

The background image shows a panoramic view of the Atlanta skyline at dusk or night. The sky is a deep blue. In the foreground, a multi-lane highway with blurred lights from moving vehicles curves through the city. Several skyscrapers are visible, with one prominent building on the right featuring a distinctive, illuminated, pyramid-shaped top. The city lights are reflected in the windows of the buildings.

THE EPIC GROOVY PUZZLERS

SEASON II:

THE REVENGE OF THE PARENTHESES

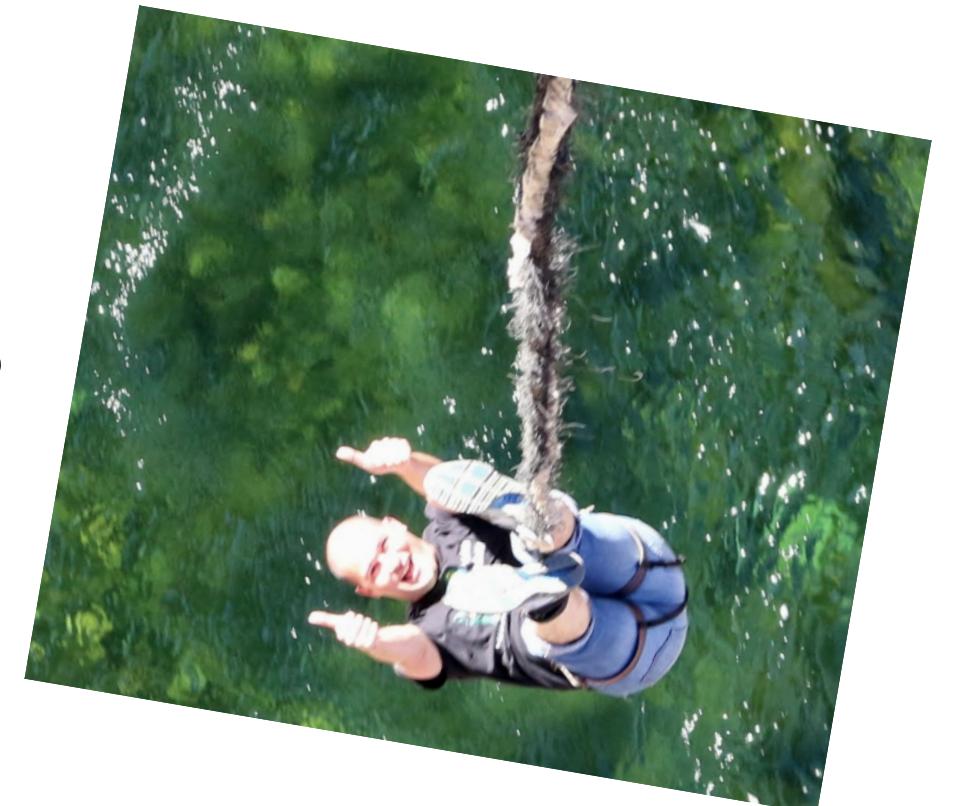
Who's speaking?
github.com/jbaruch
linkd.in/jbaruch



Baruch Sadogursky
J*, G* and Public Speaking Geek with JFrog FTW.
Israel | Computer Software

Current Developer Advocate at JFrog Ltd

View profile



stackoverflow.com/users/402053/jbaruch

Viktor Gamov
@gamussa



- Solutions Architect at Hazelcast
- International Conferences speaker
- Co-author of O'Reilly's «Enterprise Web Development»

Frogs ❤ Groovy

★ Groovy

Learn Documentation Download Community Ecosystem Socialize 

Download Groovy

- # Distributions
- # Through GVM
- # From your build tools
- # Other ways to get Groovy
- # System requirements



Download

 Improve this doc

In this download area, you will be able to download the [distribution](#) (binary and source), the Windows installer (for some of the versions) and the documentation for Groovy.

 [Download](#)

All the downloads are hosted in [Bintray's Groovy repository](#).

Registering on Bintray allows you to rate, review, and register for new version notifications.

Frogs ❤ Groovy

- ★ Hosting Groovy on Bintray
- ★ Hosting Groovy on Artifactory
- ★ Bintray UI is Grails
- ★ Artifactory plugins are Groovy
- ★ Gradle and Grails plugins on Bintray

First Season Was a Blast!



Groovy and Grails Puzzlers - As usual - Traps, Pitfalls, and End Cases

EVENT

[GR8Conf US 2014](#)



Groovy and Grails Puzzlers - As usual - Traps, Pitfalls, and End Cases

EVENT

[SpringOne2GX 2014](#)

TIME

10th September 2014 4:30pm-6:00pm



Groovy and Grails Puzzlers: As Usual—Traps, Pitfalls, and End Cases

EVENT

[JavaOne 2014](#)

TIME

30th September 2014 2:30pm-3:30pm



Groovy and Grails Puzzlers: As Usual - Traps, Pitfalls, and End Cases

EVENT

[JavaOne Latin America 2015](#)

TIME

25th June 2015 9:30am-10:30am



The Epic Groovy Puzzlers - As usual - Traps, Pitfalls, and End Cases

EVENT

[DevNexus 2015](#)

TIME

12th March 2015 10:30am-10:30am



The Epic Groovy Puzzlers - As usual - Traps, Pitfalls, and End Cases

EVENT

[DevNexus 2015](#)

TIME

12th March 2015 10:30am-10:30am

CLICK AND HACK



THE TYPING BROTHERS

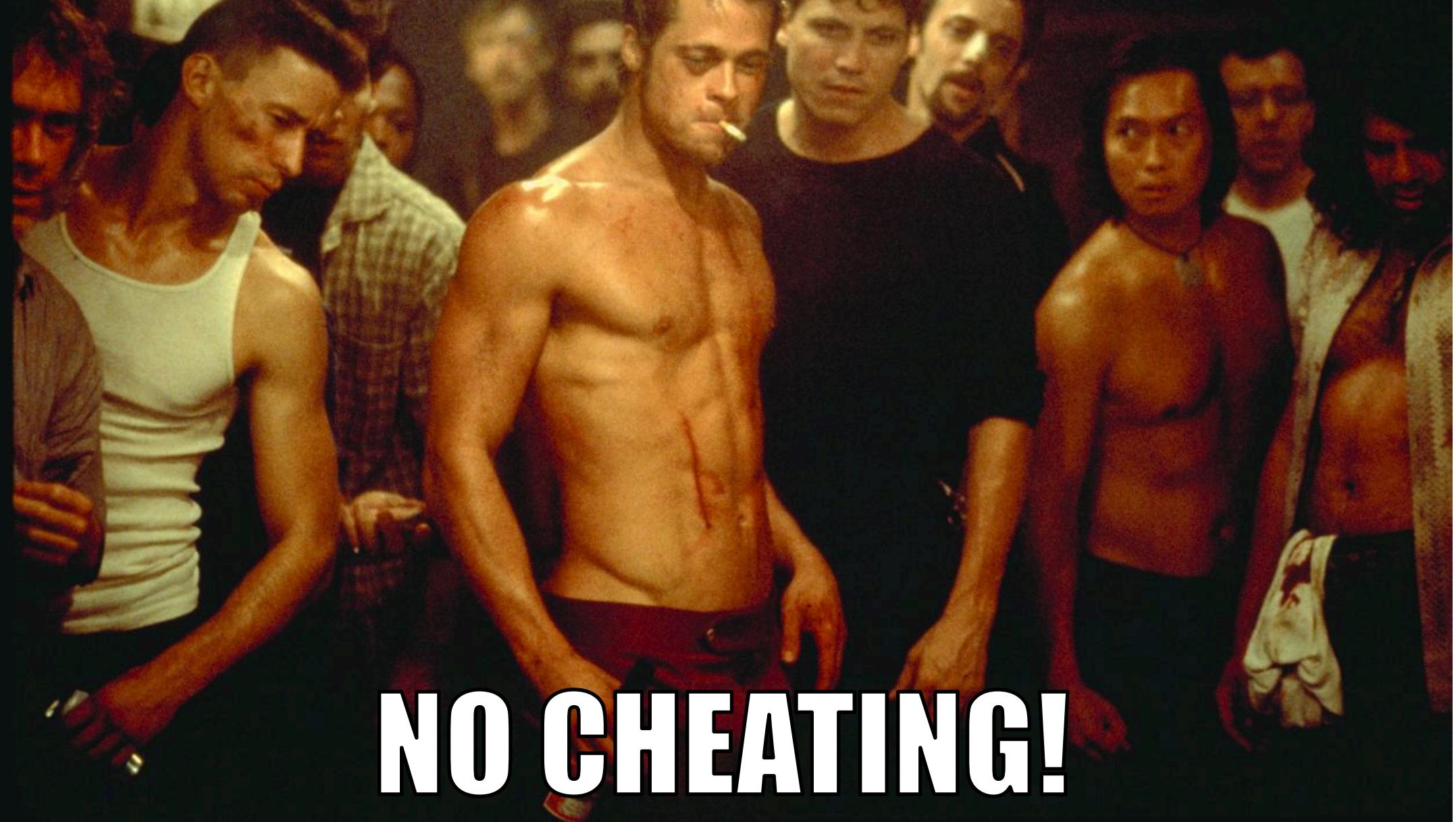
BTW,



1. Two entertaining guys on stage
2. Funny Puzzling questions
3. You think and vote
4. Awesome Groovy t-shirts fly in the air
5. Official twitter handle!
groovypuzzlers

#

FIRST RULE OF THE PUZZLERS:



NO CHEATING!

All works (or doesn't work) in Groovy 2.4.5



A photograph of a man with blonde hair, wearing a grey suit jacket, a white striped shirt, and a maroon patterned tie. He is leaning against a light-colored, textured wall, looking slightly off-camera with a neutral expression. His right hand is adjusting his tie.

CHALLENGE

ACCEPTED



© 2005 www.babybirdproductions.com

```
'a'..'z'.each { println it }
```

- A. a
 - b
 - c
 - .
 - .
 - .
 - z
- B. NoSuchElementException
 - C.
 - D. Won't run





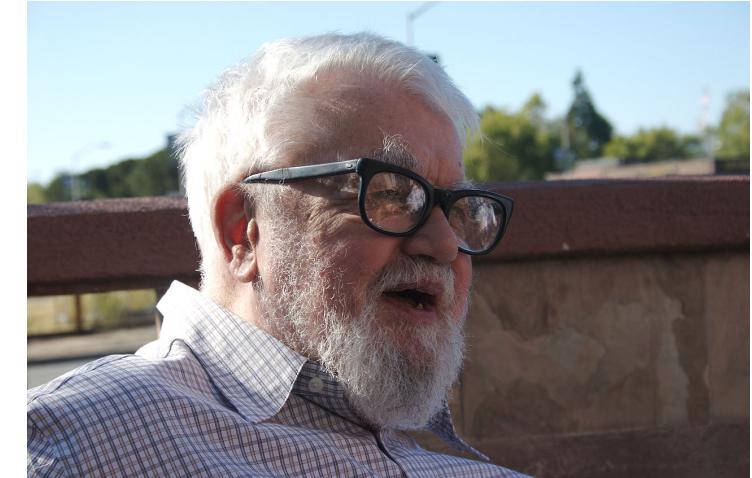
4GIFS.com

```
( 'a'..'z' ).each { println it }
```



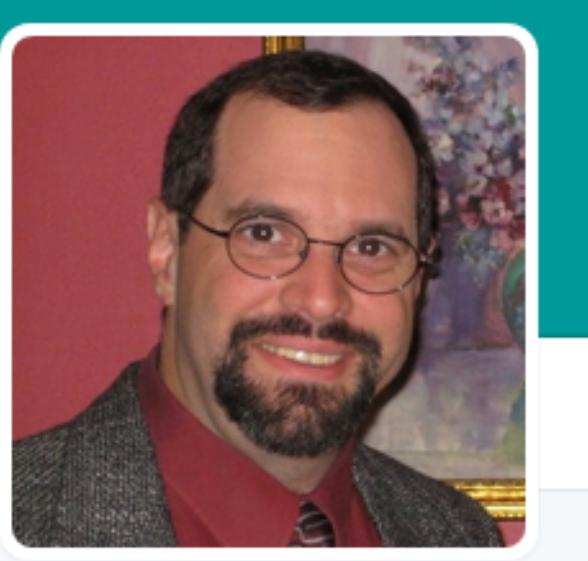
You knew it, right?

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



John McCarthy, the inventor of LISP

And the t-shirt goes to...



A portrait photograph of Ken Kousen, a man with dark hair and a beard, wearing glasses, a red shirt, and a patterned jacket. The photo is set against a red background and is framed by a white border.

Ken Kousen
@kenkousen FOLLOW YOU

Software trainer and developer, NFJS speaker, and author of *Making Java Groovy*

📍 Marlborough, CT
🔗 kousenit.com
⌚ Joined August 2008



How Many Bachs?

```
def back = 'back'  
def quotes = ["I'll be $back",  
"I'll be ${-> back}",  
"I'll be ${back}",  
"I'll be "+back]  
println quotes  
back = 'Bach'  
println quotes
```

- A.No Bachs
-  B.Groovy Bach
- C.Two Bachs
- D.Three Bachs



Only closures are evaluated at runtime
Others are inlined

Only closures are evaluated at runtime

Others are inlined

```
def back = 'back'
def quotes = ["I'll be $back",
"I'll be ${-> back}",
"I'll be ${back}",
"I'll be "+back]
```

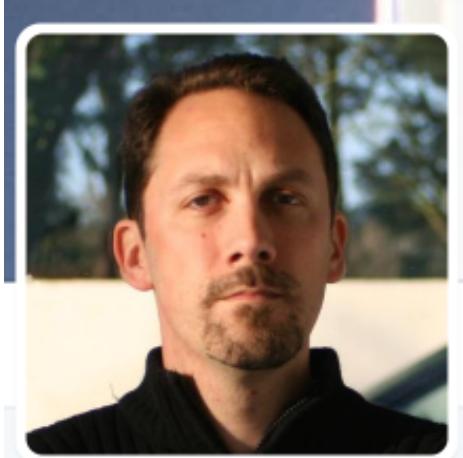
Only closures are evaluated at runtime

Others are inlined

```
def back = 'back'  
def quotes = ["I'll be $back",  
"I'll be ${-> back}",  
"I'll be ${back}",  
"I'll be "+back]
```

That is the only closure

And the t-shirt goes to...



Cédric Champeau

@CedricChampeau FOLLOW YOU

Software Engineer @Gradleware.
Conference speaker. Introvert
(carlkingdom.com/10-myths-abou
Wrote the static compiler for
#groovylang.

📍 St Hilaire de Loulay, France

🔗 melix.github.io/blog

🕒 Joined January 2010



Plugging the Users in - Extend Your Application with Pluggable Groovy DSL

Recorded at:

by Baruch Sadogursky on Mar 12, 2014 | [Discuss](#)

Share

[My Reading List](#)[Read later](#)[View Presentation](#)

A problem has been detected and windows has been shut down to prevent damage to your computer.

DRIVER_POWER_STATE_FAILURE

If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

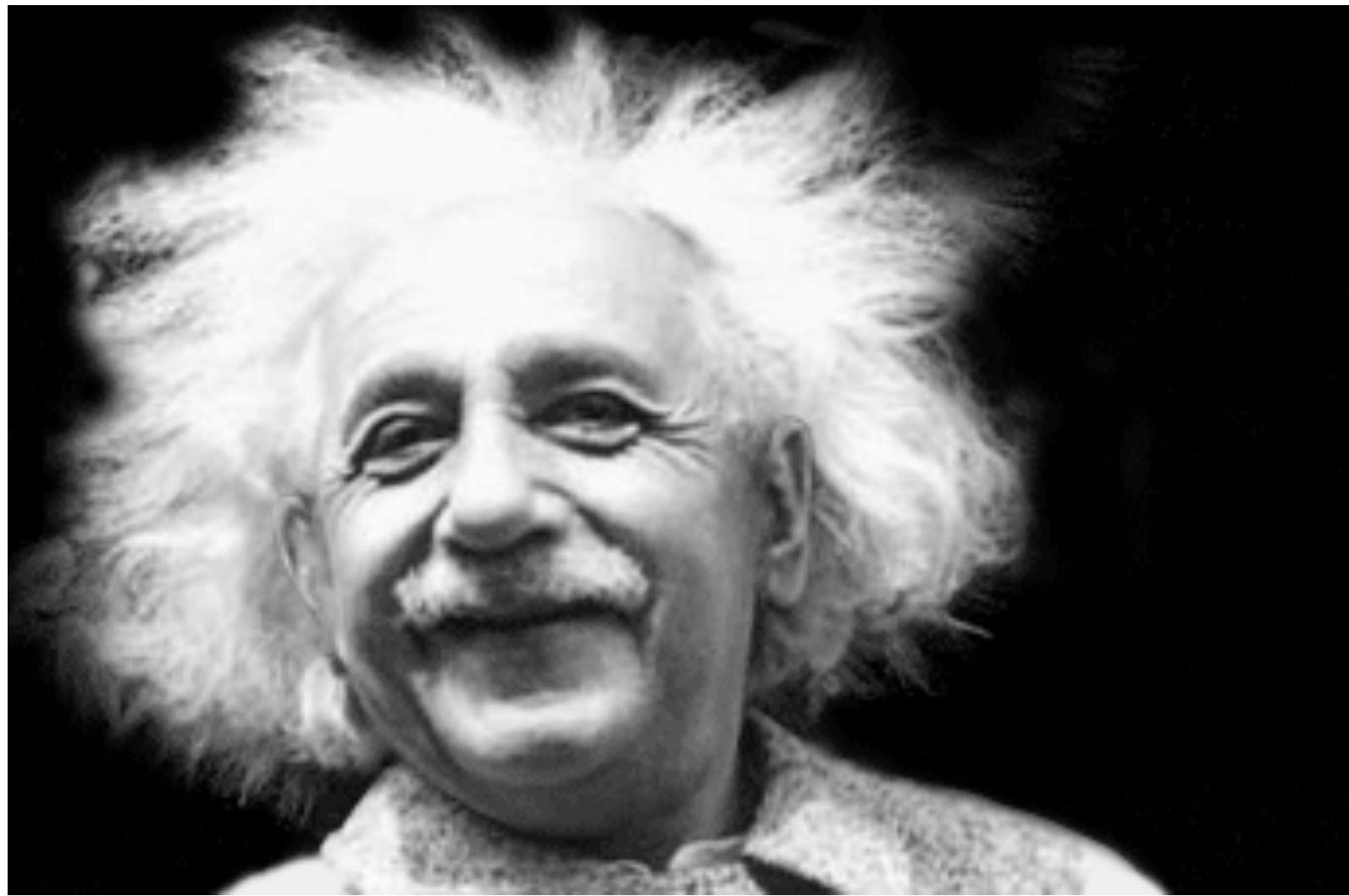
Technical information:

*** STOP: 0x0000009F (0x0000000000000003, 0xFFFFFA800FF75440, 0xFFFFF80000B9C3D8, 0xFFFFFA8010AB1270)

Collecting data for crash dump ...
Initializing disk for crash dump ...
Beginning dump of physical memory.
Dumping physical memory to disk: 5

How one prints 666?

- A. **def** beast = '6' *  PI
- B. **def** beast = '6' * '3'
- C. **def** beast = '667' - 1
- D. **def** beast = '6' + '6' + 0 + 6



```
def beast = '6' * Math.PI
```

```
def beast = '6' * 3.1415926...
```

This repository Search Explore Gist Blog Help jbaruch + ⌂ ⌂ ⌂ ⌂

groovy / groovy-core

Watch 106 Unstar 1,233 Fork 374

Groovy programming language <http://www.groovy-lang.org>

12,326 commits 35 branches 149 releases 129 contributors

branch: master groovy-core / +

Merge pull request #647 from christoph-frick/master ...

PascalSchumacher authored 6 hours ago latest commit 49c533de71

| File | Description | Time |
|--------------|--|---------------|
| benchmark | minor refactor: remove some checkstyle warnings | 8 months ago |
| buildSrc | minor refactor: remove some checkstyle warnings | 8 months ago |
| config | GROOVY-3457: Preparing for addition of new StreamingTemplateEngine | 7 months ago |
| gradle | Disable license and japicmp plugin since they still conflict with Art... | 9 hours ago |
| lib | Raw modifications to run Groovy on Android | 10 months ago |
| security | GROOVY-5305: Update dependencies (hsqldb) | 3 years ago |
| src | Merge pull request #647 from christoph-frick/master | 6 hours ago |
| subprojects | Documentation: add section on StreamingTemplateEngine | 2 days ago |
| xdocs/images | Remove the Maven 1 build files since we have now officially moved to ... | 8 years ago |
| .gitignore | Tests that fail when using @Grab with Extension Modules | 3 months ago |

Code Pull requests 32 Pulse Graphs

HTTPS clone URL <https://github.com/groovy/groovy-core> You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop Download ZIP

```
/**  
 * Repeat a String a certain number of times.  
 *  
 * @param self a String to be repeated  
 * @param factor the number of times the String should be repeated  
 * @return a String composed of a repetition  
 * @throws IllegalArgumentException if the number of repetitions is < 0  
 * @since 1.0  
 */  
public static String multiply(String self, Number factor) {  
    int size = factor.intValue();  
    ...  
}
```

```
def beast = '6' * 3.1415926...
```

```
def beast = '6' * 3
```

Pop Quiz!

- B. **def** beast = '6' * '3'
- C. **def** beast = '667' - 1
- D. **def** beast = '6' + '6' + 0 + 6



Pop Quiz!

- B. **def** beast = '6' * '3'
- C. **def** beast = '667' - 1
- D. **def** beast = '6' + '6' + (0 + 6)



Pop Quiz!

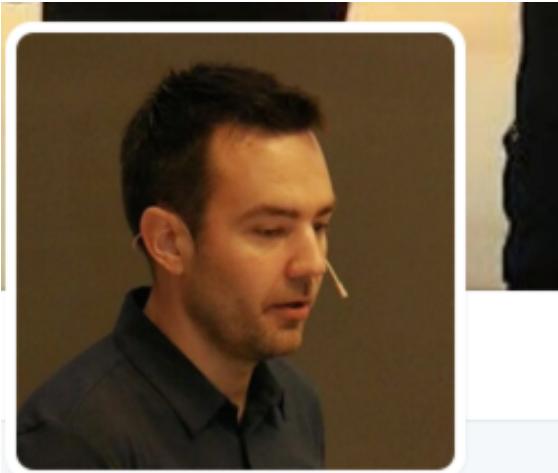
- B. **def** beast = '6' * '3'
- C. **def** beast = '667' - 1
- D. **def** beast = '6' + '6' + (0 + 6)



Yup, I am
watching you!



And the t-shirt goes to...



Lari Hotari

@lhotari FOLLOWS YOU

Software Engineer at [@Gradleware](#),
[#Grailsfw](#) core contributor, Software
Craftsman, [#DDDesign](#) & [#LeanStartup](#)
practitioner, [#InfoSec](#) paranoid, Seeking
simplicity

Ontario, Canada

github.com/lhotari

Joined April 2009



```
class THERE_CAN_BE_ONLY_ONE { }

class MacLeod {
    THERE_CAN_BE_ONLY_ONE getTHERE_CAN_BE_ONLY_ONE() {
        Class clazz = THERE_CAN_BE_ONLY_ONE
        return clazz.newInstance()
    }
}

println new MacLeod().THERE_CAN_BE_ONLY_ONE
```

```
class THERE_CAN_BE_ONLY_ONE { }

class MacLeod {
    THERE_CAN_BE_ONLY_ONE
    getTHERE_CAN_BE_ONLY_ONE() {
        Class clazz = THERE_CAN_BE_ONLY_ONE
        return clazz.newInstance()
    }
}

println new MacLeod().THERE_CAN_BE_ONLY_ONE
```

- A. Won't start
- B. No such property: THERE_CAN_BE_ONLY_ONE for class: MacLeod
- C. THERE_CAN_BE_ONLY_ONE@3d74bf60
- D. Another option





A. Multiple Compilation Errors Exception

B. Stack ~~overflow~~ flow Error

C. NullPointerException

D. Yet Another Exception



```
class MacLeod {  
  
    THERE_CAN_BE_ONLY_ONE getTHERE_CAN_BE_ONLY_ONE() {  
        Class clazz = THERE_CAN_BE_ONLY_ONE  
        return clazz.newInstance()  
    }  
  
}
```

```
class MacLeod {  
  
    THERE_CAN_BE_ONLY_ONE getTHERE_CAN_BE_ONLY_ONE() {  
        Class clazz = getTHERE_CAN_BE_ONLY_ONE()  
        return clazz.newInstance()  
    }  
  
}
```

Let's Fix It!

```
Class<THERE_CAN_BE_ONLY_ONE> clazz = THERE_CAN_BE_ONLY_ONE.class
```

```
class MacLeod {  
  
    THERE_CAN_BE_ONLY_ONE getTHERE_CAN_BE_ONLY_ONE() {  
        Class<THERE_CAN_BE_ONLY_ONE> clazz = THERE_CAN_BE_ONLY_ONE.class  
        return clazz.newInstance()  
    }  
  
}
```



```
class MacLeod {  
  
    THERE_CAN_BE_ONLY_ONE getTHERE_CAN_BE_ONLY_ONE() {  
        Class<THERE_CAN_BE_ONLY_ONE> clazz = getTHERE_CAN_BE_ONLY_ONE().class  
        return clazz.newInstance()  
    }  
  
}
```

Let's Fix It!

```
Class<THERE_CAN_BE_ONLY_ONE> clazz = (THERE_CAN_BE_ONLY_ONE as Class)
```

```
class MacLeod {  
  
    THERE_CAN_BE_ONLY_ONE getTHERE_CAN_BE_ONLY_ONE() {  
        Class<THERE_CAN_BE_ONLY_ONE> clazz = (THERE_CAN_BE_ONLY_ONE as Class)  
        return clazz.newInstance()  
    }  
  
}
```



```
class MacLeod {  
  
    THERE_CAN_BE_ONLY_ONE getTHERE_CAN_BE_ONLY_ONE() {  
        Class<THERE_CAN_BE_ONLY_ONE> clazz = (getTHERE_CAN_BE_ONLY_ONE() as Class)  
        return clazz.newInstance()  
    }  
}
```

Let's Fix It!



And the t-shirt goes to...

Толкачёв Кирилл 🔒

@tolkv FOLLOWS YOU

Joined March 2011

A close-up, slightly grainy portrait of Captain Jack Sparrow. He has his signature wild, curly brown hair and a thick, dark beard. His eyes are a light blue-grey, looking directly at the viewer with a mischievous expression. A wooden toothpick is stuck in his upper lip. He is wearing a dark, textured jacket with a small circular emblem on the left chest. The background is a blurred green, suggesting a jungle or forest setting.

MISSED ME?

```
@groovy.transform.InheritConstructors  
class TreaayeMap extends HashMap {  
}
```

```
TreaayeMap a = [5]
```

```
TreaayeMap b = [6]
```

```
println "${a.getClass()} ${a.equals(b)}"
```

- A. class HashMap true
- B. class TreaayeMap false

- C. class Tre~~oovy~~Map true
- D. class HashMap false





HATE THE CODE



RTFM!

List and map constructors

In addition to the assignment rules above, if an assignment is deemed invalid, in type checked mode, a *list* literal or a *map* literal **A** can be assigned to a variable of type **T** if:

- the assignment is a variable declaration and **A** is a list literal and **T** has a constructor whose parameters match the types of the elements in the list literal
- the assignment is a variable declaration and **A** is a map literal and **T** has a no-arg constructor and a property for each of the map keys

```
TreayeMap a = [5]
```

List and map constructors

In addition to the assignment rules above, if an assignment is deemed invalid, in type checked mode, a *list* literal or a *map* literal

A can be assigned to a variable of type T if:

- the assignment is a variable declaration and A is a list literal and T has a constructor whose parameters match the types of the elements in the list literal
- the assignment is a variable declaration and A is a map literal and T has a no-arg constructor and a property for each of the map keys

TreayMap a = [5]

List and map constructors

In addition to the assignment rules above, if an assignment is deemed invalid, in type checked mode, a *list* literal or a *map* literal

A can be assigned to a variable of type T if:

- the assignment is a variable declaration and A is a list literal and T has a constructor whose parameters match the types of the elements in the list literal
- the assignment is a variable declaration and A is a map literal and T has a no-arg constructor and a property for each of the map keys

TreayMap a = [5]

List and map constructors

In addition to the assignment rules above, if an assignment is deemed invalid, in type checked mode, a *list* literal or a *map* literal

A can be assigned to a variable of type T if:

- the assignment is a variable declaration and A is a list literal and T has a constructor whose parameters match the types of the elements in the list literal
- the assignment is a variable declaration and A is a map literal and T has a no-arg constructor and a property for each of the map keys

```
TreaayeMap a = [5]
```

List and map constructors

In addition to the assignment rules above, if an assignment is deemed invalid, in type checked mode, a *list* literal or a *map* literal

A can be assigned to a variable of type **T** if:

- the assignment is a variable declaration and **A** is a list literal and **T** has a constructor whose parameters match the types of the elements in the list literal
- the assignment is a variable declaration and **A** is a map literal and **T** has a no-arg constructor and a property for each of the map keys

```
@groovy.transform.InheritConstructors  
class TreaayeMap extends HashMap {  
}
```

```
/**  
 * Constructs an empty <tt>HashMap</tt> with the specified initial  
 * capacity and the default load factor (0.75).  
 *  
 * @param initialCapacity the initial capacity.  
 * @throws IllegalArgumentException if the initial capacity is negative.  
 */
```

```
public HashMap(int initialCapacity)
```

TreaayeMap a = [5]

List and map constructors

In addition to the assignment rules above, if an assignment is deemed invalid, in type checked mode, a *list* literal or a *map* literal

A can be assigned to a variable of type T if:

- the assignment is a variable declaration and A is a list literal and T has a constructor whose parameters match the types of the elements in the list literal
- the assignment is a variable declaration and A is a map literal and T has a no-arg constructor and a property for each of the map keys

@groovy.transform.InheritConstructors
class TreaayeMap extends HashMap {
}

equals() doesn't care about capacity, mappings only

```
/*
public boolean equals(Object o) {
    if (o == this)
        return true;
    if (!(o instanceof Map))
        return false;
    Map<?, ?> m = (Map<?, ?>) o;
    if (m.size() != size())
        return false;
    try {
        Iterator<Map.Entry<?, ?> > i1 = entrySet().iterator();
        Iterator<Map.Entry<?, ?> > i2 = m.entrySet().iterator();
        while (i1.hasNext() && i2.hasNext()) {
            Map.Entry<?, ?> e1 = i1.next();
            Map.Entry<?, ?> e2 = i2.next();
            if (!e1.getKey().equals(e2.getKey()))
                return false;
            if (!e1.getValue().equals(e2.getValue()))
                return false;
        }
        return true;
    }
}
```

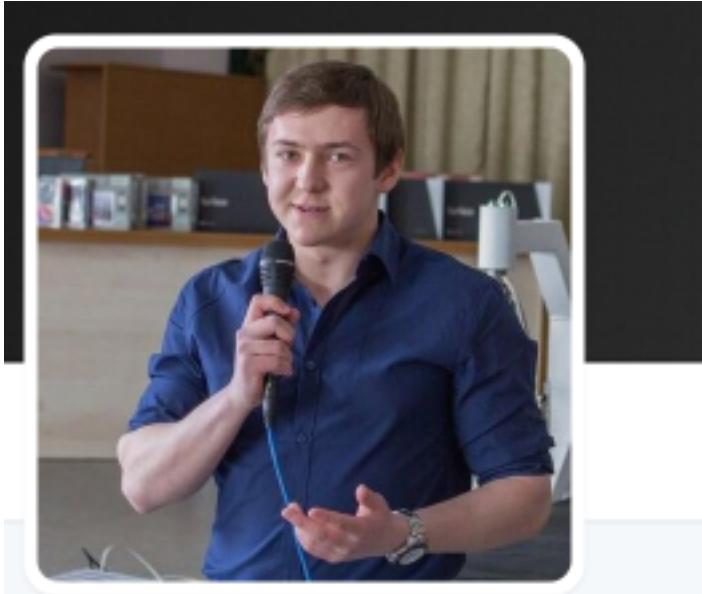
Documentation for equals(Object)

java.util.AbstractMap

public boolean equals(@Nullable java.lang.Object o)

Compares the specified object with this map for equality. Returns true if the given object is also a map and the two maps represent the same mappings. More formally, two maps m_1 and m_2 represent the same mappings if $m_1.entrySet().equals(m_2.entrySet())$. This ensures that the equals method works properly across different implementations of the Map interface.

And the t-shirt goes to...



Sergey Tselovalnikov

@SerCeMan

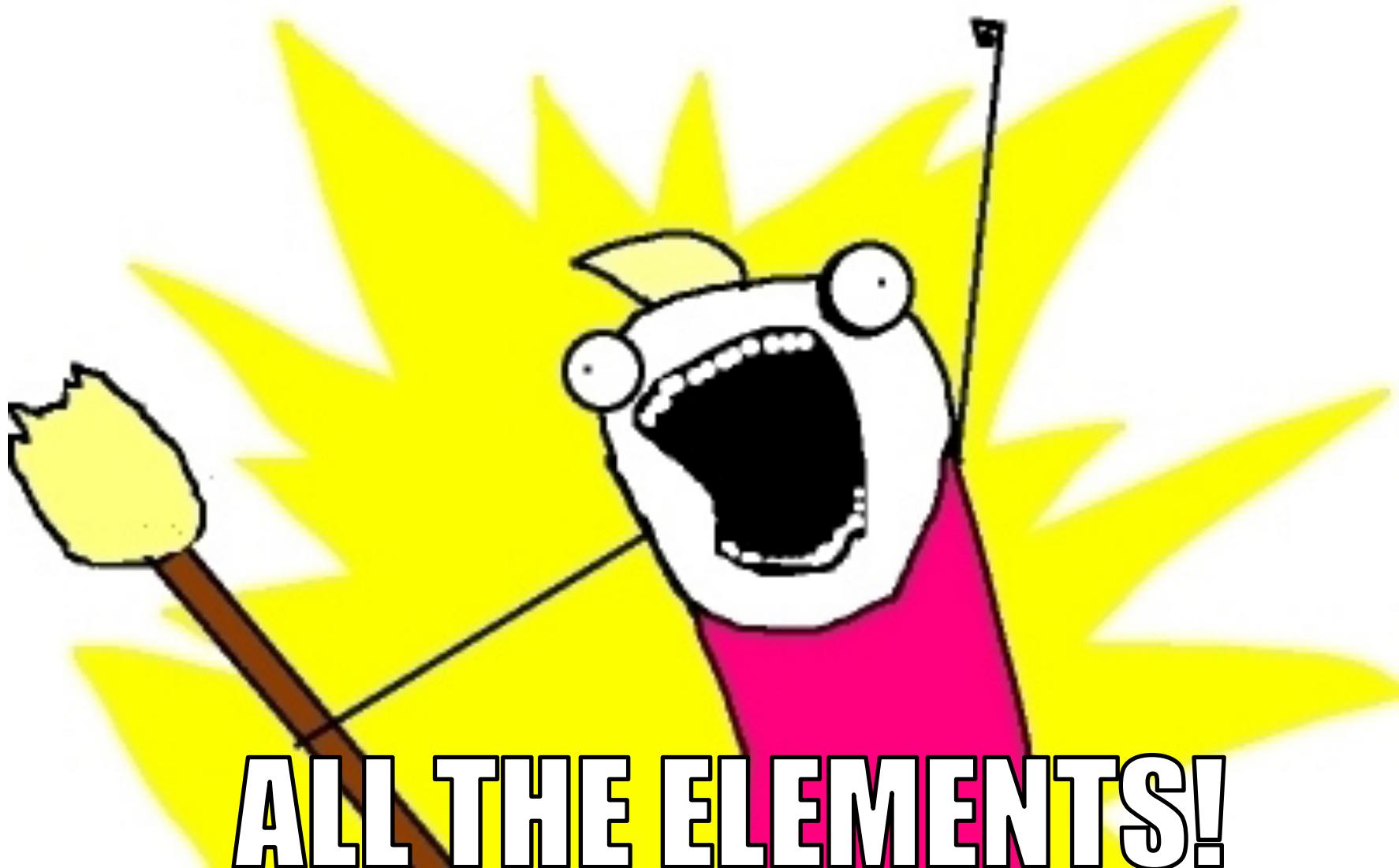
Love IT, love other interesting things.

Develop in Java, Groovy, C++. JUG.EKB
(jugekb.ru) creator.

📍 Saint-Petersburg

⌚ Joined November 2010

COLLECT



ALL THE ELEMENTS!

```
class Kitty { def fur }  
def kitties = [new Kitty(fur: 'soft'),new Kitty(fur: 'warm'),new Kitty(fur: 'purr') ]  
  
println kitties.collect { it.fur }  
println kitties*.fur  
println kitties.fur
```

How many of the printed lines will be the same?

- A. All different
- B. 2 similar, one different
- C. All the same
- D. Won't run





```
class Kitty { def fur }  
def kitties  
  
println kitties.collect { it.fur }  
println kitties*.fur  
println kitties.fur
```

How many of the printed lines will be the same?

- A. All  Groovy
- B. 2 similar, one different
- C. All the same
- D. Won't compile



```
class Kitty { def fur }  
def kitties  
  
println kitties.fur
```

```
class Kitty { def fur }  
def kitties  
  
println kitties.fur
```



```
class Kitty { def fur }
def kitties

println kitties.collect { it.fur }
```

```
class Kitty { def fur }
def kitties
```

```
println kitties.collect { it.fur }
```

java.lang
Class Object
java.lang.Object

public class Object
Class object is the root of the class hierarchy. Every class has object as a superclass. All objects, including arrays, implement the methods of this class.
Since:
JDK1.0
See Also:
Class

Constructor Summary

| Constructors |
|-----------------------------|
| Constructor and Description |
| Object() |

Method Summary

| Modifier and Type | Method and Description |
|-------------------|--|
| protected Object | clone() Creates and returns a copy of this object. |
| boolean | equals(Object obj) Indicates whether some other object is "equal to" this one. |
| protected void | finalize() Called by the garbage collector on an object when garbage collection determines that there are no more references to the object. |
| Class<?> | getClass() Returns the runtime class of this object. |
| int | hashCode() Returns a hash code value for the object. |
| void | notify() Wakes up a single thread that is waiting on this object's monitor. |
| void | notifyAll() Wakes up all threads that are waiting on this object's monitor. |
| String | toString() Returns a string representation of the object. |
| void | wait() Causes the current thread to wait until another thread invokes the <code>notify()</code> method or the <code>notifyAll()</code> method for this object. |
| void | wait(long timeout) Causes the current thread to wait until either another thread invokes the <code>notify()</code> method or the <code>notifyAll()</code> method for this object, or a specified amount of time has elapsed. |
| void | wait(long timeout, int nanos) Causes the current thread to wait until another thread invokes the <code>notify()</code> method or the <code>notifyAll()</code> method for this object, or some other thread interrupts the current thread, or a certain amount of real time has elapsed. |

```
class Kitty { def fur }
def kitties
```

```
println kitties.collect { it.fur }
```

1.8.2. Instance methods

To add an instance method to an existing class, you need to create an extension class. For example, let's say you want to add a `maxRetries` method on `Integer` which accepts a closure and executes it at most n times until no exception is thrown. To do that, you only need to write the following:

MaxRetriesExtension.groovy

```
class MaxRetriesExtension {
    static void maxRetries(Integer self, Closure code) {
        int retries = 0
        Throwable e
        while (retries<self) {
            try {
                code.call()
                break
            } catch (Throwable err) {
                e = err
                retries++
            }
        }
        if (retries==0 && e) {
            throw e
        }
    }
}
```

①

②

① The extension class

② First argument of the static method corresponds to the receiver of the message, that is to say the extended instance



```
class Kitty { def fur }
def kitties

println kitties.collect { it.fur }
```

```
public static <T> List<T> collect(Object self, Closure<T> transform) {
    Collection collection = self.asList()
    ...
}
```

```
class Kitty { def fur }
def kitties

    println kitties.collect { it.fur }

public static Collection asCollection(Object self) {
    if (self == null) {
        return Collections.EMPTY_LIST;
    }
    ...
}
```

```
class Kitty { def fur }
def kitties

println kitties*.fur
```

```
class Kitty { def fur }
def kitties

println kitties*.fur
```

The spread operator is null-safe, meaning that if an element of the collection is null, it will return null instead of throwing a [NullPointerException](#):

```
class Kitty { def fur }  
def kitties  
  
println kitties*.fur
```

The spread operator is null-safe, meaning that if an element of the collection is null, it will not throw a [NullPointerException](#):



Consistency, yeah.

[]

null

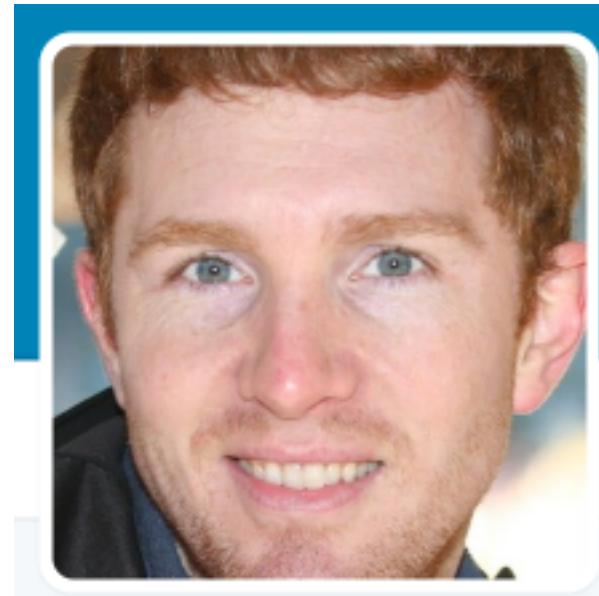
Caught: java.lang.NullPointerException:
Cannot get property 'fur' on null object

“All happy families are alike; each unhappy family is unhappy in its own way.”



Leo Tolstoy, “Anna Karenina”

And the t-shirt goes to...



Dan Tanner

@edgescope

mislabeled

📍 Minneapolis MN US

🕒 Joined April 2009



```
ArrayList<String> expendables = ['Arnold', 'Chuck', Sly']
def expendable = //someone from the list

for(String hero in expendables) {
    if(hero == expendable){
        expendables.remove(hero)
    }
}

println expendables
```

Which one won't cause a ConcurrentModificationException?

```
ArrayList<String> expendables = ['Arnold', 'Chuck', Sly']
def expendable = //someone from the list

for(String hero in expendables) {
    if(hero == expendable){
        expendables.remove(hero)
    }
}

println expendables
```

- A.Can't avoid CME
- B.Arnold
- C.C
- D.Sly



CONCURRENTMODIFICATIONEXCEPTION?



NOPE, NEVER HEARD ABOUT IT.

You have been warned.

```
public class ConcurrentModificationException  
extends RuntimeException
```

This exception may be thrown by methods that have detected concurrent modification of an object when such modification is not permissible.

For example, it is not generally permissible for one thread to modify a Collection while another thread is iterating over it. In general, the results of the iteration are undefined under these circumstances. Some Iterator implementations (including those of all the general purpose collection implementations provided by the JRE) may choose to throw this exception if this behavior is detected. Iterators that do this are known as *fail-fast* iterators, as they fail quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Note that this exception does not always indicate that an object has been concurrently modified by a *different* thread. If a single thread issues a sequence of method invocations that violates the contract of an object, the object may throw this exception. For example, if a thread modifies a collection directly while it is iterating over the collection with a fail-fast iterator, the iterator will throw this exception.

Note that fail-fast behavior cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast operations throw `ConcurrentModificationException` on a **best-effort basis**. Therefore, it would be wrong to write a program that depended on this exception for its correctness: `ConcurrentModificationException` should be used only to detect bugs.

Let's decompile this, baby!

```
List expendables = Arrays.asList(new String[]{"Arnold", "Chuck", "Sly"});  
String expendable = "Chuck";  
Iterator iterator = expendables.iterator();  
  
while(iterator.hasNext()) {  
    String hero = (String)iterator.next();  
    if(hero.equals(expendable)) {  
        expendables.remove(hero);  
    }  
}
```

```
while(iterator.hasNext()) {  
    String hero = (String) iterator.next();  
    if(hero.equals(expendable)) {  
        //      expendables.remove(hero);  
    }  
}
```

```
while(iterator.hasNext()) {  
    String hero = (String)iterator.next();  
    if(hero.equals(expendable)) {  
        //      expendables.remove(hero);  
    }  
}  
  
public E next() {  
    checkForComodification();  
    ...  
    cursor = i + 1;  
}  
  
public boolean hasNext() {  
    return cursor != size();  
}
```

Modifications are only checked in the next cycle

Getting ready for hasNext() check in the next cycle

Exit on last element +1 == size()

```
while(iterator.hasNext()) {  
    String hero = (String)iterator.next();  
    if(hero.equals(expendable)) {  
        //      expendables.remove(hero);  
    }  
}
```

```
public E next() {  
    checkForComodification();  
    ...  
    cursor = i + 1; }  
}
```

After Sly cursor is 3

```
public boolean hasNext() {  
    return cursor != size();  
}
```

 And size is 3 as well

All good. Now let's mess with it.

```
while(iterator.hasNext()) {  
    String hero = (String)iterator.next();  
    if(hero.equals(expendable)) {  
        expendables.remove(hero);  
    }  
}
```

```
public E next() {  
    checkForComodification();  
    ...  
    cursor = i + 1;  
}
```

```
public boolean hasNext() {  
    return cursor != size();  
}
```

After Chuck the cursor is 2

Then we remove the element

And now the size now is 2!

It won't get to the next() to run checkForComodification!

Hey, what about me?!



```
while(iterator.hasNext()) {  
    String hero = (String)iterator.next();  
    if(hero.equals(expendable)) {  
        expendables.remove(hero);  
    }  
}
```

```
public E next() {  
    checkForComodification();  
    ...  
    cursor = i + 1;  
}
```

```
public boolean hasNext() {  
    return cursor != size();  
}
```

After Sly the cursor is 3

Then we remove the element

But the size now is 2!

It will go to another loop and fail on checkForComodification!

Yup, you just saw the most awful
puzzler. And it's not even Groovy.



And the t-shirt goes to...



Evgeny Borisov

Senior Java Consultant at Trainologic

Israel | Education Management

Current Democracy Startup, Trainologic, JFrog Ltd

Previous IDI Israel, AlphaCSP

Education Polytechnic

Schwarzenegger

True Lies



DVD

DIGITALLY THX MASTERED
FOR SUPERIOR SOUND AND PICTURE QUALITY

```
String truth = 'false'  
boolean groovyTruth = truth  
println groovyTruth
```

```
String truth = 'false'  
boolean groovyTruth = truth  
println groovyTruth
```

A.false



C.ClassCastException

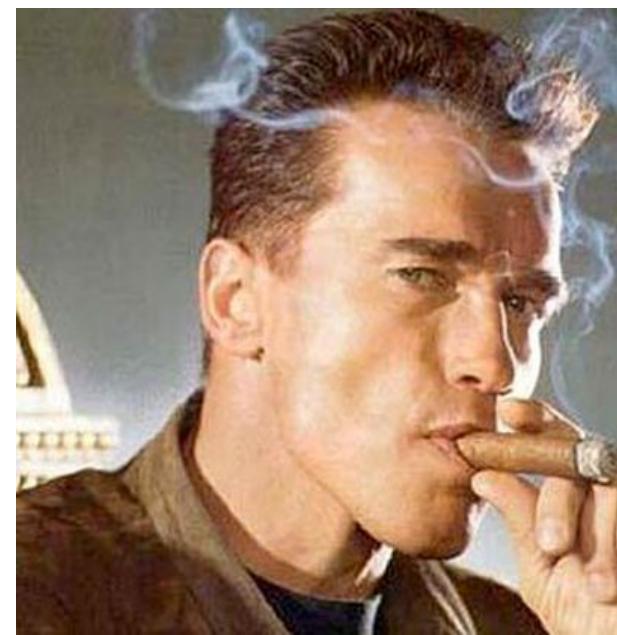
D.Startup error



DGDO | ~



RTFM!



§ Strings

Non-empty Strings, GStrings and CharSequences are coerced to true.

```
assert 'a'  
assert ''  
def nonEmpty = 'a'  
assert "$nonEmpty"  
def empty = ''  
assert !"empty"
```

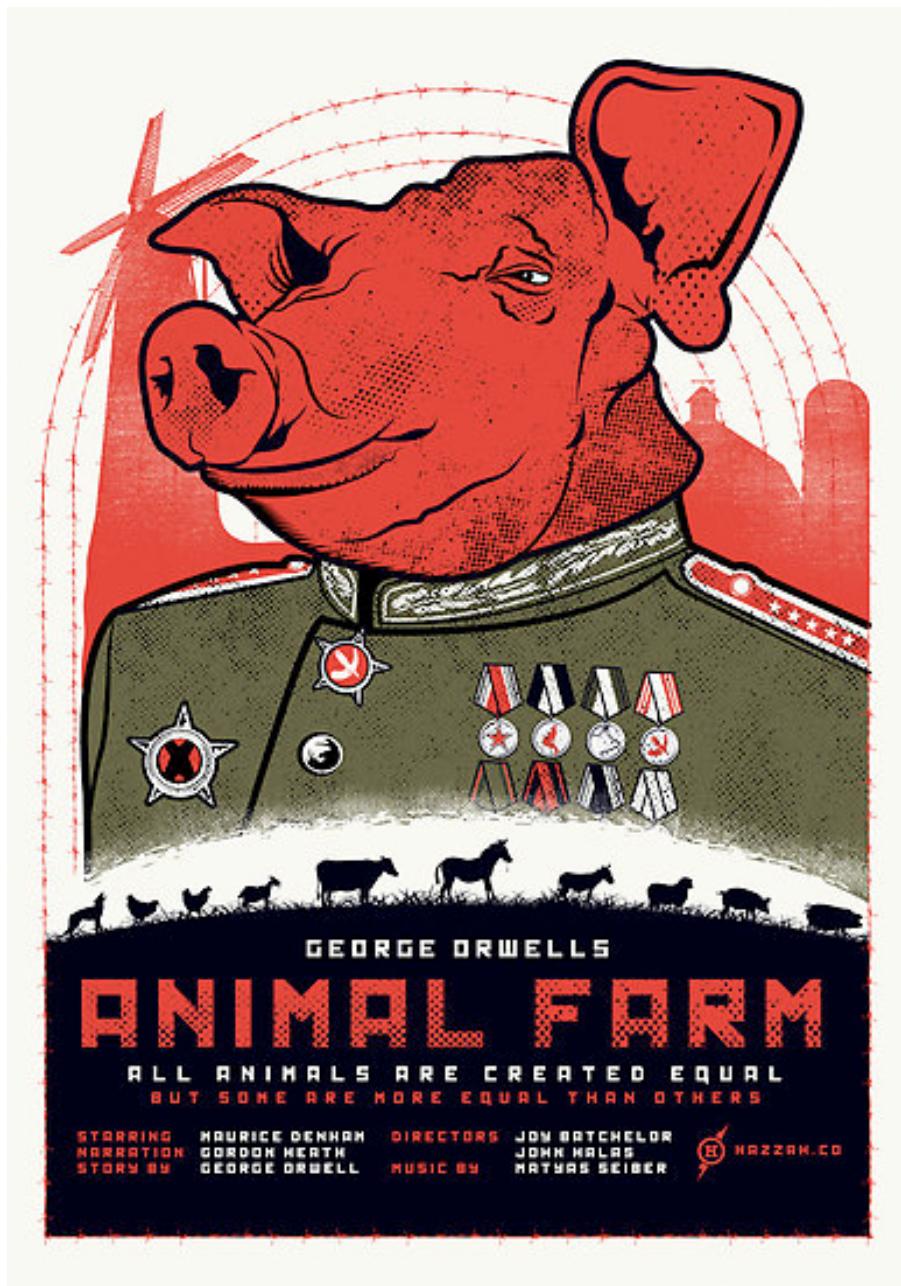
And the t-shirt goes to...



Andrey Hihlovskiy
@AndreyHihlovski FOLLOW YOU

Software Engineer at startext GmbH.
Groovy, Gradle, Spring Framework. V/I is futile. github.com/akhikhl

📍 Bonn, Germany
🔗 akhikhl.org



```
assert 1L == 1  
println 1L.equals(1)
```

- A. Assertion failed
- B. true
- C. Groovy
- D. MissingMethodException





RTFM!

10. Behaviour of `==`

In Java `==` means equality of primitive types or identity for objects. In Groovy `==` translates to `a.compareTo(b)==0`, iff they are `Comparable`, and `a.equals(b)` otherwise. To check for identity, there is `is`. E.g. `a.is(b)`.

```
/**  
 * Compare two Numbers. Equality (==) for numbers dispatches to this.  
 *  
 * @param left a Number  
 * @param right another Number to compare to  
 * @return the comparison of both numbers  
 * @since 1.0  
 */  
public static int compareTo(Number left, Number right) {  
    /* @todo maybe a double dispatch thing to handle new large numbers? */  
    return NumberMath.compareTo(left, right);  
}
```

```
/**  
 * Compares this object to the specified object. The result is  
 * {@code true} if and only if the argument is not  
 * {@code null} and is a {@code Long} object that  
 * contains the same {@code long} value as this object.  
 *  
 * @param obj the object to compare with.  
 * @return {@code true} if the objects are the same;  
 *         {@code false} otherwise.  
 */  
public boolean equals(Object obj) {  
    if (obj instanceof Long) {  
        return value == ((Long)obj).longValue();  
    }  
    return false;  
}
```

And the t-shirt goes to...



Ron Dahlgren

@ScaleItRon

Tweets generally regarding Linux, D&D, SNES, PS4, JRPGs, SRPGs, Software, and science news. PGP - FB3E4542

Mechanical Island

dahlgren.so

Joined April 2013



```
def numbers = [[2, 3, 5][2, 4, 8][42, 73, Integer.MAX_VALUE, 0]]  
println numbers
```

```
def numbers = [[2, 3, 5], [2, 4, 8], [42, 73, Integer.MAX_VALUE, 0]]  
println numbers
```

- A. [[2, 3, 5], [2, 4, 8], [42, 73, 2147483647, 0]]
- B. Won't run
- C. [[null, , null, 5]]
- D. null

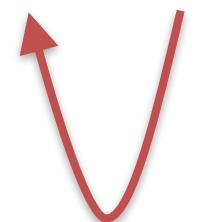


ReactionGIFS.me

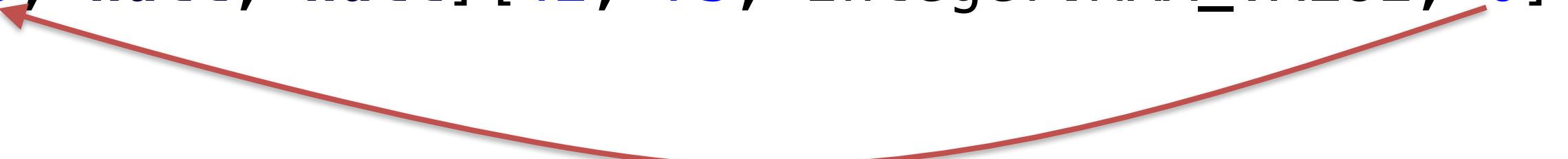
```
def numbers = [[2, 3, 5][2, 4, 8][42, 73, Integer.MAX_VALUE, 0]]  
println numbers
```

```
assert ['a', 'b', 'c'][0,2]==['a', 'c']
```

```
[2, 3, 5][2, 4, 8]
```

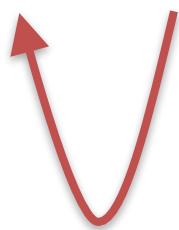


```
[5, null, null][42, 73, Integer.MAX_VALUE, 0]
```



```
def numbers = [[2, 3, 5][2, 4, 8][42, 73, Integer.MAX_VALUE, 0]]  
println numbers
```

[2, 3, 5] [2, 4, 8]



[5, null, null] [42, 73, Integer.MAX_VALUE, 0]



[[null, null, null, 5]]

No t-shirt for this guy ☹

groovy spock

share edit close flag

asked Apr 8 at 17:44



orbfish

2,157 • 4 ● 22 ● 40

We used this awesome question as a puzzler in the second season of the #groovypuzzlers and want to send you a thank-you t-shirt. How can I reach you? – [JBaruch Sep 13 at 4:49](#)

[add a comment](#)

[start a bounty](#)



```
def jailhouseRock
def loveMeTender
def rockAroundTheClock = [1,2,3]

jailhouseRock? : [] + loveMeTender? : [] + rockAroundTheClock? : []
```

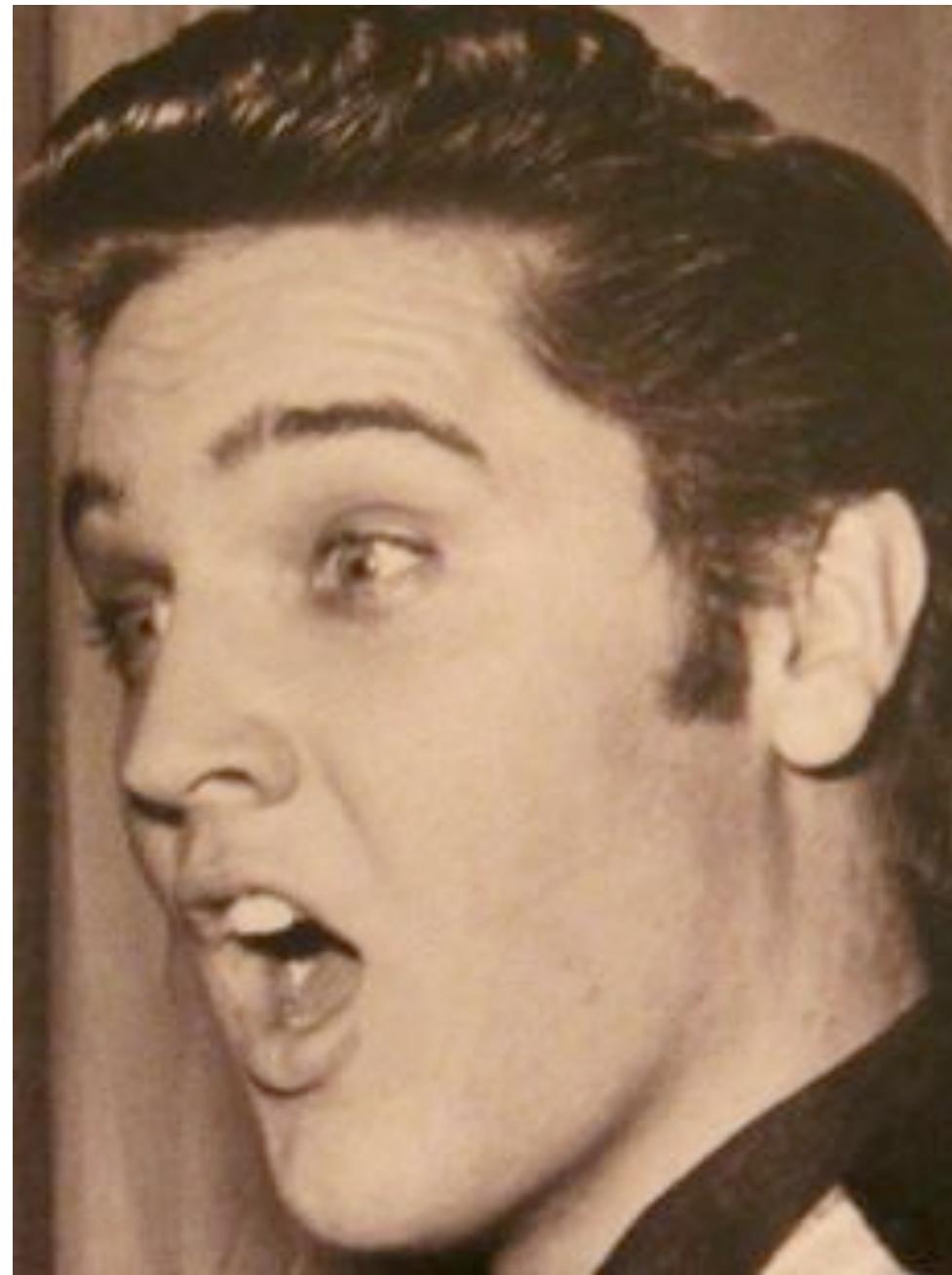
*Yes, we know, it's not Elvis' song.

```
def jailhouseRock  
def loveMeTender  
def rockAroundTheClock = [1,2,3]
```

```
jailhouseRock?[:] + loveMeTender?[:] + rockAroundTheClock?[:]
```



- A.** Groovy
- B.** null
- C.** []
- D.** [1,2,3]



```
def jailhouseRock
def loveMeTender
def rockAroundTheClock = [1,2,3]

jailhouseRock? : [] + loveMeTender? : [] + rockAroundTheClock? : []
```

```
def jailhouseRock  
def loveMeTender  
def rockAroundTheClock = [1,2,3]
```

```
jailhouseRock?: [null]? : [] + rockAroundTheClock?: []
```

```
def jailhouseRock  
def loveMeTender  
def rockAroundTheClock = [1,2,3]
```

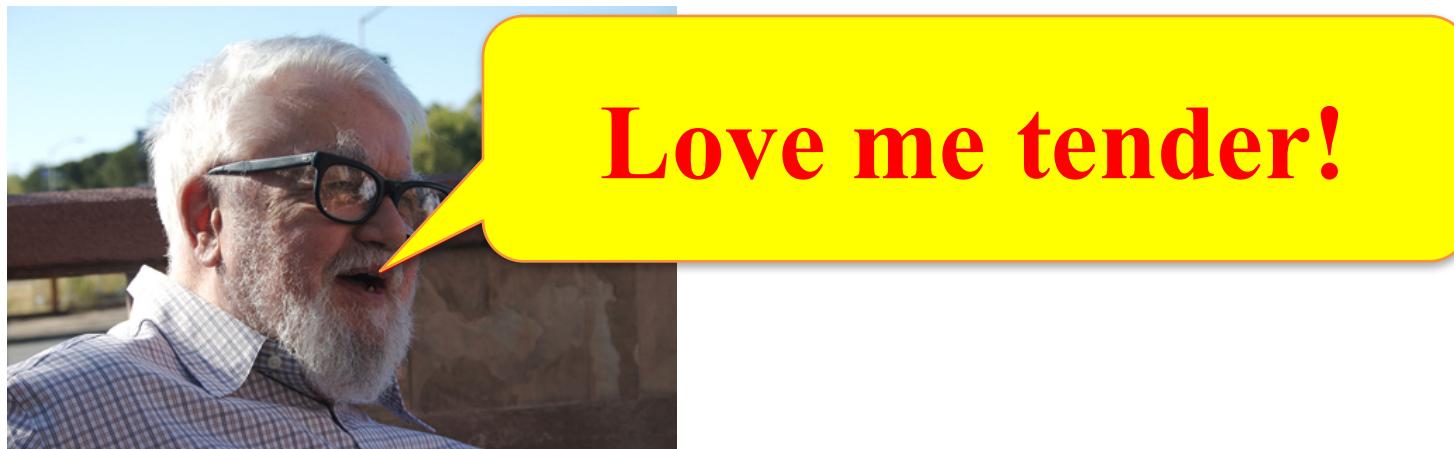
```
jailhouseRock?: [null]?:[1,2,3]?[]
```

```
def jailhouseRock  
def loveMeTender  
def rockAroundTheClock = [1,2,3]
```

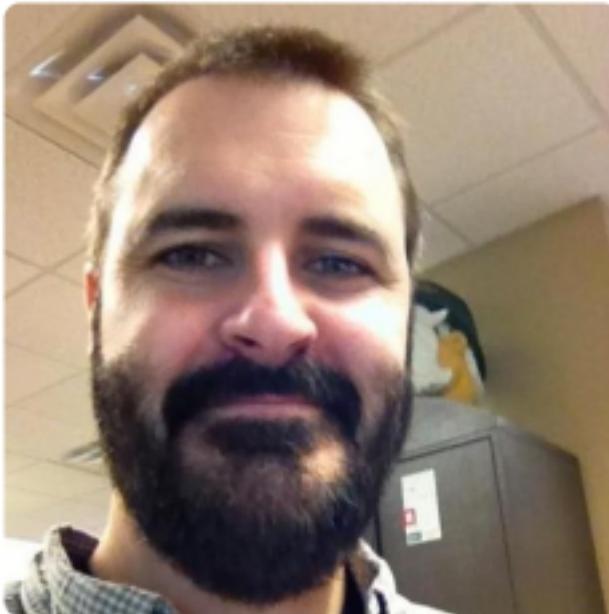
```
[null]
```

Let's fix it

(jailhouseRock?: [])+(loveMeTender?: [])+(rockAroundTheClock?: [])



And the t-shirt goes to...

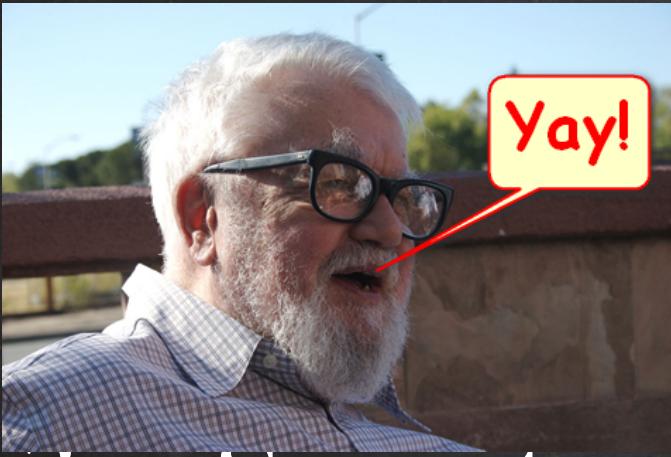


Chris Mihalcik
CMihalcik

 Cassidian Communications
 Franklin, TN
 Joined on 15 Mar 2010

Conclusions





adable code
nt neat tricks

3. Sometimes it is just a bug
4. Use static code analysis
5. Rtfm
6. Parentheses. Always use Parentheses.



We keep going! (Look at the awesome t-shirts!)

Puzzlers? Gotchas? Fetal position inducing behavior?

- puzzlers jfrog@com
- @Groovypuzzlers

Jfrog always pays its debts

 **Deigote**
@deigote

As promised by @NoamTenne, @jfrog pays its debts :-D t-shirt received for sending them a #groovylang puzzle. Thanks!

[Reply](#) [Retweeted](#) [Favorited](#) [More](#)



 **Iván López**
@ilopmar

I've received an amazing t-shirt from @jfrog for sending them a #Groovylang puzzler. Thank you @NoamTenne :-)

[Reply](#) [Retweeted](#) [Favorite](#) [More](#)



Positive feedback?

Fill the feedback form!

Praise us on twitter

#groovypuzzlers

- @Groovypuzzlers
- @gamussa
- @baruch

Negative feedback?

/dev/null

No, Thank you!

