

# Úvod do Groovy pro Java experty

**Václav Pech**

Senior Software Developer

JetBrains, Inc.



## Něco o mně

- Václav Pech
  - Programátor
  - Nadšenec do Groovy
- JetBrains
  - Dodavatel Vašich oblíbených nástrojů
    - IntelliJ IDEA, TeamCity, ReSharper a další



javamagazin



## Něco o prezentaci

- Část 1: Groovy syntaxe
  - Nové obzory pro Java vývojáře
- Část 2: Příklady uplatnění pro Groovy
  - Skriptování, Buildery, DSLs
- Otázky a odpovědi
- **POZOR!:** Během prezentace bude ukazován a editován kód

## Jeden jazyk to vše nezvládne



- SQL
- XHTML
- JavaScript
- XUL (XML UI Language)
- SVG (Scalable Vector Graphics)
- CSS (Cascading Stylesheets)
- XPath
- Velocity, XSL-T
- ...
- **Groovy** – Scripting, DSLs, builders, ...

## Groovy



- Dynamický, objektově orientovaný skriptovací jazyk
  - Syntaxe podobná Javě
    - Včetně anotací a enumerací
    - Snadné osvojení pro Java vývojáře
  - Překládá se do Java bytecode
  - Velmi snadná integrace s Javou
  - Dynamické typování

## Groovy

- Open sourced
- Od 2004 pod JSR 241: The Groovy Programming Language
- V současnosti rozvíjen společností G2one
- Podpora v IDE
  - Pluginy pro všechna IDE
  - JetGroovy plugin pro IntelliJ IDEA
- Runtime podpora
  - Spring, Seam, ...



## Properties



```
class ProgrammingLanguage {  
    String name  
    String version  
    boolean easy=true  
}  
def groovy=new ProgrammingLanguage(  
    name: 'Groovy', version: '1.5', easy: true)  
  
def java=new ProgrammingLanguage(name: 'Java')  
java.version='1.6'
```

# Closures



```
Closure multiply1 = {int a, int b -> return a * b}
```

```
Closure multiply2 = {int a, int b -> a * b}
```

```
Closure multiply3 = {a, b -> a * b}
```

```
def multiply4 = {a, b -> a * b}
```

# Closures – implicitní parametr



```
def tripple1 = {int number -> number * 3}
```

```
def tripple2 = {number -> number * 3}
```

```
def tripple3 = {it * 3}
```

## Iterace



```
(1..10).each{number -> println number * 3}
```

```
1.upto(10) {println it * 3}
```

```
Closure tripple = {it * 3}
```

```
1.step(11, 1) {println tripple(it)}
```

## GDK



- `java.util.Collection`
  - `each()`, `find()`, `join()`, `min()`, `max()` ...
- `java.lang.Object`
  - `any()`, `every()`, `print()`, `invokeMethod()`, ...
- `java.lang.Number`
  - `plus()`, `minus()`, `power()`, `upto()`, `times()`, ...
- ...

# Operator overloading



```
class Money implements Comparable {
    BigDecimal amount
    String currency

    Money plus(Money other) {
        checkCurrencies other
        return new Money(amount:this.amount
            + other.amount, currency:this.currency)
    }

    Money minus(Money other) {
        checkCurrencies other
        return new Money(amount:this.amount
            - other.amount, currency:this.currency)
    }
}
```

## Stringy a GStringy



```
String greeting1 = 'Hi ' + friend.name
```

```
String greeting2 = "Hi $friend.name"
```

## Multi-line Stringy a GStringy



```
def emailText="""
Dear ${customer.name},

thank you for your email on ${customer.lastEmail.topic}.
${customer.lastEmail.date < 1.week.ago ?
    "Sorry for not responding for so long" : ''}
...
Yours ${currentUser.name}
"""
```

## Operatory



- Spread
  - `def names = customers*.name`
  - `['Java', *['Groovy', 'Ruby', 'Scala', 'Jython']]`
- Safe dereference
  - `customer.shippingAddress?.street`
- Membership
  - `if (code.language in ['Java', 'Groovy'])`
- Elvis
  - `return user.locale?:defaultLocale`

## Operatory



- Spaceship (compareTo)
  - `a <=> b`
- Find and Match - `=~`, `==~`
  - `methodName =~ 'findAll(.*?)Employees'`
- Subscript
  - `map['key'] == map.key == map.get('key')`
  - `'VeryLongString'[4..-1]`
- As
  - `new Thread({println 'In parallel'}) as Runnable)`

## Rozšíření syntaxe



- Rozšířené JDK třídy (GDK)
- Dynamické (duck) typování – volitelně!
- Rozšíření syntaxe
  - Properties
  - Closures
  - Named parameters (`new Customer(name:'Joe')`)
  - Kolekce a mapy
  - Přetěžování operátorů
  - GString (`"Name: $name"`)
  - GPath (`employees.findAll{it.age>30}*.name`)
  - Datový typ range (e.g. `9..29` or `0..-1`)
  - Integrace regulárních výrazů

## Testování



- Testy Java i Groovy kódu
- Spouštěné pomocí Ant či Maven
- Integrovaná podpora JUnit
  - `assert...()`, `shouldFail()`, ...
- Duck typing
- Mocking
  - Integrovaná podpora stubů a mocků

## Pasáže pro starší a pokročilé



- Buildery
  - Konstrukce hierarchických datových struktur
- Skriptování
  - Přidávání a překlad kódu za běhu aplikace
- Silná podpora pro DSL
  - Využívá schopnosti meta-programování

## Buildery



- Konstrukce hierarchií

```
xml.records() {  
    order(id: 'PL19826714', date: '21-01-2008') {  
        item(quantity: 10) {  
            product(id: '76327')  
            price(base: 100) {  
                volumeDiscount(value: 5)  
            }  
        }  
    }  
}
```

## Buildery - GAnt



```
ant.sequential {
  myDir = "target/AntTest/"
  mkdir(dir: myDir)
  copy(todir: myDir) {
    fileset(dir: "src/test") {
      include(name: "**/*.groovy")
    }
  }
}
List dirs = ['core', 'lib', 'engine', 'gui', 'db']
for (String currentDir:dirs) {
  String targetDir="target/$currentDir"
  mkdir(dir:targetDir)
```

## Buildery – Spring config



```
dataSource(BasicDataSource) {
    driverClassName = "org.hsqldb.jdbcDriver"
    url = "jdbc:hsqldb:mem:shopDB"
}

sessionFactory(ConfigurableLocalSessionFactoryBean) {
    dataSource = dataSource
    hibernateProperties = ["hibernate.hbm2ddl.auto": "create-drop",
        "hibernate.show_sql": true]
}

calculator(demo.shop.CalculatorImpl) {bean ->
    bean.singleton = true
    bean.autowire = 'byType'
}
```

## Buildery – domain classes



```
DomainBuilder builder = new DomainBuilder()
builder.house(type: 'semi-detached') {
    address(street: 'Groovy lane', number: '9124', city: 'Prague', country: 'cz')
    room(name: 'kitchen', size: 10.m2) {
        table(color: 'blue')
    }
    room(name: 'hall', size: 6.m2)
}
```

## Skriptování



- Vyhodnocení Groovy kódu za běhu
  - Vlastní kód či testy
  - Implementace rozhraní

```
def classDefinition = new GroovyShell().evaluate(codePane.text)
Runnable task=classDefinition.newInstance()
new Thread(task).start()
```

## DSL



- Jednoúčelové jazyky
- Cílené na určitou problémovou doménu
- Poskytují přátelštější API
  - Externí
    - Vlastní gramatiky a parsery
    - Vlastní syntaxe
  - Interní
    - Vnořené do hostovacího jazyka
    - Syntaxe se řídí gramatikou hostovacího jazyka

## DSL – práce s datумы



```
use (org.codehaus.groovy.runtime.TimeCategory) {  
    println "Tomorrow: ${1.day.from.today}"  
    println "A week ago: ${1.week.ago}"  
    println "Date: ${1.month.ago + 1.week + 2.hours - 5.minutes}"  
    println "Date ${ (1.month + 10.days).ago}"  
}
```

## DSL – Hibernate criteria



```
def participants = Participant.createCriteria().list {
    gt('age', age)
    or{
        eq('interest', 'Java')
        eq('interest', 'Groovy')
    }
    jug {
        ilike('country', 'de')
    }
    order('lastName', 'asc')
}
```

## DSL – manipulace s účty



```
Money money = new Money(amount: 350, currency: 'eur')
getAccount('Account1').withdraw money
getAccount('Account3').deposit money
```



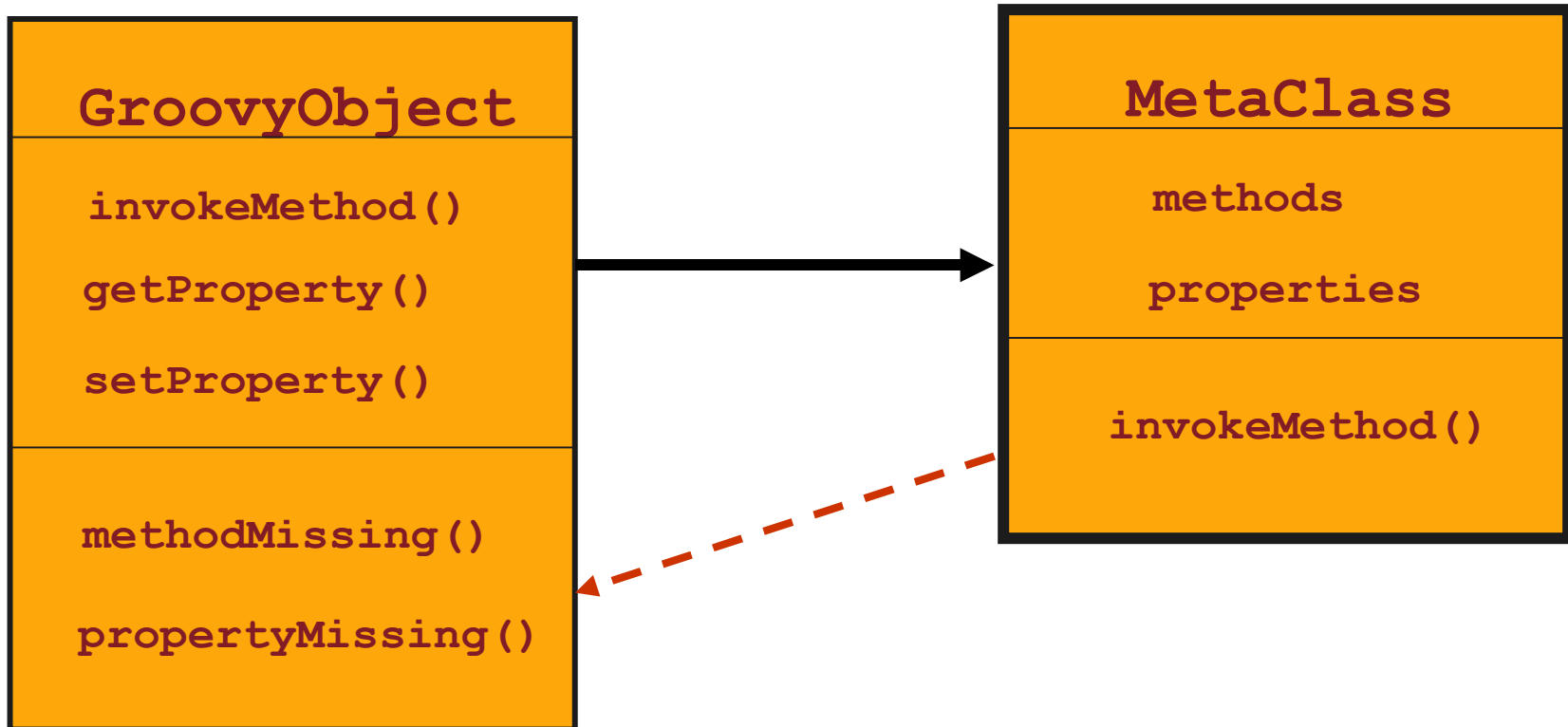
```
"Account1" >> 350.eur >> "Account3"
```

# Meta-programování



- Buildery
- Kategorie
- Změna postupu při vyvolání metod
  - Předefinovat invokeMethod()
  - Předefinovat methodMissing()
  - Předefinovat propertyMissing()
  - Přidat metodu nebo property k meta-třídě
  - Předefinovat invokeMethod() na meta-třídě
  - Přidat Expando meta-class na instanci

# Dynamické vyvolání metod

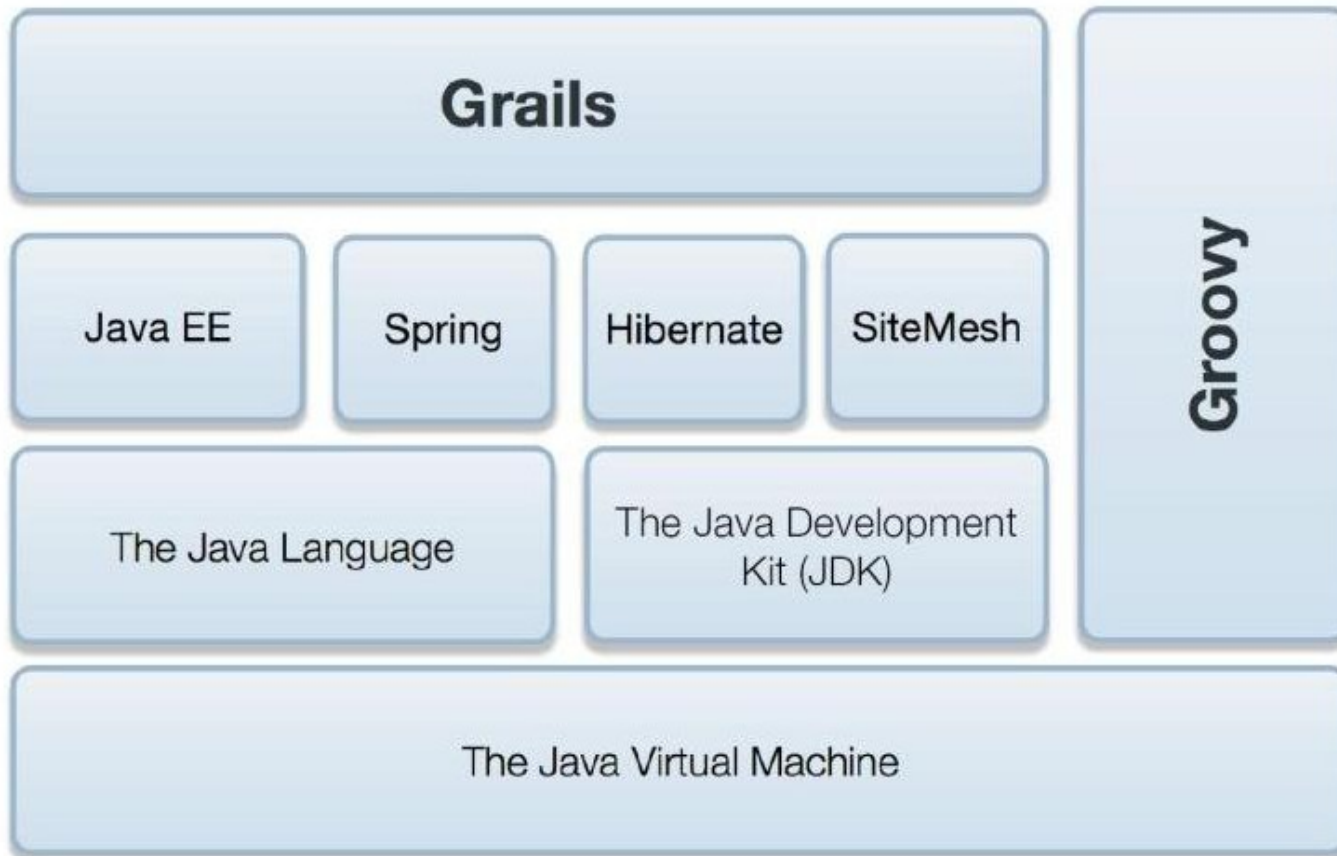


## Grails



- Framework pro tvorbu webovských aplikací
- MVC architektura
- Open sourced
  
- Cílem je co nejvíce zpříjemnit a zefektivnit vývoj webovských aplikací

# Technologie

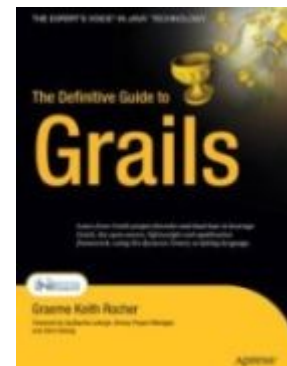
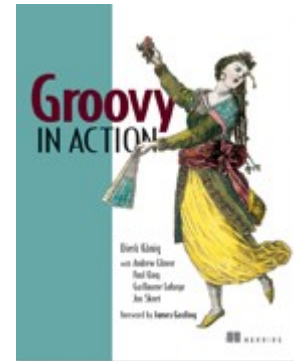


## Komunita

- G2One Inc. – G2One.com
  - Guillaume Laforge (Groovy Project Lead)
  - Graeme Rocher (Grails Project Lead)
- <http://groovy.codehaus.org/>
- <http://grails.org/>
- Novinky ze světa Groovy
  - <http://aboutgroovy.com>
  - <http://groovyblogs.org>
  - <http://groovy.dzone.com>

## Literatura

- Groovy in Action
- The Definitive Guide to Grails



## Shrnutí

- Groovy
  - Podařené rozšíření Javy
  - Vylepšení syntaxe
  - Buildery, Skriptování, DSL
  - **Kontakt: [vaclav@jetbrains.com](mailto:vaclav@jetbrains.com)**



# Otázky