



JAVA PUZZLERS NG S02

ВСЁ ЧУДЕСАТЕЕ И ЧУДЕСАТЕЕ



CLICK AND HACK
THE TYPE-IT BROTHERS



50,423 REPUTATION

7

90

166

Tagir Valeev top 0.30% this year

Author of [StreamEx](#) library.

IntelliJ IDEA Developer.

[FindBugs](#) project contributor.

Tech blog in Russian on [Habrahabr](#).

1,262
answers

12
questions

~3.1m
people reached

📍 Novosibirsk, Russia

🔄 amaembo

🕒 Member for 1 year, 11 months

👁 4,037 profile views

🕒 Last seen Apr 2 at 7:13

Top Tags (578)

java



SCORE 4,424 POSTS 1,165 POSTS % 91

java-8



SCORE 2,743 POSTS 558

java-stream



SCORE 1,827 POSTS 424

lambda



SCORE 595
POSTS 142

generics



SCORE 250
POSTS 70

collections



SCORE 208
POSTS 52

@jbaruch

@tagir_valeev

#jpoint

[View all tags →](#)



Baruch Sadogursky, JFrog Developer Advocate, @jbaruch



Cédric Champeau

@CedricChampeau FOLLOWS YOU

Software Engineer @Gradle, Inc.
Conference speaker. Introvert
(carlkingdom.com/10-myths-about...).
Wrote the static compiler for
#groovylang.

📍 St Hilaire de Loulay, France

🔗 melix.github.io/blog

📅 Joined January 2010

🎂 Born on April 04



Cédric Champeau, Software Engineer, Gradle Inc

Cédric Champeau is principal engineer at Gradle, Inc. and works on improving the open source project of the same name. He is also a core committer of Apache Groovy and spent more than 3 years developing full time the language at Pivotal, where he implemented features like the static compiler, traits, Android support, AST transformations or type checking extensions. Prior to that, he spent several years in the industry as a user, where he used Groovy in multiple industrial contexts including DSLs for natural language processing, scripting or even workflows.

🚀 JavaOne Rockstar

JAVAONE

▼ Juggling Multiple Java Platforms and *Jigsaw* with Gradle [CON4832]

Cédric Champeau, Software Engineer, Gradle Inc

The introduction of modules in Java 9 is probably one of the most disruptive additions to the platform since its inception. Until now all additions (generics, invokedynamic, lambdas, ...) have been pretty painless. Jigsaw is more likely to cause you a lot more trouble. This session describes the challenges Jigsaw poses and how with Gradle you can leverage the software model to migrate your applications smoothly, without having to migrate now. Learn how you can describe your APIs, build for multiple platforms, and benefit from compile avoidance right now. All you will have to do is switch the “Java 9 flag” when you’re ready.

📄 [Download Presentation](#)

@jbaruch

@tagir_valeev

#jpoint



 [kuksenk0](#)

Сергей Куксенко

Oracle

Java Performance Engineer. Работает с Java начиная с версии 1.0. За это время успел поучаствовать в разработке мобильных, клиентских, серверных приложений и виртуальных машин.

Производительностью Java занимается с 2005 года: сначала работал в Intel над Apache Harmony, а в данный момент в Oracle занимается производительностью OracleJDK/OpenJDK (его 3-я JVM).

@jbaruch

@tagir_valeev

#jpoint

1. ДВА КЛЕВЫХ ПАРНЯ НА СЦЕНЕ
2. СМЕШНЫЕ ГОЛОВОЛОМКИ
3. ВЫ ГОЛОСУЕТЕ ЗА ОТВЕТЫ!
4. МЫ КИДАЕМСЯ ФУТБОЛКАМИ ОТ JEROG
(СПАСИБО ИМ!)
5. ОФИЦИАЛЬНЫЕ ХЭШТЭГИ:
#JAVAPUZZLERSNG #JPOINT

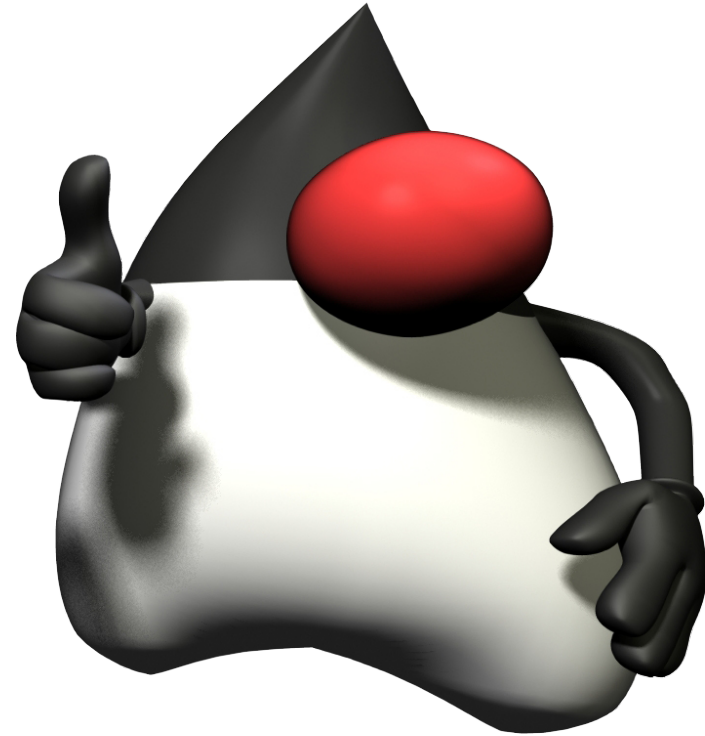
ПЕРВОЕ ПРАВИЛО ПАЗЗЛЕРОВ:



НЕ ЧИТИТЬ!

Проверка системы голосования! Какая у вас Java?

- A. Java 7
- B. Java 8
- C. Java 9
- D. Java 6
- E. Java 5
- F. Java 2



@jbaruch

@tagir_valeev

#jpoint

Готовьтесь, сейчас будет вот так:

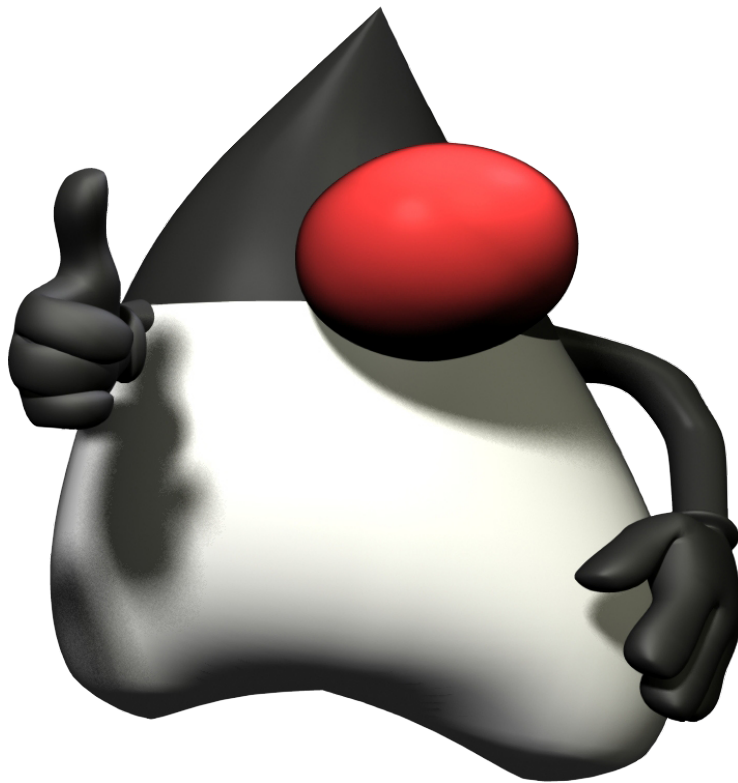


@jbaruch

@tagir_valeev

#jpoint

Всё работает (ну, или не работает) на последней версии Java 8 и 9 соответственно.



@jbaruch

@tagir_valeev

#jpoint

ТРЕТЬИМ БУДЕШЬ?



АМЕРИКАНСКИЕ ХАКЕРЫ



ХАКАЮТ БАНК


```
public class PerfectRobbery {  
    private Semaphore bankAccount = new Semaphore(-42);  
    public static void main(String[] args) {  
        PerfectRobbery perfectRobbery = new PerfectRobbery();  
        perfectRobbery.takeAllMoney();  
        perfectRobbery.checkBalance();  
    }  
    public void takeAllMoney(){  
        bankAccount.drainPermits();  
    }  
    public void checkBalance(){  
        System.out.println(bankAccount.availablePermits());  
    }  
}
```

- A. IllegalArgumentException – нельзя создавать семафор с негативным балансом
- B. UnsupportedOperationException - нельзя сливать при негативном балансе
- C. 0
- D. -42

```
public class PerfectRobbery {  
    private Semaphore bankAccount = new Semaphore(-42);  
    public static void main(String[] args) {  
        PerfectRobbery perfectRobbery = new PerfectRobbery();  
        perfectRobbery.takeAllMoney();  
        perfectRobbery.checkBalance();  
    }  
    public void takeAllMoney(){  
        bankAccount.drainPermits();  
    }  
    public void checkBalance(){  
        System.out.println(bankAccount.availablePermits());  
    }  
}
```

- A. IllegalArgumentException – нельзя создавать семафор с негативным балансом
- B. UnsupportedOperationException - нельзя сливать при негативном балансе



0

D. -42



@jbaruch

@tagir_valeev

#jpoint


```

public class PerfectRobbery {
    private Semaphore bankAccount = new Semaphore(-42);
    public static void main(String[] args) {
        PerfectRobbery perfectRobbery = new PerfectRobbery();
        perfectRobbery.takeAllMoney();
        perfectRobbery.checkBalance();
    }
    public void takeAllMoney(){
        bankAccount.drainPermits();
    }
    public void checkBalance(){
        System.out.println(bankAccount.availablePermits());
    }
}

```

- A. IllegalArgumentException – нельзя создавать семафор с негативным балансом
- B. UnsupportedOperationException – нельзя сливать при негативном балансе



0

D. -42

Semaphore

```
public Semaphore(int permits)
```

Creates a Semaphore with the given number of permits and nonfair fairness setting.

Parameters:

`permits` - the initial number of permits available. This value may be negative, in which case releases must occur before any acquires will be granted.

Доступно -42
пермита?!

drainPermits

```
public int drainPermits()
```

Acquires and returns all permits that are immediately available.

Returns:

the number of permits acquired

@jbaruch

@tagir_valeev

#jpoint

JDK-8169272 : Clarify Semaphore.drainPermits behavior when current permits are negative

Details

Type: Bug

Submit Date: 2016-11-04

Status: Resolved

Updated Date: 2017-02-16

Project Name: JDK

Resolved Date: 2016-11-29

Component: core-libs

OS:

Sub-Component: java.util.concurrent

CPU:

Priority: P4

Resolution: Fixed

Affected Versions:

Fixed Versions: 9 (b148) 

Related Reports

I agree that the current (intentional) behavior should be clarified in documentation. Just adding the following seems to suffice:

```
*** 603,608 ****  
--- 603,609 ----
```

```
    /**  
    * Acquires and returns all permits that are immediately available.  
+   * Upon return, zero permits are available.  
    *  
    * @return the number of permits acquired  
    */
```





@jbaruch

@tagir_valeev

#jpoint


```
Collections.emptyList()      == Collections.emptyList();  
Collections.emptyIterator() == Collections.emptyIterator();
```

- A. true/true
- B. true/false
- C. false/true
- D. false/false

```
Collections.emptyList() == Collections.emptyList();  
Collections.emptyIterator() == Collections.emptyIterator();
```

 true/true

B. true/false

C. false/true

D. false/false



@jbaruch

@tagir_valeev

#jpoint

```
Spliterators.emptySpliterator() == Spliterators.emptySpliterator();  
Stream.empty() == Stream.empty();
```

A. true/true

 true/false

C. false/true

D. false/false

Singleton наносит ответный удар!



@jbaruch

@tagir_valeev

#jpoint

```
Spliterators.emptySpliterator() == Spliterators.emptySpliterator();  
Stream.empty() == Stream.empty();
```

A. true/true

 true/false

C. false/true

D. false/false

Даже у пустого стрима есть состояние!

onClose

`S onClose(Runnable closeHandler)`

Returns an equivalent stream with an additional close handler. Close handler even if earlier close handlers throw exceptions. If any close handler throws as suppressed exceptions (unless one of the remaining exceptions is the same).

This is an intermediate operation.

Parameters:

`closeHandler` - A task to execute when the stream is closed.

Returns:

a stream with a handler that is run if the stream is closed.

Interface Stream<T>

Type Parameters:

`T` - the type of the stream elements

All Superinterfaces:

`AutoCloseable`, `BaseStream<T, Stream<T>>`

```
public interface Stream<T>
    extends BaseStream<T, Stream<T>>
```

A stream should be operated on (invoking an intermediate or terminal stream operation) **only once**. This rule traversals of the same stream. A stream implementation may throw `IllegalStateException` if it detects that a new stream object, it may not be possible to detect reuse in all cases.

@jbaruch

@tagir_valeev

#jpoint

ШВАРЦЕНЕГГЕР ДЕВИТО

БЛИЗНЕЦЫ

по мотивам романа РИЧАРДА МАГНА



”одинаковые”

1. Такие же по состоянию внутренней структуры.
2. Не про контракт “equals и hashCode”.
3. Не имеет отношение к проверке референсов.

```
List[] twins = new List[2];  
Arrays.setAll(twins, ArrayList::new);
```

- A. Одинаковые пустые списки
- B. Одинаковые не пустые списки
- C. Не одинаковые пустые списки
- D. Не одинаковые не пустые списки

```
List[] twins = new List[2];  
Arrays.setAll(twins, ArrayList::new);
```

- A. Одинаковые пустые списки
- B. Одинаковые не пустые списки
-  C. Не одинаковые пустые списки
- D. Не одинаковые не пустые списки



@jbaruch

@tagir_valeev

#jpoint


```
List[] twins = new List[2];  
Arrays.setAll(twins, ArrayList::new);
```

- A. Одинаковые пустые списки
- B. Одинаковые не пустые списки
-  C. Не одинаковые пустые списки
- D. Не одинаковые не пустые списки

setAll

```
public static void setAll(double[] array,  
                          IntToDoubleFunction generator)
```

Set all elements of the specified array, using the provided generator function to compute each element.

If the generator function throws an exception, it is relayed to the caller and the array is left in an indeterminate state.

Parameters:

array - array to be initialize

generator - a function **accepti**

Throws:

`NullPointerException` - if the

Since:

1.8

ArrayList

```
public ArrayList(int initialCapacity)
```

Constructs an empty list with the specified initial capacity.

Parameters:

`initialCapacity` - the initial capacity of the list

Throws:

`IllegalArgumentException` - if the specified initial capacity is negative

Дядя Сэм, он же Single Abstract Method

@jbaruch

@tagir_valeev



```
public interface Сэм<T> {  
    default void расширятьНато(String новаяСтрана) {  
        System.out.println(новаяСтрана); }  
    void расширятьНато(T songName);  
    void захватыватьНефть(T новаяСтрана);  
    void захватыватьНефть(String новаяСтрана);  
}  
@FunctionalInterface  
public interface ДядяСэм extends Single<String> { }
```

- A. Что за ерунда?! 'Single' означает один метод, не три!
- B. Проблема с методом расширятьНато(T), если его убрать, всё ОК
- C. Проблема с методами захватыватьНефть, если убрать один, то всё будет ОК.
- D. Всё путём! Дубликаты схлопываются, и мы остаёмся с одним захватыватьНефть

@jbaruch

@tagir_valeev

#jpoint

```
public interface Сэм<T> {  
    default void расширитьНато(String новаяСтрана) {  
        System.out.println(новаяСтрана); }  
    void расширитьНато(T songName);  
    void захватыватьНефть(T новаяСтрана);  
    void захватыватьНефть(String новаяСтрана);  
}  
@FunctionalInterface  
public interface ДядяСэм extends Single<String> { }
```

- A. Что за ерунда?! 'Single' означает один метод, не три!
-  Проблема с методом расширитьНато(T), если его убрать, всё ОК
- C. Проблема с методами захватыватьНефть, если убрать один, то всё будет ОК.
- D. Всё путём! Дубликаты схлопываются, и мы остаёмся с одним захватыватьНефть

@jbaruch

@tagir_valeev

#jpoint



```
public interface Сэм<T> {  
    default void расширятьНато(String новаяСтрана) {  
        System.out.println(новаяСтрана); }  
    void расширятьНато(T songName);  
    void захватыватьНефть(T новаяСтрана);  
    void захватыватьНефть(String новаяСтрана);  
}  
@FunctionalInterface  
public interface ДядяСэм extends Single<String> { }
```

- A. Что за ерунда?! 'Single' означает один метод, не три!
- 👉 Проблема с методом расширятьНато(T), если его убрать, всё ОК
- C. Проблема с методами захватыватьНефть, если убрать один, то всё будет ОК.
- D. Всё путём! Дубликаты схлопываются, и мы остаёмся с одним захватыватьНефть

@jbaruch

@tagir_valeev

#jpoint

9.4.1.3. Inheriting Methods with Override-Equivalent Signatures

It is possible for an interface to inherit several methods with override-equivalent signatures ([§8.4.2](#)).

If an interface `I` inherits a default method whose signature is override-equivalent with another method inherited by `I`, then a compile-time error occurs. (This is the case whether the other method is abstract or default.)

АМЕРИКАНСКИЕ ХАКЕРЫ



ВСЁ ЕЩЁ ХАКАЮТ БАНК

Как Прохакать Банк

- ☑ Банковский софт написан на Java
- ☑ Хакнуть!
- ☑ Найти самые крупные счета


```
Set<String> accounts =  
    new HashSet<>(Arrays.asList("Gates", "Buffett", "Bezos", "Zuckerberg"));  
System.out.println("accounts= " + accounts);
```

- A. Порядок объявления сохраняется
- B. Порядок неизвестен, но сохраняется между запусками
- C. Порядок неизвестен и меняется при каждом перезапуске JVM
- D. Порядок неизвестен и меняется при каждой распечатке

```
Set<String> accounts =  
    new HashSet<>(Arrays.asList("Gates", "Buffett", "Bezos", "Zuckerberg"));  
System.out.println("accounts= " + accounts);
```

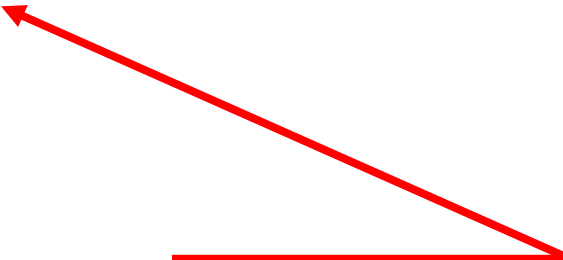
- A. Порядок объявления сохраняется
-  B. Порядок неизвестен, но сохраняется между запусками
- C. Порядок неизвестен и меняется при каждом перезапуске JVM
- D. Порядок неизвестен и меняется при каждой распечатке



```
Set<String> accounts =  
    new HashSet<>(Arrays.asList("Gates", "Buffett", "Bezos", "Zuckerberg"));  
System.out.println("accounts= " + accounts);
```

- A. Порядок объявления сохраняется
-  B. Порядок неизвестен, но сохраняется между запусками
- C. Порядок неизвестен и меняется при каждом перезапуске JVM
- D. Порядок неизвестен и меняется при каждой распечатке

```
public boolean add(E e) {  
    return map.put(e, PRESENT) == null;  
}
```



Все знают же, да?

Ваш ход, ФБР!



@jbaruch

@tagir_valeev

#jpoint

JDK 9[Downloads](#)
[Feedback Forum](#)
[OpenJDK](#)
[Planet JDK](#)**JDK™ 9 Early Access Releases****9 Build 163**

Summary of changes in JDK 9 build 163

[Feedback forum](#)
[Report Bugs](#)

The list of JDK 9 Features included in the JDK 9 Early Access Downloads can be found in JDK 9 Project page [here](#).

The download size is larger than JDK 8 downloads because the downloads include JMOD files to allow developers to try out creating custom run-time images. To find out about JMOD and creating custom run-time images see the [Quick-Start Guide](#) and [JEP 282](#).

Warning:

These early access release downloads of the JRE and JDK are based on code available on OpenJDK at the time they were built and might not include the latest security patches.

License Agreement:

You must accept the [Early Adopter Development License Agreement for Java SE](#) to download this software.

☐ Accept License Agreement | ☐ Decline License Agreement

Documentation

- [JDK Docs](#) (75.68 MB tar.gz | [HTML](#))
- [JavaFX Docs](#) (10.10 MB zip | [HTML](#))
- [Supported Platforms*](#)

Platforms		JRE	JDK
Windows	32-bit	exe (sha256) 83.35 MB	exe (sha256) 297.75 MB
	64-bit	exe (sha256) 88.33 MB	exe (sha256) 308.60 MB
Mac OS X	64-bit	dmg(sha256) 72.08 MB	dmg (sha256) 319.33 MB
	32-bit	tar.gz (sha256) 77.57 MB	tar.gz (sha256) 270.33 MB

```
Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg");  
System.out.println("accounts= " + accounts);
```

- A. Порядок объявления сохраняется
- B. Порядок неизвестен, но сохраняется между запусками
- C. Порядок неизвестен и меняется при каждом перезапуске JVM
- D. Порядок неизвестен и меняется при каждой распечатке

```
Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg");  
System.out.println("accounts= " + accounts);
```

- A. Порядок объявления сохраняется
- B. Порядок неизвестен, но сохраняется между запусками
-  C. Порядок неизвестен и меняется при каждом перезапуске JVM
- D. Порядок неизвестен и меняется при каждой распечатке

```
$ echo 'Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg")' | jshell | tail -n 4
jshell> Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg")
accounts ==> [Buffett, Bezos, Gates, Zuckerberg]
```

```
jshell> %
```

```
~
$ echo 'Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg")' | jshell | tail -n 4
jshell> Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg")
accounts ==> [Zuckerberg, Gates, Bezos, Buffett]
```

```
jshell> %
```

```
~
$ echo 'Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg")' | jshell | tail -n 4
jshell> Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg")
accounts ==> [Gates, Bezos, Buffett, Zuckerberg]
```

```
jshell> %
```

```
~
```

```
$ _
```




@jbaruch

@tagir_valeev

#jpoint

```
Set<String> accounts = Set.of("Gates", "Buffett", "Bezos", "Zuckerberg");  
System.out.println("accounts= " + accounts);
```

- A. Порядок объявления сохраняется
- B. Порядок неизвестен, но сохраняется между запусками
-  C. Порядок неизвестен и меняется при каждом перезапуске JVM
- D. Порядок неизвестен и меняется при каждой распечатке

```
private int probe(Object pe) {  
    int idx = Math.floorMod(pe.hashCode() ^ SALT, elements.length);  
    while (true) {  
        E ee = elements[idx];  
        if (ee == null) {  
            return -idx - 1;  
        } else if (pe.equals(ee)) {  
            return idx;  
        } else if (++idx == elements.length) {  
            idx = 0;  
        }  
    }  
}
```

Что уже может случиться?



Какое утверждение верно?

- A. Если сделать приложение jigsaw модулем, зависимости в classpath будут подгружаться корректно
- B. Если одна из зависимостей – jigsaw модуль, то обязательно прописать файл module-info
- C. Если вы прописали файл module-info, то все зависимости придётся прописать дважды, в classpath и в module-info
- D. Никакое не верно

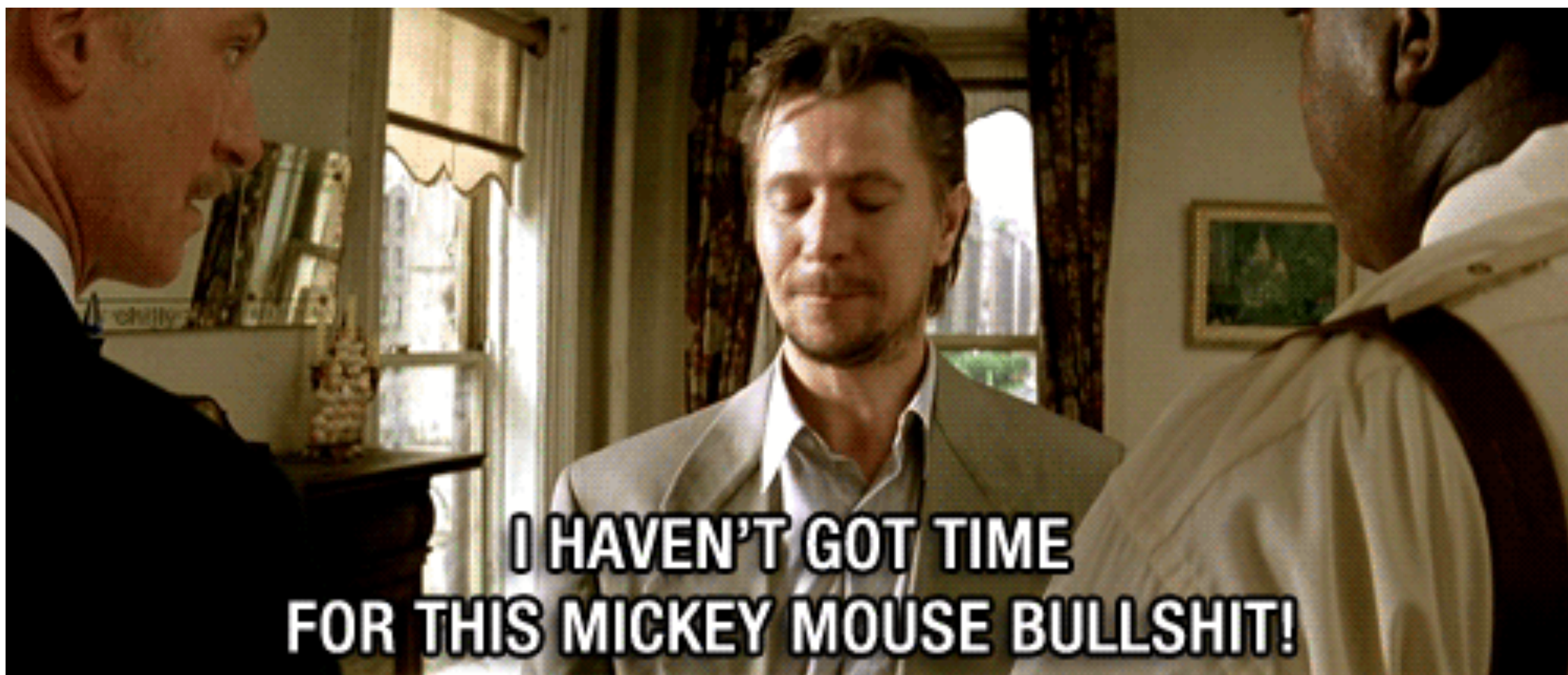
Какое утверждение верно?

- A. Если сделать приложение jigsaw модулем, зависимости в classpath будут подгружаться корректно
- B. Если одна из зависимостей – jigsaw модуль, то обязательно прописать файл module-info
-  C. Если вы прописали файл module-info, то все зависимости придётся прописать дважды, в classpath и в module-info
- D. Никакое не верно



На самом деле есть нюанс!

Есть еще `--module-path`, но это на третий сезон.



@jbaruch

@tagir_valeev

#jpoint



НЕУДЕРЖИМЫЕ 2

ОНИ ВЕРНУЛИСЬ, ЧТОБЫ ЗАЖЕЧЬ


```
static void killThemAll(Collection<Hero> expendables) {  
    Iterator<Hero> heroes = expendables.iterator();  
    heroes.forEachRemaining(e -> {  
        if (heroes.hasNext()) {  
            heroes.next();  
            heroes.remove();  
        }  
    });  
    System.out.println(expendables);  
}
```

- A. Все умерли
- B. Только чётные умерли
- C. Все выжили
- D. Только нечётные умерли
- E. Все ответы верны

```
static void killThemAll(Collection<Hero> expendables) {  
    Iterator<Hero> heroes = expendables.iterator();  
    heroes.forEachRemaining(e -> {  
        if (heroes.hasNext()) {  
            heroes.next();  
            heroes.remove();  
        }  
    });  
    System.out.println(expendables);  
}
```

A. Все умерли

B. Только чётные умерли

C. Все выжили

D. Только нечётные умерли



Все ответы верны



@jbaruch

@tagir_valeev

#jpoint

```
static void killThemAll(Collection<Hero> expendables) {  
    Iterator<Hero> heroes = expendables.iterator();  
    heroes.forEachRemaining(e -> {  
        if (heroes.hasNext()) {  
            heroes.next();  
            heroes.remove();  
        }  
    });  
    System.out.println(expendables);  
}
```

A. Все умерли

B. Только чётные умерли

C. Все выжили

D. Только нечётные умерли



Все ответы верны

Никогда. Никогда так не делайте!!!

```
killThemAll(new ArrayList<String>(Arrays.asList("N", "S", "W", "S", "L", "S", "L", "V")));  
[]
```

```
killThemAll(new LinkedList<String>(Arrays.asList("N", "S", "W", "S", "L", "S", "L", "V")));  
[S,S,S,V]
```

```
killThemAll(new ArrayDeque<String>(Arrays.asList("N", "S", "W", "S", "L", "S", "L", "V")));  
[N,S,W,S,L,S,L,V]
```

```
killThemAll(new TreeSet<String>(Arrays.asList("N", "S", "W", "S", "L", "S", "L", "V")));  
[N,W,L,L]
```


Это Дагу Ли оно unspecified, а вы уже знаете!

forEach

```
default void forEach(Consumer<? super T> action)
```

Performs the given action for each element of the `Iterable` until all elements have been processed or the action throws an exception. Actions are performed in the order of iteration, if that order is specified. Exceptions thrown by the action are relayed to the caller.

The behavior of this method is unspecified if the action performs side-effects that modify the underlying source of elements, unless an overriding class has specified a concurrent modification policy.

Незаметная разница



@jbaruch

@tagir_valeev

#jpoint

```
@FunctionalInterface
public interface OriginalPredicate<T> {
    boolean test(T t);
}
```

```
OriginalPredicate<Object> lambda = (Object obj) -> "adidas".equals(obj);
OriginalPredicate<Object> methodRef = "adidas"::equals;
```

- A. Оба скомпилируются
- B. Ламбда скомпилируется, ссылка на метод – нет
- C. Ссылка на метод скомпилируется, лямбда – нет
- D. Не функциональный интерфейс!

```
@FunctionalInterface
public interface OriginalPredicate<T> {
    boolean test(T t);
}
```

```
OriginalPredicate<Object> lambda = (Object obj) -> "adidas".equals(obj);
OriginalPredicate<Object> methodRef = "adidas"::equals;
```



Оба скомпилируются

- В. Лямбда скомпилируется, ссылка на метод – нет
- С. Ссылка на метод скомпилируется, лямбда – нет
- Д. Не функциональный интерфейс!




```
@FunctionalInterface
public interface CopyCatPredicate {
    <T> boolean test(T t);
}
```

```
CopyCatPredicate lambda = (Object obj) -> "adadas".equals(obj);
CopyCatPredicate methodRef = "adadas"::equals;
```

- A. Оба скомпилируются
- B. Лямбда скомпилируется, ссылка на метод – нет
- C. Ссылка на метод скомпилируется, лямбда – нет
- D. Не функциональный интерфейс!

```
@FunctionalInterface
public interface CopyCatPredicate {
    <T> boolean test(T t);
}
```

```
CopyCatPredicate lambda = (Object obj) -> "adadas".equals(obj);
CopyCatPredicate methodRef = "adadas"::equals;
```

- A. Оба скомпилируются
- B. Лямбда скомпилируется, ссылка на метод – нет
-  C. Ссылка на метод скомпилируется, лямбда – нет
- D. Не функциональный интерфейс!



```
@FunctionalInterface
public interface CopyCatPredicate {
    <T> boolean test(T t);
}
```

```
CopyCatPredicate lambda = (Object obj) -> "adadas".equals(obj);
CopyCatPredicate methodRef = "adadas"::equals;
```

- A. Оба скомпилируются
- B. Лямбда скомпилируется, ссылка на метод – нет
-  C. Ссылка на метод скомпилируется, лямбда – нет
- D. Не функциональный интерфейс!

```
@FunctionalInterface
public interface CopyCatPredicate {
    <T> boolean test(T t);
}
```

```
CopyCatPredicate lambda = (Object obj) -> "adadas".equals(obj);
```


Example 9.9-2. Generic Function Types

A function type may be generic, as a functional interface's abstract method may be generic. For example, in the following interface hierarchy:

```
interface G1 {  
    <E extends Exception> Object m() throws E;  
}  
interface G2 {  
    <F extends Exception> String m() throws Exception;  
}  
interface G extends G1, G2 {}
```

the function type of G is:

```
<F extends Exception> ()->String throws F
```

A generic function type for a functional interface may be implemented by a method reference expression ([§15.13](#)), but not by a lambda expression ([§15.27](#)) as there is no syntax for generic lambda expressions.

“ A generic function type for a functional interface may be implemented by a method reference expression ([§15.13](#)), but not by a lambda expression ([§15.27](#)) as there is no syntax for generic lambda expressions.



@jbaruch

@tagir_valeev

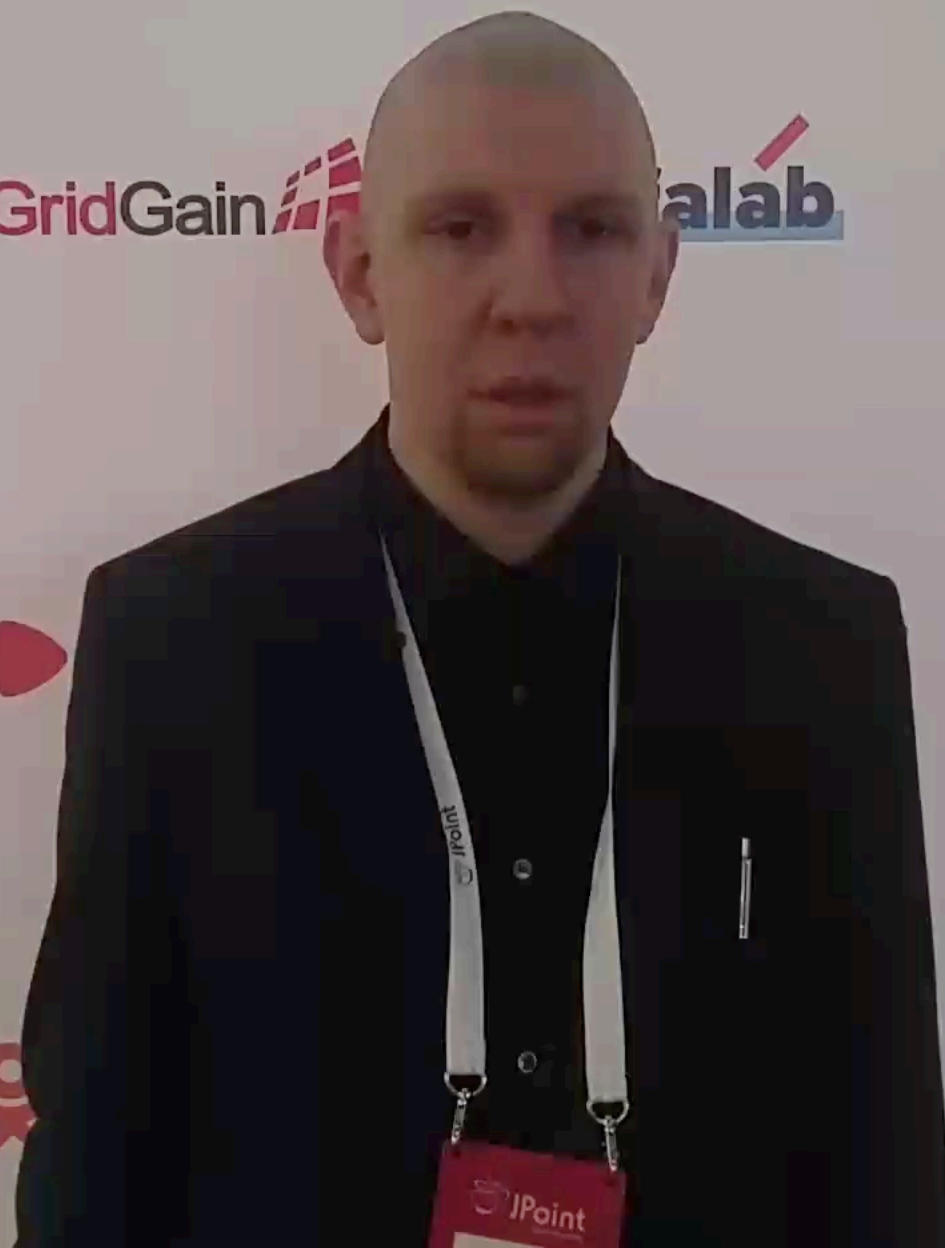
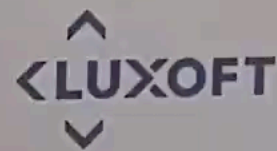
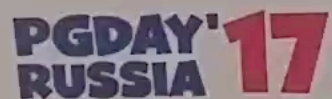
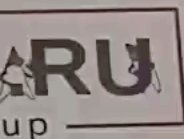
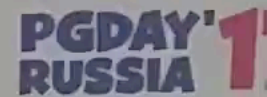
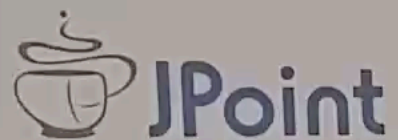
#jpoint



```
List<String> list = Stream.of("Joker", "DotNext", "HolyJS", "HolyJS",  
    "DotNext", "Joker").sequential()  
    .filter(new TreeSet<>()::add).collect(Collectors.toList());  
System.out.println(list);
```

- A. [DotNext, HolyJS, Joker]
- B. [Joker, DotNext, HolyJS, HolyJS, DotNext, Joker]
- C. [Joker, DotNext, HolyJS]
- D. [DotNext, DotNext, HolyJS, HolyJS, Joker, Joker]

НК
ОГИИ




```
List<String> list = Stream.of("Joker", "DotNext", "HolyJS", "HolyJS",  
"DotNext", "Joker").sequential()  
    .filter(new TreeSet<>()::add).collect(Collectors.toList());  
System.out.println(list);
```

- A. [DotNext, HolyJS, Joker]
- B. [Joker, DotNext, HolyJS, HolyJS, DotNext, Joker]
- C. [Joker, DotNext, HolyJS]
- D. [DotNext, DotNext, HolyJS, HolyJS, Joker, Joker]

```
List<String> list = Stream.of("Joker", "DotNext", "HolyJS", "HolyJS",  
"DotNext", "Joker").sequential()  
    .filter(new TreeSet<>()::add).collect(Collectors.toList());  
System.out.println(list);
```

A. [DotNext, HolyJS, Joker]

B. [Joker, DotNext, HolyJS, HolyJS, DotNext, Joker]



[Joker, DotNext, HolyJS]

D. [DotNext, DotNext, HolyJS, HolyJS, Joker, Joker]

РБАНК
ОЛОГИИ

JPoint

GridGain

alfalab

JUGURU
group

PGDAY
RUSSIA

JUGURU
group

PGDAY'
RUSSIA 17

LUXOFT

CU
pla

PGDAY'
RUSSIA 17

zalando

CUBA
platform

IT
BRANDS

```
List<String> list = Stream.of("Joker", "DotNext", "HolyJS", "HolyJS",  
"DotNext", "Joker").sequential()  
    .filter(new TreeSet<>()::add).collect(Collectors.toList());  
System.out.println(list);
```

A. [DotNext, HolyJS, Joker]

B. [Joker, DotNext, HolyJS, HolyJS, DotNext, Joker]



[Joker, DotNext, HolyJS]

D. [DotNext, DotNext, HolyJS, HolyJS, Joker, Joker]

```
filter(new TreeSet<>()::add)
```

!=

```
filter(i -> new TreeSet<>().add(i))
```

Тот же объект!

Новый объект
каждый раз!

@jbaruch

@tagir_valeev

#jpoint

Выводы



@jbaruch

@tagir_valeev

#jpoint

- ПИШИТЕ ЧИТАЕМАЯ КОД!
- КОММЕНТИРУЙТЕ ВСЕ ТРЮКИ
- ИНОГДА ДАЖЕ В ДЖАВЕ БЫВАЮТ БАГИ
- СТАТИЧЕСКИЕ АНАЛИЗАТОРЫ КОДА РУЛЯТ! INTELLIJ IDEA!
- RTFM!
- НЕ БОЛЕЙТЕ СТРИМОЗОМ!

-САМИ ПОНИМАЕТЕ, JAVA 8 И
9 - КЛАДЕЗЬ ПАЗЗЛЕРОВ

-ЕСЛИ ВЫ НАТКНУЛИСЬ НА
ПАЗЗЛЕР, ДАВАЙТЕ ЕГО СЮДА!

-PUZZLERS@JFROG.COM

-ПОНРАВИЛОСЬ?

-ХВАЛИТЕ НАС СКОРЕЕ В TWITTERE!

-#JAVAPUZZLERSNG #JPOINT

-@TAGIR_VALEEV

-@JBARUCH

-НЕ ПОНРАВИЛОСЬ?

-/DEV/NULL