



# Java 11: Золотой теленок

вип Тарасов Владимир  
отдел ДЦР ДКЦ

Naumen, 2019

# Disclaimer

*Доклад основан на реальных событиях. Слабонервным не слушать и не смотреть. Все, что будет сказано, является личным мнением автора. Автор не несет никакой ответственности за то, что говорит и показывает. И вообще, автор не уверен в том, что он говорит*



# План, он же Гога, он же Жора, он же оглавление, он же о чём

- А нафига оно нам?
- И как мы будем это делать?
- Чо, так просто?
- Кажется, что-то пошло не так...
- Я устал — я ухожу

# А нафига оно нам?



# Старый добрый Oracle Java 8

- С марта 2014
- Мы так их полюбили:
  - *Default interface methods*
  - *Lambda*
  - *Optional*
  - *Stream API*
  - *Java Time API*
  - *Repeatable annotations*

# Oracle Java 8 снимаю с бесплатной поддержки



- Java SE 8 *is going through the End of Public Updates process for legacy releases. Oracle will continue to provide free public updates and auto updates of Java SE 8, until at January 2019 for Commercial Users. Commercial Users continue to get free updates to Java SE 8 from OTN for free under the BCL license*

# И чо? Есть же Java 9 или 10

- Релиз в сентябре 2017 и марте 2018
- Новый релизный цикл с Java 9
- Новые фичи языка
- Улучшения JVM
- Новые GC

# Java 9 или 10 не LTS!



# Ну дак давайте перейдем на Oracle Java 11

- С сентября 2018 — свеженькая, новенькая
- **LTS!**

# пИчалька! Обновилась лицензия

- *You may not: use the Programs for any data processing or any commercial, production, or internal business purposes other than developing, testing, prototyping, and demonstrating your Application*



# Свободный задорный OpenJDK

- С 2006 года
- Огромное сообщество
- В Linux дистрибутивах
- Реализует спецификации Java
- Протестирован TCK
- GPL v2 + the Classpath exception
- Oracle JDK → OpenJDK

# А что если сразу на OpenJDK 11?

- Project Amber (var — delivered)
- Unsafe → Variable Handles
- REPL JShell
- Compact Strings
- HTTP Client API
- Flight Recorder
- **Project Jigsaw**

# А давайте!

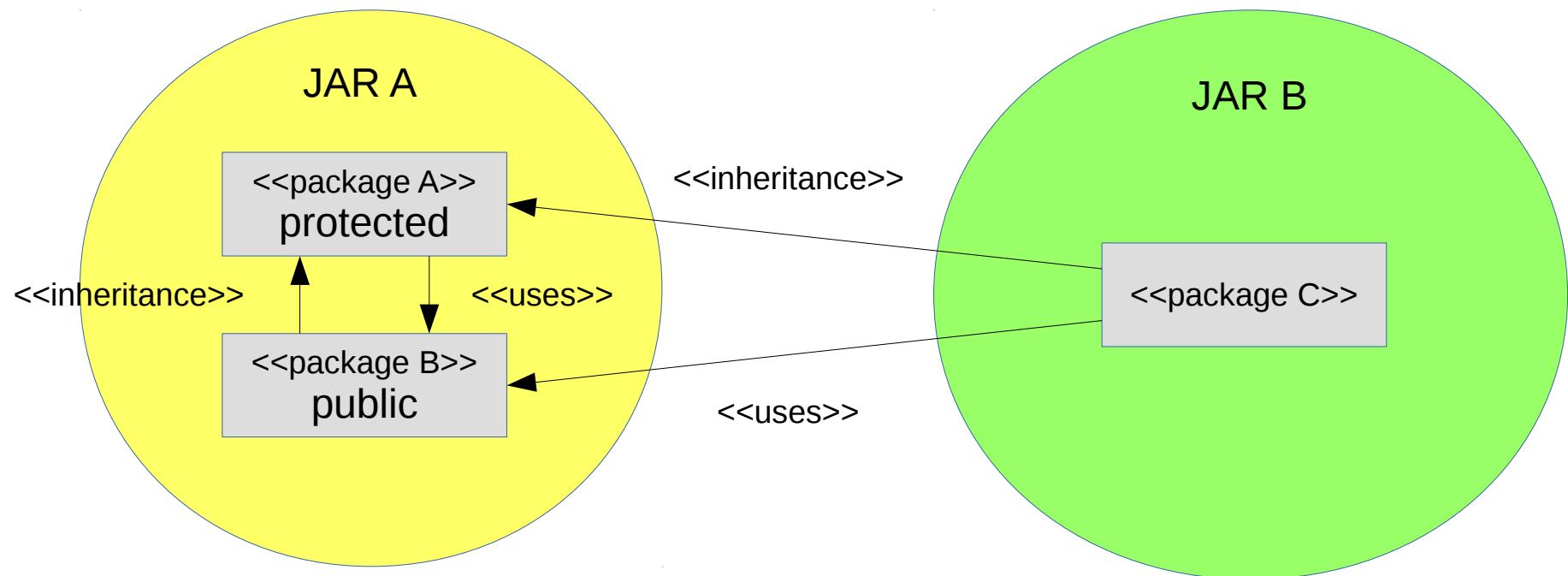


# Project Jigsaw

- Modular Java Platform (JPMS) — разделение rt.jar на модули
- Modular Java Application
- Java Application Image with JLink

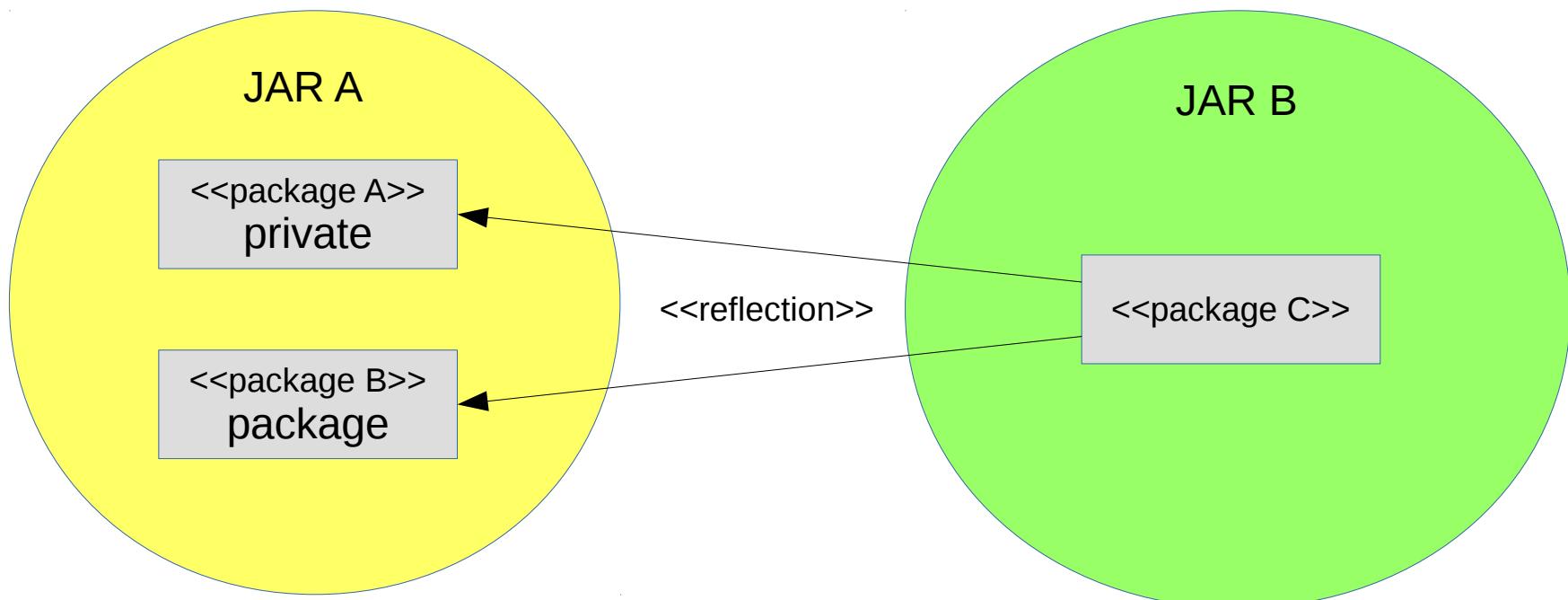
# Видите здесь модульность? А ее нет

- Проблема protected и public



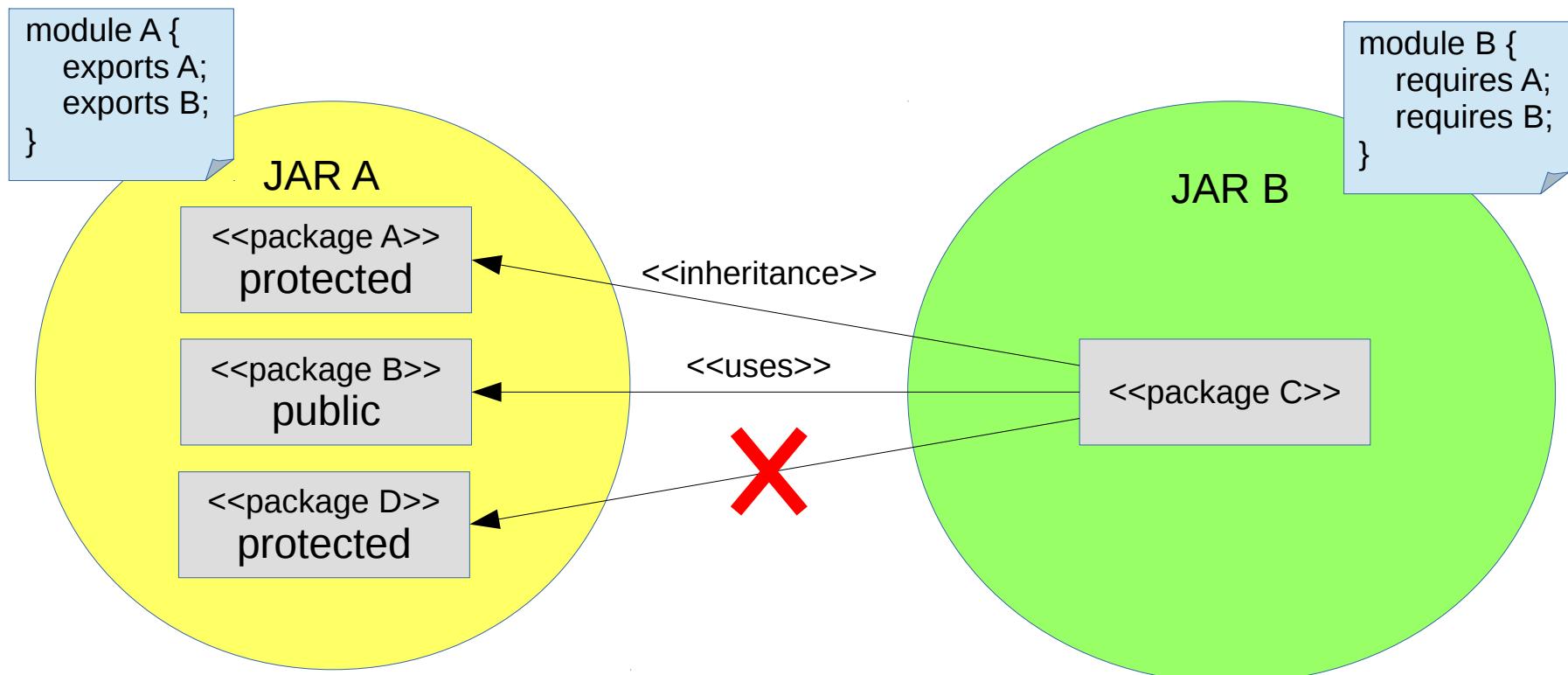
# Видите здесь модульность? А ее нет

- Проблема package и private



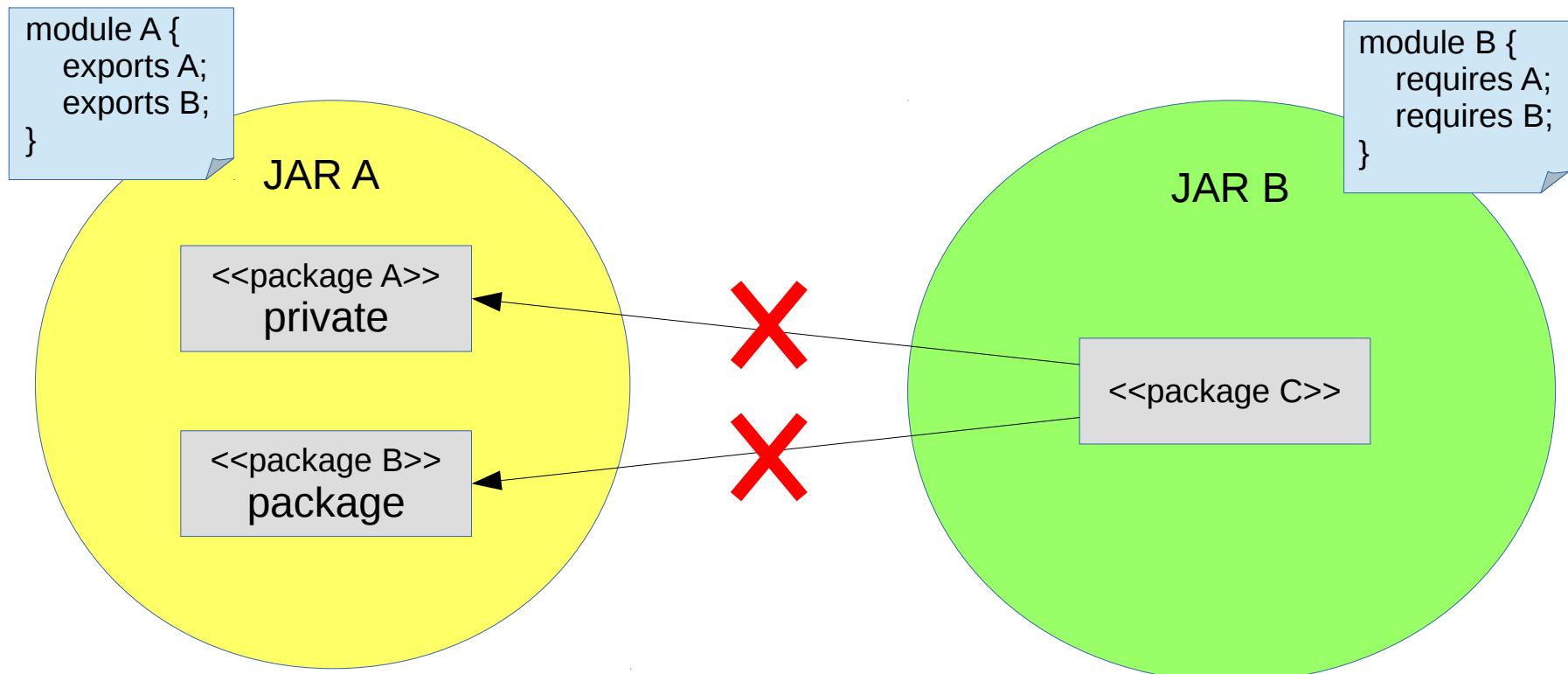
# Модульность Jigsaw — я прозрел, доктор

- Нет проблем с protected и public



# Модульность Jigsaw — я прозрел, доктор

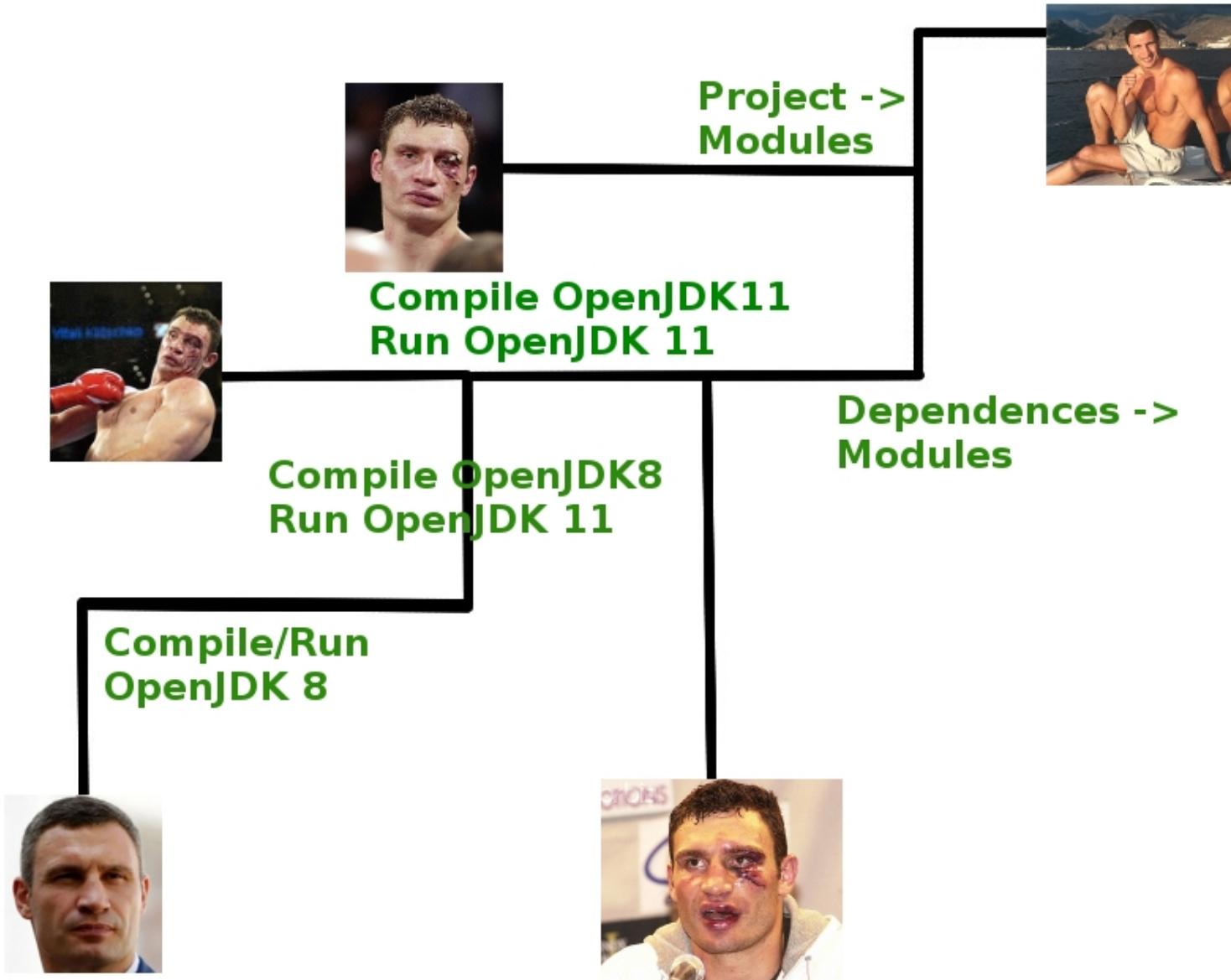
- Нет проблем с package и private



# **И как мы будем это делать?**



# Поможем Виталику добраться до отдыха?



# Компиляция и запуск на OpenJDK 8

- В скриптах .sh:

```
wget "http://link.to.oracle.jdk8.rpm"  
sudo yum localinstall jdk-8.rpm  
rm jdk-8.rpm  
sudo yum install java-1.8.0-openjdk-devel
```

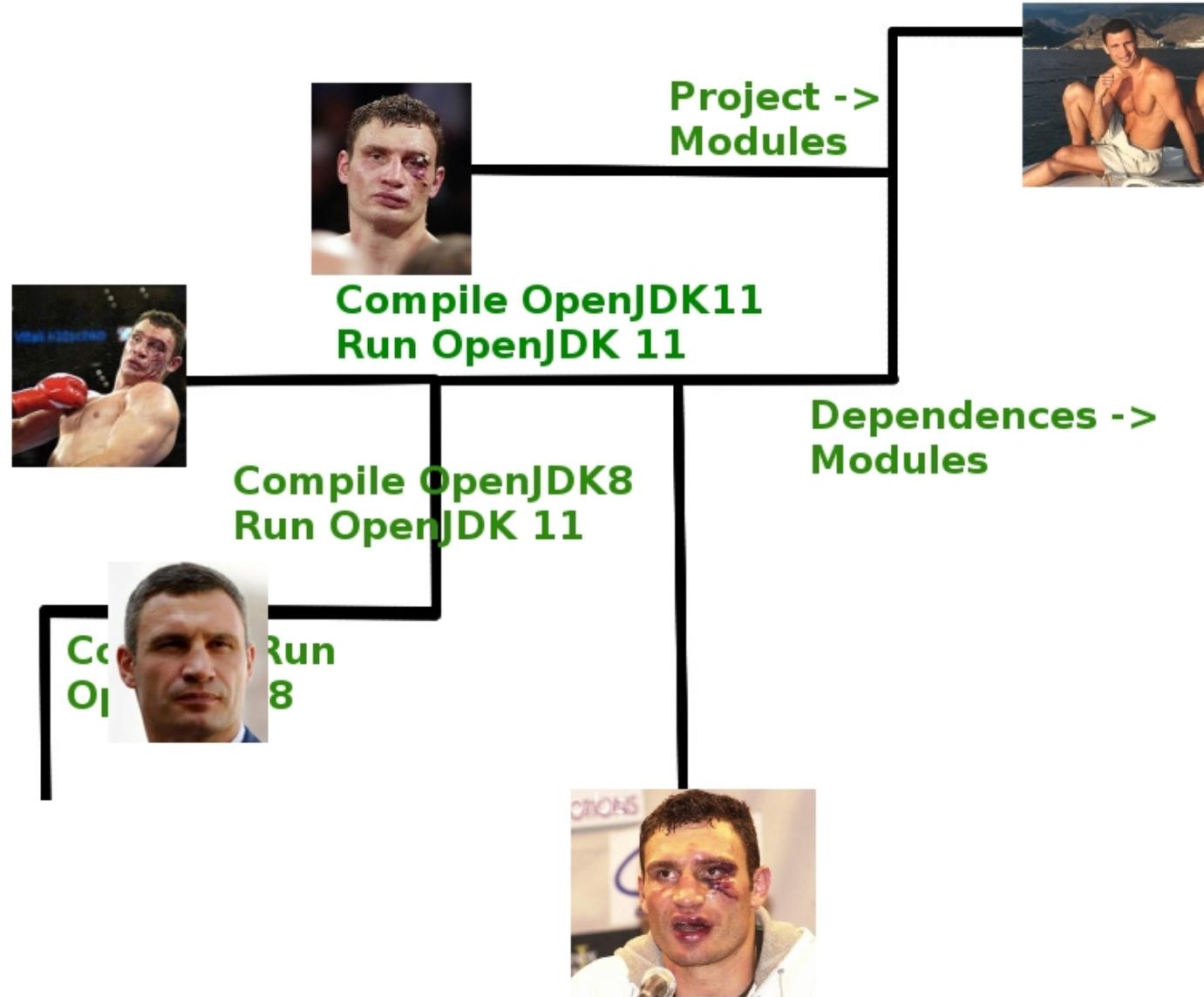
- Изменился JAVA\_HOME:

```
JAVA_HOME="/usr/java/latest"  
JAVA_HOME="/usr/lib/jvm/jre-1.8.0"
```

# Сработало!



# Где же наш Виталик?



# Компиляция на OpenJDK 8 → запуск на OpenJDK 11

- В скриптах запуска .sh:

```
sudo yum install java-1.8.0-openjdk-devel  
sudo yum install java-11-openjdk-devel
```

- Изменился JAVA\_HOME:

```
JAVA_HOME="/usr/lib/jvm/jre-1.8.0"  
JAVA_HOME="/usr/lib/jvm/jre-11"
```

# JVM не стартировала

- Unrecognized VM option 'PrintGCTimeStamps'

Error: Could not create the Java Virtual Machine.

Error: A fatal exception has occurred. Program will exit.

- `-XX:+PrintGCDetails -XX:+PrintGCTimeStamps`

# JEP 158: Unified JVM Logging

- Появилось с Java 9
- Introduce a common logging system for all components of the JVM
- Tags: compiler, gc, classload, metaspace, svc, jfr, ...
- Levels: error, warning, info, debug, trace, develop
- **-Xlog:gc\***

# JVM стартовала, но приложение упало

javax.xml.bind.JAXBException: Implementation of JAXB-API has not been found on module path or classpath.

– with linked exception:

[java.lang.ClassNotFoundException:  
com.sun.xml.internal.bind.v2.ContextFactory]

# JEP 320: Remove the Java EE and CORBA Modules

- rt.jar распилен на модули
- Java SE 9 и 10 → метамодуль java.se.ee — @Deprecated
- Java SE 11 — он удален
- Нужно добавлять зависимости на Java EE модули отдельно

# Добавляем зависимости на xml.bind

- dependencyRecommendations {  
  
    map recommendations: [ ...  
  
        'com.sun.xml.bind:jaxb-impl': '2.3.1',  
  
        'com.sun.xml.bind:jaxb-core': '2.3.0.1'  
  
    ]  
}
- dependencies {  
  
    ...  
  
    compile("javax.xml.bind", "jaxb-api")  
  
    compile("com.sun.xml.bind", "jaxb-core")  
  
    compile("com.sun.xml.bind", "jaxb-impl")  
  
}

# Приложение стартовало, но все ли ОК?

- При запуске на Java 8
  - Даты: 20.05.17
  - Проценты: 10.7%
  - Валюты: 12 руб., 12 USD, 12 EUR
- При запуске на Java 11
  - Даты: 20.05.2017
  - Проценты: 10.7 \_ %
  - Валюты: 12 ₽, 12 \$, 12 €

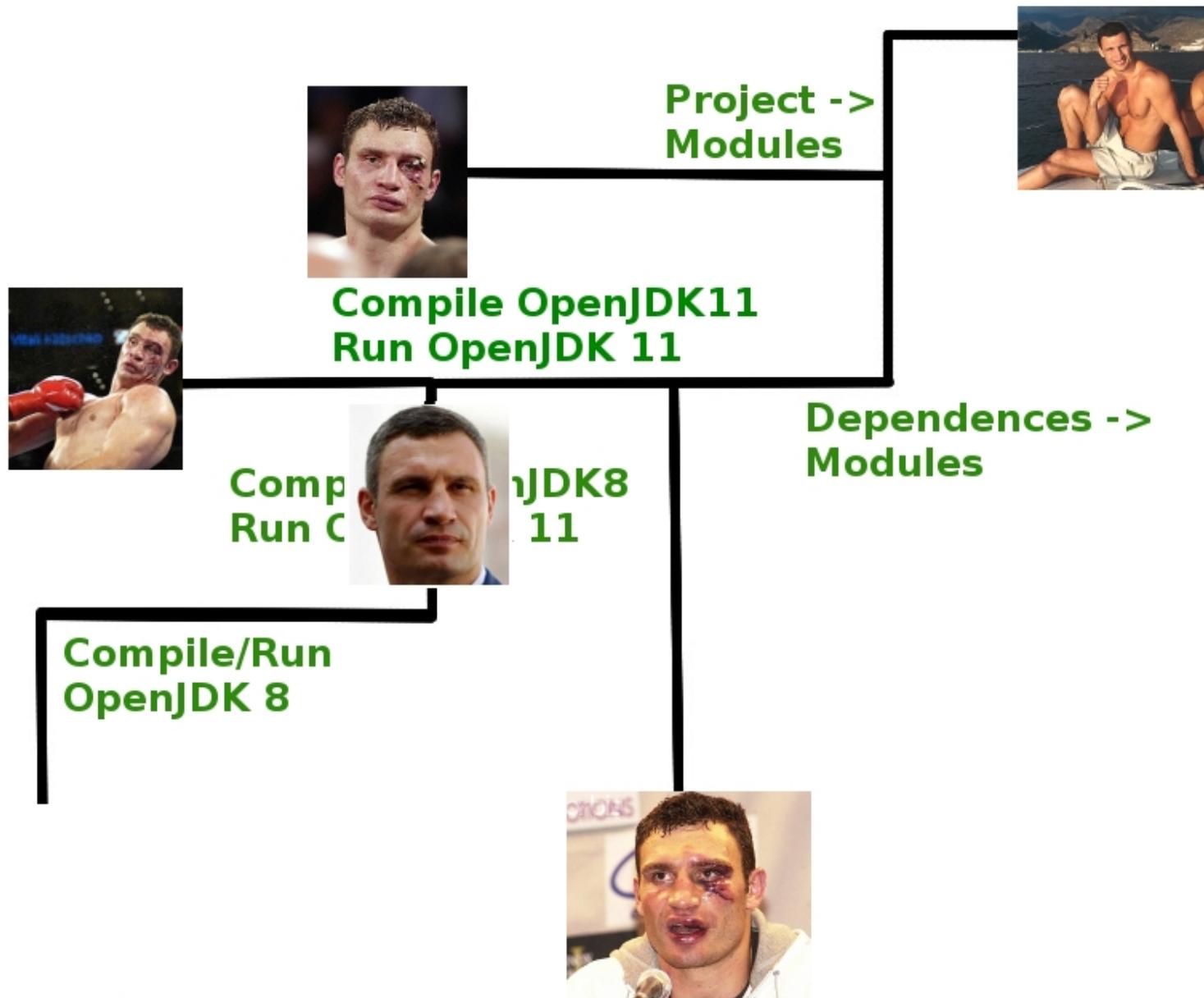
# CLDR Locale Data Enabled by Default

- Появился → Java 8
- По умолчанию → Java 9
- CLDR (Common Locale Data Repository) — проект консорциума Unicode, обеспечивающий языковые настройки в XML
- **Режим обратной совместимости:**
  - Djava.locale.providers=COMPAT

# Exyyy!



# А где Виталик?



# Компиляция и запуск на OpenJDK 11

- В скриптах сборки .gradle:

```
sourceCompatibility = JavaVersion.VERSION_1_8
targetCompatibility = JavaVersion.VERSION_1_8
sourceCompatibility = JavaVersion.VERSION_11
targetCompatibility = JavaVersion.VERSION_11
```

# Упал Java-тест с формированием XML

- Ожидали

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    ...
</root>
```

- Получили

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root>
    ...
</root>
```

# Изменилась реализация XMLEventWriter

- com.sun.xml.internal.stream.writers.XMLEventWriterImpl
- Переписали тест → добавили standalone=no в ожидаемые xml

# Упал Groovy-тест с формированием XML

- **Ожидали**

```
groovy.xml.XmlUtil.serialize(message) == """
<?xml version="1.0" encoding="UTF-8"?><NCC>
    <Request name="Beer"/>
</NCC>"""
```

- **Получили**

```
groovy.xml.XmlUtil.serialize(message) == """
<?xml version="1.0" encoding="UTF-8"?><NCC>
|
    <Request name="Beer"/>
|
</NCC>"""
```

# Просто переписали тест

- Реализация

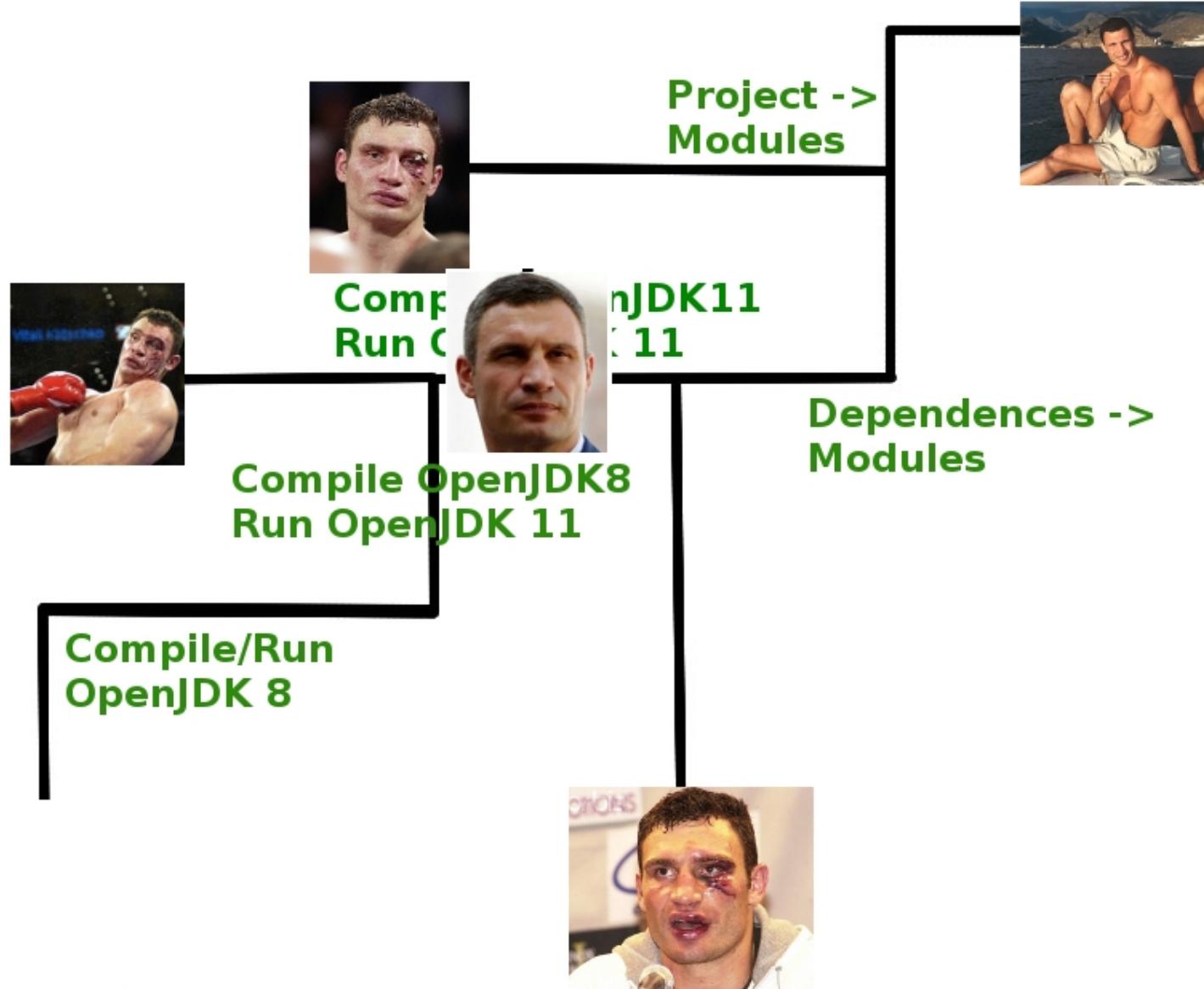
```
Transformer transformer = factory.newTransformer();
transformer.setOutputProperty(OutputKeys.INDENT, "yes");
transformer.setOutputProperty(OutputKeys.METHOD, "xml");
transformer.setOutputProperty(OutputKeys.MEDIA_TYPE, "text/xml");
transformer.transform(source, target);
```

- Просто добавили новых пустых строк в тесте

# Чо, так просто?



# УОТ ТАК УОТ!



Осталось добавить модульность и чики-пики



# Два вопроса, которые нужно решить

- Как собрать модульное приложение с помощью Gradle?
- Как запустить приложение модульное приложение в Idea?

# Сборка модульного приложения с помощью Gradle

- Gradle не поддерживает Jigsaw
- Все плагины делают одно и то же

```
compileJava {  
    doFirst {  
        options.compilerArgs = [  
            '--module-path', classpath.asPath,  
            '--add-modules', 'ALL-MODULE-PATH'  
        ]  
        classpath = files()  
    }  
}
```

- Ну мы так и сделали, и все сразу же начало падать

# Кажется, что-то пошло не так...

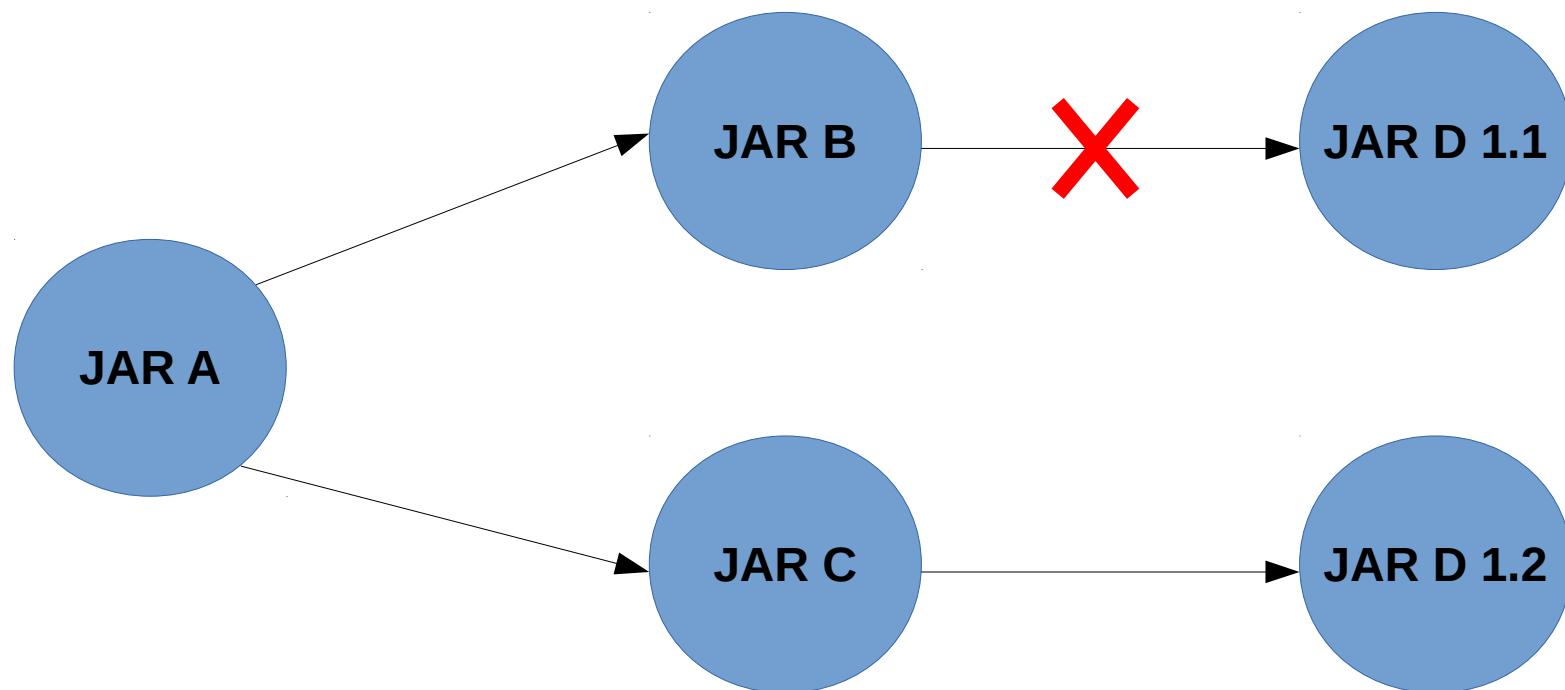


# Осторожно, в зависимостях есть СПЛИТ-пакеты!

```
> Task :fx:compileJava FAILED
error: module com.sun.istack.runtime reads package org.xml.sax from both java.xml and xml.apis
error: module com.sun.istack.runtime reads package org.w3c.dom from both java.xml and xml.apis
error: module com.sun.istack.runtime reads package javax.xml from both java.xml and xml.apis
error: module com.sun.istack.runtime reads package javax.activation from both java.activation and activation
error: module com.sun.istack.runtime reads package org.apache.commons.logging from both jcl.over.slf4j and spring.jcl
error: module com.sun.istack.runtime reads package javax.annotation from both java.annotation and tomcat.annotations.api
error: module com.sun.istack.runtime reads package org.w3c.dom.html from both xml.apis and xercesImpl
error: module com.sun.istack.runtime reads package org.apache.commons.logging from both jcl.over.slf4j and commons.logging
error: module com.sun.istack.runtime reads package org.apache.commons.logging.impl from both jcl.over.slf4j and commons.logging
error: module com.sun.istack.runtime reads package javax.annotation from both java.annotation and jsr305
error: module java.xml.bind reads package javax.activation from both java.activation and activation
error: module java.xml.bind reads package org.apache.commons.logging from both jcl.over.slf4j and spring.jcl
error: module java.xml.bind reads package javax.annotation from both java.annotation and tomcat.annotations.api
error: module java.xml.bind reads package org.w3c.dom.html from both xml.apis and xercesImpl
error: module java.xml.bind reads package javax.annotation from both java.annotation and jsr305
error: module java.xml.bind reads package javax.xml from both xml.apis and java.xml
error: module java.xml.bind reads package org.w3c.dom from both xml.apis and java.xml
error: module java.xml.bind reads package org.xml.sax from both xml.apis and java.xml
error: module org.jvnet.staxex reads package org.apache.commons.logging from both jcl.over.slf4j and spring.jcl
error: module org.jvnet.staxex reads package javax.annotation from both java.annotation and tomcat.annotations.api
error: module org.jvnet.staxex reads package org.w3c.dom.html from both xml.apis and xercesImpl
error: module org.jvnet.staxex reads package javax.xml from both xml.apis and java.xml
error: module org.jvnet.staxex reads package org.w3c.dom from both xml.apis and java.xml
error: module org.jvnet.staxex reads package org.xml.sax from both xml.apis and java.xml
error: module org.jvnet.staxex reads package org.apache.commons.logging from both jcl.over.slf4j and commons.logging
```

# И что с ними делать?

- Исключать из зависимостей

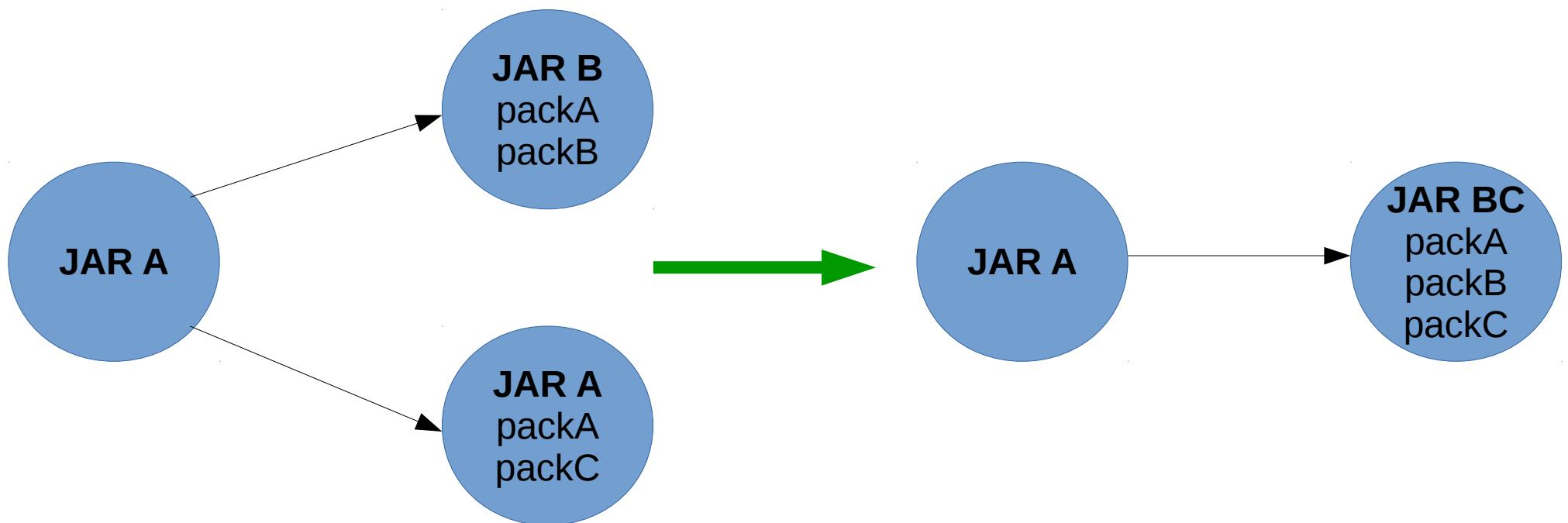


# Исключаем из зависимостей

```
compile("org.springframework.boot", "spring-boot-starter-websocket") {  
    exclude("javax.annotation", "javax.annotation-api")  
}  
  
compile("xalan", "xalan") {  
    exclude("xml-apis", "xml-apis")  
}  
  
compile("com.github.spotbugs", "spotbugs-annotations") {  
    exclude("com.google.code.findbugs", "jsr305")  
}  
  
compile("org.springframework.boot", "spring-boot-starter-test") {  
    exclude("org.hamcrest", "hamcrest-library")  
    exclude("org.hamcrest", "hamcrest-core")  
}
```

# А если нужны разные классы из одного сплит-пакета?

- Ну давайте их схлопнем в один и его импортируем



# Даже есть такой плагин

## **build.gradle**

```
plugins {  
    ...  
    id 'com.github.johnrengelman.shadow' version '4.0.3'  
}  
  
project(':hamcrest-shadow') {  
    dependencies {  
        implementation group: 'org.hamcrest', name: 'hamcrest-library', version:  
        '1.3'  
        implementation group: 'org.hamcrest', name: 'hamcrest-core', version:  
        '1.3'  
    }  
}
```

## **settings.gradle**

```
include("hamcrest-shadow")
```

Жесть, не так ли?



# Naumen PMS

- Выросло из бандлов OSGi
- Модульное приложение
- Около 60 модулей!
- *И тоже есть сплит-пакеты! И что щас делать?*

# Звать брата!



## `fx` VS. `fxlib`

- `fxlib` отпачковался от `fx`
- в обоих есть пакет `ru.naumen.fx.util`
- нельзя просто так взять и перенести `ru.naumen.fx.util` в `fx` или `fxlib`, потому что в нем лежит SPI! А это публичный интерфейс(

# Ну предположим, мы все-таки собрали приложение

- Как его запустить в Idea?
- Idea использует эвристику → запускает приложение как модульное, если есть хотя бы один `module-info.java`
- Да как так то?

# А еще у нас SpringBoot, и запускаем мы Main-класс

- Страна запуска

```
/home/vtarasov/work/java/jdk11/bin/java  
<keys>  
-m /path/to/project/starter.web/out/production/classes:  
/path/to/project/starter.web/out/production/resources:  
<dependencies>  
ru.naumen.fx.FxWebApplication
```

- Нет доступа к ресурсам из resources!

**Я устал — я ухожу**



# Выводы

- Оставаться на Oracle JDK 8 нет никакого смысла, если нет коммерческой поддержки
- Перейти на OpenJDK 8 очень просто
- Если хочется новые фичи, то нужно переходить на свежие LTS-версии
- Перейти на OpenJDK 11 не сложно
- В принципе, этого вполне достаточно, чтобы быть в тренде!

# Выводы

- Использовать модули Jigsaw в большом legacy-проекте очень не просто
- Нет хорошей поддержки со стороны сторонних библиотек
- Нет хорошей поддержки со стороны Gradle
- Нет хорошей поддержки со стороны Idea
- Остается ждать лучших времен, когда сообщество будет готово...

Ну и фиг с этой модульностью —  
все равно мы в тренде!



Спасибо за внимание  
Вопросы?