JAVA 8 PUZZLERS

ДУМАЛИ ВЫ ЗНАЕТЕ ЭТУ ВАШУ НАДЕЖНУЮ И ПОНЯТНУЮ ДЖАВУ?!

Краткое содержание

Любите нас в Твиттере!

#jpoint
#java8puzzlers
@tagir_valeev
@jekaborisov
@jbaruch
@real_jbaruch (мы не знаем, кто это, но он ОК)

История про забытые яйца

Что выведет следующий код?

```
List<String> list = new ArrayList<>();
list.add("молоко");
list.add("хлеб");
list.add("колбаса");
Stream<String> stream = list.stream();
list.add("яйца, яйца ещё!");
stream.forEach(System.out::println);
```

A. молоко/хлеб/колбаса

молоко/хлеб/колбаса/яйца, яйца ещё!

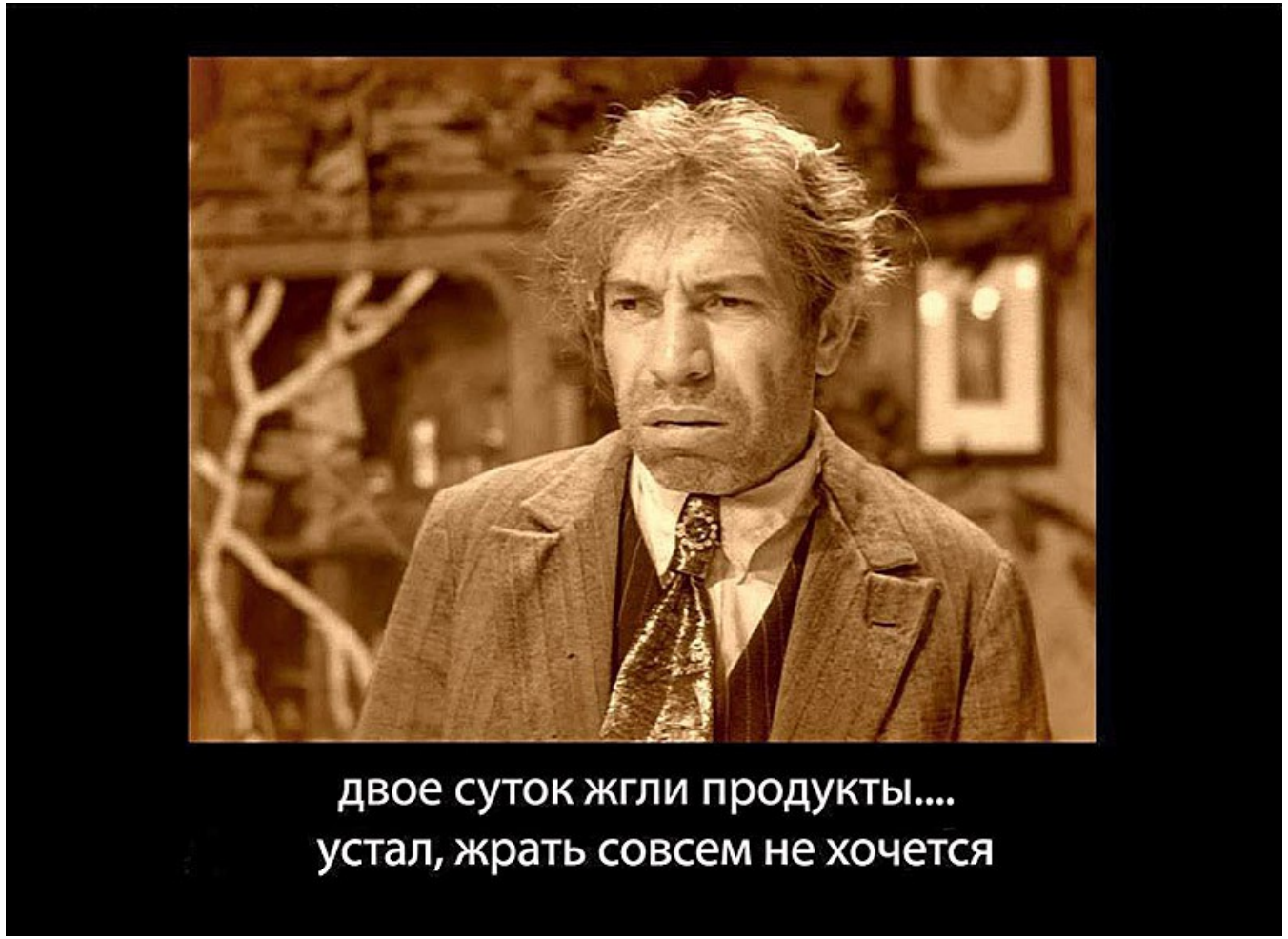
C. молоко/хлеб/колбаса/ConcurrentModificationException

D. ConcurrentModificationException

Late binding же

```
List<String> list = new ArrayList<>();
list.add("молоко");
list.add("хлеб");
list.add("колбаса");
Stream<String> stream = list.stream();
list.add("яйца, яйца ещё!");
stream.forEach(System.out::println);
```

Колбасы не надо



Что выведет следующий код?

```
List<String> list = new ArrayList<String>();
list.add("молоко");
list.add("хлеб");
list.add("колбаса");
list = list.subList(0, 2); //не надо колбасу!
Stream<String> stream = list.stream();
list.add("яйца, яйца ещё!");
stream.forEach(System.out::println);
```

A. молоко/хлеб/колбаса
B. молоко/хлеб/колбаса/яйца, яйца ещё!

молоко/хлеб/колбаса/ConcurrentModificationException
D. молоко/хлеб/яйца, яйца ещё!

Я В ШОКЕ МАЛЕНЬКО)))*
risovach.ru

- Баг!!!

JDK / JDK-8148748

ArrayList.subList().spliterator() is not late-binding

Agile Board

Details

Type:	Bug	Status:	RESOLVED
Priority:	P4	Resolution:	Fixed
Affects Version/s:	9	Fix Version/s:	9
Component/s:	core-libs		
Labels:	None		
Subcomponent:	java.util:collections		
Introduced In Version:	8		
Resolved In Build:	b112		

People

Assignee:

Tagir Valeev

Reporter:

Tagir Valeev

Votes:

0 Vote for this issue

Watchers:

3 Start watching this issue

Сага о палаче

В чём разница между строчками 1 и 2?

```
public void killAll(){
    ExecutorService ex = Executors.newSingleThreadExecutor();
    List<String> sentence = Arrays.asList("Казнить");
    ex.submit(() ->   Files.write(Paths.get("Приговор.txt"), sentence)   ); // 1
    ex.submit(() -> { Files.write(Paths.get("Приговор.txt"), sentence); }); // 2
}
```

1 компилируется, 2 нет

B. 2 компилируется, 1 нет

C. Что в лоб, что по лбу, обе нормально сработают.

D. Без разницы, обе не компилируются.

Точки с запятыми – зло!

В чём разница между строчками 1 и 2?

```
public void killAll(){
    ExecutorService ex = Executors.newSingleThreadExecutor();
    List<String> sentence = Arrays.asList("Казнить");
    ex.submit(() ->   Files.write(Paths.get("Приговор.txt"), sentence)   ); // 1
    ex.submit(() -> { Files.write(Paths.get("Приговор.txt"), sentence); }); // 2
}
```

1 компилируется, 2 нет

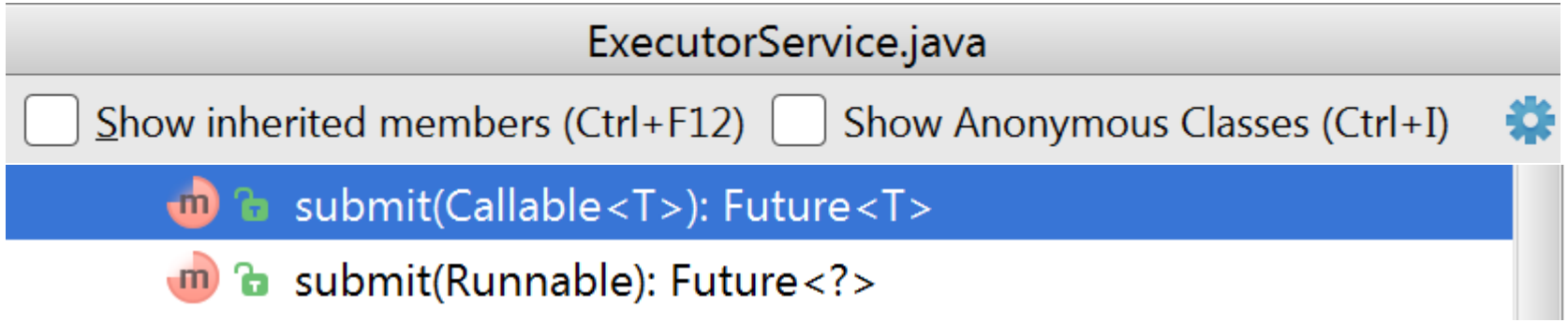
B. 2 компилируется, 1 нет

C. Что в лоб, что по лбу, обе нормально сработают.

D. Без разницы, обе не компилируются.

```java
@FunctionalInterface
public interface Runnable {
        public abstract void run();
}
```

```java
@FunctionalInterface
public interface Callable<V> {
        V call() throws Exception;
}
```




```java
public void killAll(){
    ExecutorService ex = Executors.newSingleThreadExecutor();
    List<String> sentence = Arrays.asList("Казнить");
    ex.submit(() ->   Files.write(Paths.get("Приговор.txt"), sentence)   ); // 1
    ex.submit(() -> { Files.write(Paths.get("Приговор.txt"), sentence); }); // 2
}
```

Безумный Макс

Что будет?

```
System.out.println(
        Stream.of(-3, -2, -1, 0, 1, 2, 3).max(Math::max).get()
);
```

A. Ошибка компиляции
B. Вылетит с исключением
C. 3
D. Что-то другое

Что будет?

```
System.out.println(
        Stream.of(-3, -2, -1, 0, 1, 2, 3).max(Math::max).get()
);
```

D. Что-то другое:

A. -3

B. -1

C. 0

D. 1

MAKE GIFS AT GIFSOUP.COM

```
Stream.of(-3, -2, -1, 0, 1, 2, 3).max(Math::max).get()
```

- Math.max(−3, −2) = −2 < 0 → −3 < −2, берём −2
- Math.max(−2, −1) = −1 < 0 → −2 < −1, берём −1
- Math.max(−1, 0) = 0 → −1 == 0, оставляем −1
- Math.max(−1, 1) = 1 > 0 → −1 > 1, оставляем −1
- Math.max(−1, 2) = 2 > 0 → −1 > 2, оставляем −1
- Math.max(−1, 3) = 3 > 0 → −1 > 3, оставляем −1

Настало время апгрейда

Что произойдёт после выполнения?

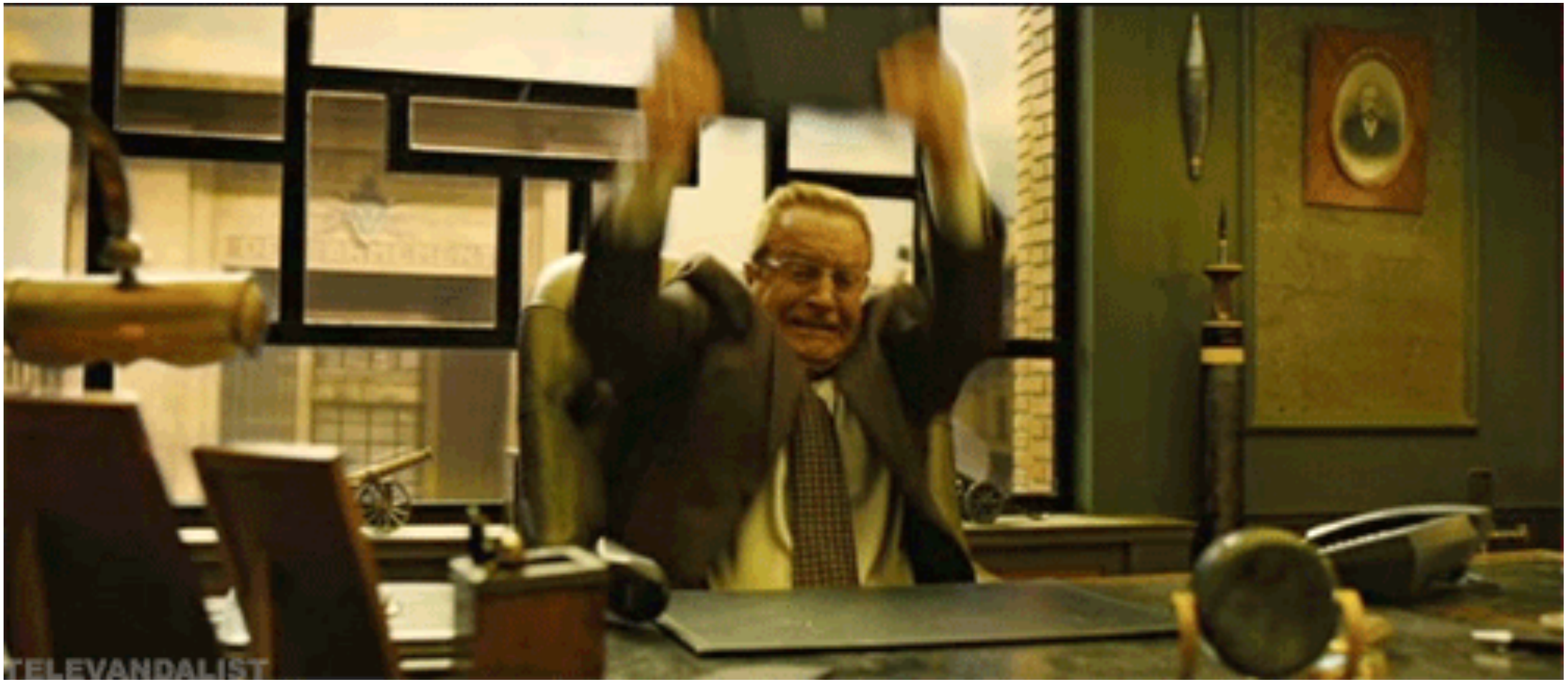
```
Map<String, String> oldSchool = initOldSchool();
// oldSchool = {buildTool=maven, lang=java, IOC=jee}
Map<String, String> proper = initКакНадо();
// proper = {buildTool=gradle, lang=groovy, IOC=spring}

oldSchool.replaceAll(proper::put);
```

Мапы поменяются местами

B. Обе станут олдскульными

C. Обе станут как надо

D. Поучитесь программировать! Это вообще не скомпилируется!


TELEVANDALIST

Map interface

```
V put(K key, V value);
```

```
void replaceAll(BiFunction<? super K, ? super V, ? extends V> function)
```

```
oldSchool.replaceAll(proper::put);
```

Одинаково ли сработают три способа?

```
List<String> list = Arrays.asList("Вронский", "Поезд", "Анна");
Comparator<String> cmp = Comparator.nullsLast(Comparator.naturalOrder());

System.out.println(Collections.max(list, cmp));
System.out.println(list.stream().collect(Collectors.maxBy(cmp)).get());
System.out.println(list.stream().max(cmp).get());
```

Все ответы будут одинаковые

B. Будет два разных ответа

C. Будет три разных ответа

D. Будет четыре разных ответа

А пазлер в чём?

А теперь?

```
List<String> list = Arrays.asList("Вронский", null, "Анна");
Comparator<String> cmp = Comparator.nullsLast(Comparator.naturalOrder());

System.out.println(Collections.max(list, cmp));
System.out.println(list.stream().collect(Collectors.maxBy(cmp)).get());
System.out.println(list.stream().max(cmp).get());
```

A. Все ответы будут одинаковые
B. Будет два разных ответа
C. Будет три разных ответа
D. Будет четыре разных ответа

Вот теперь убиться

Объяснение тут

- null
- NoSuchElementException
- NullPointerException

```
List<String> list = Arrays.asList("Вронский", null, "Анна");
Comparator<String> cmp = Comparator.nullsLast(Comparator.naturalOrder());

System.out.println(Collections.max(list, cmp));
System.out.println(list.stream().collect(Collectors.maxBy(cmp)).get());
System.out.println(list.stream().max(cmp).get());
```

Мутанты

Как сделать тип, не объявляя его?

```
interface Кот{ default void мяукать() {System.out.println("мяу");}}
interface Пёс{ default void лаять() {System.out.println("гав");}}

public static void main(String[] args) {
    class Котопёс implements Кот, Пёс {}
    test(new Котопёс());
}

static void test(Object obj) {
      def x = (?)obj;
      x.мяукать();
      x.лаять();
}
```

Какой вариант сработает?

```
interface Кот{ default void мяукать() {System.out.println("мяу");}}
interface Пёс{ default void лаять() {System.out.println("гав");}}

public static void main(String[] args) {
    class Котопёс implements Кот, Пёс {}
    test(new Котопёс());
}
```

```
static void test(Object obj) {
    // A. Это сработает?
Кот & Пёс x = (Кот & Пёс) obj;
x.мяукать();
x.лаять();
}
```

```
static void test(Object obj) {
    // B. Это сработает?
    ((Consumer<? extends Кот & Пёс>)(x -> {
    x.мяукать();
    x.лаять();
})).accept((Кот & Пёс)obj); }
```

```
static void test(Object obj) {
    // C. Это сработает?
    Optional.of((Кот & Пёс) obj)
        .ifPresent(x -> {
            x.мяукать();
            x.лаять();
        });}
```

```
// D. Да вы упоролись, в моей Джаве такого не бывает!
```

CAT-GIFS.COM
Sir are you aware you are a cat?

А как это вообще работает???

```
static void test(Object obj) {
    // С. Это сработает?
    Optional.of((Кот & Пёс) obj)
            .ifPresent(x -> {
                x.мяукать();
                x.лаять();
            });}
```

Трудности перевода

Что выведет?

```
public class Test {
    String str;

    void run() {
        str = "привет";
        Supplier<String> s1 = str::toUpperCase;
        Supplier<String> s2 = () -> str.toUpperCase();
        str = "hello";
        System.out.println(s1.get());
        System.out.println(s2.get());
    }
}
```

A. ПРИВЕТ/ПРИВЕТ

ПРИВЕТ/HELLO

C. HELLO/ПРИВЕТ

D. HELLO/HELLO

```java
public class Test {
    String str;

    void run() {
        str = "привет";
        Supplier<String> s1 = str::toUpperCase;
        Supplier<String> s2 = () -> str.toUpperCase();
        str = "hello";
        System.out.println(s1.get());
        System.out.println(s2.get());
    }
}
```

BLACKHAWK!

Что произойдёт?

```
List<String> list = new ArrayList<>(Arrays.asList("Arne", "Chuck", "Slay"));
list.stream().forEach(x -> {
    if(x.equals("Chuck")) {
        list.remove(x);
    }
});
```

A. ConcurrentModificationException
B. ArrayIndexOutOfBoundsException
C. NullPointerException
D. Успешно завершится

GIFSec.com

Java 8 vs Chuck Norris

Ад вот почему:

- stream().forEach() → spliterator().forEachRemaining()
- forEachRemaining проверяет modcount один раз в конце
- Удаление элемента приведёт к сдвигу массива с добавлением null в конце: [“Arne”, “Chuck”, “Slay”] → [“Arne”, “Slay”, null]
- На последней итерации if(null.equals(“Chuck”)) упадёт с NPE (до CME не дожили)
- Используйте list.removeIf(“Chuck”::equals);

4GIFs.com

Выводы

WITH GREAT
POWER
COMES GREAT
RESPONSIBILITY
PETER PARKER is SPIDER MAN

1. Пишите читабельный
2. Комментируйте все тр
3. Иногда это баг
4. Пользуйте static code a
   IDEA!
5. Rtfm
6. Хм, а groovy не так уж и плох!

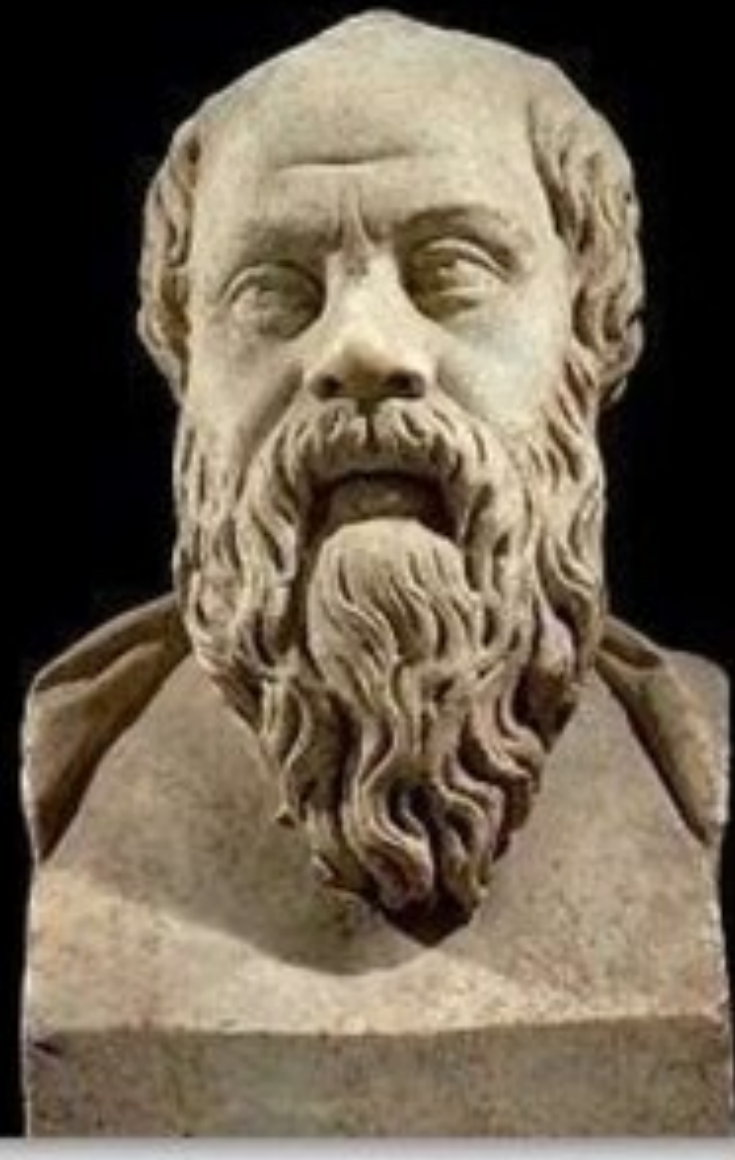
Паззлеров, конечно, полно.

Наткнулись на паззлер? Давайте его сюда!

Понравилось?
Хвалите нас в твиттере скорее!

- #java8puzzlers #jpoint
- @tagir_valeev
- @akaborisov
- @baruch

Не понравилось?
/dev/null

"Расставашки -
всегда пичалька".

(с) Сократик