

برنامه نویسی خودکار

امیر بهاء الدین سبط الشیخ

برنامه نویسی خودکار^۱ به گونه ای از برنامه نویسی کامپیوتری گفته می شود که در آن به کمک مکانیزم هایی یک برنامه کامپیوتری ایجاد می کنند. این گونه برنامه نویسی درست نقطه مقابل برنامه نویسی کامپیوتر توسط برنامه نویسی (فرد) است.

البته در این تعریف توافق کمی وجود دارد و به سختی می توان به تعریفی دقیق و جامع رسید، چرا که معنای برنامه نویسی خودکار در طول زمان تغییر کرده است.

دیوید پاراناس^۲، یکی از ارکان اساسی پنهان سازی اطلاعات در برنامه نویسی ماژولار، تاریخچه این واژه را بررسی و اشاره کرده است که در دهه ۴۰ میلادی، برنامه نویسی خودکار به مکانیزه کردن امور لازم برای سوراخ کردن کاغذهای کد می گفتند. اما بعدها به ترجمه زبان های برنامه نویسی سطح بالایی چون فورترن^۳ و الگول^۴ اطلاق شده است. در حقیقت، یکی از نخستین کامپایلرهای دنیا، به نام اتوکد^۵ ایجاد شده بود.

دیوید پاراناس همچنین نتیجه گیری کرده است که: «برنامه نویسی خودکار همواره تعبیر یک زبان سطح بالاتر به زبانی سطح پایین تر بوده که برای برنامه نویسی ساده تر باشد.»

۱. برنامه نویسی نسلی

برنامه نویسی نسلی^۶ نوعی از برنامه نویسی کامپیوتر است که طی آن، خودش اقدام به تولید کد



منبع برنامه می کند. این تولید به کمک استفاده از جنریک ها^۷، فریم ها^۸، کلاس ها، الگوها^۹ و دیدگاه ها^{۱۰} انجام می شود و باعث بالاترین میزان بهره وری می گردد. در مباحث استفاده مجدد از کد، به شیوه تولید خودکار کد، گاه به «مهندسی نرم افزار مولفه محور^{۱۱}» و «مهندسی خانواده محصول^{۱۲}» نیز نسبت داده می شود.

۲. تولید کد منبع

تولید کد منبع، یعنی عمل ایجاد کد منبع بر مبنای مدلی آنتولوژیک^{۱۳} مانند یک الگو که با یک ابزار برنامه نویسی مثل پردازنده الگو یا محیط توسعه یکپارچه (IDE)^{۱۴} همراه باشد. این ابزارها توسعه کد منبع را به مقاصد مختلفی انجام می دهند. برای مثال، یکی از مفاهیم پایه برنامه نویسی خودکار، پردازنده ماکرو در زبان C است که به سادگی کد منبع را با ماکروهای



نوشته شده در کد جایگزین می کند و قوانین ساده ای دارد.

پیاده سازی

محیط های توسعه یکپارچه ای چون Interface Builder (برای سیستم های Mac OS X) و یا مایکروسافت ویژوال استودیو (برای سیستم های ویندوزی) شکل های پیشرفته تر و پیچیده تری از تولید خودکار کد را انجام می دهند، در آن ها برنامه نویسی می تواند به صورت تعاملی و پویایی تکه کدهای منبع خود را ایجاد یا ویرایش کند. ویزاردهای برنامه به سادگی و با محیطی گرافیکی به برنامه نویسی اجازه می دهند رابط کاربری مورد نظر خود را ایجاد کنند و کامپایلر به صورت نامرئی کد منبع لازم برای ایجاد آن رابط کاربری را تولید کند. این ها هم شکل دیگری از تولید خودکار کد به شمار می روند. صرف نظر از تولید کد بر مبنای ویزارد یا الگو، محیط های توسعه یکپارچه همچنین کد را ریفاکتور^{۱۵} می کنند تا در مقابل وقوع خطا مقاوم سازی شود، این روند اگر به روش دستی انجام شود، وقت بیشتری را صرف می کند.



دشوارتر و وقت گیرتر از حالت دستی است.

چند مثال از نرم افزارهای برنامه نویسی خودکار

Altova MapForce

ابزار یکپارچه سازی، تبدیل و نگاشت داده که قادر به تولید کد به زبان های جاوا، C# یا C++ است.

NConstruct

ابزاری است برای پیاده سازی نرم افزارها با معماری توسعه سریع، در محیط های وب یا ویندوز و تحت محیط .NET فعالیت می کند و کد تولیدی آن، به زبان C# است و با ویزاردهای ساده خود، تمام کدهای لازم برای نرم افزار را تولید می کند.

MyGeneration

تولیدکننده کد بر مبنای الگوهاست که برای دات نت فریم ورک مایکروسافت نوشته اند.

Ebenzeer Eterprises

سرویس آنلاینی است که عملیات مارشال کردن کد را به زبان C++ انجام می دهد.

LatticeFramework Studio

ابزار توسعه سریع نرم افزار بر مبنای مدل و الگو است که برای محیط دات نت نوشته شده است. این ابزار کدهایی به زبان های C#، VB.NET، جاوا، PHP و دیگرام ها و مستندات را طی پروسه های ویزاردی بسیار ساده تولید می کند.

MyProLang

ابزار تولید خودکار کد است که بر مبنای الگو فعالیت می کند و می تواند از رابط کاربری تا کدهای عملیاتی را ایجاد کند و در یک برنامه اجرایی دیگر کامپایل شود.

Viewpoints

ابزاری برای مدل سازی تصویری و تولید کد برای ویژوال استودیو دات نت است که محوریت فعالیت های تجاری و مولفه های مرتبط را مدنظر قرار داده است.

منابع

- 1. http://www.codegeneration.net/files/JavaOne_OnePageGuide_v1.pdf
- 2. http://en.wikipedia.org/wiki/Automatic_programming
- 3. <http://www.generative-programming.org/>

پی نوشت ها

- 1. Automatic Programming
- 2. David Parnas
- 3. Fortran
- 4. Algol
- 5. AutoCode
- 6. Generative Programming
- ۷. Generics: شیوه ای در برنامه نویسی که در آن الگوریتم ها به نوع «بعدها معرفی خواهد شد» پیاده سازی می شوند.
- ۸. Frames: نوعی فناوری در تولید نرم افزار است که به کمک بلوک هایی از پیش ساخته، کدهای لازم را ایجاد می کند.
- 9. Templates
- 10. Aspects
- 11. Component-Based Software Engineering
- 12. Family-Based Engineering
- ۱۳. Ontological: آنتولوژیک یعنی شکل ظاهری مجموعه ای از مفاهیم مرتبط با یکدیگر در یک دامنه مشخص.
- 14. Integrated Development Environment
- ۱۵. Refactor: تغییر ساختار درونی برنامه، بدون آن که در رفتار و عملکرد آن تاثیر داشته باشد.

مزایای تولید خودکار کد تولید کد فقط سرعت کدنویسی را بالا نمی برد. بلکه چهار امتیاز برجسته نسبت به تولید کد به روش دستی دارد:

(الف) کیفیت: تولیدکننده های کد از الگوهایی برای کدهای لازم استفاده می کنند. بنابراین هر چه این الگو بهتر باشد، کد خروجی بهتر خواهد بود. بنابراین تنها کافی است برای بالابردن کیفیت الگوهای کد تلاش کرد تا تاثیر آن را در تمام کد منبع یک نرم افزار مشاهده کرد.

(ب) ثبات: کلاس، متدها و نام گذاری متغیرها در این روش ثبات بیشتری دارد و باعث می شود از رابط کاربری راحت تر استفاده کرد. همچنین امکان افزودن کدهای تولیدی دیگری روی نرم افزار فعلی ساده تر خواهد بود.

(پ) بهره وری: این عامل رابطه مستقیم با مهارت تولیدکننده کد دارد تا بتواند با تغییراتی در کد، کل پروژه را به سمتی برساند که به ذهن برنامه نویسی نزدیک تر شود.

(ت) تجرید: تولیدکننده هایی که نیازهای لازم یک نرم افزار را داخل بستری خاص نگه می دارند (مثلا به شکل دیتابیس یا فایل XML) از زبان پیاده سازی فاصله می گیرند و به تجرید نزدیک تر می شوند. به همین جهت می توان مطالعه بعدی روی تمام پروژه را صرف نظر از زبان پیاده سازی و شیوه آن انجام داد و در صورت لزوم تغییرات را روی آن ها اعمال کرد.

معایب تولید خودکار کد

درست مانند هر تکنیک دیگری، تولید کد نیز معایب خود را دارد. از مهم ترین و دغدغه هایی که در بحث تولید خودکار کد وجود دارد، به ترتیب زیر است: تولیدکننده های کد معمولاً غیر قابل تعمیرند و به همین دلیل مهندسان نرم افزار از آن دوری می کنند. همچنین گاهی اوقات کد تولیدی بسیار پیچیده می شود و مرور و رفع عیب آن به مراتب

