

آشنای اشیاء

«هر ابلمی می تواند کدی بنویسد که توسط ماشین قابل درک باشد. برنامه نویسان خوب کسانی هستند که کدهایشان برای انسان ها قابل درک باشد.» «نقل از کتاب Refactoring»

«احمد شریف پور

مارتین فاولر متولد سال ۱۹۶۳ و السال (Walsall) در انگلستان است. وی که از اواسط دهه هشتاد وارد صنعت نوپای آن زمان یعنی محاسبات شده است، از همان ابتدای ورود مجذوب زبان های برنامه نویسی شیء گرا شده و به گفته خودش بیشتر دهه نود میلادی را صرف مشاوره، آموزش و کمک به افراد برای نوشتن برنامه های شیء گرا کرده است. در این مدت، بیشتر تمرکز وی بر نرم افزارها و کاربردهای سازمانی بوده است. طراحی و تحلیل سیستم های شیء گرا، UML، الگوهای توسعه و کدنویسی چابکانه (Agile Software Development) از حیطه های تخصصی وی است. وی کار خود را در زمینه سیستم های نرم افزاری سازمانی با زبان های SmallTalk و ++C شروع کرده و اکنون به جاوا، #C و روبی روی آورده است. او در سایتش (martinfowler.com) آورده است: «علاقه مندی اصلی من درک نحوه طراحی سیستم های نرم افزاری و از این طریق افزایش بهره وری تیم های توسعه دهنده است.» به همین دلیل، او به مطالعه الگوهای توسعه درست نرم افزار پرداخته و به یکی از پایه گذاران و طرفداران سبک توسعه چابکانه یا Agile Development تبدیل شده است. فاولر که از اعضای گروه Agile Alliance است، به کمک پانزده عضو دیگر این گروه در سال ۲۰۰۱ مانیفست توسعه چابکانه نرم افزار را منتشر کرده اند.

- ۱۹۹۶- الگوهای تحلیلی: مدل های شیء قابل استفاده دوباره (Analysis Patterns: Reusable Object Models)
- ۱۹۹۹- ری فکتورینگ: بهبود طراحی کدهای موجود (Refactoring: Improving the Design of existing Code)
- ۲۰۰۰- زمینه سازی برنامه نویسی حداکثری (Planning Extreme Programming)
- ۲۰۰۲- الگوهای معماری برنامه های سازمانی (Patterns of Enterprise Application Architecture)
- ۲۰۰۳- چکیده UML: راهنمای مختصر زبان استاندارد مدل سازی شیء (UML Distilled: A Brief Guide to the Standard Object Model Language)
- ۲۰۱۰- زبان های حوزه های خاص (Domain-Specific Languages)

او علاوه بر ارائه مشاوره و نوشتن کتاب، سخنران خوبی نیز به شمار می رود و در کنفرانس هایی نظیر XP 2001، UML World، OOPSLA تا XP 2005 و همچنین Agile Universe جزء سخنرانان یا هیئت داوران بوده است.

در انتها بد نیست به Bliki سیستم انتشار محتوایی که فاولر در سایت رسمی خود از آن استفاده می کند نیز اشاره کنیم. این سیستم ترکیبی از قابلیت های blog و wiki است. به این معنی که نوشته ها یا به اصطلاح پست ها همانند وبلاگ ها با ترتیب زمانی نزولی در یک صفحه مرتب می شوند، اما ویرایش آن ها با سیستمی شبیه و یکی ها صورت می گیرد. اگرچه ابداع این اصطلاح را به فاولر نسبت می دهند، اما به اعتقاد خود وی، این سیستم و این نام در حدود سال های ۲۰۰۳ توسط یکی از بنیان گذاران ویکی، یعنی «وارد کانینگهام» ابداع شده است.

او پیش از مهاجرت به امریکا در سال ۱۹۹۴ به مدت یک دهه در لندن زندگی می کرده و از مارس ۲۰۰۰ نیز به عنوان دانشمند ارشد در شرکت ThoughtWorks مشغول کار است. شرکت ThoughtWorks در زمینه یکپارچه سازی سیستم ها و همچنین ارائه خدمات مشاوره فعالیت می کند. وی هم اکنون در ملرز (Melrose) در نزدیکی بوستون زندگی می کند.

سابقه فاولر در زمینه نویسندگی درست به اندازه سابقه اش در حیطه برنامه نویسی درخشان است. او به مدت پنج سال نویسنده ستون «طراحی» (Design) در نشریه IEEE Software بوده است و در اواخر دهه ۹۰ نیز ستون ثابتی در مجله Distributed Computing داشته است. کتاب الگوهای معماری برنامه های سازمانی

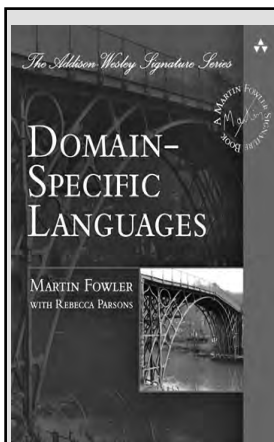
برنامه نویسی درخشان است. او به مدت پنج سال نویسنده ستون «طراحی» (Design) در نشریه IEEE Software بوده است و در اواخر دهه ۹۰ نیز ستون ثابتی در مجله Distributed Computing داشته است. کتاب الگوهای معماری برنامه های سازمانی



Domain Specific Languages

کتاب زبان های خاص

معرفی عمیق و کاملاً کاربردی زبان های DSL که با شیوه ای الگو گرایانه برای برنامه نویسان مبتدی و متوسط نوشته شده است.



عنوان: کتاب زبان های خاص
نویسنده: مارتین فاولر
تاریخ انتشار: اکتبر ۲۰۱۰
ناشر: Addison-Wesley
تعداد صفحات: ۶۴۰ صفحه

نزدیک شدن به مسئله و حل آن را به برنامه نویس آموزش می دهد. این روش ها با بیشتر زبان های شیء گرای مدرن قابل پیاده سازی هستند. همچنین در این کتاب مثال های متعددی با زبان های جاوا، C# و روبي آورده شده است و فصل های مختلف کتاب تا حد امکان مستقل از یکدیگر طراحی و نوشته شده اند. مباحثی که در این کتاب پوشش داده شده اند، عبارتند از:

- تفاوت و شباهت DSL ها با فریم ورک ها و کتابخانه ها چیست و چه زمانی استفاده از این جایگزین ها مناسب تر است.
- نحوه استفاده از Parser ها و Parser Generator ها برای Pars کردن انواع خارجی زبان های DSL - درک، مقایسه و انتخاب ساختارهای مناسب یک زبان DSL

- تعیین نیاز یا عدم نیاز به Code Generator ها و مقایسه راهبردهای مختلف این فرآیند
- بررسی اجمالی ابزارهای جدید برای ایجاد زبان های DSL جدید.
- برنامه نویسان زبان های DSL را دوست دارند، زیرا برنامه نویسی موارد خاص با یک زبان DSL که به خوبی طراحی شده باشد، بسیار ساده تر از یک کتابخانه توابع معمولی است و این امر بهره وری برنامه نویس را بالاتر می برد. این افزایش بهره وری همواره برای برنامه نویسان ارزشمند است.

از سوی دیگر، زبان های مختص یک حوزه خاص برقراری ارتباط میان توسعه دهندگان نرم افزار و متخصصان آن حوزه (که یکی از معضلات همیشگی برنامه نویسی در محیط های خاص است) را نیز ساده تر می کند. با این اوصاف، اگر در حال توسعه نرم افزار برای محیط های سازمانی یا صنعتی خاص هستید یا به این نوع زبان ها علاقه مند باشید، این کتاب اگر نگوییم تنها مرجع موجود، دست کم مشهورترین و معتبرترین آن ها در حوزه فعالیت شما به شمار می رود.

زبان های خاص بنا به تعریف به زبان هایی با خوانایی محدود (برای افراد) گفته می شود که حیطة خاصی از محاسبات را هدف گرفته اند. این زبان ها اگر به درستی انتخاب و به کار برده شوند، می توانند در حوزه کاربردی خود، کدهای پیچیده زبان های دیگر را به کدهای ساده و خوانا تبدیل کرده، ارتباط مؤثرتر با مشتریان را فراهم کرده، بهره وری را بالاتر برده و گلوگاه های فرآیند کدنویسی را از بین ببرند. این زبان ها معمولاً کوچک هستند و بر جنبه خاصی از یک سیستم نرم افزاری متمرکز شده اند. با این زبان ها نمی توان یک برنامه کامل نوشت، بلکه معمولاً از چندین و چند زبان DSL (سرنام Domain Specific Language) در یک سیستم نرم افزاری که خود معمولاً با یک زبان چندمنظوره نوشته شده است، استفاده می شود.

زبان های DSL در دو فرم کلی ظاهر می شوند. نوع نخست یا DSL های خارجی، زبان هایی هستند که مستقل از زبان چند منظوره میزبان بررسی و کنترل (parse) می شوند. کدهای CSS و Regular Expressions نمونه های بارز این نوع زبان ها هستند. زبان های DSL خارجی سابقه ای طولانی در دنیای یونیکس دارند. در مقابل زبان های DSL داخلی فرم خاصی از API ها در زبان چندمنظوره میزبان هستند. کتابخانه JMock و بسیاری از مکانیسم های به کار رفته در Ruby On Rails از نمونه های این نوع زبان ها هستند. این زبان ها به ویژه در جامعه برنامه نویسان Lisp از سابقه ای طولانی برخوردار است.

به رغم این که مدت مدیدی از پدیدار شدن زبان های DSL می گذرد، اما فقدان دانش برنامه نویسی با این زبان ها مانع بزرگی در توسعه و فراگیری آن ها است و به همان میزان هم در کاهش سرعت و بهره وری پروژه های خاص نرم افزاری مؤثر است. فاولر دقیقاً به همین دلیل به سراغ این بحث خاص رفته است که کمتر توسط سایرین مورد بررسی و تخصص قرار گرفته است. کتاب روش هایی را برای توسعه برنامه با زبان های DSL داخلی و خارجی معرفی می کند و به این ترتیب اطلاعات کافی را برای شروع کار و همین طور انتخاب بین نوع داخلی یا خارجی آن ها فراهم می کند. همچنین در کتاب بر استفاده از زبان DSL به عنوان لایه ای روی یک کتابخانه معمولی تأکید فراوانی شده است. به این ترتیب، برنامه نویس این دو مورد را با هم پیش برده و روش هایی را نیز در زمینه تولید کد (Code Generation) یاد خواهد گرفت.

همان گونه که پیش تر گفته شد، در ابتدای کتاب اطلاعات لازم برای تصمیم گیری در مورد زمان و همین طور لزوم یا عدم لزوم استفاده از زبان های DSL آورده شده است. هنگامی که استفاده از این زبان ها مناسب تشخیص داده شود، روش های ارائه شده در کتاب، شیوه های