



# Van de voorzitter



Het vierde kwartaal van 2006 kan de boeken in als een zeer gedenkwaardig kwartaal voor Java. Op 13 november 2006 annonceerde Sun Microsystems dat het Java-platform open source gemaakt is en dat de broncode onder GPL beschikbaar is. Meer dan 10 jaar werkt Sun reeds aan de ontwikkeling van het platform. Nu is deze ontwikkeling ook toegankelijk voor iedere Java-enthousiasteling in de wereld.

Het open source maken van Java is een zeer complex en tijdrovend proces. Daarom zullen de verschillende componenten afzonderlijk van elkaar beschikbaar komen. Een van de meest tijdrovende processen is het nalopen van de broncode van Java en het controleren van de eigendomsrechten. In sommige gevallen liggen niet alle rechten bij Sun Microsystems en moet dit op individuele basis opgelost worden. De eerste componenten die nu beschikbaar zijn in het OpenJDK project, zijn de Java Compiler (javac), de Hotspot-technologie en de JavaHelp software. Naar verwachting zal een volledige zelf te bouwen JDK beschikbaar zijn in het eerste kwartaal van 2007.

De licentie die gekozen is voor dit project, is 'GPL version 2 with Classpath exception'. Door keuze van de licentie is het zeer onwaarschijnlijk dat de code 'geforked' zal worden, waardoor er niet meerdere 'Java'-stromingen zullen ontstaan. De 'Classpath Exception' zorgt er tevens voor dat ontwikkelaar zonder problemen z'n applicaties kan bouwen op het Java-platform, zonder dat de broncode vrijgegeven hoeft te worden. Een vuistregel hiervoor: als je code in de JDK aanpast, dan moet je volgens de licentie deze code vrijgeven en beschikbaar stellen aan de community; als je je code baseert op de JDK,

dan hoef je deze niet vrij te geven.

Het opensourcen van Java betekent niet dat er nu allerlei extra features in het platform komen of dat er 'deprecated' functionaliteit uitgesloopt wordt. De specificaties waaraan het Java-platform moet voldoen (en wat er in komt) zal nog steeds worden geregeld via de JCP (Java Community Process). Het is wel mogelijk om het Java-platform te 'porteren' naar andere architecturen waarvoor nog geen ondersteuning aanwezig is. Dit kan een nog bredere adoptie van het platform tot gevolg hebben.

Java zal dus in de toekomst mede gedragen worden door de community. Vanuit de NL-JUG proberen we ook een sterke gemeenschap neer te zetten rondom Java. Degene die dit jaar een evenement van de NL-JUG heeft bezocht, zag dat de gemeenschap sterk groeit, zowel in aantal leden, maar ook in inhoudelijke zin; dit jaar hebben we meer leden gewonnen en meer evenementen georganiseerd dan ooit.

Vlak voor het sluiten van het jaar wil ik nog graag de volgende mensen bedanken: Het individuele NL-JUG lid; je bent lid van een platform, waarin je je kennis kunt verrijken, kunt communiceren met vakgenoten en je kennis kunt delen. De sprekers, trainers en instructeurs; zonder jullie was het ons niet gelukt evenementen

neer te zetten met het bereikte goede niveau. Namens alle leden hoop ik ook dat we volgend jaar wederom op jullie inhoudelijke bijdrage kunnen rekenen. Natuurlijk ook een woord van dank voor de partners van de NL-JUG. Zij zorgen voor de financiële continuïteit, zijn een ambassadeur waar mogelijk en zorgen tevens voor een inhoudelijke bijdrage.

Deze trend hopen we in 2007 door te zetten. Dit is echter een lastige opgave; de verwachtingen zijn hoog. Een community kan alleen leven als er ook inhoudelijk wordt bijgedragen aan de community. Dit kan een conflict geven met de zeer hoge druk die op dit moment op de Nederlandse Java-wereld ligt; er zijn niet voldoende Java-specialisten om het werk gedaan te krijgen. Toch is het belangrijk om ondanks deze druk te blijven investeren in kennis, kennisontwikkeling en kennisdeling. Vooral onze partners zou ik willen vragen goed te blijven letten op de manier waarop omgegaan wordt met de community. Een goede balans tussen het leveren van goede inhoud, profilering en financiële steun is en blijft zeer belangrijk.

Namens het bestuur en medewerkers van de NL-JUG wens ik jullie fijne feestdagen en een 'Java-rijk' 2007.

*Klaasjan Tukker*

# Masters of Java 2006

Op donderdag 23 november vond in Rotterdam de derde editie plaats van 'Masters of Java'. 24 teams streden om de befaamde titel 'Masters of Java 2006'. De opgaven dit jaar zijn weer samengesteld door onder andere Eric Hooijmeijer en Gerrit Bes en uitgetest en bijgeschaafd door Sebastiaan Wibier en Robert Bor.

De strijd was dit jaar zeer spannend. De opgaven waren over het algemeen goed te maken binnen de gestelde tijd. Voor elke goed ingeleverde opgave kreeg een team 400 punten, vermeerderd met het aantal seconden dat nog over was. Twee cases (River Crossing en Jindiana Jones) bleken wat moeilijker dan gemiddeld en werden niet door alle teams opgelost.

Op de derde plaats eindigde Team 'SOGYO 1' (Nanda Slabbers en Hugo Valk) met 6135 punten. Zij waren het enige team in de top 3 die alle cases succesvol hebben ingeleverd. Over het algemeen hadden ze echter net iets meer tijd nodig voor de opgaven, waardoor ze uiteindelijk op de derde plaats eindigden. Met slechts 59 punten verschil (6194 punten) is de tweede plaats in beslag genomen door Floris Ouwendijk. De winnaar van dit jaar met 6239 punten is team 'Haarlem Globetrotters' (Martin van Dijken en

Nico Tromp). Zij hebben 5 van de 6 cases succesvol opgelost en deden dat zo snel, dat ze voldoende punten hadden om hun voorsprong te behouden. De top 3 eindigde allemaal binnen anderhalve minuut van elkaar, wat een verschil betekent van ongeveer 15 seconden per case. Dit toont wel hoe spannend de eindstrijd was.

Een van de cases dit jaar was de 'Head on Collision': Na de zoveelste Command&Conquer kloon besluit EA eindelijk eens wat nieuws uit te brengen: ArkaPong - een combi van het klassieke Pong en Arkanoid. Geen geweldig idee, maar het is weer eens wat anders. Essentieel bij ArkaPong is een correcte botsingsdetectie tussen de ballen. Raad eens wie dat klusje mag klaren...

**IN BOTSING** Opdracht: Implementeer de routine 'collision' welke true moet terug geven wanneer de twee gegeven ballen in botsing zijn. 'In botsing' betekent dat of de randen van de bal elkaar raken of dat ze elkaar deels overlappen.

Dit probleem kan op meerdere manieren opgelost worden. De basis is dat als de afstand tussen de twee ballen kleiner of gelijk is dan de som van de stralen van de ballen er een botsing optreedt. Met behulp van de stelling van Pythagoras en de beide coördinaten kan de afstand tussen de ballen berekend worden. De Java-API

biedt echter een eenvoudiger oplossing: `java.awt.geom.Point2D.distance(x1, y1, x2, y2)`. Deze methode geeft de afstand tussen twee punten zonder dat je zelf Pythagoras hoeft te kennen (\*phew\*). De oplossing wordt dan:

```
public boolean collision(Ball
b1, Ball b2)
{
    // Som van de stralen
    double rr =
b1.getRadius() +
b2.getRadius();
    // Bereken de afstand
    tussen de ballen
    double r=Point2D.
distance(b1.getX(),b1.
getY(),b2.getX(),b2.getY());
    // het resultaat
    return r<=rr;
}
```

Deze oplossing wordt vervolgens ingevoerd in de competitie-software en door de centrale server getest. De opgave lijkt makkelijk, maar onder tijdsdruk kan hij soms toch moeilijker opgevat worden.

Wil je zelf ook kennismaken met 'Masters of Java'? Tijdens de Sun Kerstborrel op 21 december 2006 in Amersfoort zal de NL-JUG hier de software demonstreren. Je hebt dan de gelegenheid zelf kleine eenvoudige opgaven te maken.



Eén van de winnaars legt z'n gekozen oplossing uit.