DEVNEXUS™

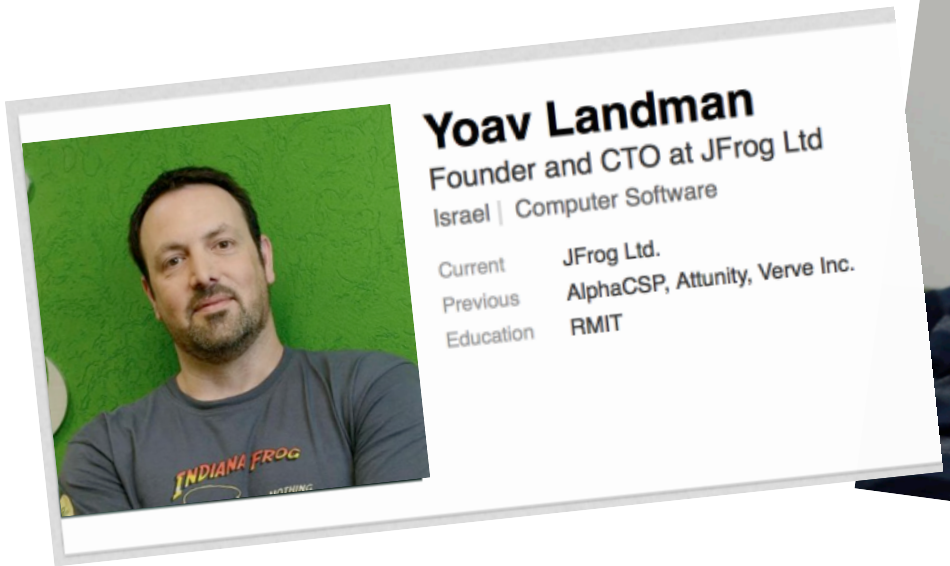
The Groovy Puzzlers

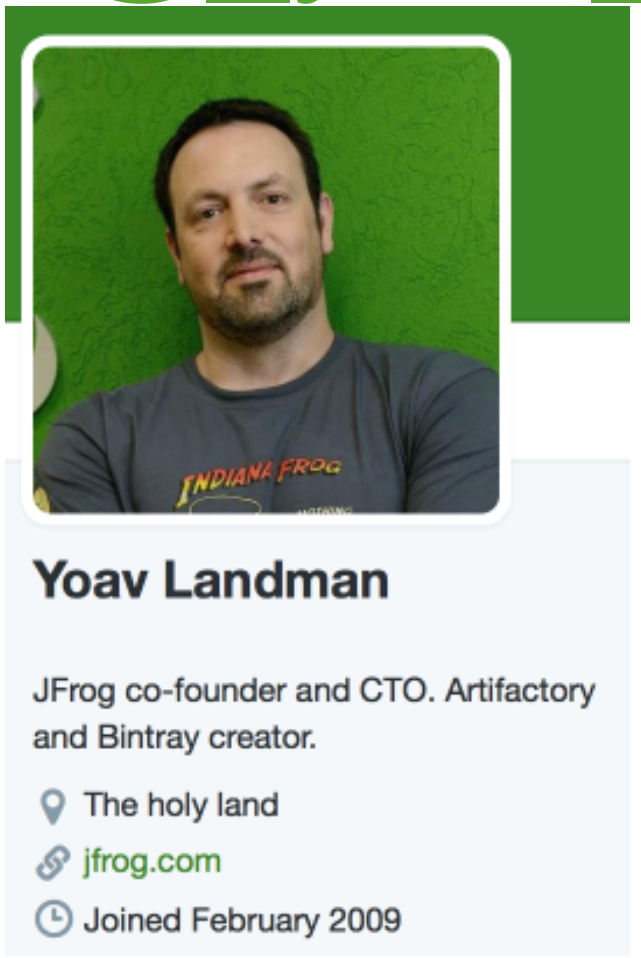
As usual - Traps, Pitfalls, and End Cases

Who we are

https://github.com/yoav

@_yoav_





Who we are?

github.com/jbaruch

@jbaruch

linkd.in/jbaruch

stackoverflow.com/users/402053/jbaruc

What Frog?

What Frog?

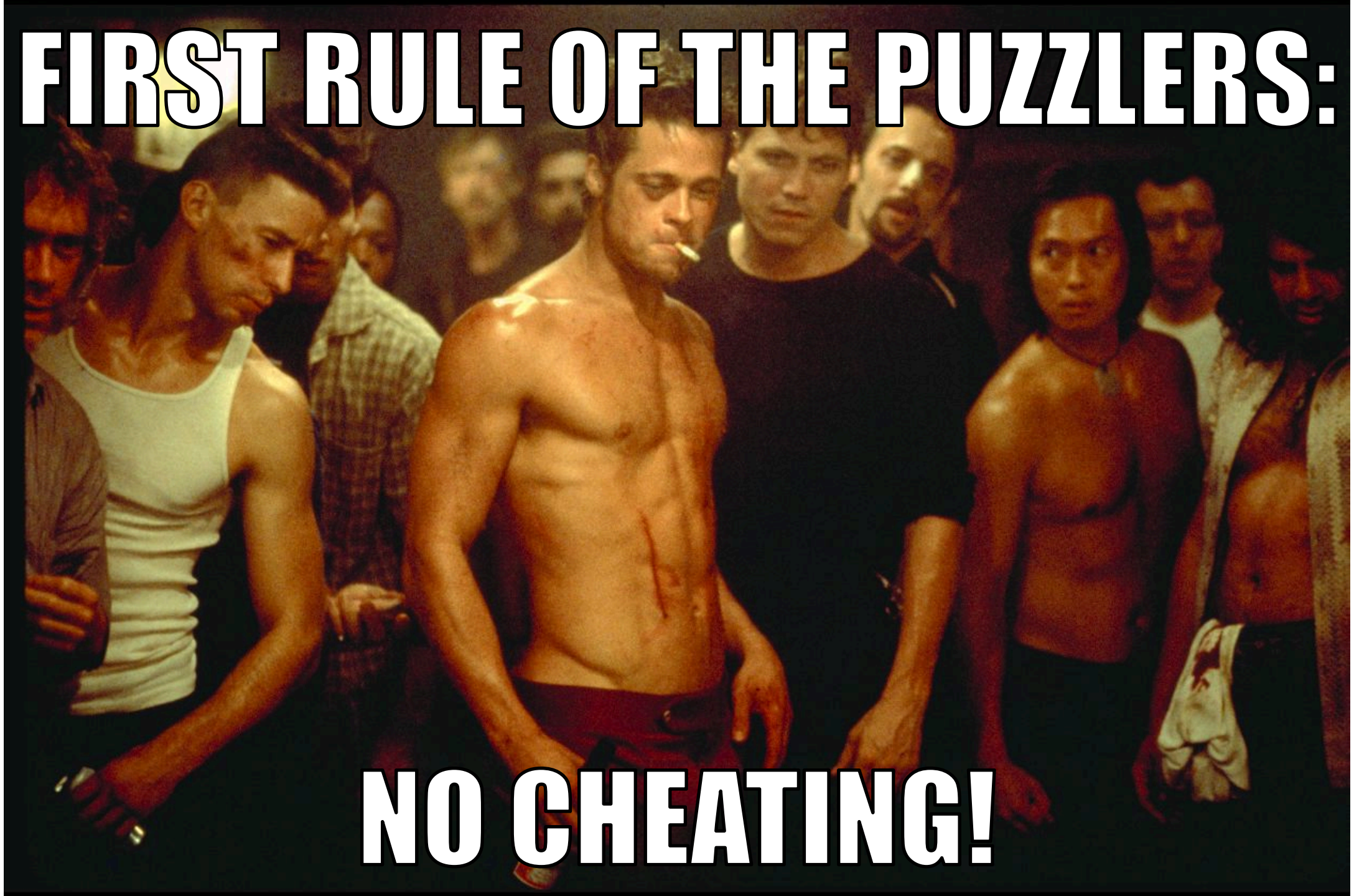
What Frog?

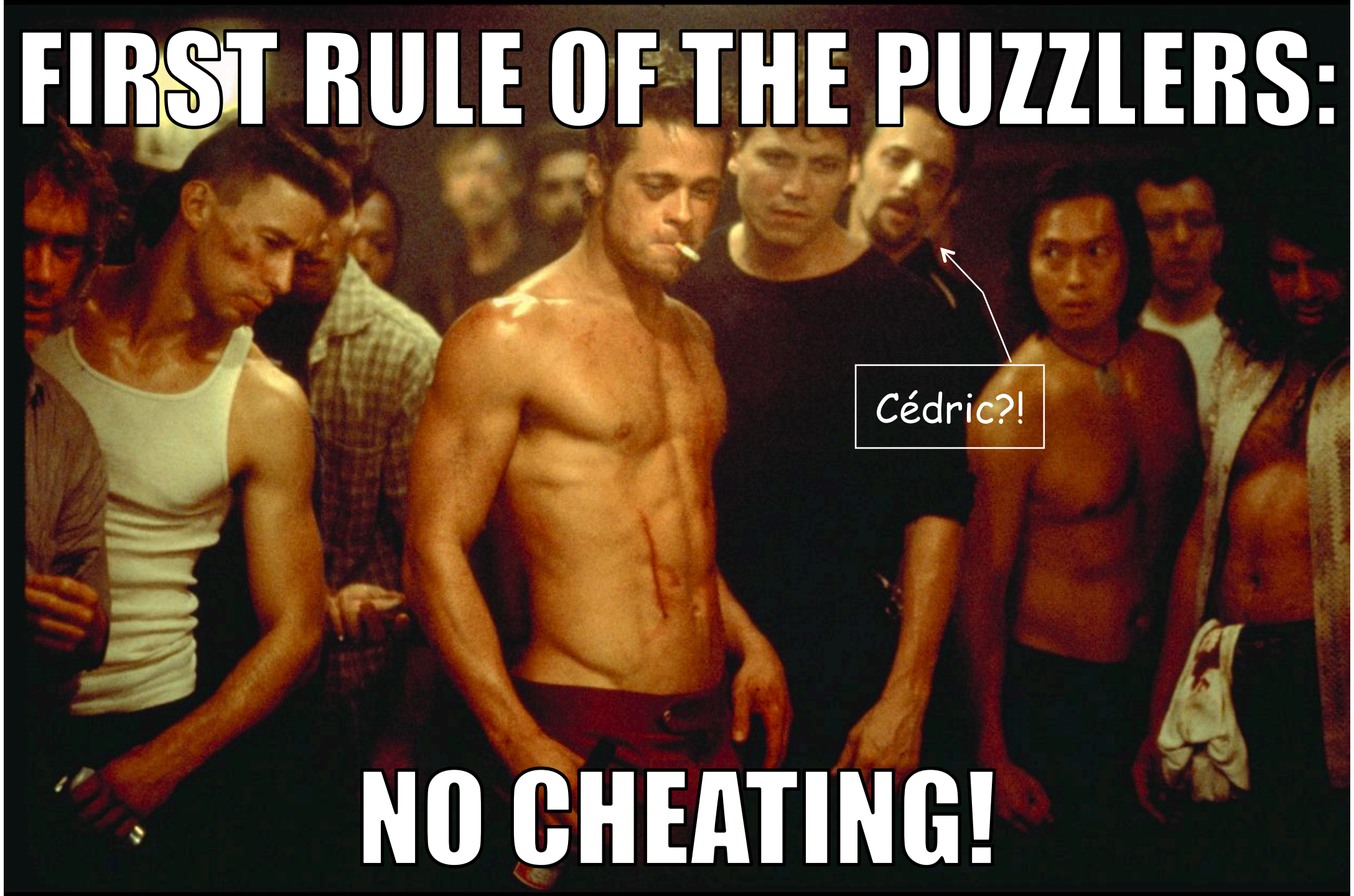
What Frog?


CLICK AND HACK
THE TYPING BROTHERS

BTW…

1. Two entertaining guys on stage
2. Funny Puzzling questions
3. You think and vote
4. Lots of T-shirts flying in the air
5. Official twitter handle!
   groovypuzzlers


FIRST RULE OF THE PUZZLERS:
NO CHEATING!


FIRST RULE OF THE PUZZLERS:
Cédric?!
NO CHEATING!

Consider the puzzlers correct as of 2.4.1


CHALLENGE
ACCEPTED
quickmeme.com


Greach
the Groovy spanish conf
MADRID - MARCH
28TH & 29TH

```
class Conference {def name; def year}

def gr = new Conference(name: 'Greach', year: 2014)

gr.each {println it}
```

A. class=class Conference
   name=Greach
   year=2014

B. Conference@XXXXXX

C. Startup failure

D. Greach
   2014

NOT EXACTLY ITERATION, IS IT?

RTFS (READ THE F***ING SOURCE)!

```
public static Collection asCollection(Object value) {
    if (value == null) {...}
    else if (value instanceof Collection) {...}
    else if (value instanceof Map) {...}
    else if (value.getClass().isArray()) {...}
    else if (value instanceof MethodClosure) {...}
    else if (value instanceof String) {...}
    else if (value instanceof GString) {...}
    else if (value instanceof File) {...}
    else if (value instanceof Class && ((Class)value).isEnum()) {...}
    else {
        // let's assume it's a collection of 1
        return Collections.singletonList(value);
    }
}
```

ABSOLUTELY GROOVY

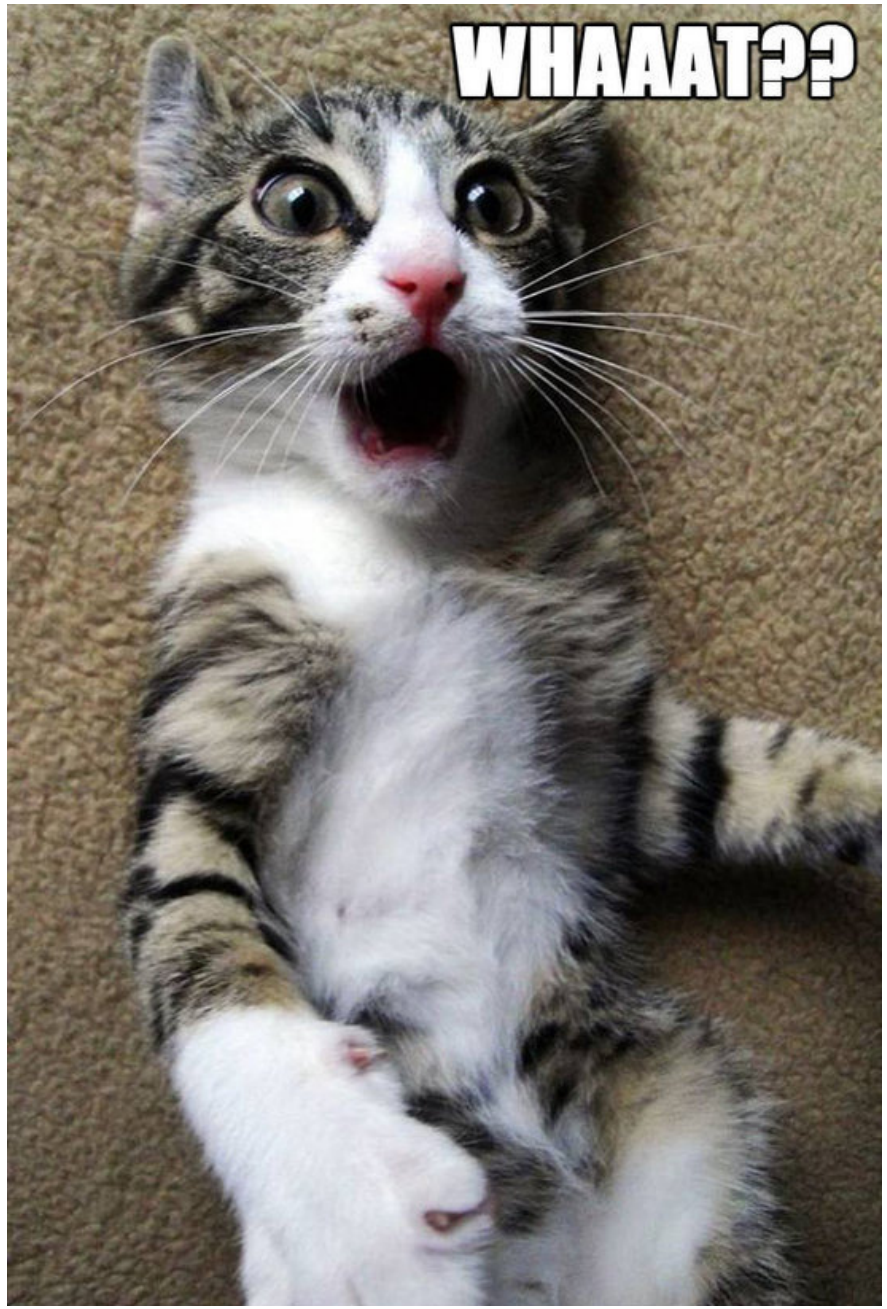
-3.abs()

A.3

B.NoSuchMethodError

C.-3

D.Execution Failure



Groovy AST Browser

Show Script View Help

At end of Phase: Conversion

ClassNode - script1406423965039

Name	Value

Source Bytecode

```
public class script1406423965039 extends groovy.lang.Script {

    public script1406423965039() {
    }

    public script1406423965039(groovy.lang.Binding context) {
        super.setBinding(context)
    }

    public static void main(java.lang.String[] args) {
        org.codehaus.groovy.runtime.InvokerHelper.runScript(scri
    }

    public java.lang.Object run() {
        -(3.abs())
    }

}
```

Unary minus is done last!

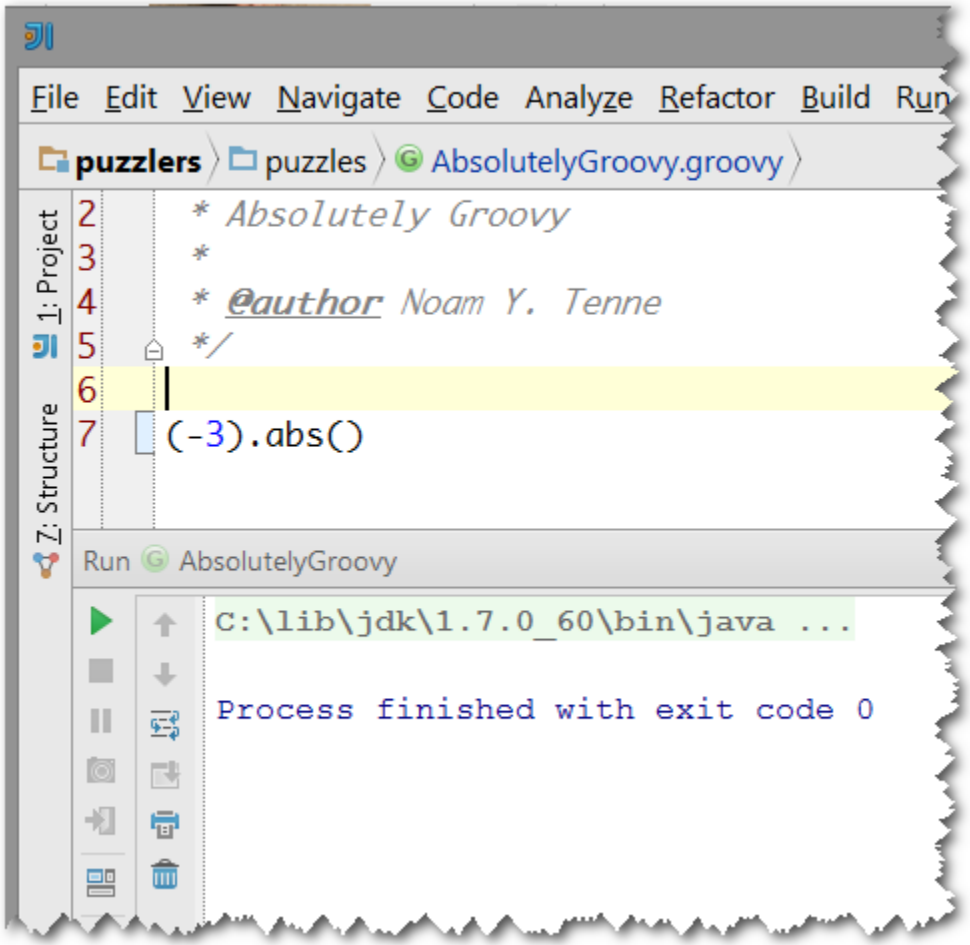
FIX IS SIMPLE

```
(-3).abs()
```

OR

```
int value = -3
value.abs()
```

MY IDEA PRINTS NOTHING WHEN I COPY PASTE THIS CODE. WHAT GIVES?



PRINTLN FOR THE RESCUE!

```
println (-3).abs()
```

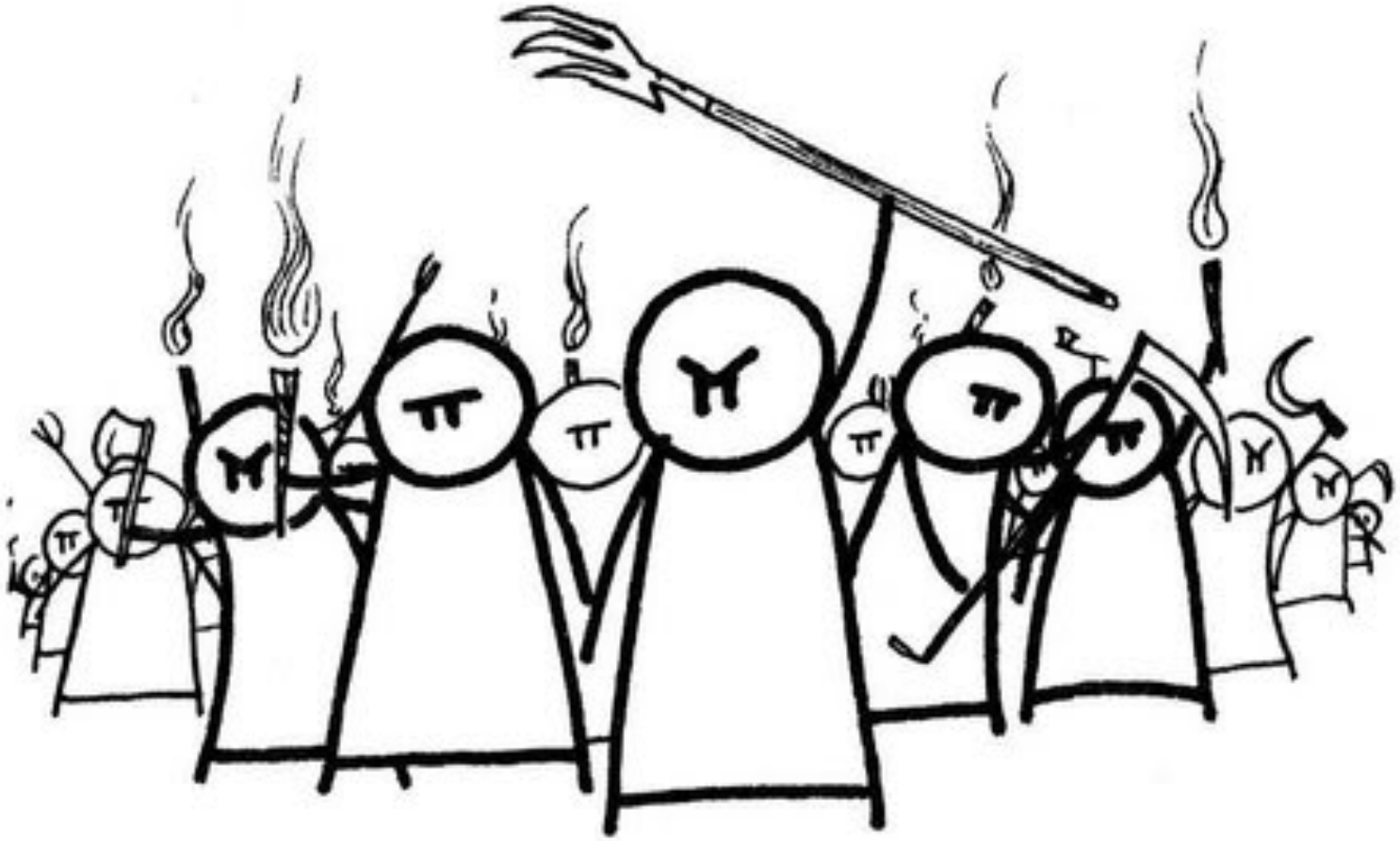
A.-3

B.3

C.3 and NullPointerException

D.-3 and NullPointerException

DELETE ALL PARENTHESES!



COMPILER WILL FIGURE IT OUT!

```
println (-3).abs()
```

```
-3
Caught: java.lang.NullPointerException: Cannot invoke method abs() on null
object
java.lang.NullPointerException: Cannot invoke method abs() on null object
	at AbsolutelyGroovy.run(AbsolutelyGroovy.groovy:7)
	at com.intellij.rt.execution.application.AppMain.main(AppMain.java:134)
```

“All problems in computer science can be solved by another pair of parentheses”

John McCarthy, the inventor of LISP

JUST THROW IN SOME MORE PARENTHESES!

```
println ((-3).abs())
```

OR

```
int value = -3
println value.abs()
```

AND THE T-SHIRT GOES TO...

WHY SO PRIMITIVE?

WHY SO PRIMITIVE?

```
def x = int
println x

if ((x = long)) {
    println x
}

if (x = boolean ) {
    println x
}
```

WHY SO PRIMITIVE?

```
def x = int
println x

if ((x = long)) {
    println x
}

if (x = boolean ) {
    println x
}
```

A. Static Error

B. int, long, boolean

C. java.lang.Integer, java.lang.Long, java.lang.Boolean

THAT WAS EASY. BUT WHY?

That's OK

```
def x = int
println x

//if ((x = long)) {
//    println x
//}
//
//if (x = boolean ) {
//    println x
//}
```

```
WhySoPrimitive
C:\lib\jdk\1.7.0_60\bin\java ...
int

Process finished with exit code 0
```

That's not OK

```
def x = int
//println x
//
//if ((x = long)) {
//    println x
//}
//
if (x = boolean ) {
```

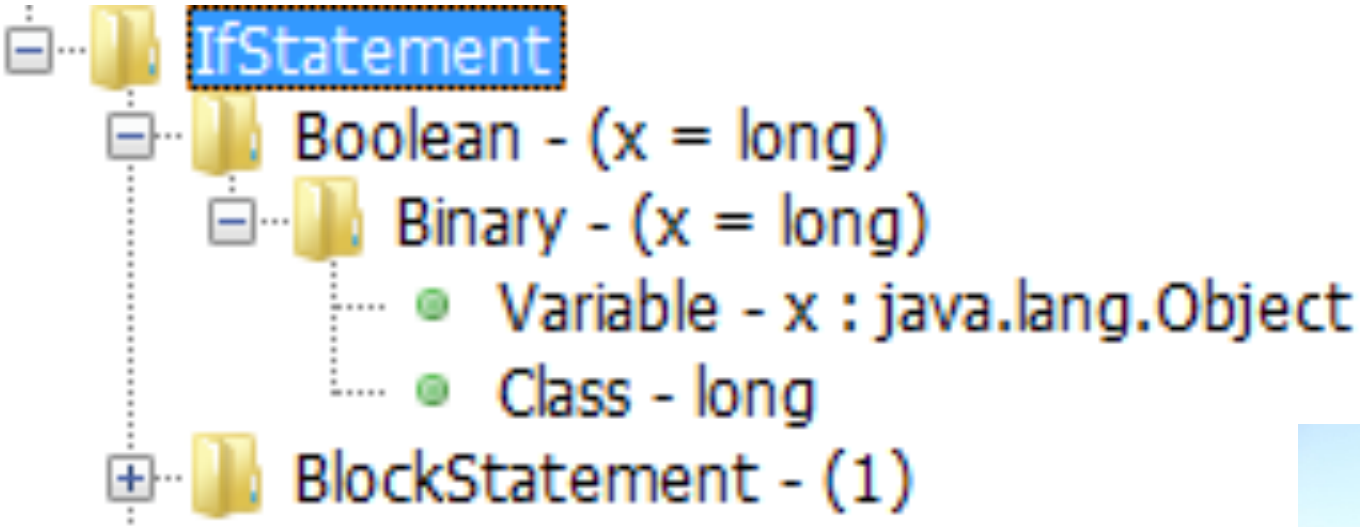
')' expected

Unexpected symbol

HOW ABOUT THAT:

```
def x = int
//println x
//
if ((x = long)) {
    println x
}
//
//if (x = boolean ) {
//    println x
//}
```

(GROOVY) TRUTH CAN HURT
SO JUST PARENTHESES FOR A RESCUE (OR NOT)



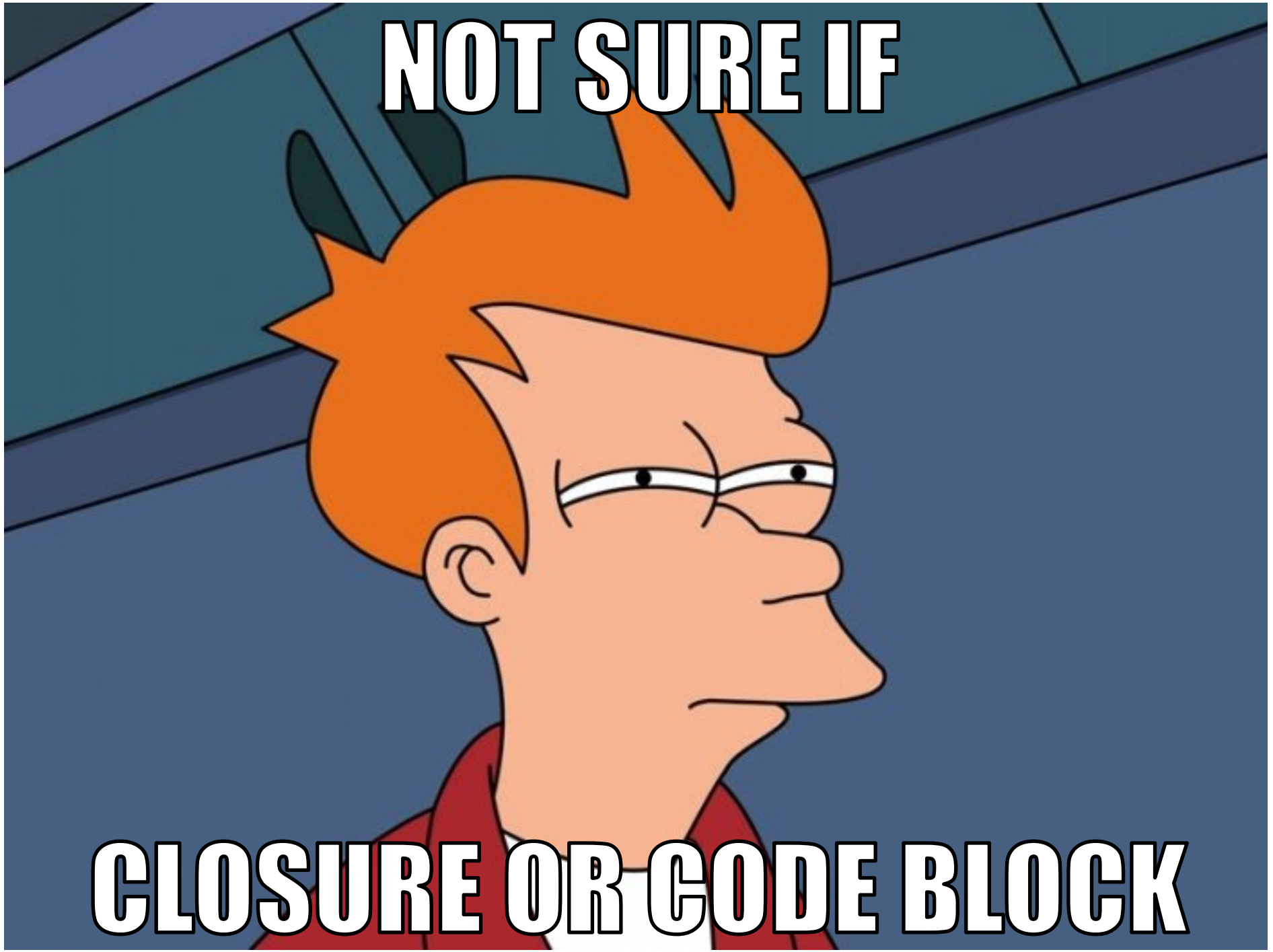
WHODUNIT

WHODUNIT

```
Closure whodunit() {
        {
            'The butler did it.'
        }
    }
    println whodunit()
```

A. NullPointerException
B. whodunit_closure1@xxxxxxx
C. StartupGroovyor
D. The butler did it.

THAT IS VERY SAD.


NOT SURE IF
CLOSURE OR CODE BLOCK

CHEER UP, THE FIX IS EASY!

```
Closure whodunit() {
    { ->
        'The butler did it.'
    }
}
```

AND THE T-SHIRT GOES TO...

PUBLIC -

GET YOUR PIECE OF PUBLIC PROPERTY.

```
trait Public {
   public String property = "I am all public!"

}

class Property implements Public {}

Property publicProperty = new Property()
```

```groovy
trait Public {
   public String property = "I am all public!"

}

class Property implements Public {}

Property publicProperty = new Property()
```

A. publicProperty.@property

B. publicProp  Public__property

C. publicProperty.getProperty()

D. publicProperty.property

TWO QUESTIONS:

1. What's up with the strange name?!
2. How the hell I am supposed to know?!

ANSWER TO 1:

IN SOVIET RUSSIA

YOU DON'T ACCESS PUBLIC PROPERTY

ANSWER TO 2:

The name of the field depends on the fully qualified name of the trait. All dots (`.`) in package are replaced with an underscore (`_`), and the final name includes a double underscore. So if the type of the field is `String`, the name of the package is `my.package`, the name of the trait is `Foo` and the name of the field is `bar`, in the implementing class, the public field will appear as:

```
String my_package_Foo__bar
```

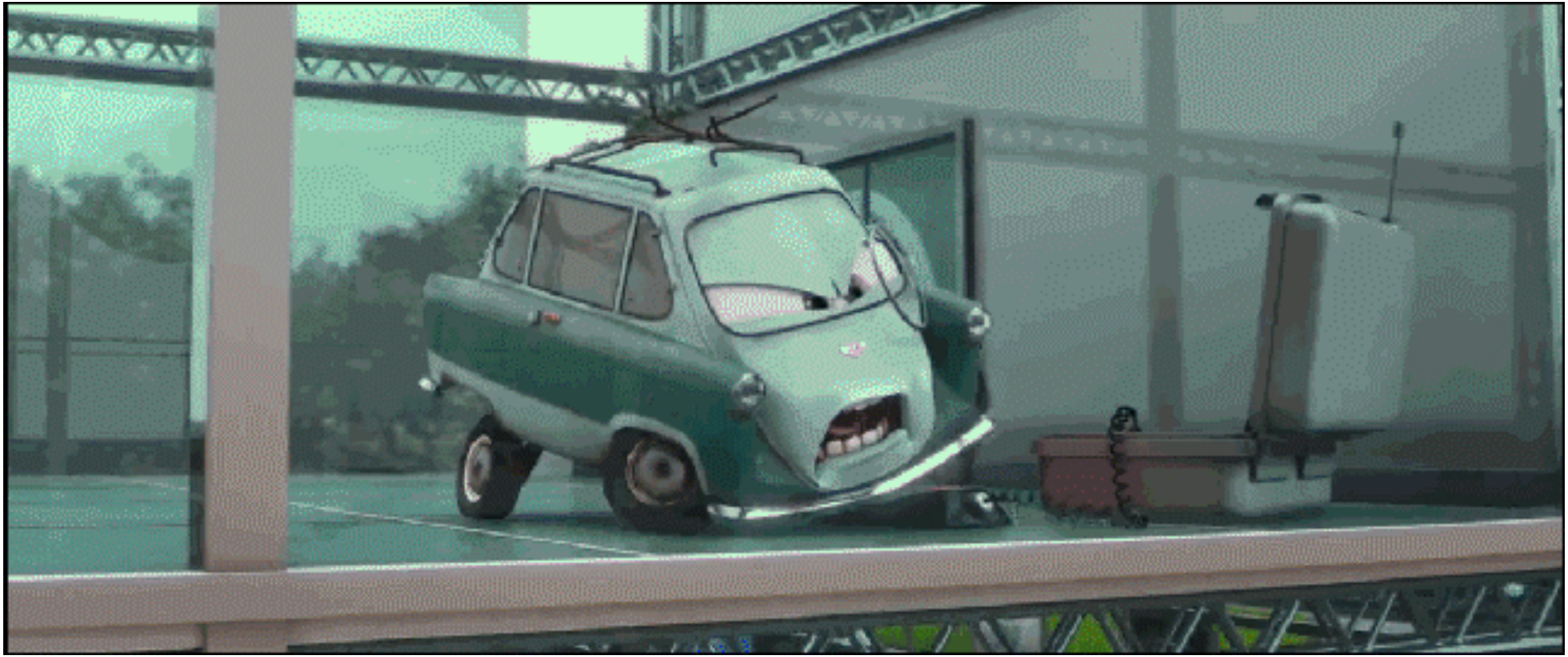
While traits support public fields, it is not recommanded to use them and considered as a bad practice.

http://beta.groovy-lang.org/docs/groovy-2.3.0/html/documentation/core-traits.html

READ THE TRAITS DOCS

IT'S ALL THERE

OUT OF RANGE



OUT OF RANGE

```
def range = 1.0..10.0
assert range.contains(5.0)
println range.contains(5.6)
```

A.Assertion Failed

B.fal 

C.true

D.NullPointerException


But... but...
why?

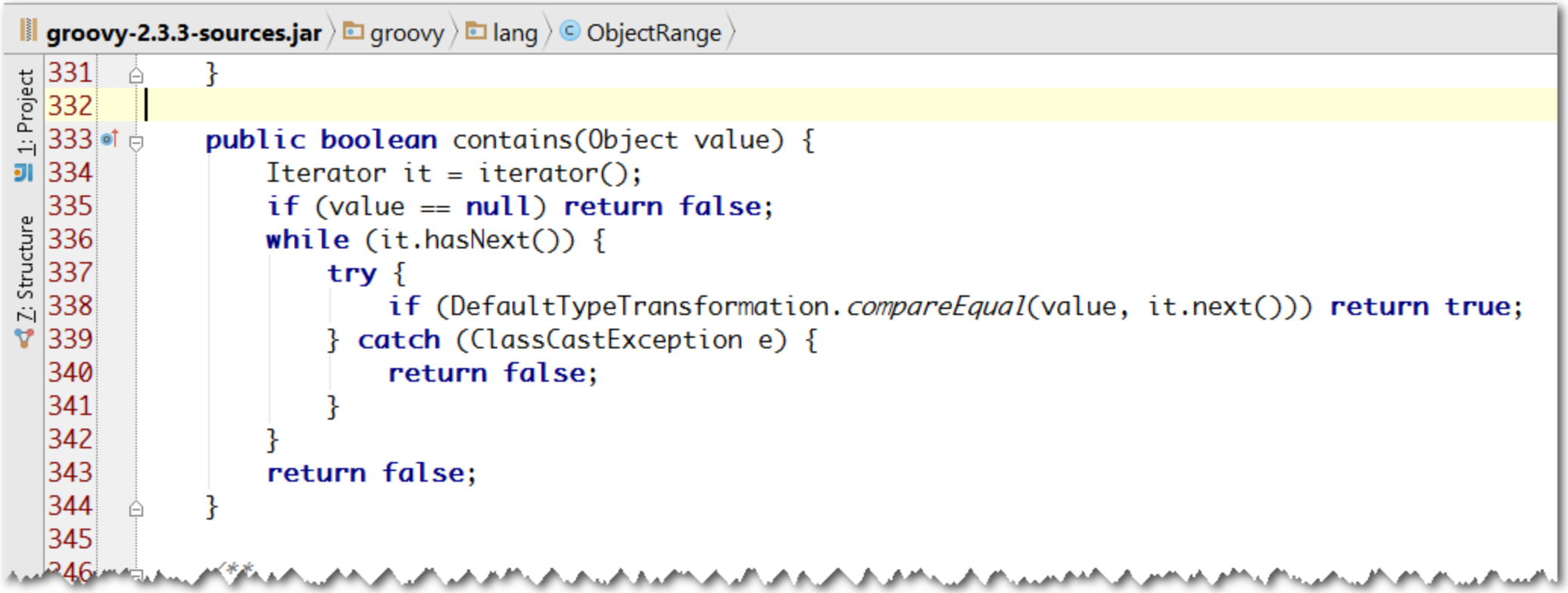
ROAD TO ENLIGHTENMENT, STEP 1:

```
6
7  def range = 1.0..10.0
8
9  println range.class.name
```

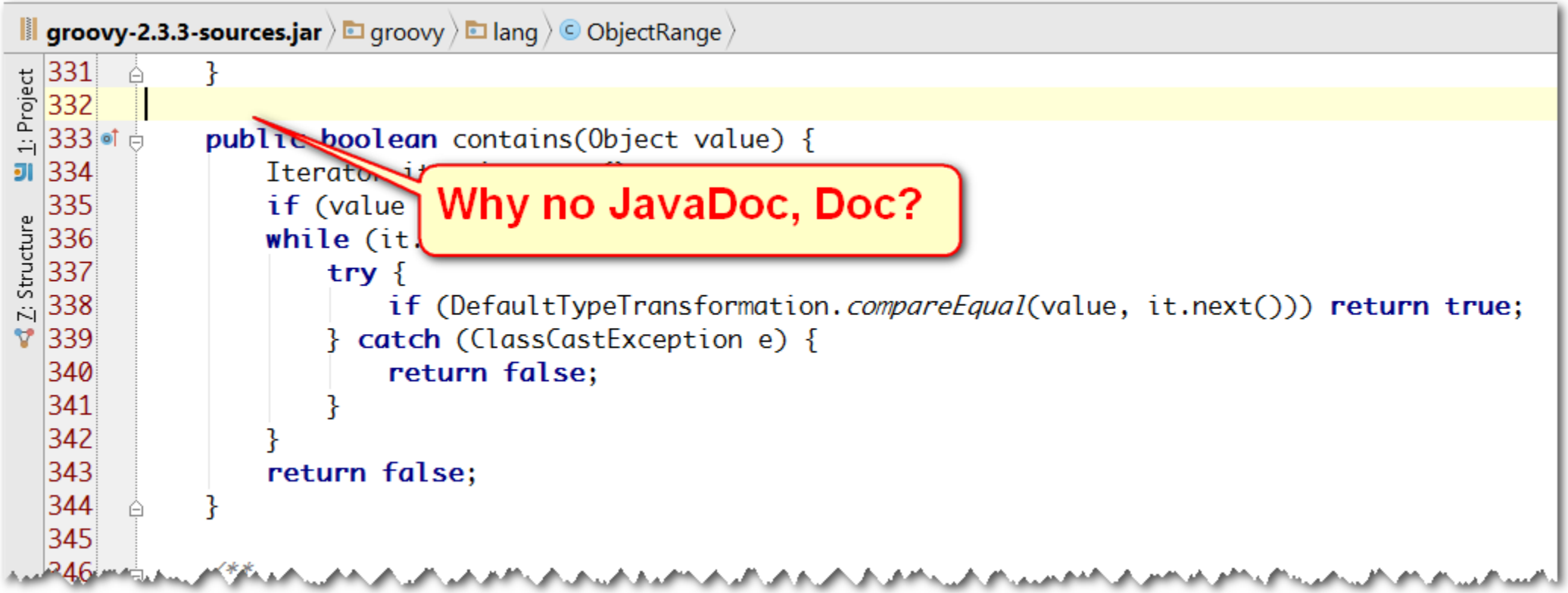
Run OutOfRange

```
C:\lib\jdk\1.7.0_60\bin\java ...
groovy.lang.ObjectRange

Process finished with exit code 0
```

ROAD TO ENLIGHTENMENT, STEP 2:



ROAD TO ENLIGHTENMENT, STEP 2:




```
331     }
332
333     public boolean contains(Object value) {
334         Iterator it
335         if (value
336         while (it.
337             try {
338                 if (DefaultTypeTransformation.compareEqual(value, it.next())) return true;
339             } catch (ClassCastException e) {
340                 return false;
341             }
342         }
343         return false;
344     }
345
346     /**
```

ROAD TO ENLIGHTENMENT, STEP 3:

```
Iterator iterator = (1.0..10.0).iterator()
while (iterator.hasNext()) {
    print "${iterator.next()} "
}
```

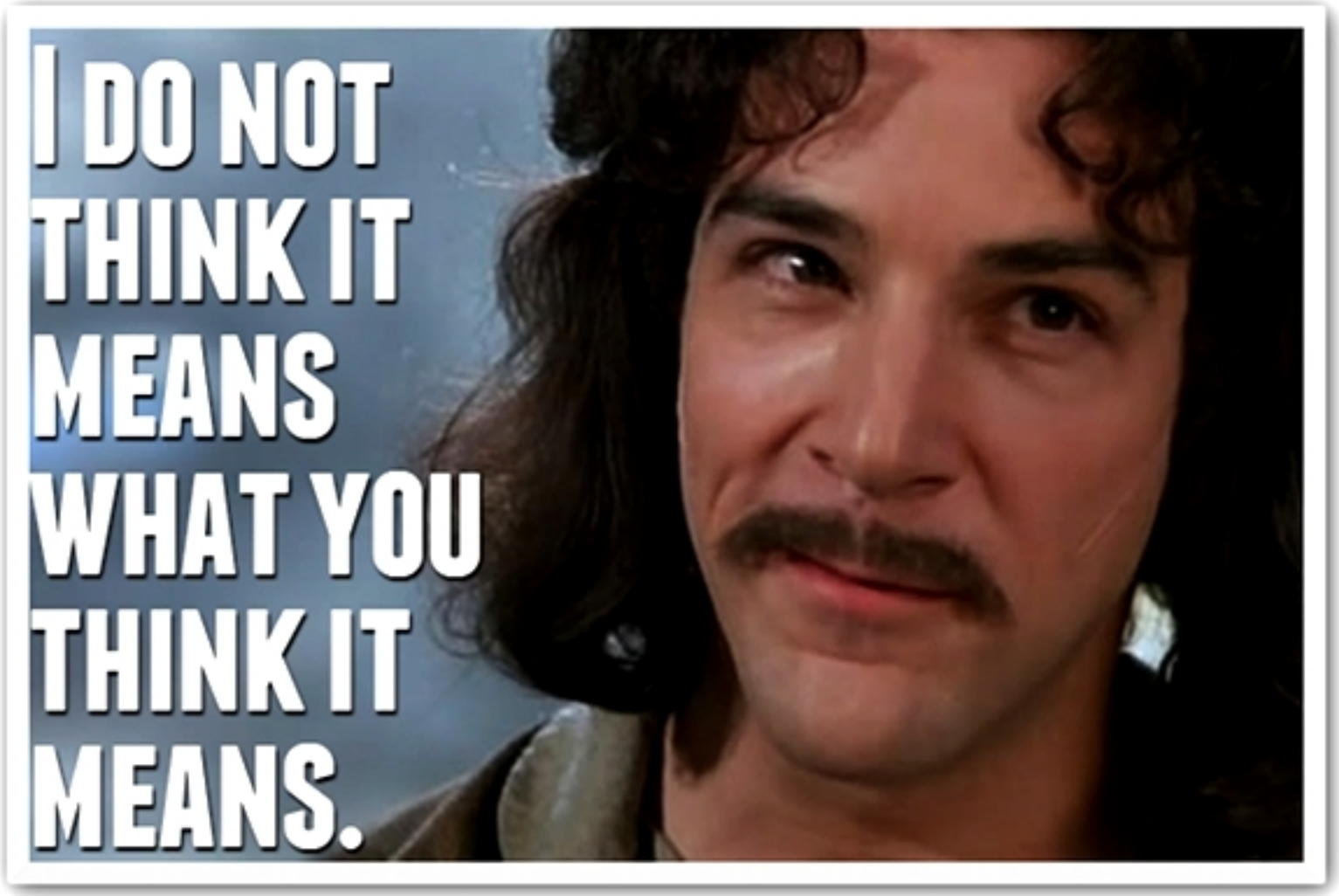
```
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
```

FIX IS SIMPLE, BUT QUESTIONABLE

groovy-2.3.3-sources.jar › groovy › lang › ObjectRange

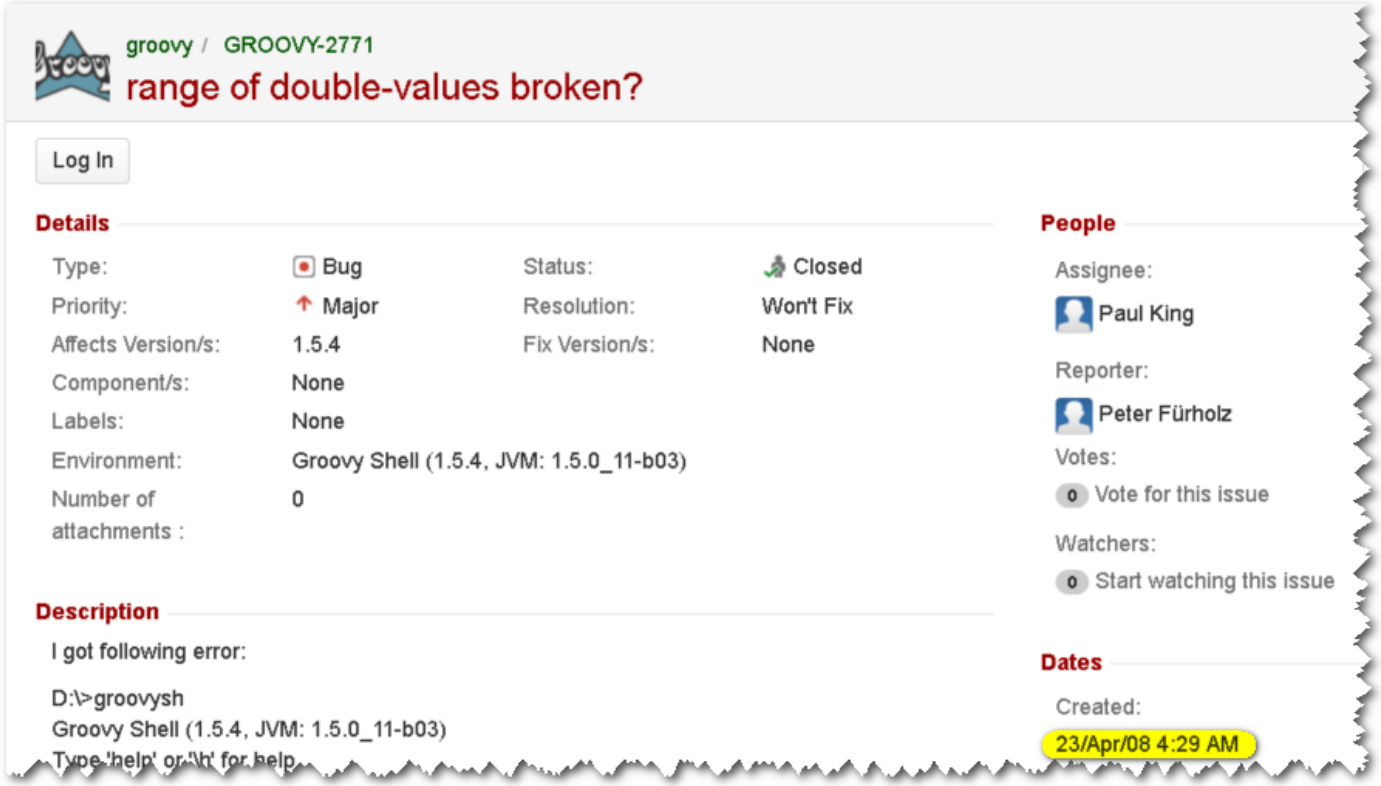
```
241     /**
242      * Checks whether a value is between the from and to values of a Range
243      *
244      * @param value the value of interest
245      * @return true if the value is within the bounds
246      */
247     public boolean containsWithinBounds(Object value) {
248         if (value instanceof Comparable) {
249             int result = compareTo(from, (Comparable) value);
250             return result == 0 || result < 0 && compareTo(to, (Comparable) value) >= 0;
251         }
252         return contains(value);
253     }
254
```

SO, GROOVY, YOU SAY THAT 1.0..10.0 DOES NOT CONTAIN 5.6?



WHAT CAN BE DONE?

- We don't care abo
  implementation cla
- Make it "RealRang
  "ContinuousRange
- In this implementa
  contains to
  containsWithinBounds.
- Fix GROOVY-2771



Thank you.

AND THE T-SHIRT GOES TO...

POWER RANGE(RS)

LET'S TAKE ONE DOWN...

```
[0..9].each { println(it - 1) }
```

A. -1
0
1
2
3
4
5
6
7
8

B. 0
1
2
3
4
5
6
7
8
9

C. [0, 2, 

5, 6, 7, 8, 9]

D. [-1, 0, 1, 2, 3, 4, 5, 6, 7, 8]

I JUST TRICKED YOU. HA!

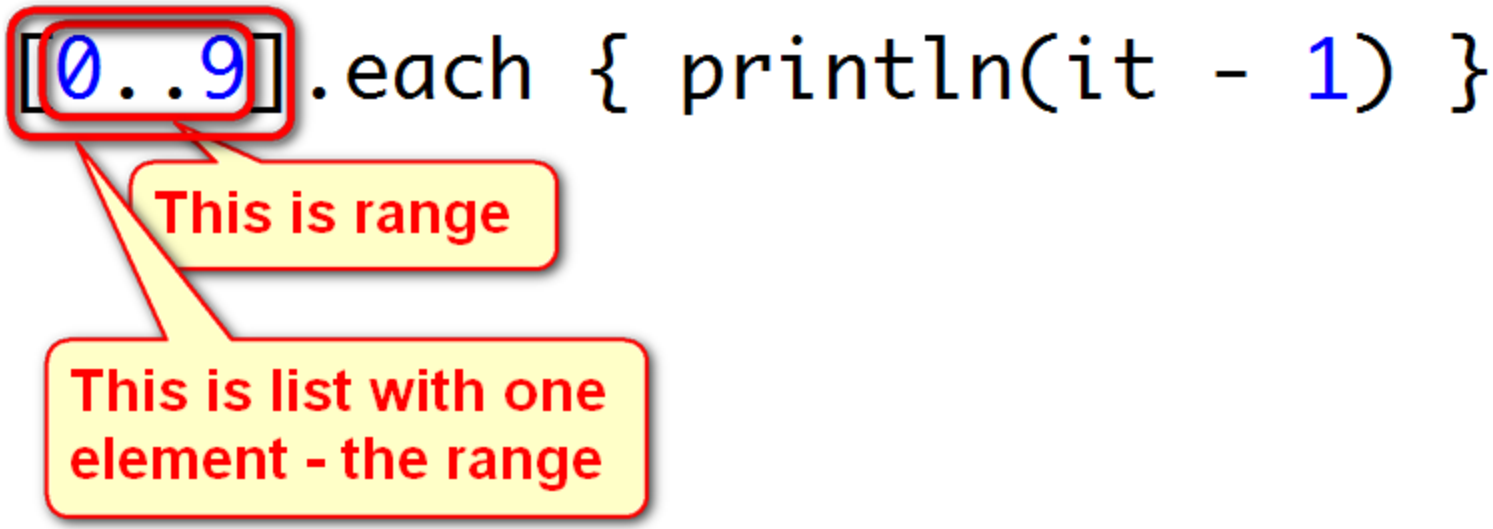
```
[0..9].each { println(it - 1) }
```

I JUST TRICKED YOU. HA!

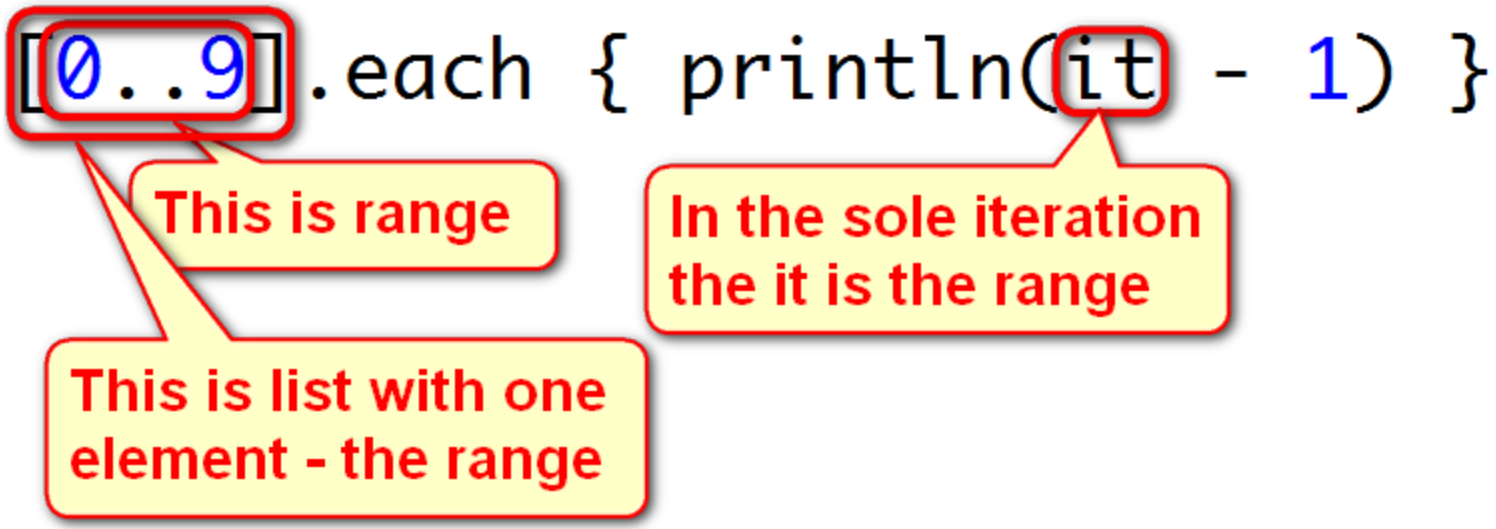
```
[0..9].each { println(it - 1) }
```

This is range

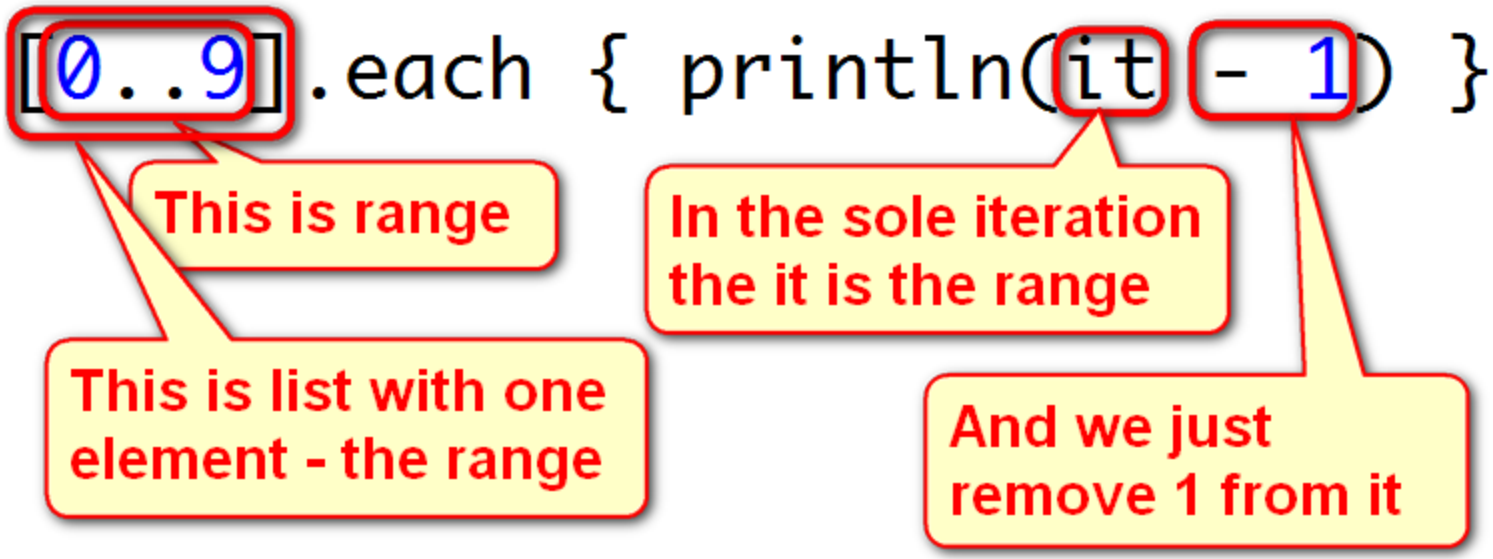
I JUST TRICKED YOU. HA!



I JUST TRICKED YOU. HA!



I JUST TRICKED YOU. HA!



AND THE T-SHIRT GOES TO...

FRESH IN MEMORY!

International Talk Like a Pirate Day

Type	Parodic
Date	September 19
Next time	**19 September 2014**
Frequency	annual

TREAYE MAP, ARRR!

TREAYE MAP, ARRR!

```
def key = 'x'
def map = [key: 'treasure']
def value = map.get(key)
println value
```

```groovy
def key = 'x'
def map = [key: 'treasure']
def value = map.get(key)
println value
```

A.NoSuchElementExceptio
n

B.null

C.treasure

D.I ily-livered swab, BSOD!

BY DEFAULT, KEY IS A LITERAL, NO MATTER WHAT

```
7   def key = 'x'
8   def map = [key: 'treasure']
9   def value = map.get(key)
10  println value
```

See the green?
Green means "String"

FIXES, LOTS OF THEM

```
1.def map = [(key): 'treasure']
2.map.put(key, 'treasure')
3.map."$key" = 'treasure'
4.map[key] = 'treasure'
```

MORE ROUTES T' TO TREAAYE!

A NUMBER OF ROUTES

```
def map = [2: 'treasure']
def key = 2
def value = map."$key"
println value
```

```groovy
def map = [2: 'treasure']
def key = 2
def value = map."$key"
println value
```

A.NoSuchElementException

B.n 

C.treasure

D.Kernel Panic, ye lice-infested

NOW WHAT?!

BY DEFAULT, KEY IS A LITERAL, NO MATTER WHAT, EXCEPT NUMBERS.

```
def map = [2: 'treasure']
println map.keySet().first().class.name
```

```
java.lang.Integer
```

AND THE T-SHIRT GOES TO...


SAVE THE DATE
JAVAONE IS COMING

JAVA2, ANYONE?

```
List<Long> list = [1,2,3]
def now = new Date()
list << now
println list
```

```
List<Long> list = [1,2,3]
def now = new Date()
list << now
println list
```

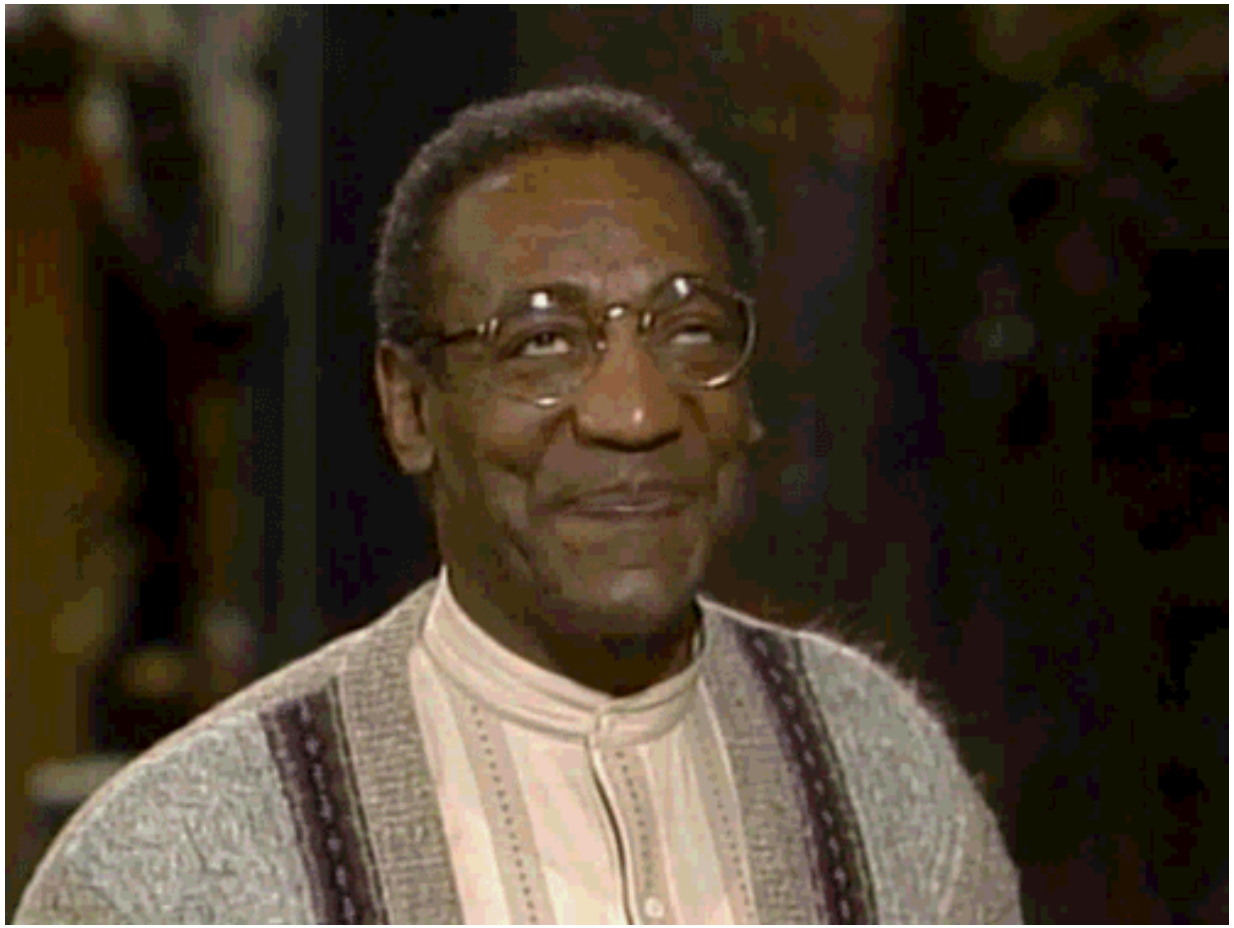
A. Startup Failure

B. [1, 2, 3, WED Oct 10 17:35:11 CDT 2014]

C. [1, 2, 3, 1410388511456]

D. ClassCastException

OH, WAIT, I KNOW WHAT HAPPENED!



ERASURE!

ERASURE!

Type Erasure

Generics were introduced to the Java language to provide tighter type checks at compile time and to support generic programming. To implement generics, the Java compiler applies type erasure to:

- Replace all type parameters in generic types with their bounds or `Object` if the type parameters are unbounded. The produced bytecode, therefore, contains only ordinary classes, interfaces, and methods.
- Insert type casts if necessary to preserve type safety.
- Generate bridge methods to preserve polymorphism in extended generic types.

Type erasure ensures that no new classes are created for parameterized types; consequently, generics incur no runtime overhead.

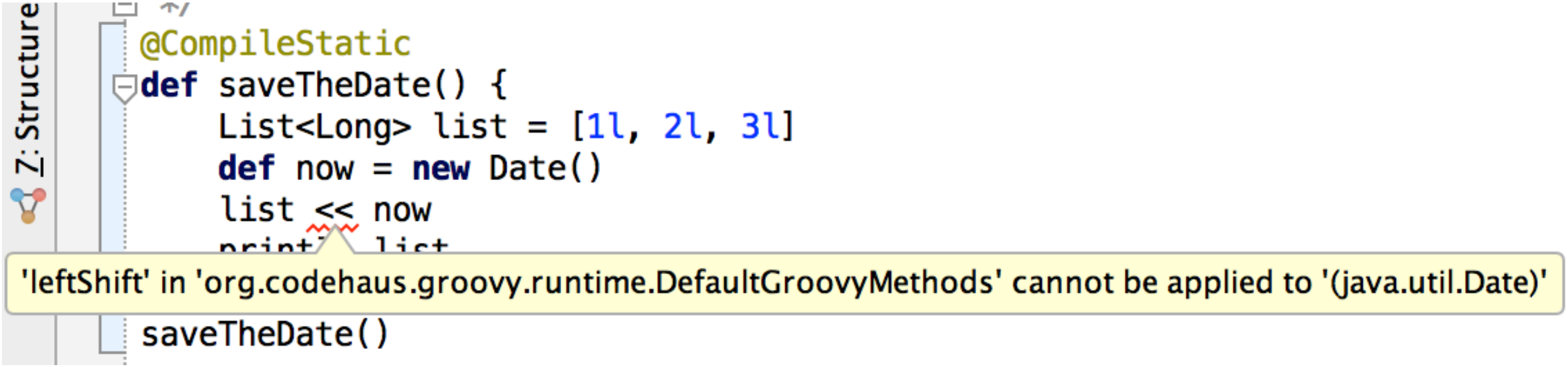

BUT WHERE ARE MY
COMPILE TIME CHECKS?!

“IN GROOVY YOU CAN INSERT ANYTHING ANYWHERE”

```
List<Long> list = [1,2,3]
def now = new Date()
list << now
list << 'foo'
println list*.class.name
```

```
[java.lang.Long, java.lang.Long,
java.lang.Long, java.util.Date,
java.lang.String]
```

COMPILESTATIC!




WHAT'S THE FREQUENCY
KENNETH?
R.E.M.
Compact Disc Maxi-Single

WHAT'S THE METACLASS, BARUCH?

```
def map = [metaClass: 'frequency']
println "What's the $map.metaClass, Baruch?"
```

A. MissingMethodException

B. What's the org.codehaus.groovy.runtime.HandleMetaClass@YYYYYYYY [groovy.lang.MetaClassImpl@YYYYYYYY[class java.util.LinkedHashMap]], Baruch?

C. What's the java.util.LinkedHashMap@XXXXXX.metaClass, Baruch?

D. What's the frequency, Baruch?


IT WORKS, BUT...
NOT SURE IT'S A GOOD THING


REMEMBER JAMES GOSLING TOLD YOU
THAT OPERATOR OVERLOADING IS BAD?

TOO MUCH OPERATOR OVERLOADING!

If `map.metaClass` is overloaded for `map.get('metaClass')`, it can't be overloaded for `map.getMetaClass()` in the same time!


MAYYCE

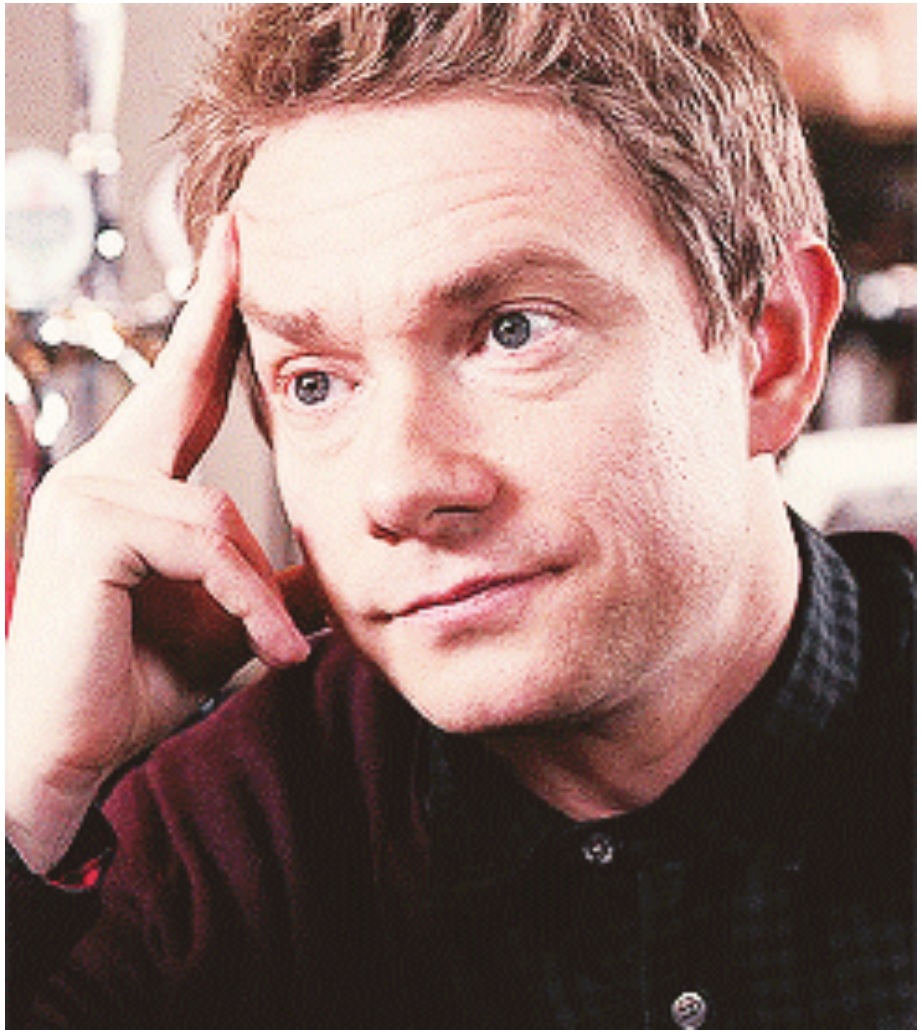
AND THE T-SHIRT GOES TO...

PRIME CUTS

PRIME CUTS

```
boolean isPrime(def x) {
    if (x == 2) return true
    int limit = Math.sqrt(x) + 1
    (2..limit).each {
        if (x % it == 0) {
            return false
        }
    }
    true
}

println isPrime("4" as Double)
```

CONFUSED? SO ARE WE.



LET'S LOOK AT IT AGAIN...

```
boolean isPrime(def x) {
    if (x == 2) return true
    int limit = Math.sqrt(x) + 1
    (2..limit).each {
        if (x % it == 0) {
            return false
        }
    }
    true
}

println isPrime("4" as Double)
```

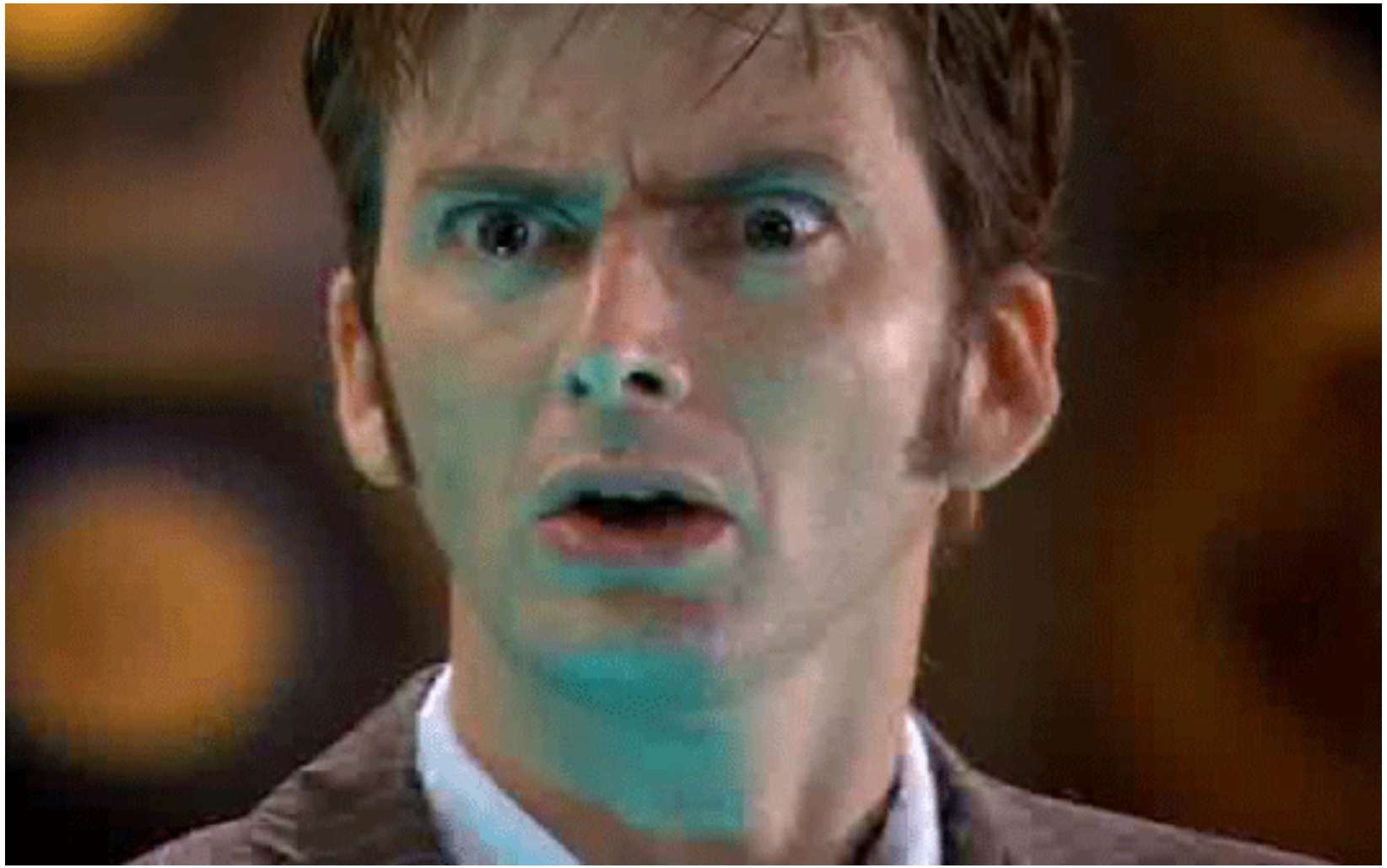
```groovy
boolean isPrime(def x) {
    if (x == 2) return true
    int limit = Math.sqrt(x) + 1
    (2..limit).each {
        if (x % it == 0) {
            return false
        }
    }
    true
}

println isPrime("4" as Double)
```

A.true

B.false

C.NumberFormatException

D.MissingMethodException



LET'S TRY TO FIGURE IT OUT

```
boolean isPrime(def x) {
    if (x == 2) return true
    int limit = Math.sqrt(x) + 1
    (2..limit).each {
        if (x % it == 0) {
            return false
        }
    }
    true
}
```

LET'S TRY TO FIGURE IT OUT

```
boolean isPrime(def x) {
    if (x == 2) return true
    int limit = Math.sqrt(x) + 1
    (2..limit).each {
        if (x % it == 0) {
            return false
        }
    }
    true
}
```

This is closure on every element in range, it has its own scope

LET'S TRY TO FIGURE IT OUT

```
boolean isPrime(def x) {
    if (x == 2) return true
    int limit = Math.sqrt(x) + 1
    (2..limit).each {
        if (x % it == 0) {
            return false
        }
    }
    true
}
```

This returns from the closure, not from method isPrime a.k.a. "local return"

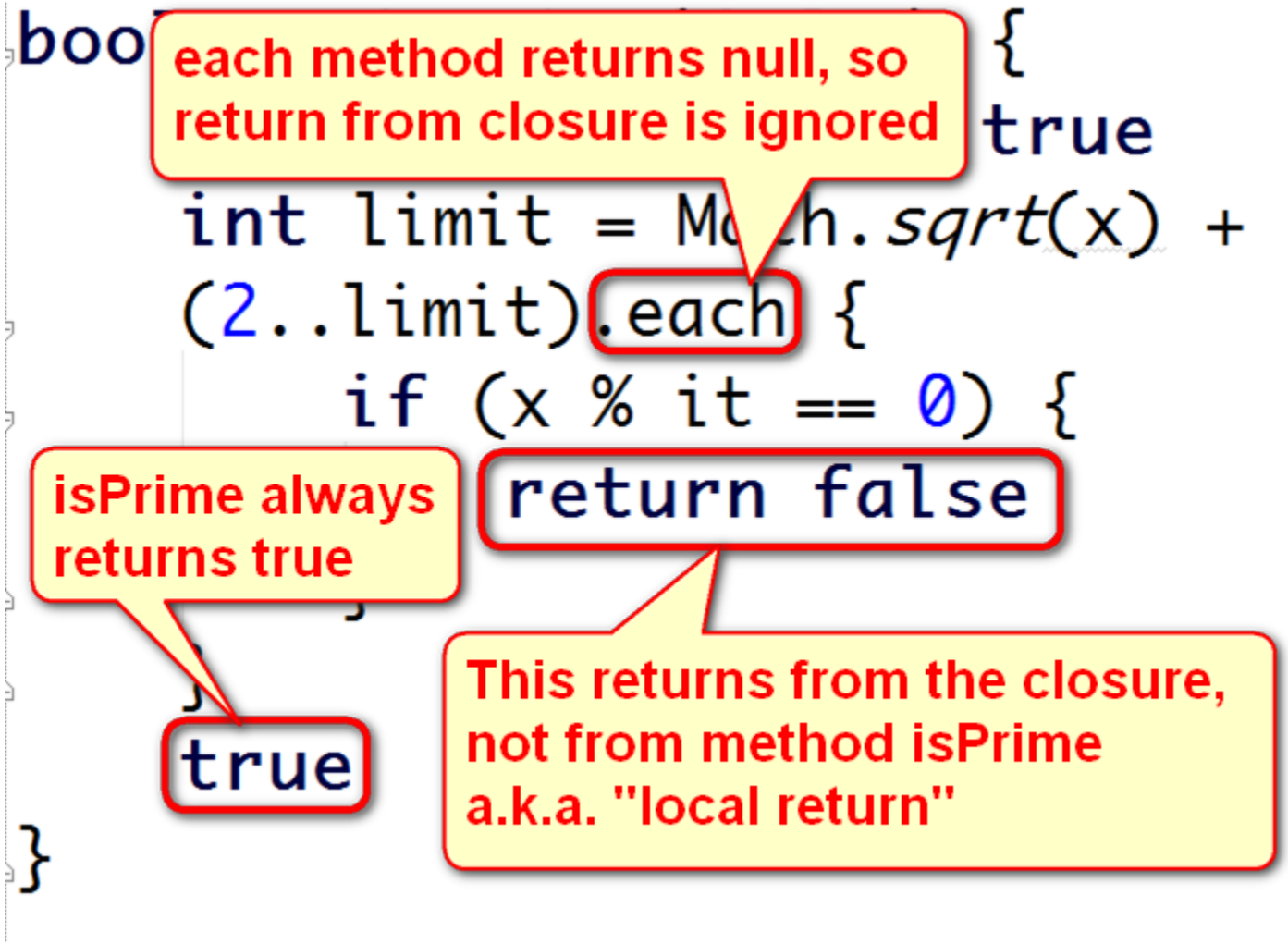
LET'S TRY TO FIGURE IT OUT

```
boo                                    {
    int limit = Math.sqrt(x) + 1                true
    (2..limit).each {
        if (x % it == 0) {
            return false
        }
    }
    true
}
```

each method returns null, so return from closure is ignored

This returns from the closure, not from method isPrime a.k.a. "local return"

LET'S TRY TO FIGURE IT OUT

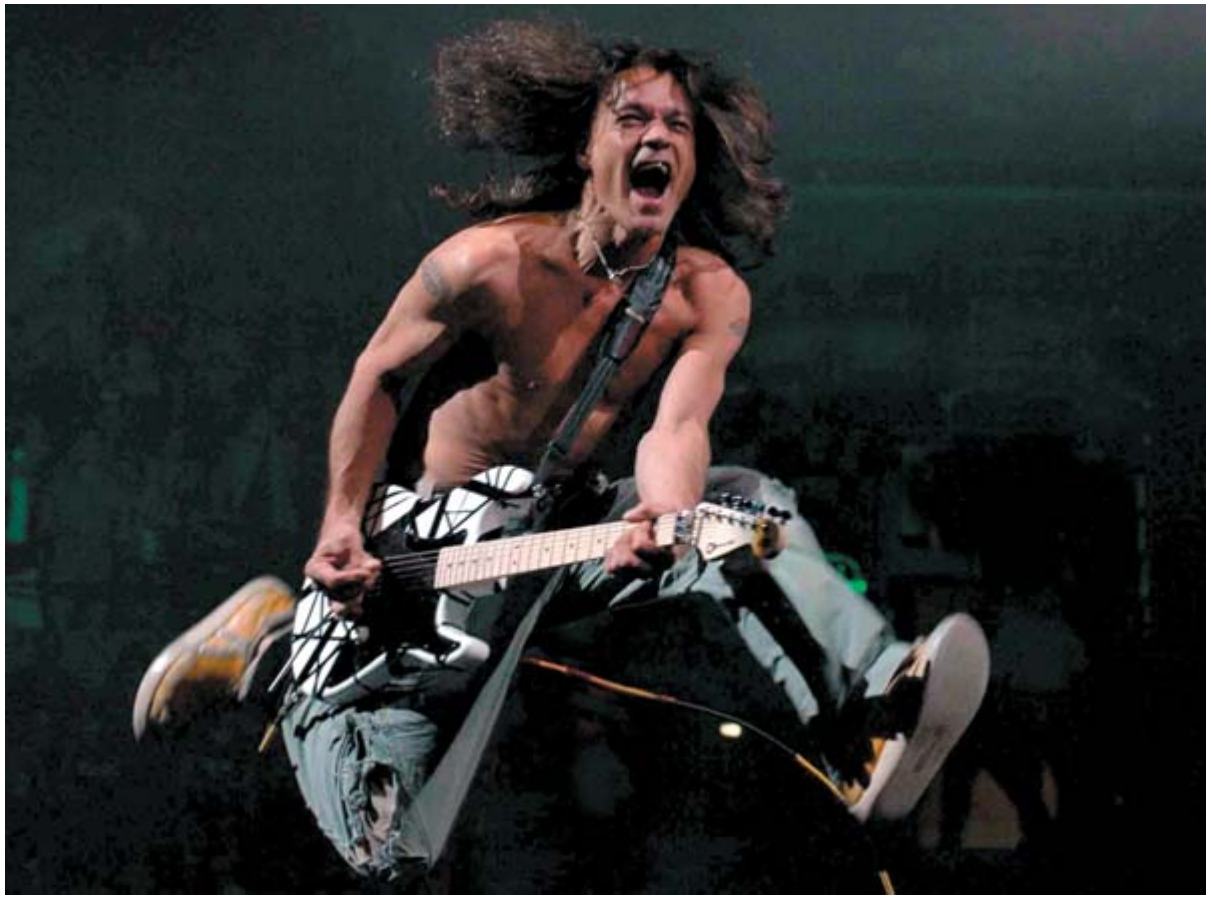


SOLUTION? USE ANY() OR RETHINK THE ALGORITHM

http://kousenit.wordpress.com/2014/04/18/responses-to-the-closure-of-no-return/

AND THE T-SHIRT GOES TO...

JUMP! A.K.A. THE MISSING LYRICS



JUMP! A.K.A. THE MISSING LYRICS

```
class VanHalen {

    public static jump() {
        "Here are the ${lyrics()}"
    }

    def methodMissing(String name, def args) {
        'lyrics'
    }
}

println VanHalen.jump()
```

JUMP! A.K.A. THE MISSING LYRICS

```
class VanHalen {

    public static jump() {
        "Here are the ${lyrics()}"
    }

    def methodMissing(String name, def args) {
        'lyrics'
    }
}

println VanHalen.jump()
```

A. Here are the lyrics

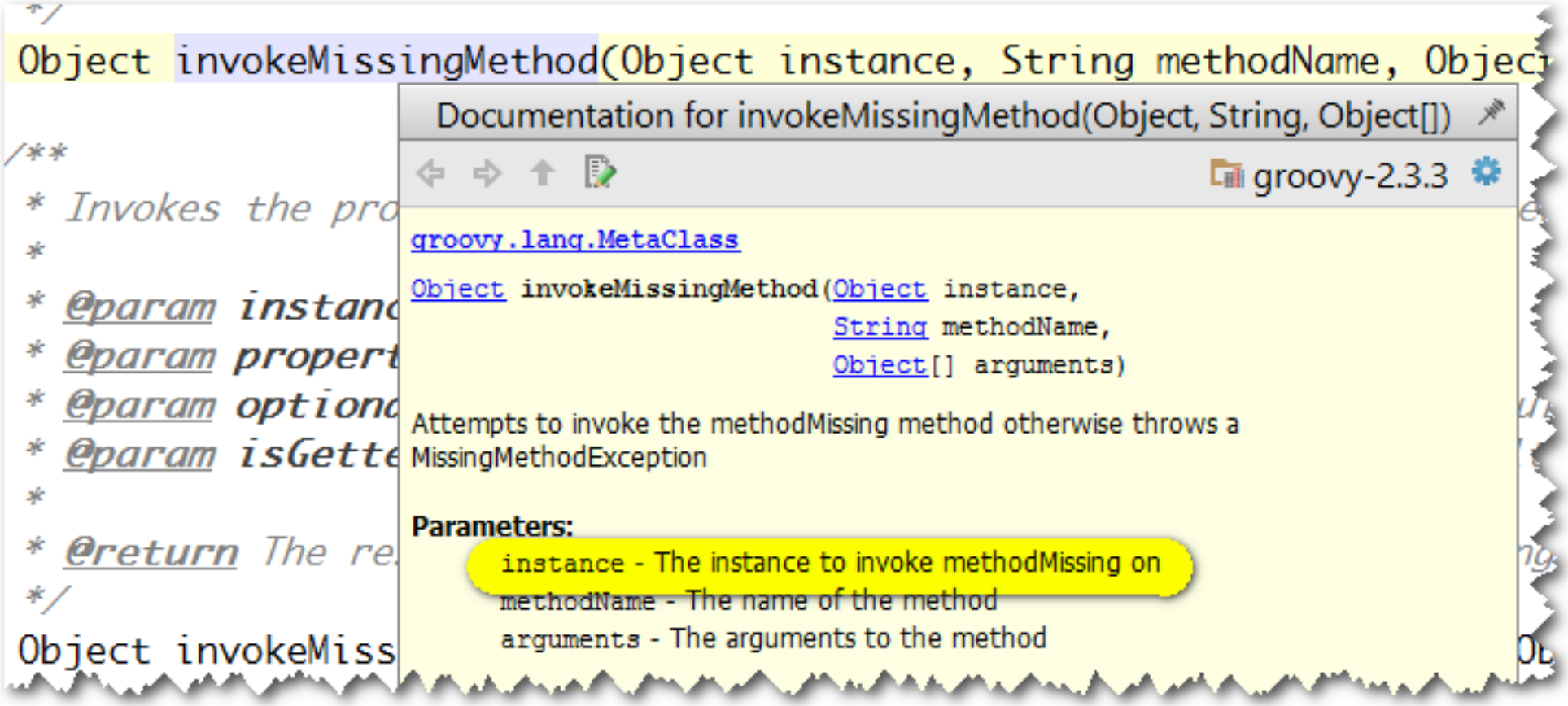
B. Here are the null

C. Startup failure

D. MissingMethodExceptio

I CAN'T BELIEVE YOU FELL FOR THE OLDEST TRICK IN THE BOOK

I CAN'T BELIEVE YOU FELL FOR THE OLDEST TRICK IN THE BOOK



I CAN'T BELIEVE YOU FELL FOR THE OLDEST TRICK IN THE BOOK

```
class VanHalen {

    public static jump() {
        "Here are the ${lyrics()}"
    }

    def methodMissing(String name, def args) {
        'lyrics'
    }
}
```

Can't invoke instance method from static context!

I CAN'T BELIEVE YOU FELL FOR THE OLDEST TRICK IN THE BOOK

```groovy
class VanHalen {

    public static jump() {
        "Here are the ${lyrics()}"
    }

    def methodMissing(String name, def args) {
        'lyrics'
    }
}

println VanHalen.jump()
```

Run TheMissingLyrics

```
C:\lib\jdk\1.7.0_60\bin\java ...
Caught: groovy.lang.MissingMethodException: No signature of method: static VanHalen.lyrics() is applicable for argument types: () values: []
Possible solutions: print(java.io.PrintWriter), print(java.lang.Object), is(java.lang.Object), notify(), wait(), grep()
groovy.lang.MissingMethodException: No signature of method: static VanHalen.lyrics() is applicable for argument types: () values: []
Possible solutions: print(java.io.PrintWriter), print(java.lang.Object), is(java.lang.Object), notify(), wait(), grep()
    at VanHalen.jump(TheMissingLyrics.groovy:11)
    at VanHalen$jump.call(Unknown Source)
    at TheMissingLyrics.run(TheMissingLyrics.groovy:19) <1 internal calls>
```

SOLUTION: STATIC METHODMISSING

```
class VanHalen {

    public static jump() {
        "Here are the ${lyrics()}"
    }

    static $static_methodMissing(String name, def args) {
        'lyrics'
    }
}

println VanHalen.jump()
```

BETTER SOLUTION: USE GOOD OOP

```
class VanHalen {

    public jump() {
        "Here are the ${lyrics()}"
    }

    def methodMissing(String name, def args) {
        'lyrics'
    }
}

println new VanHalen().jump()
```

π – THE FOOD AND THE NUMBER

DOUBLE THE PI

```
double value = 3
println "$value.14".isDouble()
```

A. true

B. MissingPropertyException

C. false

D. MissingMethodException

GOT THIS ONE!

PIECE OF PIE!

```groovy
class Person{ String name }
Person person = new Person(name: 'Andres')
println "The name is $person.name"

double value = 3
println "$value.14".isDouble()
```

Refers to property

PIECE OF PIE!

```groovy
class Person{ String name }
Person person = new Person(name: 'Andres')
println "The name is $person.name"

double value = 3
println "$value.14".isDouble()
```

Refers to property

Refers to property `14` of double? WTF!

DOUBLE THE PI

```groovy
double value = 3
println "$value.14".isDouble()
```

A. true

B. MissingPropertyException

C. fa 

D. MissingMethodException


ESPN
CELTICS
abc
00:20
03:45

IF IT CAN'T BE A PROPERTY, WHY EVEN BOTHER?

```
5806        public
5807            ret
5808        }
5809
5810        /**
```

```
tart(char ch) {
);
```

```
class Perso
Person perso
println "The
```

```
'Andres')
me"
```

```
double valu
println "$v
```

can be property

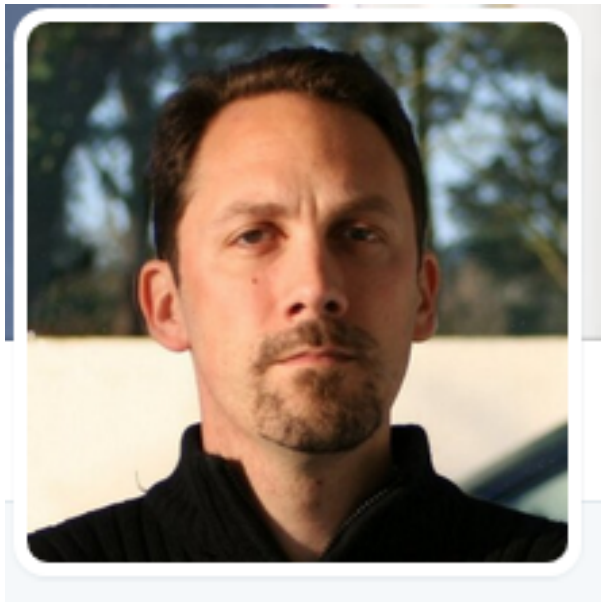
MIND THE TOSTRING() OF A DOUBLE

```
7
8    double value = 3
9    println value
10   println "$value.14"
```

Run DoubleThePi

```
C:\lib\jdk\1.7.0_60\bin\java ...
3.0
3.0.14
```

EVER WONDERED WHAT RSVP STANDS FOR?



ASK THE FRENCH GUYS!

RSVP

```
class Invite {
    int attending = 1
}

def invite = new Invite()
def attendees = (invite.attending) +1
println attendees
```

```groovy
class Invite {
    int attending = 1
}

def invite = new Invite()
def attendees = (invite.attending) +1
println attendees
```

A. StartupFailure

B. 1

C. 2

D. MissingPropertyException

```groovy
class Invite {
    int attending = 1
}

def invite = new Invite()
def attendees = (invite.attending) +1
println attendees
```

WAT?!

Rsvp

```
C:\lib\jdk\1.7.0_60\bin\java ...
org.codehaus.groovy.control.MultipleCompilationErrorsException: startup failed:
C:\Users\jbaruch\projects\puzzlers\puzzles\Rsvp.groovy: 12: unable to resolve class invite.attending
 @ line 12, column 17.
   def attendees = (invite.attending) +1
                   ^

1 error
```

THAT'S WHY:

```
def invite = new Invite()
def attendees = (invite.attending) +1
println attendees
```

This is class cast!

Package name

Class name

FIX: WHATEVER DOESN'T LOOK LIKE CASTING.

THIS WORKS:

```
def attendees = (new Invite().attending) + 1
println attendees
```

THIS WORKS AS WELL:

```
def invite = new Invite()
println (invite.attending +1)
```

AND THE T-SHIRT GOES TO....

French guy, naturally.

He's the one to blame, BTW.

CONCLUSIONS

1. Write readable code
2. Comment neat tricks
3. Sometimes it is just a
4. Use static code analy
5. Rtfm

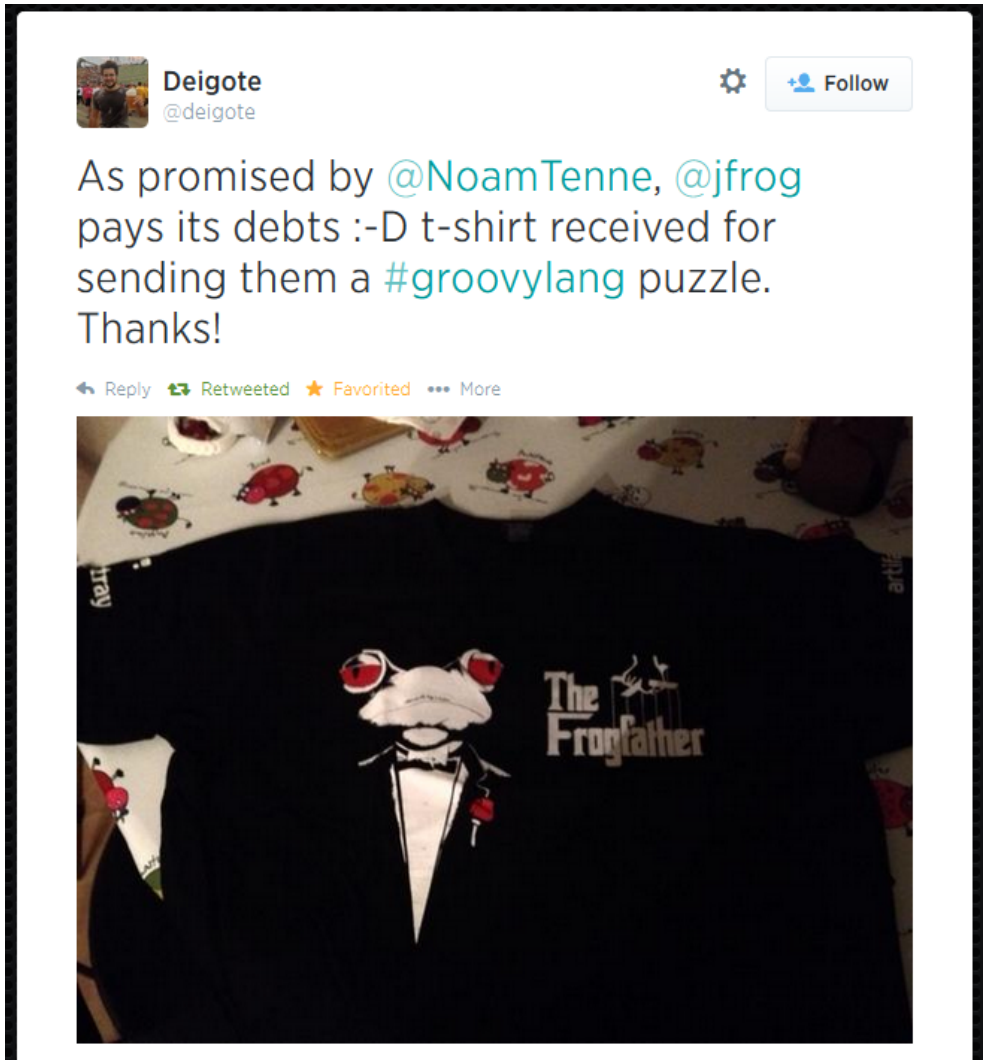
6. Don’t code like my brother

We have just started!
(may end up in proper uniform)

Puzzlers? Gotchas?

- puzzlers@jfrog.com
- Groovypuzzlers@gmail.com

JFROG ALWAYS PAYS ITS DEBTS



Positive feedback?
Praise us on twitter

- #groovypuzzlers
- @Groovypuzzlers
- @_yoav_
- @jbaruch

Negative feedback?
> /dev/null

NO, THANK YOU!