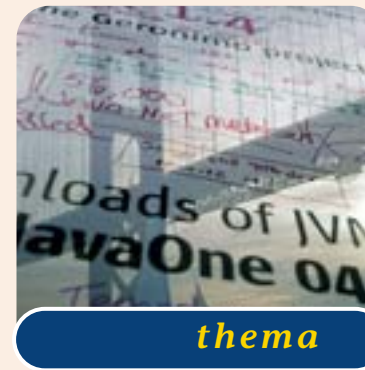


Tijdens de JavaOne is er veel aandacht besteed aan de updates in de Java 2 Standard Edition (J2SE). Deze volgende release met codenaam "Tiger" is nu nog in ontwikkeling. De release staat gepland voor 30 september 2004. Op dit moment is de bèta2 versie te downloaden van de [java.sun.com](http://java.sun.com) website. Wat staat er te gebeuren na "Tiger"?



thema

# De tijger wordt volwassen

## Veel veranderingen in Java 5

Tot nu toe kreeg de "Tiger" release het versienummer J2SE 1.5 mee. Sun heeft echter besloten om de J2SE en J2EE producten te hernummeren. Dat gebeurde al eerder met Solaris. De versie die na Solaris 2.6 kwam heette Solaris 7 en op dit moment komt Solaris 10 uit. Volledig overeenkomstig met deze nummer gaat Java 1.5 nu door het leven als Java 5. De motivatie achter deze omnummering is eenvoudig; Java is stabiel en volwassen en een versienummer 1.5 zou aangeven dat dit nog niet het geval is. De Java 5 versie heeft lang op zich laten wachten. Reeds twee jaar geleden op de JavaOne werden sessies gepresenteerd over de nieuwe functionaliteiten van "Tiger". De Java 5 release is de belangrijkste update van het Java platform sinds de verschijning van Java; de taal is uitgebreid met nieuwe mogelijkheden die het gebruik van Java moeten vereenvoudigen.

**JAVA 5 FEATURES** Het Java 5 platform brengt zeer veel nieuwe features met zich mee. De belangrijkste thema's uit deze release zijn "Quality and Stability", "Performance and Scalability", "Ease of Development" en "Desktop improvements". Java 5 is uitgerust met een JVM Monitoring en Management API (JSR 163 / JSR 174). Hierdoor kan tijdens executie allerlei informatie opgevraagd worden over de status van de virtuele machine, zoals de omvang van de heap, de status van de garbage collector en thread informatie. Daarnaast kunnen deze gegevens ook opgevraagd worden via SNMP, zodat bestaande management oplossingen gebruikt kunnen worden voor het bewaken van Java-gebaseerde applicaties. JMX is de Java Management API in "Tiger". Deze API is ook reeds aanwezig in eerder vrijgegeven J2EE applicatieservers. De JMX API's kunnen ook gebruikt worden voor het opvragen van gegevens via tools als "jps", "jstat", "jinfo" en "jstack". Daarnaast

geeft de visuele tool "JConsole" inzicht in wat er zich afspeelt in de JVM. Daarnaast kan de ontwikkelaar ook zelf MBeans gebruiken om eigen informatie prijs te geven via de JMX-API. De MBean uit codevoorbeeld 1 kan gebruikt worden via een standaard voorgedefinieerde factory vanuit de JMX architectuur.

Met Java 5 is de focus van Sun teruggekomen op de desktop. De startup time is sterk gereduceerd door het gebruik van JVM sharing en effectieve omgang met standaardbibliotheken. Daarnaast zijn de GUI look-and-feels aangepast aan Windows XP en Gnome GTK. Voor ontwikkelaars die een desktop-onafhankelijke look-and-feel willen gebruiken is nu de "Ocean" theme beschikbaar. Java 5 krijgt een uitgebreide XML-ondersteuning met technieken als een SAX 2.0 parser, DOM level 3, W3C Schema ondersteuning, XPath, JAXP 1.3 en een verbeterde XSLT engine. Ease of Development is eveneens een belangrijk thema. Sun probeert door verbeteringen in het platform het leven voor de Java-ontwikkelaar makkelijker te maken. Bekende keywords uit dit thema zijn generics, enums, for-loop verbeteringen,

```
public interface FredMBean {
    public int getCount();
}

public class Fred implements FredMBean {
    int count = 5;

    public int getCount() {
        return count;
    }
}
```

CODEVOORBEELD 1: MBean

```
import javax.management.*;
import java.lang.management.*;

public class Main {
    public static void main(String argv[]) throws
    Exception {
        MBeanServer mbs =
        ManagementFactory.getPlatformMBeanServer();
        ObjectName on = new ObjectName(":type=Fred");
        mbs.createMBean("Fred",on);
    }
}
```

#### CODEVOORBEELD 2: Publishing

varargs, annotations, autoboxing en static import. Deze onderwerpen zijn reeds eerder in dit magazine behandeld.

De Annotation Processing Tool (apt) is een universele tool die de annotations kan verwerken die in de Java sourcefiles gebruikt worden. Deze tool zoekt alle annotations op en probeert daarbij de bijbehorende "processor" op te zoeken. Deze processors kunnen via een simpele API geschreven worden. Op deze manier kunnen ontwikkelaars hun eigen metadata tags schrijven en daarvoor de bijbehorende code laten genereren. Deze ontwikkeling zorgt ervoor dat de ontwikkelaar niet verschillende tools hoeft te gebruiken om verschillende annotations te verwerken. Waar normaal "javac" gebruikt zou worden voor het compilatie proces, wordt nu "apt" gebruikt.

**TIGER'S CHILDREN** Na de release van Java 5 zal er hard gewerkt worden de volgende versies van Java. Deze dragen de codenamen "Mustang" (Java 6) en "Dolphin" (Java 7). Eén van de belangrijkste zaken in de verdere ontwikkeling van Java is dat de taal transparant moet zijn en een goede balans moet hebben tussen simplicité en kracht. De onderhoudbaarheid van de code is zeer belangrijk, zeker waar het gaat om applicaties van vele miljoenen regels code. Naast de Java-releases ontstaan er ook andere talen die gebruikt kunnen worden binnen

```
@Author(name="KT" date="29/6/2004") class
Foo{}
```

#### CODEVOORBEELD 3: annotation (JSR-175)

```
public @interface Author {
    String name();
    String date();
}
```

#### CODEVOORBEELD 4: attribute definitie

het Java-platform. Een voorbeeld hiervan is de "Groovy" language (JSR-241). Deze taal biedt mogelijkheden die niet in Java zitten met de eenvoudigheid van een scripttaal.

Op dit moment worden experimenten gestart voor de Mustang en Dolphin projecten; dit kunnen projecten zijn vanuit Sun of vanuit de Java community. Op het moment dat deze projecten succesvol blijken, worden ze opgenomen in het JCP. Na goedkeuring door de JCP worden deze technieken opgenomen in de releases. "Mustang" (Java 6) wordt verwacht in de eerste helft van 2006. "Dolphin" (Java 7) wordt verwacht in de tweede helft van 2007. Sun Microsystems streeft ernaar elke achttien maanden met een nieuwe release te komen. Een update-release "Dragonfly" (Java 5.1) wordt verwacht in de eerste helft van 2005. Om te kijken welke interesses er zijn voor de volgende versies heeft Sun een online enquête opgesteld. Deze kan ingevuld worden op <http://java.sun.com/j2se/survey/index.jsp>.

**DESKTOP EXPERIMENTEN** Als onderdeel van de voorbereidingen voor de volgende releases zijn er twee projecten geplaatst op java.net; Java Desktop Integration Components (JDIC) en Java Desktop Network Components (JDNC). JDIC moet de integratie tussen Java applicaties en de desktop verbeteren. JDNC zorgt voor een verbeterde weergave van databronnen in een desktop-omgeving. Op de desktop zal continu gewerkt worden aan look-and-feel. Tijdens een demonstratie werd getoond hoe een Java-applicatie eruit ziet op Windows Longhorn. Deze zal tegelijkertijd met de release van Microsoft Longhorn beschikbaar komen.

**NIEUWE ONDERDELEN** Naast de bestaande onderdelen zullen de toekomstige releases waarschijnlijk worden uitgebreid met webservices- en XML-ondersteuning. Hiervoor wordt momenteel een JAX-RPC 2.0 client ontwikkeld. Tevens zal JAXB 2.0 data binding beschikbaar komen. Deze techniek zorgt voor de vert-



Graham Hamilton, Jonathan Schwartz en James Gosling „releasen" Tiger

Foto: Sun Microsystems

## Hyperlinks

<http://java.sun.com/j2se/1.5.0/snapshots/>  
<http://java.sun.com/developer/technicalArticles/releases/j2se15langfeat/>

aling van XML-berichten in Java-classes en vice-versa door gebruik te maken van XML schema's. Tevens zal er een lightweight JAX-RPC worden opgenomen voor gebruik in non-J2EE omgevingen. Additionele technieken op XML gebied zoals encryptie en digital signatures zullen ook worden toegevoegd.

De JDK updates zullen bestaan uit een "Split verifier" (vereenvoudigt de runtime-verificatie en zorgt voor snelheidsverbetering). Tevens zal de compiler API uitgebreid worden, zodat externe tools beter met javac kunnen werken. De packaging wordt verbeterd met extensies op het JAR formaat. JDBC 4.0 zal worden geïntroduceerd met enkele updates. De nieuwe SQL standaard zal worden ondersteund. De verbeteringen in JDBC hebben eveneens te maken met "Ease of Development".



Foto: Sun Microsystems

Jonathan Schwartz kondigt de bèta2 release aan van Java 5 (voorheen J2SE 1.5)

Het maken van verbindingen met databases moet eenvoudiger worden en vertaling van datatypes moet transparanter gemaakt worden.

Alles bij elkaar betekent dit, dat er de komende jaren hard gewerkt wordt aan het Java platform. Op de Sun website zijn wekelijkse updates te downloaden van "Tiger" om zo op de hoogte te blijven van de laatste stand van zaken.

## Project Looking Glass - de desktop van de toekomst

Looking Glass is een mooie, moderne desktop in 3D. De huidige windowmanagers hebben een tweedimensionale weergave. De ontwikkelaars van "Project Looking Glass" hebben de Java 3D API's gebruikt voor een proof of concept van een zeer moderne desktop, die er zeer aantrekkelijk uit ziet. Deze desktop biedt een intuïtieve, driedimensionale omgeving waarin de gebruiker interactie kan hebben met de applicaties. De desktop voorziet in transparantie, rotatie, zoom, multiple desktop workspaces en miniatures. Project Looking Glass legt een fundament, waarop ontwikkelaars nieuwe toepassingen kunnen bouwen voor het visualiseren van informatie naar de gebruikers. Dit project zal ondersteund worden op het Solaris en Linux platform.

**JAVA 3D** Project Looking Glass is als open source project beschikbaar op java.net. Hier kan de laatste versie worden gedownload en zijn ook de instructies te vinden om de desktop te installeren. De huidige versie bestaat uit een 3D windowmanager platform met een 3D windowmanager en application development API. Met deze API kunnen nieuwe applicaties worden toegevoegd aan het Looking Glass platform. Een native application integration module maakt het mogelijk om ook X11 applicaties op te starten

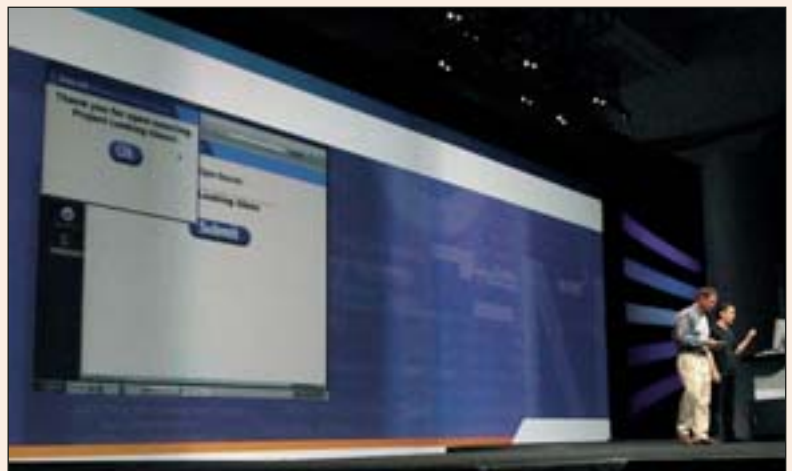


Foto: Dré de Man

Project Looking glass werd met behulp van een speciale applicatie aan de open source gemeenschap gedoneerd.

in deze 3D omgeving. Hierdoor kunnen dus ook niet-Java applicaties, zoals terminals en browsers gebruikt worden. In de SDK zit ook een 3D Environment Lite. Met deze standalone omgeving kunnen de ontwikkelde applicaties getest worden op elk Java 3D enabled platform.

Project Looking Glass maakt gebruik van de Java 3D API. Deze API is tegelijkertijd met Project Looking Glass



Op java.net is de laatste versie van Project Looking Glass beschikbaar gesteld voor download

gedoneerd aan de open source community op java.net. Het Java 3D platform maakt het mogelijk platformafhankelijke grafische 3D-applicaties te ontwikkelen. Java 3D werd eerder gebruikt door NASA voor het command- en controlsysteem voor de Mars Rover missie. Meer informatie over Java 3D is te vinden op <http://java3d.dev.java.net/>.

### Hyperlinks

[www.sun.com/software/project-looking-glass](http://www.sun.com/software/project-looking-glass)  
<http://lg3d.dev.java.net/>

## Groovy – The other language

Op dit moment wordt er binnen kringen van Java-ontwikkelaars veel gepraat over het doneren van Java aan de open source community. Daarnaast zijn meerdere open source talen ontwikkeld die kunnen integreren met het Java platform, zoals JRuby en Jython (Python voor Java). James Strachan en Bob McWhirter gingen een andere weg: zij ontwikkelden een nieuwe agile scripting taal, Groovy, die kan draaien op de bestaande Java Virtuele Machines. De ontwikkeling van deze open source taal is in het afgelopen jaar gestart. Groovy is eveneens aangemeld bij het JCP via JSR-241. Via deze weg moet Groovy tot een door het JCP erkende agile taal worden voor het Java platform. Binnen de expert-group is er reeds gestemd voor Groovy en het voorstel is met unanieme stemmen aangenomen. De omschrijving van JSR-241 is als volgt: *“This specification will standardize the Groovy programming language so that vendors can provide compliant implementations and developers will have a sanctioned scripting language they can use on the Java platform.”*

**INDRUK** In het volgende Java Magazine zal een volledig artikel over Groovy te vinden zijn. Ook tijdens de afgelopen J-Spring van de NL-JUG is er een presentatie geweest over Groovy. Om toch een indruk te geven van de taal volgen enkele voorbeelden. Groovy scripts (.groovy files) worden door de groovy compiler (groovyc) omgezet naar Java bytecode (.class files):  
prompt> groovyc <script-name>. Een voorbeeld van een Groovy statement wordt dan: [“bewolkt”, “zon-nig”].each { weer | println “Het weer vandaag is \${weer}” }

Groovy is een zeer compacte taal. Vergelijk de statements voor filehandling in Java en Groovy hiernaast.

```
import java.io.*;
public class ReadLines {
    public static void main (String [] args)
        throws FileNotFoundException, IOException {

        BufferedReader reader = new
        BufferedReader( new FileReader( „tst.txt“ ));
        String line;
        while ( (line=reader.readLine()) != null ) {
            System.out.println(line);
        }
    }
}
```

#### File handling in Java

```
file = new java.io.File(„tst.txt“)
file.eachLine { println(it) }
```

#### File handling in Groovy

Daarnaast is het in Groovy mogelijk om Swing-based user interfaces op te zetten:

```
import groovy.swing.*

builder = new SwingBuilder()
widget = builder.frame(title:'My Frame',
    size:[200,100]) {
    panel() {
        label(text:"Please enter something here:")
        tf=textField(text:"something")
        button(text:'OK', actionPerformed:{
            println(„Received Event = ${it}“)
            println(tf.text)
            System.exit(0)
        })
    }
}
```

Groovy is een krachtige taal voor het uitvoeren van "agile" werk. De taal is compact en dynamisch. Het feit dat het naar het Java platform compileert en gebruikt kan worden in combinatie met in Java ontwikkelde classes maakt het nog bruikbaar. Groovy is open source en wordt steeds verder uitgebouwd met nieuwe modules. Voorbeelden van dergelijke modules zijn de database en markup-builder modules. Onderstaande code haalt gegevens uit een database en zet deze om naar XML:

```
import groovy.sql.Sql
import groovy.sql.TestHelper
import groovy.xml.MarkupBuilder

sql = TestHelper.makeSql()

xml = new MarkupBuilder()

ignore = 'James'
sql.queryEach ("select * from person where
firstname != ${ignore}") {
    person | xml.customer(id:person.id,
type:'Customer',
                        name:person.firstname +
" " + person.lastname)
}
```

Deze code genereert de volgende XML:

```
<customers>
  <customer id="123" type="Customer"
foo="whatever">
  <role>partner</role>
  <name>James</name>
```

```
<location id="5" name="London"/>
</customer>
</customers>
```

**JONG EN COMPACT** De taal is echter ook nog vrij jong, waardoor er zeker nog enkele bugs in zitten. Tevens is de error afhandeling nog erg cryptisch en de taal is nog niet gefinaliseerd. Op het punt van documentatie moet er nog veel gebeuren. Op de Codehaus website zijn wel enkele presentaties en voorbeelden te vinden, waardoor de opstap makkelijker wordt gemaakt.

De presentatie van James Strachan op de JavaOne was trok het maximale bezoekersaantal. De interesse voor de taal neemt dus erg toe en binnen korte tijd kunnen we een nog grotere support uit de community verwachten voor Groovy. Dit wordt tevens nog aangemoedigd door de award die Groovy heeft gekregen tijdens de JavaOne. Tijdens de bijeenkomst van het Java Community Process is de award voor "Most innovative JSR" uitgereikt aan het Groovy team. Neem dus ook een kijkje en "get Groovy".

### Hyperlinks

<http://groovy.codehaus.org/>  
<http://www.nljug.org/>

## Brave new mobile world

Vrijwel iedere mobiele telefoon heeft tegenwoordig Java aan boord. Helaas hebben de operators, en mobiel-tjesmakers nogal wat chaos gecreëerd door een veel eigen, niet gestandaardiseerde functionaliteit aan de Java Omgeving toe te voegen. Sommigen hebben zelfs eigen Java mobiele standaarden ontworpen (Vodafone, NTT Docomo) Ook de toegang naar low level functionaliteit vanuit het Java-niveau op Symbian smartphones zoals de SE P900 of de Nokia serie 60 en anderen was tot nu toe onmogelijk. Pertti Korhonen, vice president mobiles van Nokia, liet op JavaOne in zijn keynote een vergezicht zien met betrekking tot de toekomstige mobiele Java wereld. In de toekomst wordt het mogelijk niet alleen monolithische Jar/Jad files te installeren, maar via een gedetailleerder component/middleware model kunnen componenten ook separaat geïnstalleerd worden. Zo kunnen C++ modules of bijvoorbeeld een Codec meegeladen worden om specifieke functionali-

teit mogelijk te maken. Ook de manageability van telefoons wordt sterk verbeterd, indien er een exception (fout) optreedt kan dit via het netwerk terug gemeld worden aan de service provider die dan min of meer automatisch een update kan uitvoeren. Zo zal software management ook voor grote groepen gebruikers mogelijk zijn en er kunnen 'over the air' updates uitgevoerd worden. Sun en Nokia werken samen in deze projecten. In 2005 en 2006 zullen we de resultaten van deze samenwerking zien. Het is echter duidelijk dat het mobiele Java-platform volwassen wordt. Standaardisering, managability en veel nieuwe functionaliteit komen ten dienste van de ontwikkelaar. Eindelijk 'a brave new mobile world' ?

*Klaasjan Tukker en Joost Backus*

*Klaasjan Tukker is voorzitter van de NL-JUG en Joost Backus is directeur van NotTheFly Component Solutions.*