواهیم در کلاس خود یک فعالیت پژوهشی با حضور ۱۰ دانش‌آموز اول لیست کلاسی انجام دهیم. برنامه‌ای بنویسید که اسامی ۳۰ دانش‌آموز را دریافت کرده و اسامی ۱۰ نفر اول دفتر کلاسی (می‌دانید که اسامی دانش‌آموزان در دفتر کلاسی بر اساس حروف الفبا مرتب شده است) را که در دفتر کلاسی پژوهش (آرایه جدید) نیز وارد می‌کند، نمایش دهد.

فعالیت

متن زیر از MSDN با موضوع آرایه برداشت شده است. آن را با کمک هم کلاسی خود ترجمه کنید و به کلاس ارائه نمایید.

Initializing Arrays

C# provides simple and straightforward ways to initialize arrays at declaration time by enclosing the initial values in curly braces ({}). The following example shows the way to initialize an array.

Note If you do not initialize an array at the time of declaration, the array members are automatically initialized to the default initial value for the array type. Also, if you declare the array as a field of a type, it will be set to the default value null when you instantiate the type.

```
int[] numbers = new int[5] {1, 2, 3, 4, 5};
string[] names = new string[3] {"Matt", "Joanne", "Robert"};
```

واژگان و اصطلاحات انگلیسی فصل چهارم

ردیف	واژه انگلیسی	معنی به فارسی
۱	Area	
۲	Array	
۳	Immutable	
۴	Upper	
۵	Lower	
۶	Radius	

فصل پنجم

ایجاد برنامه‌های ویندوزی با استفاده از ویژوال استودیو

خروجی تمام برنامه‌هایی که در کتاب برنامه سازی یک و همچنین فصل‌های قبلی در این کتاب نوشتید در محیط کنسول قابل مشاهده بود. در این فصل می‌خواهیم با روش برنامه نویسی در محیط ویندوز آشنا شویم تا بتوانیم برنامه‌هایی با ظاهری زیباتر، مانند برنامه‌های کاربردی رایج در ویندوز بنویسیم. برای محقق شدن این هدف، در دو فصل آینده برنامه‌های خود را در محیط ویندوزی خواهیم نوشت.

پس از پایان این فصل انتظار می‌رود که فراگیر بتواند :

۱ــ تفاوت بین برنامه‌های کنسولی و ویندوزی را توضیح دهد.

۲ــ واسط گرافیکی کاربر و کاربرد کلاس From را توضیح دهد.

۳ــ گزینه‌های مورد نیاز پنجره‌ها، منوها و نوار ابزارهای IDE را در محیط ویندوزی مورد استفاده قرار دهد.

۴ــ پروژه ویندوزی با یک فرم ایجاد نماید و ویژگی‌های مورد نیاز را تنظیم نماید.

۵ ــ کنترل‌های برچسب و جعبه تصویر را به فرم خود اضافه کند و برخی ویژگی‌های آنها را به دلخواه تغییر دهد.

۶ــ با استفاده از VS برنامه ویندوزی را ترجمه و اجرا نماید.

۱ــ۵ــ تفاوت برنامه‌های کنسولی و برنامه‌های ویندوزی

برنامه‌های کنسولی که تاکنون نوشته‌اید برای یادگیری مفاهیم اساسی و مقدماتی زبان برنامه‌نویسی C# و به‌دست آوردن تجربهٔ کدنویسی خوب است اما برای تولید برنامه‌های کاربردی که در زندگی روزمره کاربرد داشته باشد، چندان مناسب نیست. امروزه بیشتر کاربران کامپیوتر، با برنامه‌های کاربردی رایج کار کرده‌اند و توقع دارند برنامه‌ای که شما می‌نویسید دارای ظاهر و عملکردی شبیه برنامه‌های رایج در ویندوز باشند. در اجرای برنامه‌هایی که از این به بعد می‌نویسید انتظار می‌رود که با اجرای آن، یک پنجره باز شود و در داخل آن منوها و ابزارهای مختلفی دیده شود که با کلیک کردن روی هر منو یا روی گزینه‌ای از یک منو، پنجره دیگری ظاهر شود. در این پنجره‌ها، اطلاعاتی از کاربر در قالب یک فرم دریافت می‌شود و یا اطلاعاتی نشان داده می‌شود. در هر فرم، قسمت‌هایی برای ورود اطلاعات و کلیدهایی برای تأیید و ثبت اطلاعات و یا لغو کردن عملیات وجود دارد. به عنوان نمونه در شکل زیر نمونه ای از برنامه حدس عدد را که در کتاب برنامه‌سازی ۱ در محیط کنسول ساخته بودید، در محیط ویندوز مشاهده می کنید. این برنامه را در ادامه می‌سازید.

شکل ۱ــ۵ــ یک فرم و برنامه کاربردی

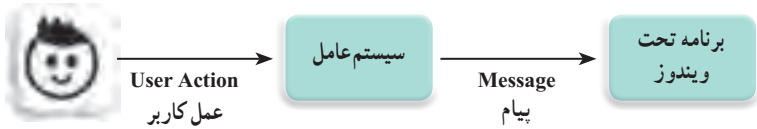
در برنامه‌های کنسولی، اجرای برنامه از متدی به نام Main() شروع می‌شود و دستورات داخل آن، به ترتیب و خط به خط اجرا می‌شوند. در چنین برنامه‌هایی با استفاده از متدهای موجود مانند ReadLine() برای دریافت یک رشته، درخواست‌هایی به سیستم عامل داده می‌شود که عملیاتی را برای ما انجام دهد.

شکل ۲ــ۵

در برنامه‌های ویندوزی نیز اجرای برنامه با متدی به نام Main() شروع می‌شود اما برخلاف برنامه‌های کنسول، برنامه در حالت انتظار[1] قرار می‌گیرد تا یک اتفاق یا رویداد[2] رخ دهد که در این صورت نسبت به آن واکنش نشان دهد. مثلاً وقتی کاربر با ماوس بر روی دکمه[3] در شکل ۱ــ۵ کلیک می‌کند، سیستم عامل که مدیر وکنترل کننده منابع سیستم است متوجه می‌شود و این اتفاق را به عنوان یک رویداد به برنامه اطلاع می‌دهد.

رویداد چیست؟ رویداد یک اطلاع[4] است که از طرف سیستم عامل به برنامه داده می‌شود تا نشان دهد که یک اتفاق رخ داده است.

برنامه‌های ویندوزی باید به رویدادهایی که برای آنها مهم است، عکس العمل نشان دهند.



شکل ۳ــ۵

برای مقایسه بهتر برنامه‌های کنسولی و برنامه‌های ویندوزی، مثال زیر را در نظر بگیرید :

فرض کنید شما در منزل هستید و قرار است برای شما مهمان بیاید. اگر بخواهید به موقع درب منزل را برای وی باز کنید می‌توانید به یکی از روش‌های زیر عمل کنید :

۱ــ هر چند لحظه یک بار، به سمت در منزل بروید و بیرون در را نگاه کنید که آیا مهمان شما آمده است.

۲ــ می‌توانید کارهای دیگر خود را رها کنید و در جلوی در منزل خود، منتظر شوید تا وی بیاید.

۳ــ می‌توانید در داخل منزل مشغول انجام کارهای خود باشید، هر وقت که مهمان آمد، زنگ زد و شما را از ورودش مطلع کرد، به استقبال وی بروید و در را برایش باز کنید.

شکل ۴ــ۵

۱ــ Wait state

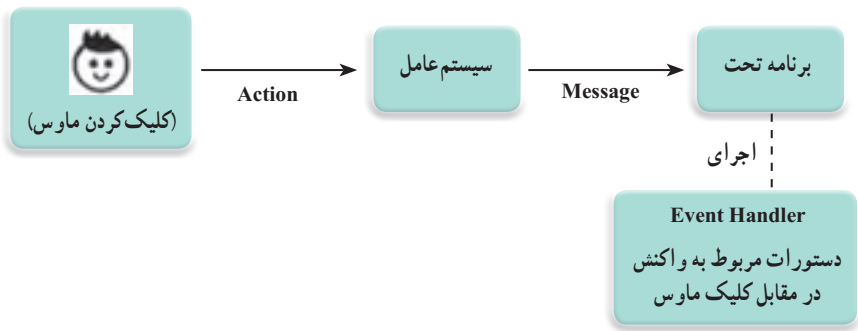
۲ــ Event

۳ــ Button

۴ــ Notification

در برنامه‌های کنسولی، در دستوراتی مانند دستور دریافت داده، اجرای برنامه متوقف می‌شود و تا زمانی که کلید Enter زده شود دستورات بعدی اجرا نمی‌شوند. پس از اینکه کاربر اطلاعات را وارد کرد و کلید Enter را زد، خطوط بعدی برنامه به ترتیب اجرا می‌شوند. در مقایسه با مثال بالا می‌توان گفت در برنامه‌های کنسولی از روشی مانند روش ۱ و یا روش ۲ استفاده می شود. به نظر شما در برنامه‌های ویندوزی، از کدام روش استفاده می‌شود؟

اما در محیط ویندوز، هنگامی‌که کاربر عملی[1] را انجام می‌دهد مثلاً کلیدی از صفحه کلید را فشار می‌دهد و یا با استفاده از ماوس بر روی منویی و یا کلیدی کلیک می‌کند، از طرف سیستم عامل به برنامه اطلاع داده می‌شود که یک رویداد رخ داده است. در واقع برنامه‌های ویندوز با دریافت پیام[2]هایی از سیستم عامل، از رخ دادن رویداد مطلع می‌شوند. بنابراین در برنامه‌های ویندوزی شما به عنوان برنامه نویس، باید پیش بینی کنید که اگر کاربر عملی را انجام دهد، برنامه چگونه نسبت به آن واکنش نشان دهد و برای این منظور باید متدهایی را بنویسید که در مواجهه با یک رخداد یا رویداد مانند کلیک ماوس در یک محل، به کامپیوتر اعلام کند که چه کاری باید انجام شود. به این متدها، Event Handler گفته می‌شود که برای سادگی در این کتاب به جای آن به اختصار EH را به کار می‌بریم.



شکل ۵ــ۵

به این ترتیب هنگامی‌که کاربر بر روی کلیدی با ماوس کلیک می‌کند، این رویداد از طریق سیستم‌عامل شناسایی می‌شود و از طرف سیستم عامل یک پیام به برنامه فرستاده می‌شود و متد EH مربوط به آن رویداد که قبلاً نوشته‌ایم به طور خودکار اجرا می‌شود. طبیعی است که اگر برنامه فاقد

۱ــ Action

۲ــ Message

متدی برای یک رویداد باشد، هر چقدر که سیستم عامل پیام بفرستد برنامه در مقابل آن واکنشی نشان نخواهد داد. شکل کلی یک متد EH به صورت زیر است :

```
void نام متد (فرستنده پیام، جزئیات رویداد)
{
دستورات واکنش به رویداد
}
```

شکل ۶ــ ۵

هر متد EH دارای دو پارامتر ورودی است که اولی نوع شئ[1] فرستنده پیام را مشخص می‌کند و پارامتر دوم مربوط به جزئیات رویداد است مثلاً در مورد کلیک ماوس شامل اطلاعاتی در مورد موقعیت مکانی ماوس در لحظه کلیک و کلیدی (کلید چپ، راست یا وسط) از ماوس است که فشار داده شده است.

هنگامی‌که یک برنامه ویندوزی مانند برنامه ماشین حساب را اجرا می‌کنید، این برنامه پس از نمایش پنجره خود، در حالت انتظار قرار می‌گیرد، تا شما مثلاً بر روی دکمه ای کلیک کنید و یا از طریق صفحه کلید، علامت یا رقمی‌را وارد کنید که در این صورت یک رویداد رخ می‌دهد و EH مربوطه اجرا می شود.

یک تفاوت دیگر بین برنامه ویندوزی و کنسولی این است که، برنامه‌های ویندوزی پس از واکنش به رویدادها و انجام عملیات مربوطه، مجدداً در حالت انتظار برای رویداد بعدی به سر می‌برند تا در نهایت کاربر از برنامه خارج گردد.

در ضمن با توجه به اینکه رویدادهای مختلفی ممکن است از طریق کاربر رخ دهد که قابل پیش‌بینی نیست، بنابراین با اجرای برنامه، مشخص نیست که کدام رویداد ممکن است ابتدا رخ دهد و به عبارت دیگر ترتیبی برای رویدادها، قابل پیش بینی نیست.

۲ــ۵ــ واسط گرافیکی کاربر

در طول دوره‌های مختلف تحصیلی، ممکن است، یک روزنامه دیواری با دوستان خود درست کرده باشید. برای ایجاد روزنامه دیواری ابتدا به یک مقوا نیاز دارید تا مطالب خود را بر روی آن

۱ــ Object

بنویسید و یا عکس‌ها و بریده‌های مجلات را بر روی آن بچسبانید. در ساخت یک برنامه ویندوزی نیز باید صفحه یا فرمی در اختیار داشته باشید تا انواع دکمه‌ها، منوها، تصاویر و نوشته‌ها را روی آن قرار دهید. این فرم، واسط گرافیکی کاربر یا GUI[1] نامیده می شود.

فرم، صفحه‌ای است که اجزای گرافیکی گوناگونی بر روی آن قرار می‌گیرد. اندازه یک فرم متناسب با تعداد و اندازه اشیای گرافیکی است که قرار است روی آن جای گیرند (شکل ۷ــ۵).

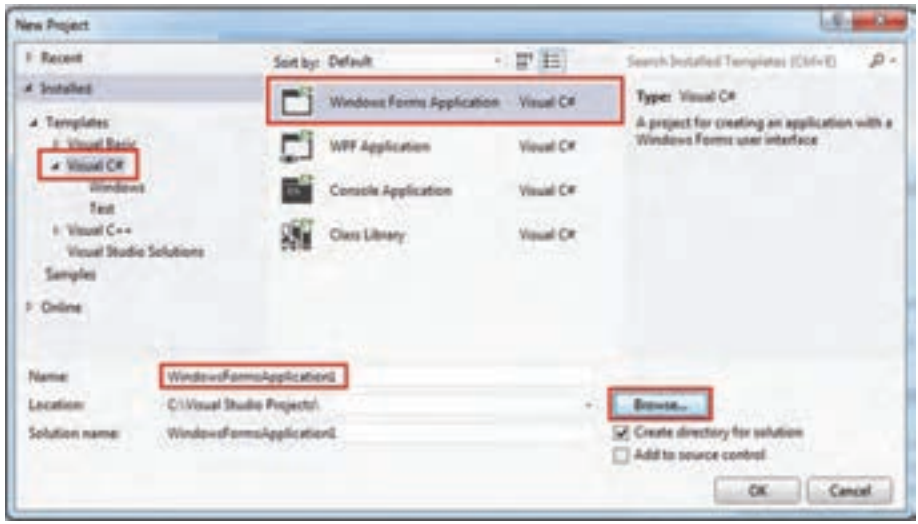
شکل ۷ــ۵ــ یک فرم خالی مانند یک مقوای روزنامه دیواری

همان طور که برای خرید مقوای روزنامه دیواری، لازم است به یک فروشگاه لوازم التحریر مراجعه کنید، در زبان برنامه نویسی C# نیز، برای ایجاد یک فرم باید به کتابخانه ارزشمند Net. مراجعه کنید. در این کتابخانه، کلاسی به نام Form تعریف شده است که برای ایجاد یک فرم لازم است از آن به عنوان پایه و مبنای کار خود استفاده کنیم.

۳ــ۵ــ ایجاد یک پروژه ویندوزی با کمک VS

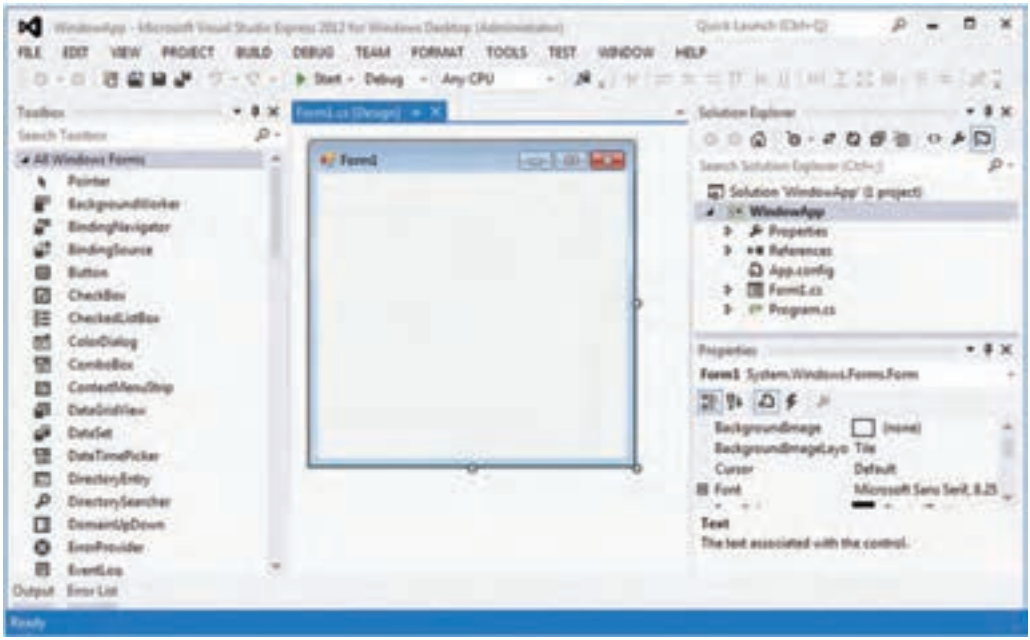
برای ایجاد یک برنامه ویندوزی، VS را اجرا کنید و در صفحه آغازین، روی گزینهٔ New Project کلیک کنید در پنجره‌ای که مطابق شکل ۸ــ۵ نمایان می‌شود، گزینهٔ Visual C# و سپس Windows Forms Application را انتخاب کنید.

۱ــ Graphical User Interface



شکل ۸ــ۵ــ پنجره ایجاد پروژه جدید

می‌توانید نام و مسیر ذخیره سازی پروژه را مانند برنامه‌های کنسولی، انتخاب کنید. پس از تأیید و کلیک بر روی دکمه OK، وارد محیط IDE (شکل ۹ــ۵) می‌شوید.



شکل ۹ــ۵ــ پنجره IDE پروژه ویندوزی

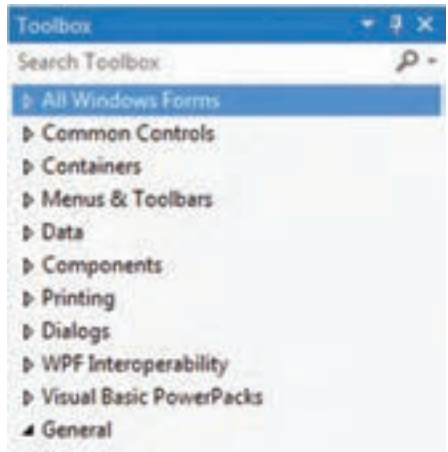
همان‌طور که در شکل ۹ــ۵ مشاهده می‌شود خیلی با این محیط غریبه نیستیم و شبیه محیط تولید برنامه کنسولی است. در اینجا گذری بر روی پنجره‌های مختلف آن خواهیم داشت.

۱ــ۳ــ۵ــ آشنایی با پنجره‌های IDE

در بالای صفحه، منوها و نوار ابزارها[1] مانند محیط تولید برنامه‌های کنسولی وجود دارد. البته همان‌طور که در شکل ۱۰ــ۵ دیده می‌شود یک منو به نام Format اضافه شده است که از گزینه‌های آن برای شکل‌دهی ظاهری و تنظیم جای دهی کنترل‌های یک فرم استفاده می‌شود.

شکل ۱۰ــ۵ــ نوار منوها و نوار ابزارهای IDE در یک پروژه ویندوزی

در سمت چپ صفحه (IDE)، پنجره ای به نام جعبه ابزار[2] دیده می‌شود. در پنجره جعبه ابزار، نام و آیکن کنترل‌های قابل استفاده، لیست شده است. کنترل‌ها در جعبه ابزار، دسته بندی شده‌اند و هنگامی که بر روی نام یک گروه، کلیک می کنید کنترل‌های مربوطه نمایش داده می شوند و اگر مجدداً روی نام آن گروه کلیک کنید، لیست مربوطه بسته می‌شود. در شکل ۱۱ــ۵ جعبه ابزار به صورت گروه بندی نشان داده شده است.

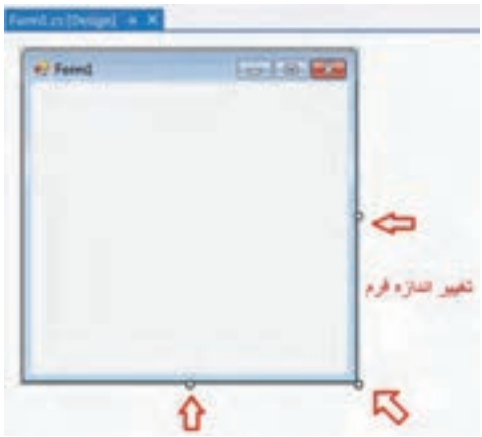


شکل ۱۱ــ۵ــ پنجره جعبه ابزار

۱ــ Tool bar

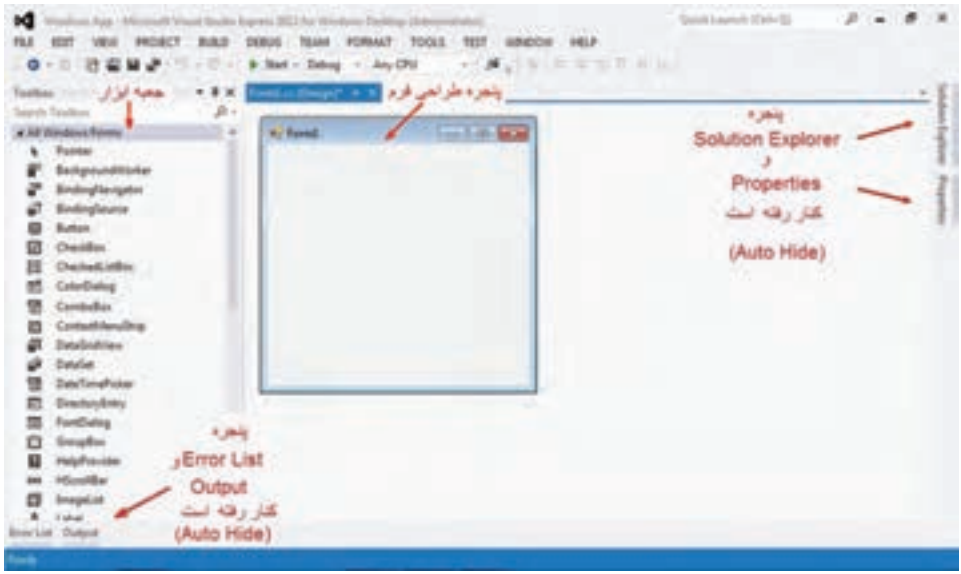
۲ــ Tool box

در وسط صفحه، یک فرم دیده می‌شود که از آن برای طراحی واسط کاربری و قراردادن کنترل‌ها مانند مقوای روزنامه دیواری، استفاده خواهیم کرد. اندازه طول و عرض فرم به راحتی با استفاده از ماوس و با کشیدن گوشه‌های آن، که در شکل ۱۲ــ۵ مشخص شده است، قابل تغییر است. با این پنجره که پنجره طراحی فرم[1] نامیده می‌شود، زیاد کار داریم.



شکل ۱۲ــ۵ــ پنجره طراحی فرم

با توجه به اینکه محیط VS قابل تنظیم است، ممکن است صفحه‌ای که روی کامپیوتر شما دیده می شود با آنچه که در شکل ۹ــ۵ مشاهده می‌کنید، کمی‌متفاوت باشد. مثلاً در شکل ۱۳ــ۵، پنجره‌هایی در حالت Auto Hide، قرار دارند و با کلیک بر روی نام آنها، پنجره دیده می‌شود و با کلیک در نقطه دیگری از صفحه، پنجره‌ها به طور خودکار کنار می‌روند و فقط نام پنجره‌ها دیده می‌شوند.



شکل ۱۳ــ۵ــ حالت Auto Hide پنجره‌ها

۱ــ Form Designer

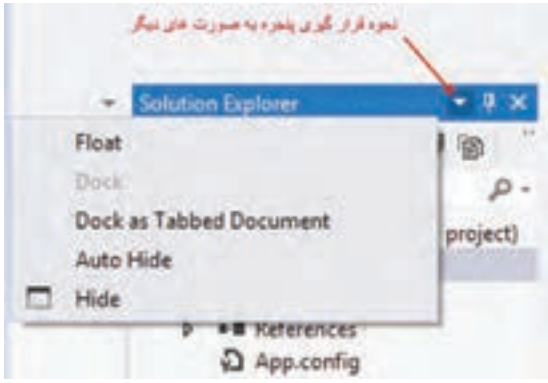
از قابلیت Auto Hide برای کنار رفتن موقتی پنجره‌ها و برای بزرگ‌شدن محیط کار استفاده می‌شود. برای اینکه پنجره‌ای را از حالت Auto Hide خارج کنید و یا این قابلیت را به آن بدهید، کافی است در روی خط عنوان پنجره بر روی آیکن پونز[1] کلیک کنید. شکل ۱۴ــ۵ را مشاهده کنید.

شکل ۱۴ــ۵ــ قابلیت کنار رفتن پنجره به طور خودکار فعال است.

اگر پونز در حالت افقی مانند شکل ۱۴ــ۵ باشد، قابلیت Auto Hide فعال است و اگر روی آن کلیک کنید، پونز به حالت عمودی قرار می‌گیرد و پنجره ثابت می‌ماند. شکل ۱۵ــ۵ را مشاهده کنید.

شکل ۱۵ــ۵ــ حالت کنار رفتن خودکار پنجره غیر فعال است.

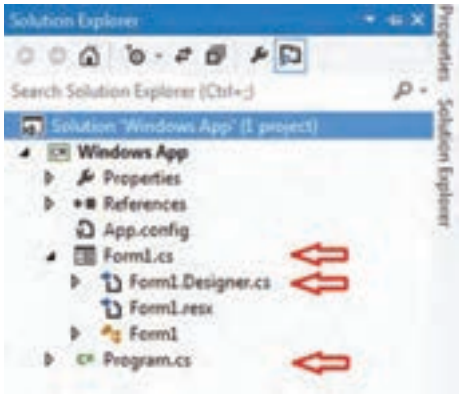
در کنار آیکن پونز، آیکن مثلثی شکلی مانند شکل ۱۶ــ۵ نیز قرار دارد که با کلیک بر آن منوی Windows Posit ion باز می‌شود تا روش‌های دیگری برای قرارگیری پنجره فراهم کند. در قسمت کار در کارگاه آنها را تجربه خواهید کرد.



شکل ۱۶ــ۵ــ نحوه قرار گیری و نمایش پنجره

۱ــ Pushpin

به پنجره Solution Explorer مطابق شکل ۱۷ــ۵ دقت کنید در این پنجره، نسبت به آنچه که در برنامه کنسولی مشاهده کردید، موارد بیشتری دیده می‌شود.



شکل ۱۷ــ۵ــ پنجره Solution Explorer در یک پروژه ویندوزی

علاوه بر فایل Program.cs که نام آن برای شما آشنا است، فایل‌های دیگری از جمله دو فایل به نام‌های Form1.cs و Form1.Designer.cs دیده می‌شوند. دستورات یک برنامه ویندوزی در این سه فایل قرار می‌گیرد. اگر در پنجره Solution Explorer روی نام فایل Program.cs، دوبار کلیک کنید، پنجره‌ای باز می‌شود و محتویات آن را نشان می‌دهد. در این فایل کلاس اصلی برنامه و متد Main()مطابق شکل ۱۸ــ۵ قرار دارد.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Windows_App
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

شکل ۱۸ــ۵ــ محتوای فایل Program.cs

اگر در پنجره Solution Explorer بر روی کلمه Form1.cs کلیک راست کنید، منویی مانند شکل ۱۹ ــ ۵ ظاهر می‌گردد که شامل گزینه‌هایی برای انجام عملیات مختلف بر روی آن فایل است. به این منوها که گزینه‌های آن مربوط به یک موضوع خاص است منوی موضوعی[1] یا محلی گفته می‌شود.

شکل ۱۹ــ۵ــ منوی موضوعی در مورد فایل Form1.cs

در منوی ظاهر شده، اگر بر روی گزینه View Designer کلیک کنید یا از میان‌بر[2] آن Shift+F7 استفاده کنید، پنجره طراحی فرم را خواهید دید.

نکته

در هنگام برنامه‌نویسی در محیط VS، سعی کنید از کلیدهای میان‌بر استفاده کنید تا سرعت شما در انجام عملیات بالا رود. مثلاً هنگامی که در پنجره Solution Expolrer فایل Form1.cs انتخاب شده است، برای مشاهده دستورات آن و یا برای مشاهده پنجره طراحی فرم که بارها مورد استفاده قرار می‌گیرد، از میان‌برهای زیر استفاده کنید.

مشاهده دستورات فایل فرم	Ctrl + Alt +0
مشاهده پنجره طراحی فرم	Shift + F7

۱ــ Context menu
۲ــ Shortcut

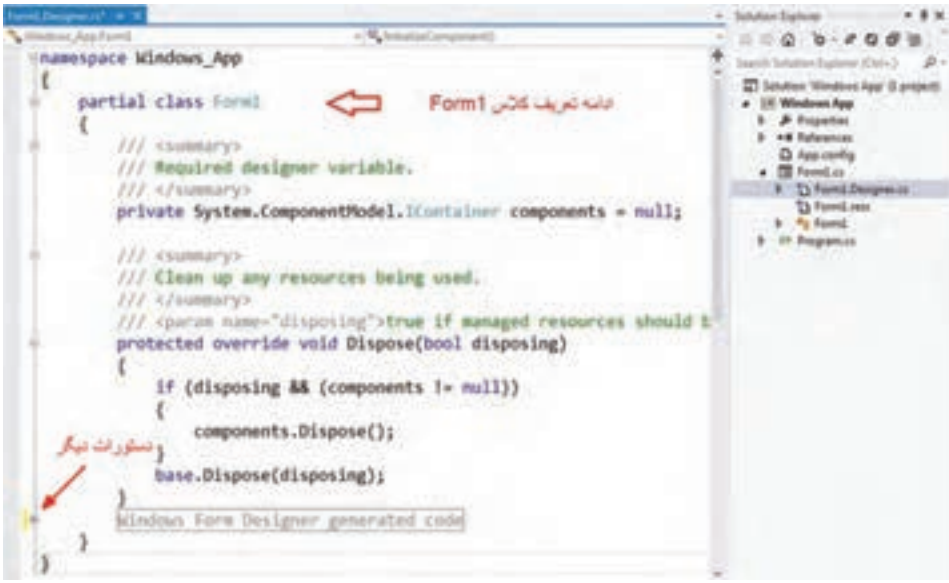
در منوی موضوعی فایل Form1.cs، اگر بر روی گزینه View Code کلیک کنید، محتوای فایل Form1.cs را مانند شکل ۲۰ــ۵ مشاهده خواهید کرد، که قسمتی از تعریف کلاس Form1 است.

شکل ۲۰ــ۵ــ محتوای دستورات و برنامه‌فایل Form1.cs

به خط عنوان تعریف کلاس Form1 توجه کنید، کلمه کلیدی partial در خط عنوان کلاس نشان می‌دهد که بخشی از تعریف کلاس Form1 در فایل دیگری قرار دارد. قسمت دیگر تعریف کلاس Form1 کجا است؟

فایل دیگری که VS ایجاد می‌کند و شما نباید محتویات آن را دستکاری کنید، فایل Form1.Designer.cs است. هنگامی که از جعبه ابزار سمت چپ صفحه، کنترلی را انتخاب و آن را در روی فرم قرار می‌دهید و ویژگی‌های آن را مطابق سلیقه خود تنظیم می‌کنید، VS تنظیمات شما را به دستورات و کدهای زبان C# تبدیل می‌کند. این دستورات به عنوان بخشی از تعریف کلاس Form1 به وسیلهٔ VS در فایل Form1.Designer.cs به‌طور خودکار نوشته می‌شود و کار شما را بسیار آسان می‌کند. بنابراین بقیه تعریف کلاس Form1 در این فایل قرار دارد. در پنجره Solution Explorer آن را پیدا کرده و با دو بار کلیک، با احتیاط آن را باز می‌کنیم. همان‌طور که در شکل ۲۱ــ۵ مشاهده می‌کنید ادامه تعریف کلاس Form1 دیده می‌شود. دستورات و محتویات این فایل، با

قراردادن کنترل‌ها بر روی فرم، افزایش می‌یابد. برای جلوگیری از دستکاری دستورات این فایل و در دسترس قرار ندادن آنها، مترجم آنها را در لحظه اول نمایش نمی‌دهد.



شکل ۲۱ــ۵ــ محتوای دستورات و برنامه فایل Form1.Designer.cs

برای نمایش دستورات ، در سمت چپ پنجره‌ای که محتوای فایل Designer را نشان می‌دهد، یک خط عمودی را مشاهده می‌کنید که در بین آن علامت‌های – و در بعضی قسمت‌ها علامت + قرار دارد. اگر بر روی علامت + کلیک کنید ، منطقه‌ای نمایان می‌شود که تعدادی از دستورات در آن قرار گرفته است.[۱] لطفا این دستورات را بدون آگاهی تغییر ندهید که در این صورت قسمت طراحی فرم دچار مشکل می‌شود.

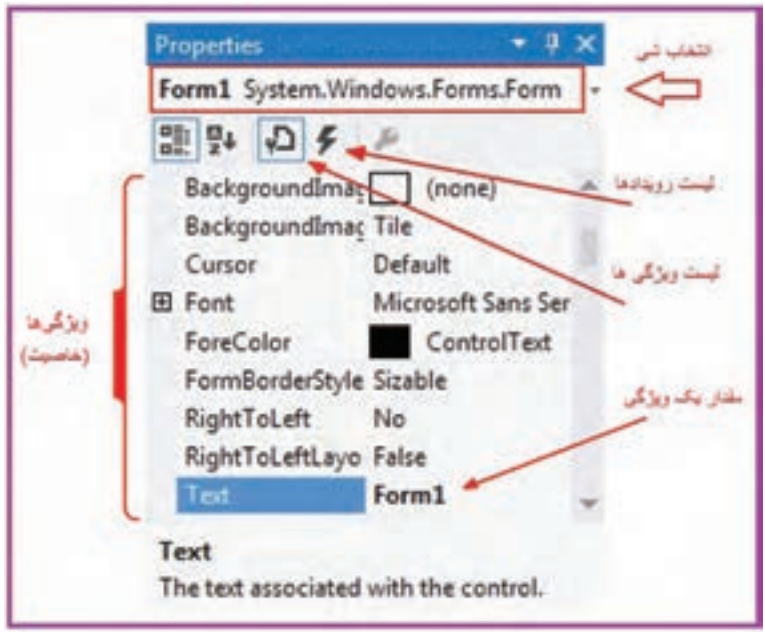
۲ــ۳ــ۵ــ پنجره ویژگی‌ها و خواص اشیاء

پنجره‌ای که در برنامه‌های ویندوزی از آن زیاد استفاده خواهیم کرد، پنجره Properties است که در شکل ۲۲ــ۵ مشاهده می‌کنید.

۱ــ با استفاده از راهنمای region# و endregion#، این امکان مهیا می‌شود که تعدادی دستور در لحظه اول نشان داده نشوند و در صورت لزوم با کلیک بر روی علامت +، نمایان شوند.

Formatted: Font: 11 pt, Complex Script Font: 11pt

Formatted: Font: 11 pt, Complex Script Font: 11pt



شکل ۲۲ــ۵ ــ پنجره Properties

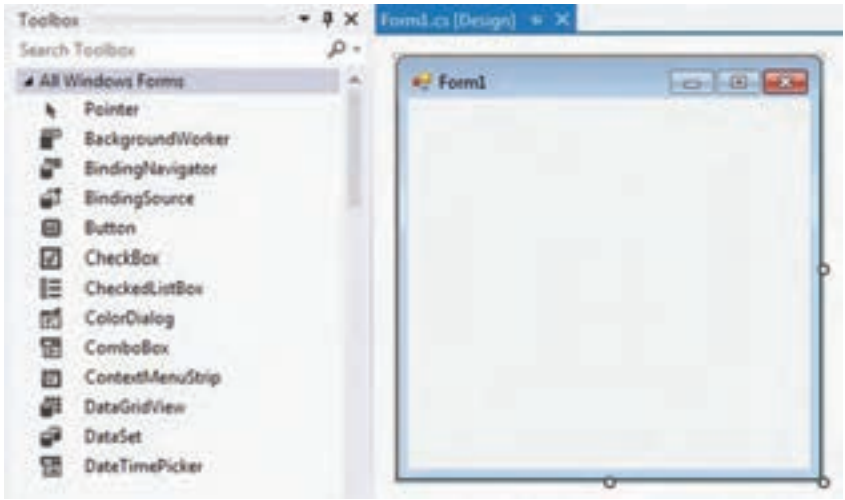
در این پنجره ویژگی‌ها و خواص شئ و همچنین واکنش به رویدادهای هر شئ را می‌توانیم مقداردهی و تنظیم کنیم. در قسمت بالای این پنجره، نام یک شئ که در فرم انتخاب شده، نشان داده شده است و از طریق مثلث کوچک کنار آن می‌توان شئ دیگری را از بین اشیای موجود روی فرم انتخاب کنیم. در زیر نام شئ، می‌توان با کلیک بر روی آیکنی که نظیر رعد و برق است، لیست رویدادهای شئ مشاهده و یا بر روی آیکن کنار آن کلیک کرد تا لیست ویژگی‌های شئ را مشاهده شود. در هر کدام از انتخاب‌ها، لیست نشان داده شده را می‌توان به ترتیب نام[1] و یا به ترتیب عملکرد که دسته‌بندی[2] شده است، مشاهده نمایید. در شکل ۲۳ــ۵، آیکن‌های مربوطه دیده می‌شوند.

شکل ۲۳ــ۵ــ آیکن‌های ترتیب نمایش ویژگی‌ها در پنجره Properties

۱ــ Alphabetical view ۲ــ Categorized view

۳ــ۳ــ۵ــ استفاده از جعبه ابزار و قراردادن کنترل‌ها در روی فرم

همان‌طورکه در فصل قبل بیان شد، یک برنامه ساده ویندوزی از یک فرم تشکیل می‌شود که به عنوان نگهدارنده[۱] برای قرار گرفتن اجزای گرافیکی دیگر، استفاده می‌شود. شکل ۲۴ــ۵ را مشاهده کنید.



شکل ۲۴ــ۵ــ پنجره طراحی و جعبه ابزار

در حالی که پنجره طراحی فرم را مشاهده می‌کنید، می‌توانید از جعبه ابزار استفاده کرده و کنترل مورد نظر خود را پیدا کنید. برای قرار دادن یک کنترل بر روی فرم، می‌توانید روی کنترل مورد نظر دوبار کلیک کنید و یا با درگ کردن کنترل و رها کردن آن روی فرم، کنترل مورد نظر را روی فرم بچسبانید. البته بعضی از کنترل‌ها مانند کنترل زمان‌سنج (تایمر)[۲]، با توجه به عملکردشان، ممکن است در قسمت پایین فرم قرار گیرند که با آنها در این کتاب آشنا می‌شوید.

اندازه فرم که برای طراحی استفاده می‌کنید قابل تغییر است. توجه داشته باشید که اگر در پنجره Form Designer، در روی فرم دوبار کلیک کنید، دستوراتی به فایل Form1.cs اضافه می‌شود که مربوط به الگوی، یک EH است که در شکل ۲۵ــ۵ نشان داده شده است، به دلیل اینکه به طور اتفاقی و اشتباهاً دوبار کلیک کرده‌اید، لذا دستورات اضافه شده را حذف کنید و یا با احتیاط، عمل انجام‌شده را لغو و یا بی‌اثر کنید.

۱ــ Container

۲ــ Timer Control

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

        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

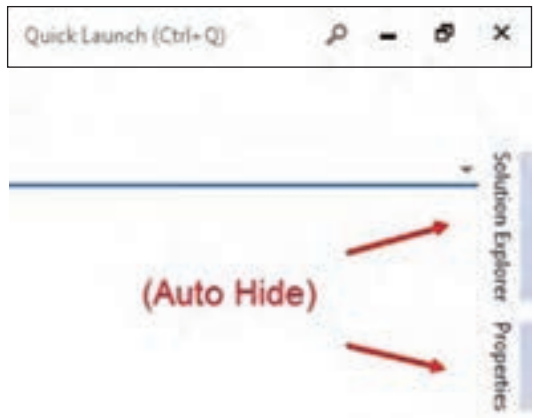
شکل ۲۵ــ۵ــ دستورات اضافه شده در اثر دو بار کلیک بر روی فرم

کار در کارگاه ۱: جابه‌جایی پنجره‌ها در VS

۱ــ برنامه VS را اجرا کنید و یک پروژه Windows Forms Application ایجاد کنید.

۲ــ پنجره Solution Explorer را در حالت AutoHide قرار دهید.

۳ــ پنجره Properties را نیز در حالت AutoHide مانند شکل ۲۶ــ۵ قرار دهید.

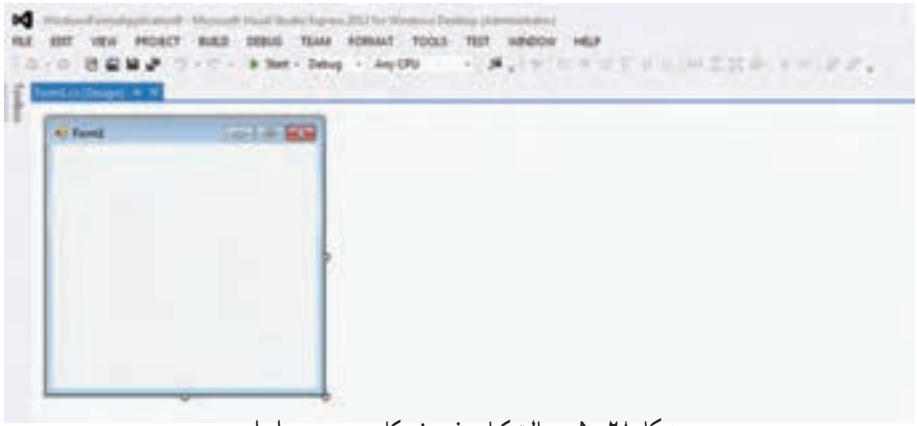


شکل ۲۶ــ۵ــ حالت کنار رفتن خودکار

۴ــ هر دو پنجره را از حالت Auto Hide خارج کرده و به حالت چسبیده[1] مانند شکل ۲۷ــ۵ قرار دهید.

شکل ۲۷ــ۵ــ پنجره‌ها در حالت چسبیده (Dock)

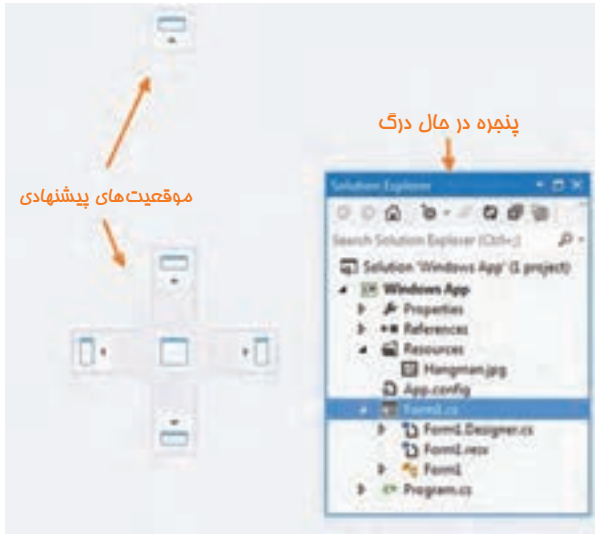
۵ــ پنجره Tool box را در حالت Auto Hide قرار دهید تا قسمت طراحی فرم مانند شکل ۲۸ــ۵ بزرگ‌تر شود. اگر از قبل در حالت Auto Hide است آن را به حالت چسبیده تغییر دهید.



شکل ۲۸ــ۵ــ حالت کنار رفتن خودکار پنجره جعبه ابزار

۱ــ Dock

۶ــ ماوس را روی خط عنوان پنجره Solution Explorer برده و با استفاده از درگ کردن، آن را به نقاط دیگر ببرید. توجه کنید که VS مکان‌هایی را مانند شکل ۲۹ــ۵ به شما پیشنهاد می‌دهد. از آنها استفاده و موقعیت‌های مختلف را تجربه کنید. سعی کنید که مفهوم موقعیت‌های پیشنهادی را بفهمید.



شکل ۲۹ــ۵ــ انتقال پنجره به نقاط دیگر صفحه

۷ــ برای اینکه هر دو پنجره Solution Explorer و Properties از کل فضای سمت راست صفحه، به طور مشترک استفاده نمایند، آنها را به صورت Tab Docked مانند شکل ۳۰ــ۵ قرار دهید. برای این منظور پنجره Properties را درگ کرده و وقتی روی خط عنوان پنجره Solution Explorer رسید، آن را رها کنید.

۸ــ مجدداً هر دو پنجره را به صورت چسبیده، مانند شکل ۲۷ــ۵ قرار دهید.

شکل ۳۰ــ۵ــ پنجره‌ها در حالت Tab Docked

اکنون که با پنجره‌های محیط طراحی برنامه ویندوزی در VS آشنا شدیم، می‌توانیم یک نمونه پروژه ویندوزی را قدم به قدم دنبال کنیم.

۴ــ۵ــ ساخت یک واسطه کاربری با استفاده از قابلیت طراحی تصویری در VS

هنگامی که از امکانات VS برای طراحی واسطه کاربری استفاده می‌کنید، بسیاری از دستورات و کدهای لازم، بدون اینکه شما متوجه شوید، به طور خودکار به وسیلهٔ VS تولید شده و در قالب فایل‌هایی که در ابتدای این فصل معرفی شد، به برنامه اضافه می‌شوند.

در قسمت کار در کارگاه ۲ ، مراحل ساخت یک نمونه فرم ساده را بدون نوشتن حتی یک خط برنامه، دنبال می‌کنیم. بدیهی است برای تولید فرم‌های کاربردی و واکنش به رویدادها، علاوه بر استفاده از ابزارها و امکانات VS ، باید چند خط برنامه نیز بنویسیم.

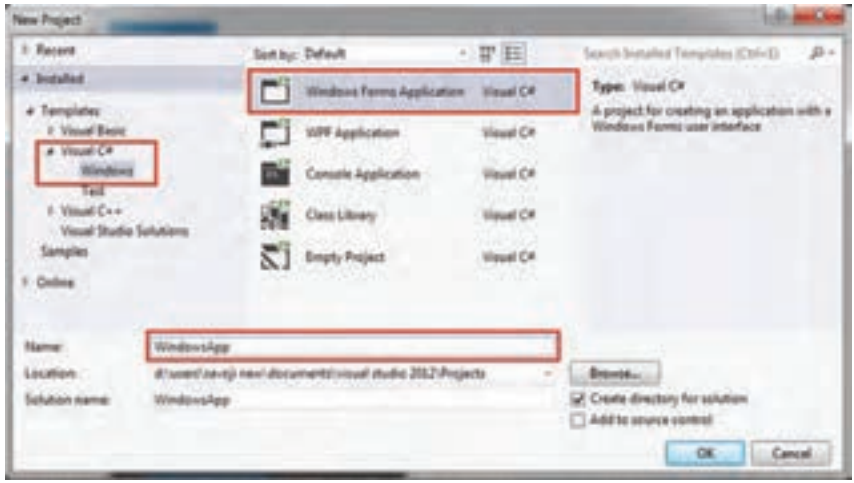
کار در کارگاه ۲: ساخت اولین فرم

ساخت یک برنامه یا یک فرم مانند شکل ۳۱ــ۵

شکل ۳۱ــ۵ــ اولین برنامه در ویژوال C#

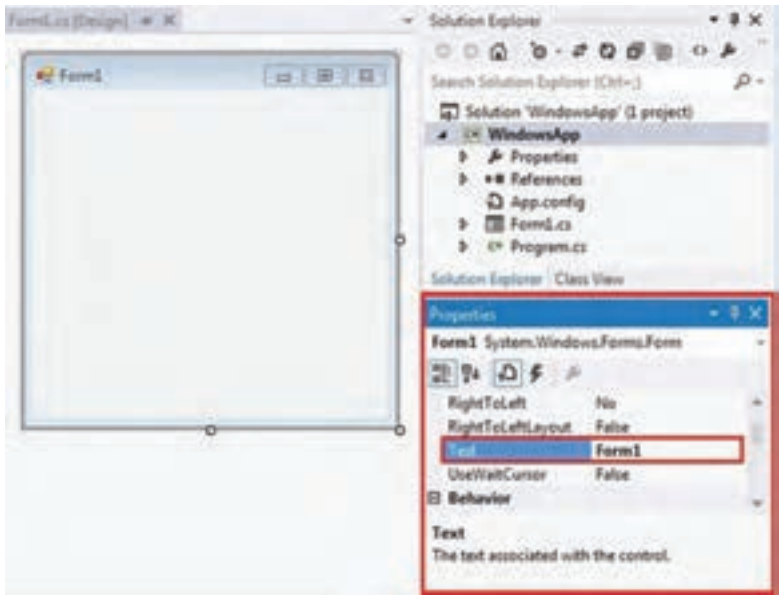
الگوریتم یا روش انجام کار : برای ایجاد چنین فرمی، مراحل زیر را به دقت انجام دهید :

۱ــ ایجاد یک پروژه ویندوزی: وارد برنامه VS شوید و یک پروژه جدید از نوع Windows Forms Application با نام پروژه WindowsApp و در مسیر مشخصی مانند شکل ۳۲ــ۵ بسازید. اگر در حال حاضر در داخل VS هستید از طریق منوی File، انتخاب گزینه Close Solution ، پروژه فعلی را ببندید و سپس یک پروژه جدید مطابق با آنچه که گفته شد، بسازید.



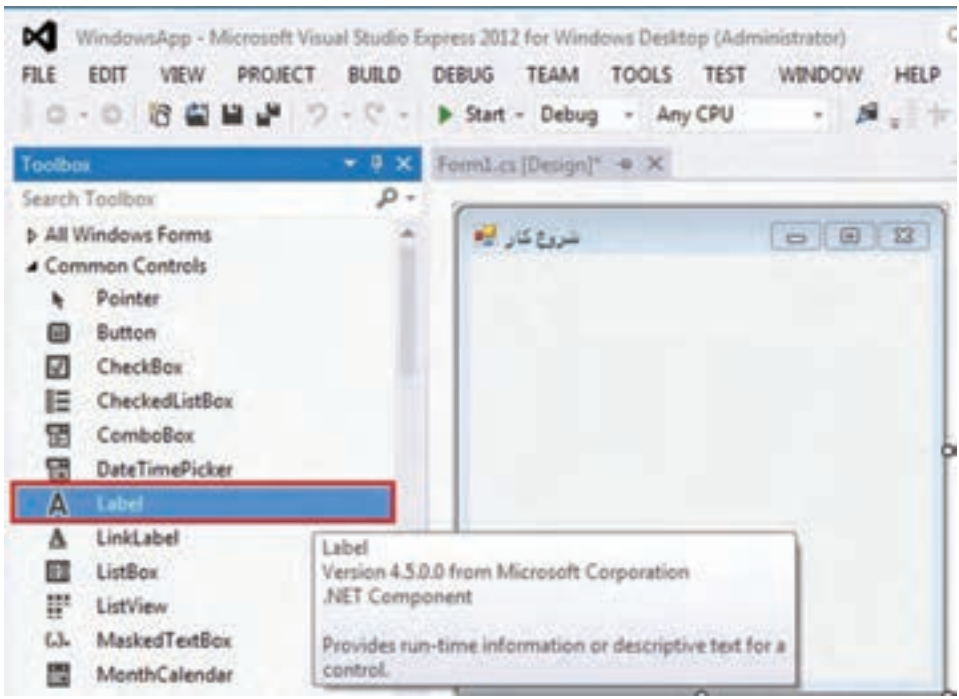
شکل ۳۲ــ۵ــ ایجاد پروژه جدید

۲ــ تعیین خط عنوان فرم: اگر به خط عنوان فرم در شکل ۳۱ــ۵ توجه کنید، عبارت فارسی «شروع کار» دیده می‌شود. برای تغییر عنوان فرم به عبارت خواسته شده، به پنجره ویژگی‌ها(Properties) رفته و مانند شکل ۳۳ــ۵ خاصیت Text را پیدا کرده و متن داخل آن یعنی "Form1" را پاک کرده و عبارت جدید «شروع کار» را تایپ کنید. اگر پنجره Properties دیده نمی‌شود از منوی View، بر روی گزینه Properties Window کلیک کنید و یا از میان بر مربوطه استفاده نمایید.



شکل ۳۳ــ۵ــ تغییر خاصیت Text مربوط به فرم از طریق پنجره Properties

۳ــ *اضافه کردن یک متن نمایشی بر روی فرم :* برای نمایش یک متن یا عبارت بر روی فرم، از کنترل برچسب (Label) استفاده می‌کنیم. به سراغ پنجره جعبه ابزار رفته و کنترل برچسب (Label) را پیدا کرده و آن را بر روی فرم قرار می‌دهیم. شکل ۳۴ــ۵ را مشاهده کنید.



شکل ۳۴ــ۵ــ کنترل برچسب در جعبه ابزار

با یکی از روش‌های زیر می‌توانید کنترل‌ها را بر روی فرم قرار دهید :

۱ــ با استفاده از دابل کلیک بر روی کنترل

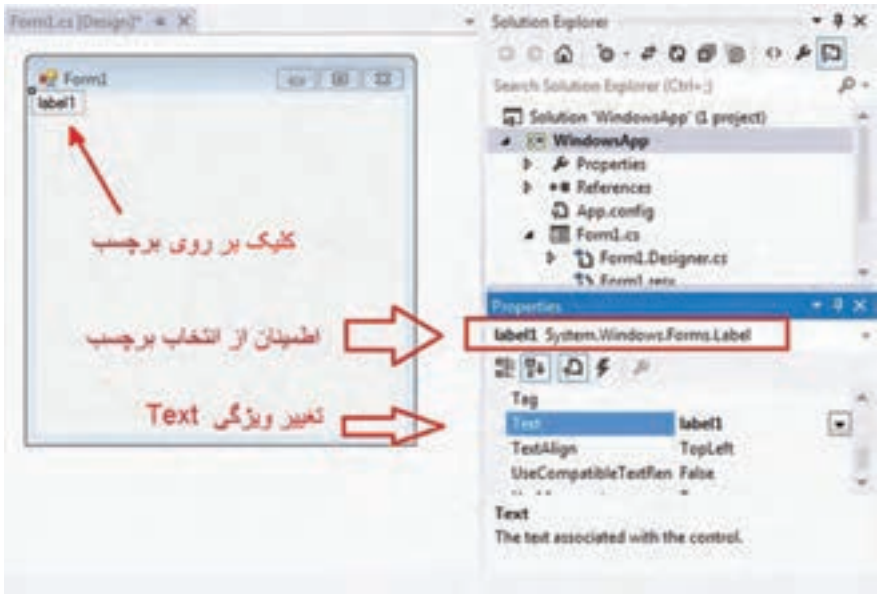
۲ــ با استفاده از روش Drag & Drop

۴ــ *تعیین ویژگی Text برای برچسب :* به پنجره طراحی فرم مراجعه کرده، بر روی برچسب یک بار کلیک کنید تا انتخاب شود. با انتخاب کنترل برچسب روی فرم، پنجره Properties، ویژگی‌های آن را نشان می‌دهد. نام این کنترل به طور خودکار به وسیلهٔ VS انتخاب می‌شود که label1 است. در واقع این نام، نام شئ است که از کلاس Label ایجاد و یا نمونه سازی شده است. در خط

اول پنجره Properties، نام شئ به همراه نام کلاس آن، نشان داده شده است.

(Label1 System.windows.Forms.Label)

حال در پنجره Properties، پس از اطمینان از اینکه برچسب انتخاب شده، خاصیت Text را مانند شکل ۳۵ــ۵ پیدا کنید.

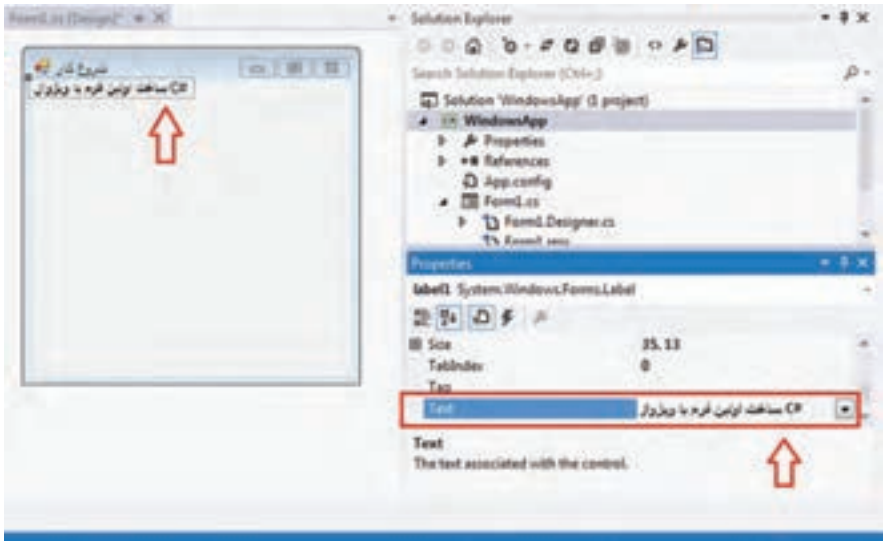


شکل ۳۵ــ۵ــ تغییر ویژگی متن کنترل برچسب

برای اطمینان از اینکه برچسب انتخاب شده است به قسمت بالای پنجره Properties توجه کنید، باید نام برچسب را مشاهده کنید. اگر چنین نیست روی فلش کوچک کنار آن کلیک کنید، نام برچسب را پیدا کرده و آن را انتخاب کنید.

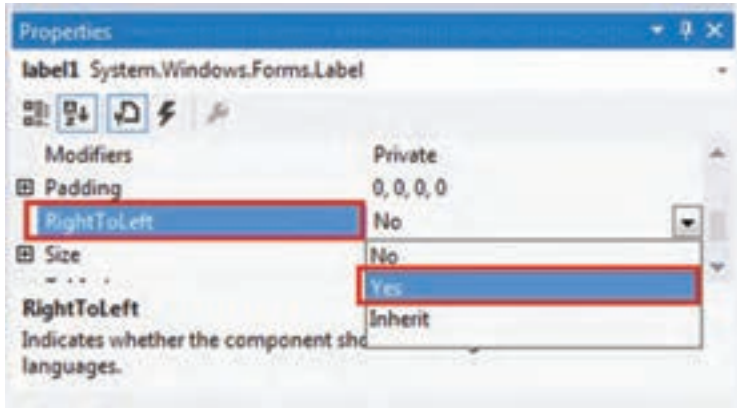
پس از انتخاب کنترل برچسب، عبارت «ساخت اولین فرم با ویژوال C#» را در قسمت Text برچسب وارد کنید.

همان‌طور که در شکل ۳۶ــ ۵ مشاهده می‌شود، عبارت مورد نظر در خاصیت Text برچسب، وارد شده اما به هم ریخته است. چون عبارت وارد شده، ترکیبی از کلمات فارسی و همچنین حروف انگلیسی است ، لذا مشکل به‌هم ریختگی کلمات پیش آمده و عبارت «C#» علاوه بر اینکه در جای خود (انتهای جمله) قرار ندارد به شکل برعکس یعنی «C#» دیده می‌شود. برای رفع این مشکل چه باید کرد؟



شکل ۳۶ــ۵ ــ متن شامل کلمات فارسی و انگلیسی

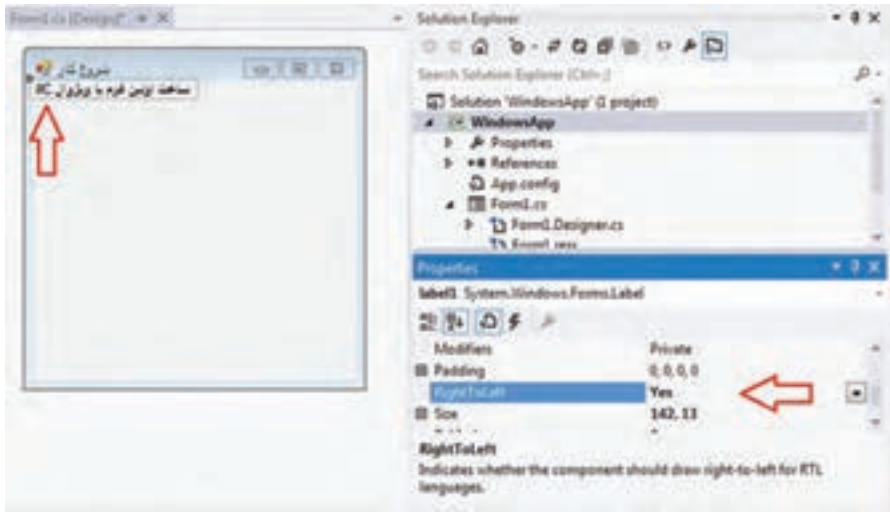
قبلاً با چنین مشکلاتی در برنامه‌هایی نظیر Word روبه‌رو شده بودید. در اغلب برنامه‌های کاربردی، از زبان‌های محاوره‌ای پشتیبانی می‌شود که در آنها کلمات و جملات از سمت راست به چپ نوشته می‌شود. در.NET نیز چنین است. در کنترل‌ها، خاصیتی به نام RightToLeft برای پشتیبانی از زبان‌های راست به چپ (RTL)[1]، پیش بینی شده است. بنابراین به سراغ خاصیت RightToLeft بروید و مقدار آن را به Yes تغییر دهید. شکل ۳۷ــ۵ را ملاحظه کنید.



شکل ۳۷ــ۵ ــ ویژگی RightToLeft برای پشتیبانی از زبان‌های RTL

۱ــ Right to Left Languages

با تغییر دادن مقدار ویژگی RightToLeft ، به مقدار Yes ، عبارت نوشته شده در برچسب، به صورت شکل ۳۸ــ۵ خواهد شد :



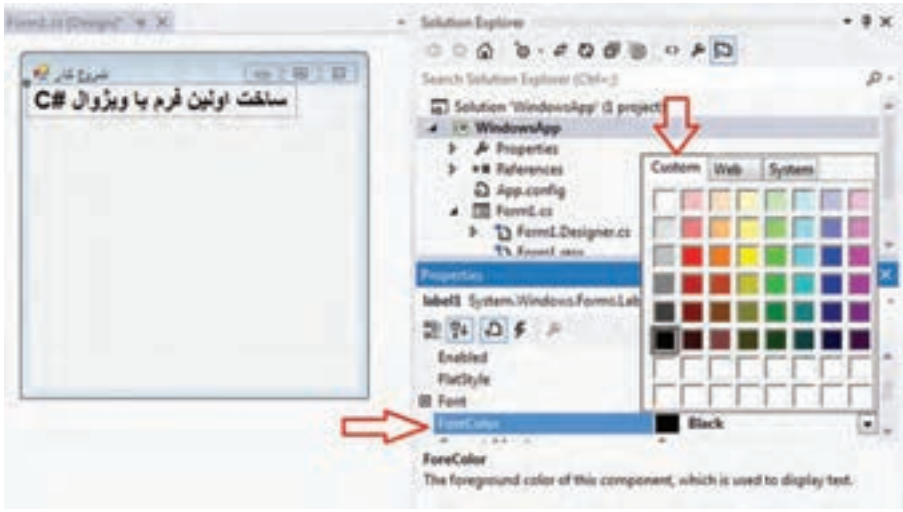
شکل ۳۸ــ۵ــ تغییر ویژگی RightToLeft

همان‌طور که در شکل ۳۸ــ۵ مشاهده می‌کنید، تنها اشکال باقیمانده، برعکس بودن عبارت "C#" است که برای تصحیح آن به خاصیت Text بروید و آن را به صورت "C#" تایپ کنید.

۵ــ تعیین ویژگی‌های فونت ، رنگ و محل قرارگیری برچسب : برای تعیین نوع فونت کنترل برچسب، از خاصیت Font در پنجره Properties استفاده کنید و روی علامت سه نقطه کلیک کنید تا لیست فونت‌های موجود در کامپیوتر ظاهر شود. پنجره‌ای مانند شکل ۳۹ــ۵ مشاهده می‌شود که در آن می‌توانید، فونت دلخواه را به همراه اندازه و سبک آن انتخاب کنید. سعی کنید، فونتی را انتخاب کنید که نوشته‌های فارسی خوب دیده شوند.

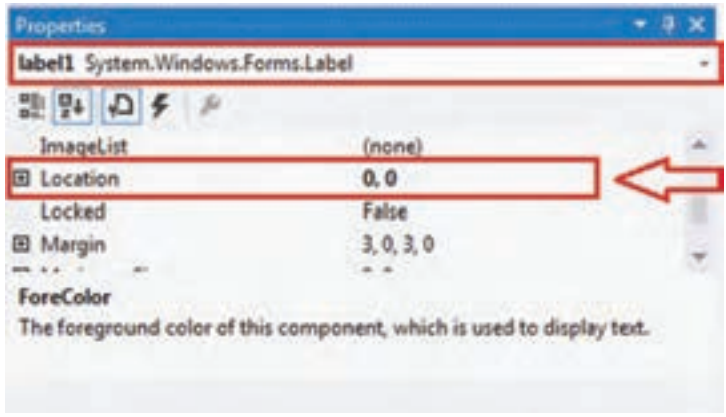
شکل ۳۹ــ۵ــ پنجره فونت

همچنین برای تعیین رنگ نوشته از خاصیت ForeColor استفاده می‌کنیم. پس از کلیک بر روی علامت فلش ، پنجره ای برای تعیین رنگ باز می‌شود، در آن پنجره مطابق شکل ۴۰ــ۵ ، بر روی سربرگ Custom کلیک کنید.



شکل ۴۰ــ۵ ــ پنجره رنگ

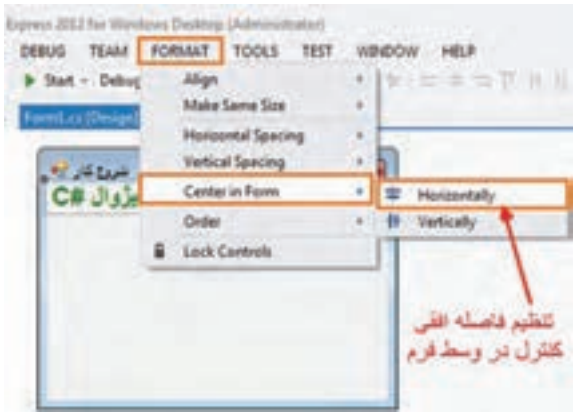
حال می‌خواهیم محل قرارگیری کنترل را تنظیم کنیم. قبل از انجام این‌کار که به سادگی با استفاده از ماوس می‌توانید انجام دهید، به پنجره Properties مراجعه کنید و ویژگی Location را مطابق شکل ۴۱ــ۵ پیدا کرده، روی علامت + کنار آن کلیک کنید.



شکل ۴۱ــ۵ ــ ویژگی موقعیت

در جلوی ویژگی Location، دو عدد ۰ و ۰ نوشته شده است. این اعداد فاصله کنترل را از سمت چپ فرم و از بالای فرم بر حسب پیکسل[1] مشخص می‌کنند. به عبارت دیگر موقعیت یک کنترل نسبت به گوشه بالا سمت چپ[2] فرم سنجیده می‌شود. بر خلاف آنچه که در ریاضیات معمول است که موقعیت یک نقطه را نسبت به مرکز صفحه (مبدأ) در نظر می‌گیرند، در کامپیوتر مبدأ سنجش موقعیت، گوشه بالا سمت چپ صفحه است.

برای تنظیم محل قرارگیری کنترل می‌توانید از ماوس استفاده کنید و یا اعداد مناسبی را به عنوان فاصله از چپ و بالا مشخص کنید. در اینجا با استفاده از ماوس، کنترل برچسب را گرفته و در نقطه دلخواه فرم قرار دهید به‌طوری که از دو طرف فرم به یک اندازه باشد. همچنین می‌توانید مانند شکل ۴۲ــ۵، از منوی Format گزینه Center in Form استفاده کنید تا با دقت بیشتری کنترل را در وسط فرم قرار دهید.



شکل ۴۲ــ۵ــ تنظیم موقعیت افقی با استفاده از منوی Format

نکته

هنگامی‌که بیش از یک کنترل در روی فرم قرار داشته باشد، اگر آنها را با هم انتخاب کنید، ابزارهای تنظیم موقعیت آنها نسبت به یکدیگر که در روی نوار ابزار و همچنین در منوی Format وجود دارد فعال شده، به سادگی و با دقت و سرعت عمل بالا می‌توانید آنها را تنظیم کنید.

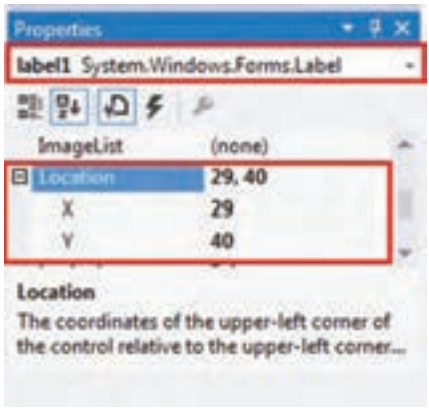
شکل ۴۳ــ۵

۱ــ Pixel

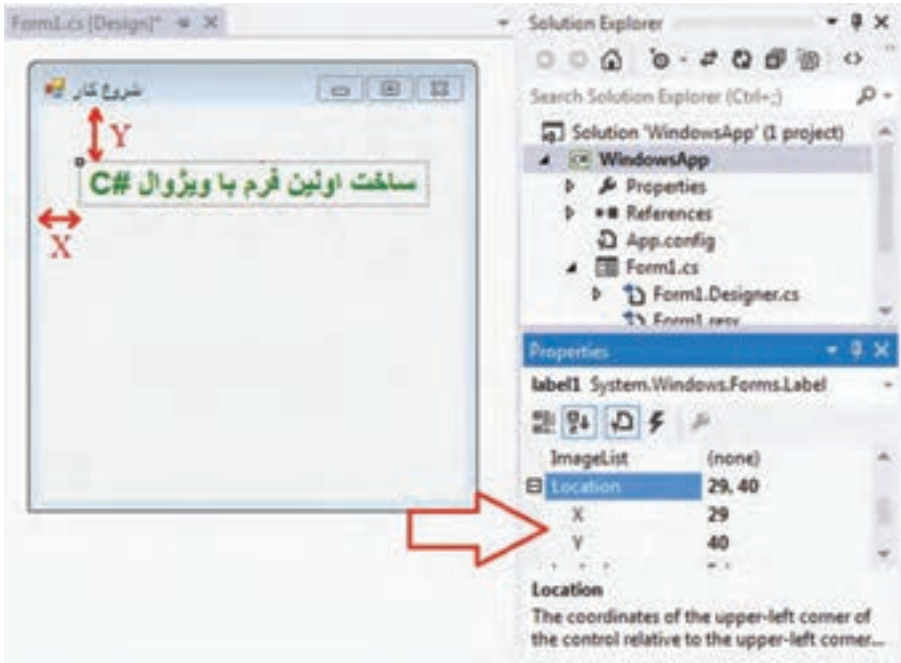
۲ــ Upper_left corner

بعد از جا به جایی کنترل برچسب و قرارگرفتن در یک موقعیت مناسب، به پنجره Properties رفته و مقدار اعداد در ویژگی Location را مشاهده کنید.

همان‌طور که در شکل ۴۴ــ۵ مشاهده می‌کنید، در اثر جا به جایی کنترل، مقدار ۰ و ۰ به اعداد دیگری مانند ۴۰ ,۲۹ تغییر کرده است. ارتباط این دو عدد با محل قرار گیری برچسب نسبت به فرم در شکل ۴۵ــ۵ نشان داده شده است. فاصله افقی[1] نسبت به لبه سمت چپ فرم با مقدار X و فاصله عمودی[2] نسبت به لبه بالای فرم با مقدار Y مشخص می‌گردد.



شکل ۴۴ــ۵ــ مقادیر ویژگی موقعیت کنترل برچسب



شکل ۴۵ــ ۵ــ ارتباط مقادیر ویژگی موقعیت با لبه‌های فرم

۱ــ Horizontally

۲ــ Vertically

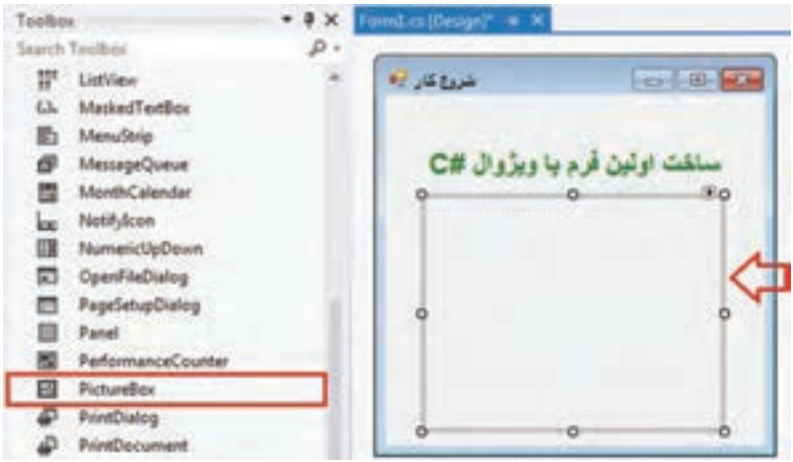
قبل از رفتن به مرحله بعدی بهتر است یک بار ویژگی‌هاو ویژگی‌های کنترل برچسب را که تنظیم کرده ایم در جدول ۱ــ۵ مشاهده کنید. رنگ نوشته و نوع فونت ممکن است با آنچه شما در انجام این مثال انجام می‌دهید، متفاوت باشد که اهمیتی ندارد.

جدول ۱ــ۵ــ مقادیر ویژگی‌های کنترل برچسب

کنترل Label	
ویژگی	**مقدار**
Name	label1
Text	ساخت اولین فرم با ویژوال C#
RightToLeft	Yes
Font	Arial Size : 16 Bold
ForeColor	Green
Location	Center in Form | Horizontally

در سطر اول جدول ۱ــ۵، نام کنترل نوشته شده است و در ستون اول نام ویژگی‌های کنترل و در ستون دوم مقدار هر ویژگی بیان شده است. با توجه به اینکه جدول ۱ــ۵ به طور واضح و گویا ویژگی‌های کنترل برچسب را نشان می‌دهد، از چنین جداولی در انجام پروژه‌های بعدی و یا حتی در امتحانات عملی، برای مشخص کردن ویژگی‌های هر کنترل که باید تنظیم شود، استفاده خواهیم کرد و دیگر لازم به ارائه توضیحات نیست.

۶ــ اضافه کردن عکس بر روی فرم : در این مرحله باید عکسی را به فرم اضافه نماییم. برای نمایش یک عکس می‌توانید از کنترل PictureBox، استفاده کنید که یک کنترل جالب و کاربردی است و آن را کنترل جعبه تصویر می‌نامیم. برای قراردادن این کنترل بر روی فرم، به جعبه ابزار مراجعه کرده ، آن را پیدا کرده و به فرم اضافه کنید. شکل ۴۶ــ۵ را مشاهده کنید.

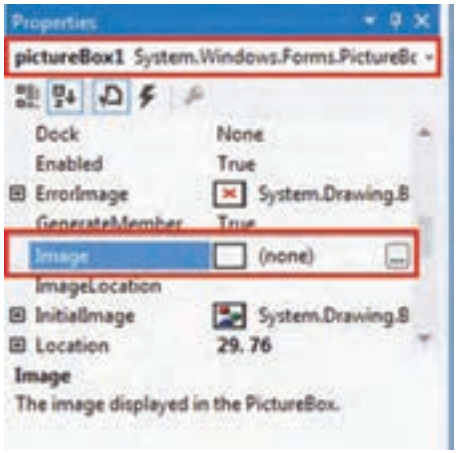


شکل ۴۶ــ۵ ــ اضافه کردن کنترل جعبه تصویر به فرم

به محض قرار گیری این کنترل بر روی فرم، پنجره Properties به طور خودکار، ویژگی‌ها این کنترل را نشان می‌دهد. نام این شئ pictureBox1 است که به وسیلهٔ VS انتخاب می‌شود و از کلاس PictureBox ساخته شده است.

برای تعیین تصویری که می‌خواهیم در جعبه تصویر نمایش داده شود، از ویژگی Image این کنترل استفاده می‌کنیم. برای مقداردهی ویژگی Image دو روش وجود دارد :

۱ــ استفاده از پنجره Properties، انتخاب شئ PictureBox و سپس کلیک روی علامت... (شکل ۴۷ــ۵).



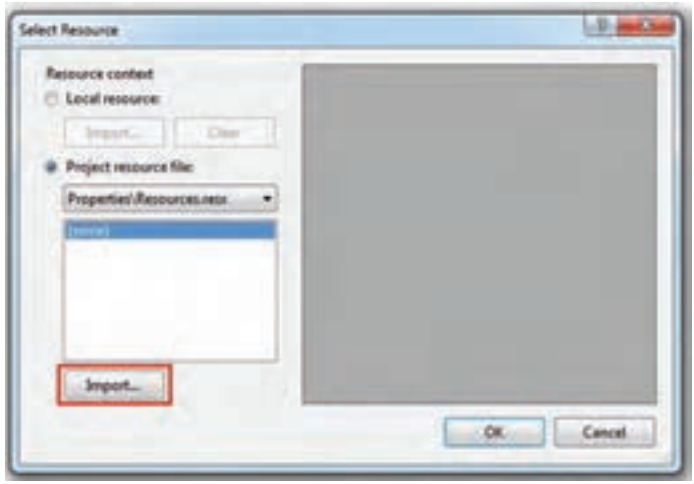
شکل ۴۷ــ۵ــ ویژگی Image برای تعیین تصویر

۲ــ کلیک روی مثلث کوچک سیاه رنگ در کادر اطراف کنترل جعبه تصویر و استفاده از گزینه Choose Image

در کادر اطراف کنترل جعبه تصویر، یک مثلث کوچک سیاه رنگ دیده می‌شود. با کلیک بر آن، منویی مانند شکل ۴۸ــ۵ ظاهر می‌گردد که در آن چند ویژگی کنترل نشان داده می‌شود.

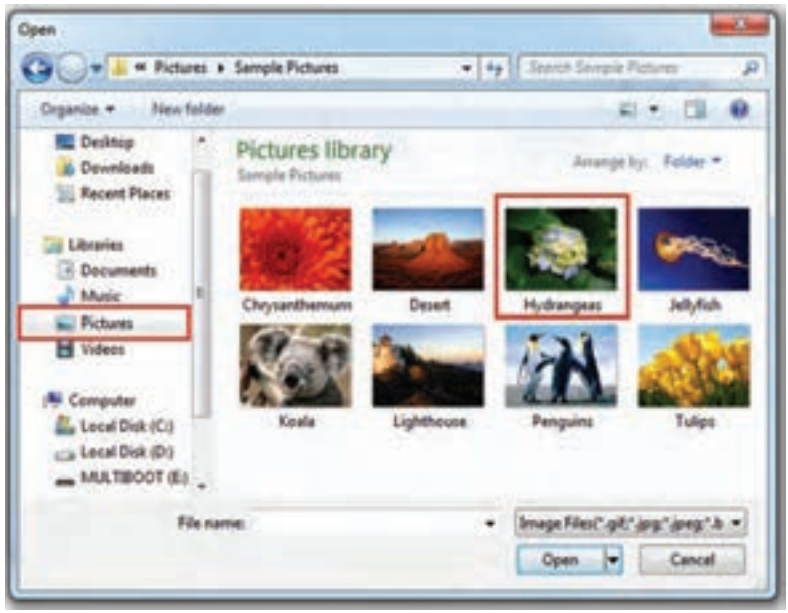
شکل ۴۸ــ۵ ــ کلیک بر روی مثلث و ظاهر شدن منو

به وسیلهٔ گزینه Choose Image در شکل ۴۸ــ۵ نیز علاوه بر پنجره Properties، می‌توانید ویژگی Image را تعیین کنید. در هر حال با انتخاب گزینه Choose Image، پنجره‌ای به نام انتخاب منبع (Select Resource)، شکل ۴۹ــ۵ باز می‌شود که در آن باید فایل مورد نظر برای اضافه‌شدن به پروژه را مشخص کنید. اگر در شکل ۴۹ــ۵ بر روی کلید Import کلیک کنید پنجره انتخاب فایل باز می‌شود تا یک کپی از فایل مورد نظر شما به لیست فایل‌های پروژه اضافه شود.



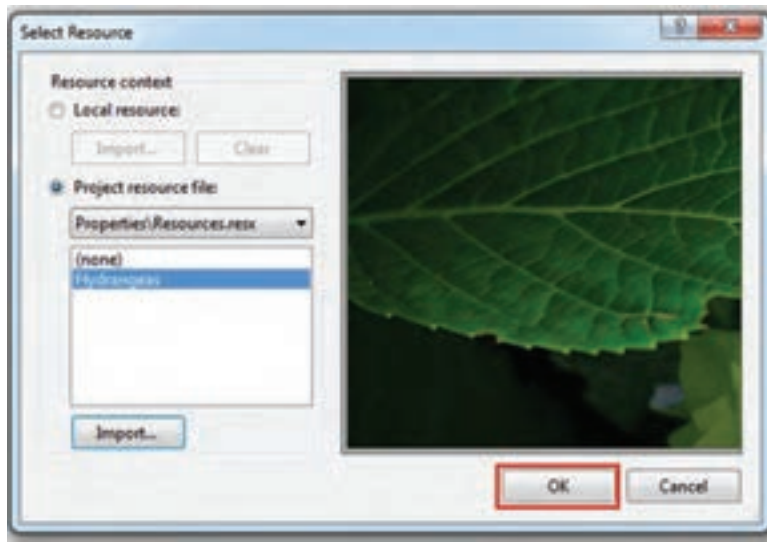
شکل ۴۹ــ۵ ــ انتخاب فایل تصویری جهت اضافه شدن به پروژه

پس از اینکه پنجره مربوط به تعیین نام فایل ظاهر شد وارد پوشهٔ Pictures شوید و یک فایل تصویری انتخاب کنید. در این مثال تصویر گل ادریسی انتخاب شده است.



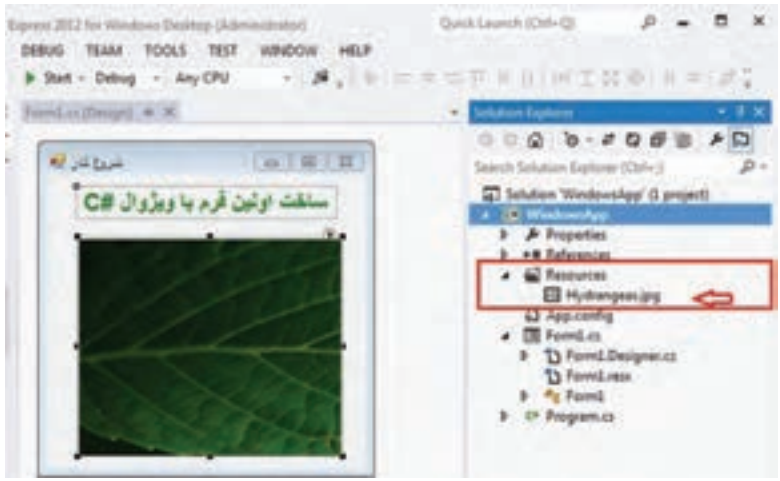
شکل ۵۰ــ۵ــ انتخاب فایل تصویری جهت نمایش در کنترل جعبه تصویر

پس از انتخاب فایل مورد نظر، به پنجره انتخاب منابع باز می‌گردید. با کلیک روی دکمه OK عملیات انجام شده را تأیید کنید (شکل ۵۱ ــ۵).



شکل ۵۱ــ۵ــ پنجره انتخاب منبع پس از انتخاب فایل تصویری

پس از انتخاب فایل تصویری و کلیک بر روی دکمه OK ، تصویر انتخاب شده در جعبه تصویر قرار می‌گیرد. اگر مانند این مثال، اندازه عکس، بزرگ‌تر از اندازه جعبه تصویر باشد، فقط بخشی از آن مانند شکل ۵ــ۵۲ دیده خواهد شد. نگران نباشید این مشکل را برطرف می‌کنیم!



شکل ۵ــ۵۲ــ اضافه شدن فایل تصویری در فولدر Resources

اگر به پنجره Solution Explorer در شکل ۵ــ۵۲ ، دقت کنید فایل تصویر انتخاب شده در پوشه Resources دیده می‌شود. اگر از طریق My Computer به مسیری بروید که در ابتدا، برای پروژه تعیین کرده‌اید، پوشه‌ای به نام Resources را پیدا خواهید کرد که در داخل آن یک کپی از فایل تصویر انتخاب شده قرار دارد. به همین دلیل به این پوشه ، پوشه منابع گفته می‌شود.

برای برطرف کردن مشکل عدم تطبیق اندازه عکس و اندازه جعبه تصویر چه کنیم؟

شاید راه حل‌های مختلفی برای برطرف کردن اِشکال فوق، از جمله کاهش اندازه عکس به وسیلهٔ برنامه‌های گرافیکی نظیر Paint و Photoshop و یا بزرگ کردن اندازه جعبه تصویر به طوری که بتواند تمام عکس را در خود جای دهد، به نظر برسد. اما بهترین راه حل این است که از ویژگی SizeMode کنترل جعبه تصویر استفاده کنیم و مقدار آن را مانند شکل ۵ــ۵۳ برابر StretchImage قرار دهیم.

شکل ۵۳ــ۵ــ تغییر خاصیت SizeMode به StretchImage

حال برنامه را اجرا می‌کنیم ، خروجی آن مانند شکل ۵۴ ــ ۵ است.

شکل ۵۴ــ۵ــ خروجی اولین برنامه نوشته شده در محیط Visual C#

تبریک می‌گوییم که توانستید اولین برنامه ویندوزی با استفاده از VS را بنویسید. در این مثال یک خط برنامه ننوشتید و همه دستورات این برنامه به وسیلهٔ VS به طور خودکار انجام شد. بر روی فایل Form1.Designer.cs دوبار کلیک کنید و مستقیم به سراغ متد ()InitializeComponent بروید و روی علامت + کلیک کنید تا محتوای آن را ببینید. توجه داشته باشید که محتویات این فایل، بخصوص این متد را نباید بدون اطمینان و آگاهی تغییر دهید و یا دستکاری کنید. شکل ۵۵ــ۵ را مشاهده کنید. دستورات مربوط به برچسب و جعبه تصویر مشخص شده است. دستورات قسمت پایین در شکل ۵۵ــ۵ مربوط به چیست؟

```
/// Required method for Designer support. do not modify
/// the contents of this method with the code editor. محتویات این متد را با ادیتور تغییر ندهید
/// </summary>
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.label();
    this.pictureBox1 = new System.Windows.Forms.pictureBox();
    ((System.ComponentModel.IsupportInitialize)(this.pictureBox1)).BeginInit();
    this.SuspendLayout();
    //label1
    //
    this.label1.AutoSize = true;
    this.label1.Font = new System.Drawing.Font("Arial", 15.75F, System.Drawing.
    this.label1.ForeColor = System.Drawing.Color.Green;
    this.label1.Location = new System.Drawing.Point(12,13);
    this.label1.Name = "label1";
    this.label1.RightTOLeft = System.Windows.Forms.RightTOLeft.Yes;
    this.label1.Size = new System.Drawing.Size(227,24);
    this.label1.TabIndex = 0;
    this.label1.Text = "ساخت اولین فرم با ویژوال #C";
    //
    // picturBox1
    //
    this.pictureBox1.Imag=global::WindowsApp.Properties.Resources.Hydrangeas
    this.pictureBox1.Location = new System.Drawing.Point(12, 57);
    this.pictureBox1.Name = "pictureBox1";
    this.pictureBox1.Size = new System.Drawing.Size(233, 193);
    this.pictureBox1.SizeMode=System.Windows.Forms.PictureBoxSizeMode.Stretcr
    this.pictureBox1.TabIndex = 1;
    this.pictureBox1.TabStop = false;
    //
    // Form1
    //
```

مقداردهی ویژگی‌های برچسب

مقداردهی ویژگی‌های جعبه تصویر

```
this. AutoScaleDimensions = new System. Drawing. SizeF (6F, 13F);
this. AutoScaleMode = System.Windows. Forms. AtoScaleMode.Font;
this. ClientSize = new System. Drawing. Size (284, 262);
this. Controls. Add (this.pictureBox1);
this. Controls. Add (this.label1);
this. Name = "Form1" ;
this. Text = «شروع کار» ;
((System. ComponentMode 1. IsupportInitialize) (this.pictureBox1)). EndInit ( );
this. ResumeLayout (false);
this. PerformLayout ( );
```

شکل ۵۵ــ۵ــ محتویات متد InitalizeComponent() در فایل Form1.Designer.cs

برای مطالعه

کنترل‌ها و ابزارهای شخص ثالث برای.NET

علاوه بر کنترل‌هایی که در جعبه ابزار VS وجود دارد، کنترل‌های دیگر و ابزارهای پیشرفته‌ای نیز توسط شرکت‌های نرم‌افزاری که با شرکت مایکروسافت همکاری دارند، ساخته شده‌اند. به این نوع کنترل‌ها که قابلیت اضافه شدن به فرم‌های ویندوزی را دارند و توسط شرکت دیگری ساخته می‌شود، کنترل‌های شخص ثالث نامیده می‌شوند. بعضی از این ابزارها و کنترل‌ها در محیط ویژوال استودیو اضافه می شوند.

شرکت DevExpress از جمله شرکت‌هایی است که مجموعه کنترل‌ها و ابزارهای واسطه کاربری را برای تولیدکنندگان برنامه‌های کاربردی، از جمله برنامه‌نویسانی که از ویژوال استودیو استفاده می‌کنند، فراهم و به صورت تجاری عرضه کرده است. با استفاده از این کنترل‌ها، واسطه‌های کاربری زیباتری، بدون کدنویسی می‌توان ایجاد کرد.

یکی از ابزارهایی که این شرکت به رایگان عرضه می‌نماید، CodeRush Express است

که می‌توانید آن را از آدرس زیر دانلود نمایید. با نصب این ابزار در محیط VS، امکانات بیشتری در هنگام برنامه‌نویسی فراهم می‌شود. مثلاً منوی Refactoring با گزینه‌های متنوعی را در اختیار خواهید داشت.

https: //www.devexpress.com/Products/CodeRush/

کار در کارگاه ۳: تنظیم اندازه کنترل

تنظیم خودکار اندازه کنترل‌ها در هنگام اجرای برنامه

۱ــ پروژه WindowsApp که در این فصل ساختید را اجرا کنید.

شکل ۵۶ــ۵ ــ تغییر اندازه پنجره برنامه در حال اجرا

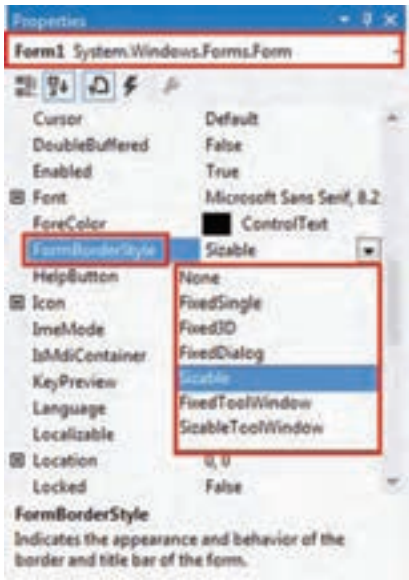
۲ــ همان‌طورکه در شکل ۵۶ــ۵ مشاهده می‌کنید، با استفاده از ماوس گوشه سمت راست پایین پنجره برنامه را گرفته و به سمت راست بکشید تا پنجره برنامه بزرگ‌تر شود.

شکل ۵۷ ــ ۵

همان‌طور که در شکل ۵ــ۵۷ مشاهده می‌کنید، در اثر تغییر اندازه پنجره، فضای خالی در سمت راست ایجاد می‌شود. اما کاربران انتظار دارند با بزرگ شدن پنجره، اندازه تصویر نیز افزایش یابد و صفحه پر شود. ولی در برنامه بالا چنین نیست و توازن صفحه به هم می‌خورد.

برای رفع اشکال بالا، دو راه حل داریم :

الف) کاربر اجازه تغییر اندازه پنجره برنامه را نداشته باشد. در واقع صورت مسئله را پاک می‌کنیم و به کاربر اجازه نمی‌دهیم که اندازه فرم را تغییر دهد تا چنین مشکلی به وجود آید. فرم‌ها دارای ویژگی به نام FormBorderStyle هستند که به وسیلهٔ آن می‌توانیم سبک یا شکل ظاهری کادر دور فرم را تعیین کنیم. در بعضی از سبک‌ها از امکان تغییر اندازه پنجره در حین اجرای برنامه جلوگیری می‌شود. مقدار پیش فرض این ویژگی Sizable است، یعنی فرم قابل تغییر اندازه است. برای انتخاب سبک‌های دیگر در حالی که فرم انتخاب شده است، در پنجره Properties به سراغ ویژگی FormBorderStyle رفته (شکل ۵ــ۵۸) و مقدار دیگری مانند FixedSingle که به معنای اندازه ثابت فرم و با کادر تک خطی می‌باشد را انتخاب کنید.



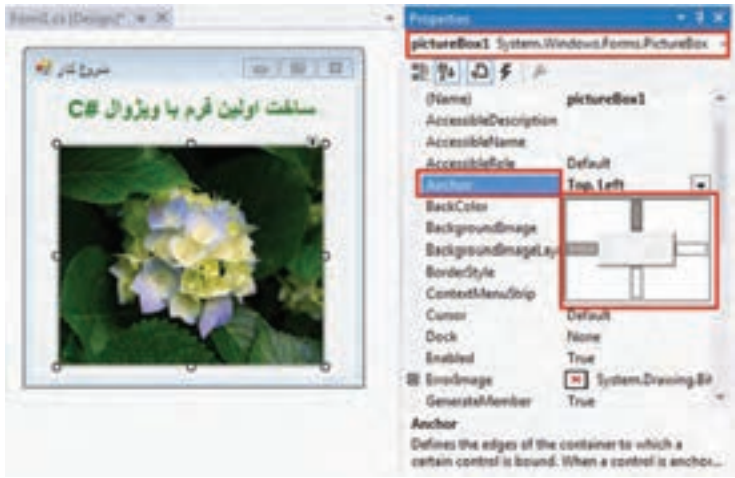
شکل ۵ــ۵۸ــ ویژگی FormBorderStyle فرم

نکته

می‌توانید منوی کنترل پنجره و آیکن‌های بیشینه، کمینه و خروج را پنهان کنید . برای این منظور ویژگی ControlBox مربوط به فرم را برابر false کنید. البته در این صورت کاربر، مجبور است برای پایان دادن به اجرای برنامه از ترکیب کلیدی Alt +F4 استفاده کند.

ب) در روش دوم محدودیتی برای کاربر ایجاد نکرده و مانع تغییر اندازه فرم نمی‌شویم. بنابراین باید اندازه جعبه تصویر در هنگام تغییر اندازه پنجره، به طور خودکار افزایش یابد. و به عبارت جالب‌تر، باید فاصله جعبه تصویر از چهار لبه فرم همواره ثابت بماند. خیلی نگران نباشید در VS برای انجام این

کار نیازی به برنامه نویسی خود شما نیست و فقط کافی است در محیط طراحی، از ویژگی Anchor کنترل جعبه تصویر برای این منظور استفاده کنید. این ویژگی‌ها می‌تواند مقادیر Left , Right , Top , Bottom و یا ترکیبی از آنها را بپذیرد. هر یک از مقادیر باعث می‌شوند که فاصله کنترل از سمت مربوطه ثابت بماند. در حالت پیش فرض مقدار ویژگی Anchor کنترل جعبه تصویر برابر Left, Top است، بنابراین فاصله کنترل از دو طرف چپ و بالای فرم ، ثابت باقی می‌ماند. شکل ۵۹ــ۵ را مشاهده کنید.



شکل ۵۹ــ۵ــ ویژگی Anchor

شکل ۶۰ــ۵ــ تغییر اندازه پنجره برنامه در حال اجرا پس از تنظیم ویژگی Anchor کنترل pictureBox

اگر در ویژگی Anchor مربوط به جعبه تصویر هر چهار مقدار را قرار دهید (از ماوس کمک بگیرید و روی دو کادر خط چین راست و پایین در شکل ۵۹ــ۵ کلیک کنید تا رنگ آنها خاکستری شود)، فاصله آن از چهار طرف ثابت می‌ماند. به عبارت دیگر اندازه آن هماهنگ با افزایش اندازه پنجره تغییر می‌کند. شکل ۶۰ــ۵ پنجره برنامه را پس از افزایش توسط کاربر نشان می‌دهد.

۳ــ در شکل فوق مشاهده می‌کنید که برچسب در وسط پنجره قرار ندارد. برای اینکه موقعیت برچسب نیز بتواند با تغییر اندازه پنجره تغییر کند و همواره در وسط باقی بماند لازم است فقط فاصله

آن تا بالای پنجره ثابت باشد. البته توجه داشته باشید برچسب باید از ابتدا در وسط فرم باشد. بنابراین برچسب را انتخاب کرده و ویژگی Anchor آن را برابر Top قرار دهید تا به طور خودکار موقعیت آن همواره در وسط پنجره قرار گیرد.

۴ــ برنامه را اجرا کنید و اندازه پنجره را تغییر دهید. آیا اندازه تصویر و موقعیت برچسب نیز مطابق با آن به طور خودکار تغییر می‌کند و کار رضایت بخش است؟

۵ــ در خیلی از برنامه‌های کاربردی نظیر Word و Excel که در سال قبل آشنا شدید، برای وارد کردن اطلاعات به زبان فارسی و یا نمایش صحیح اطلاعات فارسی، جهت صفحه را به حالت *راست به چپ*[1] تغییر می‌دادید. در فرم‌ها و کنترل‌هایی که برای نمایش و یا دریافت اطلاعات فارسی به‌کار می‌روند نیز بهتر است ظاهر و ترتیب قرارگیری اجزای فرم و همچنین ترازبندی از راست به چپ تنظیم شوند. مثلاً در شکل ۶۱ــ۵ خط عنوان فرم به صورت راست به چپ تنظیم شده است.

شکل ۶۱ــ۵ ــ خط عنوان از راست به چپ

همان‌طور که در شکل مشاهده می‌کنید، خط عنوان فرم تغییر کرده است و حالت آینه‌ای نسبت به فرم‌های معمولی دارد. در این فرم، آیکن‌های بستن پنجره ، بیشینه و کمینه در جهت سمت چپ قرار گرفته‌اند.

در فرم‌های ویندوزی خاصیت RightToLeft و خاصیت RightToLeftLayout برای پشتیبانی از زبان‌های راست به چپ ، پیش بینی شده است. برای ایجاد چنین حالتی، باید مقدار هر دو خاصیت را به true تغییر دهید.

---

۱ ــ Right to Left

خودآزمایی فصل پنجم

الف) درستی یا نادرستی هر عبارت را تعیین کنید.

۱ــ برنامه‌های کنسولی کاربرد زیادی در سیستم عامل ویندوز دارند.

۲ــ در برنامه‌های ویندوزی، واسط کاربری به صورت گرافیکی است.

۳ــ برای نمایش یک متن روی فرم از کنترل Label استفاده می‌کنیم.

۴ــ عنوان یک فرم از طریق ویژگی Text مقداردهی می‌شود.

۵ــ می‌توان منوی کنترل پنجره و آیکن‌های بیشینه، کمینه و خروج را پنهان کرد.

ب) در سؤالاتِ چند گزینه‌ای زیر، پاسخ صحیح را انتخاب نمایید.

۶ــ کدام ویژگی یک برچسب مربوط به متن داخل آن است؟

الف) RightToLeft　　ب) Text

ج) Enabled　　د) TextAlign

۷ــ برای تغییر ویژگی‌های یک کنترل از پنجره.....استفاده می‌کنیم.

الف) Solution Explorer　　ب) Properties

ج) Toolbox　　د) Class View

۸ــ برای دیدن پروژه و فایل‌های آن از پنجره..... استفاده می‌شود

الف) Solution Explorer　　ب) Properties

ج) Toolbox　　د) Class View

۹ــ اگر بخواهیم در محیط طراحی، یک کنترل را به فرم اضافه کنیم از پنجره..... آن را به داخل فرم می‌آوریم.

الف) Solution Explorer　　ب) Properties

ج) Toolbox　　د) Class View

۱۰ــ ویژگی SizeMode برای تنظیم چه چیزی از کنترل PictureBox به کار برده می‌شود؟

الف) مکان تصویر　　ب) تصویر درون کنترل

ج) نوع تعیین اندازه تصویر　　د) اندازه کنترل

۱۱ــ برای ثابت نگه داشتن فاصله از طرفین برای یک کنترل از کدام ویژگی استفاده می‌نماییم؟

الف) Dock　　ب) Text

ج) Location　　د) Anchor

۱۲ــ اگر ویژگی FormBorderStyle برابر با Sizable شود فرم..... خواهد بود.

الف) با اندازه ثابت　　ب) یک کادر محاوره‌ای

ج) بدون دکمه‌های بیشینه و کمینه　　د) با اندازه متغیر

ج) جاهای خالی را با عبارات مناسب پرکنید.

۱۳ــ اطلاعی که از طرف سیستم عامل به برنامه داده می‌شود.....نامیده می‌شود.

۱۴ــ ویژگی RightToLeft برای زبان‌های..... پیش‌بینی شده است

۱۵ــ در کنترل PictureBox، ویژگی که تصویر را در بر می‌گیرد.....است.

تمرینات برنامه نویسی فصل پنجم

۱ــ فرم زیر را برای نمایش اطلاعات شخصی خود (نام ، نام خانوادگی، نام پدر، تاریخ تولد و عکس) طراحی کنید.

۲ــ فرم تمرین ۱ برای نمایش اطلاعات به زبان فارسی که یک زبان RTL است، مناسب نیست. با تغییر ویژگی‌های فرم و برچسب‌ها، فرم را برای زبان RTL مناسب کنید.

فعالیت

متن زیر از MSDN با موضوع برنامه‌های ویندوزی برداشت شده است. آن را با کمک هم‌کلاسی خود ترجمه کنید و به کلاس ارائه دهید.

Windows Forms provide your project with components, such as dialog boxes, menus, buttons, and many other controls, that make up a standard Windows application user interface (UI). Fundamentally, these controls are just classes from the .NET Framework class library. The Designer view in Visual C# Express Edition enables you to drag the controls onto your application's main form and adjust their size and position. As you do this, the IDE automatically adds the source code to create an instance of the appropriate class and initialize it.

واژگان و اصطلاحات انگلیسی فصل پنجم

ردیف	واژه انگلیسی	معنی به فارسی
۱	Action	
۲	Alphabetical view	
۳	Button	
۴	Categorized view	
۵	container	
۶	Context menu	
۷	Event	
۸	Form Designer	
۹	Graphical User Interface	
۱۰	Horizontally	
۱۱	Message	
۱۲	Notification	
۱۳	Object	
۱۴	Pixel	
۱۵	Pushpin	
۱۶	Right to Left	
۱۷	Right to left languages	
۱۸	Shortcut	
۱۹	Timer Control	
۲۰	Tool Bar	
۲۱	Tool Box	
۲۲	Upper–left corner	
۲۳	Vertically	
۲۴	Wait state	