

JAOO – the Report

In September 2003 werd de JAOO Conferentie gehouden. Deze conferentie was gevuld met sessies rondom Java Technology, object orientatie en software ontwikkeling. Klaasjan Tukker bezocht deze conferentie namens SDGN een geeft de highlights weer.

De conferentie was goed gevuld met sessies van bekende sprekers op veel gebieden van het software ontwikkelproces.

Bekende sprekers zoals Erich Gamma (van de Gang of Four!), Pragmatic Programmers Andy Hunt en Dave Thomas, XP Guru's als Kent Beck, Kevlin Henney en Nicolai Josuttis en natuurlijk Java "kopstukken" als Joshua Bloch van de collection API en Neal Gafter, "mister javac". Alles bij elkaar een interessante line-up en goede ingrediënten voor een stevige conferentie.



minder snel "nee". Jongere ontwikkelaars zijn meestal interessanter voor werkgevers.

2. Afhankelijkheid van een bepaalde sector; uitspraken als ik ben een "XYZ" programmeur of misschien wel ik ben een "Java" programmeur. Daarmee ben je afhankelijk van een bedrijf en bedrijven houden wel eens op te bestaan. Denk bijvoorbeeld maar eens aan namen als "Digital Equipment", "Tandem", "Honeywell" en misschien "Sun".

3. Het werk verdwijnt naar andere landen; hiervoor kunnen verschillende redenen zijn, waarvan de financiële reden er een is. Het belangrijkste dat hier geldt: "Goedkope ontwikkelaars kunnen goede code schrijven". Veel ontwikkeling wordt op grote schaal uitbesteed aan "lage lonen landen" in de ICT sector.

Alles bij elkaar een interessante line-up en goede ingrediënten voor een stevige conferentie

De volgende stukken zijn korte beschrijvingen en samenvattingen van verschillende sessies. De slides van de sessies zijn te vinden op de JAOO website, www.jaoo.dk2.

How to keep your job – Andy Hunt / Dave Thomas

Een pakkende titel en zeker in de huidige economische omstandigheden een sessie om bij te wonen. Andy en Dave hadden een goed opgezet verhaal met veel redenen waarom je jezelf als software ontwikkelaar zou moeten ontwikkelen. Het uitgangspunt: "Onze baan staat op de tocht". Er zijn drie grote bedreigingen:

1. De dingen gaan te snel. De halveringstijd van ontwikkelkennis, de tijd die nodig is voordat het overbodig is, varieert van 7 tot 2 jaar. Een grotere bedreiging is de vergrijzende ontwikkelaar; jongere ontwikkelaars zijn goedkoper, flexibeler, minder gebonden en zeggen

Voor deze bedreigingen moet er natuurlijk een oplossing komen. De oplossingen kunnen vanuit verschillende hoeken komen. Er kunnen b.v. maatregelen genomen worden vanuit de overheid. Een bekende maatregel is het hantieren van het "protectionisme", maar in het verleden is al gebleken dat dit niet werkt. Een andere mogelijkheid is te vertrouwen op het bedrijf waarvoor je werkt.

Waarschijnlijk zal dit ook niet werken, want het bedrijf en jij hebben tegenstrijdige belangen. De laatste optie is om zelf actie te ondernemen: "investeer in jezelf!" Daarbij kunnen we heel veel leren van de financiële wereld. Hier geldt: goede investeerders hebben een plan, spreiden hun belang, zoeken de waarde, zijn actief en niet passief, doen het frequent. Stel voor jezelf een doel, waar je in de komende jaren heen wilt en maak een planning hoe je daar wilt komen. Richt je niet op een bepaalde techniek, maar probeer jezelf meerdere technieken meester te maken; het leren van een extra taal levert altijd iets op. Probeer te investeren in zaken die waarde hebben, het moet goed toepasbaar zijn en niet te theoretisch van aard. Stel je actief op; evalueer regelmatig je eigen voortgang en kijk of je oorspronkelijke plan nog staat. En na-

tuurlijk: besteed er regelmatig aandacht aan, want af en toe er mee bezig zijn zet geen zoden aan de dijk. Samenvattend, wil je voor jezelf je toekomst zeker stellen, dan moet je er zelf aan bouwen en niet altijd te veel vertrouwen op je huidige omgeving.

Object Relational Mapping – Gavin King

Gavin King is de lead-developer van Hibernate (www.hibernate.org). Deze software helpt de ontwikkelaar het gat tussen de relationele wereld en de objectgeoriënteerde wereld te dichten. Veel ontwikkelaars die bezig zijn met Java, vinden het lastig zich te moeten bewegen tussen de Java en de SQL-taal. Onderstaand codefragment is afkomstig uit de “Hibernate and Tomcat QuickStart”

```
SessionFactory sessionFactory =
    new Configuration().configure().buildSessionFactory();
session = sessionFactory.openSession();
transaction = session.beginTransaction();
Cat princess = new Cat();
princess.setName("Princess");
princess.setSex('F');
princess.setWeight(7.4f);

session.save(princess);
transaction.commit();
session.close();
```

Fig. 1: Sample Hibernate code

Hibernate zorgt door middel van een XML-mapping dat de data in de relationele tabel wordt vertaald naar de objectstructuur in Java. Daarnaast heeft Hibernate de beschikking over een eigen object georiënteerde querytaal, waarmee queries samengesteld kunnen worden. Voor ontwikkelaars die veel met databases werken en het schrijven van een eigen persistence laag zat zijn, is Hibernate een welkome afwisseling.

Forthcoming Java Programming Language Features – Joshua Bloch / Neal Gafter

Deze twee heren zijn bekende kopstukken achter de Java ontwikkelomgeving. Joshua is verantwoordelijk voor de complete Collection-API binnen de Java2 Standard Edition en Neal is het brein achter “javac”, de Java Compiler. In een sessie van 45 minuten worden de nieuwe features in de “Tiger” release (Java 1.5) belicht.

In een sessie van 45 minuten worden de nieuwe features die in de “Tiger” release (Java 1.5) belicht

In het kort de kreten die deze sessie bepaalden, elders in dit magazine worden enkele nieuwe features uitgebreid beschreven.

- **Generics** – Het toevoegen van compile-time checks op collections

- **Verbeteringen voor Loops**
Het versimpelen van “loop” constructs
- **Autoboxing / unboxing** – Java primitive types eenvoudig combineren met collecties
- **Typesafe enumerations**
Leesbare code voor vaste variabelen
- **Varargs** – Voor het schrijven van methods met een wisselend aantal parameters, te vergelijken met de “printf” methode in C
- **Static imports**
Importeer statische members uit een klasse
- **Annotations** – Metadatering voor het beschrijven van klassen en methoden

Wildcards in Java – Mads Torgersen

Een onderdeel van de nieuwe programming features in de 1.5 release van Java is het “Generics” principe. Op deze manier worden compile-time checks ingebouwd voor collections. Een voorbeeld hiervan is hieronder weergegeven:

```
interface List extends Collection {
    void add(Object element);
    Object get(int index);
    ...
}

List numbers = new ArrayList();
numbers.add(new Integer(7));
Number n = (Number)numbers.get(0);
Fig. 2: Collection oude-stijl met type-casting

interface List<T> extends Collection<T> {
    void add(T element);
    T get(int index);
    ...
}

List<Number> numbers = new ArrayList<Number>();
numbers.add(new Integer(7));
Number n = numbers.get(0);
```

Fig. 2: Collection oude-stijl met type-casting

Fig. 3: Collection met geparаметriseerd type <T>, type-casting is niet nodig

Op het moment dat dit mechanisme gebruikt gaat worden in een Collections en Inheritance omgeving treden er problemen op. Het Java Type “Object” is het supertype van alle java objecten. Daarmee kunnen verschillende Lists gecombineerd worden, waarbij ze alle naar het abstracte type “Object” worden vertaald.

Om dit te verbeteren is de wildcard ingevoerd. Het wildcard teken in Java is “?”. Op het moment dat twee listen van verschillende typen worden gecombineerd, wordt de wildcard gebruikt. In het onderstaande voorbeeld wordt een list gedefinieerd, waar objecten in kunnen zitten die allemaal afgeleid zijn van het objecttype “Number”. In het onderstaande code voorbeeld kunnen wel de objecten aan hun abstracte type eruit gehaald worden, maar hier mogen geen objecten in geplaatst worden.

```
List<Integer> ints = ...;
List<? extends Number> numbers = ints;
Number n = numbers.get(0);
```

Fig. 4: Een lijst met abstracte types dmv de Wildcard

Door gebruik te maken van superbounded wildcards kunnen Lists gemaakt worden met elementen die als supertype het abstracte type hebben. In deze situatie kunnen

wel objecten in de lijst gestopt worden, maar mogen geen objecten uit de lijst worden gehaald van het abstracte type.

```
List<Number> numbers = ...;  
List<? super Integer> ints = numbers;  
ints.add(7); //autoboxed add
```

Fig. 5: Een lijst met abstract type *dmv Superbounded Wildcards*

De functionaliteit van Wildcards is absoluut noodzakelijk voor het goed functioneren van Generics binnen de collection API van Java. De getoonde onderdelen zijn nog in een experimentele fase, maar worden zeker verwacht in de eerste Beta van "Tiger" release.

Eclipse@SAP – Erich Gamma / Ivo Totev

Eclipse is inmiddels een zeer bekende ontwikkelomgeving voor Java ontwikkelaars. De omgeving is echter opgezet als een modulair applicatie framework, waarin developers hun eigen functionaliteiten kunnen ontwikkelen door middel van Plugins. Bij het ontwikkelen van deze plugins kunnen ze op hun beurt weer een beschrijving maken op welke manier de desbetreffende plugin uitgebreid kan worden. De Java ontwikkelomgeving bestaat in totaal uit 13 plugins (Java Development Tools) die boven op het Eclipse Framework worden geplaatst. SAP heeft deze functionaliteit gebruikt om boven op het Eclipse Framework hun eigen complete ontwikkelomgeving te maken, NetWeaver genaamd.

Eric Gamma is project-lead voor het Eclipse project. Dit project wil uitgroeien tot het belangrijkste platform voor het bouwen van IDE applicaties en development tools. Wil je zelf ook eens kennismaken met dit platform, neem dan een kijkje op www.eclipse.org.

Social events

Natuurlijk was er tijdens de JA00 voldoende tijd om een gesprek aan te gaan met een van de vele sprekers. Dit kon tijdens de lunch, maar ook waren er verschillende activiteiten gedurende de avonden. Een van deze activiteiten was de JA00 IT-Run; een hardlooptwedstrijd over 6.6 kilometer door het centrum van Arhus. Ook waren er de nodige diners op maandag- en dinsdag-avond waar nader van gedachten gewisseld kon worden met collega ontwikkelaars.

Volgend jaar vindt er van 20 tot 24 september 2004 weer een JA00 plaats in Arhus. Vroegtijdig zal hiervan een aankondiging gedaan worden op de SDGN site. Dit jaar was de conferentie goed gevuld, zo goed, dat zich dat niet laat beschrijven in een artikel. Wil je meer weten, dan kun je gerust nog even kijken op de JA00 website.

(Footnotes)

- ¹ Gang of Four, de vier schrijvers van het bekende boek "Design Patterns".
- ² www.jaoo.dk inloggegevens voor de slides JA00slides2003 / Java00