



Эпичные Груви
Паззлеры
Второй Сезон:
Месть Скобок!

Ты кто такой?

github.com/jbaruch
linkd.in/jbaruch



stackoverflow.com/users/402053/jbaruch

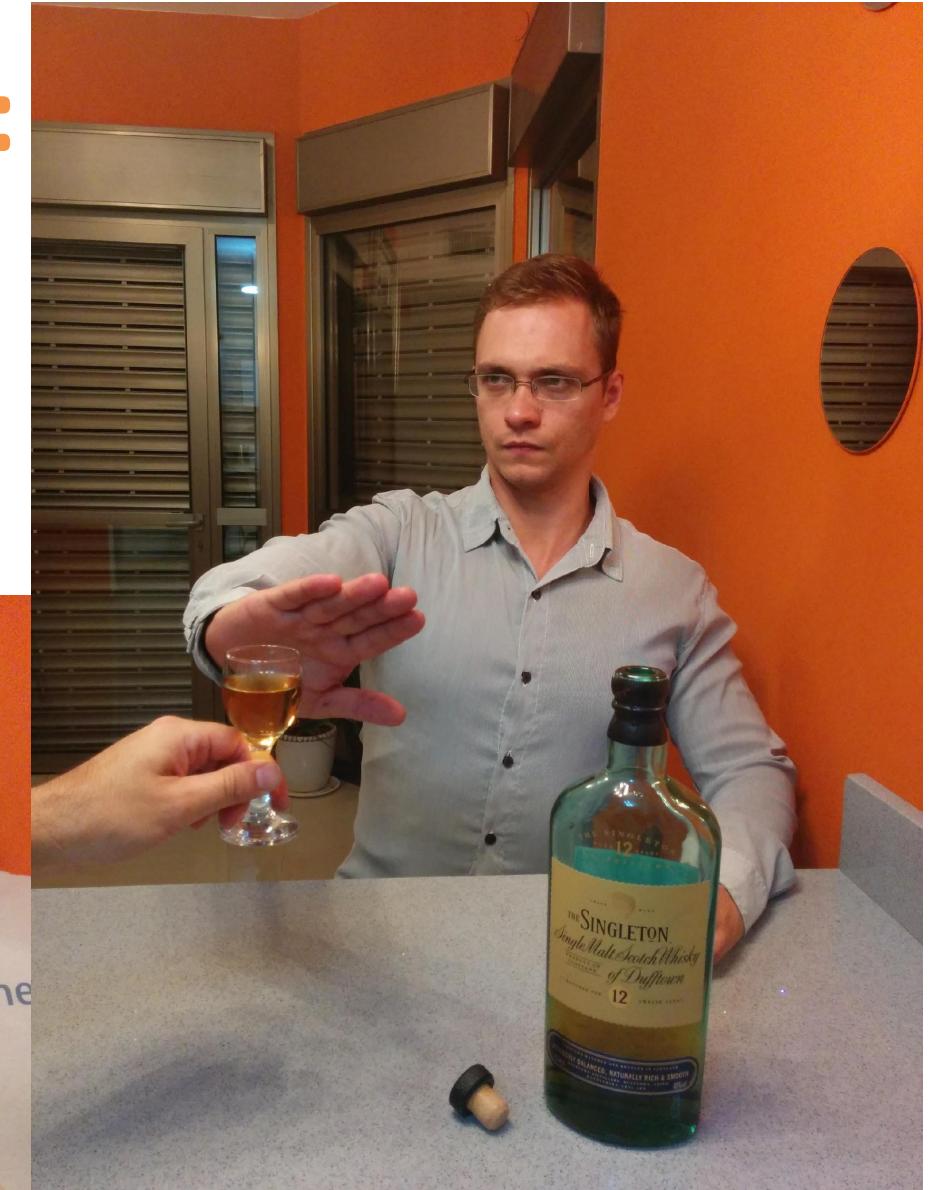
Ты кто такой?

jeka@inwhite.pro
@jekaborisov



linkedin.com/in/evborisov

Вместе мы:



Первый Сезон Хорошо Пошёл!



CLICK AND HACK



THE TYPING BROTHERS

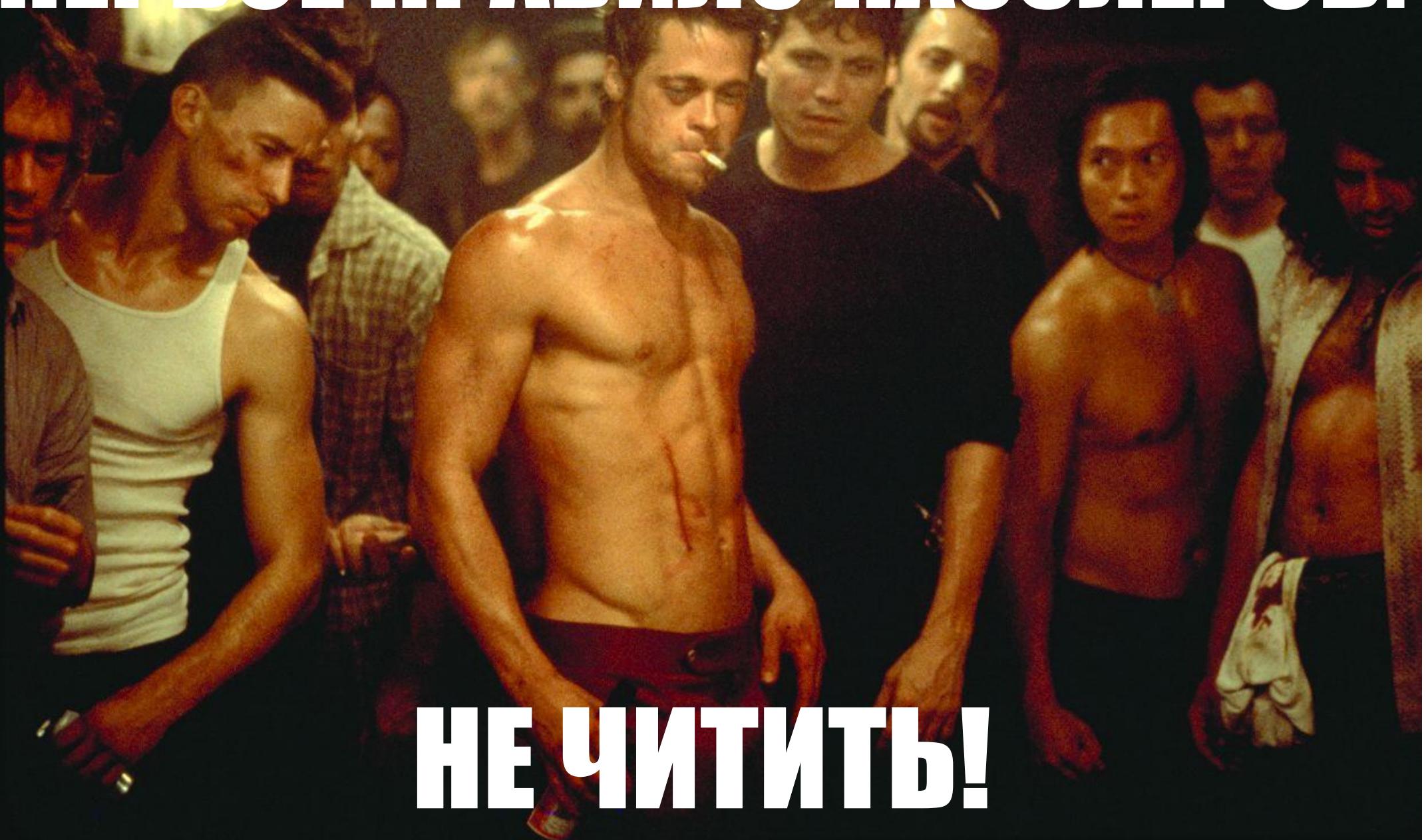
Ну этих точно никто не знает...



1. Два клевых пацана на сцене
2. Смешные задачки
3. Вы голосуете! Да-да!
4. Новые клевые футболки летят в зал! Лягушки тоже.
5. У нас есть хэштег!
groovypuzzlers

#

ПЕРВОЕ ПРАВИЛО ПАЗЗЛЕРОВ:



НЕ ЧИТИТЬ!

Все работает (т.е. не работает) на версии 2.4.3







```
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. Никаких Бахов
- B. О,  Бах
- C. Два Баха
- D. Три Баха



**В рантайме отрабатывают только closures,
всё остальное инлайнится**

**В рантайме отрабатывают только closures,
всё остальное инлайнится**

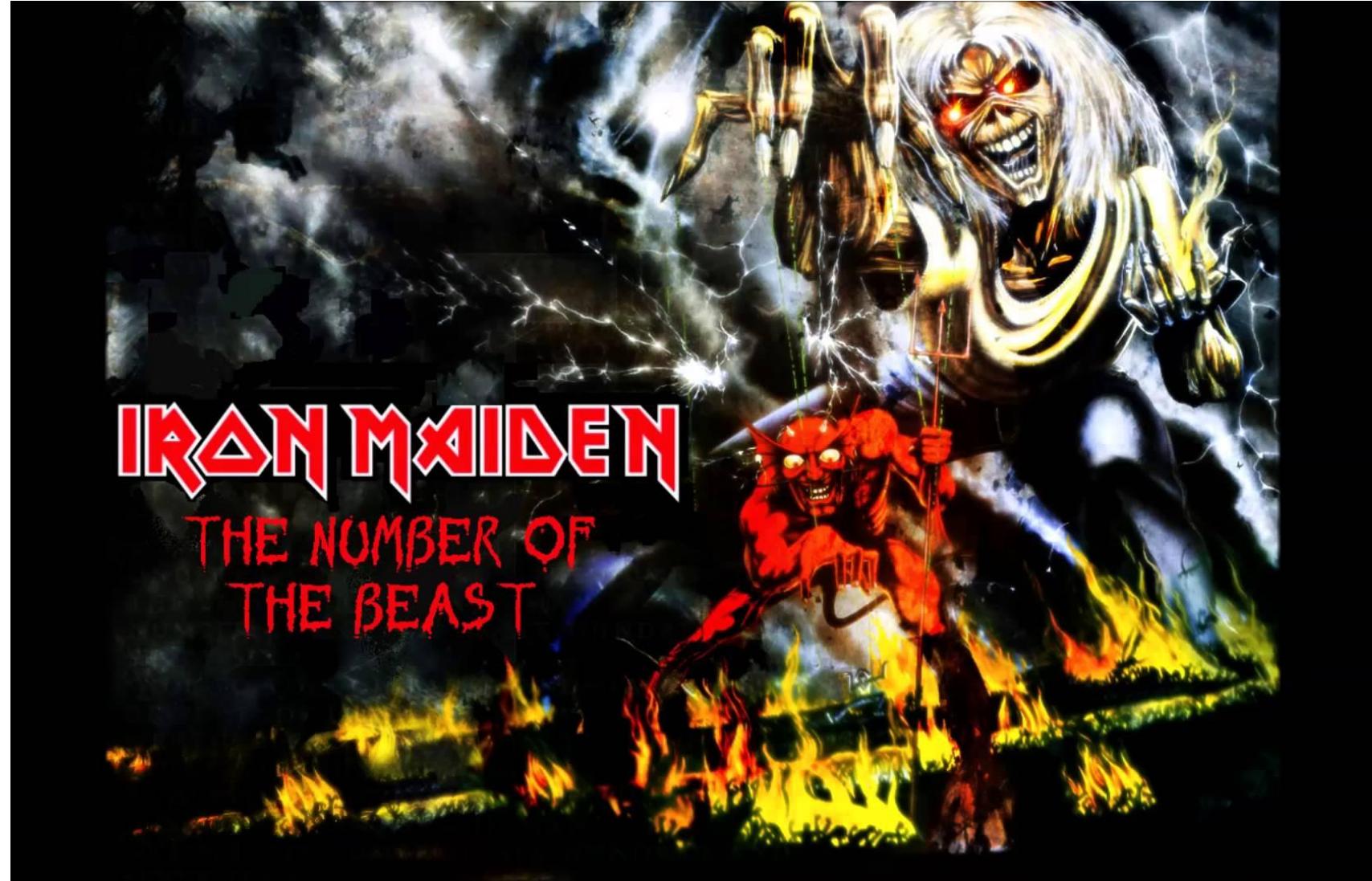
```
def back = 'back'  
def quotes = ["I'll be $back",  
"I'll be ${-> back}",  
"I'll be ${back}",  
"I'll be "+back]
```

**В рантайме отрабатывают только closures,
всё остальное инлайнится**

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

А это единственный
closure

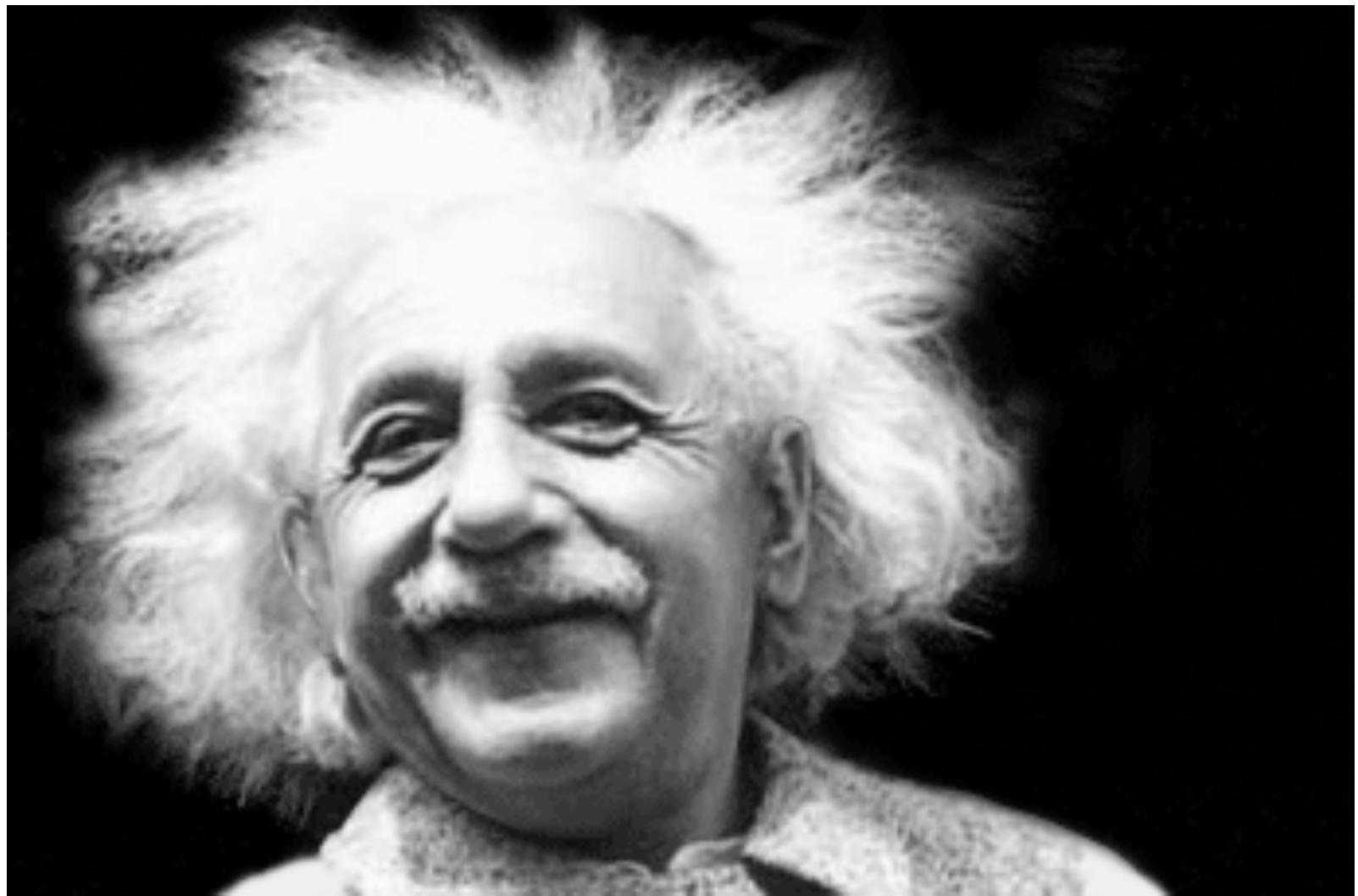




Как напечатать 666?

- A. **def** beast = '6' * Math.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



А я вам
говорил!





соскучились?

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

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

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

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

```
@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 TreaayeMap true
- D. class HashMap false







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

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

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

TreaayeMap a = [5]

List and map constructor

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 constructor

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 constructor

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 HashMa

}

```
/*
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<Entry<?,?> >
```

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 m1 and m2 represent the same mappings if
m1.entrySet().equals(m2.entrySet()). This ensures that the equals method works properly across different
implementations of the Map interface.



```
class MacLeod {  
  
    THERECANBEONLYONE getTHERECANBEONLYONE() {  
        Class clazz = THERECANBEONLYONE  
        return clazz.newInstance()  
    }  
  
}  
  
class THERECANBEONLYONE {}  
  
println new MacLeod().THERECANBEONLYONE
```

```
class MacLeod {  
  
    THERECANBEONLYONE getTHERECANBEONLYONE() {  
        Class clazz = THERECANBEONLYONE  
        return clazz.newInstance()  
    }  
  
}  
  
class THERECANBEONLYONE {}  
  
println new MacLeod().THERECANBEONLYONE
```

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



- A. MultipleCompilationErrorsException
- B. StackOverflowError
- C. NullPointerException
- D. Yet Another Exception



```
class MacLeod {  
  
    THERECANBEONLYONE getTHERECANBEONLYONE() {  
        Class clazz = THERECANBEONLYONE  
        return clazz.newInstance()  
    }  
  
}
```

```
class MacLeod {  
  
    THERECANBEONLYONE getTHERECANBEONLYONE() {  
        Class clazz = getTHERECANBEONLYONE()  
        return clazz.newInstance()  
    }  
  
}
```

Как мы это чиним?

Class<THERECANBEONLYONE> clazz = THERECANBEONLYONE.class

```
class MacLeod {  
    THERECANBEONLYONE getTHERECANBEONLYONE() {  
        Class<THERECANBEONLYONE> clazz =  
            THERECANBEONLYONE.class  
        return clazz.newInstance()  
    }  
}
```



```
class MacLeod {  
    THERECANBEONLYONE getTHERECANBEONLYONE() {  
        Class<THERECANBEONLYONE> clazz =  
            getTHERECANBEONLYONE().class  
        return clazz.newInstance()  
    }  
}
```

Как мы это чиним?

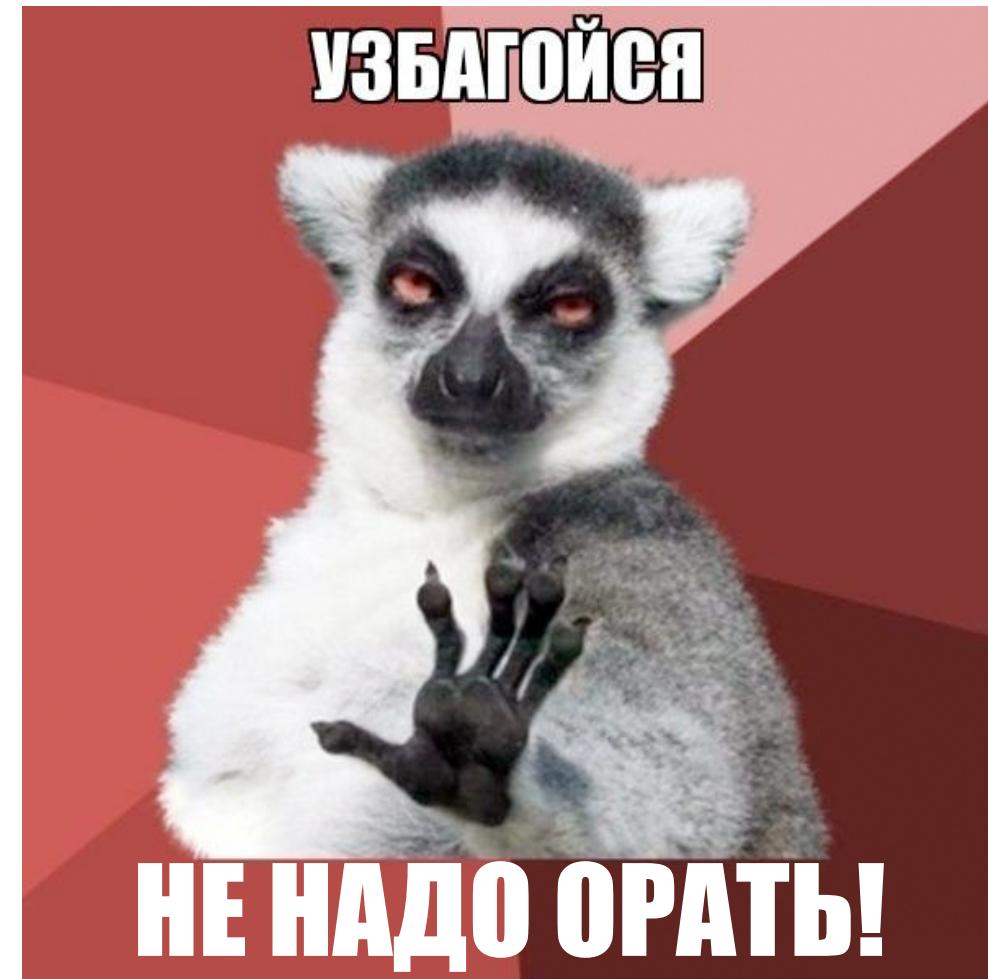
Class<THERECANBEONLYONE> clazz = (THERECANBEONLYONE as Class)

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



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

Как мы это чиним?



УЗБАГОЙСЯ

НЕ НАДО ОРАТЬ!

Schwarzenegger

True Lies



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
```

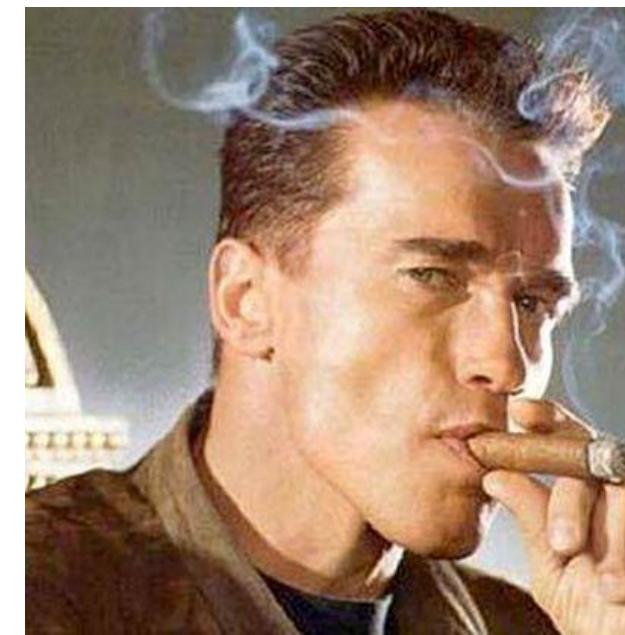
- A. false
- B. Groovy
- C. ClassCastException
- D. Startup error



DGDO | ~



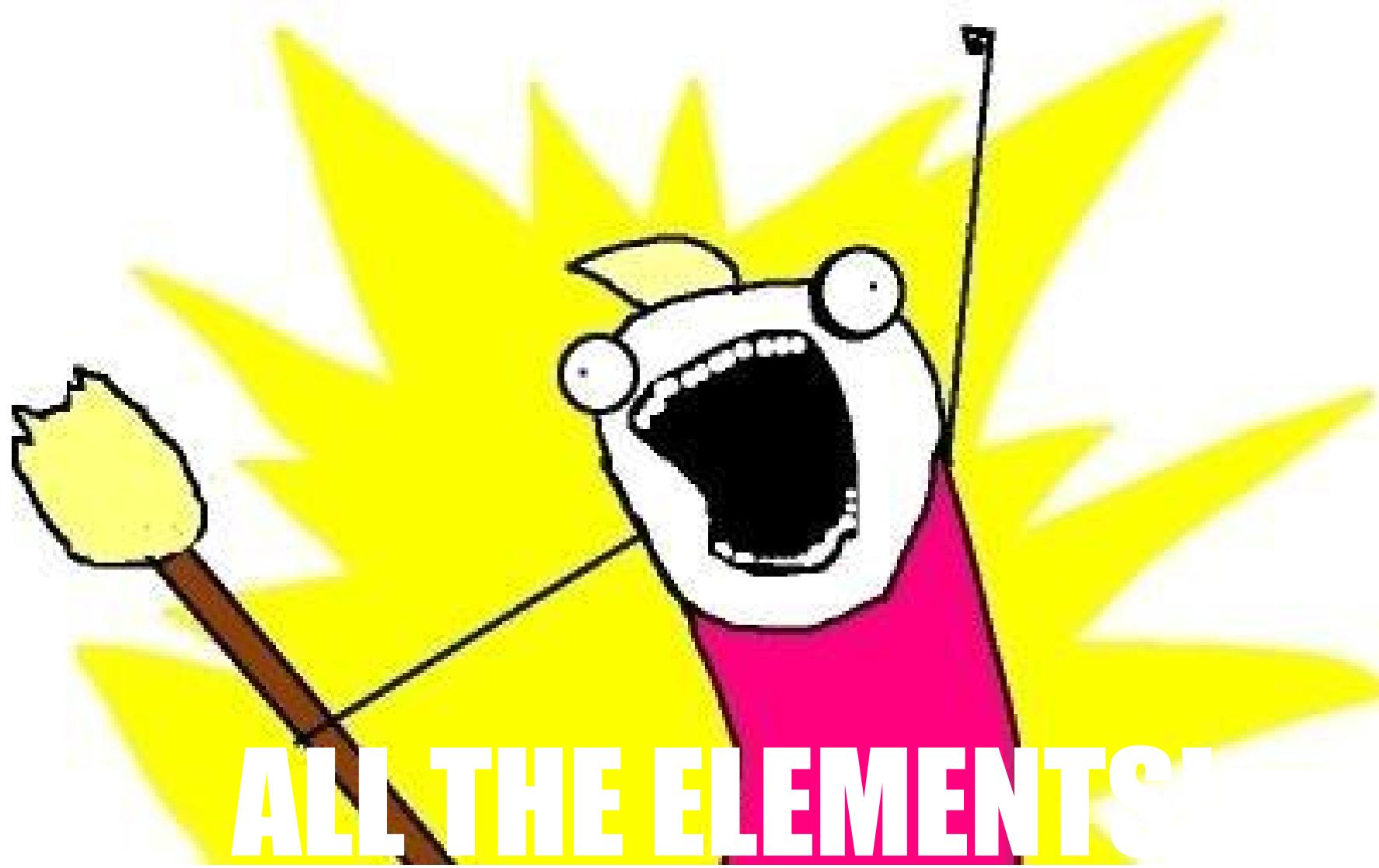
RTFM!



§ Strings

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

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

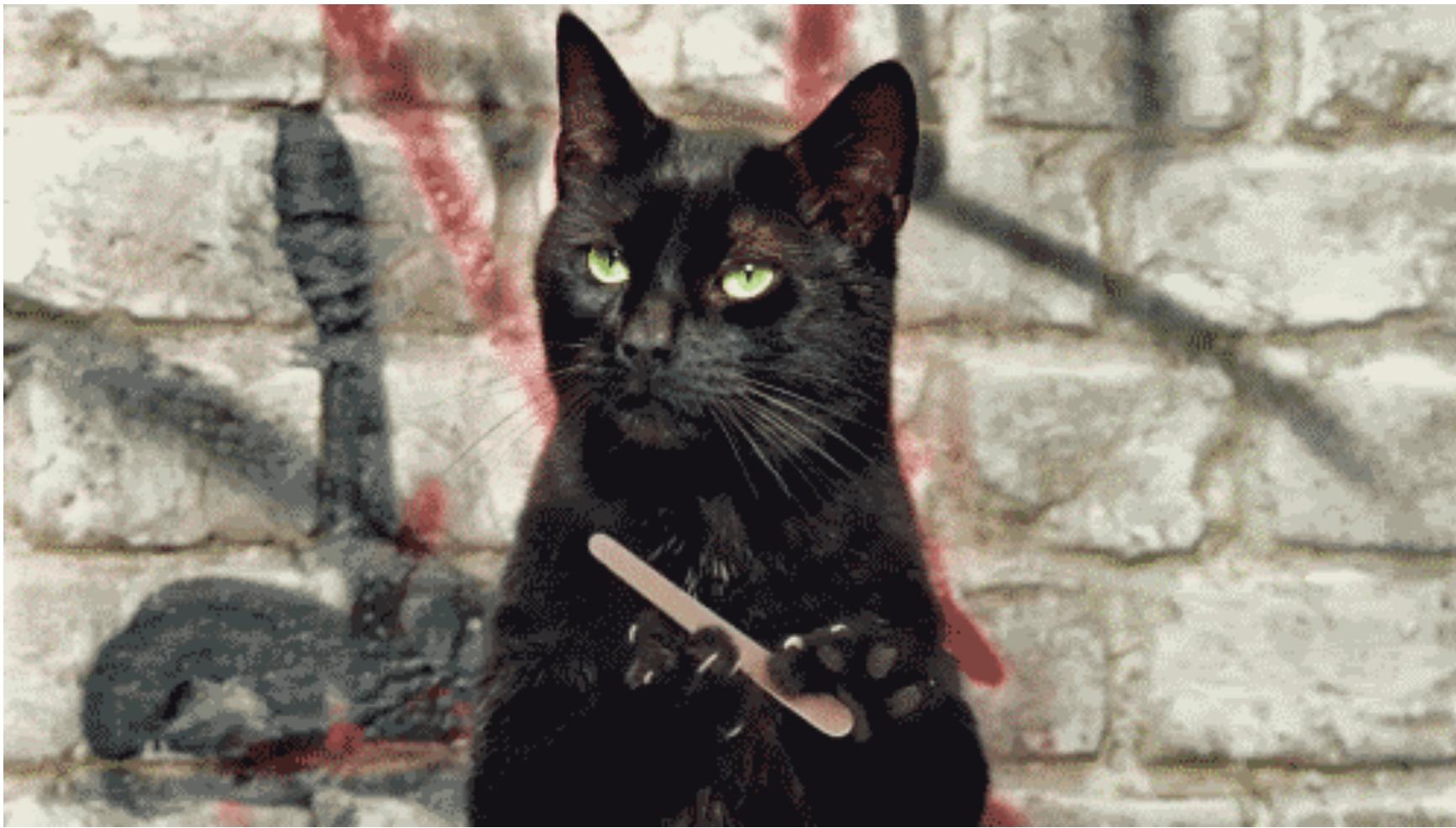


```
class Kitty { def fur }  
def kitties = [new Kitty(fur: 'soft'),new Kitty(fur: 'warm'),new Kitty(fur: 'purr') ]  
  
println kitties.collect { it.fur }  
println kitties*.fur  
println kitties.fur
```

Сколько будет одинаковых строк?

- A. Все разные
- B. Две одинаковые
- C. Все одинаковые
- D. Не побежит





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

```
def kitties
```

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

```
println kitties*.fur
```

```
println kitties.fur
```

Сколько будет одинаковых строк?

- A. Все 
- B. Две одинаковые
- C. Все одинаковые
- D. Не побежит



```
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 }
```

```
public static Collection asCollection(Object value) {
    if (value == 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`:



[]

null

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



```
ArrayList<String> expendables =  
['Arnold', 'Chuck', 'Silvester']  
  
def expendable = '?'  
  
for (String hero in expendables) {  
    if (hero == expendable) {  
        expendables.remove(hero)  
    }  
}  
println expendables
```

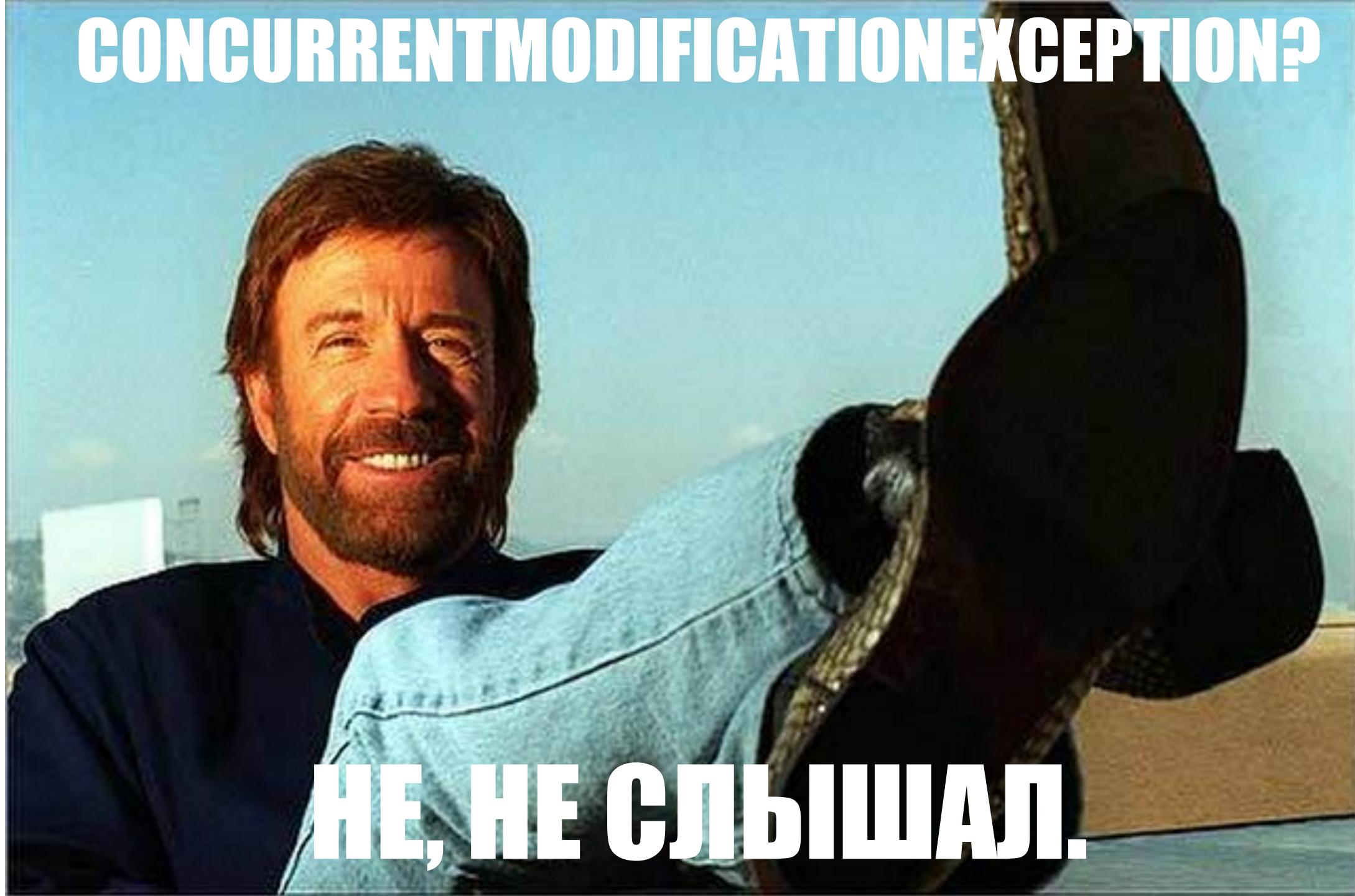
Кого поставить вместо '?' чтобы не было ConcurrentModificationException?

```
ArrayList<String> expendables =  
['Arnold', 'Chuck', 'Silvester']  
  
def expendable = '?'  
  
for (String hero in expendables) {  
    if (hero == expendable) {  
        expendables.remove(hero)  
    }  
}  
println expendables
```

- A. Никого
- B. Arnold
- C. Chuck
- D. Silverster



CONCURRENTMODIFICATIONEXCEPTION?



НЕ, НЕ СЛЫШАЛ.

А вас предупреждали

```
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.

foreach He фурычыт!

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

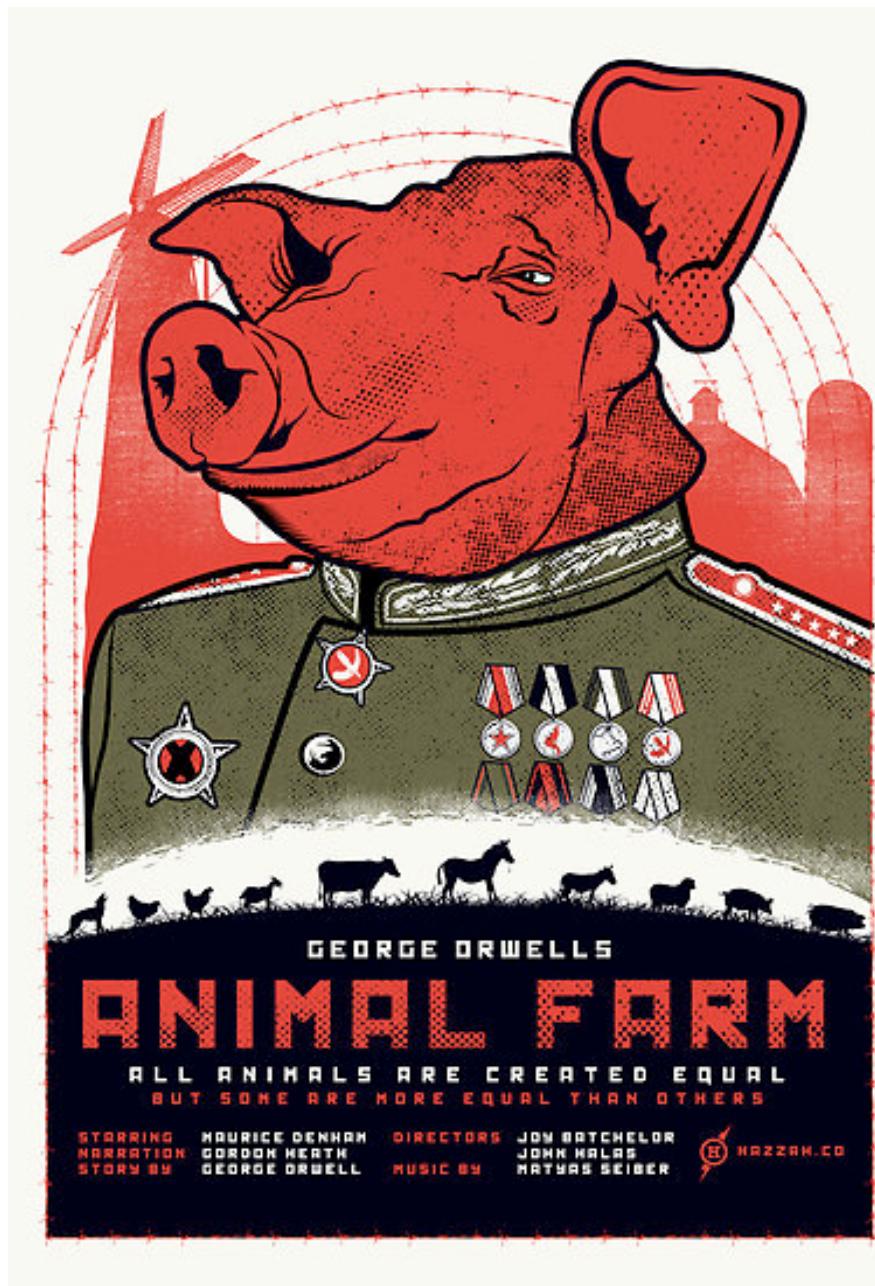
def iterator = expendable.iterator()
while (iterator.hasNext()) {
    Object element = iterator.next();
    if (element == expendable) {
        expendables.remove(element)
    }
}
```

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

public E next() {
 checkForComodification();

final void checkForComodification() {
 if (modCount != expectedModCount)
 throw new ConcurrentModificationException();

```
graph TD; A[for (String name in expendables)] --> B[def iterator = expendable.iterator()]; B --> C[Object element = iterator.next()]; C --> D[if (element == expendable)]; D --> E[public boolean hasNext()]; D --> F[public E next()]; F --> G[final void checkForComodification()];
```



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

```
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;  
}
```



```
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, null, 5]]
The Groovy logo is a blue star with white outlines and a white center. The word "Groovy" is written in a white, stylized font across the center of the star.
- 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]
```

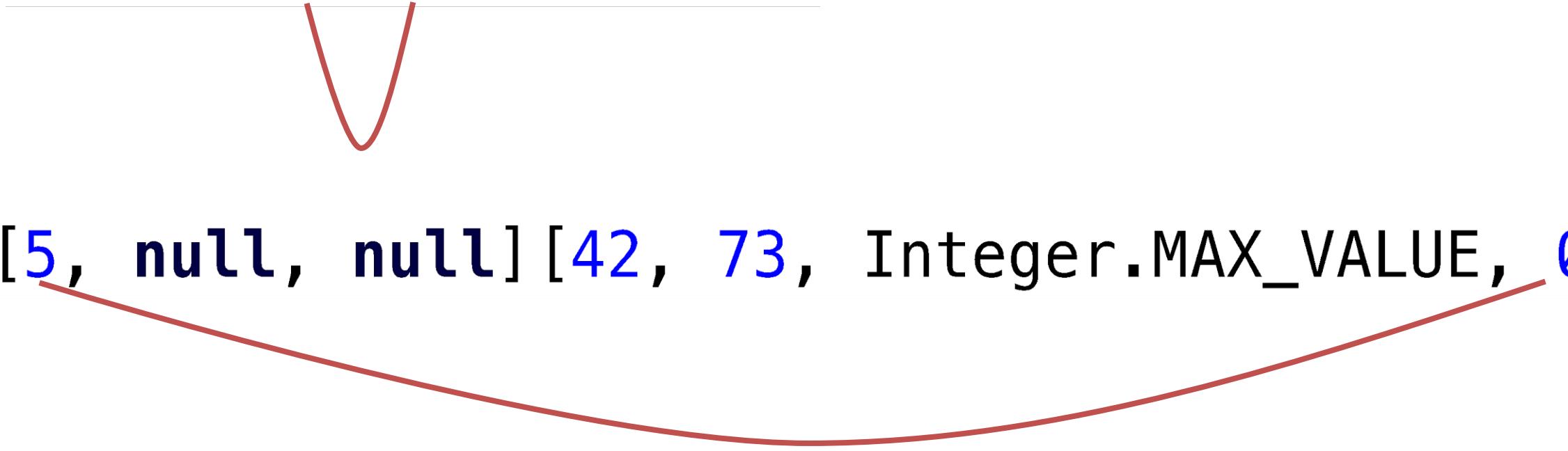
```
[5, null, null][42, 73, Integer.MAX_VALUE, 0]
```

```
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]]



```
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]
```

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

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

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

```
def jailhouseRock  
def loveMeTender  
def rockAroundTheClock = [1,2,3]  
  
false?null:true?[null]:true?[1,2,3]:[]
```

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

```
false?null:true?[null]:[1,2,3]
```

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

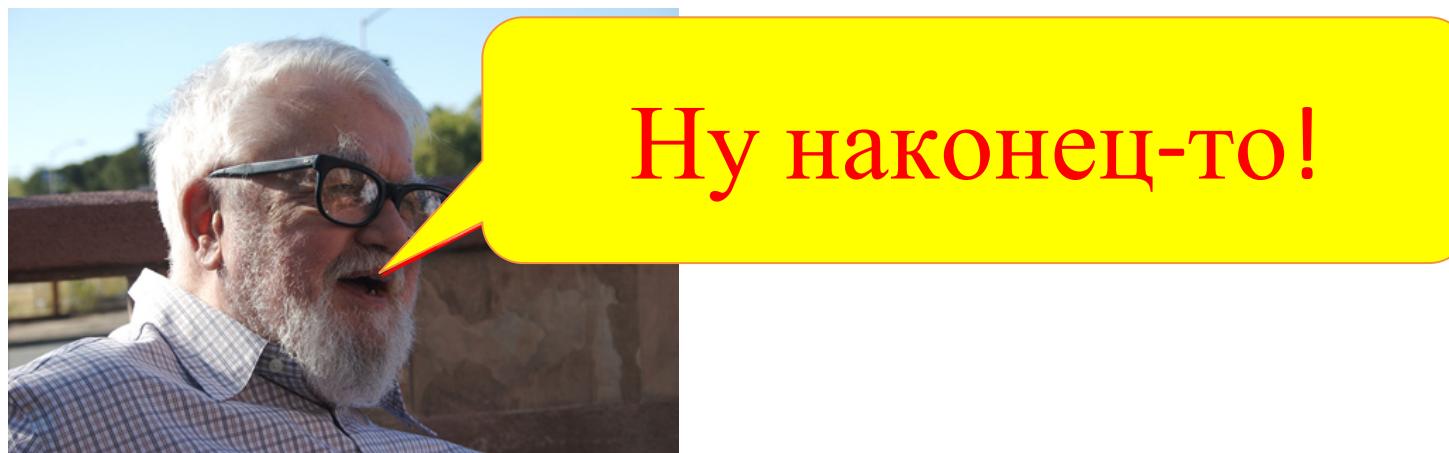
```
false?null:[null]
```

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

[null]

Как мы это чиним?

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



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.  Stargroovy or
- B. int, long, boolean
- C. java.lang.Integer,
java.lang.Long,
java.lang.Boolean
- D. int

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

The screenshot shows a Java code editor with several syntax errors. The first error is at the start of the second block: 'def' is used instead of 'if'. The second error is at the start of the third block: 'boolean' is misspelled as 'boolean'. The code editor highlights these errors with red squiggly lines under the respective words.

точно непорядок

def x = int
//println x
//
//if ((x = long)) {
// println x
//}
//
if (x = boolean) {
 '}' expected
 Unexpected symbol

A tooltip from the Java code editor highlights a syntax error. It points to the line 'if (x = boolean) {' and displays two messages: "'}' expected" and "Unexpected symbol". The word 'boolean' is also underlined with a red squiggle.

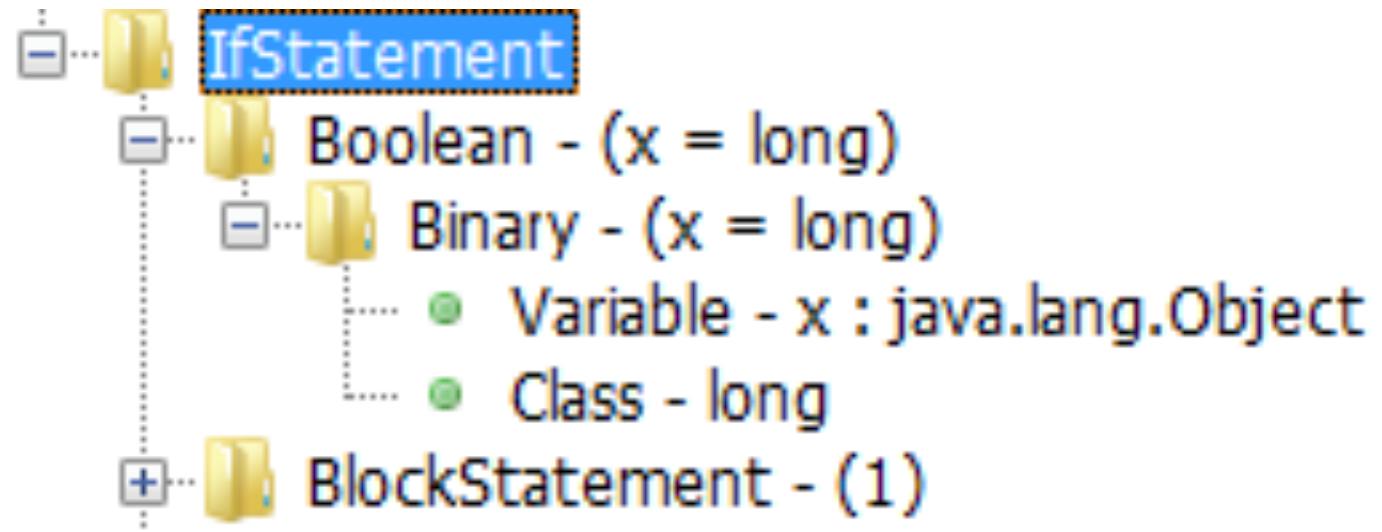
А ЭТО?

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

```



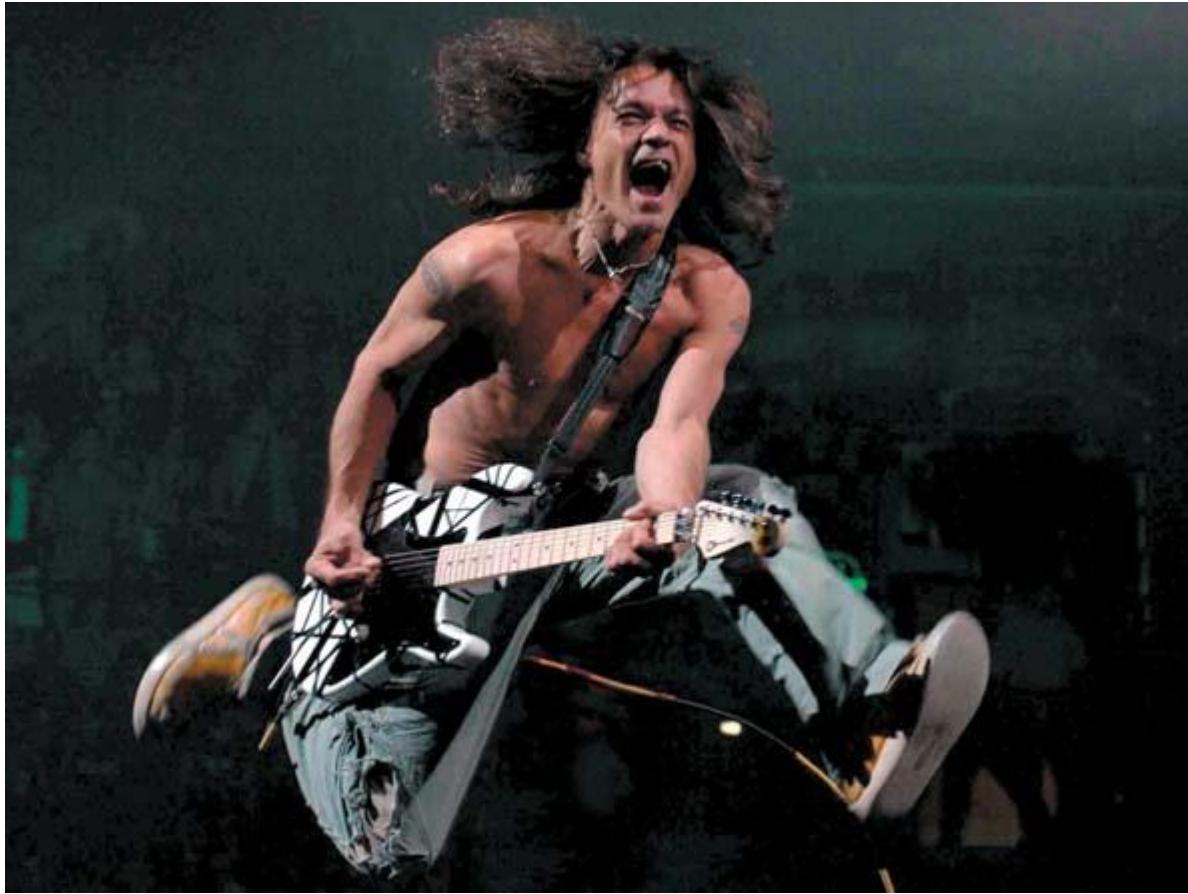
(Groovy) truth может ранить, так что просто добавь скобок



Пошла, пошла!



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

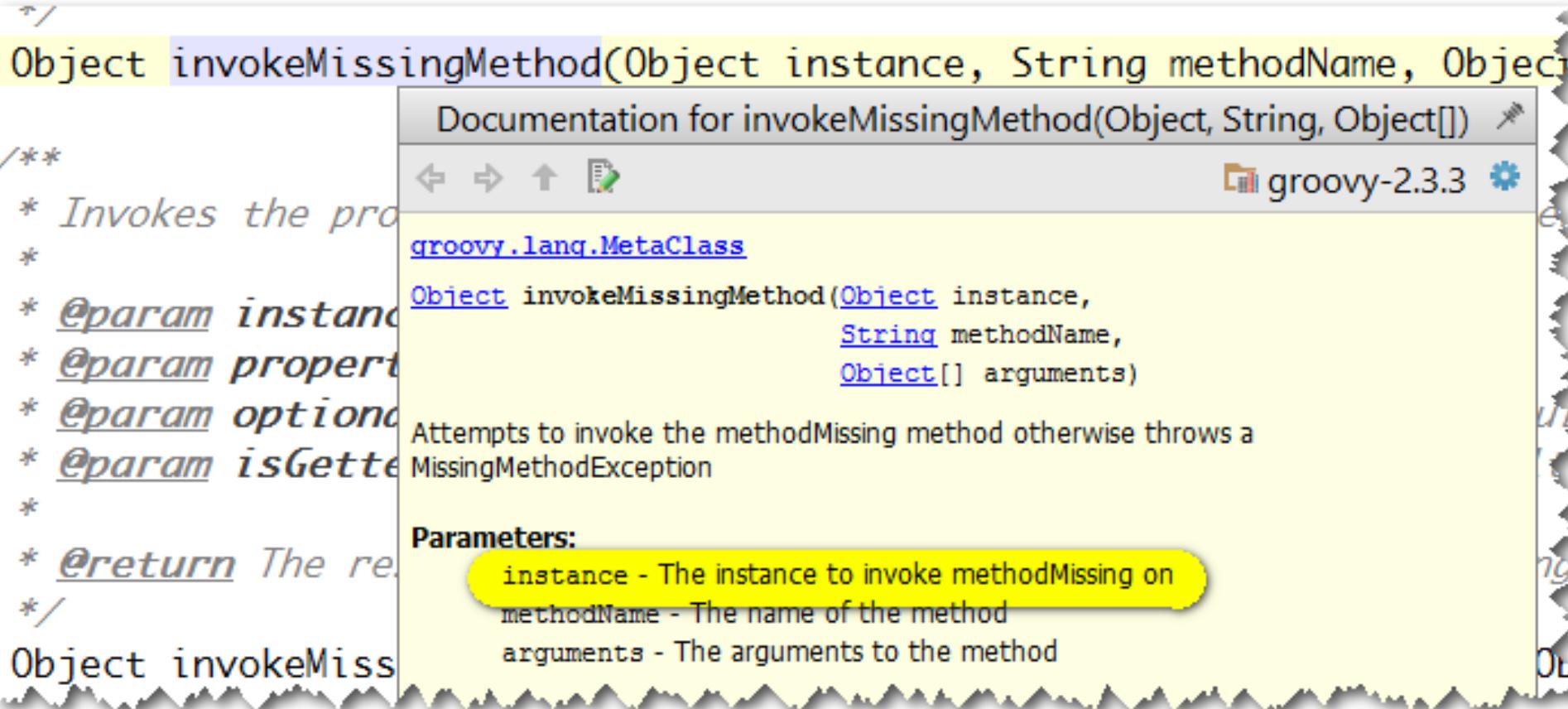
```
class VanHalen {  
  
    public static jump() {  
        "Here are the ${lyrics()}"  
    }  
  
    def methodMissing(String name, def args) {  
        'lyrics'  
    }  
  
    println VanHalen.jump()
```

- A. Here are the lyrics
- B. Here are the null
- C. Startup failure
- D. MissingMethodException





I can't believe you fell for the oldest trick in the book



The screenshot shows a Java IDE displaying Groovy code. The code is annotated with Javadoc-style comments. A tooltip is open over the parameter `instance`, highlighting its documentation.

```
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

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

```
'  
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>
```

Как мы это чиним? 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'S THE FREQUENCY,
KID?

R.E.M.

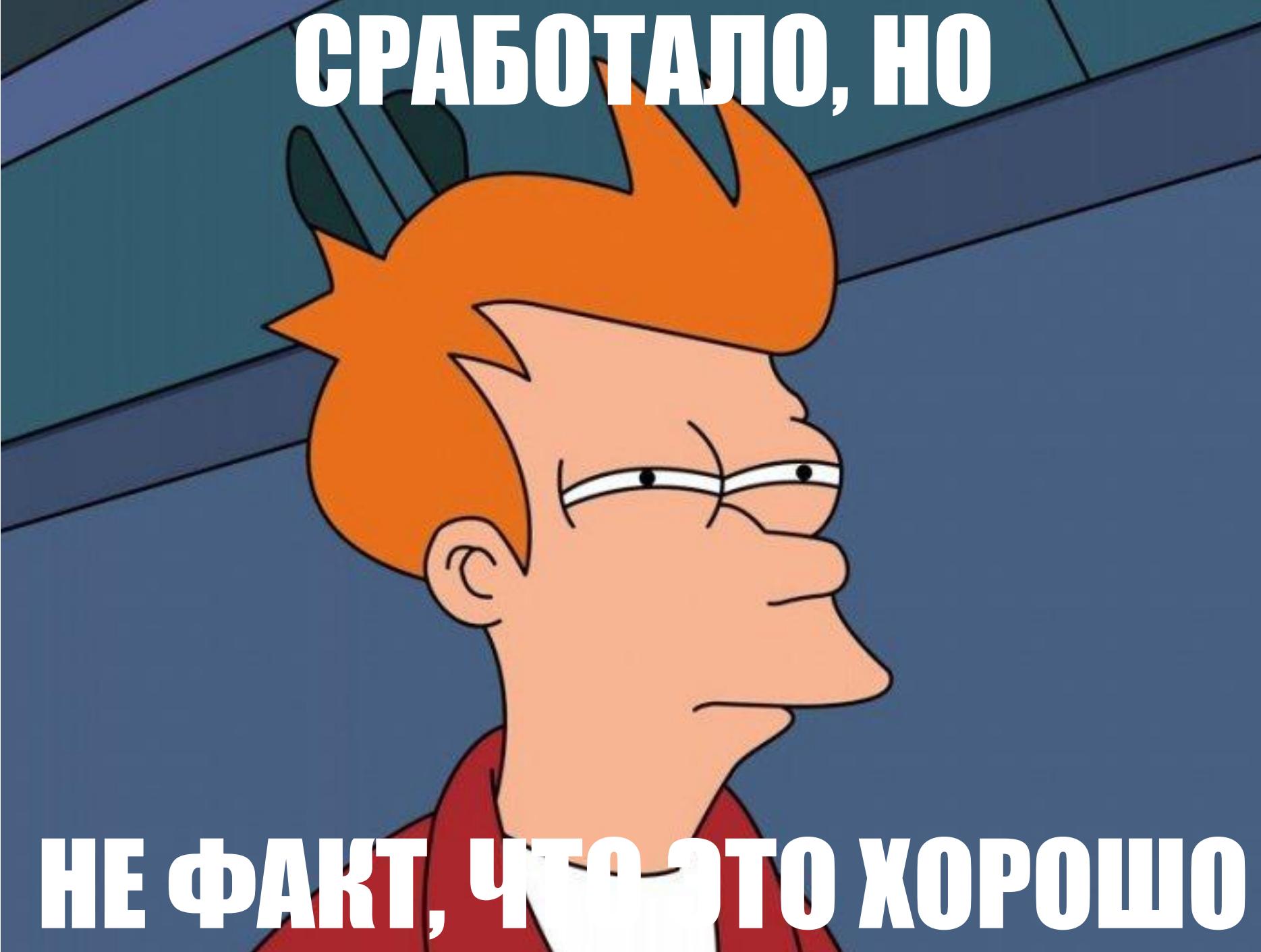
Compact Disc Maxi-Single

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@YYYYYYYY [groovy.lang.MetaClassImpl@YYYYYYYY[class java.util.LinkedHashMap]], Барух?
- C. What's the java.util.LinkedHashMap@XXXXXX, Барух?
- D. What's the frequency,  Барух?

СРАБОТАЛО, НО



НЕ ФАКТ, ЧТО ЭТО ХОРОШО

ПОМНИШЬ, ДЖЕЙМС ГОСЛИНГ ПРЕДУПРЕЖДАЛ

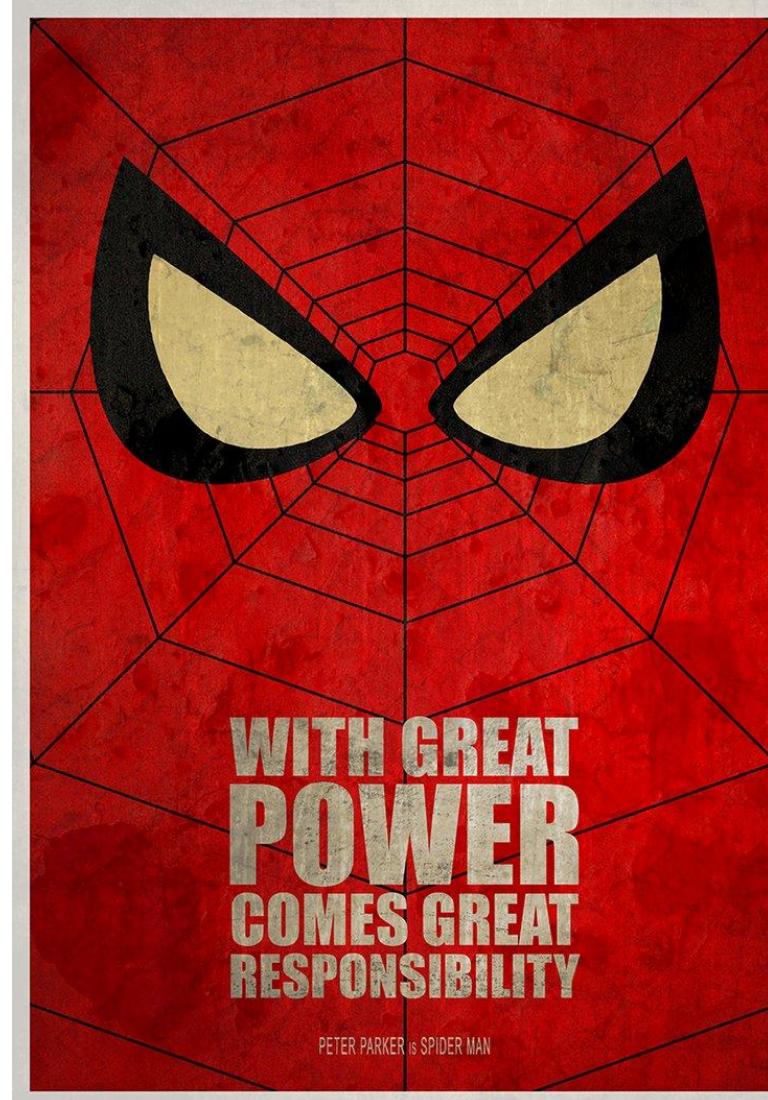


ЧТО OPERATOR OVERLOADING ЭТО ПЛОХОЭ

operator overloading!

Если `мар.metaClass` переопределён
для `мар.get('metaClass')`, он не
может быть переопределён и для
`map.getMetaClass()`

Выводы



1. Пишите читабельны
2. Комментируйте все
3. Иногда это баг
4. Пользуйте static code
intelliJ IDEA!
5. Rtfm
6. Скобки! Всегда добавляйте скобки!



Мы продолжаем развлекаться!
(смотрите какие уже футболочки
няшные!)

Наткнулись на паззлер? Давайте его
сюда!

- puzzlers jfrog.com
- Groovypuzzlers
@

Вот, мы рассылаем футболки, честно!

 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

[Follow](#)

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

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



Понравилось?

Хвалите нас в твиттере!

#groovypuzzlers

- @Groovypuzzlers

- @ekaborisov

- @baruch

Не понравилось?

/dev/null

**"Расставашки -
всегда пичалька".**

(с) Сократик

