

چندی قبل مطلبی را در این سایت در مورد معرفی الگویی که توسط آن می‌توان اعمال غیر همزمان را به صورت پی در پی انجام داد، مطالعه کردید:

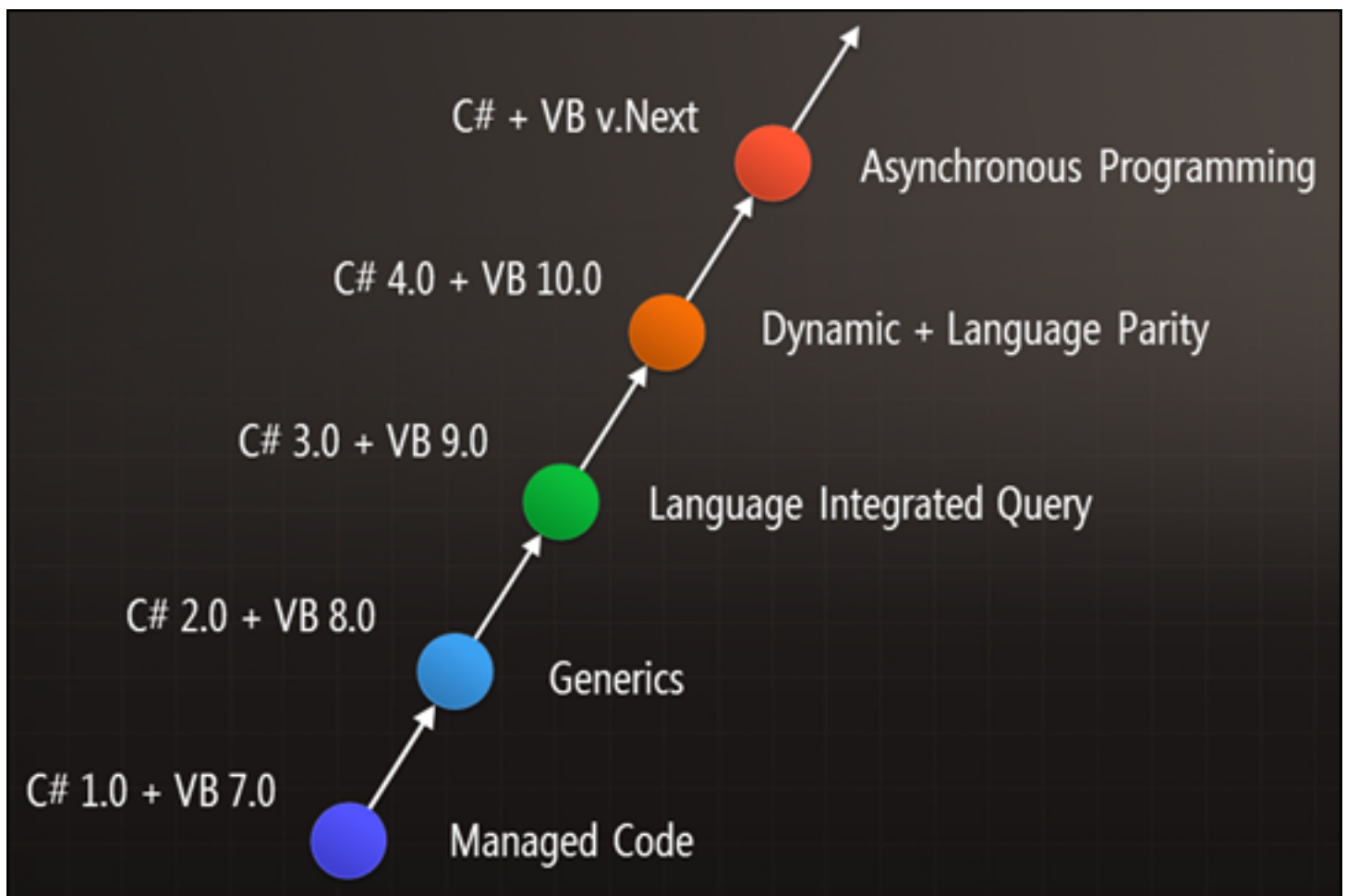
[انجام پی در پی اعمال Async به کمک Iterators - قسمت اول](#)

[انجام پی در پی اعمال Async به کمک Iterators - قسمت دوم](#)

و بحث اصلی مطالب فوق هم این است:

"در برنامه نویسی متداول همیشه عادت داریم که اعمال به صورت  $A \rightarrow B \rightarrow C$  انجام شوند. اما در Async programming ممکن است ابتدا C انجام شود، سپس A و بعد B یا هر حالت دیگری صرفنظر از تقدم و تاخر آن‌ها در حین معرفی متدهای مرتبط در یک قطعه کد. همچنین میزان خوانایی این نوع کدنویسی نیز مطلوب نیست...."

خبر خوش آن است که پشتیبانی از این نوع مدل پی در پی برنامه نویسی در نگارش‌های بعدی سی شارپ و VB.NET اضافه شده است.



لیستی از مقالات منتشر شده در این مورد را در ادامه ملاحظه خواهید کرد:

[?What is Visual Studio Async](#)

[...Visual Studio Async CTP for the rest of us](#)

[Making Asynchronous Programming Easy](#)

[C# 5 Async Exception Handling](#)

[C# 5.0 Asynchrony – A Simple Intro and A Quick Look at async/await Concepts in the Async CTP](#)

[C# 5 Async, Part 1: Simplifying Asynchrony – That for which we await](#)

[Asynchrony in C# 5, Part One](#)

[Async, Await and C# vNext](#)

[C# 5.0 Asynchronous Made Easy](#)

[!What's Next in C#? Get Ready for Async](#)

علاوه بر آن یک سری ویدیوی مرتبط با این بحث نیز منتشر شده است:

[مصاحبه با تیم طراح](#)

[مصاحبه با Anders Hejlsberg](#)

## نظرات خوانندگان

نویسنده:

محمود رمضانی

تاریخ:

۱۳۸۹/۰۸/۱۹ ۱۱:۵۰:۰۸

راستش اگه بخوام نظر شخصیم رو بگم باید بگم از سیاست جدید ماکروسافت مبنی بر تمرکز بر روی زبان برنامه نویسی به جای API ها زیاد خوشم نیاد.

درسته که نتایجش سرعت بیشتر در برنامه نویسی اما یک مشکل بزرگ داره. تا قبل از این اگه شما یک زبان برنامه نویسی رو یاد میگرفتید میتونستید خیلی آسون و سریع اون رو در زبان های دیگه هم یاد بگیرید و ازش استفاده کنید چون بیشتر API ها با زبان های C/C++ نوشته شدن بنابراین زبان ها معمولا یک API رو فراخوانی می کردن فقط نحوه فراخوانی اون ها در زبان ها فرق می کرد.

به عنوان مثال برنامه نویسی سوکت در Java, CSharp, CPlusPlus ... کاملا شبیه هم هستش. Multithreading هم به همین صورت. اگه شما با یکی از این زبان ها این Concept

ها رو یاد بگیرید میتونید همون Concept رو در بقیه زبان ها هم سریع یاد بگیرید.

اما با این سیاست جدید اگه شما مثلا برنامه نویسی Async رو در CSharp یاد بگیرید نمیتونید معلومات خودتون رو در این زمینه به زبان های دیگه ببرید و فقط میتونید ازش در دات نت استفاده کنید. خوشحال می شم نظر بقیه اساتید رو در این زمینه بدونم.

نویسنده:

وحید نصیری

تاریخ:

۱۳۸۹/۰۸/۱۹ ۱۲:۰۲:۲۵

- اتفاقا اخیرا ماکروسافت تمرکز کمتری روی زبان ها داشته و تمرکز بیشتر اون بر روی کتابخانه ها و افزونه های مرتبط با دات نت فریم ورک بوده مثل افزونه های برنامه نویسی موازی و بهبودهای زیادی که در زمینه برنامه نویسی چند ریسمانی به دات نت 4 اضافه شده. این مبحث ساده سازی async هم در جهت تکمیل این بحث است.

- هنوز هم با تمام این اضافات، حجم مستندات زبان سی شارپ در حد یک دانشنامه ی قطور مثل مستندات زبان CPP در نیامده است.

- اینکه مثلا زبان جاوا در چند سال اخیر درجا زده و مثلا در مورد LINQ یا قابلیت های پویای زبان یا همین مبحث ساده سازی Async کاری نکرده آیا به نظر شما دلیل مناسبی است برای کم کاری دیگران؟