

Orientation in Objects

Groovy und Grails

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Gliederung

Orientation in Objects

- Groovy
- Grails
- Live Session
- Wenn noch Zeit ist
- Quellen

Gliederung

Orientation in Objects

- **Groovy**
- Grails
- Live Session
- Wenn noch Zeit ist
- Quellen

Groovy Buzz Words

Orientation in Objects

new agile **dynamic** language JVM

Java-like syntax **quicker** more concise **fun way**

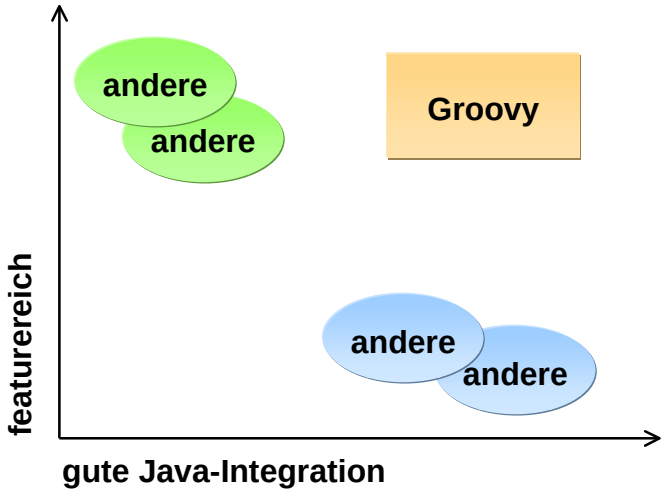
expressive syntax **feature-rich** like Python, Ruby and Smalltalk

Closures GDK DB XML Groovlets **Builder**

Prototyping **Scripting** Templating **Unit Testing** DSL

Was macht Groovy besonders?

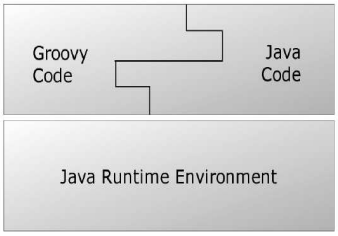
Orientation in Objects



aus Dierk König, Groovy in Action (Manning)

Groovy und Java

Orientation in Objects



aus Dierk König, Groovy in Action (Manning)

- Groovy läuft in der Java VM
- Groovy nutzt die Bibliotheken des JDK
- Groovy erweitert das JDK (GDK)
- Groovy kann Java-Klassen aufrufen
- Java kann Groovy-Klassen aufrufen
- Groovy kann nach Java Classfiles compiliert werden
- Leichte, risikolose Migration von Java nach Groovy

Historie

Orientation in Objects

- 2003:
  - Start durch James Strachan und Bob McWriter
  - Grundidee: Eleganz von Ruby in Java
- 2004:
  - Beginn der Standardisierung mit dem JSR-241
  - GroovyOne (Treffen von Groovy Entwicklern in London)
- Ende 2006: Release 1.0
  - Projektleitung: Guillaume Laforge (Groovy), Graeme Rocher (Grails)
- 2007:
  - 1. Platz JAX Innovation Award
  - G2One gegründet (Firma hinter Groovy und Grails)

Groovy - 1x1

Orientation in Objects

=

ausdrucksstarke Syntax

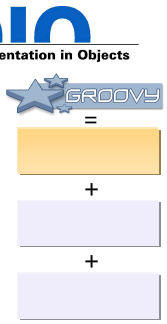
+

mächtige Bibliotheken

+

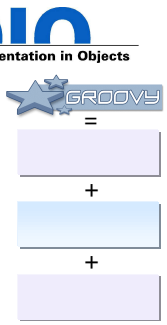
Meta-Programmierung

Ausdrucksstarke Syntax



- kurz und leserlich
- keine primitiven Datentypen -> alles ist ein Objekt
- literale Unterstützung von Collections, regulären Ausdrücken
- Closures, Überladen von Operatoren, GPath, Builder

Mächtige Bibliotheken

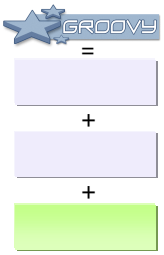


- verwendet Javas Klassenbibliothek
- GDK
  - JDK-Erweiterungen
  - Wrapper-APIs
    - **JDBC**
    - **XML**
    - **GString, Template**
    - **Swing**
    - **SOAP, Webservices, REST**
    - **COM-Scripting**

Meta-Programmierung

Orientation in Objects

- Java:
  - Felder und Methoden zur Compile-Zeit bekannt
- Groovy:
  - dynamisches Anhängen von Feldern und Methoden zur Laufzeit
  - Meta Object Protocoll - MOP
  - Builder Patterns in Groovy



Hello World in Groovy

Orientation in Objects

```
println 'Hello Groovy-World!'
```

Nochmal Hello World in Groovy

Orientation in Objects

```
public class SayHello {
   public static void main(String[] args) {
       System.out.println("Hello Java-World!");
   }
}
```


Alles ist ein Objekt

Orientation in Objects

- keine primitiven Datentypen (alles Integer-Objekte)
- Gleichheit mit '==' (entspricht equals() in Java)
- Identität mit 'is()' (entspricht '==' in Java)

```
int i = 55
assert i == 55
assert 24 == new Integer(24)
assert i.is(i)
```

Typsystem

Orientation in Objects

- statische Typisierung optional
- dynamische Typen (Überprüfung zur Laufzeit)

```
String s = "Test"

def i = 1
```

- Duck Typing:

```
class Auto {
   def fahren() {println 'Auto fährt'}
}
class BobbyCar {
   def fahren() {println 'BobbyCar fährt'}
}

def auto = new Auto()
def bobbyCar = new BobbyCar()

[auto, bobbyCar].each {it.fahren}
```

Zeichenketten

Orientation in Objects

3 Arten:

```
def s1 = 'String'
def s2 = "GString"
def s3 = /\w[2]/
```


```
// in Java
System.out.println("Hallo " + user.getName() + "!");

// in Groovy
println "Hallo ${user.name}!"
```

Collections

Orientation in Objects

- Listen und Maps sind literale Sprachkonstrukte in Groovy

```
def list = [1, 2, 3]
def map = [1:'a', 2:'b']
```

- Ranges: Wertebereiche

```
1..10 // Intervall von 1 bis 10
'A'..<'F' // Intervall von A bis E

// Schleife von 1 bis 3
(1..3).each {println i}

def substr = 'Kleiner Test'[0..6] // --> Kleiner
```


Operatorüberladung

Orientation in Objects

Operator	Methode
a + b	a.plus(b)
a | b	a.or(b)
a[b]	a.getAt(b)
a[b] = c	a.putAt(b, c)
a << b	a.leftShift(b)
[...]	

```
i = 1 + 1
list = [1,2,3] + 4 // --> [1,2,3,4]
list = [1,2,3] * 2 // --> [1,2,3,1,2,3]
tomorrow = new Date() + 1
```

Groovy Beans

```
public class Person {

   private String firstName;
   private String lastName;

   public String getFirstName() {
        return firstName;
   }

   public void setFirstName(String firstName){
        this.firstName = firstName;
   }

   public String getLastName() {
        return lastName;
   }

   public void setLastName(String la
   this.lastName = lastName;
   }
}
```

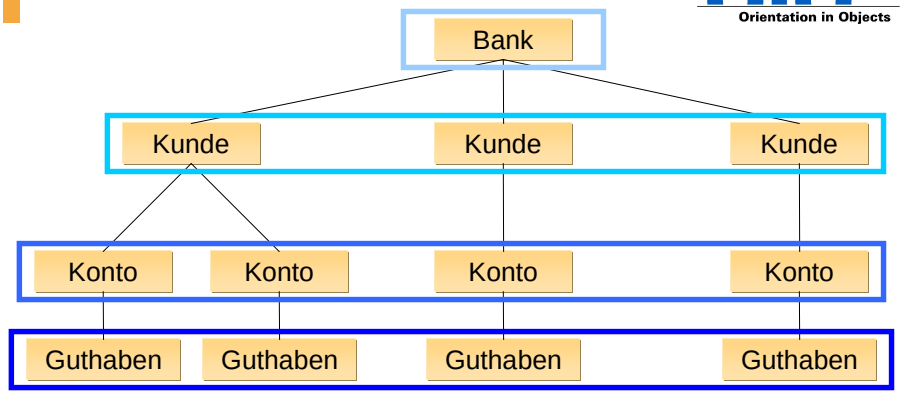
```
class Person {
   String firstName;
   String lastName;
}
```

```
p = new Person()

p.setFirstName("Dieter")
p.firstName = "Otto"
p['firstName'] = "Manfred"

println p.getFirstName()
println "Hallo $p.firstName"
```


GPath - Effiziente Objektnavigation



- Summe der Guthaben aller Konten der Bank:

  bank.kunden.konten.guthaben.sum()

Closures

Orientation in Objects

- Ausführbarer Codeblock
- kann weitergereicht werden

```
[1, 2, 3, 4].each {print it * 2}     // =>  2468

5.times { println 'Groovy!' }

mails.sort { mail -> mail.dateReceived }
```


Builder - Beispiel HTML

Orientation in Objects

```
def writer = new FileWriter("mypage.html")
def builder = new groovy.xml.MarkupBuilder(writer)
builder.html {
   head {
        title 'My Homepage'
   }
   body {
        h1 'Welcome'
        p 'Welcome to my Homepage'
   }
}
```

```
<html>
  <head>
    <title>My Homepage</title>
  </head>
  <body>
    <h1>Welcome</h1>
    <p>Welcome to my Homepage</p>
  </body>
</html>
```

Gliederung

Orientation in Objects

- Groovy
- **Grails**
- Live Session
- Wenn noch Zeit ist
- Quellen

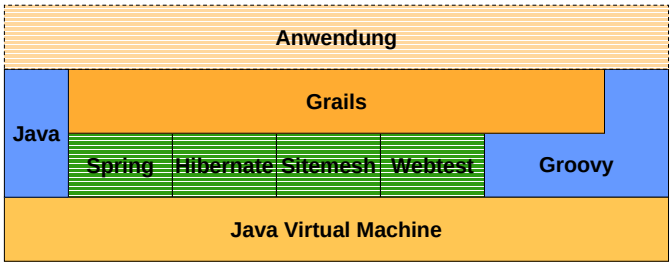
Grails - Einführung

Orientation in Objects

- MVC-Web-Framework
- inspiriert von Ruby on Rails
- Prinzipien
  - DRY - Don't Repeat Yourself
  - Convention over Configuration
- produktives und zügiges Entwickeln

Technologien in Grails

Orientation in Objects



Domänen-Modell

Orientation in Objects

- Kern jeder Grails-Anwendung
- besteht aus Domänen-Klassen (Kunde, Artikel, Rechnung)
- Grundlage für Scaffolding

```
class Benutzer {
   String name
   String passwort

   static constraints = {
        name(size:4..16, blank:false, unique:true)
        passwort(size:4..8, blank:false, passwort:true)
   }
}
```

Objekt-Relationales Mapping

Orientation in Objects

- GORM - Grails Object Relational Mapping
- Methoden für CRUD-Operationen
- dynamische Finder-Methoden

```
def item = new Item(name:'meinItem')

item.save()
def geladenesItem = Item.get(1)
def gefundenesItem = Item.findByName('meinItem')
item.delete()
```


Scaffolding

Orientation in Objects

- Generierung von Artefakten (Controller, Views)
- statisches vs. dynamisches Scaffolding
- Ausgangspunkt für Entwicklung oder für Prototyping
- Templates änderbar

Controller, Actions

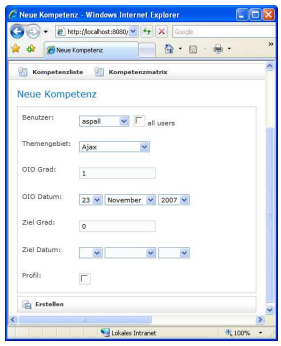
Orientation in Objects

- Aufgaben
  - Verarbeitung von Anfragen
  - direkt Ergebnis an Client senden
  - Model aufbauen und View bestimmen
- Actions werden als Closures definiert
- DataBinding

Views

Orientation in Objects

- GSP oder JSP
- Einfache Nutzung von Grails Tag-Library
- Nutzung selbstentwickelter TagLibs



Dynamic Tag Libraries

Orientation in Objects

- Bereits mitgeliefert Tags für:
  - logische Tags (If-Else, ...)
  - Iteration über Collections
  - Vereinfachung von HTML-Formularen
  - Ajax
- Eigene Tags sehr einfach zu implementieren
  - TagLib Klasse - Tags werden als Closures definiert

Dynamic Tag Libraries - Beispiel

Orientation in Objects

TagLib-Klasse:

```
class MyTagLib {

    def formatTime = {
        DateFormat formater = new SimpleDateFormat("h:m:s")
        out << formater.format(attrs['time'].getTime())
    }
}
```

Einsatz in View:

```
<g:formatTime time="${new Date()}"/>
```

Services

Orientation in Objects

- für Business-Logik
- Methoden können transaktional ablaufen
- automatische Injection von Services

```
class UserController {
    UserService userService
    ...

}
```


Gliederung

Orientation in Objects

- Groovy
- Grails
- **Live Session**
- Wenn noch Zeit ist
- Quellen

Live Session

Orientation in Objects

- einfaches Weblog-System
  - Blogeinträge anzeigen
  - Blogeintrag hinzufügen

Gliederung

Orientation in Objects

- Groovy
- Grails
- Live Session
- **Wenn noch Zeit ist**
- Quellen

Einsatzmöglichkeiten Groovy

Orientation in Objects

- Punktuell in Java-Anwendungen
- Steigern der Code-Qualität
- Testing
- Prototyping
- Dynamik für Java Enterprise Anwendungen
  - DSLs
  - ausgelagerte Logik

Einsatzmöglichkeiten Grails

Orientation in Objects

- Kleine bis mittelgroße Webanwendungen
- Alternative zu anderen Frameworks
  - Ideal wenn Spring und Hibernate eingesetzt werden sollen
- Prototyping
- Rapid Application Development
  - kleines Budget
  - Zeitdruck

Wahr oder falsch?

Orientation in Objects

```
// klassisch
assert true
assert !false

// Strings
assert "irgendein Text"
assert !""

// Collections
assert [1, "abc", 1.3]
assert ![]

// Zahlen
assert 1.1
assert !0

// Objekte
assert new Object()
assert !null
```

Typ	Wahr, wenn
Boolean	true
Matcher (Regexp)	Übereinstimmung
Collection, Maps	nicht leer
Strings	nicht leer
Number, Character	Wert nicht 0
Objekte	Referenz nicht null

Reguläre Ausdrücke

Orientation in Objects

- Pattern Operator: ~"muster"
- Finder Operator: =~
- Matcher Operator: ==~

```
if ("Hello World!" =~ /Hello/)

if ("Hello World!" ==~ /Hello\b.*/)

// -> 2.34
"1.23".replaceAll(/\d+/){ num -> num.toInteger() + 1}

// -> Hallo: 5 Welt: 4
println "Hallo Welt".replaceAll(/\w+/){match -> "$match:
   ${match.size()}"}
```

Builder - Beispiel Swing

Orientation in Objects

```
def swing = new SwingBuilder()
def frame = swing.frame(title:'Groovy Swing Demo', size:[300,300]) {
   borderLayout()
   panel(constraints:BorderLayout.CENTER) {
        label(text:"Hello World")
        button(text:'Click Me', actionPerformed: { println "clicked" })
   }
   panel(constraints:BorderLayout.SOUTH) {
        label(text:"How are you?")
   }
}
frame.pack()
frame.show()
```

Verwendung Groovy - Scripting

Orientation in Objects

- Shell-Scripting
- Build-Management
- Scriptom
- Makros

Verwendung Groovy - Punktuell

Orientation in Objects

- XML Verarbeitung
- Datei IO
- Webservices
- DB Zugriff
- Templating

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <tag attribute="foo">
        <bar>
            <value>Bar</value>
        </bar>
    </tag>
    <tag attribute="" />
</root>
```


Verwendung Groovy - Prototyping, Testing

Orientation in Objects

- GUI
- schnelle Test-Entwicklung
  - Mocking
- Expando, MOP

Verwendung Groovy - JEE

Orientation in Objects

- für kleine bis mittelgroße Anwendungen
  - Code-Qualität, Code-Lesbarkeit
- Integration in JEE Projekten
  - DSL
  - Business Rules auslagern
  - programmatische Konfiguration
  - UI / Application Customizing
  - Änderungen am laufenden System durchführen

Verwendung Grails

Orientation in Objects

- Prototyping
- komplette "Industrial Strength" Webanwendungen
  - Entwicklungsprozess wird beschleunigt
  - Allround Lösung für Web-Projekte
  - Rapid Application Development

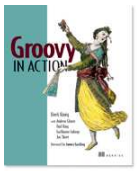
Gliederung

Orientation in Objects

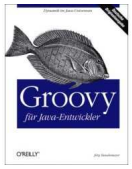
- Groovy
- Grails
- Live Session
- Wenn noch Zeit ist
- **Quellen**

Literaturhinweise

Orientation in Objects



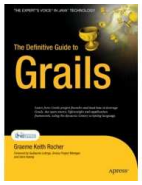
- Groovy in Action
  - **Sprache: Englisch**
    Broschiert - 696 Seiten - Manning
    Erscheinungsdatum: Januar 2007
    ISBN: 1-932394-84-2
  - **auch in Deutsch: „Groovy im Einsatz“**



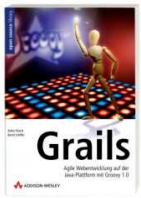
- Groovy für Java-Entwickler
  - **Sprache: Deutsch**
    Broschiert - 352 Seiten - O'Reilly
    Erscheinungsdatum: September 2007
    ISBN: 978-3-89721-483-5

Literaturhinweise

Orientation in Objects



- The Definitive Guide to Grails
  - **Sprache: Englisch**
    Broschiert - 384 Seiten - Springer-Verlag
    Erscheinungsdatum: Dezember 2006
    ISBN: 978-1590597583



- Grails
  - **Sprache: Deutsch**
    Broschiert - 300 Seiten - Addison-Wesley
    Erscheinungsdatum: Dezember 2007
    ISBN: 978-3827324986

Links

Orientation in Objects

- Groovy
  - http://groovy.codehaus.org/
- Grails
  - http://grails.org
- Schnellstart mit InfoQ-Buch
  - http://www.infoq.com/minibooks/grails
- News über Groovy und Grails
  - http://aboutgroovy.com

Orientation in Objects

Fragen?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Orientation in Objects

Vielen Dank für Ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0