

MARKO BIKKER

"GOTO JAVA"

'Evolutie of nieuwe creatie?'

Ik herinner het me als de dag van (eer)gisteren. Lekker programmaatjes schrijven in BASIC op de P2000, Atari, MSX, C64 of welke nostalgische voorloper van de huidige duo core machines dan ook. Van objectgeoriënteerd programmeren had nog nooit iemand gehoord (ik in ieder geval niet). Enige gestructureerdheid was vaak ver te zoeken, vanwege misschien wel het meest misbruikte BASIC statement: GOTO. Maar als ik alle BASIC programma's die ik ooit geschreven heb achter elkaar zou uitprinten, zou de stapel kettingpapier (ik blijf in nostalgische stijl) waarschijnlijk net zo hoog worden als ikzelf.

Omdat BASIC ook zijn beperkingen kent en bovendien niet voorhanden was op de UNIX terminals waar ik op werkte, ging ik steeds meer over op de meer 'professionele' talen Pascal en C/C++. En dat laatste werd bij mij favoriet. Heerlijk om te programmeren in een taal die zo low-level is, dat je er eigenlijk alles in kon doen – en er daarmee ook alles in fout kon doen. Maar als het debuggen eenmaal achter de rug was dan had je ook wat!

Aan het eind van de twintigste eeuw nam de rekenkracht en de geheugengrootte van computers zo snel toe, dat snelheid en efficiëntie niet de enige argumenten bleven bij het ontwikkelen van software. Een programma dat geschreven is in een taal als C is weliswaar heel snel en efficiënt, maar het kan niet zomaar op een ander platform worden uitgevoerd. Wat zou het toch gemakkelijk zijn als...

Java werd geboren. Of hoe je het maar wil uitdrukken: misschien kun je het zien als evolutie vanuit oudere programmeertalen of juist als een totaal nieuwe creatie. Eén ding is zeker: er zit een bepaalde mate van intelligent design achter.

Kritisch als ik ben stond ik in het begin nogal sceptisch tegenover het concept van Java: een platform-onafhankelijke taal die in gecompileerde vorm niet op een echte (fysieke) processor draait, maar op een virtuele machine. De processor wordt als het ware gesimuleerd

Marko Bikker is afgestudeerd in de technische cognitiewetenschap (tegenwoordig: kunstmatige intelligentie).



door een stukje software. Daar gaat je snelheid en efficiëntie! En in het begin was dat dan ook een probleem: een Java programma was eigenlijk per definitie log en traag. Ik herinner me hoe ik de Oracle Universal Installer (gemaakt in Java) moest uitvoeren op een Windows 95 machine met 16 MB geheugen. Je kunt nog beter op je spreekwoordelijke hoofd gaan staan en wachten tot je geitenwollen sokken afzakken.

Maar ik moet eerlijk toegeven: er is sindsdien echt wat veranderd! In de nieuwere Java versies worden technieken gebruikt als just-in-time compilation (JIT) en dynamic recompilation. Deze features zorgen voor een veel efficiëntere uitvoering, waarbij (een deel van) de code on-the-fly wordt doorgecompileerd naar processorniveau. Hierdoor is de performance vergelijkbaar geworden met die van in andere talen geschreven en gecompileerde software. Het grote voordeel van Java blijft echter de platformonafhankelijkheid. Deze eigenschap is in combinatie met de opkomst van het internet gedurende de afgelopen tien jaar waarschijnlijk de belangrijkste reden geweest voor het succes van Java.

En terecht. Ruim tien jaar na de introductie kan ik nog maar één conclusie trekken. In BASIC terminologie: GOTO Java!