

Ik moet toegeven dat ik er in eerste instantie blij verrast van opveerde. Iemand wees me op een recent artikel van twee professoren aan de Universiteit van New York die betogen dat Java een ongeschikte taal is om programmeren mee te leren. Wat had ik daar gelijk een herkenning bij!

Programmerende manager

Java en ik, we gaan al jaren nogal moeizaam door één deur. Echt een naar, dikdoenerig programmeertaaltje. Veel te complex ook. Syntactisch gezien een beetje alles in huis hebben om een marslander of een kerncentrale mee aan te sturen en er dan heel saai de zoveelste voorraad-, polis- of klantenadministratie mee uitpersen. Alsof iemand in een fout, zilverkleurige hemdje in de sportschool zijn spieren staat te showen, zo vliegen je de *inheritance*, *multiple polymorphism* en *inner classes* om de oren. Protserig krachtsvertoon dat je ondertussen bijna jankend smeekt om fouten te maken en improductief te zijn.

En dan met objectoriëntatie integraal ingebakken, toch al zo'n vreselijk *overrated* concept. Hoewel, het is ook wel weer knap van het vakgebied om een structureringsprincipe te vinden dat zó optimaal slecht aansluit op het relationele datamodel. En iedereen maar dikke boeken schrijven over *object-to-relational mappers*: met veel geweld een appel door een theezeef weten te duwen en nog trots zijn op de pulp ook. Ja, we kunnen ècht wel wat in de IT.

Maar komaan, dat artikel dus.

Eindelijk worden man en paard genoemd, dacht ik nog. Maar al snel kwam ik erachter dat beide wetenschappers aan de andere kant van het spectrum verkeren. Vanuit hun perspectief is Java juist een lichtgewicht middel waarmee je lachend snel wat geinige Internetdingetjes in elkaar zet. Als het zo eenvoudig wordt om te programmeren, dan hoeft straks niemand meer verstand te hebben van formele methoden en complexe algoritmen, daar komt hun beklag op neer. Sterker nog, op deze manier hebben we helemaal geen programmeurs meer in het Westen nodig, want zo simpel systemen produceren, ja! dat kunnen ze in India ook nog wel. Als alternatief bevelen ze daarom *Ada* aan: een nagenoeg onbegrijpelijke taal waarmee zelden een serieus programma is geschreven.

Programmeren wordt zo weer een ècht vak voor ingewijden met een wiskunde-knobbel.

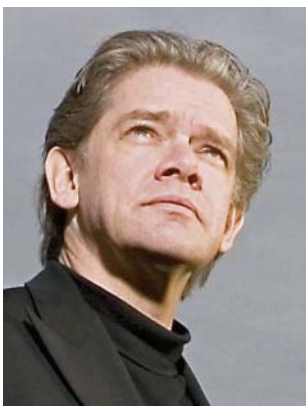
Wel geruststellend, de houding van zo'n universiteit. En toch weer die knappe prestatie om zo compromisloos afstand te nemen van de dagelijkse realiteit. Het wordt al een hele tijd gedaan, dus van ervaring kunnen we ondertussen wel spreken.

Ik heb het van horen zeggen, maar toen COBOL68 op de markt werd gebracht, werd geroepen dat het vak van programmeur geen toekomst meer had. COBOL leek ondertussen zoveel op natuurlijke taal dat *managers vanaf nu zelf konden programmeren*.

Vooruit, het zal de hippietijd zijn geweest en de wereld zag er roze uit. Maar wat is er in die 40 jaar eigenlijk gruwelijk weinig veranderd. Ook anno 2008 roepen velen dat we nu wel zo'n beetje klaar zijn met programmeren. De moeilijke dingen zitten steeds meer ingebakken in standaardpakketten. Voor de rest is het een kwestie van *mashuppen*, het aan elkaar lijmen van herbruikbare componenten, steeds krachtiger wordende *frameworks* en zelfs executeerbare modellen. Komt weinig coderen meer bij kijken.

En kunnen de managers ondertussen zelf programmeren?

Weinig kans. De IT kent een zelfreinigend vermogen dat er voor zorgt dat zo'n gruwel niet snel werkelijkheid wordt. Het is het vermogen dat er een tijdje geleden voor zorgde dat we 4GL's inruilden voor OO programmeertalen; hetzelfde vermogen dat ons een discussie doet voeren over of Java, Ada of C++ nou het beste is. Het is dezelfde Stille Kracht die sommigen doet terugverlangen naar *assembler* en die ons steeds weer wegvoert van eenvoudige oplossingen. Geen zorgen, dit vakgebied blijft met gemak moeilijk.



Ron Tolido
tolido.blogspot.com