زنگ سی‌شارپ – قسمت دوم

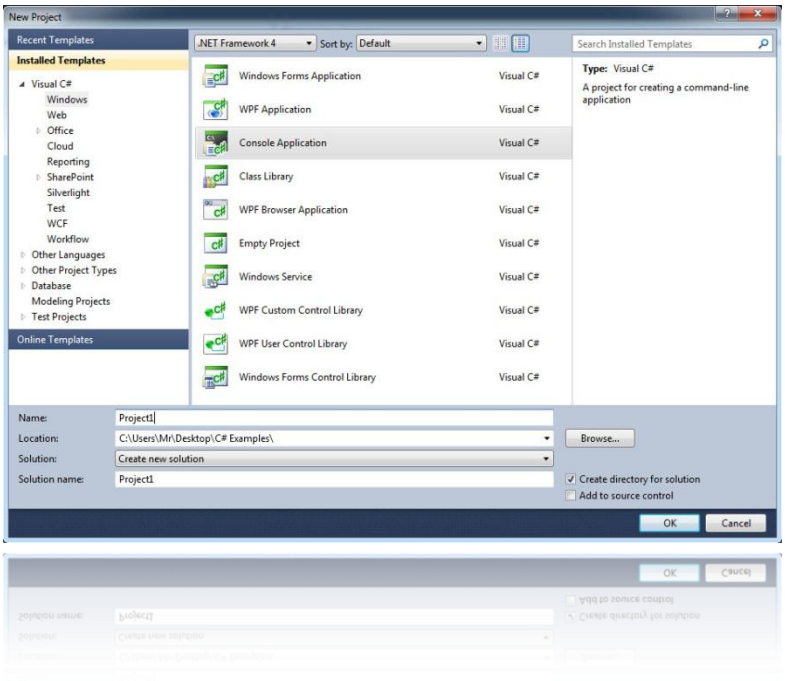
نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در قسمت قبل اندکی با تاریخچه سی‌شارپ و توضیحات مختصری در مورد آن و در مورد این سری از مقالات آموزشی آشنا شدیم. در این قسمت قصد داریم بیشتر با کدنویسی و محیط برنامه Visual Studio IDE آشنا شویم. ویژوال استودیو، IDE شرکت مایکروسافت است، IDE مخفف Integrated Development Environment (محیط یکپارچه توسعه نرم‌افزار) است. Visual Studio به شما اجازه ویرایش، کامپایل، اجرا و خطایابی (Debug) برنامه‌های سی‌شارپ را می‌دهد پس برای دنبال‌کردن این سری مقالات و یادگیری C# نیاز دارید که یک نسخه از ویژوال استودیو را روی سیستم خود نصب کنید و پیشنهاد من نسخه‌ی ۲۰۱۰ به بعد است چراکه ممکن است در برخی از موارد از ویژگی‌های C# 4.0 استفاده کنیم.

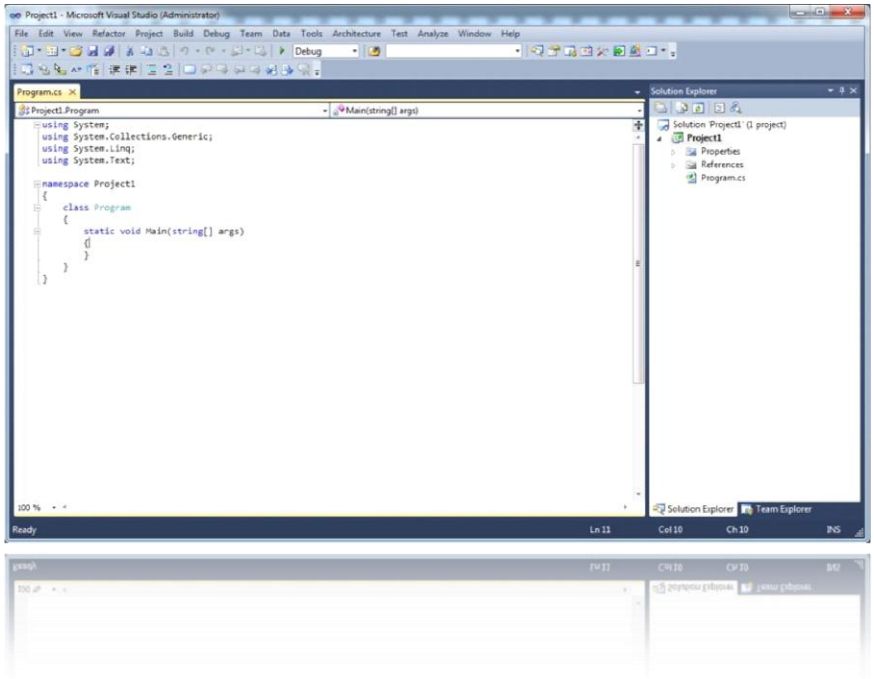
همچنین می‌توانید نسخه‌ی رایگان را از وب‌سایت مایکروسافت دریافت کنید. دیگر وقت آن رسیده است که اولین برنامه سی‌شارپ را بنویسیم. برای این منظور باید یک پروژه برای سی‌شارپ در ویژوال استودیو بسازیم. پس مراحل زیر را برای ساخت یک پروژه خالی سی‌شارپ که در اینجا از Visual Studio 2010 Professional استفاده شده است دنبال کنید.

۱. ویژوال استودیو را اجرا کنید و از منوی بالا به File => New => Project بروید، سپس Windows را از لیست قالب‌های نصب‌شده انتخاب کنید، بعد از آن Console Application را انتخاب کنید:



سپس نام پروژه خودتان را در قسمت Name وارد کنید و روی OK کلیک کنید تا پروژه شما ساخته شود.

۲. **زمانی که پروژه شما ساخته شد، ویژوال استودیو بدین شکل به‌نظر می‌رسد:**



اگر به هر دلیلی پنجره **Solution Explorer** را در سمت راست مشاهده نکردید می‌توانید از منوی **View** با انتخاب **Solution Explorer** آن را فعال کنید.

تا اینجا شما توانستید پروژه خودتان را که هم‌اکنون آماده برای کدنویسی است تهیه کنید. همان‌طور که مشاهده می‌کنید مقداری کد به صورت پیش‌فرض در پنجره **Program.cs** قرار دارد که در ادامه به‌شرح مختصر آنها می‌پردازیم.

```
using System;
```

این خط کد مشخص می‌کند که برنامه شما از فضای‌نام **System** استفاده می‌کند که یکی از فضاهای‌نام **(namespaces)** پایه‌ای در **Net Framework.** است. احتمالاً هنوز نمی‌دانید **Net Framework.** چیست، پس حتماً مقاله‌ی چهارچوب دات‌نت را در ویکی‌پدیا مطالعه بفرمایید.

Namespace روشی برای مدیریت کد و گروه‌بندی کلاس‌های مرتبط به‌هم است.

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

توضیح این چند خط کد در مقالات بعدی و در جای خود دنبال خواهد شد.

```
class Program
```

این خط کد از کلمه‌کلید `class` برای اعلان یک کلاس جدید که از قبل در **Net Framework.** تعریف شده است استفاده می‌کند. `Program` نام این کلاس است. تعریف کلاس با آکولاد باز **{** شروع و با آکولاد بسته **}** تمام می‌شود که عناصر بین آکولاد، اعضای کلاس هستند. از آن‌جا که سی‌شارپ یک زبان برنامه‌نویسی تماماً شی‌گرا و همه‌ی سروکارش با کلاس‌ها است باید در نحوه‌ی تعریف کلاس و استفاده از آن‌ها تسلط کافی داشته باشیم که این کار را در مقالات آینده انجام خواهیم داد.

```
static void Main(string[] args)
```

متد **()Main** در اینجا قسمتی است که عملیات اصلی برنامه در آن انجام می‌شود. بدون متد **()Main** برنامه‌های سی‌شارپ قابل اجرا نخواهند بود، نوع آن در اینجا **void** تعریف شده است یعنی این متد خروجی ندارد، متد **()Main** هم با آکولاد باز و بسته محدوده خودش را مشخص می‌کند.

در مورد **Method** و **Class** بعداً مفصل توضیح خواهیم داد پس اصلاً نگران نباشید زیرا تا این‌جا فقط یک توضیح مختصر در مورد کدهای پیش‌فرضی بود که در ابتدا می‌دیدید.

اکنون اندکی با برنامه **Visual Studio IDE** و نحوه‌ی پروژه ساختن آشنا شدید. همین‌طور کدهای پیش‌فرضی که در ابتدا می‌دیدید را تاحدی شناختید. هم‌اکنون قصد داریم با چند کد ساده شروع به برنامه‌نویسی کنیم. ممکن است در ابتدا چیزی ببینید که اصلاً از آن سر در نمی‌آورید اما اصلاً نترسید چون همه‌چیز را تا حدی که مفهوم مطلب بیان شود و اطمینان حاصل شود که شما خوانندگان عزیز آن موضوع را درک کرده‌اید، توضیح خواهیم داد. فقط از خواندن باز نایستید!

همان‌طور که گفته شد عملیات اصلی برنامه در متد **()Main** نوشته می‌شود و فعلاً با بقیه‌ی قسمت‌ها کاری نداریم چراکه در جای خود به آن‌ها می‌رسیم و به شرح آنها می‌پردازیم.

ویژوال استودیو را اجرا کنید و یک پروژه جدید بسازید.

به کد زیر نگاه کنید:

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

namespace Project1
{
    class Program
    {
        // A C# program begins with a call to Main().
        Static void Main(string[] args)
        {
            /*
                This is a simple C# program.
                Green lines are comments.
            */

            Console.WriteLine(“A Simple C# Program.”);
        }
    }
}
```

همان‌طور که مشاهده می‌کنید مقداری از کدها به صورت پیش‌فرض در صفحه ما بوده‌اند و فقط این چند خط جدید است:

```
// A C# program begins with a call to Main().
```

```
/*
       This is a simple C# program.
       Green lines are comments.
*/

Console.WriteLine("A Simple C# Program.");
```

خطوط سبزی که مشاهده می‌کنید **comment** (توضیح) هستند. در سی‌شارپ ما همانند سایر زبان‌های برنامه‌نویسی اجازه داریم که از کامنت استفاده کنیم. همان‌طور که می‌بینید به دو طریق می‌توانیم **comment** بگذاریم. یکی نوشتن جلوی دو اسلش **//** و دیگری نوشتن مابین ستاره و اسلش **/* */** محتوای کامنت‌ها توسط کامپایلر نادیده گرفته می‌شوند، کامنت‌ها فقط برای این منظور توسط برنامه‌نویس نوشته می‌شوند که هرکس در حال خواندن کد و برنامه است آن قسمت از کد را بفهمد زیرا کامنت‌ها اکثراً توضیح و شرحی در مورد کد هستند و نوشتن آنها اختیاری است.

```
Console.WriteLine("A Simple C# Program.");
```

خط کدی که در بالا مشاهده می‌کنید، رشته‌ی **"A simple C# Program."** را در یک خط جدید در خروجی نمایش می‌دهد. فعلاً در همین حد بدانید که `Console.WriteLine()` در این مثال یک **String** (رشته‌ای از کاراکترها) را در خروجی نمایش می‌دهد. بعداً متوجه خواهید شد که چطور این اتفاق می‌افتد. پس دانستید که **String** رشته‌ای از کاراکترها است که حتماً باید بین این علامت **" "** (**Double quotation**) قرار بگیرد و `Console.WriteLine()` برای چاپ کردن اطلاعات و پیغام در خروجی است و آن‌چه را که می‌خواهید در خروجی نمایش دهید باید بین () قرار دهید، درست طبق مثال. در انتهای آن نقطه و ویرگول بدین شکل ; قرار می‌گیرد که اصطلاحاً به آن **Semicolon** (بخوانید سِمی‌کالن) گفته می شود. بدین صورت:

```
Console.WriteLine();
```

توجه داشته باشید که سی‌شارپ به حروف بزرگ و کوچک حساس و اصطلاحاً **case-sensitive** است و این بدین معنی است که اگر فراموش کنید کجا باید از حروف بزرگ و کوچک استفاده کنید برنامه شما دچار خطا می‌شود. برای مثال اگر نام یک متغیر را **phoneNum** انتخاب کرده باشید و در جای دیگر برای استفاده از آن بنویسید **PhoneNum** یا **Phonenum** آنگاه برنامه شما دچار مشکل می‌شود. دستور `Console.WriteLine()` هم به همین روال است همین‌طور بقیه دستورها، هرچند که کامپایلر سی‌شارپ

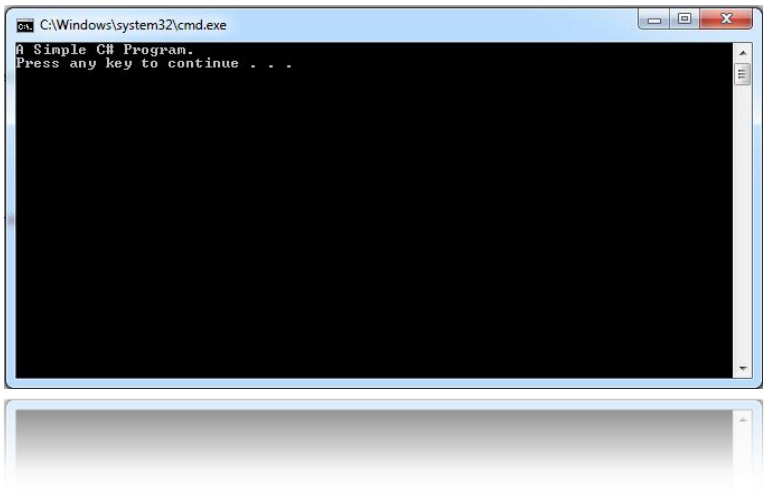
فوق‌العاده قدرتمند است و در صورت بروز کوچکترین خطا شما را باخبر می‌کند، به‌جز خطاهای منطقی. در مورد خطاهای منطقی بعداً بیشتر صحبت خواهیم کرد.

تا اینجا با یک سری از اصطلاحات و مفاهیم آشنا شدید، بعد از اینکه یک سری از مطالب را با هم مورد بررسی قرار دادیم سراغ یک مثال درست و حسابی می‌رویم.

```
Static void Main(string[] args)
{
	Console.WriteLine("A Simple C# Program.");
}
```

پس طبق کد بالا دستور چاپ یک **string** را بنویسید و سپس **Ctrl + F5** را باهم بگیرید تا خروجی برای شما نمایش داده شود:

(به جای فشردن **Ctrl + F5**، از منوی **Debug** با انتخاب **Start Without Debugging** هم می‌توانید این‌کار را انجام دهید)



همان‌طور که می‌بینید **A Simple C# Program** در پنجره خروجی نمایش داده شد.

هدف از این سری مقالات آموزشی یادگیری سی‌شارپ به طور گام‌به‌گام و تقریباً کامل برای هر مبحث است. هرچند مجبوریم بعضی از مسائل را فعلاً به‌صورت کامل باز نکنیم تا در فهم موضوع دچار مشکل نشوید اما در آینده به آن‌ها پی خواهید برد. اگر همیشه این سری مقالات را دنبال کنید کم‌کم هر قسمت از زبان سی‌شارپ

را به طور مفهومی آموزش می‌بینید. اگر در مورد مطلب هر قسمت سوالی برایتان پیش آمد و مشکلی داشتید می‌توانید در قسمت نظرات سوالات خود را مطرح کنید یا از شبکه‌های اجتماعی که در پروفایل من در همین سایت موجود است با من تماس بگیرید.

به‌یاد داشته باشید آنکه می‌خواهد روزی پریدن آموزد، نخست می‌باید ایستادن، راه‌رفتن، دویدن و بالارفتن آموزد. پرواز را با پرواز آغاز نمی‌کنند.

کلیه حقوق مادی و معنوی برای وب‌سایت وب‌تارگت محفوظ است.
استفاده از این مطلب در سایر وب‌سایت‌ها و نشریات چاپی تنها با ذکر و درج لینک منبع مجاز است.