Java Mapping для прагматичных программистов

Vladimir.Krasilschik@gmail.com
VKrasilschik@luxoft.com

Что Вы узнаете за следующие 45 минут?

LUXOFT
Engineering Business Performance

- Что такое Java mapping?
- Немного про проект “X”
- Проблемы маппинга
- Критерии “правильного” маппинга
- Обзор технологий маппинга
- ИТОГИ

Поехали? =)

“Java mapping” - о чем вообще речь?

LUXOFT
Engineering Business Performance

Классы с именами типа:

- Mapper
- Convertor || Converter
- Transformer
- Builder, Assembler (редко)

“Java mapping” - о чем вообще речь?

LUXOFT
Engineering Business Performance

Классы с методами типа:

- map
- convert
- transform
- build, enrich

- В смысле речь о формализмах типа ORM или компиляторов?

*- Нет, это же презентация для **прагматичных** программистов*

Какие Java маппинги рассматриваем?

LUXOFT
Engineering Business Performance

1. Простые: long <-> String
2. Между структурами данных: Array <-> Collection
3. Между архитектурными слоями: Entity <-> DTO
4. Кросс-доменные:

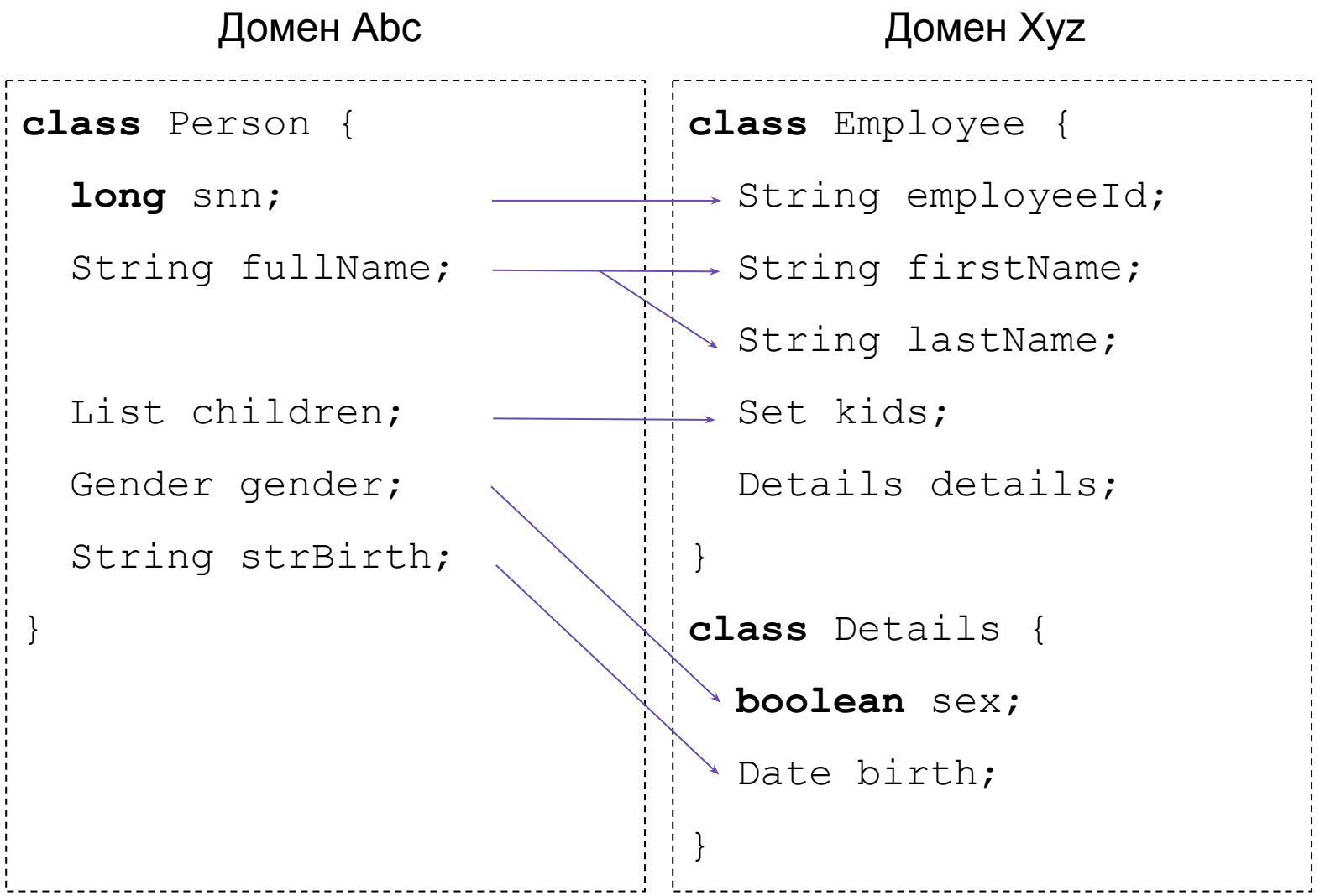
    JavaBean <-> JavaBean

    XML <-> XML

Есть API не удовлетворяющие JavaBean спецификации, как быть с ними?

Кросс-доменный Java маппинг - что?

LUXOFT
Engineering Business Performance



Какие Java маппинги **НЕ** рассматриваем?

LUXOFT
Engineering Business Performance

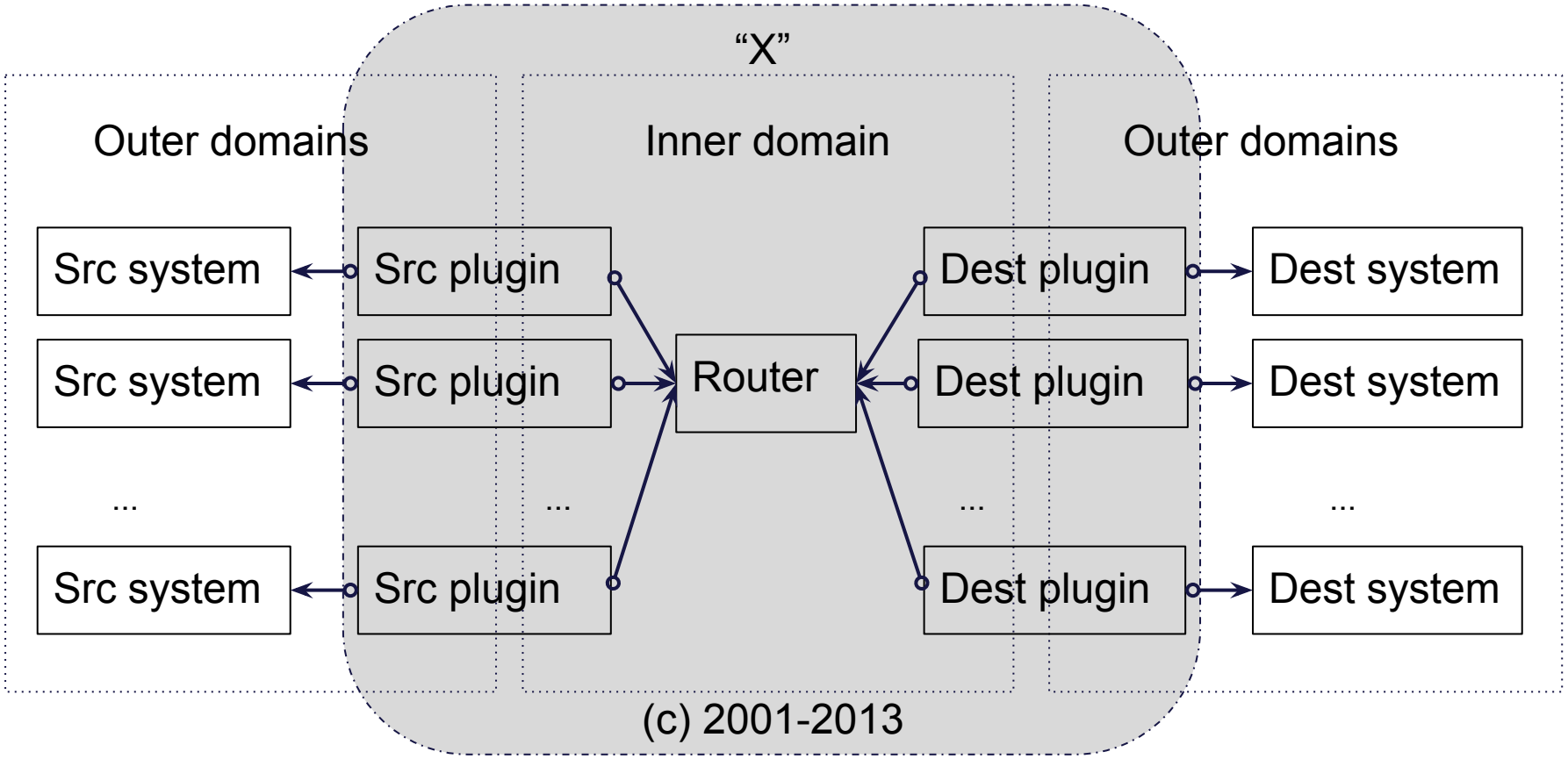
1. JavaBean <-> XML: JAXB, XmlBeans, XStream
2. DB <-> Java: Hibernate, JPA, GORM

Какие Java маппинги **НЕ** рассматриваем?

LUXOFT
Engineering Business Performance

1. ~~JavaBean <-> XML: JAXB, XmlBeans, XStream~~
2. ~~DB <-> Java: Hibernate, JPA, GORM~~

Проект “X” - интеграционная платформа, вид с высоты 1000 футов



Основные компоненты: router, плагины-источники, целевые плагины и внешние системы

Проект “X” - некоторые аспекты архитектуры

LUXOFT
Engineering Business Performance

- около 15 плагинов
- более 30 кросс-доменных маппингов

4 различных технологий маппинга:

1. XSLT: более 15
2. Java: около 10
3. DOM: 1
4. Nomin: 1

Проект “X” - сколько стоит исправить маппинг из домена Abc в домен Xyz

LUXOFT
Engineering Business Performance

Проект “X” - сколько стоит исправить маппинг из домена Abc в домен Xyz

- анализ требований и поиск плагина: 1 час

LUXOFT
Engineering Business Performance

Проект “X” - сколько стоит исправить маппинг из домена Abc в домен Xyz

- анализ требований и поиск плагина: 1 час
- поиск кода маппинга в плагине: 1 час

LUXOFT
Engineering Business Performance

Проект “X” - сколько стоит исправить маппинг из домена Abc в домен Xyz

LUXOFT
Engineering Business Performance

- анализ требований и поиск плагина: 1 час
- поиск кода маппинга в плагине: 1 час
- поиск места в маппинге + вникание в технологию: 2 часа

Проект “X” - сколько стоит исправить маппинг из домена Abc в домен Xyz

LUXOFT
Engineering Business Performance

- анализ требований и поиск плагина: 1 час
- поиск кода маппинга в плагине: 1 час
- поиск места в маппинге + вникание в технологию: 2 часа
- фикс: 5 минут

Проект “X” - сколько стоит исправить маппинг из домена Abc в домен Xyz

LUXOFT
Engineering Business Performance

- анализ требований и поиск плагина: 1 час
- поиск кода маппинга в плагине: 1 час
- поиск места в маппинге + вникание в технологию: 2 часа
- фикс: 5 минут
- исправление рухнувших тестов: 2 часа

Проект “X” - сколько стоит исправить маппинг из домена Abc в домен Xyz

LUXOFT
Engineering Business Performance

- анализ требований и поиск плагина: 1 час
- поиск кода маппинга в плагине: 1 час
- поиск места в маппинге + вникание в технологию: 2 часа
- фикс: 5 минут
- исправление рухнувших тестов: 2 часа

Итого: 6 часов 5 минут

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance

- Непонятно **где искать** маппинг в коде приложения

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance

- Непонятно **где искать** маппинг в коде приложения
- Маппинги **сложно читать**

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance

- Непонятно **где искать** маппинг в коде приложения
- Маппинги **сложно читать**
- Маппинги **страшно рефакторить**

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance

- Непонятно **где искать** маппинг в коде приложения
- Маппинги **сложно читать**
- Маппинги **страшно рефакторить**
- Нет **поддержки компилятора** или IDE (удивлены?)

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance

- Непонятно **где искать** маппинг в коде приложения
- Маппинги **сложно читать**
- Маппинги **страшно рефакторить**
- Нет **поддержки компилятора** или IDE (удивлены?)
- Не про**дебажить** код маппинга (опять удивлены?)

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance

- Непонятно **где искать** маппинг в коде приложения
- Маппинги **сложно читать**
- Маппинги **страшно рефакторить**
- Нет **поддержки компилятора** или IDE (удивлены?)
- Не про**дебажить** код маппинга (опять удивлены?)
- Сложно проследить соответствие кода маппинга и **требований**

Проект “X” - что не так с маппингами?

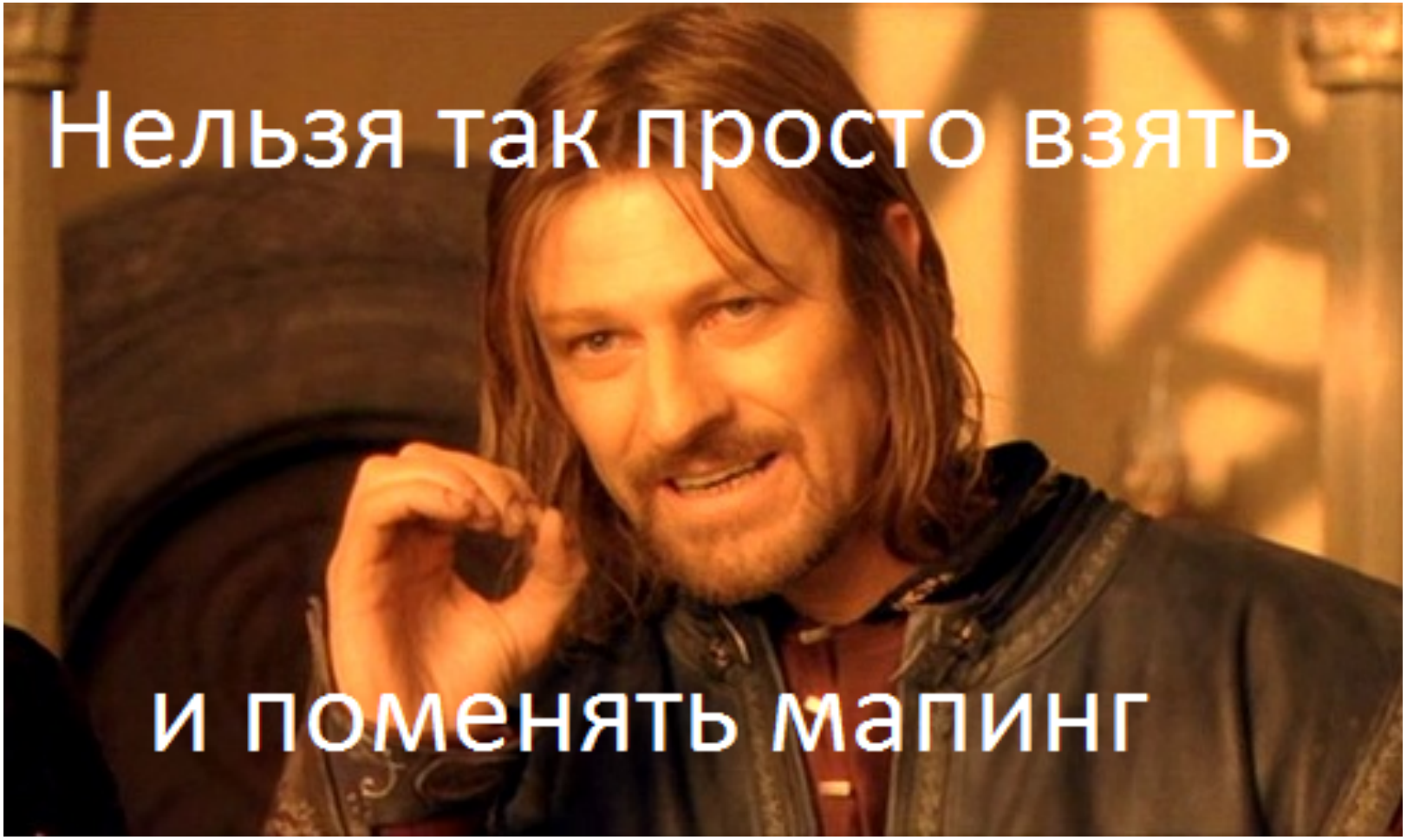
LUXOFT
Engineering Business Performance

- Непонятно **где искать** маппинг в коде приложения
- Маппинги **сложно читать**
- Маппинги **страшно рефакторить**
- Нет **поддержки компилятора** или IDE (удивлены?)
- Не про**дебажить** код маппинга (опять удивлены?)
- Сложно проследить соответствие кода маппинга и **требований**

Да ладно, чувак, мы молоды и круты, для нас не вопрос разобраться с любым мапингом, зафиксить любую багу и завтра выкатить следующий релиз!

Проект “X” - что не так с маппингами?

LUXOFT
Engineering Business Performance



Критерии “правильного” маппинга для прагматичного программиста

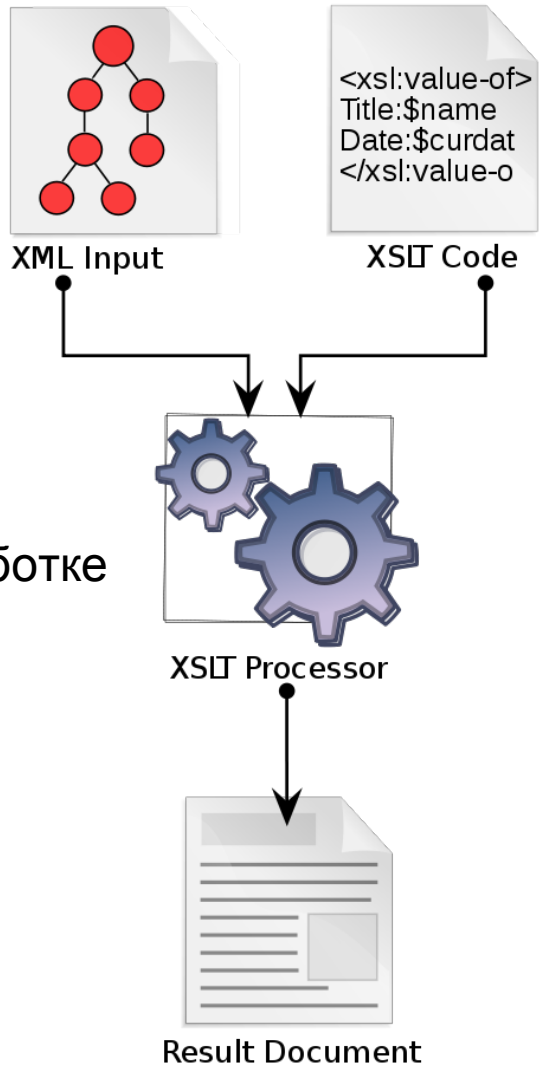
LUXOFT
Engineering Business Performance

1. Понятно **где искать** маппинг в коде приложения
2. Маппинги **легко читать** и делать reverse engineering
3. Есть **поддержка компилятора**
4. Есть **поддержка IDE**
5. Можно **дебажить** код маппинга
6. Легко проследить соответствие **требований** и кода

Так какая технология маппинга у Вас самая популярная?

XSLT

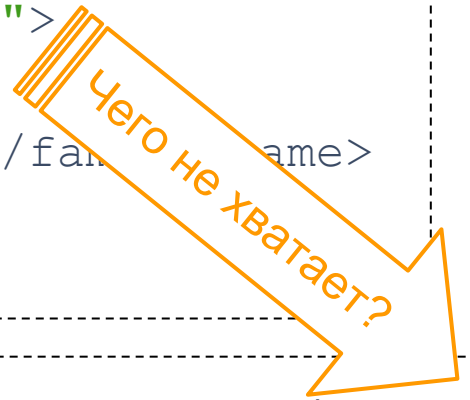
LUXOFT
Engineering Business Performance



1999 г. - v1.0
2001 г. - v2.0
с 2012 г. - v3.0 в разработке

XSLT - пример

```
<?xml version="1.0" ?>
<persons>
  <person username="JL1">
    <name>John</name>
    <family-name>Lennon</family-name>
  </person>
  <person username="AS1">
    <name>Adam</name>
    <family-name>Smith</fa    ame>
  </person>
</persons>
```




```
<?xml version="1.0" encoding="UTF-8"?>
 <result>
   <name username="JL1">John</name>
   <name username="AS1">Adam</name>
 </result>
```

XSLT - пример

LUXOFT
Engineering Business Performance

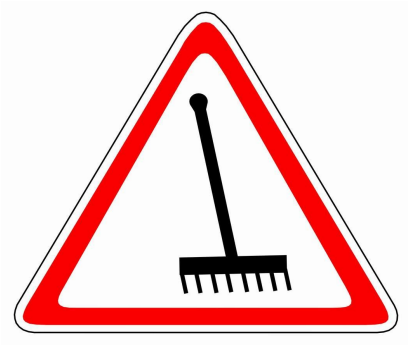
```
<?xml version="1.0" encoding="UTF-8"?>
 <xsl:stylesheet xmlns:xsl="http://www.w3.
 org/1999/XSL/Transform" version="1.0">
   <xsl:output method="xml" indent="yes"/>
   <xsl:template match="/persons">
     <result>
       <xsl:apply-templates select="person"/>
     </result>
   </xsl:template>
   <xsl:template match="person">
     <name username="{@username}">
       <xsl:value-of select="name" />
     </name>
   </xsl:template>
 </xsl:stylesheet>
```

XSLT - плюсы

LUXOFT
Engineering Business Performance

1. декларативный стиль
2. часто используется
3. подходит для XML
4. есть готовые трансформеры для Java
5. из XSLT вызываются статические методы Java
6. в XSLT из Java передаются именованные параметры

XSLT - минусы



1. не дебажится в runtime
2. не подходит для сложных трансформаций
3. нет инструментальной поддержки
4. xslt надо постоянно писать
5. не для JavaBeans
6. не-Java подход

Что сегодня доступно на "рынке мапперов" из java решений?

LUXOFT
Engineering Business Performance

- Commons BeanUtils
- Spring 3 Type Conversion
- Otom
- Transmorph
- EZMorph
- Morph
- Dozer
- OMapper
- JMapper
- java-object-merger
- GeDA
- JBeanMapper
- MapStruct
- MOO
- Smooks
- Orika
- ModelBridge
- Nomin
- ModelMapper

Что сегодня доступно на "рынке мапперов" из **прагматичных** java решений?

LUXOFT
Engineering Business Performance

~~Commons BeanUtils~~

Spring 3 Type Conversion

~~Otom~~

Transmorph

~~EZMorph~~

~~Morph~~

Dozer

~~OMapper~~

~~JMapper~~

~~java-object-merger~~

~~GeDA~~

~~JBeanMapper~~

~~MapStruct~~

~~MOO~~

~~Smooks~~

~~Orika~~

ModelBridge

Nomin

ModelMapper

LUXOFT
Engineering Business Performance

- **Spring 3 Type Conversion**
- Transmorph
- Dozer
- ModelBridge
- Nomin
- ModelMapper

Spring 3 Type Conversion

LUXOFT
Engineering Business Performance

Пакет **org.springframework.core.convert**

- встроеные конвертеры для простых типов: StringToInteger
- SPI и API для кастомных конвертеров

Spring 3 Type Conversion - SPI

LUXOFT
Engineering Business Performance

```
public interface Converter<S, T> {

    T convert(S source);

}
```

Spring 3 Type Conversion - как конвертеры организовать через SPI

LUXOFT
Engineering Business Performance

```
public interface ConverterFactory<S, R> {

 <T extends R> Converter<S, T> getConverter(Class<T> targetT);

}
```

Spring 3 Type Conversion - как конвертеры организовать через API

LUXOFT
Engineering Business Performance

```
public interface ConversionService {

    boolean canConvert(Class<?> sourceT, Class<?> targetT);

    boolean canConvert(TypeDescriptor sourceT,

                       TypeDescriptor targetT);

    <T> T convert(Object source, Class<T> targetT);

    Object convert(Object source, TypeDescriptor sourceT,

                                  TypeDescriptor targetT);

}
```

Spring 3 Type Conversion - как конвертеры организовать через API

LUXOFT
Engineering Business Performance

```
<bean id="conversionService"

class=

"org.springframework.context.support.ConversionServiceFactoryBean">

   <property name="converters">

       <list>

           <bean class="example.MyCustomConverter1"/>

           <bean class="example.MyCustomConverter2"

       </list>

    </property>

</bean>
```

Spring 3 Type Conversion - выводы

LUXOFT
Engineering Business Performance

1. Есть готовые конвертеры
2. Конвертеры находятся в контексте Spring
3. Понятно где искать конвертеры в приложении
4. Общий подход к организации маппинга в приложении

А в стандартной жаве разве ничего такого нет?

Может в JCP что-то сделали в этом направлении?

LUXOFT
Engineering Business Performance

- Spring 3 Type Conversion
- **Transmorph**
- Dozer
- ModelBridge
- Nomin
- ModelMapper

Transmorph (by Cedric Chabanois)

LUXOFT
Engineering Business Performance

1. Много готовых конвертеров:
   - ~40 простых конвертеров: `StringToEnum`
   - `ArrayToArray`: Object[] ->String[], int[][] ->String[][]
   - `ArrayToCollection`: int[] -> Collection<Integer>
2. Можно реализовать кастомный конвертер

Transmorph - пример №1

LUXOFT
Engineering Business Performance

```
class BeanToBeanTO
      extends AbstractSimpleBeanConverter<MyBean, MyBeanTO> {

      @Override
      public MyBeanTO doConvert(ConversionContext context,
                  MyBean source, TypeReference<?> destinationType)
                  throws ConverterException {
        MyBeanTO result = new MyBeanTO();
        result.setLength(source.getSize());
        result.setMyStrings(convertElement(context,
                    source.getMyStrings(), String[].class));
        return result;
      }
}
```

Transmorph - пример №1

LUXOFT
Engineering Business Performance

```
DefaultConverters defaultConverters =
        new DefaultConverters();
defaultConverters.addConverter(new BeanToBeanTO());
defaultConverters.add…
Transmorph transmorph = new Transmorph(defaultConverters);
ConversionContext context = new ConversionContext();
context.add("key", value);
MyBean myBean = new MyBean();
myBean.set…
MyBeanTO myBeanTO = transmorph.convert(
        context, myBean, MyBeanTO.class);
```

Transmorph - пример №2

LUXOFT
Engineering Business Performance

```
String[] arrayOfStrings = new String[]
    {"0","1","2","3","4","5"};

TypeReference<List<? extends Number>> typeReference =
    new TypeReference<List<? extends Number>>() {};

List<? extends Number> listOfNumbers =
    transmorph.convert(arrayOfStrings, typeReference);
```

Почему нельзя написать `List<? extends Number>.class`?

Transmorph - пример №2

LUXOFT
Engineering Business Performance

```
String[] arrayOfStrings = new String[]
    {"0","1","2","3","4","5"};

TypeReference<List<? extends Number>> typeReference =
    new TypeReference<List<? extends Number>>() {};

List<? extends Number> listOfNumbers =
    transmorph.convert(arrayOfStrings, typeReference);
```

Почему нельзя написать `List<? extends Number>.class`?

Из-за `type-erasure`.

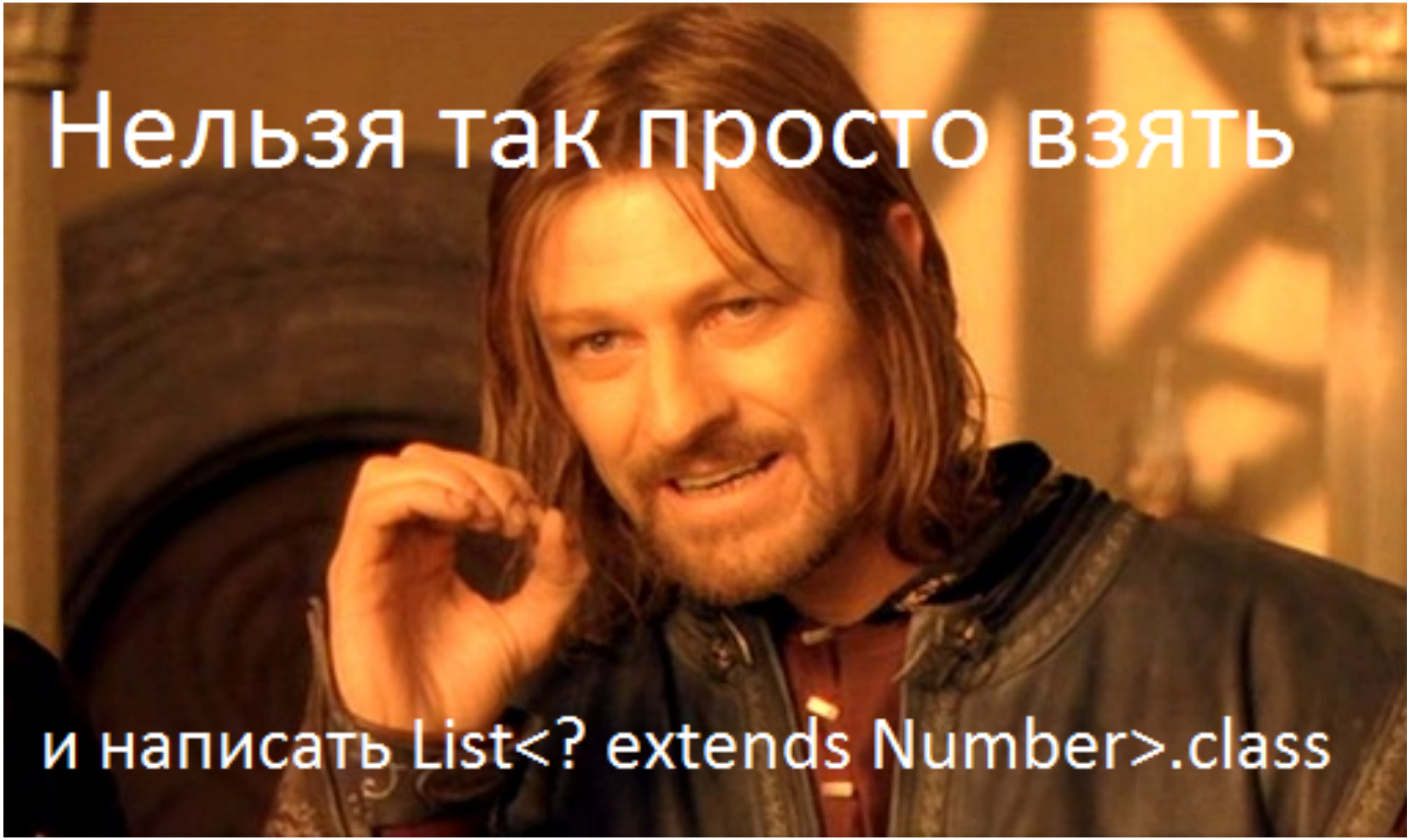
Решение (by Neal Gafter): `super type tokens`

Базовая реализация: **`public abstract class`** *TypeReference*`<T> {}`

Подробнее тут - http://gafter.blogspot.fr/2006/12/super-type-tokens.html

Transmorph - пример №2

LUXOFT
Engineering Business Performance



Transmorph - выводы

1. Проект новый и “живой”
2. Маппинг “ручной” и быстрый
3. Нет внешних зависимостей
4. Хороший код, покрыт тестами
5. Конвертация type-safe
6. Дебажится
7. Поддержка компилятора и IDE
8. Готовые конвертеры
9. Кастомные конвертеры
10. ~~`convert(context, sourceObject, targetObject)`~~

А нет чего-нибудь такого, чтобы самому писать поменьше java кода и чтобы как-то все само мапилось?

LUXOFT
Engineering Business Performance

- Spring 3 Type Conversion
- Transmorph
- **Dozer**
- ModelBridge
- Nomin
- ModelMapper

Dozer - API basics

LUXOFT
Engineering Business Performance

```
<mapping>
  <class-a>yourpackage1.YourSourceClassName</class-a>
  <class-b>yourpackage2.YourDestinationClassName</class-b>
    <field>
      <A>yourSourceFieldName</A>
      <B>yourDestinationFieldName</B>
    </field>
</mapping>
```

```
DestinationObject destObject1 =
    mapper.map(sourceObject, DestinationObject.class);
DestinationObject destObject2 = new DestinationObject();
mapper.map(sourceObject, destObject2);
```

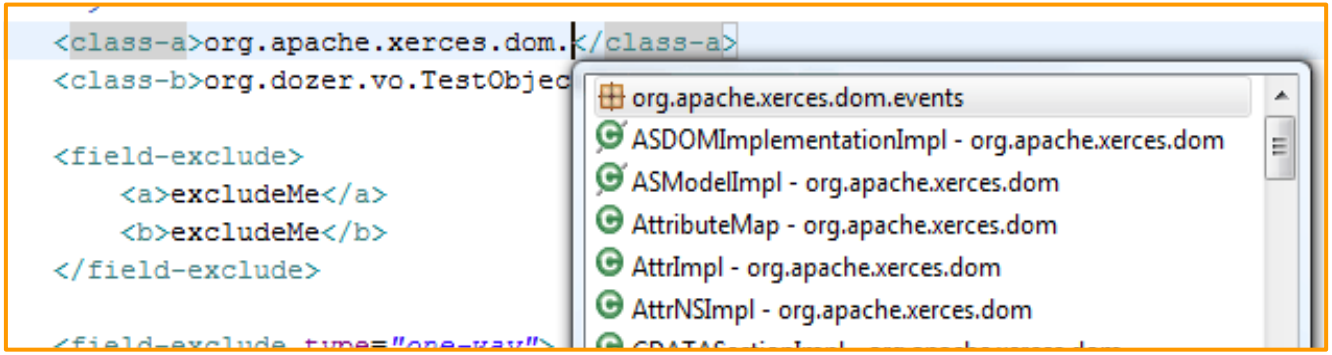
Dozer - некоторые фишки

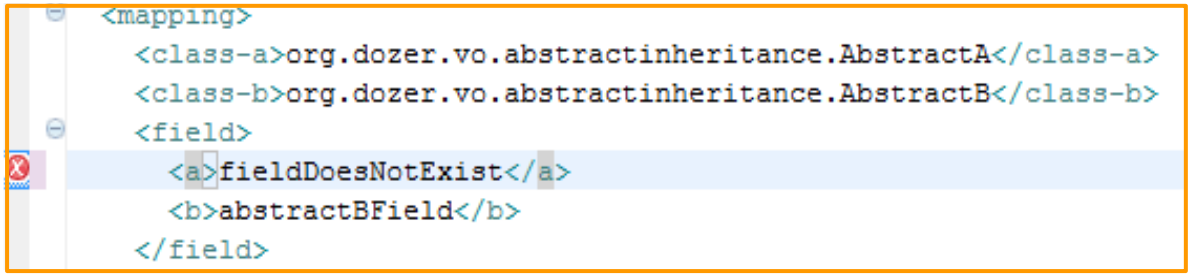
LUXOFT
Engineering Business Performance

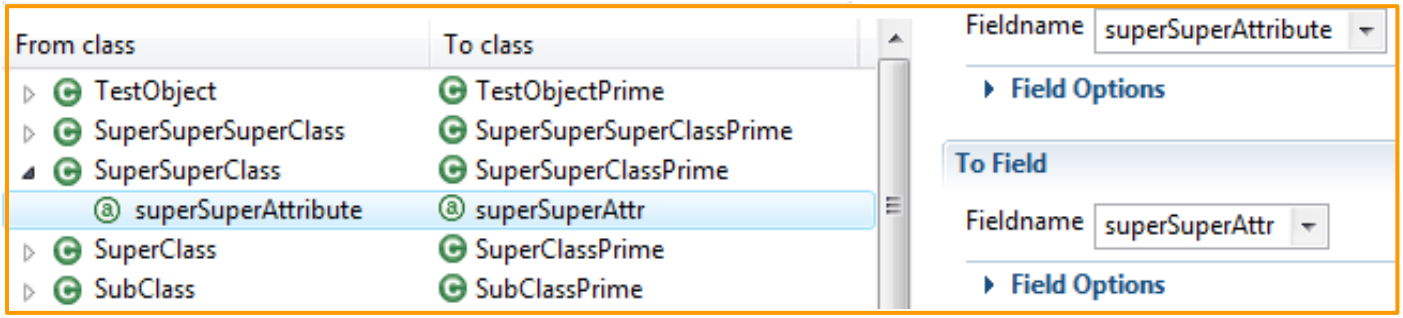
1. автоматическое приведение типов
2. рекурсивный маппинг
3. wildcard="false"
4. маппинг коллекций
5. map-id="caseA"
6. create-method

Dozer - Eclipse Plugin

1. поддержка синтаксиса и автодополнение
2. валидатор
3. редактор маппинга

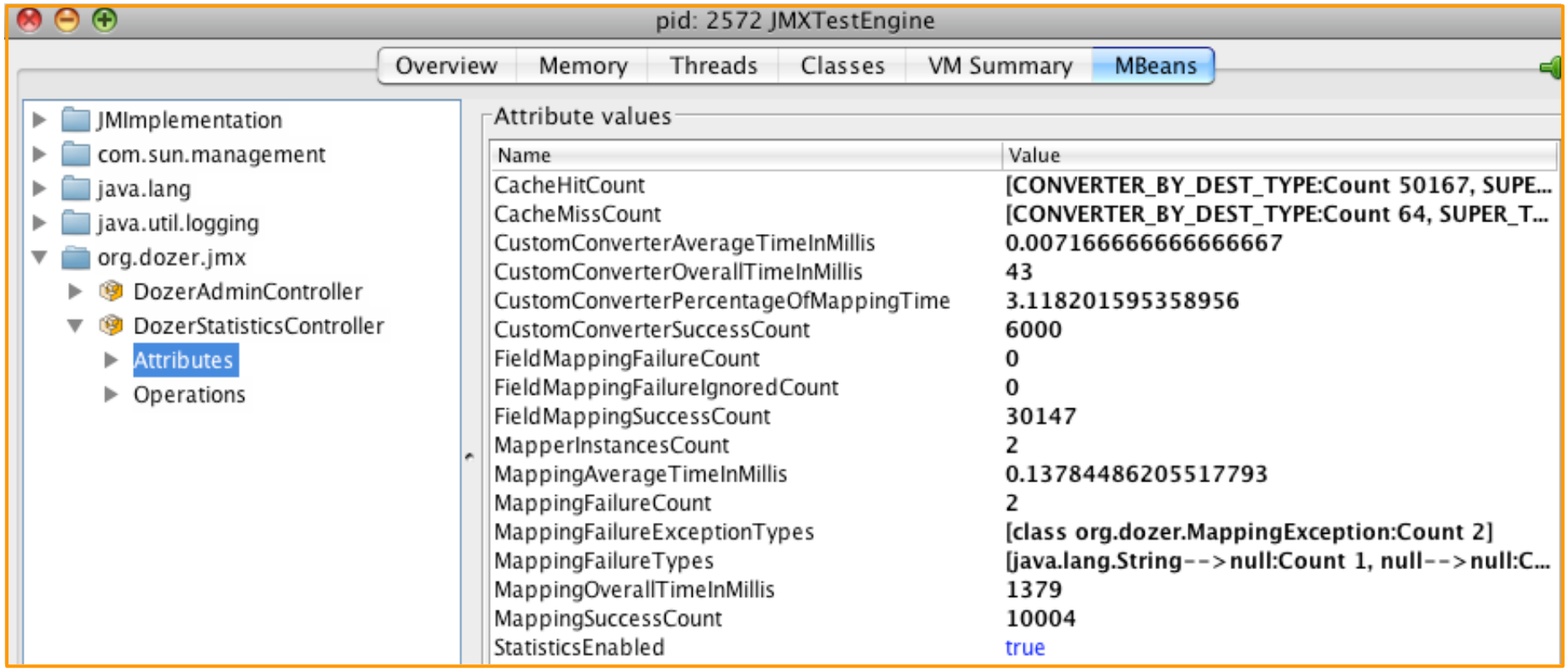






Dozer - стастистика

LUXOFT
Engineering Business Performance



Dozer - IoC, JAXB и XMLBeans

LUXOFT
Engineering Business Performance

1. Интеграция со Spring через DozerBeanMapperFactoryBean
2. Интеграция с JAXB и XMLBeans через JAXBBeanFactory и XMLBeanFactory

Dozer - annotations или как начать терять веру в человечество

LUXOFT
Engineering Business Performance

```
public class SourceBean {
    private Long id;
    private String name;
    @Mapping("binaryData")
    private String data;
    @Mapping("pk")
    public Long getId() {
        return id;
    }
    public String getName() {
        return name;
    }
}
```

Annotation-based libs или как окончательно потерять веру в человечество

LUXOFT
Engineering Business Performance

OMapper (by sachin.magician)

GeDA (by Denis Pavlov) - Javassist / BCEL / Sun JavaC

JMapper (by Alessandro Vurro)

MapStruct (by Gunnar Morling) - JSR 269

MOO (by Geoffrey Wiseman)

Dozer - уродливый программный маппинг

LUXOFT
Engineering Business Performance

```
BeanMappingBuilder builder = new BeanMappingBuilder() {
      protected void configure() {
        mapping(Bean1.class, Bean2.class,
                oneWay(),
                mapId("A"),
                mapNull(true)
        )
                .exclude("excluded")
                .fields("src1", "dest1",
                        copyByReference(),
                        collectionStrategy(true,
                            RelationshipType.NON_CUMULATIVE),
                        hintA(String.class),
                        hintB(Integer.class),
                        fieldOneWay(),
                )
                .fields("src2", "dest2");
      }
    };
```

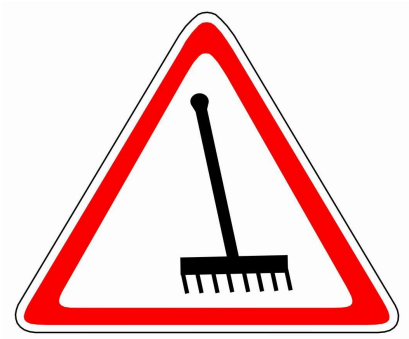
Dozer - плюсы

LUXOFT
Engineering Business Performance

1. Самый популярный проект
2. Очень гибкая конфигурация
3. Плагин под *Eclipse*
4. Статистика и интеграция
5. `org.dozer.CustomConverter`

Dozer - минусы

LUXOFT
Engineering Business Performance



1. Маппинг описывается в XML
2. Не дебажится
3. Нет поддержки компилятора
4. Есть ограничение на глубину интроспекции
5. Стектрейс ошибки непросто понять
6. Требуется высокий % покрытия юнит тестами
7. Маппинг через аннотации
8. Уродливый программный маппинг

LUXOFT
Engineering Business Performance

- Spring 3 Type Conversion
- Transmorph
- Dozer
- **ModelBridge**
- Nomin
- ModelMapper

ModelBridge (by Ramy Hardan)

LUXOFT
Engineering Business Performance

Плагин для Eclipse, генерит java код с языка m2m

```
rule WorkOrderToInvoice
    from SrcWorkOrder wo
    create SinkInvoice
    map
        setTotal = wo.price
        setPositions = wo.lineItems
rule LineItemToPosition
    from SrcLineItem li where li.part != null
    create SinkPosition
        map
            setInvoice = parent
            setReference = li.lineItemRef
            setAmount = li.part.price?.value.multiply(
                new BigDecimal(li.part.quantity))
            setDescription = li.part.partRef
```

ModelBridge (by Ramy Hardan)

LUXOFT
Engineering Business Performance

Построен на технологиях Eclipse:

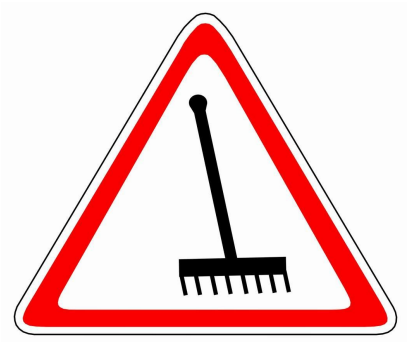
1. Xtext - framework для разработки ЯП и DSL
2. EMF - моделирование и генерация кода

ModelBridge (by Ramy Hardan)

LUXOFT
Engineering Business Performance

Плюсы:

1. Плагин генерит обычный java код
2. Полноценная поддержка m2m



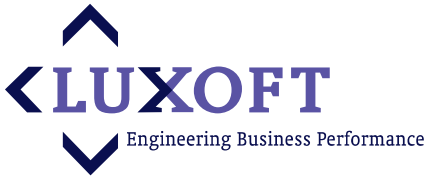
Минусы:

1. Около года нет комитов на GitHub
2. Нет доки, только 5-ти минутное видео
3. Запутанный код проекта

LUXOFT
Engineering Business Performance

- Spring 3 Type Conversion
- Transmorph
- Dozer
- ModelBridge
- **Nomin**
- ModelMapper

Nomin (by Dmitri Dobrynin, Luxoft alumni)




```
mappingFor a: Person, b: Employee
a.name = b.name
a.fullName = { b.firstName + " " + b.lastName }
a.children = b.details.kids
hint a: List[ExtendedChildren], b: Set[ExtendedKid]
a.strDate = b.details.birth
dateFormat "dd-MM-yyyy"
a.gender = b.details.sex
simple ([Gender.MALE, true], [Gender.FEMALE, false])
a.snn = b.employeeId
convert to_a: { eId -> repository.findByEmployeeId(eId) },
        to_b: { snn -> repository.findBySnn(snn) }
```

Nomin - некоторые фишки

LUXOFT
Engineering Business Performance

- рекурсивный
- поддерживает неявный маппинг
- поддерживает коллекции
- с FastIntrospector в 10 раз быстрее Dozer (CGLIB FastMethod и FastClass)
- декларативная обработка ошибок:

```
handle throwables:[IllegalArgumentException,
NullPointerException],
handler: {
 logger.logError("Exception in " + mapping + ". Failed
 rule " + failed, throwable)
}
```

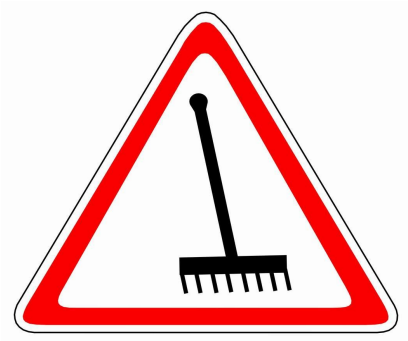
Nomin - плюсы

LUXOFT
Engineering Business Performance

1. Компактный DSL
2. Код на Groovy компактнее Java
3. Nomin сам создаёт коллекции
4. Можно писать на Java
5. Код проекта хорош, доступен в hg и покрыт тестами
6. Groovy сейчас под SpringSource и под VMware
7. В Groovy 2.0 появились @TypeChecked и @CompileStatic

Nomin - минусы

LUXOFT
Engineering Business Performance

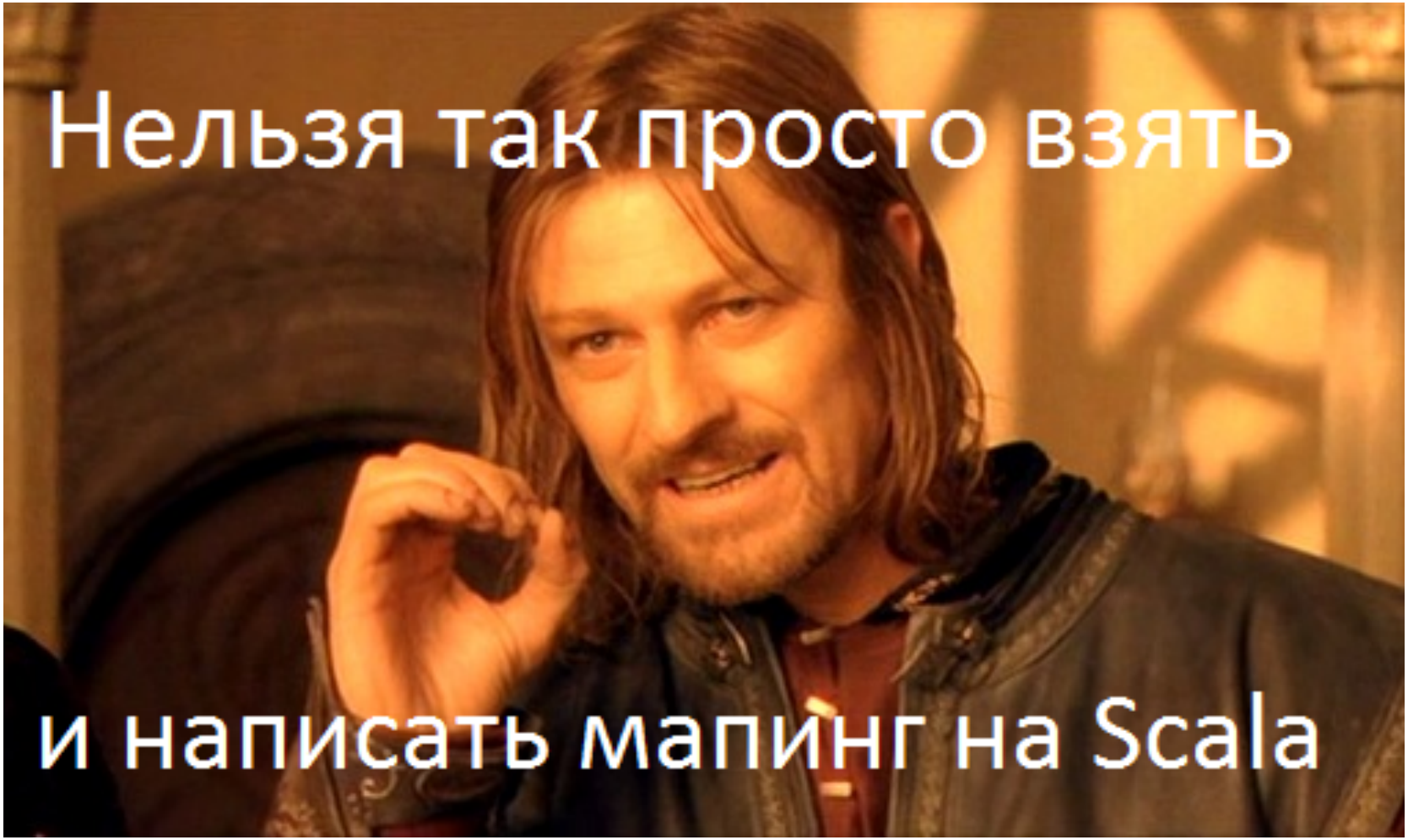


1. Это не маппинг, а конфигурация маппинга
2. Нельзя продебажить
3. Нет поддержки компилятора и IDE
4. Описки маппинга вылетают в runtime
5. Требует высокой степени покрытия unit tests
6. Cobertura ошибочно показывает ~100%
7. Stacktrace неочевиден
8. Нет активности на SourceForge с 2011 года

LUXOFT
Engineering Business Performance

Маппинг на Scala DSL решит все проблемы прагматичного программиста?

LUXOFT
Engineering Business Performance



LUXOFT
Engineering Business Performance

- Spring 3 Type Conversion
- Transmorph
- Dozer
- ModelBridge
- Nomin
- **ModelMapper**

ModelMapper (by Jonathan Halterman)

LUXOFT
Engineering Business Performance

```
public class PersonMap
        extends PropertyMap<Person, PersonDTO>(){
   ...
    map().setName(source.getFirstName());
    map().setEmployer("Luxoft");
    map(source.getAge()).setAgeString(null);
    map(source.getAgeString()).setAge((short) 0);
    map(21).setAgeString(null);
    map().getCustomer().setName(
        source.getPerson().getFirstName());
    using(personToStreetConvertor).map(source).setStreet(null);
    with(managerProvider).map().setManager(source.getPerson());
}
```

ModelMapper - converters

LUXOFT
Engineering Business Performance

```
Converter<Person, Street> personToStreetConverter =
    new AbstractConverter<Person, Street>() {
      @Override
      protected Street convert(Person person) {
        return person == null ? null : person.getStreet();
      }
    };
```

ModelMapper - providers

LUXOFT
Engineering Business Performance

```
Provider<Person> managerProvider =
    new AbstractProvider<Person>() {
      @Override
      public Person get() {
        return new Person("Vasya", "Pupkin", 42);
      }
    }
```

ModelMapper - conditions

LUXOFT
Engineering Business Performance

```
Condition notNull = new Condition() {
  public boolean applies(MappingContext<?, ?> context) {
    return context.getSource() != null;
  }
};
```

```
when(notNull).map().setName(source.getName());
when(someCondition)
    .with(personProvider)
    .using(personConverter)
    .map().setPerson(source.getPerson());
when(Conditions.and(txActive, inScope))
    .map().setRequest(source.getRequest());
```

Что-то напоминает... Mockito?

ModelMapper - generics

LUXOFT
Engineering Business Performance

Так сработает ?

```
List<Integer> numbers = buildIntegers();
List<String> characters = new ArrayList<String>();
modelMapper.map(integers, characters);
```

ModelMapper - generics

LUXOFT
Engineering Business Performance

~~Так сработает ?~~ НЕТ!

```
List<Integer> numbers = buildIntegers();
List<String> characters = new ArrayList<String>();
modelMapper.map(integers, characters);
```

Сработает так:

```
List<Integer> numbers = buildIntegers();
Type listType = new TypeToken<List<String>>(){}.getType();
List<String> characters = modelMapper.map(numbers, listType);
```

ModelMapper - matching

LUXOFT
Engineering Business Performance

```
public class Person {
   private Address address;
}
public class Address {
   private String street;
   private String city;
}
```

```
public class PersonDTO {
   private String addressStreet;
   private String addressCity;
}
```

Фазы матчинга и точки кастомизации:

eligibility -> transformation -> tokenization -> matching -> handling ambiguity

`NamingConvention,NameTransformer,NameTokenizer,MatchingStrategy`

ModelMapper - плюсы

LUXOFT
Engineering Business Performance

1. “Живой” проект на github версии 0.6.1 с юнит-тестами
2. Поддержка компилятора, поддержка IDE, рефакторинги
3. Маппинг декларативный, простой и понятный
4. Минимальный набор фишек: условия, конвертеры
5. “Умный” матчинг для имплицитного маппинга

ModelMapper - минусы

1. это не маппинг, а конфигурация маппинга
2. не продебажить
3. нет поддержки маппинга коллекций

LUXOFT
Engineering Business Performance

ИТОГИ

ИТОГИ - инструменты для маппинга

- reflection и cglib FastMethod
- Аннотации: compile-time and runtime
- JSR 269
- Генерация java кода
- Генерация байт кода: Javasist, Bcel, javac, ecj
- Магия Eclipse: Xtext и EMF
- Магия Groovy
- Магия Scala
- Декларативная java
- Старая добрая императивная java

ИТОГИ - номинации и победители

LUXOFT
Engineering Business Performance

ИТОГИ - номинации и победители

1. Самый анти-java маппинг - XSLT

ИТОГИ - номинации и победители

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring

ИТОГИ - номинации и победители

LUXOFT
Engineering Business Performance

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf

ИТОГИ - номинации и победители

LUXOFT
Engineering Business Performance

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf
4. Самая универсальная либа - Dozer

ИТОГИ - номинации и победители

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf
4. Самая универсальная либа - Dozer
5. Самый уродливый маппинг - программный Dozer

ИТОГИ - номинации и победители

LUXOFT
Engineering Business Performance

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf
4. Самая универсальная либа - Dozer
5. Самый уродливый маппинг - программный Dozer
6. Самый лаконичный маппинг - Nomin

ИТОГИ - номинации и победители

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf
4. Самая универсальная либа - Dozer
5. Самый уродливый маппинг - программный Dozer
6. Самый лаконичный маппинг - Nomin
7. Самый многообещающий маппинг - Scala DSL

ИТОГИ - номинации и победители

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf
4. Самая универсальная либа - Dozer
5. Самый уродливый маппинг - программный Dozer
6. Самый лаконичный маппинг - Nomin
7. Самый многообещающий маппинг - Scala DSL
8. Самый быстрый маппинг - императивная java

ИТОГИ - номинации и победители

LUXOFT
Engineering Business Performance

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf
4. Самая универсальная либа - Dozer
5. Самый уродливый маппинг - программный Dozer
6. Самый лаконичный маппинг - Nomin
7. Самый многообещающий маппинг - Scala DSL
8. Самый быстрый маппинг - императивная java
9. Самая “типичная” проблема маппинга - type-erasure

ИТОГИ - номинации и победители

1. Самый анти-java маппинг - XSLT
2. Самый малоинтрузивный маппинг - Spring
3. Самый “правильный” маппинг - Transmorf
4. Самая универсальная либа - Dozer
5. Самый уродливый маппинг - программный Dozer
6. Самый лаконичный маппинг - Nomin
7. Самый многообещающий маппинг - Scala DSL
8. Самый быстрый маппинг - императивная java
9. Самая “типичная” проблема маппинга - type-erasure
10. Самое редкое слово доклада - EJB

P.S. Что делать, если я проспал весь доклад?

LUXOFT
Engineering Business Performance

```
if (iHaveSpring()) useSpring3TypeConverterAPI()
else createCustomConverterFactory();
if (iLikeGroovy()) {
    useNomin(); createManyUnitTests();
} else if (iWantToWriteInJavaOnly()){
    if (iWantToDebug()) useTransmorf();
    else { useModelMapper(); createManyUnitTests(); }
} else if (iWantSomeMagic()){
    createMyOwnScalaMappingDSL(); tryCodeGeneration();
} else {
    useDozerLikeOthersDo(); createManyUnitTests();
}
```

LUXOFT
Engineering Business Performance

Спасибо за внимание и лёгкого Вам маппинга!

А можно вопрос?