

مایکل اولسون

آموزش برنامه نویسی

به زبان C#

مرجع سریع دستورات C#

مترجم: مروت گیوی

توجه :

کتاب حاضر ترجمه مروت گوی از انتشارات ترجمک می‌باشد. فایل کتاب حاوی اطلاعات DRM (مدیریت حقوق دیجیتال) است. وقتی برای اولین بار فایل را باز می‌کنید، کد شناسایی کتاب به همراه آدرس IP سیستم شما ذخیره شده و زمانیکه آنلاین شوید، به سرور انتشارات ترجمک انتقال می‌یابد.

خواهشمند است به حقوق مترجمان و گروه فنی ترجمک احترام گذاشته و از توزیع بدون مجوز فایل کتاب اجتناب نمایید. شما با خرید و دانلود این کتاب موافقت نموده اید که اطلاعات فایل DRM به سرور انتشارات ترجمک انتقال یابد و در صورت محرز شدن نقض حقوق صاحب اثر، کلیه خسارات حاصله در طی فرآیند حقوقی و مطابق قانون حمایت حقوق مؤلفان و مصنفان و هنرمندان و ناشران جمهوری اسلامی (مصوب دوازده اسفند ۱۳۶۵ یا بعد از آن) از شما دریافت شود.

از اینکه با عرضه مقرون به صرفه کتاب های الکترونیک و شکوفایی انتشارات ترجمک همیاری می‌کنید، سپاسگزاریم.

مایکل اولسون

آموزش برنامه نویسی

به زبان C#

مرجع سریع دستورات C# 8.0

راهنمای جیبی زبان، API و کتابخانه ها

ادیشن سوم

ترجمه

مرّوت گیوی



شناسنامه کتاب

نام کتاب: آموزش برنامه نویسی به زبان C# 8.0

ترجمه: مروت گیوی

ناشر: انتشارات ترجمک

صفحه آرای: انتشارات ترجمک

طراحی جلد: محمدحسین گیوی

نوبت چاپ: چاپ اول، ۱۴۰۰

قیمت: ۲۵۰۰۰ تومان

چاپ: گروه نشر الکترونیک ترجمک

شابک: ۹۷۸-۶۲۲-۹۸۳۹۸-۵-۰

تلفن تماس: ۰۹۱۸۱۵۰۶۱۰۰

تارنمای اینترنتی: <https://tarjomac.com>

ISBN:978-622-98398-5-0



9

مقدمه

زبان برنامه نویسی C# یک زبان شی محور است که توسط مایکروسافت برای فریم ورک دات نت (NET Framework) ساخته شد. C# (که بصورت سی شارپ تلفظ می شود) براساس بعضی از بهترین خصیصه های زبان های برنامه نویسی بزرگ و اصلی ساخته شده است. سی شارپ ترکیب قدرت C++ و سادگی ویژوال بیسیک است و همچنین مقداری از جاوا به قرض گرفته است. اینکار موجب تولید زبانی شده است که به راحتی آموخته و استفاده می شود، قدرتمند و در مقابل خطا مصون است و تولید سریع اپلیکیشن ها را ممکن کرده است. همه اینها بدون فدا کردن مقدار زیادی از قدرت یا سرعت برنامه نویسی بدست آمده است، وقتی با C++ مقایسه می شود.

در سال های پس از انتشار آن در سال ۲۰۰۲، C# به سرعت رشد کرده و به یکی از زبان های برنامه نویسی معروف تبدیل شد. این یک زبان برنامه نویسی همه منظوره است، بنابراین برای خلق دامنه وسیعی از برنامه ها مناسب است. همه چیز از یک برنامه کاربردی کوچک تا بازی های کامپیوتری، اپلیکیشن های دسکتاپ یا حتی سیستم های عامل را می توان با C# ساخت. این زبان را می توان با ASP.NET بکار گرفته و اپلیکیشن های تحت وب نوشت.

زمان برنامه نویسی در دات نت، برنامه نویسان دامنه وسیعی از گزینه ها را دارند و می توانند زبان مورد علاقه خود را انتخاب کنند. بعضی از زبان های خیلی معروف دات نت شامل VB.NET، C++/CLI، F# و C# است. از بین اینها، اغلب C# زبان انتخابی است. همانند سایر زبان های دات

نت، C# ابتدای امر به یک زبان میانه کمپایل می شد. این زبان را زبان عمومی میانه (CIL)^۱ می نامیدند و روی فریم ورک دات نت اجرا می شود. یک برنامه دات نت بنابراین فقط روی سیستمی اجرا می شود که از قبل این فریم ورک در آن نصب شده باشد.

فریم ورک دات نت یک فریم ورک نرم افزاری است که شامل موتور اجرای عمومی و یک کتابخانه کلاس غنی است. این فریم ورک بر روی مایکروسافت ویندوز اجرا می شود و بنابراین فقط برای نوشتن اپلیکیشن های تحت ویندوز استفاده می شود. هر چند درگاه های بین پلتفرم آن ایجاد شده است و دو تا پورت اصلی آن یعنی Mono^۲ و NET Core^۳ است. این دو هر دو پروژه های متن باز هستند که اجازه می دهند اپلیکیشن های دات نت در سایر پلتفرم ها مثل لینوکس، مکینتاش و سیستم های مستتر اجرا شوند.

^۱ Common Intermediate Language

^۲ www.mono-project.com

^۳ <https://docs.microsoft.com/dotnet/core>

محتوای کتاب در یک نگاه

صفحه	فصل
۱.....	فصل ۱- دنیا سلام
۴.....	فصل ۲- کمپایل و اجرا
۷.....	فصل ۳- متغیرها
۱۲.....	فصل ۴- اپراتورها
۱۷.....	فصل ۵- رشته ها
۲۱.....	فصل ۶- آرایه ها
۲۴.....	فصل ۷- دستورات شرطی
۲۸.....	فصل ۸- حلقه ها
۳۱.....	فصل ۹- متدها
۴۰.....	فصل ۱۰- کلاس
۵۳.....	فصل ۱۱- وراثت
۵۸.....	فصل ۱۲- تعریف مجدد اعضا
۶۲.....	فصل ۱۳- سطوح دسترسی
۶۸.....	فصل ۱۴- استاتیک
۷۳.....	فصل ۱۵- ویژگی ها
۷۷.....	فصل ۱۶- ایندکس کننده ها

۸۱	فصل ۱۷- واسط ها
۸۵	فصل ۱۸- انتزاع
۸۸	فصل ۱۹- فضای نام
۹۲	فصل ۲۰- برشمارش
۹۵	فصل ۲۱- رفع و رجوع استثناء
۱۰۲	فصل ۲۲- سربارگذاری اپراتور
۱۰۶	فصل ۲۳- تبدیل های سفارشی (دلخواه)
۱۰۸	فصل ۲۴- کلمه کلیدی struct
۱۱۱	فصل ۲۵- پیش پردازشگرها
۱۱۵	فصل ۲۶- تفویض به نماینده
۱۲۱	فصل ۲۷- وقایع
۱۲۵	فصل ۲۸- ژنریک ها
۱۳۴	فصل ۲۹- ثابت ها
۱۳۸	فصل ۳۰- متدهای آسنکرون
۱۴۳	واژه نامه

فهرست مندرجات کتاب

صفحه	عنوان
۱	فصل ۱- دنیا سلام.....
۱	انتخاب محیط توسعه تلفیقی (IDE).....
۲	ایجاد یک پروژه.....
۳	IntelliSense.....
۴	فصل ۲- کمپایل و اجرا.....
۴	کمپایل کردن در ویژوال استودیو.....
۴	کمپایل کردن در کنسول.....
۵	نسخه زبان.....
۵	کامنت و توضیحات.....
۷	فصل ۳- متغیرها.....
۷	انواع داده.....
۷	اعلان متغیر.....
۸	مقداردهی به متغیر.....
۸	انواع متغیر عدد صحیح.....
۹	انواع اعشاری.....
۱۰	انواع رشته ای.....
۱۰	انواع بولی.....
۱۰	حوزه متغیر.....

۱۲	فصل ۴- اپراتورها
۱۲	اپراتورهای جبری
۱۳	اپراتورهای تخصیص
۱۳	اپراتورهای تخصیص ترکیبی
۱۳	اپراتورهای یک واحد اضافه و کم
۱۴	اپراتورهای مقایسه
۱۴	اپراتورهای منطقی
۱۴	اپراتورهای بیتی
۱۵	تقدم اپراتورها
۱۷	فصل ۵- رشته ها
۱۷	متصل کردن رشته ها
۱۸	کاراکتر فرار
۱۹	مقایسه رشته ها
۱۹	اعضای رشته
۱۹	کلاس <code>StringBuilder</code>
۲۱	فصل ۶- آرایه ها
۲۱	اعلان آرایه
۲۱	تخصیص آرایه
۲۲	مقداردهی آرایه
۲۲	دسترسی به آرایه
۲۲	آرایه مربعی
۲۳	آرایه دندان‌دار
۲۴	فصل ۷- دستورات شرطی
۲۴	دستور <code>If</code>
۲۵	دستور <code>Switch</code>
۲۶	دستور <code>Goto</code>
۲۶	عبارت <code>Switch</code>
۲۷	عملگر سه تایی
۲۸	فصل ۸- حلقه ها

۲۸	While حلقه
۲۸	Do-While حلقه
۲۹	For حلقه
۲۹	Foreach حلقه
۳۰	Continue و Break
۳۱	فصل ۹- متدها
۳۱	تعریف کردن متد
۳۲	فراخوانی متد
۳۲	پارامترهای متد
۳۳	کلمه کلیدی Params
۳۳	بیش بار متد
۳۳	پارامترهای اختیاری
۳۴	آرگومان های نامگذاری شده
۳۵	Return دستور
۳۵	انواع ارزشی و رفرنس
۳۶	ارجاع با مقدار
۳۶	ارجاع با مرجع
۳۷	کلمه کلیدی Ref
۳۸	کلمه کلیدی Out
۳۸	متدهای محلی
۴۰	فصل ۱۰- کلاس
۴۰	خلق شی
۴۱	دسترسی به اعضای شی
۴۱	سازنده
۴۲	کلمه کلیدی this
۴۳	بیش بارگذاری سازنده
۴۳	زنجیره کردن سازنده
۴۴	مقادیر اولیه فیلد
۴۴	سازنده پیش فرض
۴۴	مقداردهی اولیه شی

۴۵	کلاس نسبی
۴۶	آشغال جمع کن
۴۶	تخریب کننده سازه
۴۷	کلمه کلیدی Null
۴۸	انواع مقدار قابل نول شدن
۴۸	انواع مرجع قابل نول شدن
۴۹	اپراتور نول-ادغام شونده
۵۰	عملگر شرطی نول
۵۱	مقادیر پیش فرض
۵۱	استنباط نوع
۵۲	انواع بی نام
۵۳	فصل ۱۱- وراثت
۵۴	بالادست و پایین دست (Downcast and Upcast)
۵۵	باکس کردن
۵۵	باکس گشایی
۵۵	کلمات کلیدی Is و As
۵۶	تطبیق الگو
۵۸	فصل ۱۲- تعریف مجدد اعضا
۵۸	مخفی کردن اعضا
۵۹	رونویسی اعضا
۵۹	مخفی کردن و رونویسی
۶۰	کلمه کلیدی Sealed
۶۰	کلمه کلیدی Base
۶۲	فصل ۱۳- سطوح دسترسی
۶۲	دسترسی خصوصی
۶۲	دسترسی محافظت شده
۶۴	دسترسی داخلی
۶۴	دسترسی داخلی محافظت شده

۶۵ دسترسی خصوصی محافظت شده
۶۵ دسترسی عمومی
۶۶ سطوح دسترسی رده بالا
۶۶ کلاس های درونی
۶۷ دستورالعمل سطح دسترسی
۶۸ فصل ۱۴- استاتیک
۶۹ دسترسی به اعضای استاتیک
۶۹ متدهای استاتیک
۶۹ فیلدهای استاتیک
۷۰ کلاس های استاتیک
۷۰ سازنده استاتیک
۷۰ توابع محلی استاتیک
۷۱ متدهای افزونه
۷۳ فصل ۱۵- ویژگی ها
۷۴ فایده های ویژگی
۷۵ ویژگی های فقط خواندنی و فقط نوشتنی
۷۶ سطوح دسترسی به ویژگی
۷۶ ویژگی های خود-اجرا
۷۷ فصل ۱۶- ایندکس کننده ها
۷۸ سربار گذاری ایندکس کننده
۷۹ دامنه ها و ایندکس کننده ها
۸۱ فصل ۱۷- واسط ها
۸۱ امضای واسط
۸۲ مثال واسط
۸۲ واسط کارکردی
۸۳ واسط کلاس
۸۴ پیاده سازی پیش فرض
۸۵ فصل ۱۸- انتزاع

۸۵	اعضای انتزاعی
۸۶	مثال Abstract
۸۷	کلاس های انتزاعی و واسط ها
۸۸	فصل ۱۹ - فضای نام
۸۸	فضای نام تو در تو
۸۹	دسترسی به فضای نام
۹۰	استفاده از امریه
۹۲	فصل ۲۰ - برشمارش
۹۲	مثال برشمارش
۹۳	مقادیر ثابت برشمارش
۹۳	نوع ثابت برشمارش
۹۳	سطح دسترسی و حوزه برشمارش
۹۴	متدهای برشمارش
۹۵	فصل ۲۱ - رفع و رجوع استثناء
۹۶	دستور Try-Catch
۹۶	قطعه Catch
۹۶	فیلترهای استثناء
۹۸	قطعه Finally
۹۹	دستور Using
۹۹	صدور استثناء
۱۰۲	فصل ۲۲ - سربارگذاری اپراتور
۱۰۲	مثال سربارگذاری اپراتور
۱۰۳	سربارگذاری عملگر باینری
۱۰۳	سربارگذاری اپراتور یکانی
۱۰۴	انواع برگشتی و پارامترها
۱۰۴	عملگرهای قابل سربارگذاری
۱۰۵	سربارگذاری عملگرهای true و false
۱۰۶	فصل ۲۳ - تبدیل های سفارشی (دلخواه)
۱۰۶	روش های تبدیل ضمنی

۱۰۷ روش های تبدیل صریح
۱۰۸ فصل ۲۴ - کلمه کلیدی struct
۱۰۸ متغیر struct
۱۰۹ سازنده های struct
۱۰۹ مقدار دهنده فیلد struct
۱۱۰ وراثت struct
۱۱۰ دستورالعمل struct
۱۱۱ فصل ۲۵ - پیش پردازشگرها
۱۱۲ سینتاکس پیش پردازشگرها
۱۱۲ نمادهای کمپایل شرطی
۱۱۲ کمپایل کردن شرطی
۱۱۳ رهنمودهای تشخیصی
۱۱۳ رهنمود Line
۱۱۳ رهنمود Region
۱۱۵ فصل ۲۶ - تفویض به نماینده
۱۱۶ متدهای بی نام
۱۱۶ عبارت لامبدا
۱۱۷ اعضای تته عبارت
۱۱۸ تفویض چند تبدیلی
۱۱۹ امضای تفویض
۱۱۹ تفویض ها به عنوان پارامتر
۱۲۱ فصل ۲۷ - وقایع
۱۲۱ منتشر کننده (ناشر)
۱۲۲ کلمه کلیدی Event
۱۲۲ فراخوانی کننده واقعه
۱۲۳ وقوع واقعه
۱۲۳ مشترک
۱۲۴ مشترک واقعه شدن

۱۲۵.....	فصل ۲۸- ژنریک ها
۱۲۵	متدهای ژنریک
۱۲۶.....	فراخوانی متدهای ژنریک
۱۲۷.....	پارامترهای نوع ژنریک
۱۲۷.....	مقدار پیش فرض
۱۲۷	کلاس های ژنریک
۱۲۸.....	وراثت کلاس ژنریک
۱۲۹	واسط ژنریک
۱۲۹	تفویض ژنریک
۱۳۰	وقایع ژنریک
۱۳۰	ژنریک ها و شی
۱۳۲.....	محدودیت های چندگانه
۱۳۲.....	علت استفاده از محدودیت ها
۱۳۴.....	فصل ۲۹- ثابت ها
۱۳۴	ثابت های محلی
۱۳۵	فیلدهای ثابت
۱۳۵	فقط خواندنی
۱۳۶.....	پارامترهای IN
۱۳۷	دستورالعمل ثابت
۱۳۸.....	فصل ۳۰- متدهای آسنکرون
۱۳۸	کلمات کلیدی Await و Async
۱۳۹.....	انواع برگشتی Async
۱۳۹.....	متدهای Async سفارشی
۱۴۰	انواع برگشتی بسط یافته
۱۴۱	استریم های آسنکرون
۱۴۳.....	واژه نامه



فصل ۱ :: دنیا سلام

انتخاب محیط توسعه تلفیقی (IDE)

برای شروع کد نویسی در C# به یک محیط توسعه تلفیقی (IDE)^۱ نیاز دارید که از میکروسافت دات نت فریم ورک^۲ پشتیبانی کند. معروف ترین گزینه خود میکروسافت ویژوال استودیو^۳ است. این IDE در نسخه سبک تر خود به نام ویژوال استودیو جامعه^۴ بصورت رایگان در دسترس است. که می توان آن را از وب سایت ویژوال استودیو^۵ دانلود کرد.

زبان C# (سی شارپ) از زمان معرفی نسخه C# 1.0 در سال ۲۰۰۲ به بعد تحت به روز رسانی های متعددی قرار گرفته است. در زمان نوشتن این کتاب نسخه جاری زبان C# 8.0 است که در سال ۲۰۱۹ منتشر شده است. هر نسخه از زبان با نسخه ویژوال استودیو متناظر است، بنابراین برای استفاده از خصیصه های C# 8.0، به نصب ویژوال استودیو نسخه ۲۰۱۹ (نسخه ۱۶/۳ یا بالاتر) نیاز دارید.

1 Integrated Development Environment

2 Microsoft .NET Framework

3 www.visualstudio.com

4 Visual Studio Community

5 www.visualstudio.com/vs/community/

توجه: زمان نصب ویژوال استودیو، اطمینان حاصل کنید که گزینه های “NET desktop development” و “NET Core cross-platform development” انتخاب شده باشد که بتوانید از C# 8.0 استفاده کنید.

ایجاد یک پروژه

پس از نصب IDE، برنامه را باز کنید. لازم است که یک پروژه جدید بسازید، که فایل های منبع C# و سایر منابع لازم را مدیریت کند. برای نمایش پنجره پروژه جدید (New Project)، به مسیر `File ► New ► Project`، در منوی فایل ویژوال استودیو بروید. از آنجا قالب “Console App (.NET Core)” را انتخاب کرده و دکمه Next را بزنید. نام و مکان پروژه را مشخص کرده و سپس دکمه Create را کلیک کنید و صبر کنید تا ویزارد برنامه پروژه جدید شما را بسازد.

حالا پروژه شما خلق شده است. در ستون Solution Explorer (View ► Solution Explorer) می توانید محتویات پروژه که شامل یک فایل منبع C# (.cs)، منفرد است را ببینید که باز است. اگر باز نبود، روی آن دابل کلیک کنید تا باز شود. در فایل منبع، مقداری کد پایه وجود دارد که برای کمک به برنامه نویس درج شده است. هرچند برای ساده ماندن محیط در این مرحله، کد زیر را در این فایل تایپ کنید:

```
class MyApp
{
    static void Main()
    {
    }
}
```

حالا این اپلیکیشن شامل یک کلاس به نام MyApp است که حاوی یک متد Main خالی است، هر دو توسط کروشه باز و بسته محدود شده اند. متد Main نقطه ورود برنامه است و بایستی این قالب را داشته باشد. حروف کوچک و بزرگ مهم است، زیرا زبان C# نسبت به حروف حساس است. کروشه مشخص می کند که چه چیزی به یک کلیت در کد تعلق دارد، مثل متعلقات کلاس یا متد که بایستی داخل کروشه باز و بسته باشند. کروشه و محتویات آن را یک قطعه یا بلوک کد یا بطور مختصر بلوک می نامند.

همانطور که همیشه در یادگیری زبان های برنامه نویسی مرسوم است، ما هم اولین برنامه خود را با سلام دنیا شروع می کنیم. در اینجا کد برنامه ای را می نویسیم که رشته متنی “Hello World” را نمایش می دهد. اینکار با اضافه کردن خط کد زیر در داخل کروشه باز و بسته متد Main انجام می شود.

```
System.Console.WriteLine("Hello World");
```

این خط کد از متد WriteLine استفاده می کند، که یک پارامتر رشته ای منفرد جدا شده با گیومه دوپل را می پذیرد. این متد در داخل کلاس کنسول (Console class) قرار داد که متعلق به فضای نام System است. توجه شود که عملگر یا اپراتور نقطه ^۱ (.) برای دسترسی به عضوی از یک فضای نام و کلاس استفاده می شود. دستورات بایستی به سمی کالن (;) ختم شود و این در مورد همه دستورات C# مصداق دارد. حالا کد شما باید شبیه این باشد:

```
class MyApp
{
    static void Main()
    {
        System.Console.WriteLine("Hello World");
    }
}
```

متد WriteLine در پایان رشته پرینت شده خط جدید ایجاد می کند. برای نمایش رشته بدون خط جدید، بایستی بجای آن از متد Write استفاده شود.

IntelliSense

زمان نوشتن کد در ویژوال استودیو، پنجره کوچکی به نام IntelliSense بصورت حبابی ظاهر می شود که چندین بدیل از پیش تعیین شده را به شما پیشنهاد می دهد تا از بین آنها انتخاب کنید. این پنجره خیلی مفید است و می توان آن را به صورت دستی و با زدن کلید ترکیبی یا شورت کات Ctrl+Space فعال کرد. این پنجره برای شما دسترسی سریع به کد کلیت های موجود در برنامه را می دهد، که شامل کلاس ها و متدهای دات نت فریم ورک همراه با شرح ها آنها است. این ویژگی خیلی قوی است که باید استفاده از آن را بیاموزید.

¹ dot operator



فصل ۲ :: کمپایل و اجرا

کمپایل کردن در ویژوال استودیو

در فصل قبلی برنامه سلام دنیا ساخته شد، حالا نوبت **کمپایل کردن**^۱ و اجرای آن است. برای انجام اینکار، منوی Debug برنامه را باز کرده و گزینه Start Without Debugging را کلیک می کنیم یا به سادگی شورت کات **Ctrl+F5** را می زنیم. سپس ویژوال استودیو برنامه را کمپایل کرده و اجرا می کند، بنابراین رشته سلام دنیا در پنجره کنسول نمایش می یابد.

کمپایل کردن در کنسول

اگر IDE چون ویژوال استودیو ندارید، هنوز می توانید برنامه را کمپایل کنید، اگر که دات نت فریم ورک روی سیستم شما نصب باشد. برای اینکار پنجره کنسول را باز کنید (C:\Windows\System32\cmd.exe) و به پوشه پروژه خود بروید، جایی که فایل منبع در آن قرار دارد. سپس لازم است که کامپایلر C# بنام csc.exe را پیدا کنید که در مسیر شما مسیر زیر قرار دارد. حال کامپایلر را با نام فایل منبع به عنوان آرگومان آن اجرا کنید و اینکار یک فایل اجرایی^۲ در فولدر جاری ایجاد می کند.

```
C:\MySolution\MyProject>  
C:\Windows\Microsoft.NET\Framework64\v3.5\ csc.exe Program.cs
```

^۱ Compilation

^۲ executable