

# آموزش سی شارپ (C#) قسمت ۴ : آشنایی با CLR و JIT و مدیریت حافظه (نسخه چاپی)

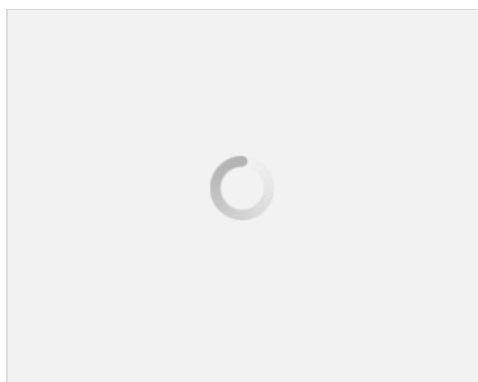
در بخش قبلی آموزش سی شارپ، با برخی مفاهیم در .NET Framework آشنا شدیم. در ادامه قصد داریم مفاهیم را بیشتر باز کرده و برخی خدمات .NET Framework را با هم بررسی کنیم. مطالبی که در این قسمت به آنها خواهیم پرداخت به شرح زیر می باشند:

۱. آشنایی با برخی سرویس های ارائه شده توسط CLR
  ۲. اجرای کدها در دات نت و سرویس JIT و اجرای مدیریت شده کدها
  ۳. مدیریت حافظه و سرویس GC
- قسمت های قبلی این آموزش از طریق لینک های زیر قابل دسترس می باشند:

۱. دوره آموزشی .NET Framework و زبان برنامه نویسی سی شارپ - مقدمه
۲. آموزش سی شارپ - قسمت اول معرفی و بررسی ابزارهای مورد نیاز دوره
۳. آموزش زبان سی شارپ - قسمت دوم آشنایی با .NET Framework و ساختار آن (بخش اول)

## آشنایی با برخی سرویس های ارائه شده توسط CLR

همانطور که در بخش قبلی خدمت دوستان گفتیم، CLR قلب تپنده دات نت بوده و اکثر وظایف زمان اجرای برنامه های دات نت به عهده CLR می باشد. در حقیقت، برای اجرای برنامه های مبتنی بر دات نت، حتماً باید نسخه مربوطه .NET Framework بر روی سیستم مقصد نصب شده باشد. در این بخش قصد داریم تا برخی از سرویس های CLR را خدمت دوستان معرفی کرده و در مورد دو سرویس، یعنی JIT و GC به تفصیل صحبت کنیم. در تصویر زیر نمای کلی از سرویس های CLR را مشاهده می کنید:



۱. **Class Loader**: این سرویس وظیفه بارگذاری کلاس ها در زمان اجرا در محیط CLR را به عهده دارد
۲. **IL to Native**: در قسمت قبل گفتیم که کدهای نوشته شده در زبان های مبتنی بر دات نت، بعد از کامپایل به یک زبان میانی به نام IL تبدیل می شوند، این سرویس وظیفه تبدیل کدهای IL به کد ماشین جهت اجرای برنامه را به عهده دارد. در این بخش به تفصیل به معرفی این سرویس می پردازیم.
۳. **Code Manager**: این سرویس وظیفه مدیریت کدها در زمان اجرا را به عهده دارد.
۴. **Garbage Collector**: با کمک این سرویس، CLR کار مدیریت حافظه را انجام می دهد، پاک سازی حافظه از اشیاء بلااستفاده از وظایف GC می باشد. در این بخش به تفصیل به معرفی این سرویس می پردازیم.
۵. **Debug Engine**: یکی از امکانات مورد نیاز برنامه نویسان، قابلیت خطایابی برنامه ها می باشد که این سرویس، قابلیت خطایابی یا همان Debugging را به برنامه نویسان می دهد.
۶. **Type Checker**: یکی از ویژگی های دات نت، **Type Safety** می باشد، بدین معنی که تمام **Data Type** ها یا نوع های داده باید مشخص باشند. برای مثال شما در دات نت نمی توانید در یک متغیر از نوع عددی، رشته ای را ذخیره کنید یا بالعکس، این سرویس وظیفه کنترل نوع های داده را به عهده دارد!
۷. **Security Engine**: یکی دیگر از سرویس های CLR می باشد که وظیفه امنیت اجرای کدها در محیط دات نت را به عهده دارد. در مورد این سرویس و امنیت در دات نت به تفصیل در فصل های آتی صحبت خواهیم کرد.

۸. Exception Manager: همدمی نه در برنمه های مبسی بر دات نت، حصایی انعاق می افند، CLR یب اسسنا یا Exception برای ان خطا ایجاد می کند، وظیفه کنترل و مدیریت استثنایا به عهده این سرویس می باشد.
۹. COM Marshaler: یکی از مزایای دات نت، قابلیت ارتباط با اجزای COM که در بخش قبلی در مورد آن صحبت کردیم می باشد، این سرویس وظیفه انتقال اطلاعات بین برنامه های دات نت و اجزای COM را به عهده دارد.
۱۰. Thread Support: یکی از ویژگی های برنامه های امروزی، قابلیت Multi-Threading جهت همزمانی انجام وظایف و عملیات ها می باشد که باعث بالارفتن کارایی برنامه می شود، این سرویس وظیفه مدیریت اجرای کدها به صورت Multi-Thread را به عهده دارد.
۱۱. Base Class Library: امکان دسترسی به کتابخانه پایه دات نت را به برنامه ها می دهد. در مورد BCL در قسمت قبلی توضیح دادیم.

## اجرای کدها در دات نت و سرویس JIT و اجرای مدیریت شده کدها

روند اجرای کدها در دات نت، کمی پیچیده ولی در عین حال، بسیار کارآمد است. زیرا طراحی این قسمت توسط شرکت میکروسافت طوری انجام شده که اجرای کدها به بهینه ترین شکل ممکن انجام شود. همانطور که در قسمت قبلی گفتیم، سرویسی که وظیفه اجرای کدها در دات نت را دارد، کامپایلر JIT یا Just-In-Time می باشد. در زیر به صورت قدم به قدم به تشریح روند کامپایل و اجرای کدها، از زمان کامپایل توسط کامپایلر زبان برنامه نویسی تا زمان اجرای کد توسط JIT می پردازیم:

۱. در قدم اول شما کد مورد نظر خود را با زبان برنامه نویسی مورد نظرتان مانند سی شارپ می نویسید.
  ۲. در ادامه، با کمک کامپایلر زبان مورد نظر، کدی که شما نوشتید به زبان IL ترجمه شده و داخل یک فایل به نام Assembly قرار داده می شوند. Assembly ها فایل هایی هستند که بعد از کامپایل کدها توسط کامپایلر ایجاد می شوند. Assembly ها شامل کد IL برنامه، Metadata ها که اطلاعاتی در مورد کد نوشته شده توسط شما در اختیار CLR قرار می دهد و Assembly Manifest که اطلاعات اولیه در مورد اسمبلی مانند نسخه و ... می باشند. (در مورد Metadata ها Manifest در بخش Reflection به تفصیل صحبت خواهیم کرد)
  ۳. بعد از ایجاد Assembly توسط کامپایلر، با اجرای برنامه، CLR وارد صحنه می شود. بدین صورت که ابتدا توسط سرویس Class Loader که در بخش قبلی در مورد آن توضیح دادیم، اطلاعات اولیه مورد نیاز جهت اجرای کد مانند کد IL و Metadata را داخل حافظه بارگذاری شده، و سپس کد بارگذاری شده توسط Class Loader توسط JIT به کد ماشین تبدیل شده و اجرا می شود.
- نکته ۱: هر اسمبلی دات نت برای اجرا نیاز به یک نقطه شروع یا Entry Point دارد که Class Loader در ابتدا این بخش، یعنی Entry Point را در حافظه بارگذاری کرده و JIT آن را اجرا می کند.
- نکته ۲: برنامه های نوشته شده توسط دات نت، می توانند شامل چندین هزار خط و بخش مجزا باشند، از مزیت های JIT این است که برای اجرای برنامه ها، کل کدها را یکجا به کد ماشین تبدیل نمی کند، بلکه تنها بخش مورد نیاز برای اجرا توسط Class Loader در حافظه بارگذاری شده و توسط JIT اجرا می شود. با این کار سرعت بارگذاری اولیه برنامه های دات نت بسیار بالا می رود.
- اما MCE یا Managed Code Execution دقیقاً" به چه معناست؟ همانطور که گفتیم، CLR پروسه ای برای اجرای کدها دارد که به شرح زیر می باشد:

۱. بارگذاری کد IL و Meta Data های مربوطه
۲. تبدیل کد IL به کد Native و اجرای آن
۳. کنترل و مدیریت حافظه در زمان اجرای برنامه
۴. اعمال Type Safety
۵. کنترل امنیت در اجرای برنامه ها
۶. مدیریت Exception ها

پروسه ایی در بالا برای اجرای کدها در دات نت ذکر شد به MCE یا همون Managed Code Execution شناخته میشه. البته مطالب در مورد این موضوع بسیار است که به دلیل تخصصی بودن مطالب، در بخش پیشرفته آموزش سی شارپ به تفصیل به موضوعات مربوط به CLR و اجرای کدها خواهیم پرداخت.

## مدیریت حافظه و سرویس GC

یکی از کابوس های برنامه نویسان، بهتره بگیم بزرگترین کابوس، مدیریت و استفاده بهینه از حافظه کامپیوتر می باشد. برنامه ای که از حافظه استفاده بهینه داشته باشد، سرعت اجرا و عملکرد بهتری دارد. اگر شما با زبان هایی که مدیریت حافظه در آنها به صورت خودکار نباشد، کار کرده باشید، حتما "واژه کابوس را تأیید می کنید. اما در دات نت، مسئله مدیریت و استفاده بهینه از حافظه بوسیله سرویس GC یا Garbage Collector به صورت کامل حل شده است. در حقیقت سرویس GC وظیفه مانیتور کردن حافظه در حین اجرای برنامه و پاک سازی آن را به عهده دارد. به همین دلیل شما به عنوان برنامه نویس، هیچ گونه نگرانی در مورد پاک سازی حافظه را نخواهید داشت.

پروسه آزاد سازی حافظه در دات نت، کمی پیچیده است، اما به طور کلی، در مدیریت حافظه توسط CLR، دو وظیفه کلی وجود دارد. تخصیص حافظه و آزاد سازی حافظه که این وظایف به صورت خودکار توسط CLR انجام می شوند. در ادامه آموزش بخشی را به سرویس GC اختصاص خواهیم داد که در آن بخش به طور کامل با پروسه های تخصیص و آزاد سازی حافظه، Finalizer ها و Generation ها در GC آشنا خواهیم شد. با پایان بخش چهارم آموزش سی شارپ، مفاهیم ابتدایی دات نت نیز به پایان می رسد. به امید خدا از بخش بعدی، کار اصلی ما، یعنی شروع نوشتن کدها به زبان سی شارپ آغاز خواهد شد. در بخش بعدی با محیط Visual Studio آشنا شده و اولین برنامه خود را به زبان سی شارپ خواهیم نوشت.

نویسنده : حسین احمدی

منبع : جزیره برنامه نویسی وب سایت توسینسو

هرگونه نشر و کپی برداری بدون ذکر منبع و نام نویسنده دارای اشکال اخلاقی است

#آموزش\_دات\_نت #آموزش\_سی\_شارپ #فریم\_ورک\_دات\_نت #آموزش\_برنامه\_نویسی #زبان\_برنامه\_نویسی\_سی\_شارپ

ارسالان علیزاده

با سلام

بنده خیلی علاقمند به یادگیری یک زبان برنامه نویسی مبتنی بر وب هستم ، متأسفانه علیرغم اینکه این توانایی رو در خودم میبینم و احساس میکنم که می بایست از ابتدا بنده در این زمینه تحصیل می کردم در دانشگاه رشته مدیریت صنعتی رو خوندم و الان ۶ ساله که توی یک بانک دولتی کار می کنم .

کار روتین دهنم رو سرویس کرده و دلم می خواد به علاقه اصلیم ( برنامه نویسی ) برگردم ، به گفته اطرافیان ذهن بسیار خلاق دارم و احساس می کنم با یادگیری برنامه نویسی میتونم از این ذهن خلاق بهترین استفاده رو بکنم .

می خواستم بنده رو راهنمایی کنید با توجه به اینکه تقریباً پیش زمینه لازم برای یادگیری مطالبی مانند مقاله فوق ندارم ( هرچند به ظاهر سری مقالات فوق قدم به قدم و از ابتدا داره مباحث رو تشریح میکنه ) باید از کجا شروع کنم

ممنون دوستان

لیلا

به نظر من کارت ول نکن

محمدجواد

من باویژوال فاکس پرو برنامه نویسی می کنم و در حال حاضر نسبت به برنامه نویسی تحت وب ناآشنا هستم فوق العاده عالیه ممنون

hani۶۶

فوق العاده عالیه ممنون

مهندس من ی خواهشی دارم اگه میشه PDF هر صفحه را زیر هر صفحه برارید تا بچه های مثل ما که اینترنت کمی دارند بتونیم بعداً از طریق آفلاین استفاده می کنیم.

با تشکر.....

حسین احمدی

ممنون از شما، از قسمت بالای صفحه، امکان دریافت مطالب به صورت PDF و همچنین نسخه قابل چاپ وجود دارد.

zahra .m

سلام استاد

میشه لطفا راجع به Exception Manager بیشتر توضیح بدید و اینکه منظور از استثناها دقیقا چیه؟

حسین احمدی

سلام، Exception ها در حقیقت خطاهایی هستند که در برنامه های دات نت اتفاق می افتن، برای مثال، زمانی که شما عددی رو تقسیم بر صفر می کنید خطای Divide By Zero رخ میده، حالا هر Exception ای که اتفاق میافته یک کلاس معادل داره، همین خطای تقسیم بر صفر کلاسی با نام DivideByZeroException براش تعریف شده که زمان وقوع اطلاعات مربوط به خطا در این کلاس در اختیار کاربر قرار میگیره. Exception Manager یک سرویس در .NET Framework هست که کار مدیریت Exception ها رو بر عهده داره.

zahra .m

متشکر استاد.

مطلب اصلی