

یکی از قدیمی‌ترین روش‌های برنامه نویسی روش برنامه نویسی تابع گراست. زبان IPL به عنوان قدیمی‌ترین زبان برنامه نویسی تابع گرا در سال ۱۹۵۵ (یک سال قبل از خلق فرترن) است. دومین زبان تابع گرا زبان LISP بوده است که در سال ۱۹۵۸ (یک سال قبل از خلق کوپول) متولد شد. هر دو زبان کوپول و فرترن زبان‌های امری و رویه ای بودند. بعد از آن‌ها در سال ۱۹۷۰ شروع عرصه زبان‌های شی گرا بود و تا امروز بیشترین کاربرد را در تولید نرم افزارها داشته اند.

F# یک زبان برنامه نویسی تابع گرا است و گزینه ای بسیار مناسب برای حل مسایل کامپیوتری. اما استفاده از زبان برنامه نویسی تابعی محض برای نوشتن و تولید پروژه‌های نرم افزاری مناسب نمی‌باشد. به همین دلیل نیاز به استفاده از این زبان‌ها در کنار سایر زبان‌های شی گرا احساس می‌شود. F# یک زبان همه منظوره دات نت است که برای حالت اجرا به صورت همه منظوره استفاده می‌شود. برخی زبان‌های تابع گرا دیگر نظیر Lisp و Haskell و OCaml (که F# بسیار نزدیک به این زبان می‌باشد) با دستورات زبان اجرای سفارشی کار می‌کنند و این مسئله باعث نبود زبان برنامه نویسی چند فعالیته می‌شود. شما می‌توانید از برنامه نویسی توصیفی هم استفاده کنید و توابع را به راحتی با هم ترکیب کنید و یا روش‌های شی گرایی و دستوری را در همان برنامه استفاده کنید.

تاریخچه

F# توسط دکتر دون سیم ابداع شد. در حال حاضر F# وابسته به تیمی کوچک ولی پیشرفته واقع در مرکز تحقیقات شرکت مایکروسافت می‌باشد. F# مدل خود را از روی زبان برنامه نویسی OCAML انتخاب کرد و سپس با گسترش قابلیت‌های فنی، خود را در دات نت گنجانده. F# در بسیاری از برنامه‌های بزرگ دنیای واقعی استفاده شده است که این خود نمایانگر آکادمیک نبودن محض این زبان است. با توجه به اینکه زبان تابع گرای دیگر به ندرت در دات نت توسعه پیدا کرده است F# به عنوان استاندارد این مقوله در آمده است. زبان F# از نظر کیفیت و سازگار بودن با دات نت و VisualStudio بسیار وضعیت بهتری نسبت به رقبا خود دارد و این خود دلیلی دیگری است برای انتخاب این زبان.

استفاده در دات نت

F# کاملاً از دات نت پشتیبانی می‌کند و این قابلیت را به برنامه نویسان می‌دهد که هر چیزی را که در سایر زبان‌های دات نت استفاده می‌کنند در این زبان نیز قابل استفاده باشد. همچنین می‌تواند برای کد نویسی IL نیز استفاده شود. F# به راحتی قابل اجرا در محیط لینوکس و مکینتاش نیز است.

استفاده کنندگان F#

F# در شرکت مایکرو سافت به شدت استفاده می‌شود. رالف هربریش که یکی از مدیران دوگانه گروه بازی‌های مایکروسافت و از متخصصین آموزش ماشین است در این باره می‌گوید:

*اولین برنامه کاربردی برای انتقال ۱۱۰ گیگا بایت از طریق ۱۱۰۰۰ فایل متنی در بیش از ۳۰۰ دایرکتوری و وارد کردن آن‌ها در دیتابیس بود. کل برنامه ۹۰ خط بود و در کمتر از ۱۸ ساعت توانست اطلاعات مربوطه را در SQL ذخیره کند. یعنی ده هزار خط برنامه متنی در هر ثانیه مورد پردازش قرار گرفت. همچنین توجه کنید که من برنامه را بهینه نکردم بلکه به صورت کاملاً عادی نوشتم. این جواب بسیار قابل توجه بود زیرا من انتظار داشتم حداقل یک هفته زمان ببرد.

دومین برنامه، برنامه پردازش میلیون‌ها Feedback مشتریان بود. ما روابط مدلی زیادی را توسعه دادیم و من این روابط را در F# قرار دادم و داده‌های مربوط به SQL را در آن فراخوانی کردم و نتایج را در فایل داده ای MATLAB قرار دادم و کل پروژه در حد صد خط بود به همراه توضیحات. زمان اجرای پروژه برای دریافت خروجی ده دقیقه بود در حالی که همین کار را توسط برنامه C# قبلاً توسعه داده بودیم که بیش از هزار خط بود و نزدیک به دو روز زمان می‌برد.*

استفاده از F# تنها در مایکروسافت نیست بلکه در سایر شرکت‌های بزرگ و نام دار نیز استفاده می‌شود و همچنان نیز در حال افزایش است. شرکت Derivative One که یک شرکت بزرگ در تولید نرم افزارهای شبیه ساز مالی است مدل‌های مالی نرم افزارهای خود را در F#

پیاده سازی کرده است.

چرا F#؟

همیشه باید دلیلی برای انتخاب یک زبان باشد. در حال حاضر F# یکی از قدرتمندترین زبان‌های برنامه نویسی است. در ذیل به چند تا از این دلایل اشاره خواهیم کرد:

F# یک زبان استنباطی است. برای مثال در هنگام تعریف متغیر و شناسه نیاز به ذکر نوع آن نیست. کامپایلر با توجه به مقدار اولیه تصمیم می‌گیرد که متغیر از چه نوعی است.

بسیار راحت می‌توان به کتابخانه قدرتمند دات نت دسترسی داشت و از آن‌ها در پروژه‌های خود استفاده کنید.

F# از انواع روش‌های برنامه نویسی نظیر **تابعی، موازی، شی گرا و دستوری** پشتیبانی می‌کند.

برخلاف تصور بعضی افراد، در F# امکان تهیه و توسعه پروژه‌های وب و ویندوز و حتی WPF و Silverlight هم وجود دارد.

نوع کدنویسی و syntax زبان F# به برنامه نویسان این اجازه را میدهد که الگوریتم‌های پیچیده مورد نظر خود را بسیار راحت‌تر پیاده سازی کنند. به همین دلیل بعضی برنامه نویسان این زبان را با Python مقایسه می‌کنند.

F# به راحتی با زبان C# و VB تعامل دارد. یعنی می‌تونیم در طی روند تولید پروژه از قدرت‌های هر سه زبان بهره بگیریم.

طبق آمار گرفته شده از برنامه نویسان، F# به دلیل پشتیبانی از نوع داده ای قوی و مبحث Unit Measure، خطاها و Bug‌های نرم افزار را کاهش می‌دهد.

به دلیل پشتیبانی VS.Net از زبان F# و وجود ابزار قدرتمند برای توسعه نرم افزار به کمک این زبان (unitTesting و ابزارهای debugging و ..) این زبان تبدیل به قدرت‌های دنیای برنامه نویسی شده است.

F# یک زبان بسیار مناسب برای پیاده سازی الگوریتم‌های data-mining است.

F# از immutability در تعریف شناسه‌ها پشتیبانی می‌کند. (در فصل‌های مربوطه بحث خواهد شد)

.....

چرا F# نه؟

F# هم مانند سایر زبان ها، علاوه بر قدرت بی همتای خود دارای معایبی نیز می‌باشد. (مواردی که در پایین ذکر می‌شود صرفاً بر اساس تجربه است نه مستندات).

نوع کدنویسی و syntax زبان F# برای برنامه نویسان دات بیگانه (و البته کمی آزار دهنده) است. اما به مرور این مشکل، تبدیل به قدرت برای مانورهای مختلف در کد می‌شود.

درست است که در F# امکان تعریف اینترفیس وجود دارد و یک کلاس می‌تواند اینترفیس مورد نظر را پیاده سازی کند ولی هنگام فراخوانی متدهای کلاس (اون هایی که مربوط به اینترفیس است) حتماً باید instance کلاس مربوطه به اینترفیس cast شود و این کمی آزار دهنده است. (در فصل شی گرای در این مورد شرح داده شده است).

زبان F# در حال حاضر توسط VS.Net به صورت Visual پشتیبانی نمی‌شود. (امکاناتی نظیر drag drop کنترل‌ها برای ساخت فرم و). البته برای حل این مشکل نیز افزونه هایی وجود دارد که در جای مناسب بحث خواهیم کرد.

آیا برای یادگیری F# نیاز به داشتن دانش در برنامه نویسی C# یا VB داریم؟

به طور قطع نه. نوع کد نویسی (نه مفاهیم) در F# کاملاً متفاوت در C# است و این دو زبان از نظر کد نویسی شباهتشان در حد صفر است. برای یادگیری F# بیشتر نیاز به داشتن آگاهی اولیه در برنامه نویسی (آشنایی با تابع، حلقه تکرار، متغیرها) و شی گرای (مفاهیم کلاس، اینترفیس، خواص، متدها و...) دارید تا آشنایی با C# یا VB.

چگونه شروع کنیم؟

اولین گام برای یادگیری آشنایی با نحوه کد نویسی F# است. بدین منظور در طی فصول آموزش سعی بر این شده است از مثال‌های بسیار زیاد برای درک بهتر مفاهیم استفاده کنم. تا جای ممکن برای اینکه تکرار مکررات نشود و شما خواننده عزیز به خاطر مطالب واضح و روشن خسته نشوید از تشریح مباحث واضح خودداری کردم و بیشتر به پیاده سازی مثال اکتفا نمودم.

نظرات خوانندگان

نویسنده: نریمان
تاریخ: ۱۵:۱۶ ۱۳۹۲/۰۴/۲۳

با تشکر از این مجموعه خیلی خوب. مفاهیم کلی F شارپ رو به طور مناسبی خلاصه کردید. اما چندتا نکته به نظرم رسید که اگر اصلاح بشه، بهبود قابل توجهی ایجاد می‌شه.

۱- خیلی از اصطلاح‌ها به فارسی برگردانده شده‌اند اما معادل انگلیسی همه آن‌ها ذکر نشده. به نظرم ذکر عبارت انگلیسی آن حتی از استفاده عبارت فارسی مهمتره، چون در نهایت کاربر برای یادگیری بیشتر باید از منابع انگلیسی استفاده کنه. پس چه بهتر که این‌جا یک مقدار آشنا بشه.

۲- کاش جایی که درباره تابع‌گرا بودن صحبت می‌کردید، از ابتدا درباره مفهوم اصلی اون و تفاوتی که بین شی‌گرایی وجود داره بحث می‌شد و اون رو زمان یاد دادن نحوه تعریف تابع و ... موکول نمی‌کردید.

یعنی مقداری درباره تفاوتی که شی‌گرایی بین فیلد و متد می‌گذاره و این‌جا با تابع و شناسه از یک جنس برخورد می‌شه (اگه اشتباه می‌کنم لطفاً تصحیح بفرمایین)، از نظر ریاضی چند تا مثال می‌زدید تا مفهوم اون جا بیفته.

۳- یه جاهایی لحن متن بین رسمی و محاوره گیر کرده. یعنی یه جایی خیلی رسمی شده، یه جایی نه. اگه یه مقدار یکدست‌تر بنویسید، تمرین نگارشیه خوبیه: دی