BRON družba za informacijsko tehnologijo

Skriptni jeziki v JVM

Borut Korošin
borut@bron.si

16. strokovno srečanje
SIOUG 2011
Kongresni center Bernardin
Portorož, 18. - 21. September 2011

BRON

družba za informacijsko tehnologijo

- 16 redno zaposlenih, 2 redna zunanja sodelavca (polni delovni čas), stalno izobraževanje v tujini in pridobivanje certifikatov
- združujemo poslovna, tehnična in praktična znanja pri uvajanju zahtevnih informacijskih sistemov
- sodelujemo v vseh fazah projektov, od zasnove, planiranja, implementacije do upravljanja
- področja dela:
  - Oracle Business Intelligence (OBI, ODI, OWB)
  - Oracle Hyperion Planning
  - Razvoj aplikativne programske opreme (Oracle, Java)
  - načrtovanje in implementacija zahtevnih storitveno orientiranih sistemov (SOA)
- zastopamo ORACLE sgi INNOVATION FOR RESULTS

ORACLE Gold Partner
Specialized
Oracle Business Intelligence
Foundation

Java == JVM ?

"Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less." - James Gosling, TSSJS 2011

Java != JVM

"What I really care about is the Java Virtual Machine as a concept, because that is the thing that ties it all together; it's the thing that makes Java the language possible; it's the thing that makes things work on all kinds of different platforms; and it makes all kinds of languages able to coexist." - James Gosling, TSSJS 2011

Zakaj mi samo Java ni dovolj?

Ker mi je včasih hitrost razvoja pomembnejša od hitrosti izvajanja.

Za JVM obstaja skoraj 100 implementacij različnih programskih jezikov

Skriptni jeziki

Običajno interpretirani

Namenjeni kontroliranju ene ali več aplikacij

- Shell skripte (sh, ...)
- GUI scripting (JavaFX)
- Aplikacijsko specifični (Emacs LISP)
- Spletni brkljalniki (JavaScript)
- Procesiranje teksta (awk, sed, perl)
- Splošni dinamični jeziki (Ruby, Python)
- Vgrajeni v aplikacije (za razširitev)

Podpora skriptnim jezikom v JVM

- Java SE6 – JSR223(Scripting for the Java Platform)
  - Vgrajen Rhino Javascript engine
  - Lahko uporabljamo za zagon skriptne kode napisane v katerem od dinamičnih jezikov (Ruby, Python, JavaScript,...)
- Java SE7 - JSR292(Supporting Dynamically Typed Languages on the Java Platform)
  - Dodana `invokedynamic` bytecode
  - Da Vinci Machine – referenčna implemetacija JSR292

Zanimivi skriptni jeziki v JVM

- Jython
- Scala
- Fantom
- JRuby
- Groovy

Jython

- Python za JVM
- Eden prvih skriptnih jezikov na JVM (1997)
- Eden od dveh skriptnih jezikov za WebSphere (drugi je Jacl – Tcl/Java)
- Med 2005 in 2008 se je razvoj ustavil
- Trenutno Jython 2.5.2 – marec 2011
- Dinamični podatkovni tipi
- Počasen

Scala

- Kombinira elemente objektno orientiranega in funkcionalnega programiranja
- Statični podatkovni tipi
- Closures
- Java concurrency API in actor model

Fantom

- Izvaja v JVM in .NET CLR
- Statični podatkovni tipi
- Omogoča tudi dinamične klice
- Poenostavljeni API (npr. java.io)
- Podpira funkcionalno programiranje
- Actor model concurrency

JRuby

- Java implementacija Ruby
- Začetek 2001
- 2007 izvaja Ruby on Rails približno enako hitro kot običajen Ruby
- Objektno orientiran
- Closures
- Duck typing
- Compiler in interpreter

Groovy

- Sintaksa zelo podobna Javi
- Dinamični podatkovni tipi
- Duck typing
- Closures
- Grails
- Zelo uporaben za testiranje

Zakaj Groovy?

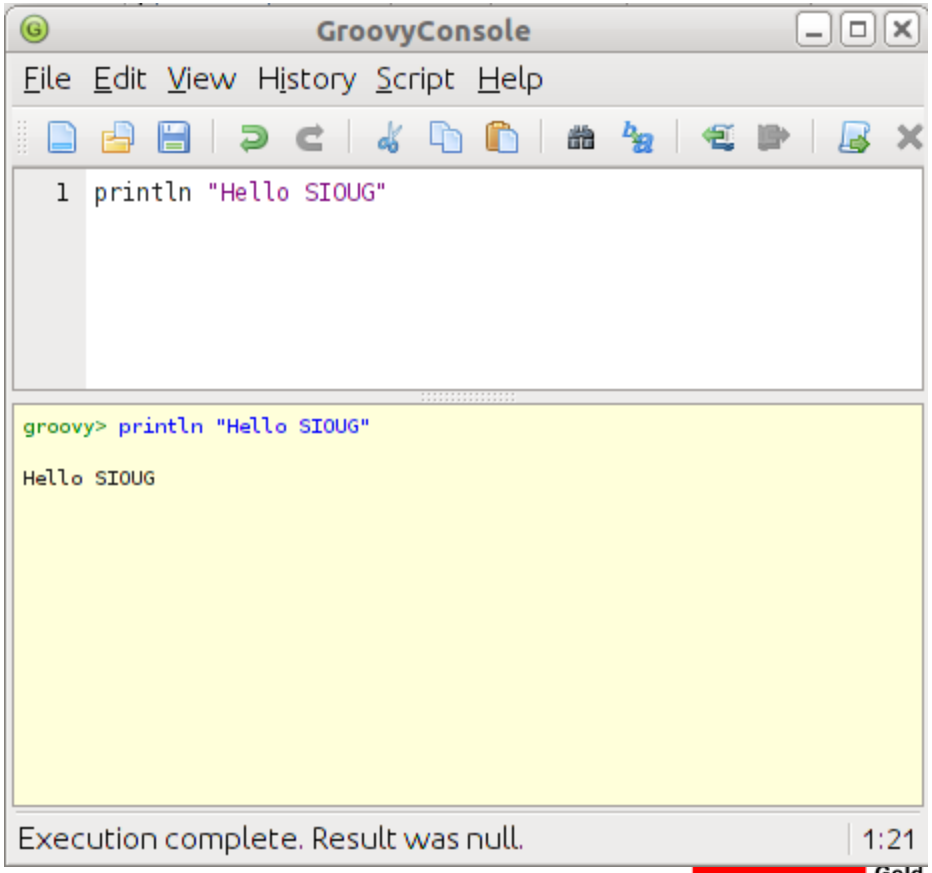
- Ne znam Ruby
- Jython je prepočasen
- Prenosljivost na .NET CLR me ne zanima
- Scala mi je sicer všeč, vendar me pritiskajo časovni okviri projekta
- Dokaj dobra podpora v IDE
- Znam Javo

Kako začeti

- http://groovy.codehaus.org/Download
- Dolpotegnem Binary release in ga razpihnem
- Kreiram GROOVY_HOME in dodam %GROOVY_HOME%/bin v PATH
- Vtipkam groovy -version v komandno vrstico

Prvi korak

- V komandni vrstici vtipkam groovyConsole
- println “Hello SIOUG”
- Ctrl+R



Primitivni tipi

- Številke so objekti
- 1.equals(2) // false
- 9.99.class

Collections

- Liste: def zvezek=[]
- Več kontrole: def zvezek=[] as String[]
- Tudi mape so enostavne
  - def sioug=[vceraj:“PON“, danes:“TOR“, jutri:“SRE“]

Duck typing

„If it walks like a duck and swims like a duck and quacks like a duck, it must be a duck”

- Statični podatkovni tipi

```
Raca mojaRaca = new Raca()
mojaRaca.gaga()
```

- Dinamični (duck) podatkovni tipi

```
def raca = new Raca()
mojaRaca.gaga()
```

Operator overloading

- a+b a.plus(b)
- a==b a.equals(b)
- a++ a.next()
- a[b] a.getAt(b)
- a<<b a.leftShift(b)
- a<=>b a.compareTo(b)

Closures

- Objekt - blok kode ali kazalec na metodo
- (1..5).each {print it}
- Closure je lahko tudi argument metode
- .& pretvorimo metodo v closure
- Currying:

```
def add = {x, y -> return x+y}
def addOne = add.curry(1)
def sest = addOne(5)
```

I/O

- Branje datoteke

  new File(“out.txt“).eachLine {println it}

- Pisanje datoteke

  file.write(“tekst”)

  file.append(“se vec“)

  file << “dodajam”

XML

- XmlSlurper

```
def pijace = new XmlSlurper().parse(new File("test.xml"))
```

```
<pijace>
  <steklenica vsebina='pivo' sorta='Union'>
    <namen>zeja</namen>
  </steklenica>
  <steklenica vsebina='voda'>
    <namen>pranje</namen>
  </steklenica>
 <steklenica vsebina='pivo' sorta='Lasko'>
    <namen>zeja</namen>
  </steklenica>
</pijace>
```

XML

```
def vseSteklenice = pijace.steklenica
    def prva = pijace.steklenica[0]
    prva.vsebina()
    prva.@sorta.text()
    prva.namen.text()
    def pivske =
    vseSteklenice.findAll{it.vsebina().contains('pivo'
    )}
```

Domensko specifični jeziki

- Opišemo rešitev v jeziku, ki je blizu problemu
- Izkoristimo notranje mehanizme, ki so na voljo v groovyu

```
def spisek = []
def dodaj = spisek.&add
dodaj “mleko”
dodaj “kruh”
```

Java + Groovy

```
ClassLoader parent = this.getClass().getClassLoader();
GroovyClassLoader gcl = new GroovyClassLoader(parent);
Class groovyValidator = gcl.parseClass(validatorSource);
GroovyObject validator = (GroovyObject) groovyValidator.newInstance();
Object[] params = new Object[2];
params[0] = sb;
params[1] = ub;
result =  (BlokadeResult) validator.invokeMethod("validate", params);
```

Groovy++

- Statični podatkovni tipi
- Omogoča mešanje statičnih in dinamičnih pod. tipov
- Performančno se približuje čisti javi
- Podpora za repno rekurzijo
- Traits – Interface s statičnimi metodami in možnimi defaultnimi implementacijami nekaterih metod

Kako naprej

- Začel sem tako, da sem v Groovy classu pisal skoraj čisto Javo
- Ko spoznavam možnosti Groovya jih sproti uporabljam
- Uporabljam ga tam, kjer mi hitrost ni tako pomembna
- Uporabljam ga tam, kjer moram spreminjati logiko delovanja in ne želim pripravljati novih deployev

BRON

družba za informacijsko tehnologijo

ORACLE Gold Partner

Specialized
Oracle Business Intelligence
Foundation