

Hoe de kunst om 'mooie' Java-programma's te maken eigenlijk hetzelfde is als het waarderen van een goed glas wijn en het leren eten van oesters.

# Programmeren is een uiting van emotie

Ik heb regelmatig de eer om als instructeur op te treden bij een of andere cursus die tracht de deelnemers het programmeren in Java bij te brengen. Aan dit soort cursussen zitten zowel wetenschappelijke als kunstzinnige aspecten. Aan de ene kant krijgen de cursisten de eigenschappen en mogelijkheden onderwezen van Java plus de meegeleverde hulpmiddelen zoals class libraries en documentatietools. Deze kennis is noodzakelijk om ervoor te zorgen dat de cursisten ooit nog eens een zinvolle, en met name werkende, Java-programma's produceren. Programmeren bestaat echter niet alleen uit het achter elkaar zetten van statements totdat het programma werkt. Het behoort ook tot de taak van de programmeur om – en dat is de andere kant – een 'mooi' programma te maken. Een mooi programma is namelijk beter gestructureerd, en dus makkelijker te doorzien, en daardoor dus eenvoudiger te onderhouden. Niet zelden is een mooi programma ook sneller door esthetisch verantwoorde toepassing van objectmodellen, algoritmes en datastructuren. Dit alles valt onder de kunst van het programmeren.

Het behoort tot de taak van een docent om de cursisten de schoonheid van een taal over te dragen en te onderrichten in de esthetiek van

de oplossingen die met die taal kunnen worden bewerkstelligd. Wat mooi is en wat niet, is in hoge mate subjectief en van taal tot taal verschillend. Zelfs de basis van 'mooi-zijn' kan per taal verschillen. In Perl heeft 'mooi' bijvoorbeeld doorgaans te maken met de lengte van de code (vaak: hoe korter hoe mooier) en het creatief gebruik van zaken zoals reguliere expressies. In Java is 'mooi' meestal gekoppeld aan de implementatie van zeer abstracte objectmodellen en het gebruik van de dynamisch, zelfbeschrijvende eigenschappen zoals Reflection en class loaders.

Kunst heeft alles te maken met emotie en het is met name dit aspect dat voor cursisten als een volledige verrassing komt. "Programmeren is een uiting van emotie", mijmerde een collega-docent toen we dit soort aspecten aan het koffie-apparaat stonden te bespreken. Het maken van Java-kunst vereist een diep inzicht in de taal en de oplossingspatronen die de taal 'voorschrijft'. En zoals het waarderen van een mooi schilderij, een goede wijn en oesters tijd en ervaring vergt, zo is ook voor het waarderen en zelf produceren van (Java-) programmeerkunst ervaring nodig. En niet te vergeten: het gebruik van de juiste materialen en hulpmiddelen. Met name op dit laatste vlak ben ik somber gesteld. De Java-program-

meeromgevingen die de meeste programmeurs moeten of willen gebruiken werken het creëren van kunst helaas niet in de hand. Je kunt als docent natuurlijk niet verwachten dat de cursisten al na een of twee weken onderricht in Java in staat zijn om de schoonheid van de taal te doorvoelen, de kunstwerken (of -werkjes) van anderen volledig te waarderen, of zelf kleine parels te produceren. Dit moet de docent er echter niet van weerhouden om ook de kunstzinnige aspecten van Java aan de man te brengen. Als docent laat je jezelf natuurlijk wel eens gaan, en als ik dan probeer uit te leggen dat je met generieke adapter-classes die gebruik maken van Java Reflection een prachtige oplossing kunt bouwen om hetzelfde event van verschillende event-sources naar verschillende handler-functies te leiden, kijkt men je aan alsof je zojuist van een andere planeet bent gekomen. Soms zie je echter in een aantal ogen iets opblinken, en daar doe je het dan allemaal voor.

Jos Visser

is IT-estheticus bij Open Solution Providers en bereikbaar via [josv@osp.nl](mailto:josv@osp.nl)