Let's move the Java world!
GeeCON

Polyglot Programming in the JVM

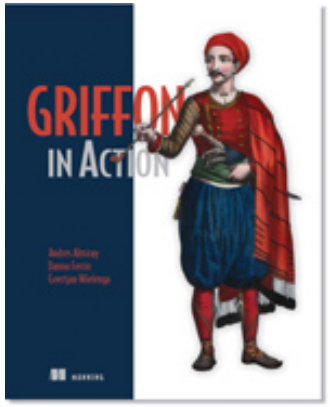
Or how I Learned to Stop Worrying and Love the JVM

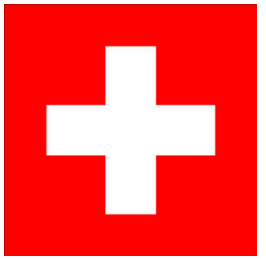
Andres Almiray | Canoo Engineering AG

Let's move the Java world!
GeeCON

About the Speaker

- **Java** developer since the beginning
- True believer in **Open Source**
- **Groovy** committer since 2007
- Project lead of the **Griffon** framework
- Currently working for canoo





JAVA™
CHAMPIONS

Let's move the Java world!
GeeCON

Some facts about Java

- Previous name was Oak. Bonus points for knowing its real name before that
- Made its public appearance in 1995
- C/C++ were king at the time
- Networking, multithreading were baked right into the language
- Developers came for the applets and stayed for the components (JEE and all that jazz)

Let's move the Java world! GeeCON

However...

- It‘s already in its teens
- It has not seen a major feature upgrade since JDK5 was released back in 2004 -> generics (and we do know how that turned out to be, don’t we?)
- JDK7 has been delayed again (late 2010). Some features might or might not make the cut (the closures debacle)

More so...

- It is rather verbose when compared to how other languages do the same task
- Its threading features are no longer enough. Concurrent programs desperately cry for immutability these days

Let's move the Java world!
GeeCON

Truth or myth?

- Is Java oftenly referred as **overengineered**?
- Can you build a Java based web application (for arguments sake a basic Twitter clone) in less than a day‘s work **WITHOUT** an IDE?
- Did James Gosling ever say he was **threatened with bodily harm** should operator overloading find its way into Java?

The JVM is a great place to work however Java makes it painful sometimes...

What can we do about it?!

Let's move the Java world!
GeeCON

Disclaimer

(this is not a bash-the-other-languages talk)

Reduced Verbosity

Let's move the Java world!
GeeCON

Standard Beans

```
public class Bean {
   private String name;

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }
}
```

Let's move the Java world! GeeCON

Standard Beans

```
public class Bean {
   private String name;

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }
}
```

Let's move the Java world!
GeeCON

Standard Beans


```
class Bean {
   String name
}
```

Let's move the Java world! GeeCON

Standard Beans

```
class Bean(var name:String)
```

```
class Bean {
  @scala.reflect.BeanProperty
  var name: String
}
```

Let's move the Java world!
GeeCON

Standard Beans

```
(defstruct Bean :name)
```

Let's move the Java world! GeeCON

Closures (or Functions)

```
public interface Adder {
   int add(int a, int b);
}

public class MyAdder implements Adder {
   public int add(int a, int b) {
      return a + b;
   }
}
```

Let's move the Java world! GeeCON

Closures (or Functions)

```
// JDK7 Closures proposal

#(int a, int b)(a + b)

#(int a, int b) { return a + b; }
```

Let's move the Java world!
GeeCON

Closures (or Functions)


```
def adder = { a , b -> a + b }
```

Let's move the Java world!
GeeCON

Closures (or Functions)

```
val adder = (a:Int, b:Int) => a + b
```

Let's move the Java world!
GeeCON

Closures (or Functions)

```
(defn adder [a b] (+ a b))
```

Let's move the Java world!
GeeCON

Enhanced switch

```
char letterGrade(int grade) {
  if(grade >= 0 && grade <= 60) return 'F';
  if(grade > 60 && grade <= 70) return 'D';
  if(grade > 70 && grade <= 80) return 'C';
  if(grade > 80 && grade <= 90) return 'B';
  if(grade > 90 && grade <= 100) return 'A';
  throw new IllegalArgumentException("invalid
grade "+grade);
}
```

Let's move the Java world! GeeCON

Enhanced Switch


```
def letterGrade(grade) {
  switch(grade) {
      case 90..100: return 'A'
      case 80..<90: return 'B'
      case 70..<80: return 'C'
      case 60..<70: return 'D'
      case 0..<60: return 'F'
      case ~"[ABCDFabcdf]":
           return grade.toUpperCase()
  }
  throw new IllegalArgumentException('invalid
grade '+grade)
}
```

Let's move the Java world! GeeCON

Enhanced Switch

```
val VALID_GRADES = Set("A", "B", "C", "D", "F")
def letterGrade(value: Any):String = value match {
  case x:Int if (90 to 100).contains(x) => "A"
  case x:Int if (80 to 90).contains(x) => "B"
  case x:Int if (70 to 80).contains(x) => "C"
  case x:Int if (60 to 70).contains(x) => "D"
  case x:Int if (0 to 60).contains(x) => "F"
  case x:String if VALID_GRADES(x.toUpperCase) =>
x.toUpperCase()
}
```

Let's move the Java world! GeeCON

Enhanced Switch

```
(defn letter-grade [grade]
  (cond
    (in grade 90 100) "A"
    (in grade 80 90) "B"
    (in grade 70 80) "C"
    (in grade 60 70) "D"
    (in grade 0 60) "F"
    (re-find #"[ABCDFabcdf]" grade)
(.toUpperCase grade)))
```

Let's move the Java world!
GeeCON

Java Interoperability

Let's move the Java world! GeeCON

All of these are true

- Java can call Groovy, Scala and Clojure classes as if they were Java classes
- Groovy, Scala and Clojure can call Java code without breaking a sweat

- In other words, interoperability with Java is a given. No need for complicated bridges between languages (i.e. JSR 223)

Ok, so...
What else can these languages do?

Let's move the Java world!
GeeCON

All of them

- Native syntax for collection classes
- Everything is an object
- Closures!
- Regular expressions as first class citizens
- Operator overloading

Let's move the Java world!
GeeCON

Groovy

- Metaprogramming (HOT!) both at buildtime and runtime
- Builders
- Healthy ecosystem: Grails, Griffon, Gant, Gradle, Spock, Gaelyk, Gpars, CodeNarc, etc...

- Not an exhaustive list of features!

Let's move the Java world!
GeeCON

Scala

- Richer type system
- Type inference
- Pattern matching (case classes)
- Actor model
- Traits

- Not an exhaustive list of features!

Let's move the Java world!
GeeCON

Clojure

- Data as code and viceversa
- Immutable structures
- STM

- Not an exhaustive list of features!

Demo

Other places where you'll find Polyglot Programming

Let's move the Java world!
GeeCON

Web app development

- XML
- SQL
- JavaScript
- JSON
- CSS
- Flash/Flex/ActionScript

Let's move the Java world!
GeeCON

Next-Gen datastores (NoSQL)

- FleetDB -> Clojure
- FlockDB -> Scala
- CouchDB, Riak -> Erlang
- By the way, watch out for Polyglot Persistence ;-)

Let's move the Java world!
GeeCON

Build systems

- Gradle, Gant -> Groovy
- Rake -> Ruby/JRuby
- Maven3 -> XML, Groovy, Ruby

Let's move the Java world!
GeeCON

Parting thoughts

- Java (the language) may have reached its maturity feature wise
- Other JVM languages have evolved faster
- Polyglot Programming is not a new concept
- Download and play with each of the demoed languages, maybe one of them strikes your fancy

Let's move the Java world!
GeeCON

Resources

Let's move the Java world!
GeeCON

Q & A

Thank you!