

Erik Meijer was een aantal jaren geleden nog hoogleraar in Utrecht.

De laatste jaren werkt hij echter in Redmond. Na zich onder meer met Linq bezig gehouden te hebben, is zijn huidige taak het invullen van de inhoud van de volgende versies van VB. NET. Meijer blijkt daar geheel eigen ideeën over te hebben.

thema

De toekomst van Visual Basic

Programmeren voor de gewone man

VB was oorspronkelijk vooral een instapmogelijkheid voor ontwikkelaars die VB 6.0 deden, maar de functionaliteit van beide talen lijkt steeds meer op elkaar. Heeft het nog wel zin om met VB door te gaan?

Meijer: 'Tot een jaar geleden dacht ik zelf ook dat VB eigenlijk hetzelfde was als C#, maar dan met een iets andere syntax, totdat ik er echt goed naar keek. VB is uniek temidden van alle programmeertalen die ik heb gezien, doordat je er zowel dynamisch als statisch getypeerd in kunt werken. De scripting-talen die nu in Amerika heel erg in zijn, zoals Ruby en Python, zitten helemaal in de dynamisch getypeerde hoek. Bij Java en C# zit je helemaal in de statische typering. VB is de enige taal die in hetzelfde programma zowel dynamisch als statisch getypeerd kan zijn. Je kunt kiezen, als een slider die je op het een of het ander kunt zetten, je kunt echt het hele spectrum af. Bij VB kun je per functie en per module kiezen, dus je hebt veel keuzemogelijkheden.'

Is dat het bestaansrecht?

'Dat is wat ik eraan ontdekt heb, dat is het unieke. Er zijn voorstellen om aan Javascript ook optional static typing toe te voegen, Actionscript (voor Flash) heeft ook optional static typing, maar dat is het eigenlijk. VB had dat allemaal al. VB1 democratiseerde het programmeren voor Windows, daarvoor was het C++ heel ingewikkeld. Toen VB kwam, kon iedereen het, mijn schoonmoeder, iedereen. Nu is het programmeren voor het web te ingewikkeld, dat

wil ik ook democratiseren. In de volgende versie van VB wil ik bereiken dat iedereen, de gewone man, webservices kan maken en aan elkaar kan plakken. Webprogrammeren toegankelijk maken voor de massa.'

Is dat met ASP.NET al niet gemakkelijk?

'Nog niet gemakkelijk genoeg. Het web heeft een beetje te veel over-engineering. Laten we proberen

'Conceptueel is XML eenvoudig maar aan al die kleine lettertjes zitten heel veel haken en ogen'

het gewoon weer simpel te maken. Internet Messenger heeft 250 miljoen gebruikers, als je die mensen de mogelijkheid biedt om te programmeren, de boel aan elkaar te plakken. Dan praten we over de doelgroep die ik voor ogen heb, maar het moet voor hen dus aanzienlijk simpeler worden gemaakt.'

Moet het voor de business niet eenvoudiger worden, want dat is ook een oud ideaal?

'Wat mij betreft, er zijn al genoeg slimme mensen bij Microsoft die zich daarover ontfermen, maar ik

richt me op de gewone man. Iemand die een postzegelverzameling heeft en die iedere dag wil zien hoeveel zijn verzameling waard is.'

En het MKB?

'Ja, voor zover ze geen informaticakennis hebben.'

DREMPEL 'Misschien speelt mijn leeftijd ook een rol. Als ik mensen van mijn leeftijd vraag hoe ze heb-

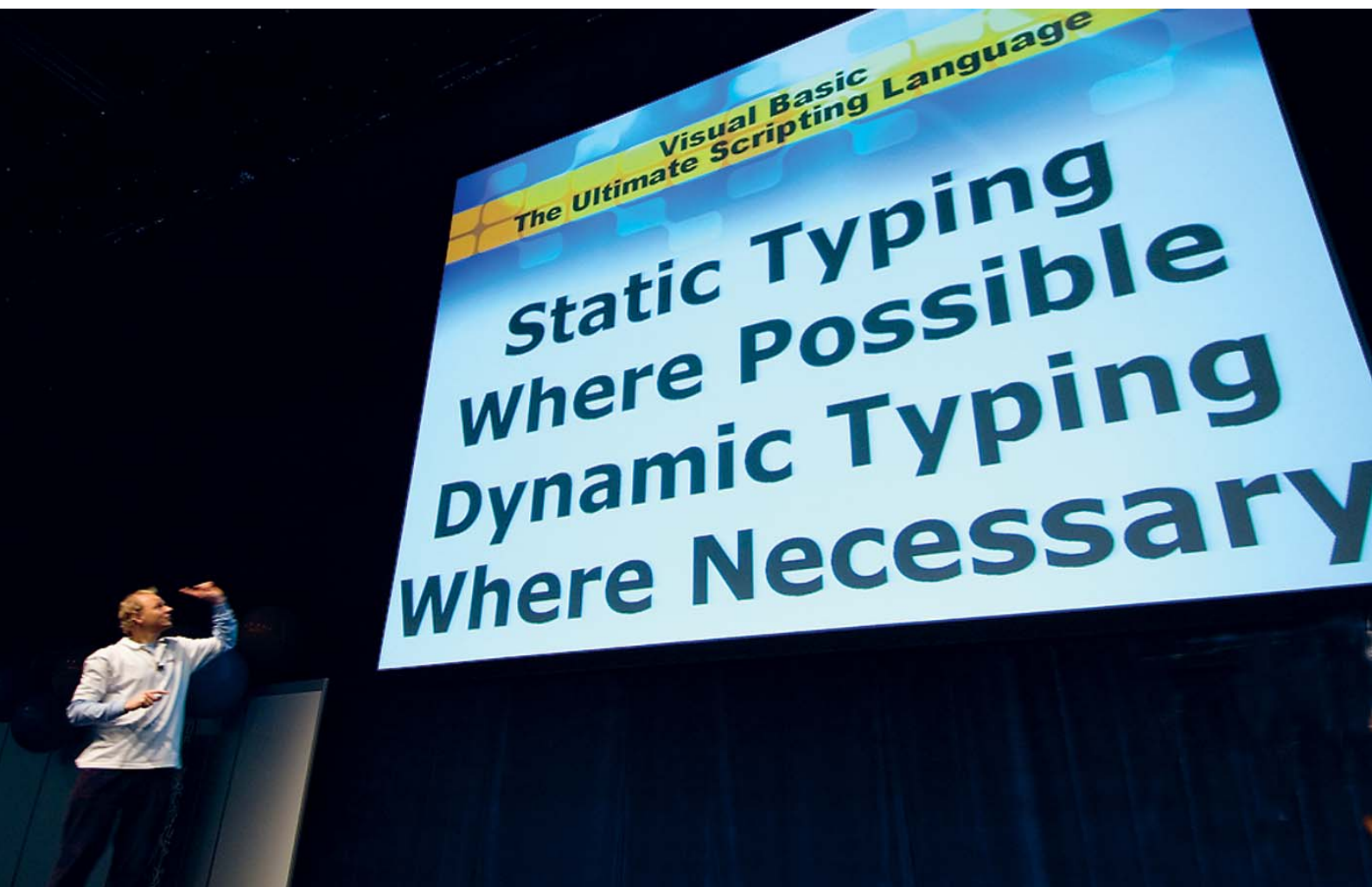
ben leren programmeren dan zeggen ze: op de Commodore 64 met Basic, op de Apple 2 met Basic, op de TRS 80 met Basic. Hoe zorg je er nu voor dat de jeugd op een leuke en gemakkelijke manier in aanraking komt met programmeren. Op dit moment spelen ze videospelletjes, maar voor het zelf in elkaar knutselen is eigenlijk niets meer. De drempel is heel hoog.'

Maar PHP is wel degelijk populair, ook bij niet-programmeurs.

'PHP is gebaseerd op de ideeën van ASP, de oudere wette ASP. Er zit een hoop goede ideeën in maar de taal zelf is eigenlijk niet echt goed. In PHP is alles dynamisch getypeerd, en als je bijvoorbeeld een variabele declareert met als waarde een string dan weet de compiler dat het een string is, dan kun je IntelliSense gebruiken en op het moment dat je die string als button wilt gebruiken een foutmelding eraan koppelen, en op het moment dat je alleen dynamische getypeerd hebt, krijg je al die fouten *at runtime*. Het is goed om zoveel mogelijk fouten tijdens het ontwikkelen van je programma af te vangen. Veel mensen denken dat als je geen types schrijft dat dit meteen

'Hoe zorg je er voor dat de jeugd op een leuke en gemakkelijke manier in aanraking komt met programmeren?'

ben leren programmeren dan zeggen ze: op de Commodore 64 met Basic, op de Apple 2 met Basic,



Erik Meijer: 'VB is uniek temidden van alle programmeertalen die ik heb gezien, doordat je er zowel dynamisch als statisch getypeerd in kunt werken'

betekent: dynamische typering, maar in de programmeertheorie weten we al dertig, veertig jaar dat de compiler die types kan afleiden, dus dat betekent dat je geen enkel type hoeft te schrijven en tóch statisch getypeerd bent. In zo'n taal kun je dat veel gemakkelijker maken en toch veel robuuster. In versie 9 zitten een aantal van die dingen en in de toekomst willen we nog veel meer van dat soort dingen toevoegen. Een ander voorbeeld wat we in VB 9.0 doen: XML. Veel mensen denken dat XML eenvoudig is, maar dat is helemaal niet zo. Conceptueel is het eenvoudig, maar aan al die kleine lettertjes zitten heel veel haken en ogen. Als implementator moet je daar rekening mee houden. In VB hebben we gezegd: hoe kunnen we XML nu zo eenvoudig mogelijk maken in het gebruik. Daarom hebben we van XML een eerste klasconcept in de taal gemaakt, net zoals strings, integers, booleans. Je kunt een stukje XML knippen en plakken in je VB-programma. XML is nu een datatype dat dezelfde rechten heeft als alle andere. In die XML kun je weer stukjes open laten als je het dynamisch wilt genereren. Op die manier maken we het werken met XML helemaal toegankelijk. Als je in Notepad een XML-documentje kunt schrijven, dan kun je het ook in VB. De VB-editor weet dat het XML is en kleurt de syntax enzovoorts.

ROLLBACK 'Iets anders is concurrency bij webtoepassingen. Als je kijkt naar multicore processoren, dan heeft iedereen straks een parallelle machine op zijn bureau met 4 tot wel 32 processoren. Hoe moeten we die programmeren? Hoe moet de gewone man dat doen? Wij zien wel wat in het concept van transacties. Dat is een concept dat iedereen kan begrijpen. Een stukje gebeurt óf niet en als het gebeurt, gebeurt het in isolatie. Wij willen transacties als universeel middel om concurrency te doen. Excepties zijn eigenlijk heel vervelend: ergens gaat iets mis, dan zoek je *at runtime* naar een *caller* die die exceptie kan afhandelen. Degene die het afhandelt zit alleen heel ver weg van waar de exceptie is gebeurd, je kunt niks meer doen. Je kunt alle veranderingen die je in je toestand hebt gedaan niet meer ongedaan maken, eigenlijk zijn excepties vrij nutteloos. Als je daarentegen met transacties werkt, kun je gebruik maken van een *roll back*, dat is een heel ander concept. Zeker als concurrency gemeengoed wordt, denken wij dat dit de manier is om het toegankelijk te maken. Bij aankopen op internet heb je natuurlijk ook dat soort situaties. Dat is ook weer iets dat in de databasewereld al dertig jaar bestaat, maar niet is doorgedrongen tot de gewone programmeerwereld.'

OLIFANTEN *Dat is allemaal voor VB, maar zou zo'n concept niet ook bij C# op z'n plaats zijn?*

'Misschien wel, maar op dit moment richt ik me echt op de massa. Een voorbeeld: de massa van alle bacteriën op de wereld is veel groter dan die van alle olifanten. Omdat je kleine dingetjes over het oog ziet richten mensen zich op die olifanten. Maar het is veel interessanter je te richten op de grote massa van

'Het is goed om zoveel mogelijk fouten tijdens het ontwikkelen van je programma af te vangen'

kleine dingetjes. Er zijn genoeg mensen die achter de olifanten aan gaan, maar ik richt me op de massa. Dat vind ik veel interessanter. We kunnen echt de wereld verbeteren met de sociale functie van het internet, en die bedrijven die redden het echt wel. Maar de scenario's zijn ook anders: in een bedrijf zijn dingen veel strakker. Als het nu *mission critical* is, dan kan het allemaal wat lossier, en dan maakt het niet zo uit of dat programma over twintig jaar nog onderhoudbaar is. Voor een bedrijf zijn hele andere dingen belangrijk. Mijn hoogleraar in Nijmegen, Kees Kusters, was altijd al bezig met programmeren voor beginners en kinderen. Dat heeft blijkbaar toch ergens een zaadje in mijn hersenen gestopt dat nu is ontkiemd.'