Мифы о Спарке

О себе

- Пишу на джаве с 2001
- Преподаю джаву с 2003
- Консультирую с 2008
- Арихтектор с 2009
- Пилю стартап с 2014
- Техлид по биг дате с 2015
- Naya technologies с 2015

За что я люблю Питер…

Иду я как то с барышней, да по Невскому…

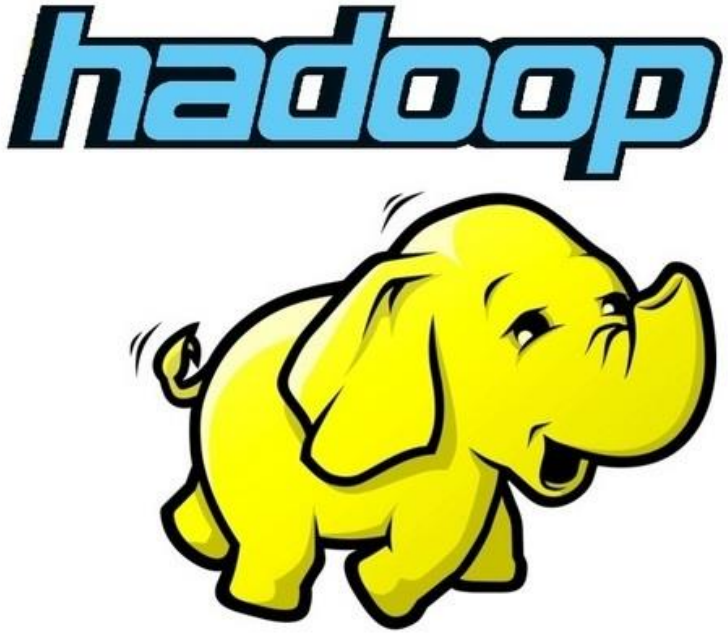
Опровержение мифов

- Концептуальные мифы
  - Spark это какая то хрень для Hadoop-а
  - Spark надо писать на скале
  - Spark и Spring не совместимы, и вообще там всё по другому
  - Для Spark-а ~~нельзя~~ сложно писать unit тесты
  - Чтобы запускать спарк нужен как минимум докер, а лучше кубернетис
- Главный миф

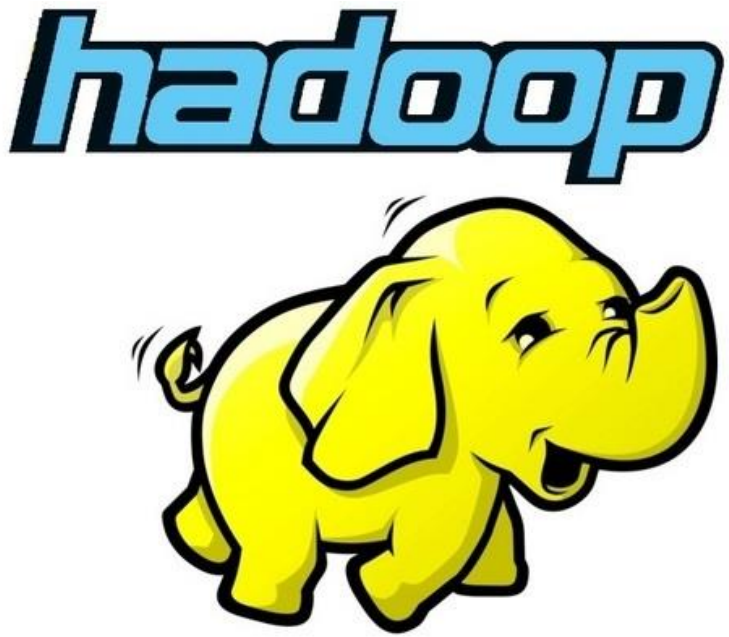
Главный миф:

Pink Floyd это вам не Бритни Спирс или Кэтти Пери

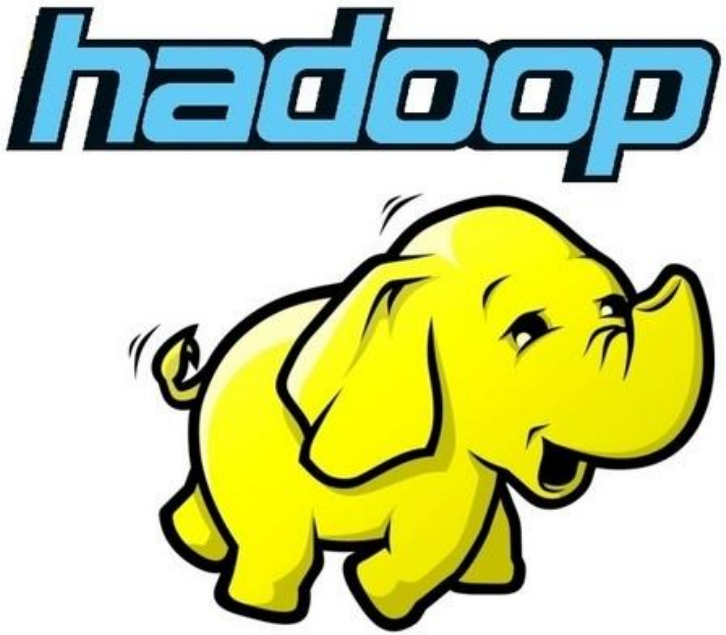
Миф первый о Spark-е и Hadoop-е



Что вообще мы знаем про Hadoop



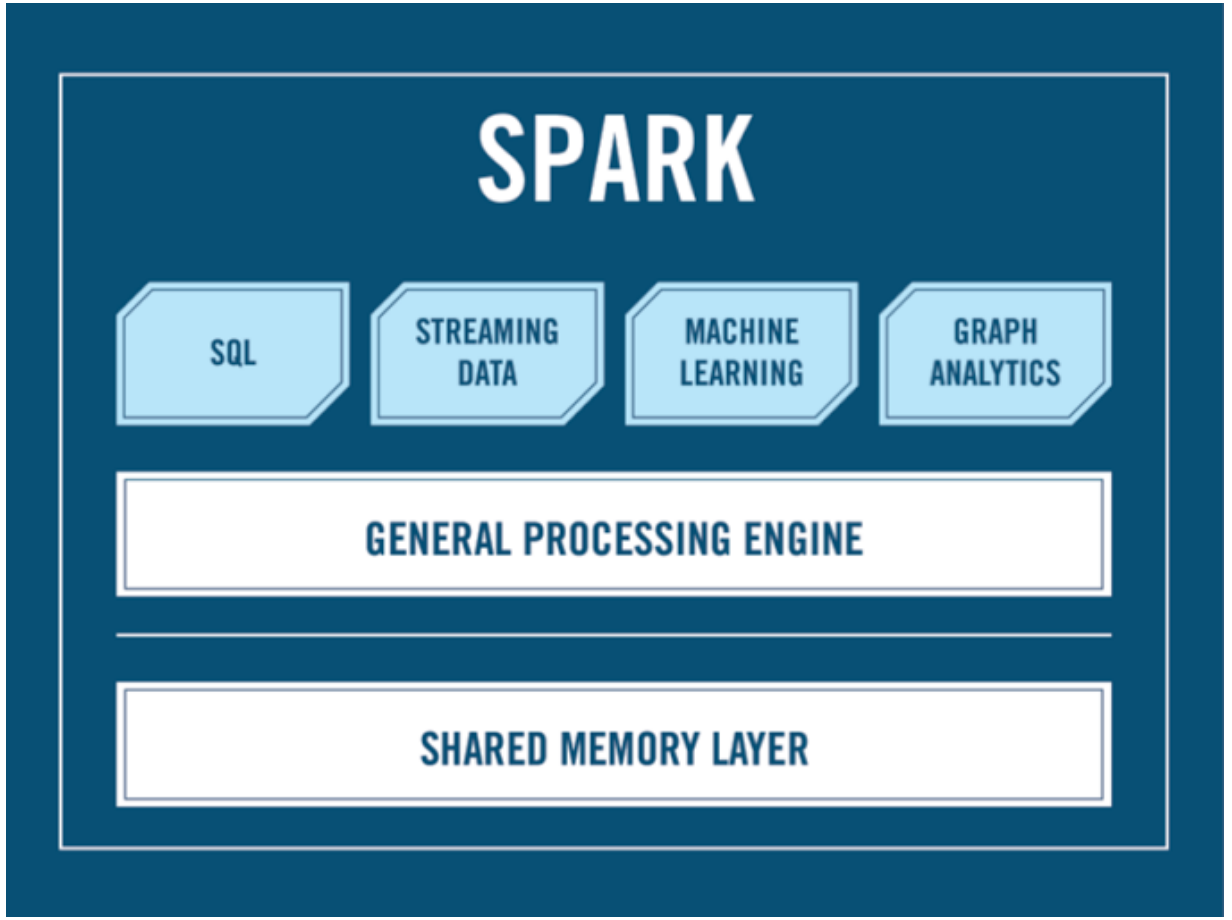
Говно ваш Hadoop



Data locality


ЕСЛИ ГОРА НЕ ИДЁТ К МАГОМЕТУ
ЗНАЧИТ МАГОМЕТА ОТПУСТИЛО

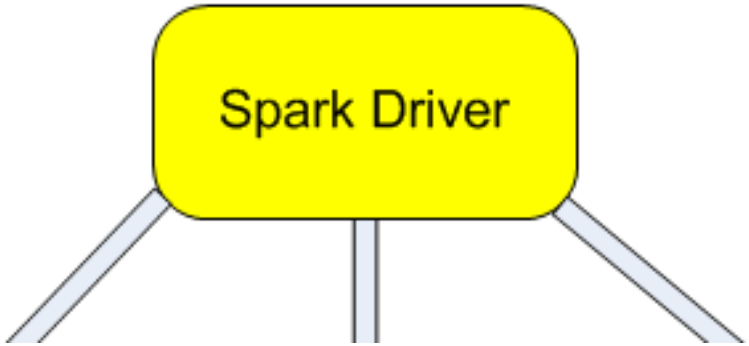
И при чём тут Hadoop?



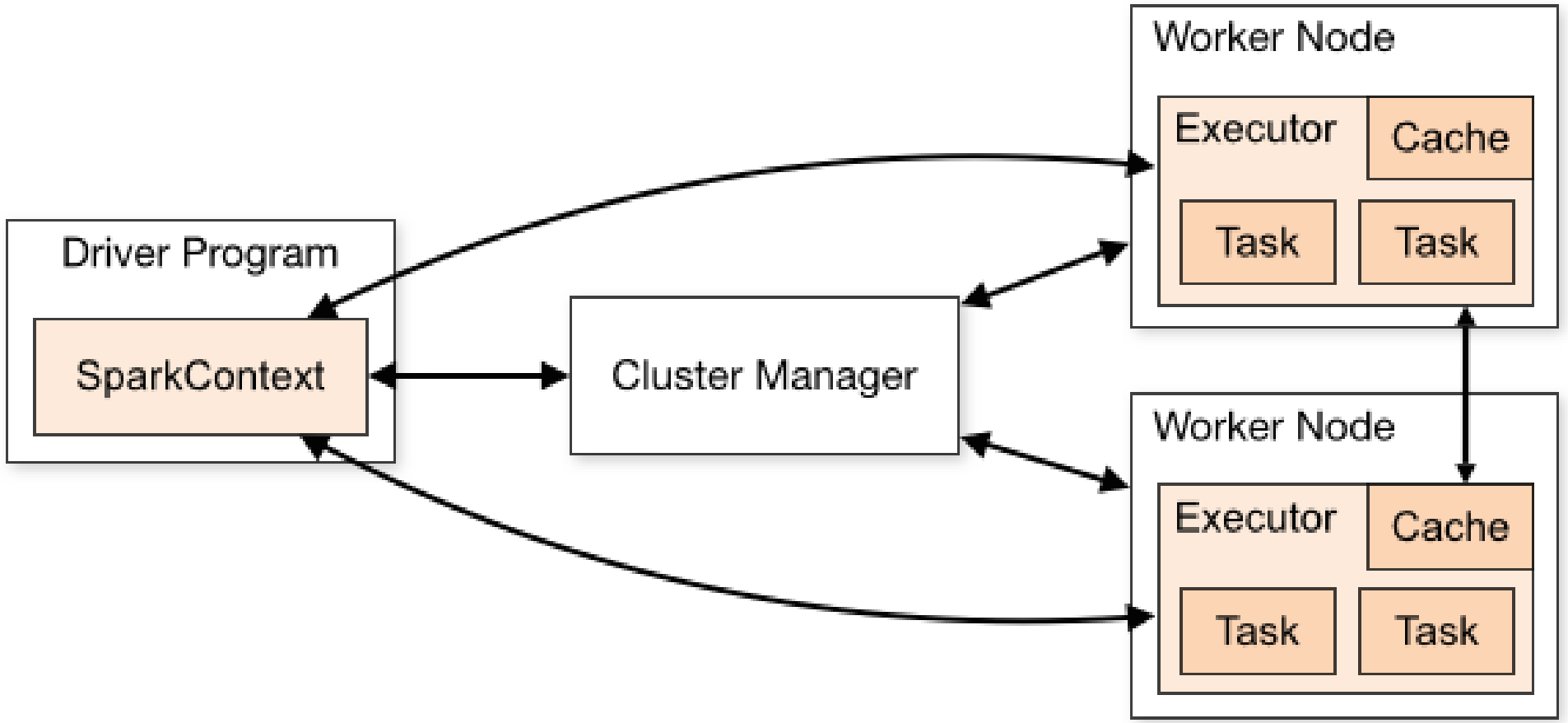
Spark

- Развитие
  - Идея зародилась в UC Berkeley примерно в 2009
  - Первый релиз в 2012
  - В января 2016 Spark 1.6
  - Сегодня Spark 2.3+
- Написан на скале
- Есть API для
  - Scala, Python, Java, Java 8, R
- Где писать
  - InteliJ, Eclipse, Spark-shell, Notebooks
- Запускать
  - Spark-Shell, Notebooks, Spark-submit
  - Как обычную Джаву
- Что нужно чтобы можно было запускать на кластере
  - Какой-нибудь cluster manager

Driver, driver, а что такое driver?

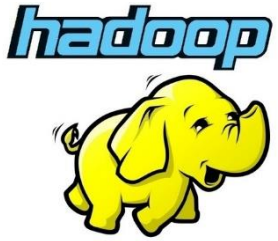


Как оно работает на кластере



Cluster manager

- Yarn

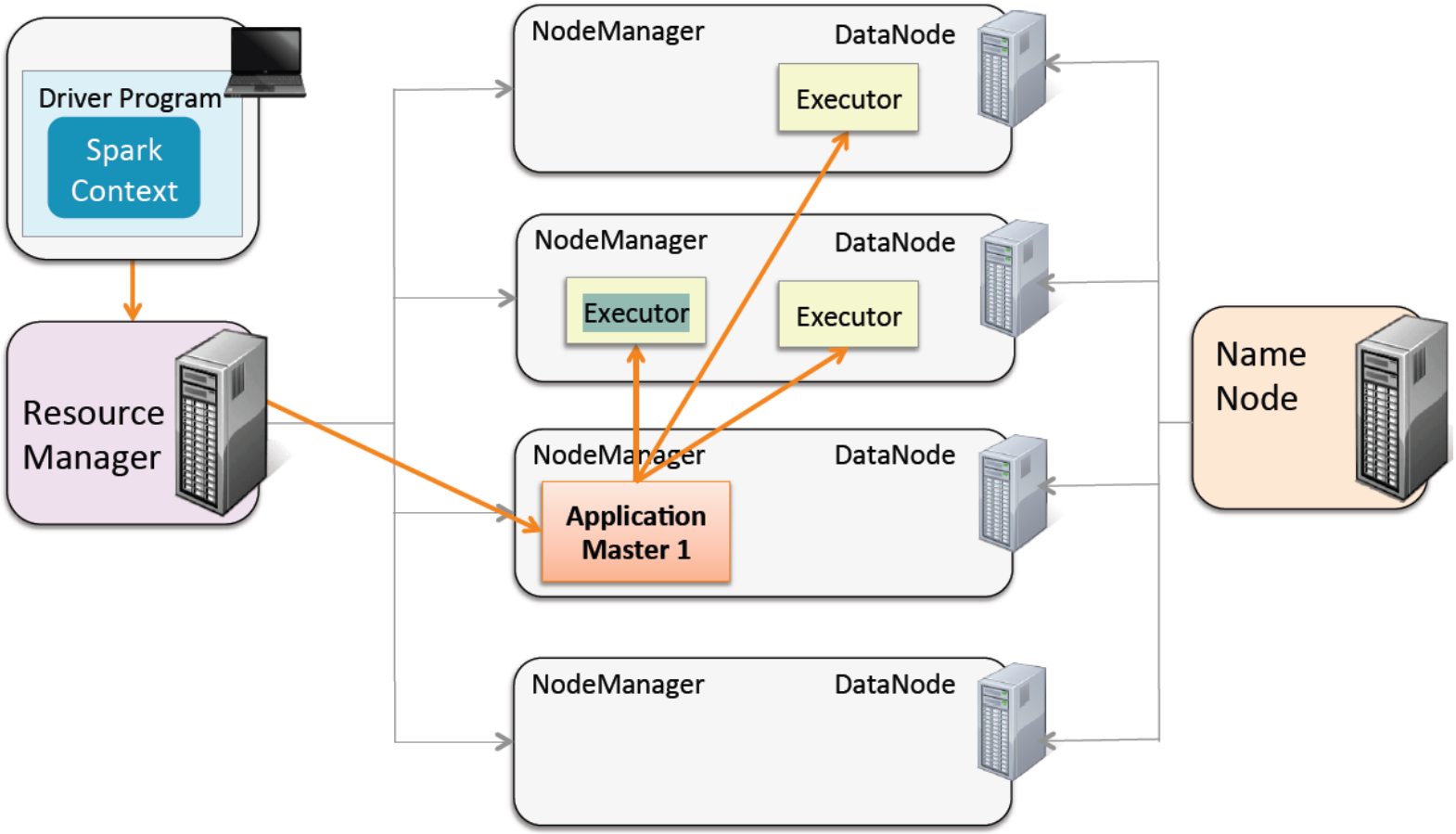


DATASTAX

- Mesos

Spark with yarn

Очень важно!
Часть кода исполняется на кластере
Часть исполняется на драйвере



RDD

Resilient

Distributed

Dataset

+-Stream

Как можно получить RDD?

- Из файла (файлов в директории)
- Из памяти (обычно используется для тестов)
- Из другого RDD (ну да да, как Стримы)

Из файла / директории

```
// from local file system
JavaRDD<String> rdd = sc.textFile("file:/home/data/data.txt");

// from Hadoop using relative path of user, who run spark
application
rdd = sc.textFile("/data/data.txt")

// from hadoop
rdd = sc.textFile("hdfs://data/data.txt")

// all files from directory
rdd = sc.textFile("s3://data/*")

// all txt files from directory
rdd = sc.textFile("s3://data/*.txt")
```

Из памяти

```
sc.pa
```

```
parallelize(List<T> list)                                  JavaRDD<T>
parallelize(List<T> list, int numSlices)                   JavaRDD<T>
parallelizeDoubles(List<Double> list)                      JavaDoubleRDD
parallelizeDoubles(List<Double> list, int numSl…           JavaDoubleRDD
parallelizePairs(List<Tuple2<K, V>> list)                  JavaPairRDD<K, V>
parallelizePairs(List<Tuple2<K, V>> list, i…               JavaPairRDD<K, V>
```

А что это за sc?

- SparkContext / JavaSparkContext
- Main entry point for Spark functionality

```
SparkConf conf = new SparkConf();
conf.setAppName("my spark application");
conf.setMaster("local[*]");
JavaSparkContext sc = new JavaSparkContext(conf);
```

Spark Context локально и продакшн

```java
@Bean
@Profile("LOCAL")
public JavaSparkContext sc() {
    SparkConf conf = new SparkConf();
    conf.setAppName("music analyst");
    conf.setMaster("local[1]");
    return new JavaSparkContext(conf);
}
```

```java
@Bean
@Profile("PROD")
public JavaSparkContext sc() {
    SparkConf conf = new SparkConf();
    conf.setAppName("music analyst");
    return new JavaSparkContext(conf);
}
```

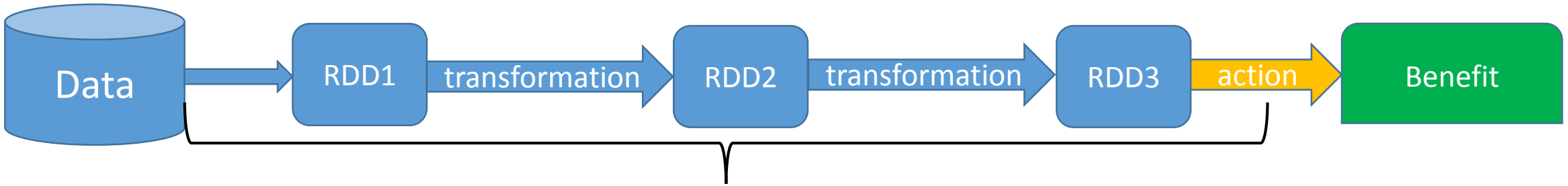
Transformations

- **map**
- **flatMap**
- **filter**
- **mapPartitions, mapPartitionsWithIndex**
- **sample**
- **union, intersection, join, cogroup, cartesian** (*otherDataset*)
- **distinct**
- **reduceByKey, aggregateByKey, sortByKey**
- **pipe**
- **coalesce, repartition, repartitionAndSortWithinPartitions**

Actions

- **reduce**
- **collect**
- **count, countByKey, countByValue**
- **first**
- **take, takeSample, takeOrdered**
- **saveAsTextFile, saveAsSequenceFile, saveAsObjectFile**
- **foreach**

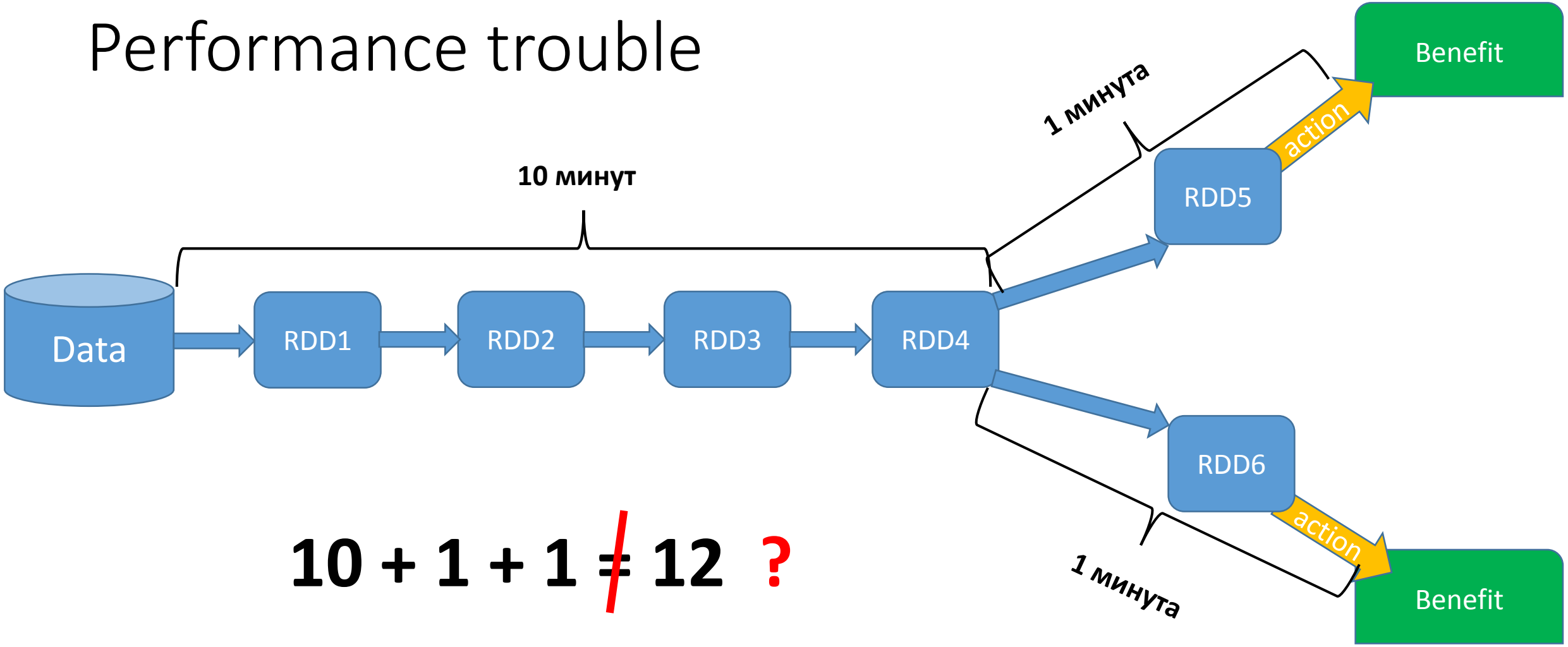
Rdd flow



Выполняется на кластере

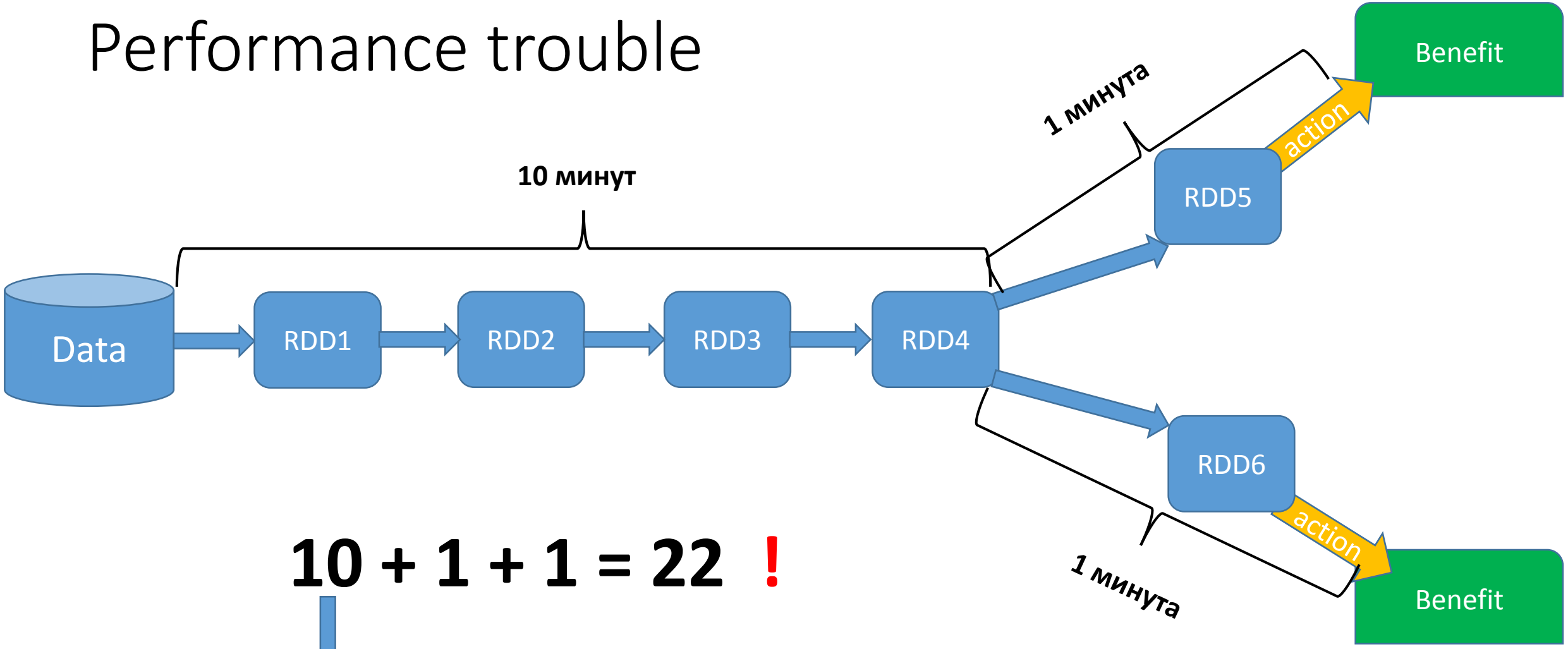
Да да, это всё Lazy как в стримах

Performance trouble



10 + 1 + 1 ≠ 12 ?

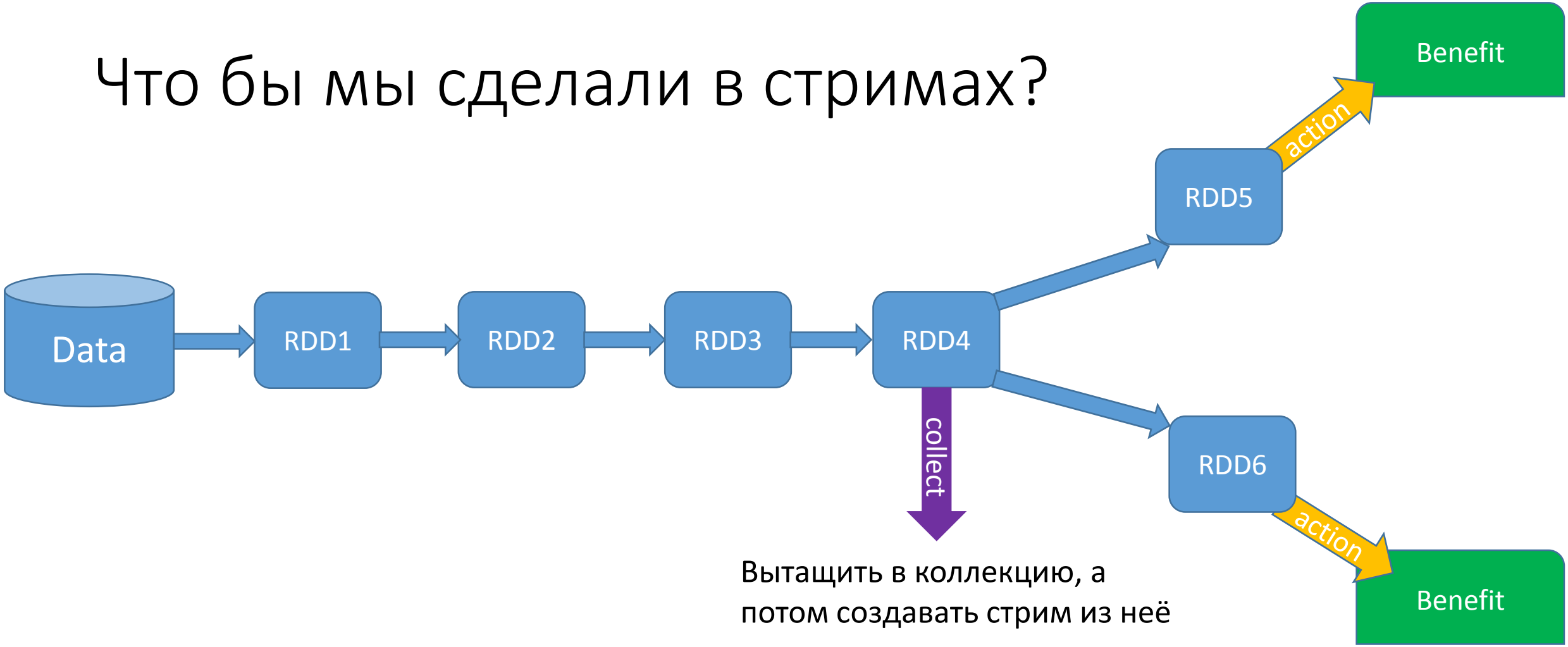
Performance trouble



10 + 1 + 1 = 22 !

(10 x 2)

Что бы мы сделали в стримах?

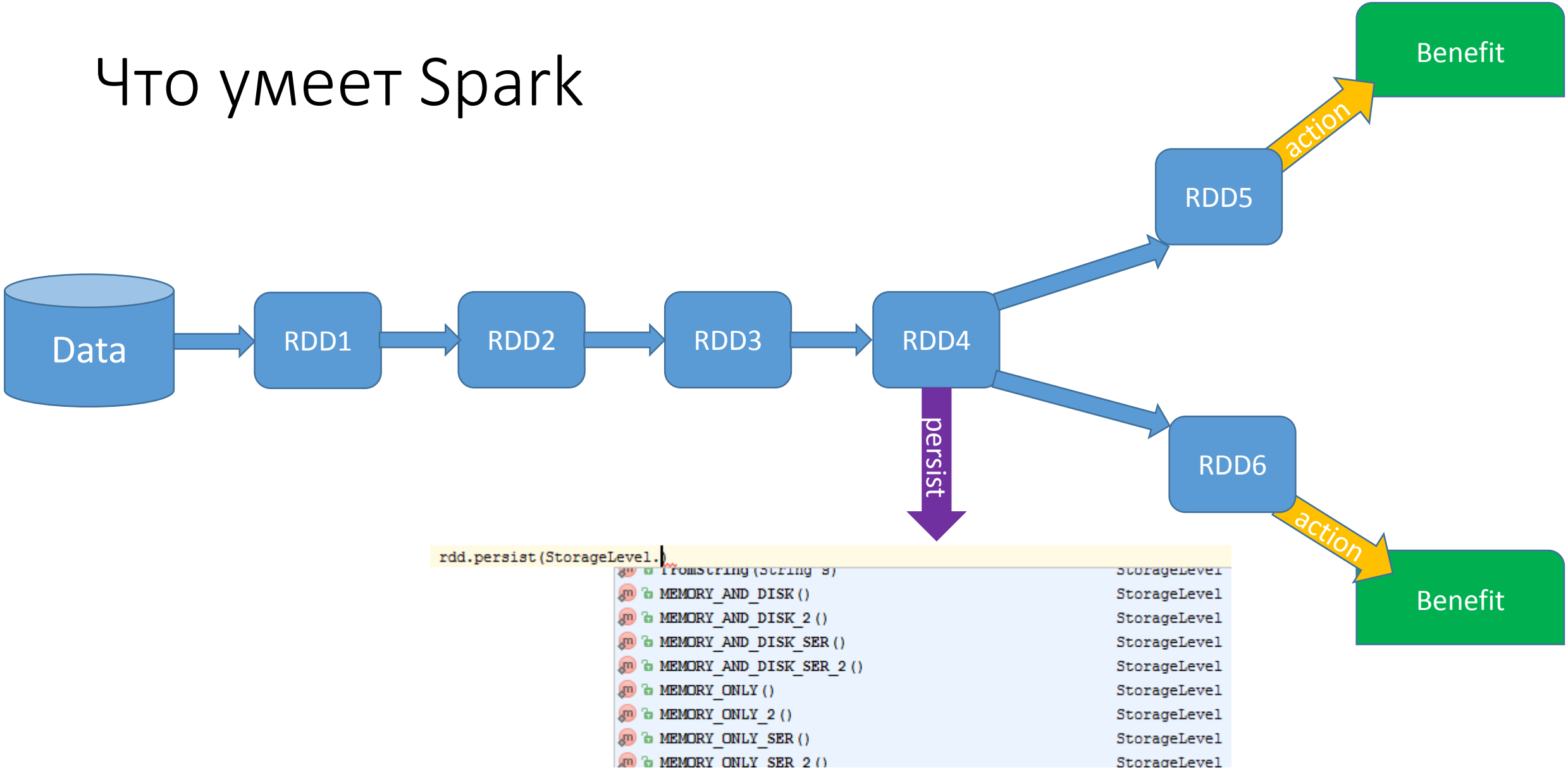


Вытащить в коллекцию, а
потом создавать стрим из неё

тоже так пробовали

Out of memory

Что умеет Spark



На чём будем писать?

- Scala
- Java
- ~~Python~~
- ~~R~~

Что думает о Скале обычный Java developer

What is Scala


Martin Odersky

- 纯面向对象,class\trait\mixin
- 函数式first class,lambda,closure,curry,lazy,tail recursive opt
- Actor, pattern-match
- Jvm bytecode(1.5 compatible)
- 强类型,静态语言,

продвинутый

Что думает о Скале ~~обычный~~ Java developer

What is Scala

Martin Odersky

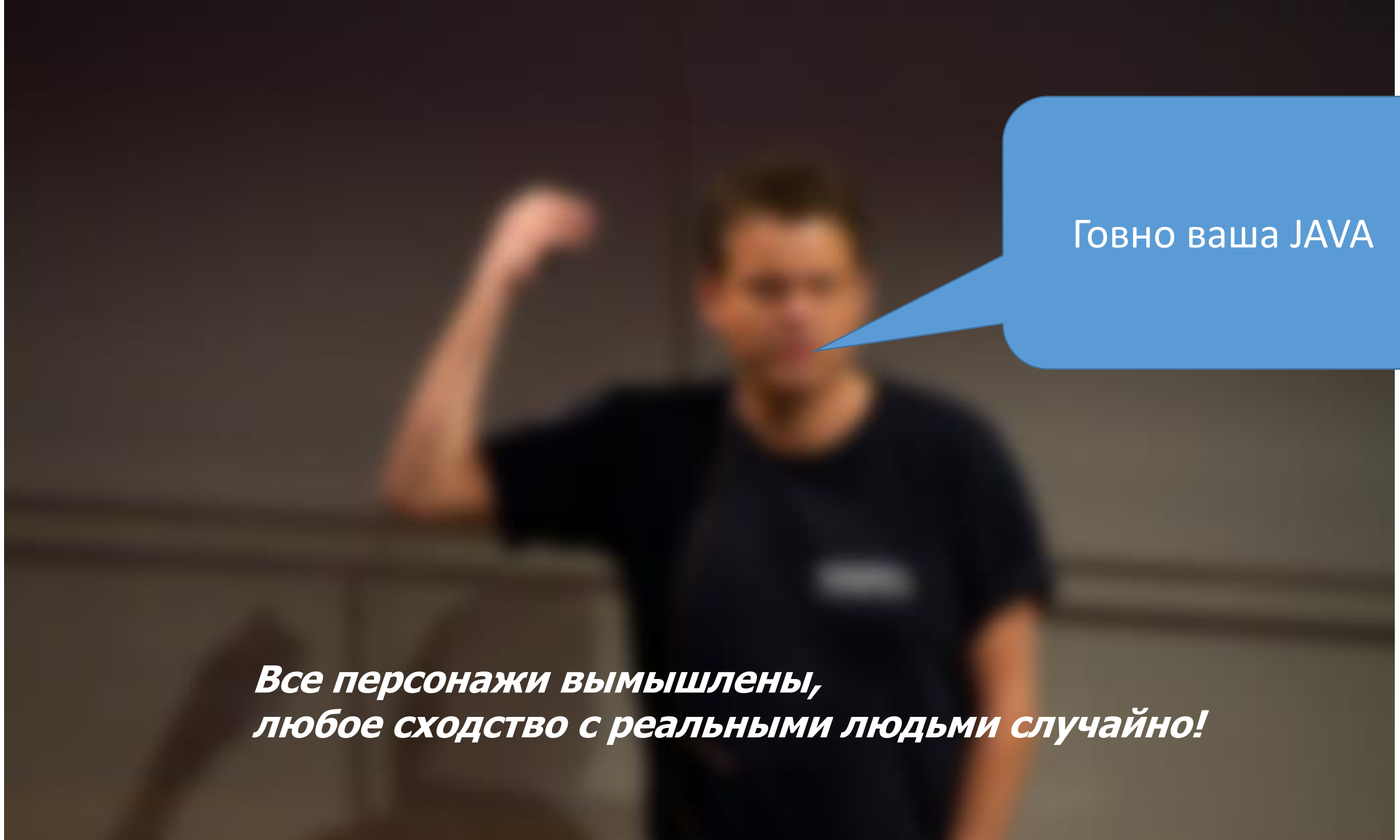
- 纯面向对象,class\trait\mixin
- 函数式first class,lambda,closure,curry,lazy,tail recursive opt
- Actor, pattern-match
- Jvm bytecode(1.5 compatible)
- 强类型,静态语言,

Scala

```scala
val lines = sc.textFile("data.txt")
val lineLengths = lines.map(_.length)
val totalLength = lineLengths.reduce(_+_)
```

Java

```java
JavaRDD<String> lines = sc.textFile("data.txt");
JavaRDD<Integer> lineLengths = lines.map(new Function<String, Integer>() {
    @Override
    public Integer call(String lines) throws Exception {
        return lines.length();
    }
});
Integer totalLength = lineLengths.reduce(new Function2<Integer, Integer, Integer>() {
    @Override
    public Integer call(Integer a, Integer b) throws Exception {
        return a + b;
    }
});
```



Все персонажи вымышлены,
любое сходство с реальными людьми случайно!

```scala
val lines = sc.textFile("data.txt")
val lineLengths = lines.map(_.length)
val totalLength = lineLengths.reduce(_+_)
```

Scala

```java
JavaRDD<String> lines = sc.textFile("data.txt");
JavaRDD<Integer> lineLengths = lines.map(String::length);
int totalLength = lineLengths.reduce((a, b) -> a + b);
```

Java 8

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	Большинство джава программистов знает джаву

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	Большинство джава программистов знает джаву
Более лаконичный и удобный синтаксис	

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	Большинство джава программистов знает джаву
Более лаконичный и удобный синтаксис	Знакомый нам мир (Spring, шаблоны проектирования)

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	Большинство джава программистов знает джаву
Более лаконичный и удобный синтаксис	Знакомый нам мир (Spring, шаблоны проектирования)
Spark API заточен под Скалу в первую очередь	

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	**Большинство джава программистов знает джаву**
Более лаконичный и удобный синтаксис	Знакомый нам мир (Spring, шаблоны проектирования)
Spark API заточен под Скалу в первую очередь	

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	**Большинство джава программистов знает джаву**
Более лаконичный и удобный синтаксис	Знакомый нам мир (Spring, шаблоны проектирования)
Spark API заточен под Скалу в первую очередь	
Java API выходит чуть позже, и не всегда всё есть	

Миф второй – Spark надо писать на Скале

За Скалу	За Джаву
Скала это круто	**Большинство джава программистов знает джаву**
Более лаконичный и удобный синтаксис	Знакомый нам мир (Spring, шаблоны проектирования)
Spark API заточен под Скалу в первую очередь	**Большинство джава программистов знает джаву**
Java API выходит чуть позже, и не всегда всё есть	Знакомый нам мир (Spring, шаблоны проектирования)

Чем будем крыть?

Главная проблема цитат в интернете в том, что люди слепо верят в их подлинность

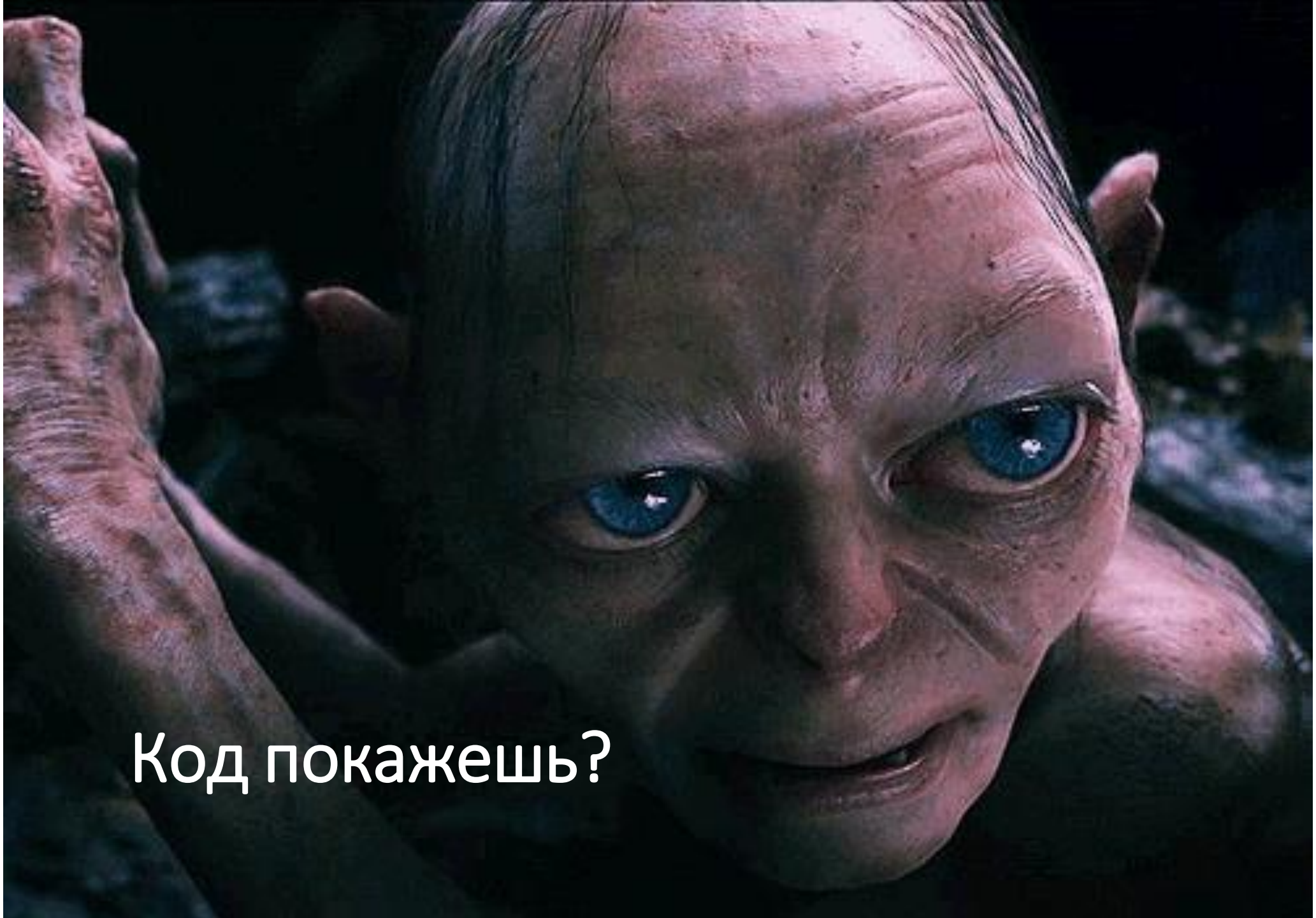
В. И. Ленин

Выписка из последнего API

- The Dataset API, released as an API preview in Spark 1.6, aims to provide the best of both worlds; the familiar object-oriented programming style and compile-time type-safety of the RDD API but with the performance benefits of the Catalyst query optimizer. Datasets also use the same efficient off-heap storage mechanism as the DataFrame API.
- Dataset API is designed to work equally well with both **JAVA** and **SCALA**

Миф третий

- Spark и Spring не совместимы, и вообще там всё по другому


Код покажешь?

Чтобы работать со спарком не обязательнон нужен докер

```
<dependency>
    <groupId>org.apache.spark</groupId>
    <artifactId>spark-core_2.11</artifactId>
    <version>2.3.1</version>
</dependency>
```

Shared data

Пример

```
Israel, +9725423632
Israel, +9725454232
Israel, +9721454232
Israel, +9721454232
Spain, +34441323432
Spain, +34441323432
Israel, +9725423232
Israel, +9725423232
Spain, +34441323432
Russia, +78123343434
Russia, +78123343434
```

Пример

Israel, +9725423632
Israel, +9725454232
Israel, +9721454232
Israel, +9721454232
~~Spain, +34441323432~~
~~Spain, +34441323432~~
Israel, +9725423232
Israel, +9725423232
~~Spain, +34441323432~~
Russia, +78123343434
Russia, +78123343434

Пример

```
Israel, Orange
Israel, Orange
Israel, Pelephone
Israel, Pelephone
Israel, Hot Mobile
Israel, Orange
Russia, Megaphone
Russia, MTC
```

Пример

```
Orange
Orange
Pelephone
Pelephone
Hot Mobile
Orange
Megaphone
MTC
```

Пример

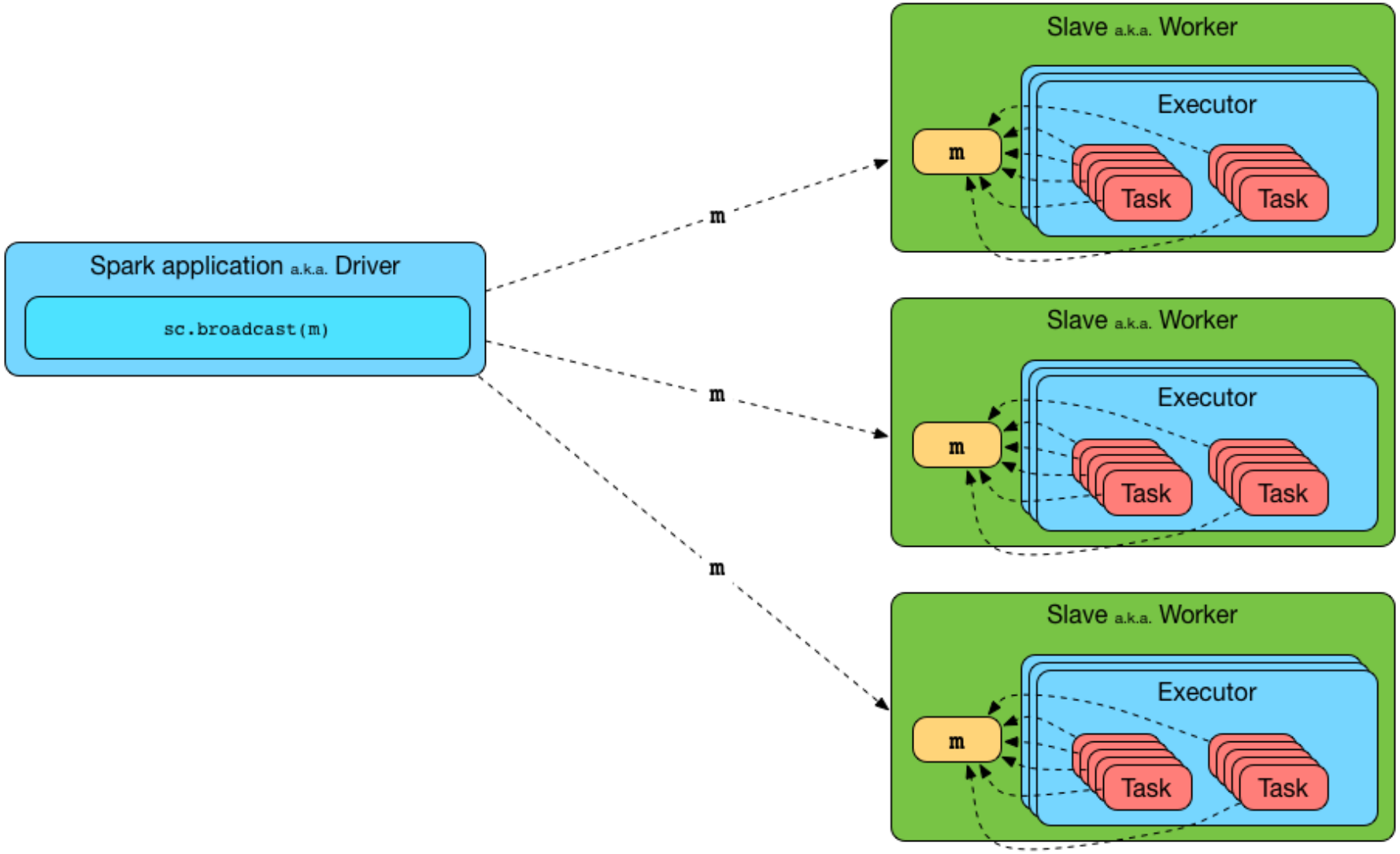
```
Israel, +9725423632
Israel, +9725454232
Israel, +9721454232
Israel, +9721454232
Spain, +34441323432
Spain, +34441323432
Israel, +9725423232
Israel, +9725423232
Spain, +34441323432
Russia, +78123343434
Russia, +78123343434
```

```
public interface CommonConfig {
    Operator getOperator(String phone);

    List<String> countries();
}
```

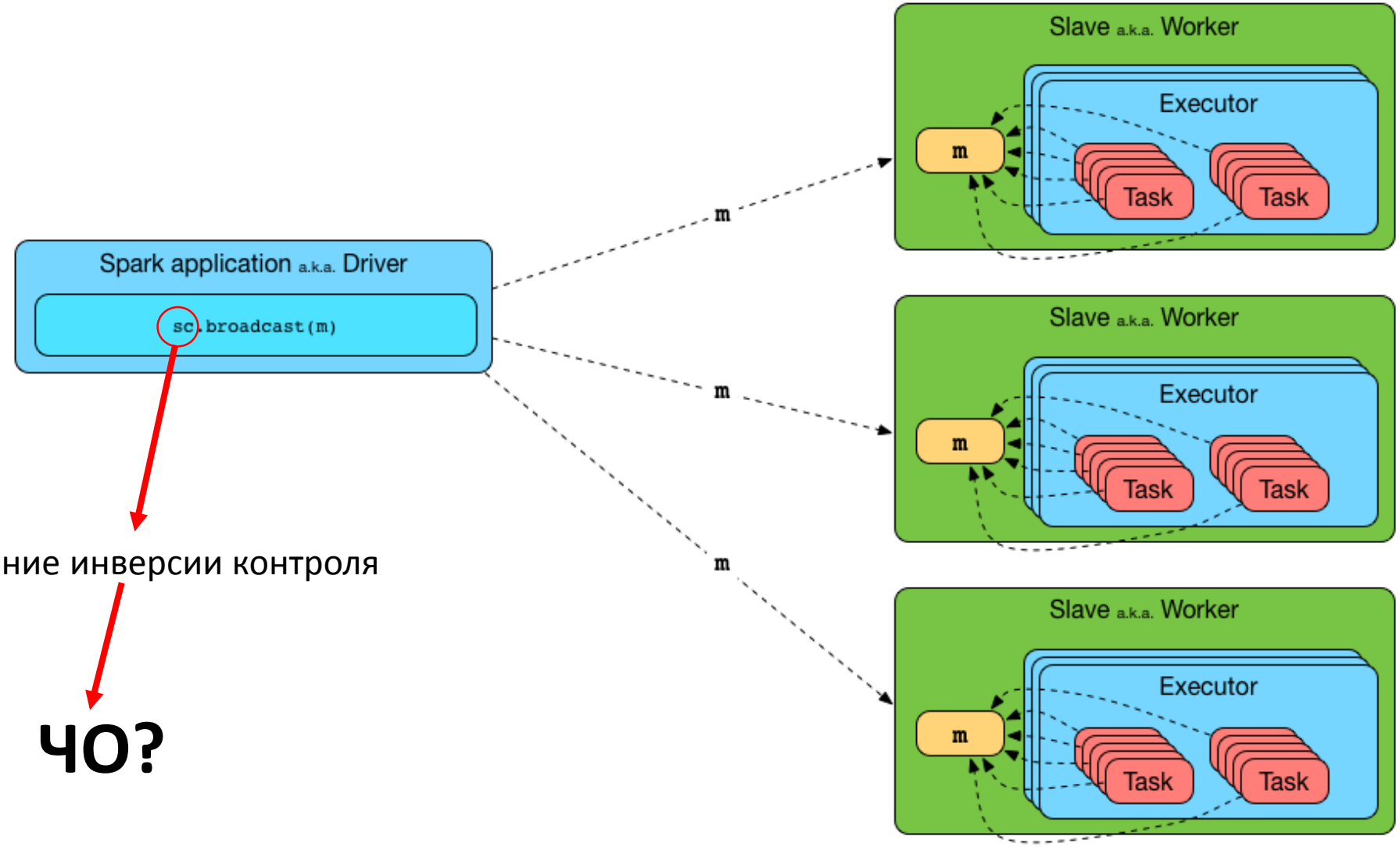
Это не эффективно

```
@Service
public class HrenoviyService {

    @Autowired
    private CommonConfig config;

    public JavaRDD<String> resolveOperatorNames(JavaRDD<Tuple2<String,String>> pairs){

        return
               pairs.filter(pair-> config.countries().contains(pair._1))

                .map(pair-> config.getOperator(pair._2).getName());
    }
}
```

Broadcast


Spark application a.k.a. Driver
sc.broadcast(m)
m
m
m
Slave a.k.a. Worker
Executor
m
Task
Task
Slave a.k.a. Worker
Executor
m
Task
Task
Slave a.k.a. Worker
Executor
m
Task
Task

А в чём проблема?



Нарушение инверсии контроля

ЧО?

```
@Service
public class HrenoviyService {

    @Autowired
    private JavaSparkContext sc;

    @Autowired
    private CommonConfig commonConfig;

    private Broadcast<CommonConfig> configBroadcast;

    @PostConstruct
    public void wrapWithBroadCast(){
        configBroadcast = sc.broadcast(commonConfig);
    }

    public JavaRDD<String> resolveOperatorNames(JavaRDD<Tuple2<String,String>> pairs){

        return pairs.filter(pair-> configBroadcast.value().countries().contains(pair._1))

                .map(pair-> configBroadcast.value().getOperator(pair._2).getName());
    }
}
```

И в чём проблема?

Нарушение инверсии контроля

Ну так себе нарушение, не контекст же Спринга инжектим

Много копи паста

Не так уж и много, пара строчек

```
@Autowired
private JavaSparkContext sc;

@Autowired
private CommonConfig commonConfig;

private Broadcast<CommonConfig> configBroadcast;

@PostConstruct
public void wrapWithBroadCast(){
    configBroadcast = sc.broadcast(commonConfig);
}
```

}

Технический спарковый код в бизнес логике

А он у тебя всё равно где то будет в бизнес логике

Но не в каждом же сервисе, которому нужен броадкаст


Не убедил

...

Дополнительные сложности для тестов

Они и так не 100% unit тесты

Есть ещё аргумент


Убедил

Caused by: java.io.NotSerializableException: com.inwhite.HrenoviyService

Serialization stack:

- object not serializable (class: com.inwhite.HrenoviyService, value: com.inwhite.HrenoviyService@4fa4f485)

- element of array (index: 0)

- array (class [Ljava.lang.Object;, size 1)

- field (class: java.lang.invoke.SerializedLambda, name: capturedArgs, type: class [Ljava.lang.Object;)

- object (class java.lang.invoke.SerializedLambda, SerializedLambda[capturingClass=class com.inwhite.HrenoviyService, functionalInterfaceMethod=org/apache/spark/api/java/function/Function.call:(Ljava/lang/Object;)Ljava/lang/Object;, implementation=invokeSpecial com/inwhite/PopularWordsResolver.lambda$mostUsedWords$29bd749d$1:(Ljava/lang/String;)Ljava/lang/Boolean;, instantiatedMethodType=(Ljava/lang/String;)Ljava/lang/Boolean;, numCaptured=1])

- writeReplace data (class: java.lang.invoke.SerializedLambda)

- object (class com.inwhite.HrenoviyService$$Lambda$6/1775488894, com.inwhite. com.inwhite.HrenoviyService $$Lambda$6/1775488894@5ec1963c)

- field (class: org.apache.spark.api.java.JavaRDD$$anonfun$filter$1, name: f$1, type: interface org.apache.spark.api.java.function.Function)

- object (class org.apache.spark.api.java.JavaRDD$$anonfun$filter$1, <function1>)

at org.apache.spark.serializer.SerializationDebugger$.improveException(SerializationDebugger.scala:40)

at org.apache.spark.serializer.JavaSerializationStream.writeObject(JavaSerializer.scala:47)

at org.apache.spark.serializer.JavaSerializerInstance.serialize(JavaSerializer.scala:101)

at org.apache.spark.util.ClosureCleaner$.ensureSerializable(ClosureCleaner.scala:301)

... 44 more


```
@Service
public class HrenoviyService {

    @Autowired
    private CommonConfig commonConfig;

    public JavaRDD<String> resolveOperatorNames(JavaRDD<Tuple2<String,String>> pairs){
        return pairs.filter(pair-> commonConfig.countries().contains(pair._1))
                .map(pair->commonConfig.getOperator(pair._2).getName());
    }
}
```


```
//@Service
public class HrenoviyService {

//     @Autowired
    private Broadcast<CommonConfig> commonConfig;

    public void setCommonConfig(Broadcast<CommonConfig> commonConfig) {
        this.commonConfig = commonConfig;
    }

    public JavaRDD<String> resolveOperatorNames(JavaRDD<Tuple2<String,String>> pairs){
        return pairs.filter(pair-> commonConfig.value().countries().contains(pair._1))
                .map(pair->commonConfig.value().getOperator(pair._2).getName());
    }
}
```


```
@Configuration
@ComponentScan(basePackages = "techtrain")
public class SpringConfig {
    @Autowired
    private CommonConfig commonConfig;

    @Bean
    public HrenoviyService hrenoviyService(){
        HrenoviyService hrenoviyService = new HrenoviyService();
        hrenoviyService.setCommonConfig(sc().broadcast(commonConfig));
        return hrenoviyService;
    }

    @Bean
    public JavaSparkContext sc() {
        SparkConf conf = new SparkConf();
        conf.setAppName("songs");
        conf.setMaster("local[*]");
        return new JavaSparkContext(conf);
    }
}
```

COME WITH ME IF YOU WANT TO LIVE

Ты сейчас опять будешь рассказывать про BeanPostProcessor-ы?

Ты сейчас опять будешь рассказывать про BeanPostProcessor-ы?

Сравнение синтаксиса

Java

```
lines.map(String::toLowerCase)
 .flatMap(WordsUtil::getWords)
 .filter(word-> !Arrays.asList(garbage).contains(word))
 .mapToPair(word-> new Tuple2<>(word, 1))
 .reduceByKey((x, y)->x+y)
 .mapToPair(Tuple2::swap)
 .sortByKey(false)
 .map(Tuple2::_2)
 .take(amount);
```

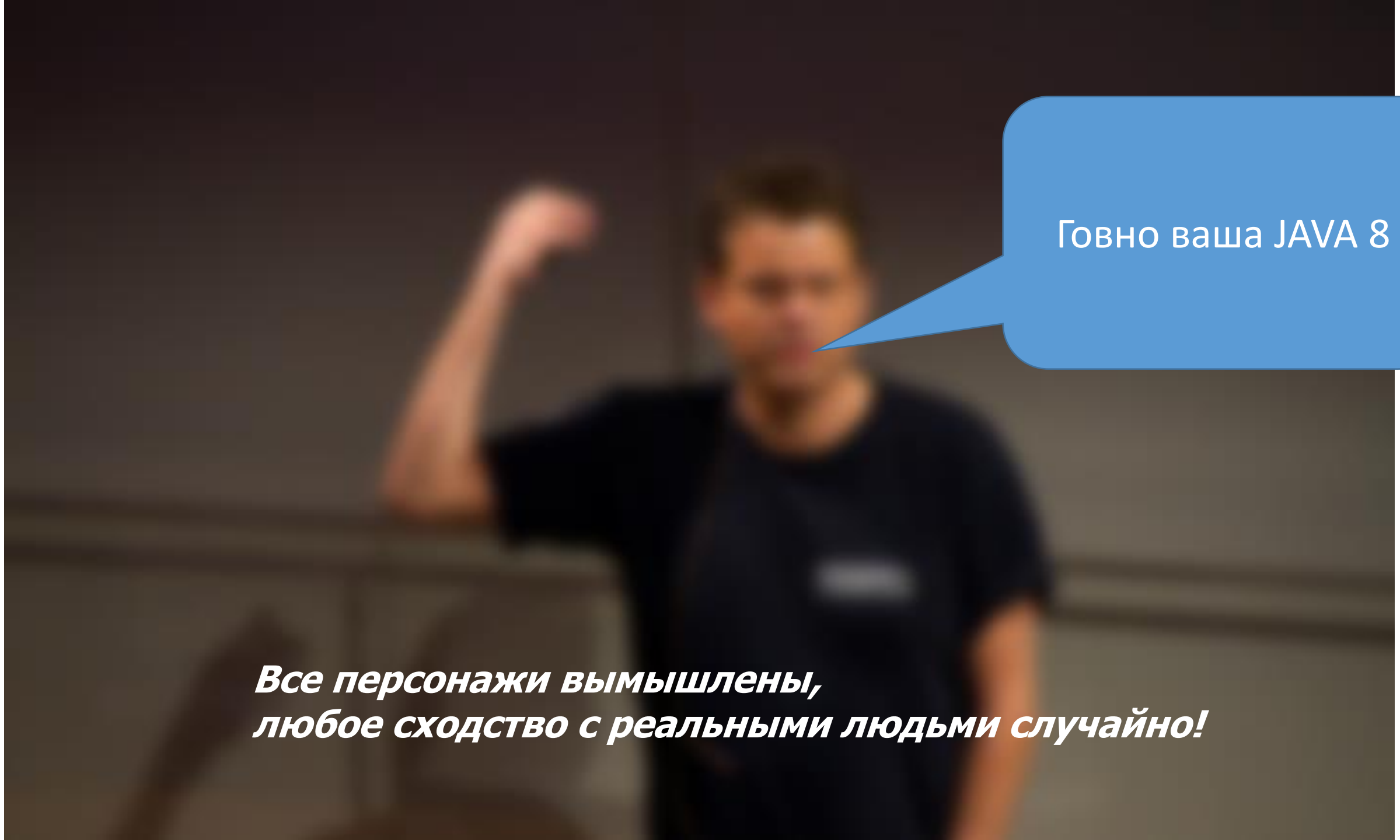
Сравнение синтаксиса

Java

```
lines.map(String::toLowerCase)
 .flatMap(WordsUtil::getWords)
 .filter(word-> !Arrays.asList(garbage).contains(word))
 .mapToPair(word-> new Tuple2<>(word, 1))
 .reduceByKey((x, y)->x+y)
 .mapToPair(Tuple2::swap)
 .sortByKey(false)
 .map(Tuple2::_2)
 .take(amount);
```

Scala

```
lines.map(_.toLowerCase())
  .flatMap("\\w+".r.findAllIn(_))
  .filter(!garbage.contains(_))
  .map((_,1)).reduceByKey(_+_)
  .sortBy(_._2,ascending = false)
  .take(amount)
```



Все персонажи вымышлены,
любое сходство с реальными людьми случайно!


ИВАН ВАСИЛЬЕВИЧ
ВПАДАЕТ В ДЕПРЕССИЮ

Чем будем крыть?


чёт мне кажется
это уже было

Чем будем крыть?

DataFrames

Co Spark-a 2 Datasets

Dataframes – since 1.3

- DataFrame is a distributed collection of data organized into named columns
- Под капотом RDD, но там много оптимизаций
- Датафрэймы требуют меньше памяти, если их умно использовать
- Можно создать из hive таблиц, json подобных файлов, обычных RDD, внешних баз данных, и из всего имеющего структуру
- Имеет очень широкий DSL
- Связан с SqlContext

Dataframes methods and functions

Agg, columns, count, distinct, drop, dropDuplicates, filter

groupBy, orderBy, registerTable, schema, show, select, where, withColumn

Немного методов Датафрэймов

Agg, columns, count, distinct, drop, dropDuplicates, filter

groupBy, orderBy, registerTable, schema, show, select, where, withColumn

```
dataFrame.registerTempTable("persons");

DataFrame frame = sqlContext.sql("select cl_id, cl_grp_id, dk_org_snw, dk_org_hnw,
         dk_org_cnp, dk_dir, dk_dat, DK_TIM_HR as dk_tim_hr, dk_spe, dk_sgt, dk_pet, dk_sgs, dk_sbp,\n" +
        "SUM(slu_atpt) slu_atpt, SUM(slu_succ) slu_succ, SUM(slu_fail) slu_fail, SUM(slu_dly) slu_dly\n" +
        "FROM persons f join tdtim t on f.dk_tim = t.DK_TIM\n" +
        "WHERE dk_pet IN (1, 4)\n" +
        "group by cl_id, cl_grp_id, dk_org_snw, dk_org_hnw, dk_org_cnp, dk_dir, dk_dat, DK_TIM_HR,
        dk_spe, dk_sgt, dk_pet, dk_sgs, dk_sbp").toDF();
```

Не говорите ему про SqlContext

Почему я не люблю sql нигде

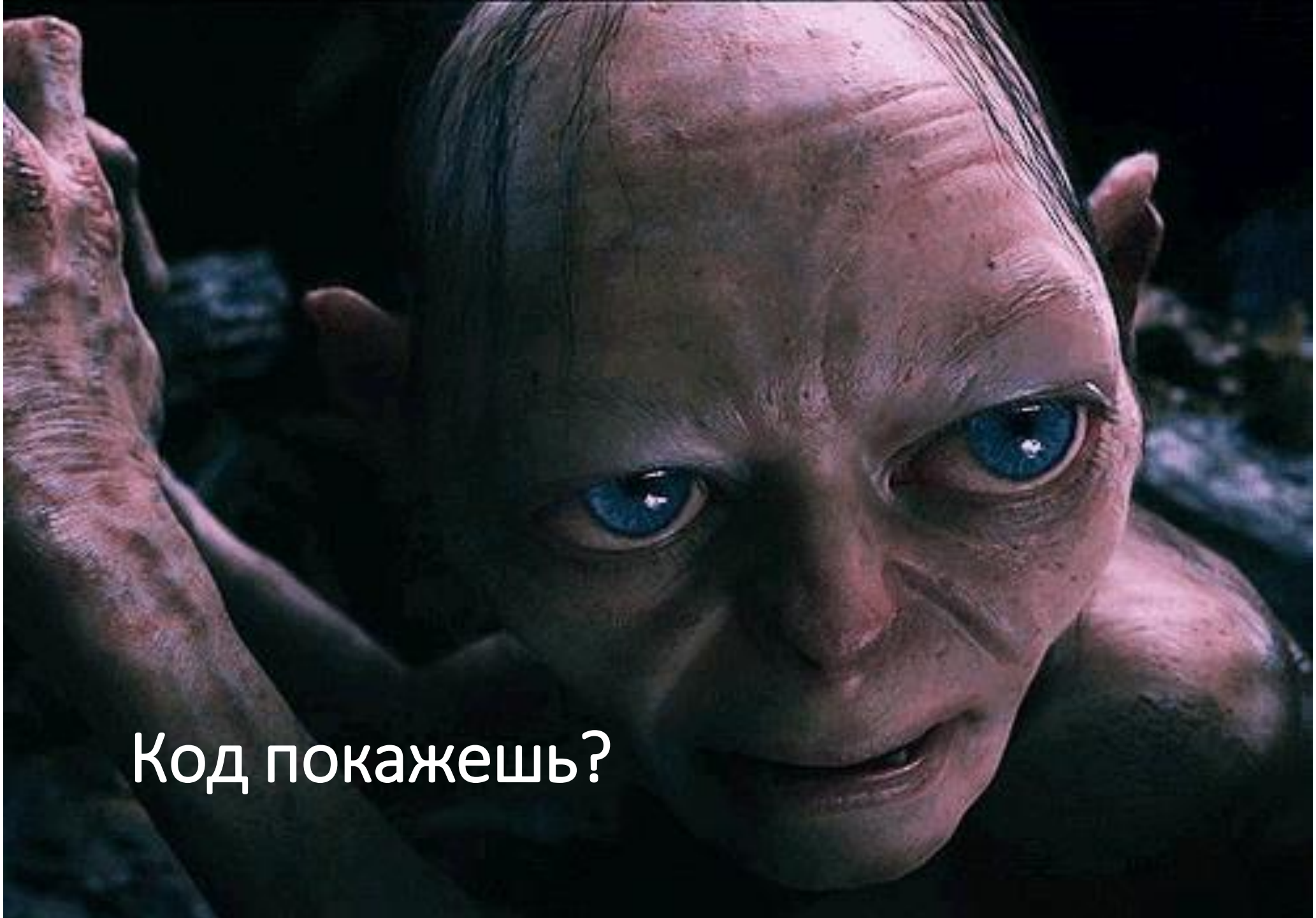
```
DataFrame dataFrame1 = sqlContext.sql("select  tdmco.CL_ID as cl_id, \n" +
        "\t\ttmdcl.CL_GRP_ID as cl_grp_id, \n" +
        "\t\ttdpet.DK_PET as dk_pet, \n" +
        "\t\ttdsgt.DK_SGT as dk_sgt, \n" +
        "\t\ttdsgs.DK_SGS as dk_sgs, \n" +
        "\t\ttddir.DK_DIR as dk_dir, \n" +
        "\t\ttdorg.DK_ORG as dk_org_cnp,\n" +
        "\t\tcase when data.INBOUND_IND = 0 then tdorg.DK_ORG when data.INBOUND_IND = 1 then tdmco.CL_ID end as dk_org_hnw,\n" +
        "\t\tcase when data.INBOUND_IND = 1 then tdorg.DK_ORG when data.INBOUND_IND = 0 then tdmco.CL_ID end as dk_org_snw,\n" +
        "\t\ttddat.DK_DAT as dk_dat,\n" +
        "\t\ttdtim.DK_TIM as dk_tim,\n" +
        "\t\tcast(case when spe1.DK_SPE is not null then spe1.DK_SPE when spe2.DK_SPE is not null then spe2.DK_SPE when spe3.DK_SPE is not null then spe3.DK_SPE else 0 end as bi
        "\t\tcast(-999 as bigint) as dk_sgl, \n" +
        "\t\tcast(-999 as bigint) as dk_adt,\n" +
        "\t\tcast(0 as bigint) as dk_sbp,\n" +
        "\t\tdata.IMSI as dd_imsi,\n" +
        "\t\tMSISDN as dd_msisdn_min, \n" +
        "\t\tMSC as dd_msc, \n" +
        "\t\tHLR as dd_hlr, \n" +
        "\t\tSGSN_GT as dd_sgsn_gt, \n" +
        "\t\tSGSN_IP as dd_sgsn_ip,\n" +
        "\t\tcast(1 as bigint) as slu_atpt,\n" +
        "\t\tcast(case when data.STATUS = 'OK' then 1 else 0 end as bigint) as slu_succ,\n" +
        "\t\tcast(case when data.STATUS = 'Reject' then 1 else 0 end as bigint) as slu_fail,\n" +
        "\t\tcast(((unix_timestamp(substring(data.TIME_END,1,19))*1000)+(substring(data.TIME_END,21,3)))-((unix_timestamp(substring(data.TIME,1,19))*1000)+(substring(data.TIME,2
        "\t\tdata.STATUS as status\n" +
        "\t\t-- first_value(tmdsbsprf.PRF_ID) OVER (PARTITION BY tdmco.CL_ID,data.IMSI ORDER BY length(tmdsbsprf.MATCH_STR) desc,tmdsbsprf.MATCH_START) AS sbs_prf_if,\n" +
        "from data left join tdmco on (data.OWNER_CODE=tdmco.MCO_TAP_CD)\n" +
        "left join tmdcl on (tdmco.CL_ID=tmdcl.CL_ID)\n" +
        "left join tdpet on (data.OPCODE=tdpet.PET_CD)\n" +
        "left join tdsgt on (case when data.OPCODE = 2 then 'C' when data.OPCODE = 23 then 'P' end = tdsgt.SGT_CD)\n" +
        "left join tdsgs on (case when data.STATUS = 'OK' then 'S' when data.STATUS = 'Reject' then 'F' end = tdsgs.SGS_CD)\n" +
        "left join tddir on (case when data.INBOUND_IND = 0 then 'V' when data.INBOUND_IND = 1 then 'O' end = tddir.DIR_CD)\n" +
        "left join tdorg on (data.OP_ID = tdorg.ORG_TAP_CD)\n" +
        "left join tddat on (to_date(data.TIME) = to_date(tddat.DAT))\n" +
        "left join tdtim on (substring(data.TIME,12,8) = tdtim.TIM)\n" +
        "left join tdspe as spe1 on (case when data.STATUS = 'Reject' and data.MAP_ERR is null and TCAP_ERR is null then 255 end = spe1.SPE_CD and spe1.SPE_CAT = 'MAP')\n" +
        "left join tdspe as spe2 on (case when data.TCAP_ERR is null or (data.MAP_ERR is not null and data.TCAP_ERR is not null) then data.MAP_ERR end = spe2.SPE_CD and spe2.SPE_
        "left join tdspe as spe3 on (case when data.MAP_ERR is null and data.TCAP_ERR is not null then data.TCAP_ERR end = spe3.SPE_CD and spe3.SPE_CAT = 'TCAP')\n" +
        "-- left join tmdsbsprf on (tdmco.CL_ID = tmdsbsprf.org_tech_id and SUBSTR(data.IMSI, tmdsbsprf.MATCH_START, LENGTH(tmdsbsprf.MATCH_STR)) = tmdsbsprf.MATCH_STR and to_da
```

Dataframes methods and functions

- abs, cos, asin, isnull, not, rand, sqrt, when, expr, bin, atan, ceil, floor, factorial, greatest, least, log, log10, pow, round, sin, toDegrees, toRadians, md5, ascii, base64, concat, length, lower, ltrim, unbase64, repeat, reverse, split, substring, trim, upper, datediff, year, month, hour, last_day, next_day, dayofmonth, explode, udf

Dataframes methods and functions

- abs, cos, asin, isnull, not, rand, sqrt, when, expr, bin, atan, ceil, floor, factorial, greatest, least, log, log10, pow, round, sin, toDegrees, toRadians, md5, ascii, base64, concat, length, lower, ltrim, unbase64, repeat, reverse, split, substring, trim, upper, datediff, year, month, hour, last_day, next_day, dayofmonth, explode, udf


Код покажешь?

Миф четвёртый

- Датафрэймы можно использовать только для файлов со схемой

```
public DataFrame load() {
    JavaRDD<String> rdd = sc.textFile("data/transactions.csv");
    JavaRDD<Row> rowJavaRDD = rdd.map(line -> {
        String[] data = line.split(";");
        return RowFactory.create(Integer.parseInt(data[0]),
                Integer.parseInt(data[1]), Integer.parseInt(data[2]));
    });

    return sqlContext.createDataFrame(rowJavaRDD,createSchema());
}

private static StructType createSchema() {
 return DataTypes.createStructType(new StructField[]{
        DataTypes.createStructField("code",DataTypes.IntegerType,false),
        DataTypes.createStructField("amount",DataTypes.IntegerType,false),
        DataTypes.createStructField("account",DataTypes.IntegerType,false)
    });
}
```

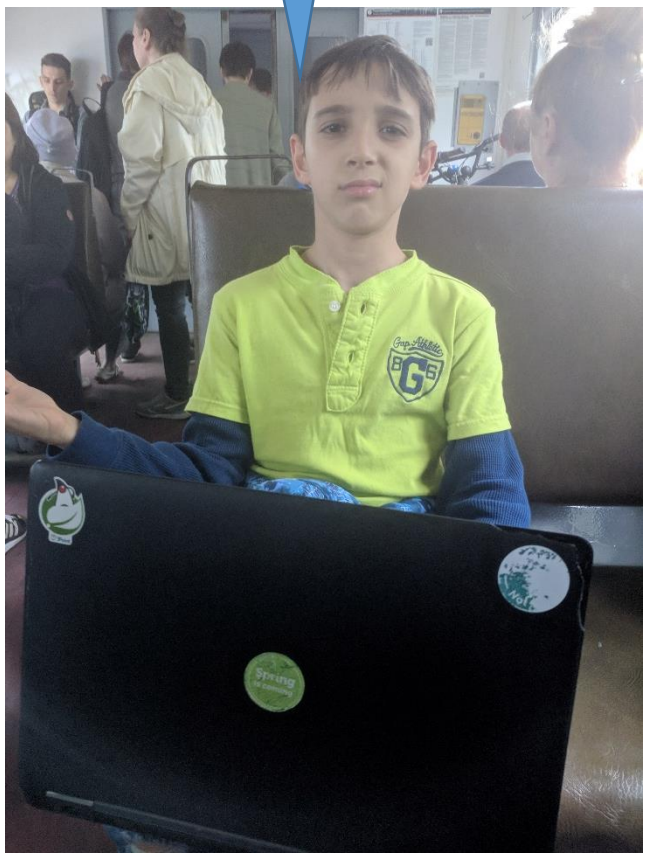
Функций много, но если надо свою?

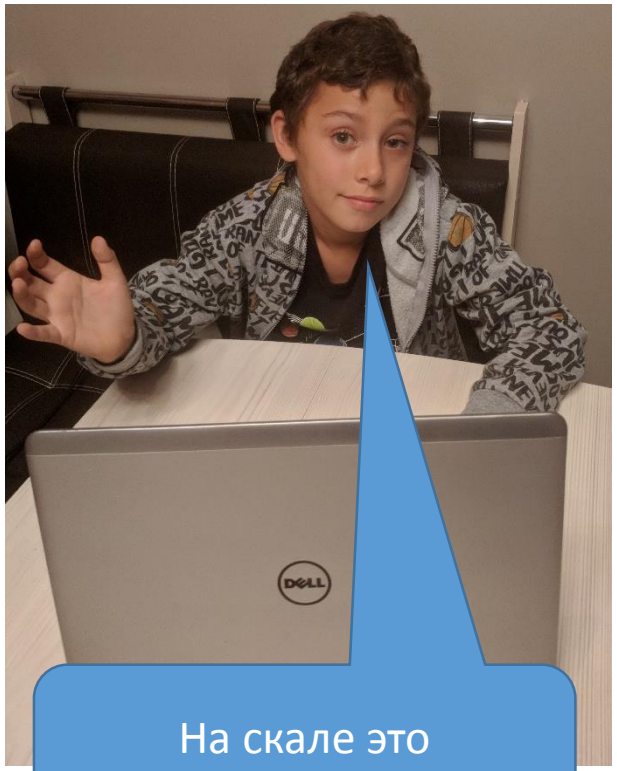
```
@Component
public class GarbageFilter implements UDF1<String,Boolean> {
    @AutowireBroadcast
    private Broadcast<UserConfig> userConfig;

    public String udfName(){return "notGarbage";}

    @Override
    public Boolean call(String word) throws Exception {
        return ! userConfig.value().garbage.contains(word);
    }
}
```

Как пользоваться UDF функциями?

```
@Service
public class PopularWordsResolverWithUDF {

    @Autowired
    private GarbageFilter garbageFilter;

    @Autowired
    private SQLContext sqlContext;

    @PostConstruct
    public void registerUdf(){
        sqlContext.udf().register(garbageFilter.udfName(),garbageFilter, DataTypes.BooleanType);
    }

    public List<String> mostUsedWords(DataFrame dataFrame, int amount) {
        DataFrame sorted = dataFrame.withColumn("words", lower(column("words")))
                                    .filter(callUDF(garbageFilter.udfName(),column("words")))…
```

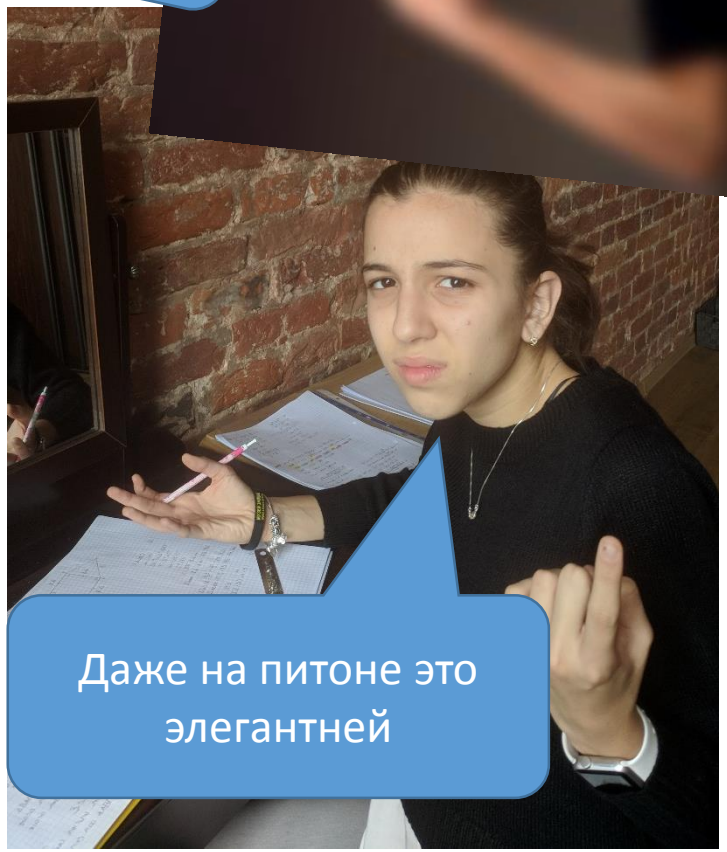
Как то это сложно...





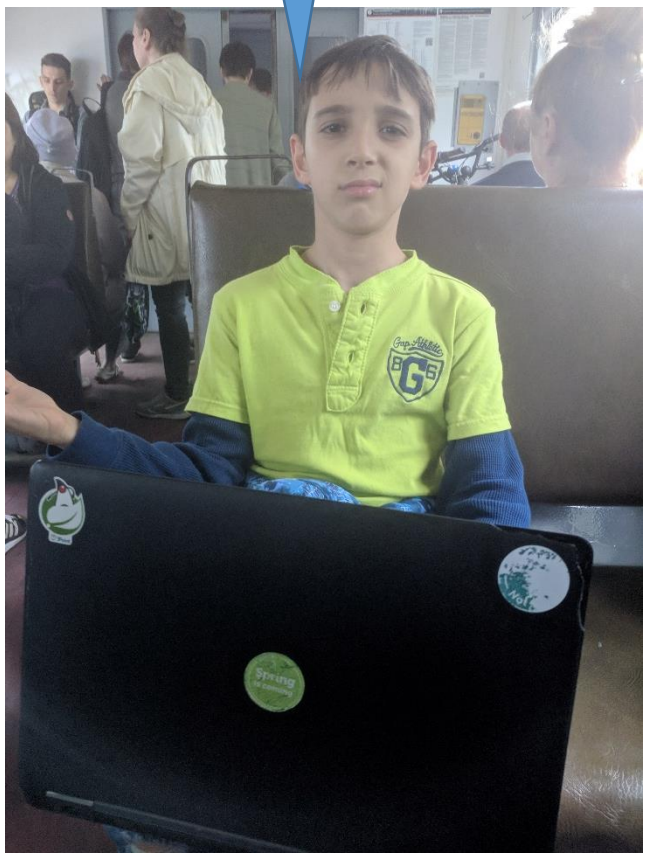
На скале это красивее...

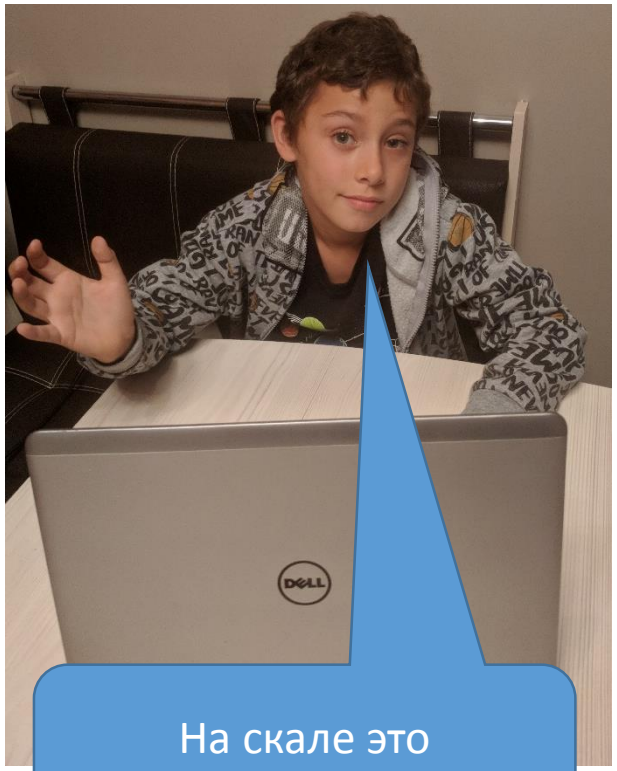
Говно ваша Джава



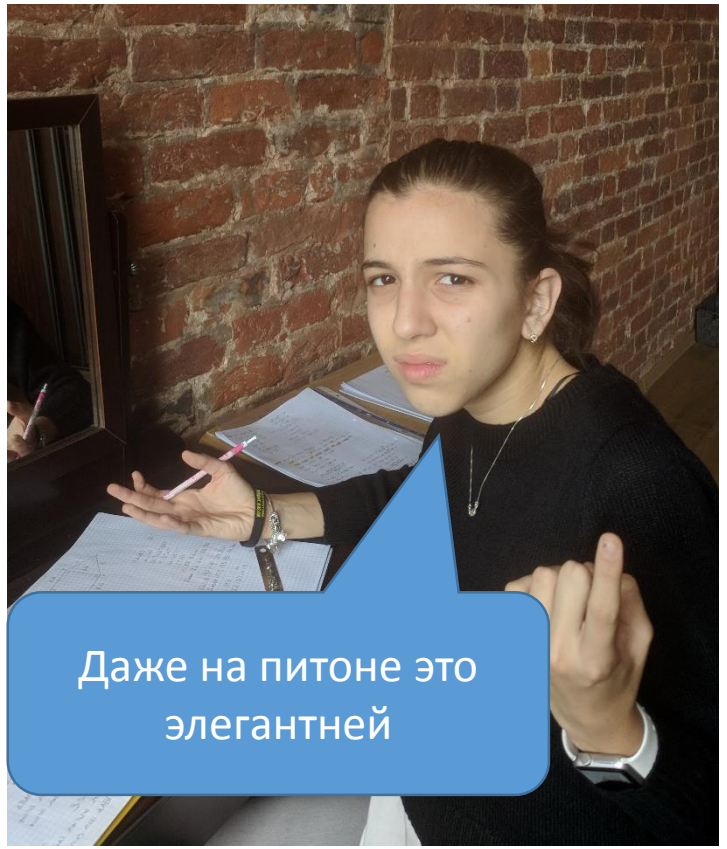
Даже на питоне это элегантней

Как то это сложно…





На скале это красивее…



Даже на питоне это элегантней

А если со спрингом?

И всё

```
@RegisterUDF
public class CountryCodeConverter implements UDF1<Integer,String>,
 Serializable {

    @AutowiredBroadcast
    private Broadcast<UserConfig> broadcast;

    @Override
    public String call(Integer countryCode) throws Exception {
        return broadcast.value().getMap().get(countryCode);
    }
}
```

А если перейти на второй спарк?

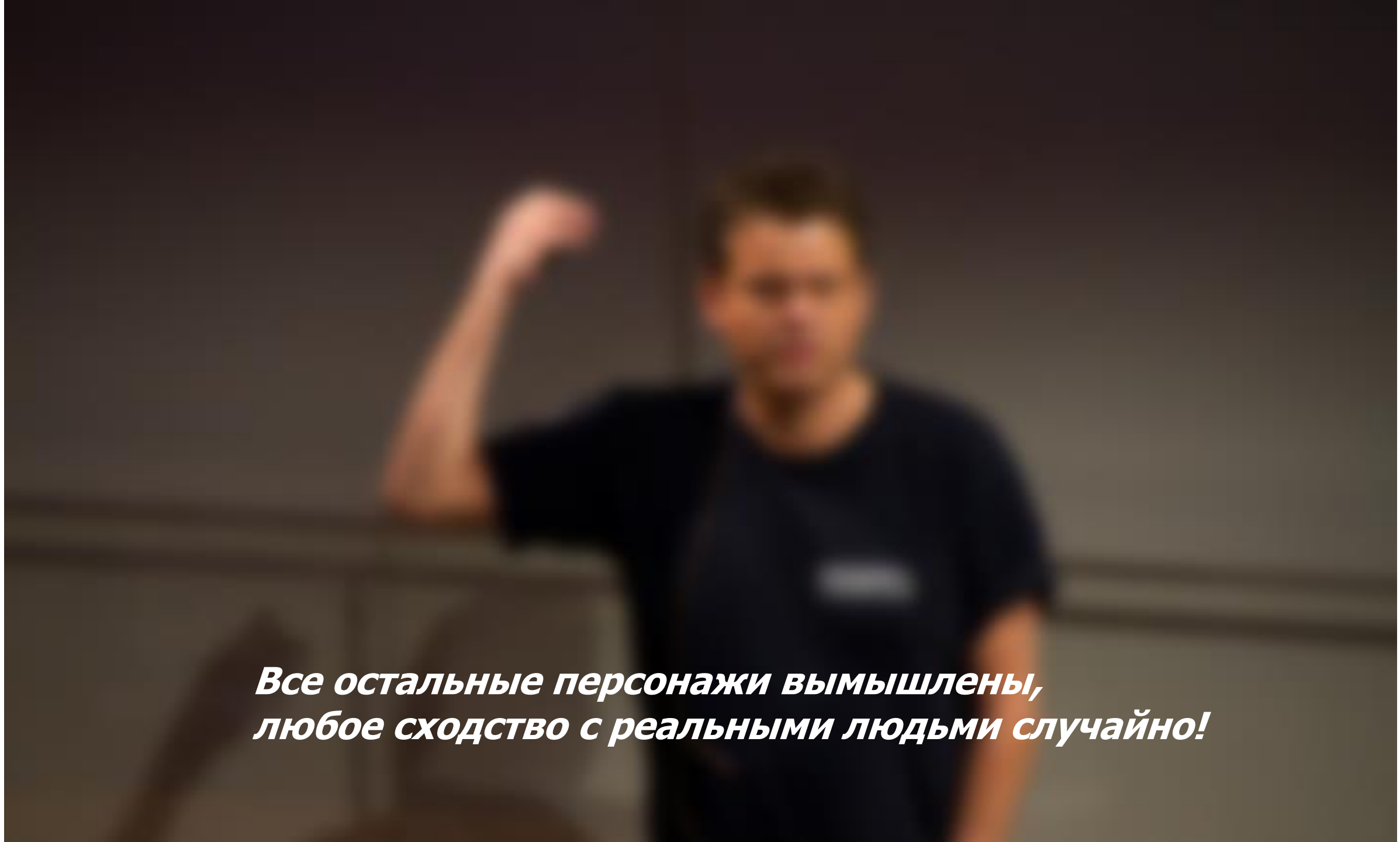
Итоги и выводы

- Hadoop нуждается в Спарке, а не наоборот
- Можно отлично обойтись без Скалы
- Можно использовать привычный вам подход:
  - Инверсия контроля, Spring, Шаблоны проектирования
- Можно прекрасно писать тесты
- Можно везде (~~ну почти~~) пользоваться Датафрэймами
- И главное:


Pink Floyd не попса!!!

Спасибо моей дочке за предоставленные тексты песен

С днём рождения Элинор!!!


Все остальные персонажи вымышлены,
любое сходство с реальными людьми случайно!