Vavr vs the World: меряемся фичами

Даниил Царёв

CUBA.platform

HAULMONT

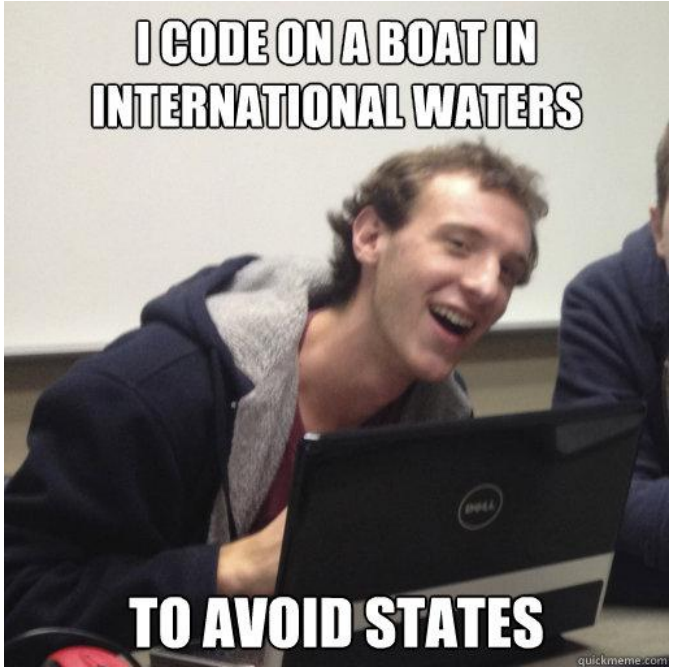
Table of Contents

Три столпа функционального программирования

1. Состояние отсутствует
2. Чистые функции
3. Рекурсия

Три столпа ФП на самом деле

1. Холивары
2. Мемы
3. Мунспик*



*Мунспик (вольный перевод) - язык, использующий большое количество заимствований из предметной области, что приводит к непониманию сути.

Дисклеймер

ФП - инструмент для достижения цели, а не панацея.

Не предлагаю перейти на тёмную ~~тему~~ / светлую сторону.

Результат не гарантирован, использовать с осторожностью.

О, Vavr!

Vavr (*aka Javaslang*) - библиотека, ~~добавляющая~~ расширяющая поддержку ФП в Java.

Разрабатывается с 2014 года (5 лет) силами @danieldietrich.

GitHub: [vavr-io/vavr]

FunctionalJava

Имеет схожие возможности и функциональность,

но пациент скорее мёртв:

Home » org.functionaljava » functionaljava

Functional Java

Functional Java is an open source library that supports closures for the Java programming language

Central (16) | Kantega (1)

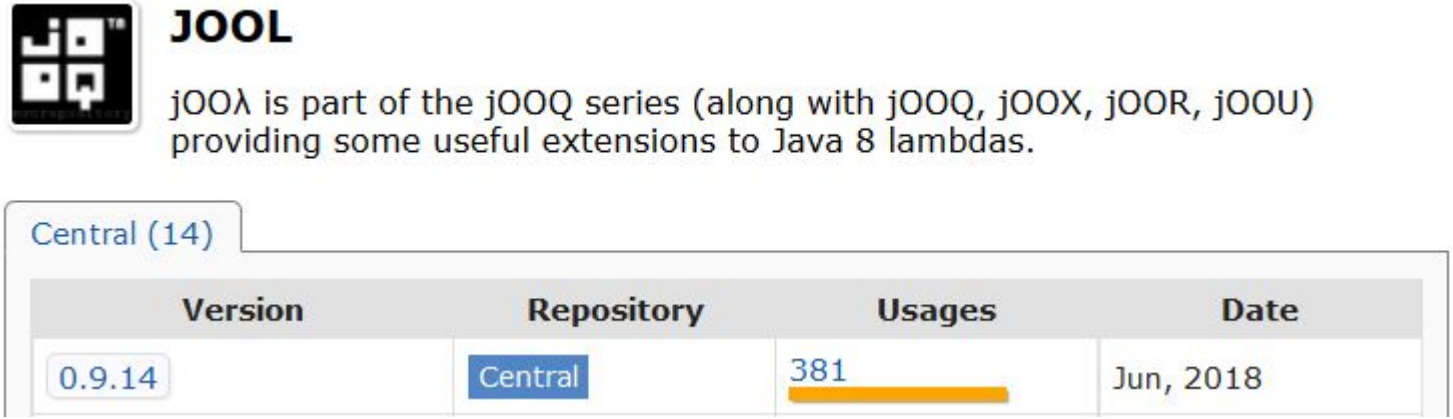
	Version	Repository	Usages	Date
4.8.x	4.8.1	Central	4	Oct, 2018
	4.8	Central	3	Aug, 2018

jOOL

Субпродукт jOOQ - ещё ничего не значит:

Единственная документация - README.md

С релизами не лучше:



JOOL

jOOλ is part of the jOOQ series (along with jOOQ, jOOX, jOOR, jOOU) providing some useful extensions to Java 8 lambdas.

Central (14)

Version	Repository	Usages	Date
0.9.14	Central	381	Jun, 2018

Выбора нет

- Примерно одни и те же возможности
- Проблемы с документацией
- Выжили не все

“First Class Citizen”

Java

```java
public static void firstClassCitizen() {
    Function<Integer, String> foo = num ->
            num == 42 ? "Answer" : "Not answer";

    System.out.println(foo.apply( t: 42));
}
```

Kotlin

```kotlin
fun firstClassCitizen() {
    val foo: (Int) -> String = { it ->
        if (it == 42) "Answer" else "Not answer"
    }
    println(foo(42))
}
```

Scala

```scala
def firstClassCitizen(): Unit = {
  val foo = (num: Int) =>
    if (num == 42) "Answer" else "Not answer"

  println(foo(42))
}
```

Композиция функций

Последовательная обработка результатов:

```
public static void composition() {
    Function0<Integer> getRandom = () ->
            new Random().nextInt( bound: 100);

    Boolean result = getRandom
            .andThen(JFunc::isAnswer) Function0<String>
            .andThen(JFunc::charsCount) Function0<Integer>
            .andThen(JFunc::isEven) Function0<Boolean>
            .apply();
}
```

Горизонт завален, и композиция не очень

Scala:

```scala
val composed =
  ((num: Int) =>
    if (num == 42) "Answer"
    else "Not answer")
  .andThen((s: String) => s.length)
  .andThen((n: Int) => n % 2 == 0)

val result = composed(new Random()
  .nextInt( bound = 100))
```

Kotlin:

```kotlin
fun <A, B> compose(g: (A) -> B,
                   f: (B) -> A): (A) -> A {
    return { it -> f(g(it))}
}

fun composition() {
    val foo: (Int) -> String = { it ->
        if (it == 42) "Answer"  ^lambda
        else "Not answer" ^lambda
    }
    val composed = compose(foo, { s -> s.length})
}
```

Мемоизация

```
public static void memoization() {
    int[] counter = {0};
    Function1<Integer, Integer> compute = n -> {
        counter[0]++;
        return n;
    };
    Function1<Integer, Integer> memoized =
            compute.memoized();

    memoized.apply( t1: 42);
    memoized.apply( t1: 42);

    assert counter[0] == 1;
}
```

Kotlin / Scala:

Lazy - я уже работал на этой неделе

Memoized Supplier:

```
public static void lazy() {
    // Trust me - i'm engineer
    Lazy<Integer> getRandom = Lazy.of(
            () -> 4);

    assert getRandom.get() == 4;
}
```

Lazy - Kotlin / Scala

```kotlin
fun lazy() {
    val getRandom = lazy { -> 4 }

    assert(getRandom.value == 4)
}
```

```scala
def scalaLazy(): Unit = {
  lazy val getRandom = () => 4

  assert(getRandom() == 4)
}
```

Попытка - не пытка

Try - контейнер для “небезопасных” операций

```java
public static void javaCompute() {
    int computed;
    try {
        computed = compute();
    } catch (Exception e) {
        computed = -1;
    }

    assert computed == -1;
}
```

```java
public static void vavrCompute() {
    Try.of(JFunc::compute)
            .onFailure((ex) -> {
                // Handle exception
            })
            .andThen(JFunc::doSomethingElse);
}
```

Try в Kotlin / Scala

```scala
def tryToCompute(): Unit = {
  val res = tryTo(compute)

  assert(res == null)
}

def tryTo(f: () => Int): Any = {
  try {
    f()
  } catch {
    case _: Exception => null
  }
}
```

Either - быть или не быть

Долой POJO для результатов!

```
public static Either<String, Integer> doSomething() {
    if (isGoingWrong()) {
        // Error value
        return Either.left("Something is going wrong");
    }
    // Success value
    return Either.right(compute());
}
```

Either + Try

```
public static void eitherDemo() {
    Integer result = process()
            .getOrElseThrow((msg) ->
                    new RuntimeException(msg));
}

public static Either<String, Integer> process() {
    return Try.of(JFunc::compute)
            .toEither()
            .mapLeft(Throwable::getMessage);
}
```

И на нашей улице будет Pattern Matching!

Kotlin:

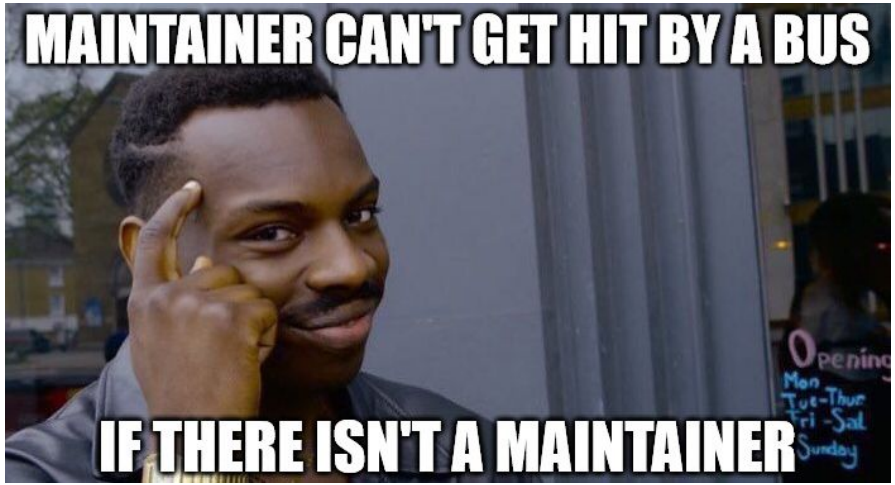
```kotlin
fun patternMatching() {
    val x = 42;
    val result = when (x) {
        42 -> "Answer"
        else -> "Not Answer"
    }
    assert("Answer".equals(result))
}
```

Vavr

```vavr
public static void patternMatching() {
    Integer x = 42;
    String res = Match(x).of(
            Case($(42), "Answer"),
            Case($(), "Not answer"));
    assert "Answer".equals(res);
}
```

Что с Vavr не так?

- Неактуален?
- Документация
- Фактически поддерживается в одиночку



Нужно ли оно нам?

“Use vavr if you want to program in scala and your boss don’t let you.”

Reddit

Vavr - динозавр?

Всё ещё ~~подаёт признаки жизни~~ релизится.

Теоретически неограниченная область применения.

Пишутся посты:

- https://www.baeldung.com/vavr-tutorial
- https://tomassetti.me/functional-programming-for-java-getting-started-with-javaslang/
- https://medium.com/@sderosiaux/using-vavr-to-write-more-robust-java-code-87ccb0be12d7

Всё!

Искать меня здесь:

Email: daniil.tsaryov@gmail.com

Tg: @d_tsaryov