آموزش C#

آشنایی با زبان C#

زبان C# یک زبان سطح بالا، شی گرا و همه منظوره است که به وسیله شرکت مایکروسافت هم زمان با پیدایش لایه نرم افزاری جدید آن به نام.NET ابداع و توسعه پیدا کرده است. از نرم افزارهای متنوع و گوناگونی از جمله نرم افزارهای اداری و برنامه های کاربردی تحت وب گرفته تا نرم افزارهایی برای تلفن همراه و بازی های کامپیوتری، با زبان C# و با استفاده از لایه .NET تولید می شود .

زبان C# شباهت زیادی به زبان های C++ و Java دارد و ویژگی هایی را از آنها تقلید کرده، یا بعضی امکانات آنها را بهبود داده است.

الگوی یک برنامه ساده به زبان C#

یک برنامه ی کاربردی نوشته شده به زبان C# شامل مجموعه ای از کلاس ها است که هر یک از آنها نیز شامل تعدادی متد هستند .اما در یک برنامه ساده تنها یک کلاس وجود دارد که در آن نیز فقط یک متد به نام( )Main تعریف می شود که نقطه آغاز اجرای برنامه است و الگوریتم خود را با رعایت قوانین زبان C# در آن می نویسیم.

الگو یا ساختارکلی یک برنامه ساده به زبان C# در زیر آمده است .الگوی زیر را به خاطر بسپارید.

```
class یک نام دلخواه
{
      static void Main()
        {
            دستورات مربوط به انجام یک کار
        }
}
```

کلاس (class) چیست؟

کلاس یک مفهوم اساسی در برنامه نویسی شی گرا است که در اینجا اگر بخواهیم به طور ساده در مورد معنی و مفهوم کلاس صحبت کنیم، باید بگوییم که کلاس به عنوان یک قالب یا الگویی می باشد که در آن داده هایی تعریف می شود .این داده ها مربوط به یک موضوع است و عملیاتی که می توان بر روی آنها انجام داد .در زبان C# گنجینه ای از کلاس های مختلف و کاربردی، از قبل تعریف شده و آماده وجود دارد که برنامه نویس کافی است آنها را بشناسد و در برنامه استفاده نماید.

Console یک کلاس آماده در زبان C# است که عملیات مختلف ورودی و یا خروجی(بر روی صفحه نمایش و یا صفحه کلید)در آن تعریف شده است.

نحوه تعریف کلاس: در زبان C# این امکان برای برنامه نویس فراهم است که کلاس جدیدی را تعریف کند. همان طور که در زیر مشاهده می کنید از کلمه کلیدی class برای تعریف و مشخص کردن یک کلاس جدید استفاده می شود. در جلوی کلمه class یک نام دلخواه ذکر می گرددکه نام کلاس است.

```
class نام کلاس
{
        تعریف داده ها و عملیات بر روی آن ها
}
```

متد چیست؟

همان طور که گفته شد در داخل کلاس، عملیات بر روی داده ها و یا الگوریتم انجام یک کار تعریف می شود .متد مجموعه ای از دستورات است که برای انجام یک کار لازم است .هر متد مطابق با عملکردش نام گذاری می شود و همچنین دارای یک جفت پرانتز باز و بسته است که در آن ممکن است ورودی هایی ذکر شود که برای انجام کار لازم است.

در برنامه های زبان C# ممکن است متدهای زیادی تعریف و یا مورد استفاده قرارگیرند، اما حتماً باید متدی به نام Main() تعریف شده باشد که نقطه آغاز اجرای برنامه است و اجرای یک برنامه از اولین دستور داخل آن شروع می شود.

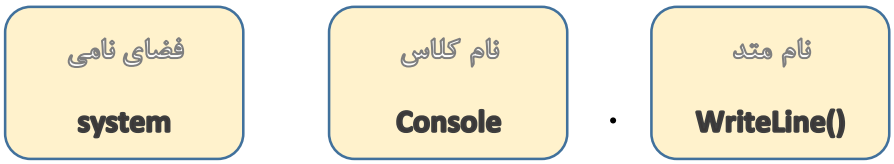
کلمات static و void در قالب کلی متد Main() ویژگی های متد را مشخص می کنند.

```
static void Main()
{
      دستور شماره ۱;
      دستور شماره ۲;
      دستور شماره ۳;
            .
            .
            .

}
```

استفاده از متدهای آماده: تعداد زیادی متد در کلاس های آماده زبان#c وجود دارد که هر یک از آنها، برای انجام کاری در نظر گرفته شده است. مثلاً متدWriteLine() **از کلاس Console** برای نشان دادن پیام روی صفحه نمایش در نظر گرفته شده است.

برای استفاده از یک متد نیز باید نام فضا یا حوزه، نام کلاس و سپس نام متد را مشخص کنید و برای جدا کردن آنها از یکدیگر، علامت نقطه بین آنها قرار دهید.



اولین برنامه به زبان C#

با یک برنامه ساده آشنا می شویم:

```
class WelcomeToCSharp
{
    static void Main( )
    {
        System.Console.WriteLine("Welcome To C# !");
     }
}
```

این برنامه کوچک فقط یک پیام خوش آمدگویی بر روی صفحه نمایش نشان می دهد.
با نگاهی جزیی تر به برنامه مشاهده می کنید که این برنامه از تعدادی کلمه و علامت تشکیل شده است.
بعضی از کلمات مانند void ،class و static کلمات شناخته شده برای زبان C# هستند و دارای معنی و مفهوم ثابتی هستند به این نوع کلمات ، کلمات کلیدی یا رزرو شده گفته می شود .کلمات رزرو شده به رنگ آبی نوشته شده اند و به تدریج با آنها آشنا می شوید.
بعضی از کلمات دیگر مانند WelcomeToCSharp نامی است که به وسیله برنامه نویس و طبق سلیقه وی انتخاب می شود. به این نام ها شناسه می گویند .برنامه نویس در انتخاب شناسه ها باید ضوابطی را رعایت کند.
با نگاهی دیگر و کلی تر به برنامه، مشاهده می کنیم که یک برنامه ساده از یک قسمت کلی به نام کلاس تشکیل شده است که با کلمه کلیدی class مشخص می شود و شروع و پایان آن با علامت آکولاد باز و بسته تعیین می گردد .در جلوی کلمه کلیدی class، یک نام(شناسه)دلخواه مثلاً WelcomeToCSharp نوشته می شود که بیان کننده کار برنامه است.

قسمت کلاس برنامه را در شکل زیر مشاهده کنید.

```
class WelcomeToCSharp
{

}
```

اگر درون کلاسWelcomeToCSharp را نگاه کنیم یک قسمت دیگر را خواهیم دید که چنین شروع شده است:

```
static void Main( )
```

شروع و پایان این قسمت نیز با علامت های آکولاد باز و بسته، مشخص شده است .به این قسمت متد Main می گوییم که بدنه اجرایی برنامه است هر دستوری که در این قسمت نوشته شود به وسیلهٔ کامپیوتر به ترتیب اجرا می شود . دستورهای برنامه خود را در این قسمت می نویسیم.

آخرین قسمتی که در این برنامه، در داخل متدMain قابل تشخیص است، یک دستور اجرایی است و به کامپیوتر اعلام می کند که چه باید انجام دهد که در این برنامه ، نمایش یک پیام است:

```
System.Console.WriteLine(" Welcome To C#!");
```

با اجرای دستوربالا، پیام خوش آمدگویی Welcome to C#! بر روی صفحه نمایش، نشان داده می شود.

آشنایی با Visual Studio

تایپ برنامه در یک ویرایشگر، ورود به پنجره فرمان، ترجمه کردن، عیب یابی و اشکال زدایی برنامه، همگی عملیاتی هستند که وقت گیر و پر زحمت اند، چون از یک محیط باید وارد محیط دیگری شوید. برای این که راحت تر بتوانیم برنامه نویسی کنیم لازم است از محیطی استفاده کنیم که همه ابزارها و لوازم مورد نیاز برنامه نویسی در آن گردآوری و متمرکز شده باشد .به چنین محیط برنامه نویسی که در آن می توان تمام مراحل برنامه نویسی، ترجمه، اشکال یابی و سرانجام اجرا را انجام داد، IDE گفته می شود که به معنای محیط تولید برنامه ی متمرکز می باشد .یعنی همه ابزارها و امکانات لازم برای تولید برنامه در یک جا گردآوری شده است.

ویژوال استودیو یک محیط برنامه نویسی بسیار قوی برای تولید برنامه های کاربردی تحت ویندوز و بر پایه Net Framework. می باشد. ویژوال استودیو از چند زبان برنامه نویسی نظیر VBو C++ ، C# پشتیبانی می کند .در این محیط علاوه بر تایپ برنامه، می توان برنامه را ترجمه، عیب یابی و سرانجام اجرا کرد.

در این محیط هنگامی که مشغول تایپ برنامه هستید باید تفاوت قابل ملاحظه ای را با روشهای قبل که برنامه را در محیط Notepad ویندوز می نوشتید احساس کنید .اولاً کلمات با توجه به نوع آنها رنگی نوشته شده اند، مثلاً کلمات کلیدی با رنگ آبی نشان داده شده اند و ضمناً در هنگام تایپ برنامه به محض نوشتن یک حرف کلمهConsole لیستی از کلمات مشابه نمایش داده می شود. با تایپ چند حرف دیگر کلمهConsole در لیست نشان داده می شود و در کنار آن یک توضیح مختصر دیده می شود.

مقداردهی متغیرها

پس از تعریف یا ایجاد متغیر، می توانید در آن، مقداری را با توجه به نوع متغیر ذخیره کنید.

توجه داشته باشید که در یک متغیر همواره فقط یک مقدار نگهداری می شود و با ذخیره کردن داده جدید در یک متغیر، مقدار قبلی آن از بین می رود .مقداردهی متغیرها به چند روش صورت می گیرد .با دستور زیر مستقیما مقداری در متغیر قرار می گیرد به این دستور، دستور انتساب می گویند.

```
نام متغیر = مقدار ;
```

در هنگام تعریف یا ایجاد متغیر نیز می توانید آن را مستقیما مقداردهی کنید که به آن **مقداردهی اولیه** می گویند .الگوی آن چنین است:

```
نام متغیر  نوع داده = مقدار ;
```

نشان دادن محتوای متغیرها بر روی صفحه نمایش

معمولاً در برنامه ها لازم است محتوای متغیرها که شامل داده ها و یا نتایج پردازش با اطلاعات بر روی صفحه نشان داده شود تا کاربر از آنها آگاه شود .بدین منظور از متد WriteLine() یا Write() استفاده می کنیم.

دریافت رشته

تاکنون داده های مشخص و ثابتی رادر داخل برنامه استفاده کردیم .این داده ها به وسیله ی برنامه نویس درون برنامه تعیین شده بود .حال می خواهیم برنامه های خود را کاربردی کنیم و داده ای را از کاربر دریافت کنیم .برای این منظور از متد ReadLine() استفاده می کنیم که به کاربر اجازه می دهد تا داده مورد نظر خود را از طریق صفحه کلید وارد کند.

متد ReadLine () مانند متد هایی که تاکنون خوانده ایم در کلاس Console تعریف شده است و در فضای نامی System قرار دارد بنابراین به صورت زیر استفاده می شود:

```
System.Console.ReadLine();
```

مثال۱: نام کاربر از ورودی دریافت شود و خطاب به او پیام خوشامدگویی اعلام شود.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Hello
{
    class Program
    {
        static void Main(string[] args)
        {
            string name;

            Console.Write("Enter your name:" );
            name = Console.ReadLine();
            Console.WriteLine("Hello "+name);

            Console.ReadKey();
        }
    }
}
```

مثال ۲:می خواهیم به مثال قبل دستوراتی اضافه کنیم که علاوه بر دریافت نام کاربر، نام خانوادگی وی نیز سؤال شود و سپس نام ونام خانوادگی را در یک خط نمایش دهد.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace NameFamily
{
    class Program
    {
        static void Main(string[] args)
        {
            string name, family;

             Console.Write("Enter your name:");
             name = Console.ReadLine();

             Console.Write("Enter your family:" );
             family = Console.ReadLine() ;

             Console.WriteLine("Hello "+ name +" "+ family);

            Console.ReadKey();
        }
    }
}
```

توضیحات: کافی است یک متغیر رشته ای به نام family تعریف کرده و از متد ReadLine() برای دریافت نام خانوادگی استفاده کنیم .برای نمایش نام و نام خانوادگی در یک خط نیز، از علامت +برای الحاق رشته ها استفاده می کنیم .