

Joker<?>
GROOVY
PUZZLERS

СТРАННОЕ,
НЕПОНЯТНОЕ И
МАМАЮЩАЯ ЛЕТО СТО

CLICK AND HACK



THE TYPING BROTHERS

1. Два клЕвых пацана на сцене
 2. Прикольные загадки
 3. Вы голосуете за правильный
ответ
 4. Мы швыряемся вещами
 5. Официальный хэш!
groovypuzzlers
- #

ПЕРВОЕ ПРАВИЛО ПАЗЗЛЕРОВ:



НЕ ЧИТИТЬ!



ВСЁ ВЕРНО ДЛЯ ВЕРСИИ
2.3.7

НА НАС ИДУТ

**РЫЖИЕ!
СОБАКИ!**

**МЫ
ПРИНИМАЕМ БОЙ!**



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

Name	Value
ClassNode - script1406423965039	

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!

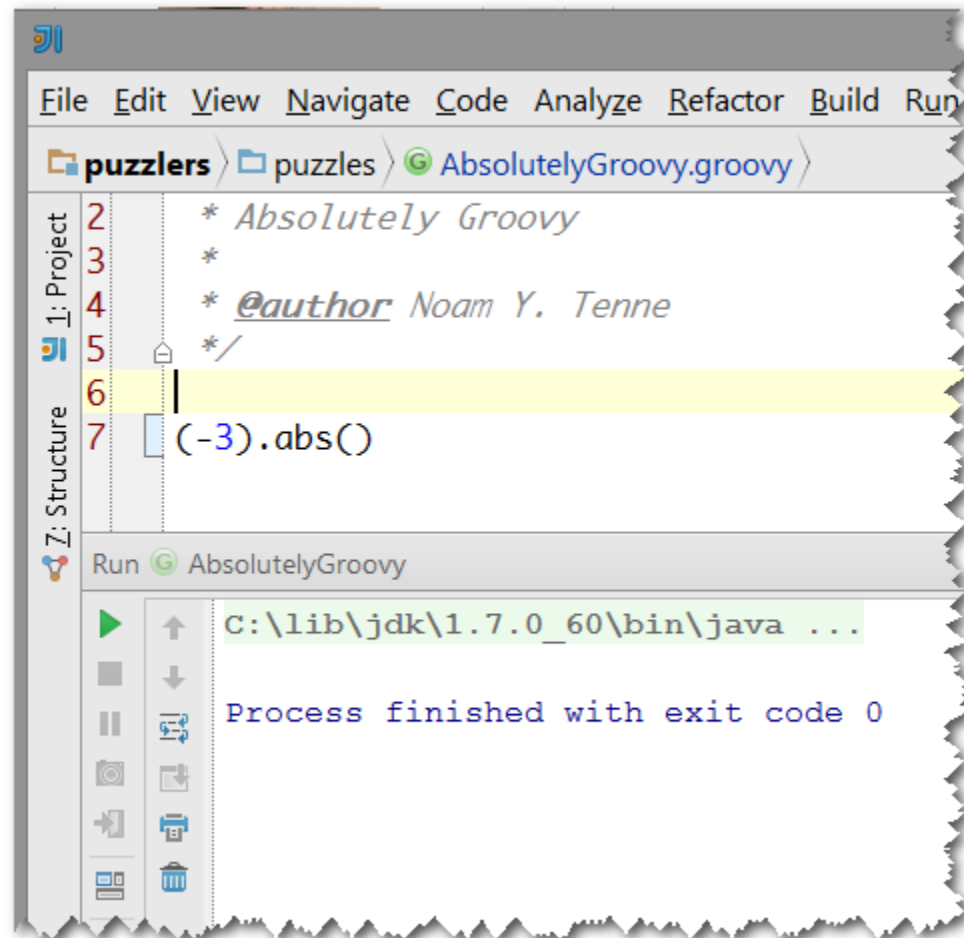
ЧИНИМ

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

ИЛИ

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

ИДЕЯ НИЧЕГО НЕ ПЕЧАТАЕТ. ПОЧЕМУ?



ДОБАВИМ PRINTLN!

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

A.-3

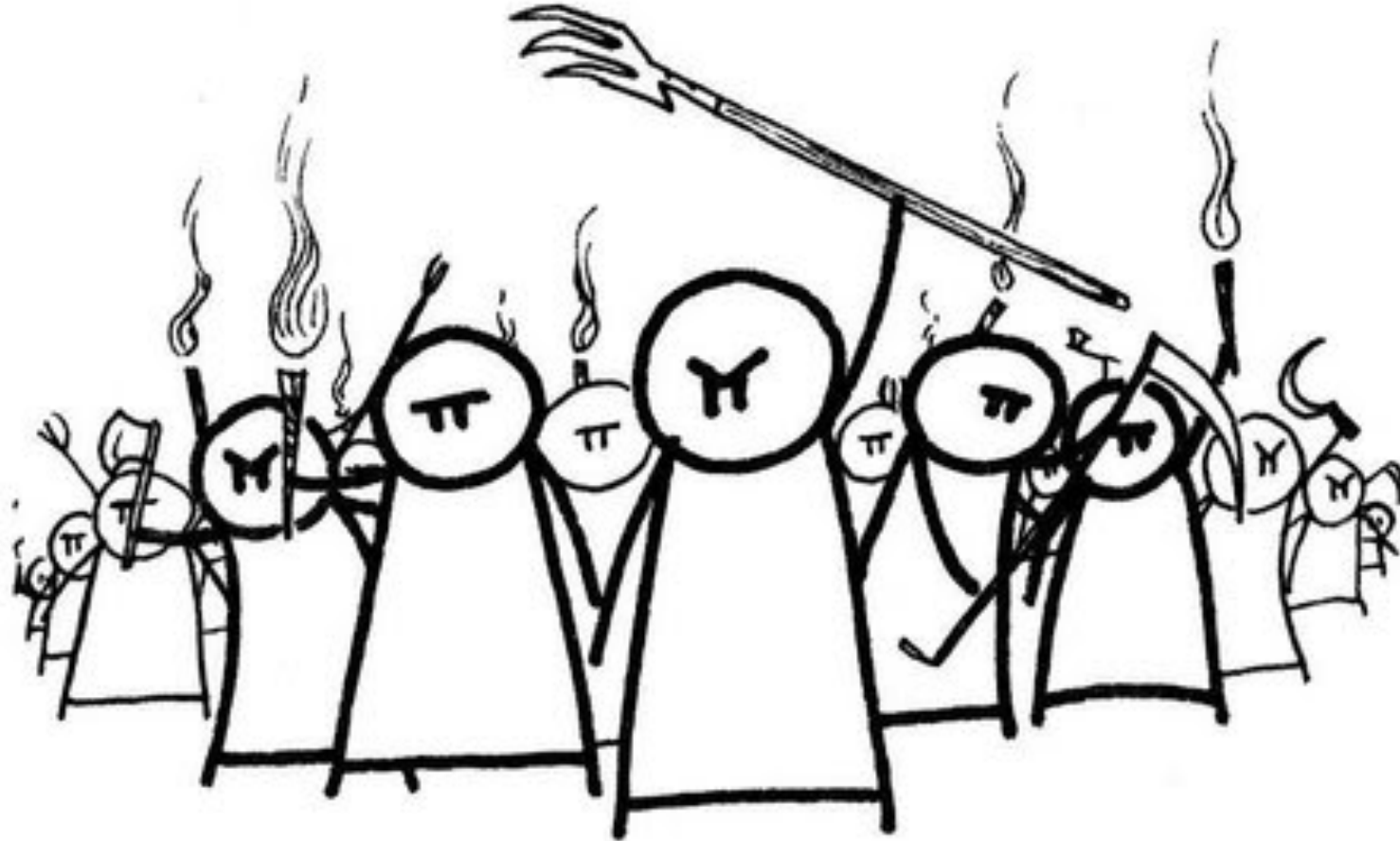
B.3

C.3 and NullPointerException

D.-3 and  NullPointerException



УБЕЙТЕ ВСЕ СКОБКИ!



КОМПИЛЯТОР РАЗБЕРЕТСЯ!

println (-3)

-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)

*“Все проблемы в программировании
можно решить добавив пару скобок”*



John McCarthy, изобретатель LISP

ПРОСТО ДОБАВЬ СКОБОК!

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

ИЛИ

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

PRIME CUTS



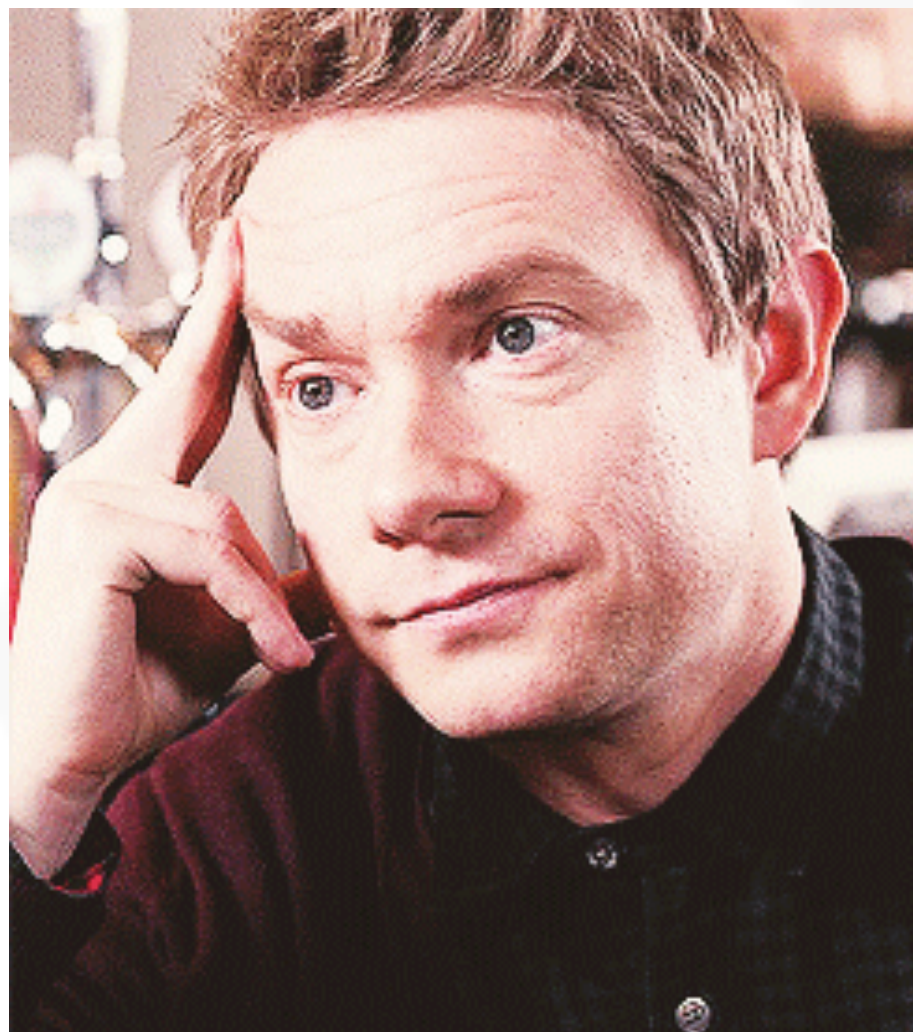
HEY VEGETARIANS

Explain these!

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)
```

CAM В ШОКЕ



ПОСМОТРИМ ЕЩЕ РАЗ

```
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)
```

```
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

КАК 4 МОГЛО ОКАЗАТЬСЯ ПРОСТЫМ
ЧИСЛОМ...



ДАВАЙТЕ РАЗБИРАТЬСЯ

```
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  
}
```


ДАВАЙТЕ РАЗБИРАТЬСЯ

```
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

ДАВАЙТЕ РАЗБИРАТЬСЯ

```
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"

ДАВАЙТЕ РАЗБИРАТЬСЯ

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

each method returns null, so
return from closure is ignored

return false

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

ДАВАЙТЕ РАЗБИРАТЬСЯ

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

each method returns null, so
return from closure is ignored

isPrime always
returns true

return false

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

true

РЕШЕНИЕ: ИСПОЛЬЗУЙ ANY()
ИЛИ ПЕРЕОСМЫСЛИ АЛГОРИТМ



<http://kousenit.wordpress.com/2014/04/18/responses-to-the-closure->

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. Conf@XX
XXXX



C. Startup failure

D. Greach
2014

КАКОЙ ЖЕ ЭТО ITERABLE?



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);  
    }  
}
```

MAX POWER



MAX POWER

```
List<Integer> list = [56, '9', 74]
def max = list.max { item ->
    (item < 50) ? item : null
}
println max
```



```
List<Integer> list = [56, '9', 74]
def max = list.max { item ->
    (item < 50) ? item : null
}
println max
```

A. ClassCastException

B. 9

C. 56

D. null

РАСКУСИЛ!



```
>groovysh (('9' as Character) as Integer)  
==> 57
```

```
List<Integer> list = [56, 57, 74]
def max = list.max { item ->
    (item < 50) ? item : null
}
println max
```

СЕКУНДОЧКУ, ТЕПЕРЬ ОНИ
ВСЕ NULL?



```
List<Integer> list = [56, '9', 74]
def max = list.max { item ->
    (item < 50) ? item : null
}
println max
```

A. ClassCastException

B. 9

C. 50 

D. null



КАК ЭТО ПОЛУЧИЛОСЬ?!

```
def random = new Random()
```

```
def randomList = []
```

```
0..10.each {randomList << random.nextInt()}
```

```
assert randomList.max{ null } == randomList[0]
```

ПЛОХОЕ ЗНАЧЕНИЕ ПО
УМОЛЧАНИЮ, НИГДЕ НЕ
ОПИСАНО



ДЕТЕКТИВ



КТО УБИЙЦА?

```
Closure ктоУбийца() {  
    {  
        'Мориарти.'  
    }  
}  
println ктоУбийца()
```

A. NullPointerException

C. StackError 

B. ктоУбийца!_closure1@x

D. Мориарти.

xxxxxxx

ПИЧАЛЬКА



```
Closure ктоУбийца () {  
    {  
        'Мориарти'  
    }  
}
```

Ambiguous code block

ПУСТЯКИ, ДЕЛО ТО ЖИТЕЙСКОЕ

```
Closure ктоУбийца() {
```

```
  return {
```

```
    'Мориарти.'
```

```
  }
```

```
}
```

```
Closure ктоУбийца() {
```

```
  { ->
```

```
    'Мориарти.'
```

```
  }
```

```
}
```



Не удается отобразить рисунок. Возможно, рисунок поврежден или недостаточно памяти для его открытия. Перезагрузите компьютер, а затем снова откройте файл. Если вместо рисунка все еще отображается красный крестик, попробуйте удалить рисунок и вставить его заново.

EUROPE



THE FINAL COUNTDOWN

```
class Countdown { int counter = 10 }
```

```
Countdown finalCountDown() {  
    def countdown = new Countdown()  
    try {  
        countdown.counter = --countdown.counter  
    } catch (ignored) {  
        println "That will never happen."  
        countdown.counter = Integer.MIN_VALUE  
    } finally {  
        return countdown  
    }  
}
```

```
println finalCountDown().counter
```

A. Startup failure

B. That will never happen.

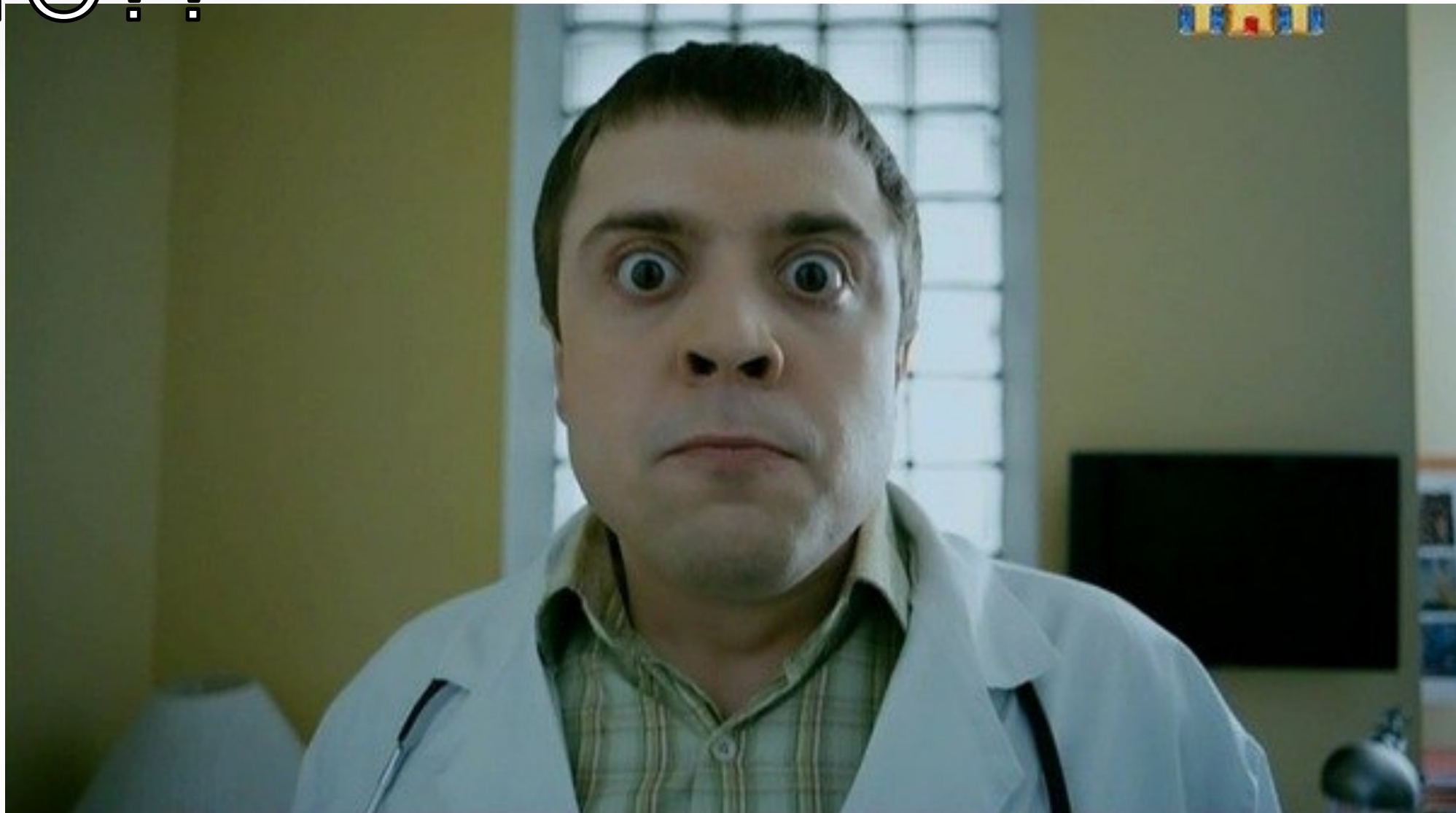
-2147483648  10s CountDown { **int** counter = 10 }

C. 9

D. 10

```
CountDown finalCountDown() {  
    def countDown = new CountDown()  
    try {  
        countDown.counter = --countDown.counter  
    } catch (ignored) {  
        println "That will never happen."  
        countDown.counter = Integer.MIN_VALUE  
    } finally {  
        return countDown  
    }  
}  
  
println finalCountDown().counter
```

KAK TAK TO??? HY KAK TAK
TO??



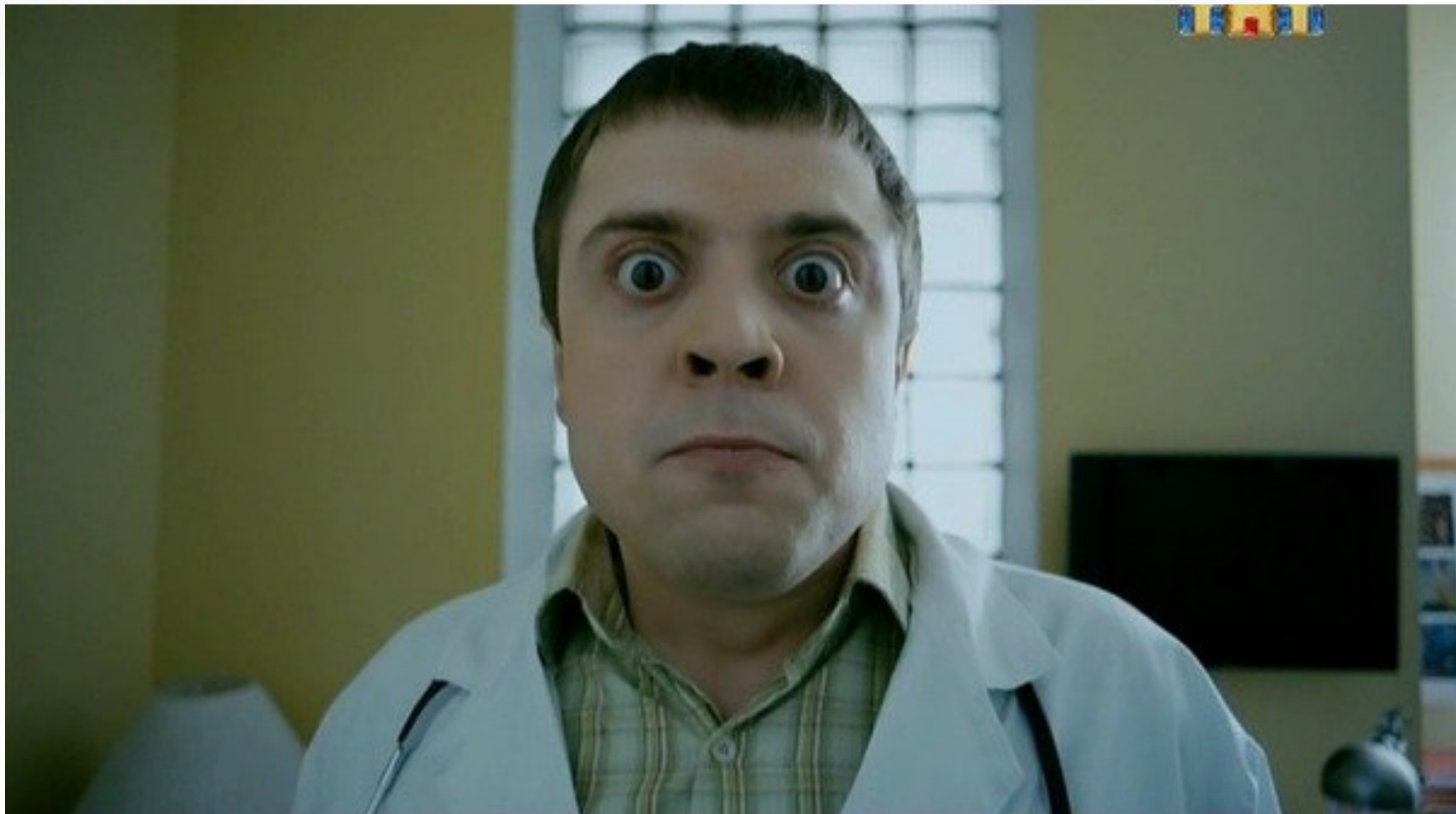
```
class Countdown { int counter = 10 }
```

```
Countdown finalCountDown() {  
    def countdown = new Countdown()  
    try {  
        countdown.counter = --countdown.counter  
    } catch (ignored) {  
        ignored.printStackTrace()  
        countdown.counter = Integer.MIN_VALUE  
    } finally {  
        return countdown  
    }  
}
```

```
println finalCountDown().counter
```

```
org.codehaus.groovy.runtime.typehandling.GroovyCastException:  
Cannot cast object '9' with class 'java.lang.Integer' to class 'Countdown'  
-2147483648
```


ЛУЧШЕ НЕ СТАЛО...



ЧТО ПРОИЗОШЛО?

```
public Countdown finalCountDown() {  
    java.lang.Object countDown = new Countdown()  
    try {  
        return countDown .counter = --( countDown .counter)  
    }  
    catch (java.lang.Exception ignored) {  
        this.println('That will never happen.')  
        return countDown .counter = java.lang.Integer.MIN_VALUE  
    }  
    finally {  
        return countDown  
    }  
}
```

ИСПОЛЬЗУЙ ОТВЕТ

```
class Countdown { int counter = 10 }

Countdown finalCountDown() {
    def countDown = new Countdown()
    try {
        countDown.counter = --countDown.counter
    } catch (ignored) {
        println "That will never happen."
        countDown.counter = Integer.MIN_VALUE
    } finally {
        return countDown
    }
    42
}

println finalCountDown().counter
```



ПОЧИНИЛИ!

A. Startup failure

B. That will never happen.

```
class Countdown { int counter = 10 }
```

```
Countdown finalCountDown() {  
    def countDown = new Countdown()  
    try {  
        countDown.counter = --countDown.counter  
    } catch (ignored) {  
        println "That will never happen."  
        countDown.counter = Integer.MIN_VALUE  
    } finally {  
        return countDown  
    }  
    42  
}
```

```
println finalCountDown().counter
```

-2147483648



D.42

?!?!



ЭТО ВОООБЩЕ КАК?!

```
public Countdown finalCountDown() {  
    java.lang.Object countDown = new Countdown()  
    try {  
        countDown .counter = --( countDown .counter)  
    }  
    catch (java.lang.Exception ignored) {  
        this.println('That will never happen.')        countDown .counter = java.lang.Integer.MIN_VALUE  
    }  
    finally {  
        return countDown  
    }  
    return 42  
}
```



ReturnAdder shouldn't add return if there's one in a finally block

Details

Type:	Bug	Status:	Open
Priority:	Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	None
Component/s:	None		
Labels:	None		
Number of attachments :	0		

Description

The following code was written for a "Groovy puzzlers" talk:

```
class CountDown { int counter = 10 }

CountDown finalCountDown() {
    def countDown = new CountDown()
    try {
```

People

Assignee:	Unassigned
Reporter:	Cédric Champeau
Votes:	0 Vote for this issue
Watchers:	1 Start watching this issue

Dates

Created:	Today 4:59 PM
Updated:	Today 4:59 PM



PUBLIC -

ПОПРОБУЙ ПОЛУЧИТЬ PUBLIC PROPERTY.

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

```
class Property implements Public {}
```

```
Property publicProperty = new Property()
```

```

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

```

```

class Property implements Public {}

```

```

Property publicProperty = new Property()

```

A. publicProperty.@property publicProperty.getProperty()

B. publicProperty.public__prop publicProperty.property

У МЕНЯ ДВА ВОПРОСА

1. Что за дебильное имя?
2. Откуда я мог это знать?



ОТВЕТ НА 1:

IN SOVIET RUSSIA



**YOU DON'T ACCESS
PUBLIC PROPERTY**

ОТВЕТ НА 2:



А ты документацию читать не пробовал???

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 recommended to use them and considered as a bad practice.

ГДЕ КАРТА БИЛЛИ?

Где карта,
Билли?
Нам нужна
карта.....



ГДЕ КАРТА БИЛЛИ?

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

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

A.NoSuchElementException

B.nt

C.treasure

D.Синий экран смерти, 1000
центов!

КАК ЭТО NULL?



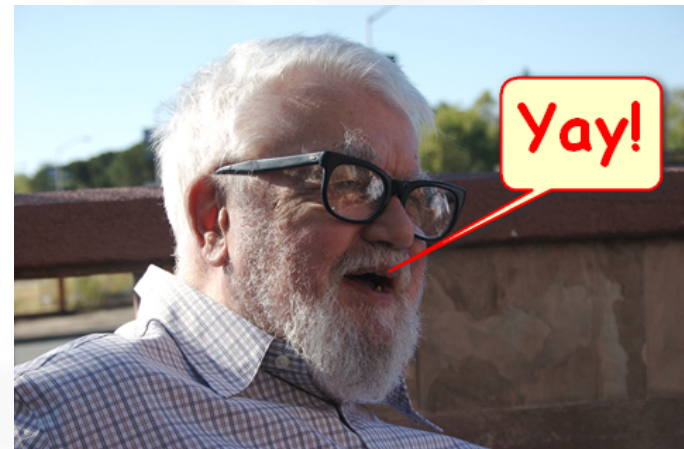
ПО УМОЛЧАНИЮ КЛЮЧ ЭТО СТРИНГ, ПОЛЮБОМУ!

```
6  
7 def key = 'x'  
8 def map = [key: 'treasure']  
9 def value = map.get(key)  
10 print
```

**See the green?
Green means "String"**

ЧИНИМ

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



ЕЩЕ КАРТЫ, БИЛЛИ!



ЕЩЕ КАРТЫ, БИЛЛИ!

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

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

A.NoSuchElementException

B.B.n

C.treasure

D.Kernel Panic киту в глотку!

А СЕЙЧАС ТО-ЧТО?



ПО УМОЛЧАНИЮ КЛЮЧ ЭТО
СТРИНГ, ПОЛЮБОМУ, ЕСЛИ НЕ
ЦИФРЫ!

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

java.lang.Integer



НАМ НУЖНА КАРТА!



HAM HУЖHA KAPTА!

```
def key = 'x'  
def map = ["${key}": 'treasure']  
def value = map['x']  
println value
```

```
def key = 'x'  
def map = ["${key}": 'treasure']  
def value = map['x']  
println value
```

A.NoSuchElementException

B.B.n

C.treasure

D.Полундра!!! Startup error!

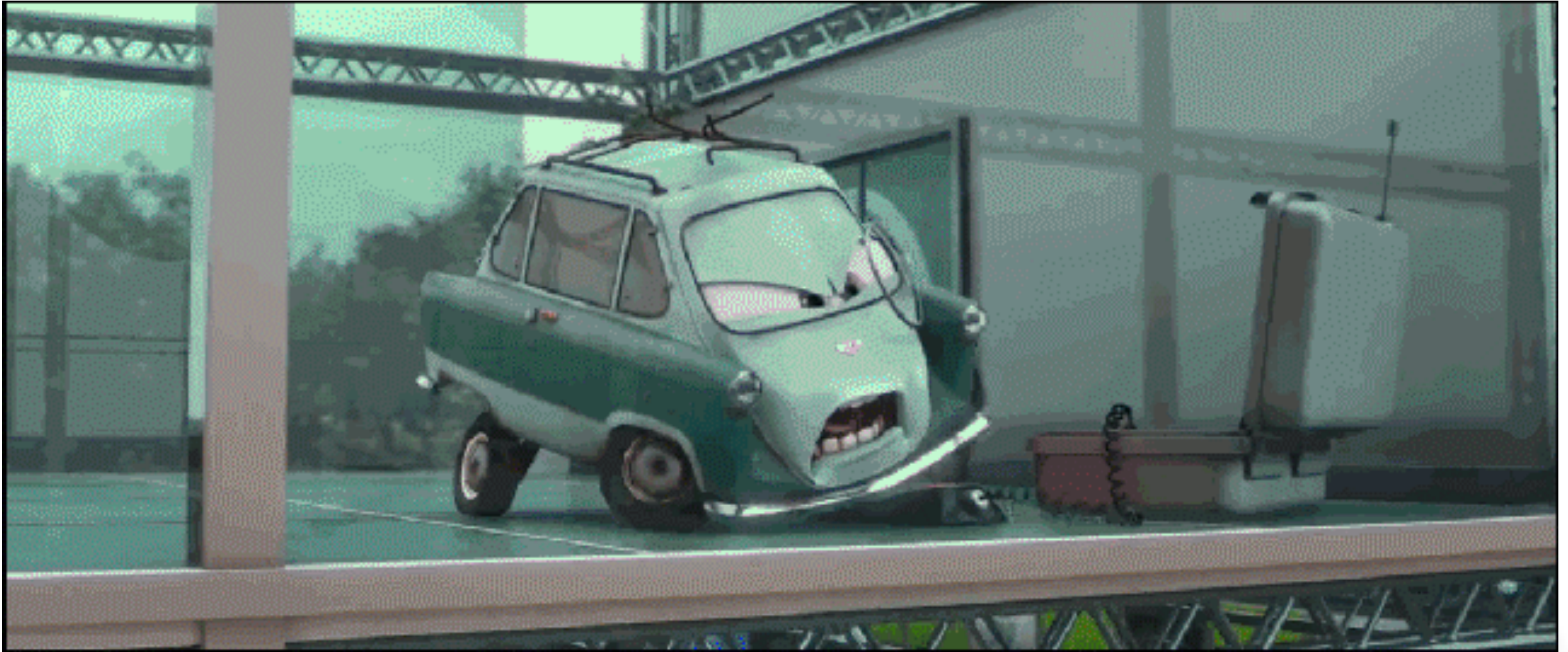
ЧТО ОПЯТЬ?



STRING ЭТО HE GSTRING, A
ТИП ВАЖЕИ

```
def map = ["${key}": 'treasure']  
println map.keySet().first().class.name  
  
org.codehaus.groovy.runtime.GStringImpl
```


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.false



C.true

D.NullPointerException



ПУТЬ К ОСМЫСЛЕНИЮ. ШАГ 1:

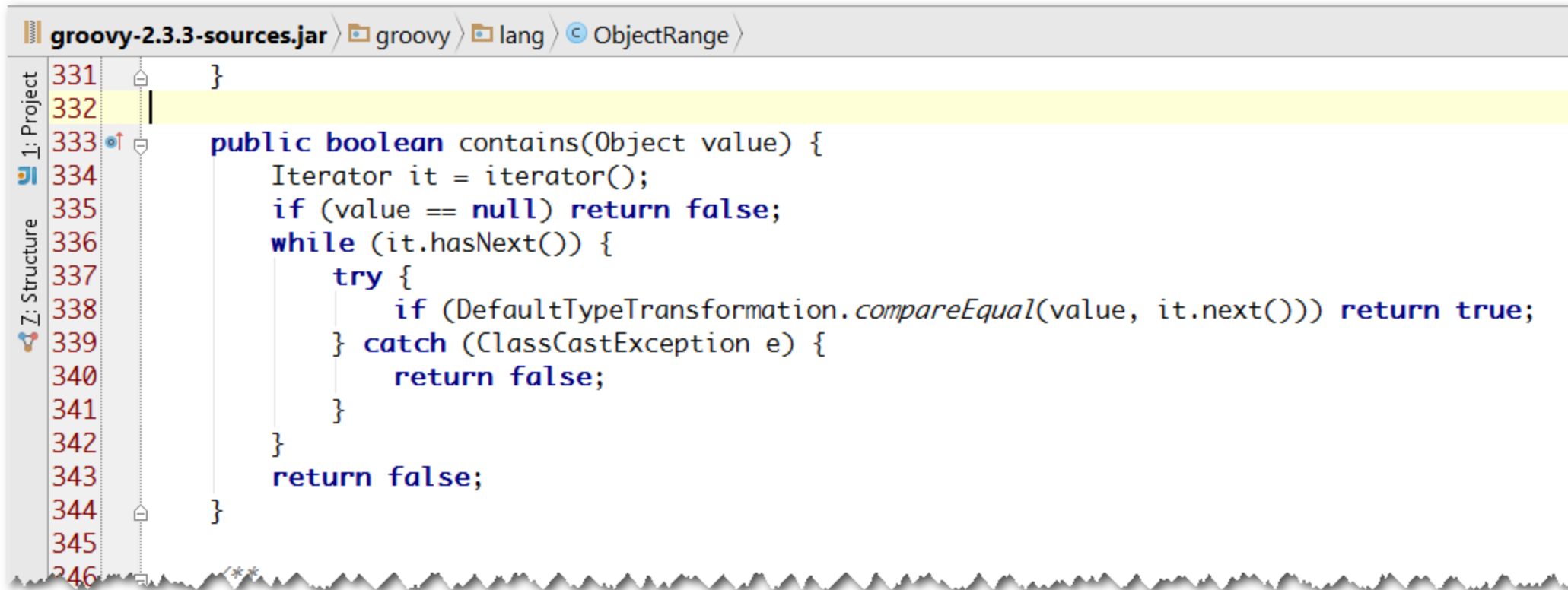
```
6
7 def range = 1.0..10.0
8
9 println range.class.name
```

Run OutOfRange

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

Process finished with exit code 0

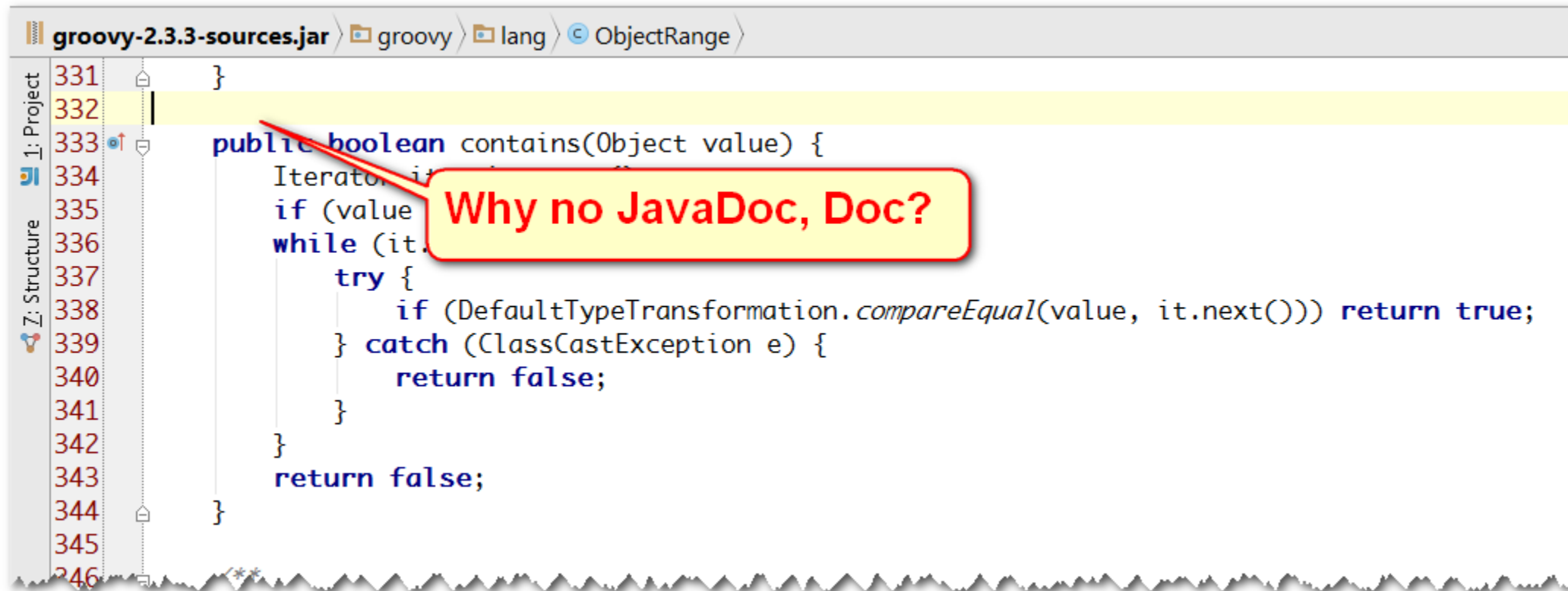
ПУТЬ К ОСМЫСЛЕНИЮ. ШАГ 2:



The screenshot shows an IDE window with the file path `groovy-2.3.3-sources.jar > groovy > lang > ObjectRange`. The code is for the `ObjectRange` class. Line 332 is highlighted in yellow. The code includes a `contains` method that uses an iterator and a `try-catch` block for comparison.

```
331     }  
332  
333     public boolean contains(Object value) {  
334         Iterator it = iterator();  
335         if (value == null) return false;  
336         while (it.hasNext()) {  
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 }
```

ПУТЬ К ОСМЫСЛЕНИЮ. ШАГ 2:



The screenshot shows an IDE window with the file path `groovy-2.3.3-sources.jar > groovy > lang > ObjectRange`. The code editor displays the `contains` method of the `ObjectRange` class. The method signature is `public boolean contains(Object value)`. The method body includes a `while` loop that iterates over the `ObjectRange` and checks if the `value` is equal to the current element using `DefaultTypeTransformation.compareEqual`. If a match is found, it returns `true`; otherwise, it returns `false` after the loop. A red callout box with the text "Why no JavaDoc, Doc?" points to the method signature.

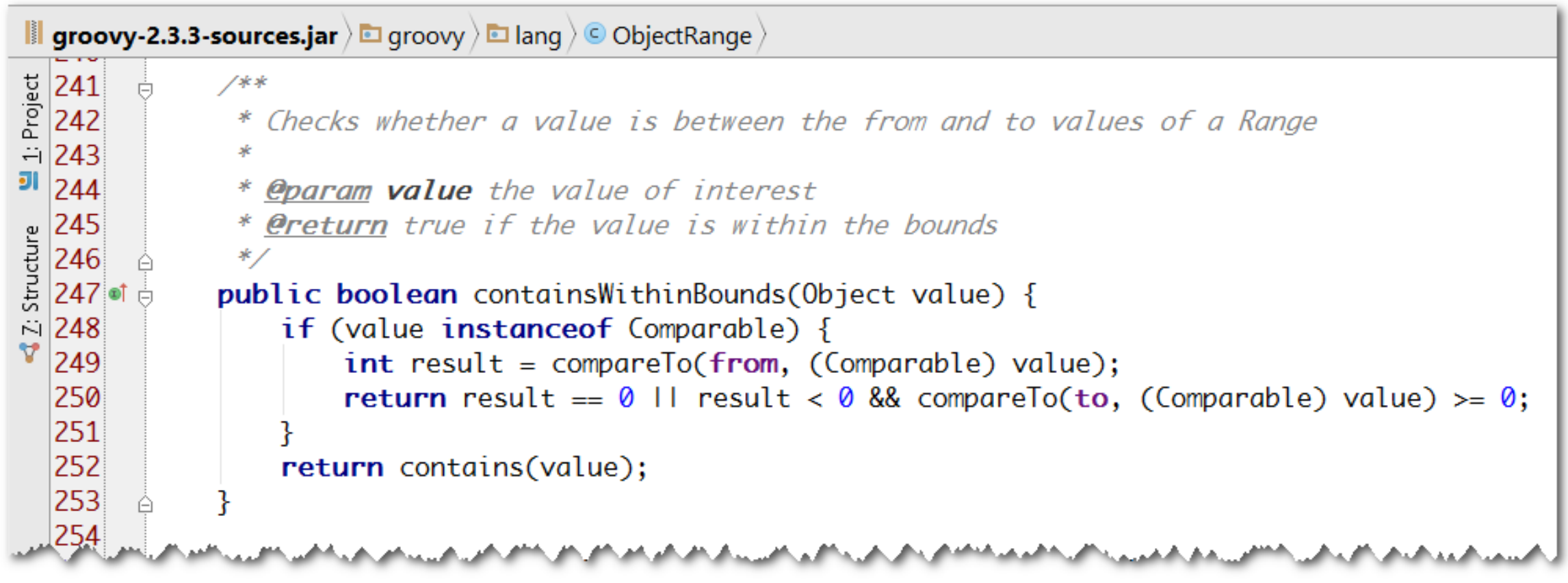
```
331 }
332
333 public boolean contains(Object value) {
334     Iterator it = iterator();
335     if (value != null)
336         while (it.hasNext()) {
337             try {
338                 if (DefaultTypeTransformation.compareEqual(value, it.next())) return true;
339             } catch (ClassCastException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

ПУТЬ К ОСМЫСЛЕНИЮ. ШАГ 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

РЕШЕНИЕ ПРОСТОЕ, НО СТРАННОЕ

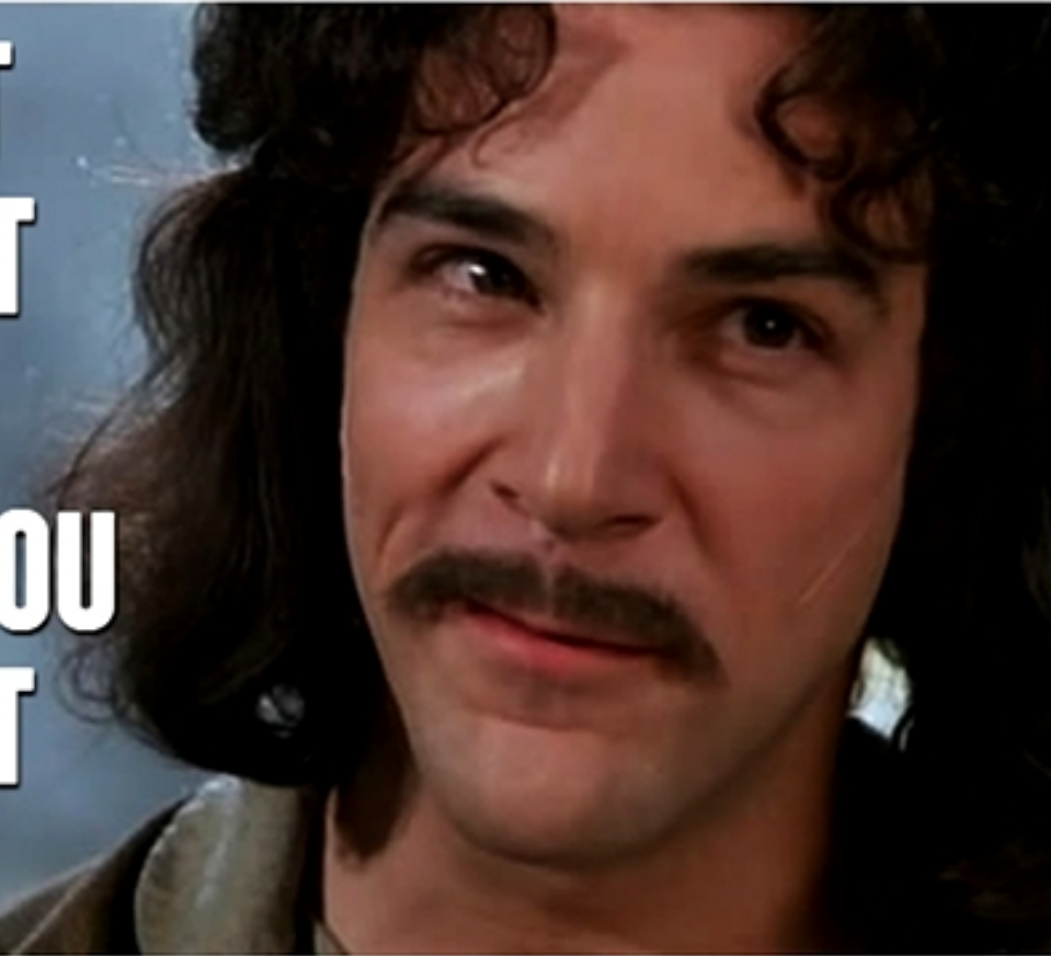


The screenshot shows an IDE window with the file path `groovy-2.3.3-sources.jar > groovy > lang > ObjectRange`. The left sidebar displays a project structure with '1: Project' and 'Z: Structure'. The main editor area shows the following Groovy code:

```
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
```


GROOVY, ТЫ УТВЕРЖДАЕШЬ ЧТО
ДИАПАЗОН 1.0..10.0 НЕ СОДЕРЖИТ
5.6?

**I DO NOT
THINK IT
MEANS
WHAT YOU
THINK IT
MEANS.**



ЧТО БЫ СДЕЛАЛИ МЫ

- Нас не интересует
- представл
- Назовите «Continuous
- И пусть де containsWi
- Почините

 groovy / GROOVY-2771
range of double-values broken?

Log In

Details

Type:	 Bug	Status:	 Closed
Priority:	 Major	Resolution:	Won't Fix
Affects Version/s:	1.5.4	Fix Version/s:	None
Component/s:	None		
Labels:	None		
Environment:	Groovy Shell (1.5.4, JVM: 1.5.0_11-b03)		
Number of attachments :	0		

People

Assignee:
 Paul King

Reporter:
 Peter Fürholz

Votes:
 Vote for this issue

Watchers:
 Start watching this issue

Description

I got following error:

D:\>groovysh
Groovy Shell (1.5.4, JVM: 1.5.0_11-b03)
Type 'help' or '?h' for help

Dates

Created:
23/Apr/08 4:29 AM

POWER RANGE(RS)



УБИРАЕМ ЕДИНИЧКУ

```
[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]



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



Это неправильные
скобки!


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



Другое дело!

ОБМАНУЛИ ДУРАКА

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

ОБМАНУЛИ ДУРАКА

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

This is range

ОБМАНУЛИ ДУРАКА

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

This is range

This is list with one
element - the range

ОБМАНУЛИ ДУРАКА

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

This is range

This is list with one
element - the range

In the sole iteration
the it is the range

ОБМАНУЛИ ДУРАКА

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

This is range

This is list with one
element - the range

In the sole iteration
the it is the range

And we just
remove 1 from it

ОБМАНУЛИ ДУРАКА

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

This is range

This is list with one
element - the range

In the sole iteration
the it is the range

And we just
remove 1 from it

[0, 2, 3, 4, 5, 6, 7, 8, 9]



SAVE THE DATE

#JOKERCONF IS COMING



JAVA2, КТО ПОМНИТ?

```
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
```

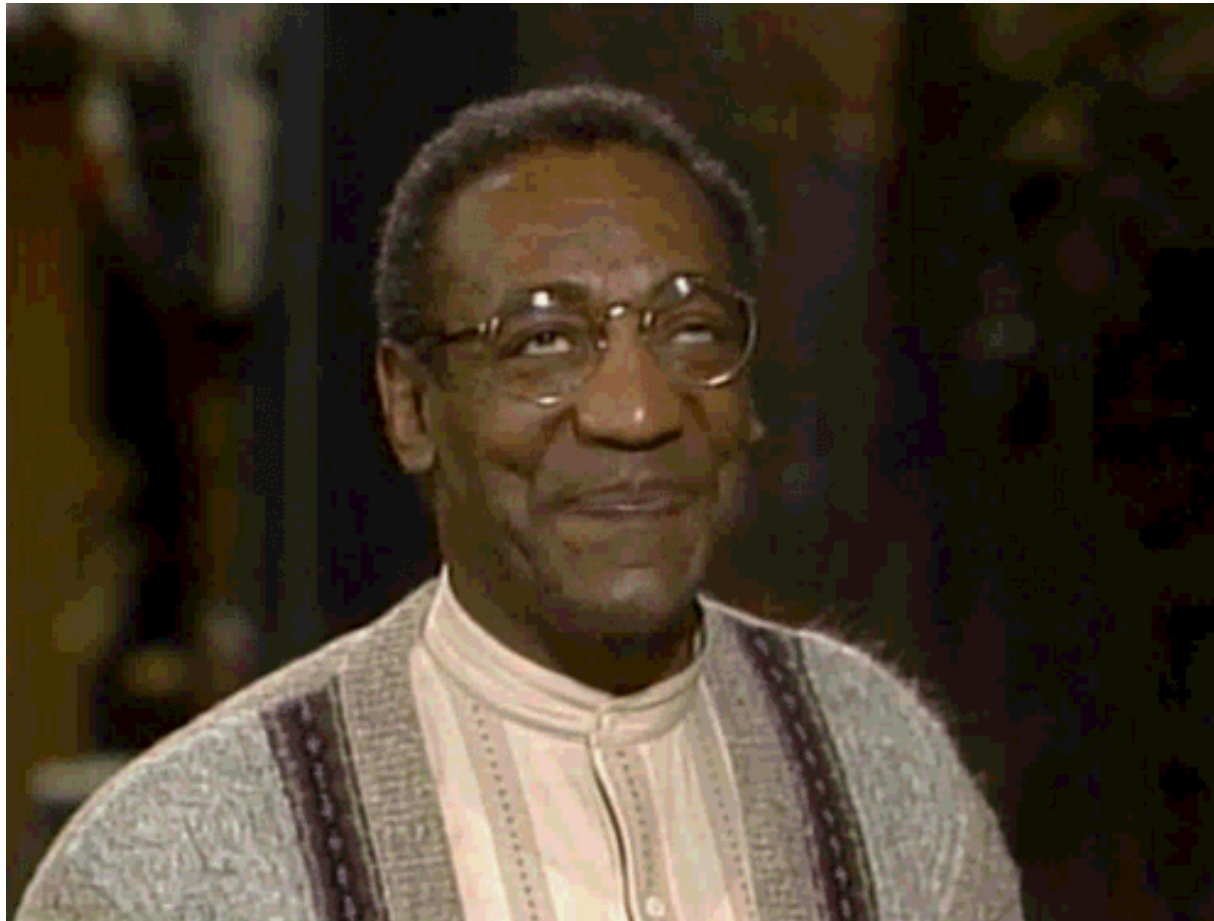
A.Startup Failure

B.[1, 2, 3, WEEKDAY 17 16:15:11
CDT 2014]

C.[1, 2, 3, 1410388511456]

D.[1, 2, 3, 15 11:15:11]

А, Я ЗНАЮ ЧТО СЛУЧИЛОСЬ!



ERASURE!



ERASURE!



The Java™ Tutorials

[Download Ebooks](#)
[Download JDK](#)
[Search Java Tutorials](#)
[Hide TOC](#)

[Generics \(Updated\)](#)

[Why Use Generics?](#)

[Generic Types](#)

[Raw Types](#)

[Generic Methods](#)

[Bounded Type Parameters](#)

[Generic Methods and](#)

[Bounded Type](#)

[Parameters](#)

[Generics, Inheritance, and
Subtypes](#)

[Type Inference](#)

[Wildcards](#)

[Upper Bounded](#)

[Wildcards](#)

[Unbounded Wildcards](#)

[Lower Bounded](#)

[« Previous](#) • [Trail](#) • [Next »](#) [Home Page](#) > [Learning the Java Language](#) > [Generics \(Updated\)](#)

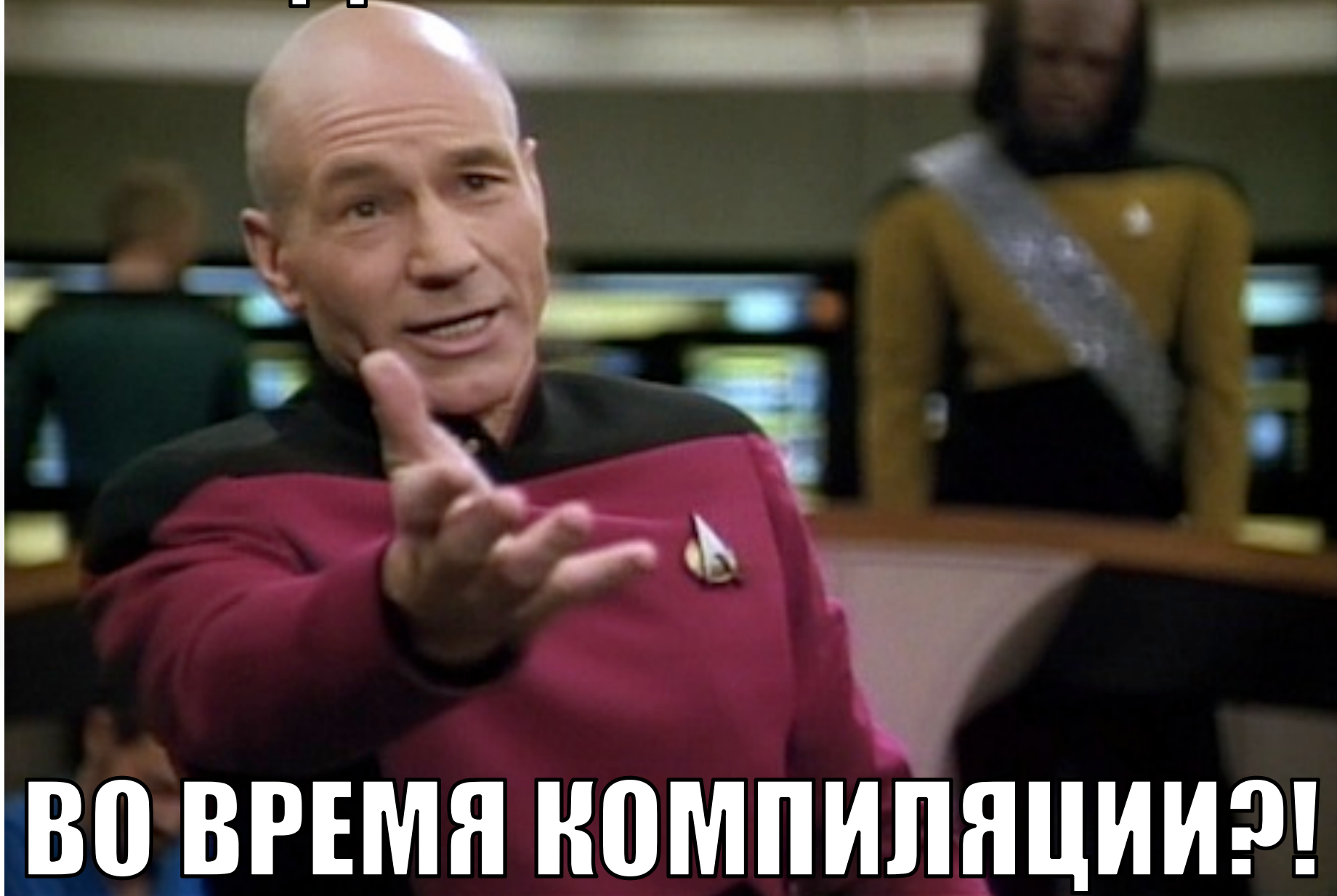
Type Erasure

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.

А ГДЕ МОИ ПРОВЕРКИ



ВО ВРЕМЯ КОМПИЛЯЦИИ?!

«В GROOVY МОЖНО ЗАСУНУТЬ
ЧТО УГОДНО, КУДА УГОДНО»

```
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!

Structure

```
@CompileStatic
def saveTheDate() {
    List<Long> list = [1l, 2l, 3l]
    def now = new Date()
    list << now
    print list
}
```

'leftShift' in 'org.codehaus.groovy.runtime.DefaultGroovyMethods' cannot be applied to '(java.util.Date)'

```
saveTheDate()
```


π – ТОРТ И ЧИСЛО



DOUBLE THE PI

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

- A. true
- B. MissingPropertyException
- C. false
- D. MissingMethodException

ПРОСЁК!



ПРОСЁК!

```
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

ΠΡΟCΕΚ!

```
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

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

A. true

B. MissingPropertyException

C. false 

D. MissingMethodException



ЕСЛИ НЕ МОЖЕТ БЫТЬ ПРОПЕРТИ, ЗАЧЕМ ПАРИТЬСЯ?

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

```
class Person  
Person person  
println "The
```

```
double value  
println "$vo
```



Trust
IntelliJ!

```
start(char ch) {  
};
```

'Andres')

me"

can be property

ПОСМОТРИ НА TOSTRING() В КЛАССЕ 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

ЗНАЕШЬ ЧТО ОЗНАЧАЕТ
СОКРАЩЕНИЕ RSVP?



СПРОСИ ФРАНЦУЗОВ!

RSVP

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

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

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

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

- A. StartGroovyFailure
- B. 1
- C. 2
- D. MissingPropertyException




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

WAT?!

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

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



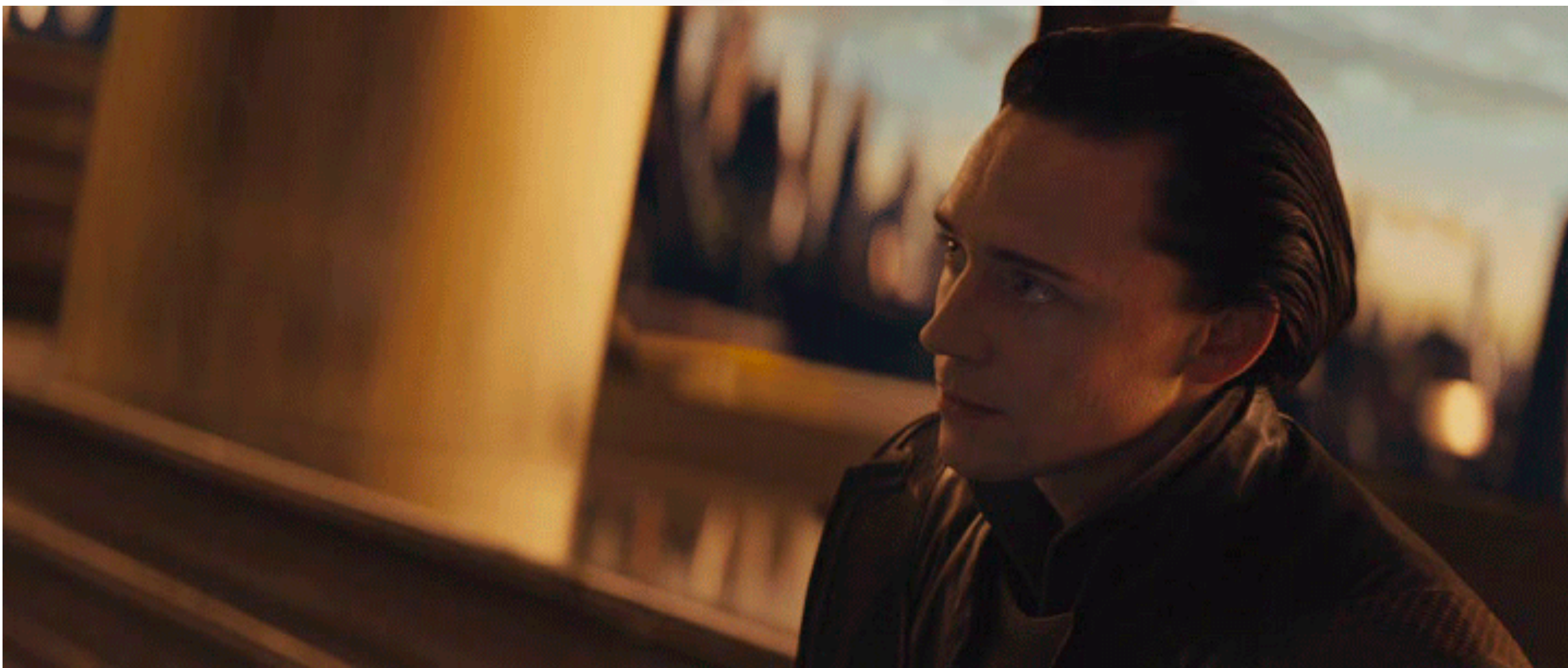
A BOT ПОЧЕМУ

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

This is class cast!

Package name

Class name



ПОЧИНИТЬ: ЧТО УГОДНО, ЧТО
НЕ ПОХОЖЕ НА КАСТИНГ

НАПРИМЕР:

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

ПОЧИНИТЬ: ЧТО УГОДНО, ЧТО НЕ ПОХОЖЕ НА КАСТИНГ

НУ, ИЛИ, УБРАТЬ СКОБКИ:

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



Как убрать скобки?!

Strange case of
and Dr Jekyll
MR. HYDE



R.L STEVENSON

```
class MrHyde {  
    def me() {  
        return this  
    }  
}
```

```
class DrJekyll {  
}
```

```
DrJekyll.mixin MrHyde
```

```
def drJekyll = new DrJekyll().me() as DrJekyll
```

```
def mrHide = new DrJekyll().me()  
println "$drJekyll and $mrHide, are they the  
same? ${ (drJekyll.class).equals(mrHide.class) }"
```

A. DrJekyll@xxxxxxx and MrHyde@yyyyy  are they the same? true

B. DrJekyll@xxxxxxx and MrHyde@yyyyyyyyy, are they the same? false

```
class MrHyde {  
  def me() {  
    return this  
  }  
}
```

C. DrJekyll@xxxxxxx and DrJekyll@xxxxxxx, are they the same? true

```
class DrJekyll {  
  
}
```

D. MrHyde@yyyyyyyyyy and MrHyde@yyyyyyyyy, are they the same? false

```
DrJekyll.mixin MrHyde
```

```
def drJekyll = new DrJekyll().me() as DrJekyll
```

```
def mrHide = new DrJekyll().me()
```

```
println "$drJekyll and $mrHide, are they the same? $  
{(drJekyll.class).equals(mrHide.class)}"
```



ВСТРАИВАЕМ ОДИН КЛАСС В ДРУГОЙ



В КОДЕ MIXINMETACLASS

core > src > main > org > codehaus > groovy > reflection > MixinInMetaClass >

```
public static void mixinClassesToMetaClass(MetaClass self, List<Class> categoryClasses) {  
    for (Class categoryClass : categoryClasses) {  
        final CachedClass cachedCategoryClass = ReflectionCache.getCachedClass(categoryClass);  
        final MixinInMetaClass mixin = new MixinInMetaClass(mc, cachedCategoryClass);  
  
        final MetaClass metaClass = GroovySystem.getMetaClassRegistry().getMetaClass(categoryClass);  
        final List<MetaProperty> propList = metaClass.getProperties();  
        for (MetaProperty prop : propList)  
            if (self.getMetaProperty(prop.getName()) == null) {  
                mc.registerBeanProperty(prop.getName(), new MixinInstanceMetaProperty(prop, mixin));  
            }  
    }  
}
```




groovy / GROOVY-7066

MixinInMetaClass shouldn't consider "class" as a property

Log In

Export

Details

Type:	Bug	Status:	Open
Priority:	Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	None
Component/s:	None		
Labels:	None		
Number of attachments :	0		

People

Assignee:	Unassigned
Reporter:	Cédric Champeau
Votes:	0 Vote for this issue
Watchers:	1 Start watching this issue

Dates

Created:	Today 5:11 PM
Updated:	Today 5:11 PM

Description

The following code was showcased in a "Groovy Puzzlers" talk:

```
class MrHyde {  
    def me() {  
        return this  
    }  
}
```

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.  Stack Error

B. int, long,
boolean

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

НУ ЭТО ЯСНО, А ПОЧЕМУ?

порядок

```
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

непорядок

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

')' expected

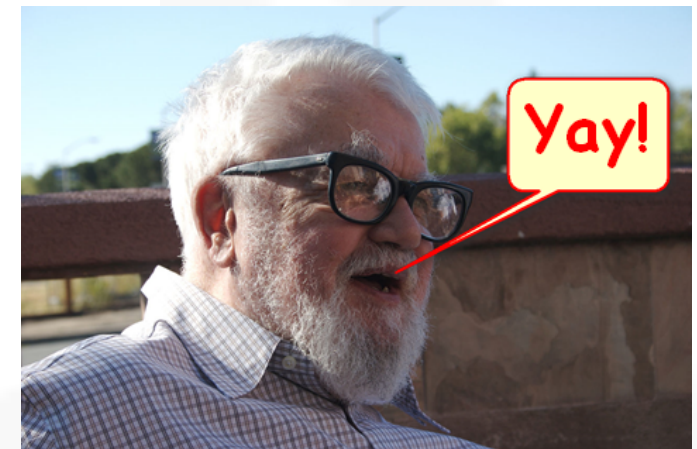
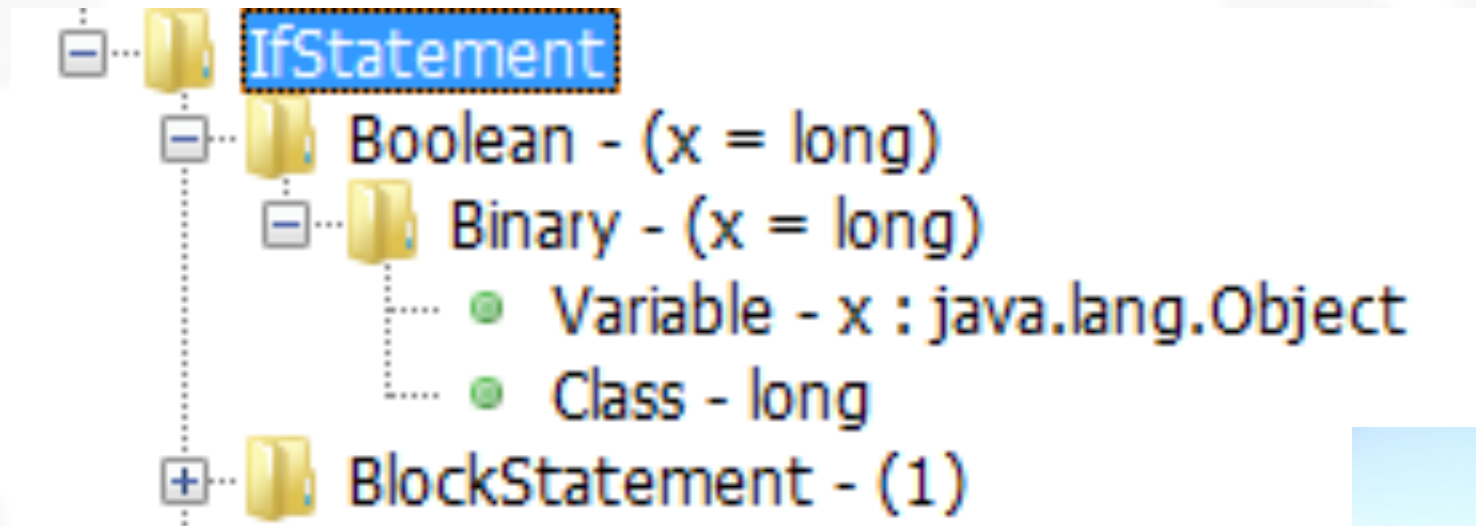
Unexpected symbol

А ЭТО?

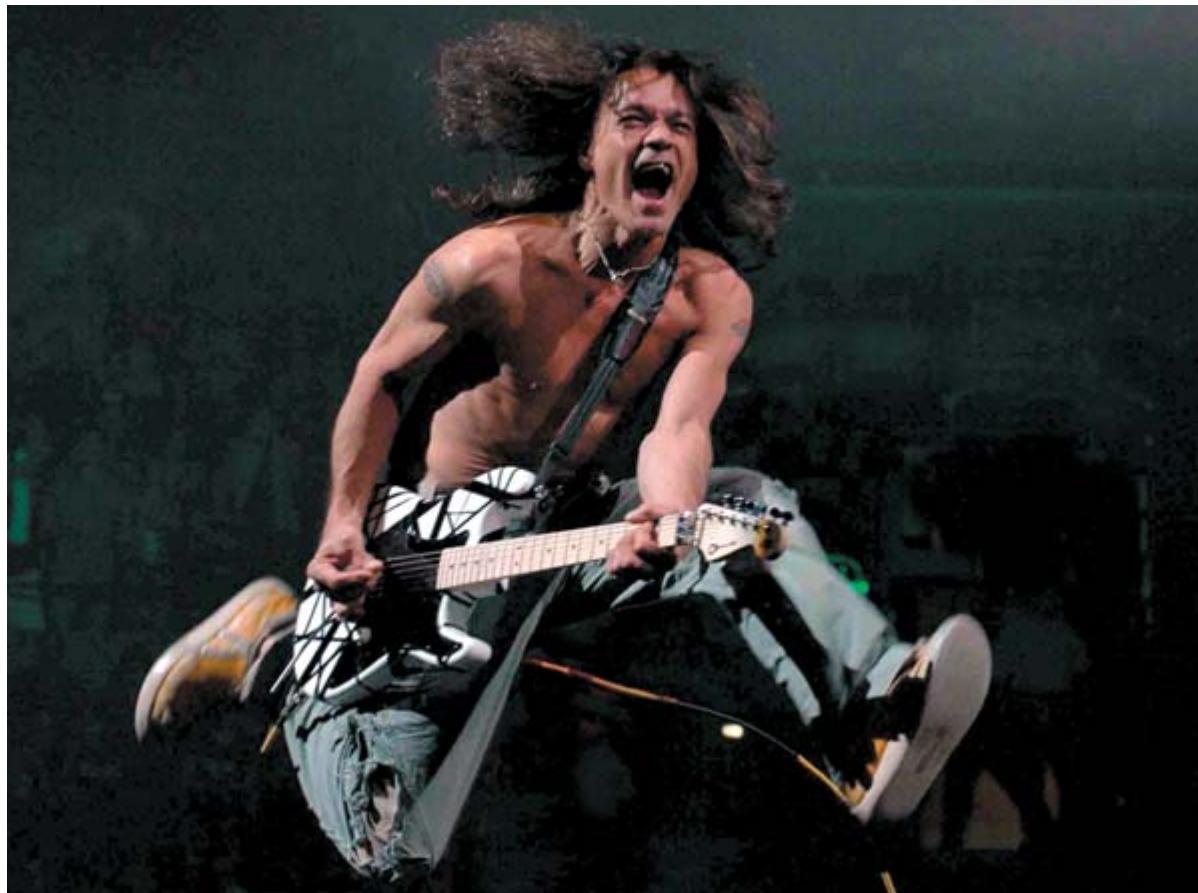
```
def x = int
//println x
//
if ((x = long)) {
  println x
}
//
//if (x = boolean ) {
//  println x
//}
```



(GROOVY) TRUTH МОЖЕТ
РАНИТЬ, ТАК ЧТО ПРОСТО
ДОБАВЬ СКОБОК



JUMP! A.K.A. ПРОПАВШИЕ СЛОВА



JUMP! A.K.A. ПРОПАВШИЕ СЛОВА

```
class VanHalen {  
  
    public static jump() {  
        "Here are the ${lyrics()}"  
    }  
  
    def methodMissing(String name, def args) {  
        'lyrics'  
    }  
}  
  
println VanHalen.jump()
```

JUMP! A.K.A. ПРОПАВШИЕ

СЛОВА

```
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.  up failure

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

```
Object invokeMissingMethod(Object instance, String methodName, Object[] arguments)
```

Documentation for invokeMissingMethod(Object, String, Object[]) 

← → ↑  groovy-2.3.3 

[groovy.lang.MetaClass](#)

[Object](#) invokeMissingMethod([Object](#) instance, [String](#) methodName, [Object](#)[] arguments)

Attempts to invoke the methodMissing method otherwise throws a MissingMethodException

Parameters:

- instance - The instance to invoke methodMissing on
- methodName - The name of the method
- arguments - The arguments to the method

```
/**
 * Invokes the propertyMissing method on the given instance
 *
 * @param instance the instance to invoke the method on
 * @param propertyName the name of the property
 * @param optionalArguments optional arguments to the method
 * @param isGetter true if the method is a getter, false otherwise
 *
 * @return The result of the method invocation
 */
Object invokeMiss
```


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

```
7 |  
8 | class VanHalen {  
9 |  
10 |     public static jump() {  
11 |         "Here are the ${lyrics()}"  
12 |     }  
13 |  
14 |     def methodMissing(String name, def args) {  
15 |         'lyrics'  
16 |     }  
17 | }  
18 |  
19 | 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>

РЕШЕНИЕ: СТАТИЧЕСКИЙ METHODMISSING

```
class VanHalen {  
    public static jump() {  
        "Here are the ${lyrics()}"  
    }  
  
    static $static_methodMissing(String name, def args) {  
        'lyrics'  
    }  
}  
  
println VanHalen.jump()
```

ЕЩЁ ЛУЧШЕ – ИСПОЛЬЗУЙ ООП

```
class VanHalen {  
    public jump() {  
        "Here are the ${lyrics()}"  
    }  
  
    def methodMissing(String name, def args) {  
        'lyrics'  
    }  
}  
  
println new VanHalen().jump()
```

WHAT IS THE FREQUENCY
KENNETH?

R.E.M.

Compact Disc Maxi-Single

WHAT'S THE METACLASS, БАРУХ?

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

A. MissingMethodException

B. What's the

org.codehaus.groovy.runtime.HandleMetaClass@YYY
YYYY [groovy.lang.MetaClassImpl@YYYYYYYYYY[class
java.util.LinkedHashMap]], Барух?

C. What's the java.util.LinkedHashMap@XXXXXXX,
Барух?



D. What's the frequency of the word "Baruch"?

СРАБОТАЛО, НО

НЕ ФАКТ, ЧТО ЭТО ХОРОШО

ПОМНИШЬ, ДЖЕЙМС ГОСЛИНГ ПРЕДУПРЕЖДАЛ



ЧТО OPERATOR OVERLOADING ЭТО ПЛОХО?

ПЕРЕБОР OPERATOR OVERLOADING!

Если `map.metaClass`

переопределён (`map.get('metaClass')`)

для

`map.getMetaClass` ~~он не может быть~~

переопределён и для

,



ВЫВОДЫ



1. Пишите читабельный код
2. Комментируйте все что не очевидно
3. Иногда это баг
4. Пользуйтесь static code analysis tools
intellij IDEA!
5. Rtfm
6. Don't code like my brother



Мы только начали! (надо еще форму придумать)

Засылайте Ваши паззлеры,
поделки и рисунки

- puzzlers jfrog@com
- @roovypuzzlers

ПАЗЗЛЕРЫ ВСЕГДА ПЛАТЯТ СВОИ ДОЛГИ



Вам понравилось?

Хвалите нас в твиттере

#groovypuzzlers

- @groovypuzzlers
- @ekaborisov
- @baruch

Вам не понравилось?

/dev/null

**"Расставашки -
всегда пичалька".**

(с) Сократик

