

Groovy für Fortgeschrittene

Dierk König

dierk.koenig@canoo.com

Herzlich Willkommen!



Dierk König

- Canoo Engineering AG, Basel (CH)
 - Rich Internet Applications
Produkte, Projekte, Beratung, Schulung
www.canoo.com
- Trainer, Berater, Softwareentwickler
- Groovy- und Grails-Committer
- Autor: Groovy in Action

Fortgeschrittenes Groovy am Beispiel von 7 Patterns

- Alleskleber
- Weiches Herz
- Endoskopische Operation
- Kluge Anpassung
- Grenzenlose Offenheit
- Heintzelmannchen
- Prototyp



canoo

#1 Alleskleber



- Applikationen aus bestehenden Komponenten zusammenbauen
- Java ist gut geeignet für stabile Infrastruktur: Middleware, Frameworks, Widget Sets, Services
- Scripting ist gut geeignet flexible (agile) Applikationslayer: View und Controller
- Grails, GSP, JBoss Seam, WebWork, Struts 2 Actions,...
- Beispiel: Zusammenzug von XML Parser, Java Networking und Swing Widget Bibliothek, um einen Standard-RSS Feed darzustellen

canoo

Alleskleber Beispiel: RSS Reader

```
def base = 'http://news.bbc.co.uk/rss/newsonline_uk_edition/'
def url  = base + 'front_page/rss091.xml'
def items = new XmlParser().parse(url).channel[0].item

def swing = new groovy.swing.SwingBuilder()
def frame = swing.frame(title: 'Top 10 BBC News') {
  scrollPane {
    table() {
      tableModel(list: items[0..9]) {
        closureColumn(header: 'title',
          read: { item -> item.title.text() } )
        closureColumn(header: 'text',
          read: { item -> item.description.text() } )
      } } } }
frame.pack(); frame.show()
```

canoo

#2 Weiches Herz



- Fachliche Modelle auslagern
- Vorgegebenes Applikationsgerüst in Java
- Fachlichen Erkenntnisfortschritt ermöglichen:
Entitäten, Beziehungen und Verhalten durch Scripting
flexibel halten
- Anwender: Rule Engines (JBoss Rules,
groovyrules.dev.java.net, JSR-94), Grails,
Versicherungswesen: Mutual of Omaha
- Beispiel: Berechnungsregeln für Boni

canoo

Weiches Herz Beispiel: Bonusberechnung

```
umsatz = mitarbeiter.umsatz
switch(umsatz / 1000) {
  case 0..100 : return umsatz * 0.04
  case 100..200 : return umsatz * 0.05
  case {it > 200} : bonusClub.add(mitarbeiter)
                  return umsatz * 0.06
}
```



```
Binding binding = new Binding();
binding.setVariable("mitarbeiter", mitarbeiter);
binding.setVariable("bonusClub", bonusClub);
GroovyShell shell = new GroovyShell(binding);
File script = new File(filename);
float bonus = (float) shell.evaluate(script);
```



canoo

#3 Endoskopische Operation

- Minimal-invasive Eingriffe "in vivo"
- Viele Notwendigkeiten für Anpassungen ad-hoc Anfragen sind nicht vorhersehbar
- Schlüsselloch für die Live Ausführung von Scripts
- Unglaublich wertvoll für Produkt-Support, Fehleranalyse, Hot-Fixes, Notfälle
- Anwender: Oracle JMX Beans, XWiki, SnipSnap, Ant, Canoo WebTest, Grails Console, ULC Admin Console
- Beispiel: ein Live-Groovy Servlet



canoo

Endoskopische Operation: im Servlet

Probleme mit der Datenbank Verbindung?

```
def ds = Config.dataSource
ds.connection = new DebugConnection(ds.connection)
```

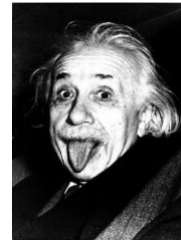
Gefährliche Benutzer rauswerfen

```
users = servletContext.getAttribute('users')
bad = users.findAll { user -> user.cart.items.any { it.price < 0 } }
servletContext.setAttribute('users', users - bad)
```

canoo

#4 Kluge Anpassung

- Konfigurationen mit Ausführungs-Logik, aka Smart Configuration
- Ersatz für XML-Konfigurationen
- Mit Referenzen, Schleifen, Bedingungen, Vererbung, Ausführungslogik, Umgebungsermittlung, ...
- Typischer Anwendungsfall für domänen-spezifische Sprachen (DSLs), Groovy Builder, Grails plugins, Benutzer-Macros, Function Plotter
- Beispiel: Navis SPARCS N4



canoo

Smart Config Beispiel: Container Routing

```
def ensureEvent = { change ->
  if (! event.getMostRecentEvent(change) {
    event.postNewEvent(change)
  }
}

switch (event.REROUTE_CTR) {
  case 'OutboundCarrierId' :
    ensureEvent('CHANGE_VSL')
    break
  case 'POD' :
    if (! event.CHANGE_VSL) ensureEvent('CHANGE_POD')
    break
}
```



canoo

#5 Grenzenlose Offenheit

- Jede Zeile Code wird änderbar
- Manchmal sind die vorgesehenen Variationspunkte nicht ausreichend
- Einfache Änderungen ohne langwierigen Setup für Kompilation und Deployment
- Perl, PHP, Python, etc. machen es vor
- Beispiel: groovyblogs.org (Grails Applikation)



canoo

#6 Heinzelmännchen



- Repetitive Aufgaben automatisieren
- Automatisierter Build, kontinuierliche Integration, Deployment, Installationen, Server-Überwachung, Reports, Statistiken, Erzeugen von Dokumentation, funktionale Tests, HTML Scraping, Web Fernbedienung, XML-RPC, WebServices
- Anwendungen mit Ant, Maven, AntBuilder, Gant, Canoo WebTest, Grails scaffolding, ...
- Beispiele: dynamisch Mails per Ant verschicken, Skripte mit Ant verbinden

canoo

Heinzelmännchen Beispiel: Mail schicken

```
def users = [ [name:'Dierk', email:'dierk.koenig@canoo.com'],
              [name:'Other', email:'other@no.such.server' ]

def ant = new AntBuilder()
for (user in users) {
    ant.mail(mailhost: 'my.email.server', subject: 'build ist fertig') {
        from(address: 'my@email.com')
        to (address: user.email)
        message( """
            Hallo ${user.name},
            Der Build ist mal wieder fertig:
            ${new Date().toGMTString()} """ )
    }
}
```

canoo

Heinzelmännchen II: Muisches Ant

```
<groovy>
import org.apache.tools.ant.*
import org.jfugue.*
project.addBuildListener(new PlayListener())
class PlayListener implements BuildListener {
    def play = { new Player().play(new Pattern(it)) }
    void buildStarted(event) { }
    void buildFinished(event) { }
    void messageLogged(event) { }
    void targetStarted(event) { play("D E") }
    void targetFinished(event) { play("C5maj") }
    void taskStarted(event) { }
    void taskFinished(event) { }
}
</groovy>
```

canoo

#7 Prototyp



- Machbarkeitsstudien auf der Zielplattform
- "Spikes" für technologische oder algorithmische Ideen mit besserer Ausdrucksmächtigkeit, schnellerem Feedback und besseren Analysemöglichkeiten
- Wahlmöglichkeit für spätere (Teil-)Portierung nach Java
- Anwendungen: Benutzerfeedback über Domänenmodell mit funktionalem Grails Prototyp abfragen, Java Bildbearbeitungs-Algorithmen
- Beispiel: Primzahlzerlegung

canoo

Prototyp Beispiel: Primzahlzerlegung

```
boolean isPrime(x) { return ! (2..<x).any { y -> x % y == 0 } }

int primeBelow(x) { (x..1).find { isPrime(it) } }

List primeFactors(x) {
  if (isPrime(x)) return [x]
  int p = primeBelow(x)
  while (p > 1) {
    if (x % p == 0) return [p, *primeFactors(x.intdiv(p))]
    p = primeBelow(p-1)
  }
}

for (n in 100..110) { println "$n : "+primeFactors(n)}
```

canoo

Primzahlzerlegung: Modulo ops zählen

```
class ModCountCategory {
  static int count = 0
  static int mod(Integer self, Integer argument) {
    count++
    return self - argument * self.intdiv(argument)
  } }

use (ModCountCategory) {
  for (n in 1000..1010) {
    ModCountCategory.count = 0
    factors = primeFactors(n)
    println "$n : $factors".padRight(30) + "(in " +
      "${ModCountCategory.count}".padLeft(5) + " steps)"
    assert n == factors.inject(1){result, item -> result * item }
  } }
}
```

canoo

Pattern Zusammenfassung

- Alleskleber
- Weiches Herz
- Endoskopische Operation
- Kluge Anpassung
- Grenzenlose Offenheit
- Heizelmännchen
- Prototyp



Keep groovin' !

canoo

Weitere Informationen

- groovy.codehaus.org, grails.org
- aboutgroovy.com, www.svenhaiges.de, groovyblogs.org, searchgroovy.org
- **Groovy in Action** groovy.canoo.com/gina
Manning, 2007, Vorwort von James Gosling
König mit Glover, Laforge, King, Skeet
- + **Groovy Recipes**



canoo

Offene Fragen



cano