

Екатеринбург, 2020

Java 11: Золотой теленок

Тарасов Владимир
Naumen, Contact Center

Disclaimer

Все изложенные мысли являются личным мнением автора. Автор, Naumen и JUG.EKB не несут никакой ответственности за использование изложенного материала

Обо мне



Naumen, Contact Center

Java

Server-side

Naumen CRM, Naumen CC PMS

 @vannoying

План

Naumen CC PMS

Эволюция Java

Переход на OpenJDK

Проект Jigsaw

Модульный Naumen CC PMS

План

Naumen CC PMS

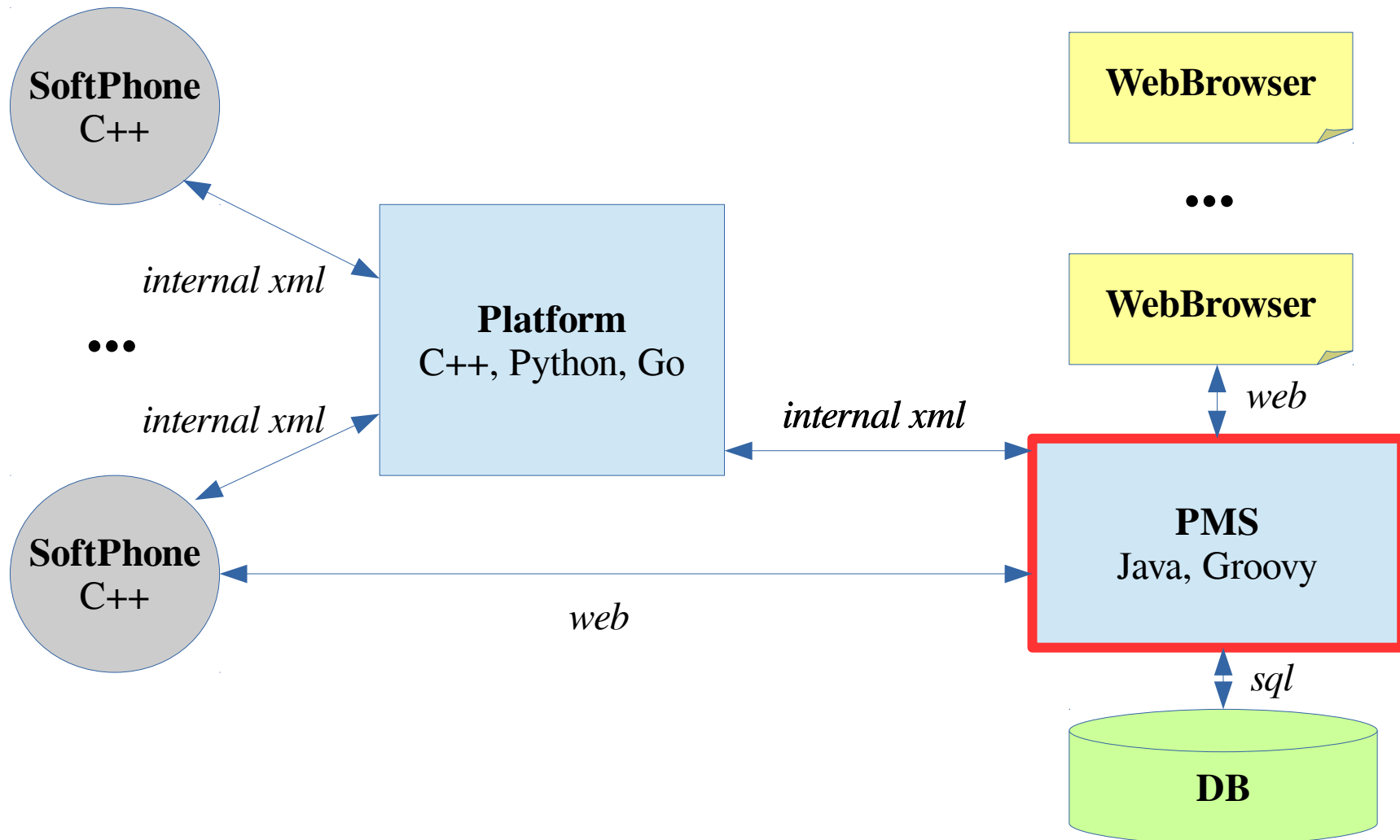
Эволюция Java

Переход на OpenJDK

Проект Jigsaw

Модульный Naumen CC PMS

Архитектура Naumen Contact Center



Задачи Naumen CC PMS

Конфигурация контактного центра

Настройка и отображение анкет оператору

Отчетность

Технологический стек Naumen CC PMS

Платформа Oracle JDK 8

Фреймворк SpringBoot

ORM Hibernate

Кодогенерация Lombok

Скрипты Groovy

Запросы SQL

Технологический стек Naumen CC PMS

Платформа **Oracle JDK 8**

Фреймворк SpringBoot

ORM Hibernate

Кодогенерация Lombok

Скрипты Groovy

Запросы SQL

Изменение лицензии Oracle JDK

~~Oracle Bytecode Licence~~

The screenshot shows the Oracle JDK 8 Downloads page. At the top, there's a navigation bar with the Oracle logo, a search bar containing the text "Ask 'Where do independent analysts rank Oracle's software?'" and links for "View Accounts" and "Try Oracle Cloud Free Tier". Below the navigation bar, the breadcrumb "Java / Technologies /" is visible, followed by buttons for "Java SE Downloads" and "Java SE Subscriptions". The main heading is "Java SE Development Kit 8 Downloads". The text below states: "Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language." and "The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform." A prominent green box contains the "Important Oracle JDK License Update" section, which states: "The Oracle JDK License has changed for releases starting April 16, 2019." It explains that the new "Oracle Technology Network License Agreement for Oracle Java SE" is substantially different from prior licenses, permitting personal and development use at no cost, but other uses may no longer be available. It also mentions that commercial license and support are available with a "Java SE Subscription" and that the latest OpenJDK release is available under the "GPL License" at jdk.java.net. Below the green box, the "See also:" section lists links to the "Java Developer Newsletter", "Java Developer Day hands-on workshops (free) and other events", "Java Magazine", and "JDK 8u241 checksum".

oracle.com/java/technologies/javase/javase-jdk8-downloads.html

Java / Technologies /

Java SE Downloads

Java SE Subscriptions

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Important Oracle JDK License Update

The Oracle JDK License has changed for releases starting April 16, 2019.

The new [Oracle Technology Network License Agreement for Oracle Java SE](#) is substantially different from prior Oracle JDK licenses. The new license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. An FAQ is available [here](#).

Commercial license and support is available with a low cost [Java SE Subscription](#).

Oracle also provides the latest OpenJDK release under the open source [GPL License](#) at jdk.java.net.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)
- [JDK 8u241 checksum](#)

Решения проблемы

Продолжать использовать Oracle JDK 8 и платить за Oracle Technology Network License

Переходить на другую JDK

Решения проблемы

Продолжать использовать Oracle JDK 8 и платить за Oracle Technology Network License

Переходить на другую JDK

План

Naumen CC PMS

Эволюция Java

Переход на OpenJDK

Проект Jigsaw

Модульный Naumen CC PMS

Java SE 8

С марта 2014

НОВЫЕ ВОЗМОЖНОСТИ

Default interface methods

Lambda

Optional

Stream API

Java Time API

Repeatable annotations

Последний релиз старого релизного цикла

Начиная с Java SE 9

Смена релизного цикла на полугодовой (март, сентябрь)

Появление LTS версий

Версии Java SE

Java SE 9 (сентябрь 2017 — март 2018)

Java SE 10 (март 2018 — сентябрь 2018)

Java SE 11 LTS (сентябрь 2018 — сентябрь 2022)

Java SE 12 (март 2019 — сентябрь 2019)

Версии Java SE

Java SE 9 (сентябрь 2017 — март 2018)

Java SE 10 (март 2018 — сентябрь 2018)

Java SE 11 LTS (сентябрь 2018 — сентябрь 2022)

Java SE 12 (март 2019 — сентябрь 2019)

НОВЫЕ ВОЗМОЖНОСТИ Java 9 — 11

Project Amber (var — delivered)

Unsafe → Variable Handles

REPL JShell

Compact Strings

HTTP Client API

Flight Recorder

Project Jigsaw

НОВЫЕ ВОЗМОЖНОСТИ Java 9 — 11

Project Amber (var — delivered)

Unsafe → Variable Handles

REPL JShell

Compact Strings

HTTP Client API

Flight Recorder

Project Jigsaw

Изменение лицензии Oracle JDK

Oracle Technology Network License с Oracle JDK 11 и выше
(Oracle JDK 8 с 16 апреля 2019)

Java SE

Java EE

Java ME

Java SE Subscription

Java Embedded

Java Card

Java TV

Community

Java Magazine

Overview

Downloads

Documentation

Community

Technologies

Training

Note: In the event of a discrepancy between any part of this FAQ and the license under which you receive Oracle software, the latter shall be considered correct. This FAQ pertains to Oracle Java SE releases starting April 16, 2019.

Oracle Java SE Licensing FAQ

What does the licensing change mean for me?

The previous Oracle Java SE license model had several options – some free under the [Binary Code License \("BCL"\)](#) and some paid under Oracle commercial terms. To simplify and provide full licensing transparency and clarity, Oracle as of Java 9 provides two distinct Java releases:

- Oracle OpenJDK releases under the open source [GNU General Public License v2, with the Classpath Exception \(GPLv2+CPE\)](#) (since Java 9), and
- Oracle Java SE product releases, which includes the Oracle JDK for Java 8 and later, and Oracle JRE with Java Web Start in Java 8, under the [OTN License Agreement for Java SE](#). This license permits personal use, development, testing, prototyping, demonstrating and some other uses at no cost.

Java SDKs and Tools

Java SE

Java EE and Glassfish

Java ME

Java Card

NetBeans IDE

Java Mission Control

Java Resources

Java APIs

Technical Articles

Demos and Videos

Forums

Java Magazine

Developer Training

Альтернативы

Corretto (Amazon)

GraalVM (Oracle)

Liberica (BellSoft)

OpenJDK (AdoptOpenJDK)

OpenJDK (openjdk.java.net)

Zulu (Azul Systems)

...

Альтернативы

Corretto (Amazon)

GraalVM (Oracle)

Liberica (BellSoft)

OpenJDK (AdoptOpenJDK)

OpenJDK (openjdk.java.net)

Zulu (Azul Systems)

...

Important Oracle JDK License Update

The Oracle JDK License has changed for releases starting April 16, 2019.

The new [Oracle Technology Network License Agreement for Oracle Java SE](#) is substantially different from the previous license for personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK license are no longer permitted. If you are downloading and using this product. An FAQ is available [here](#).

Commercial license and support is available with a low cost [Java SE Subscription](#).

Oracle also provides the latest OpenJDK release under the open source [GPL License](#) at [jdk.java.net](#).

Проект OpenJDK

Существует с 2006 года

Огромное сообщество

Присутствует в Linux дистрибутивах

Реализует спецификации Java

Протестирован ТСК

Лицензия GPL v2 + the Classpath exception

Oracle JDK расширяет → OpenJDK

<https://openjdk.java.net/>

План

Naumen CC PMS

Эволюция Java

Переход на OpenJDK

Проект Jigsaw

Модульный Naumen CC PMS

Переход на OpenJDK 8

Скрипты .sh

```
wget "http://link.to.oracle.jdk8.rpm"  
sudo yum localinstall jdk-8.rpm  
rm jdk-8.rpm
```

JAVA_HOME

```
JAVA_HOME="/usr/java/latest"
```

Переход на OpenJDK 8

Скрипты .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
```

JAVA_HOME

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

Переход на OpenJDK 8

Сборка...

Запуск...

Работа...

Переход на OpenJDK 8

Сборка... **Ok**

Запуск...

Работа...

Переход на OpenJDK 8

Сборка... **Ok**

Запуск... **Ok**

Работа...

Переход на OpenJDK 8

Сборка... **Ok**

Запуск... **Ok**

Работа... **Ok**

Переход на OpenJDK 11

Long-Term Support до 2022 года

Новые фишки JLS, JVM, JDK

<https://openjdk.java.net/projects/jdk/9/>

<https://openjdk.java.net/projects/jdk/10/>

<https://openjdk.java.net/projects/jdk/11/>

Переход на OpenJDK 11

Миграция с JDK 8 на JDK 9 — 11 описаны в различных инструкциях, в том числе и от Oracle

<https://docs.oracle.com/javase/9/migrate/toc.htm#JSMIG-GUID-7744EF96-5899-4FB2-B34E-86D49B2E89B6>

<https://docs.oracle.com/javase/10/migrate/toc.htm#JSMIG-GUID-7744EF96-5899-4FB2-B34E-86D49B2E89B6>

<https://docs.oracle.com/en/java/javase/11/migrate/migration-guide.pdf>

Общий план миграции с OpenJDK 8 на OpenJDK 11

Компиляция на OpenJDK 8, запуск на OpenJDK 11

Компиляция и запуск на OpenJDK 11

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Скрипты .sh

```
sudo yum install java-1.8.0-openjdk-devel
# compile application
sudo yum remove java-1.8.0-openjdk-devel
sudo yum install java-11-openjdk-devel
# start application
```

JAVA_HOME

```
JAVA_HOME="/usr/lib/jvm/jre-1.8.0"
# compile application
JAVA_HOME="/usr/lib/jvm/уjre-11"
# start application
```

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск...

Работа...

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Failed**

Работа...

JVM не стартовала

Ошибка

Unrecognized VM option '**PrintGCTimeStamps**'

Error: Could not create the Java Virtual Machine.

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

Старт JVM

```
java ... -XX:+PrintGCDetails -XX:+PrintGCTimeStamps ...  
ru.naumen.fx.FxWebApplication
```

JEP 158: Unified JVM Logging

JEP 271: Unified GC Logging

Появились 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

Запуск JVM

```
java ... -XX:+PrintGCDetails -XX:+PrintGCTimeStamps ...  
ru.naumen.fx.FxWebApplication
```

```
java ... -Xlog:gc* ... ru.naumen.fx.FxWebApplication
```

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск...

Работа...

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Failed**

Работа...

JVM стартовала, но приложение упало

Ошибка

```
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]K
```


JEP 320: Remove the Java EE and CORBA Modules

монолитный `rt.jar` распилен на модули

метамодуль `java.se.ee`

OpenJDK 9 и 10 → @Deprecated

OpenJDK 11 → удален

удаленные Java EE модули можно скачать с Maven Central

JDK 6
JDK 7
JDK 7 Updates
JDK 8
JDK 8 Updates
JDK 9

Since standalone versions of the Java EE technologies are readily available from third-party sites, such as Maven Central, there is no need for the Java SE Platform or the JDK to include them.

Добавляем зависимости на `xml.bind`

В файле сборки `build.gradle.kts`

```
...  
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")  
}  
...
```

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск...

Работа...

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Ok**

Работа...

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Ok**

Работа... **Warning**

Проблемы с локализацией

даты Java 8

20.05.17

даты Java 11

20.05.2017

проценты Java 8

10.7%

проценты Java 11

10.7 %

валюта Java 8

12 руб., 12 USD, 12 EUR

валюта Java 11

12 ₽, 12 \$, 12 €

JEP 252: Use CLDR Locale Data by Default

Появился → Java 8

По умолчанию → Java 9

CLDR (Common Locale Data Repository) — проект консорциума Unicode, обеспечивающий языковые настройки в XML

Старт JVM (режим обратной совместимости)

```
java ... -Xlog:gc* ... ru.naumen.fx.FxWebApplication
```

```
java ... -Xlog:gc* -Djava.locale.providers=COMPAT ...  
ru.naumen.fx.FxWebApplication
```

МЫ РАЗОЗЛИЛИ ВРЕМЯ,
И ТЕПЕРЬ У НАС ВСЕГДА
ПОЛГОДА ДО ВЫХОДА
НОВОЙ ВЕРСИИ
JAVA



Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Ok**

Работа...

Сборка на OpenJDK 8 → запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Ok**

Работа... **Ok**

Сборка и запуск на OpenJDK 11

Сборочные файлы *.gradle.kts

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

Скрипты .sh

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

JAVA_HOME

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

Сборка и запуск на OpenJDK 11

Сборка...

Запуск...

Работа...

Сборка и запуск на OpenJDK 11

Сборка... **Failed**

Запуск...

Работа...

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 XMLStreamWriter

`com.sun.xml.internal.stream.writers.XMLStreamWriterImpl`

Было

```
fStreamWriter.writeStartDocument(  
    startDocument.getCharacterEncodingScheme(),  
    startDocument.getVersion());
```

Стало

```
fStreamWriter.writeStartDocument(  
    startDocument.getCharacterEncodingScheme(),  
    startDocument.getVersion(),  
    startDocument.isStandalone(),  
    startDocument.standaloneSet());  
standaloneSet() = (encoding == true)
```

Переписали тест → добавили `standalone=no` в ожидаемые xml

Сборка и запуск на OpenJDK 11

Сборка...

Запуск...

Работа...

Сборка и запуск на OpenJDK 11

Сборка... **Failed**

Запуск...

Работа...

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> ""
```

Изменилась реализация

`groovy.xml.XmlUtil.serialize(...)`

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

Сборка и запуск на OpenJDK 11

Сборка...

Запуск...

Работа...

Сборка и запуск на OpenJDK 11

Сборка... **Ok**

Запуск...

Работа...

Сборка и запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Ok**

Работа...

Сборка и запуск на OpenJDK 11

Сборка... **Ok**

Запуск... **Ok**

Работа... **Ok**

Итоги по переходу на OpenJDK 8

Подправили скрипты сборки и запуска на использование OpenJDK 8

Итоги по переходу на OpenJDK 11

Сборка

Подправили скрипты сборки на использование OpenJDK 11

Подправили `build.gradle.kts` на использование Java 11

Добавили в `build.gradle.kts` зависимость на `xml.bind`

Подправили пару тестов, в которых использовалась сериализация в XML

Запуск

Подправили скрипты запуска на использование OpenJDK 11

При старте JVM передаем новые ключи на логирование событий GC

При старте JVM передаем ключ совместимости локализации с предыдущими версиями

План

Naumen CC PMS

Эволюция Java

Переход на OpenJDK

Проект Jigsaw

Модульный Naumen CC PMS

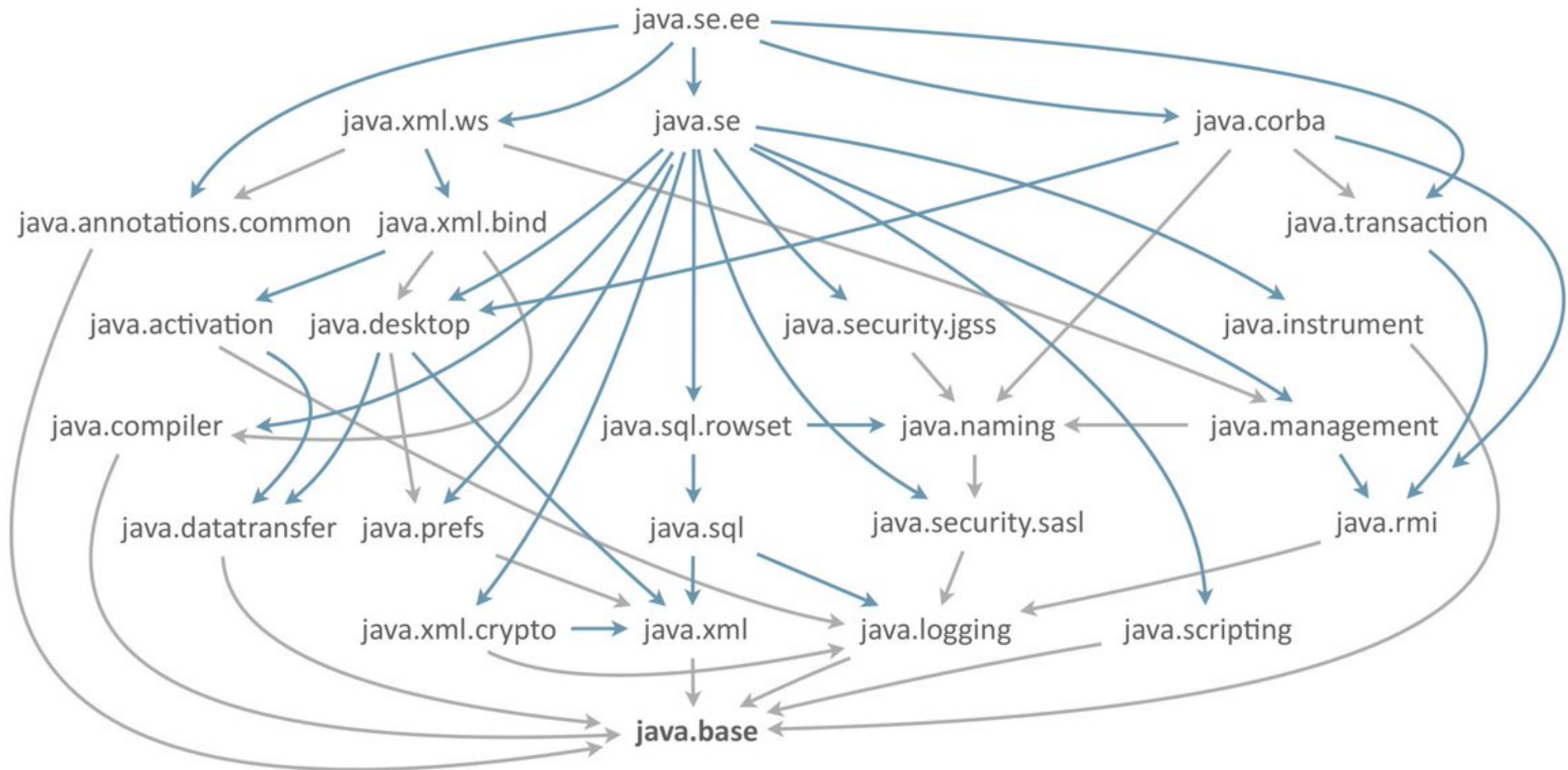
Проект Jigsaw

Modular Java Platform (JPMS) — разделение `rt.jar` на модули

Modular Java Application — возможность написания модульных приложений

Java Application Image with Jlink — возможность получения образа модульного приложения

Modular Java Platform



Modular Java Application

reliable configuration

зависимости между модулями разрешены при компиляции и при старте приложения

запрещены сплит-пакеты

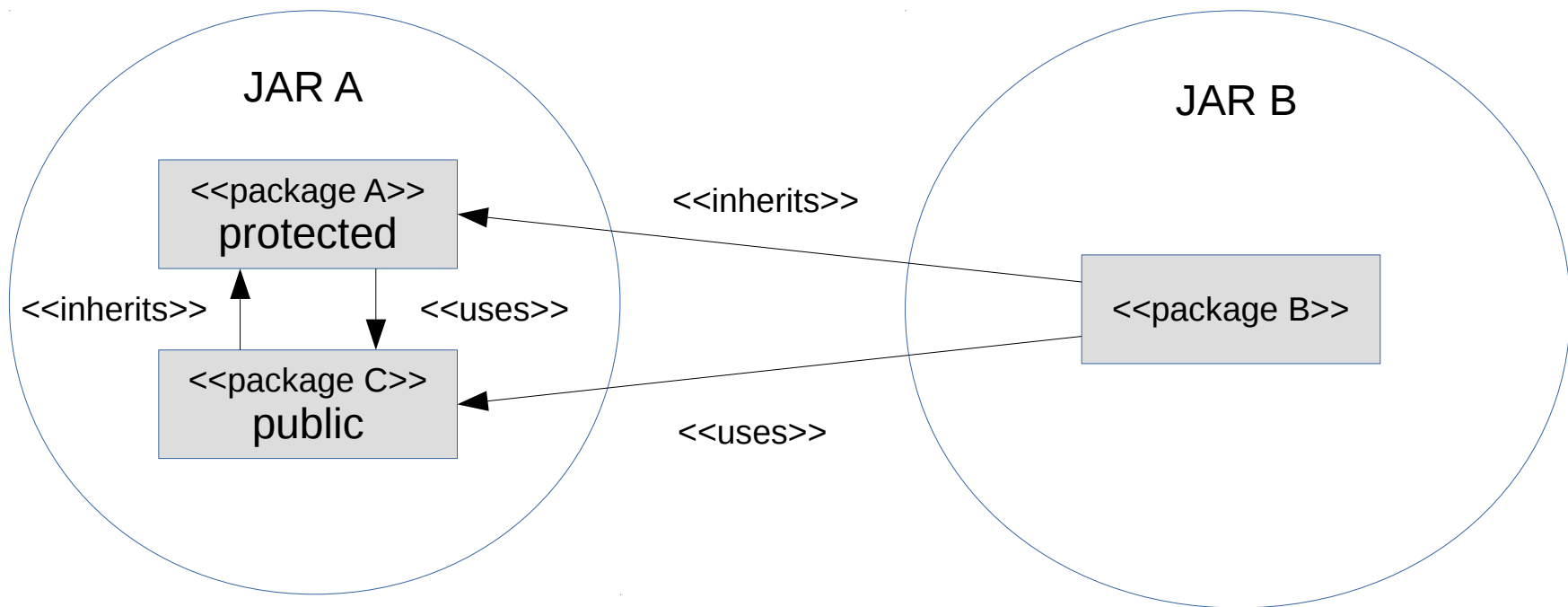
запрещены циклы в графе зависимостей

strong encapsulation

ограничение видимости пакетов

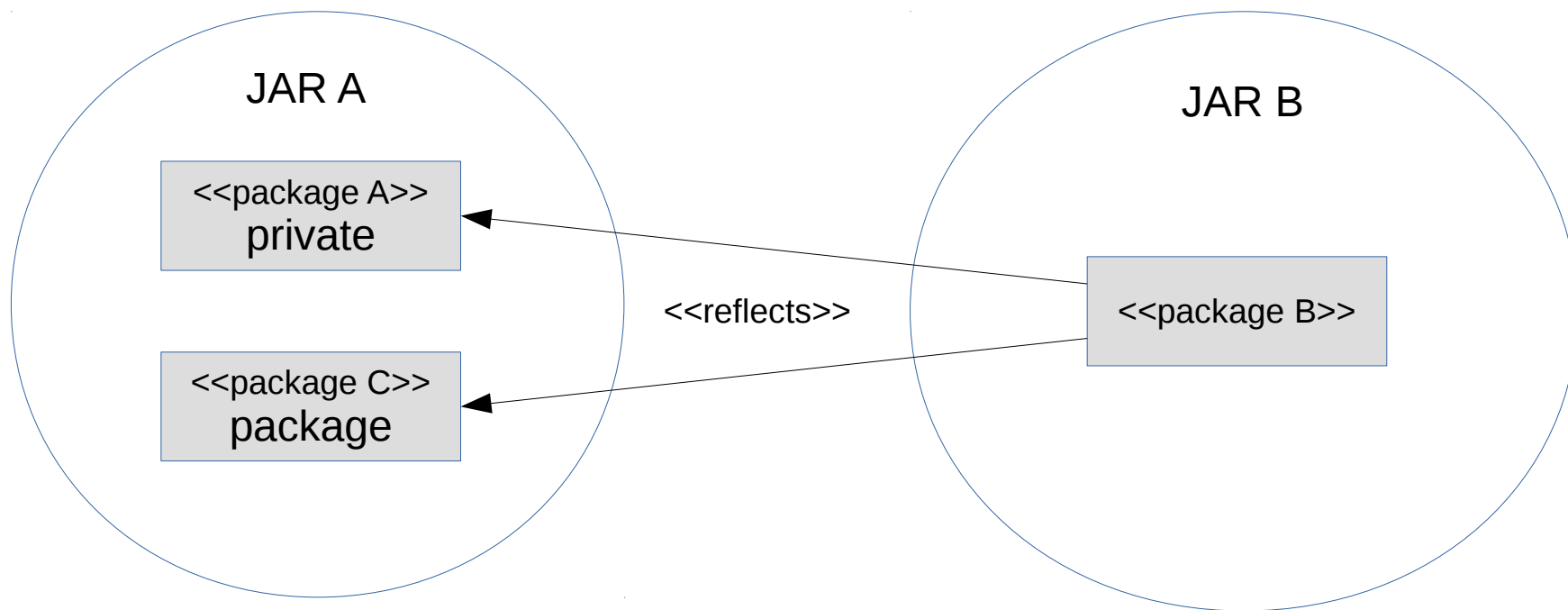
Модульность до Jigsaw

проблема `protected` и `public`



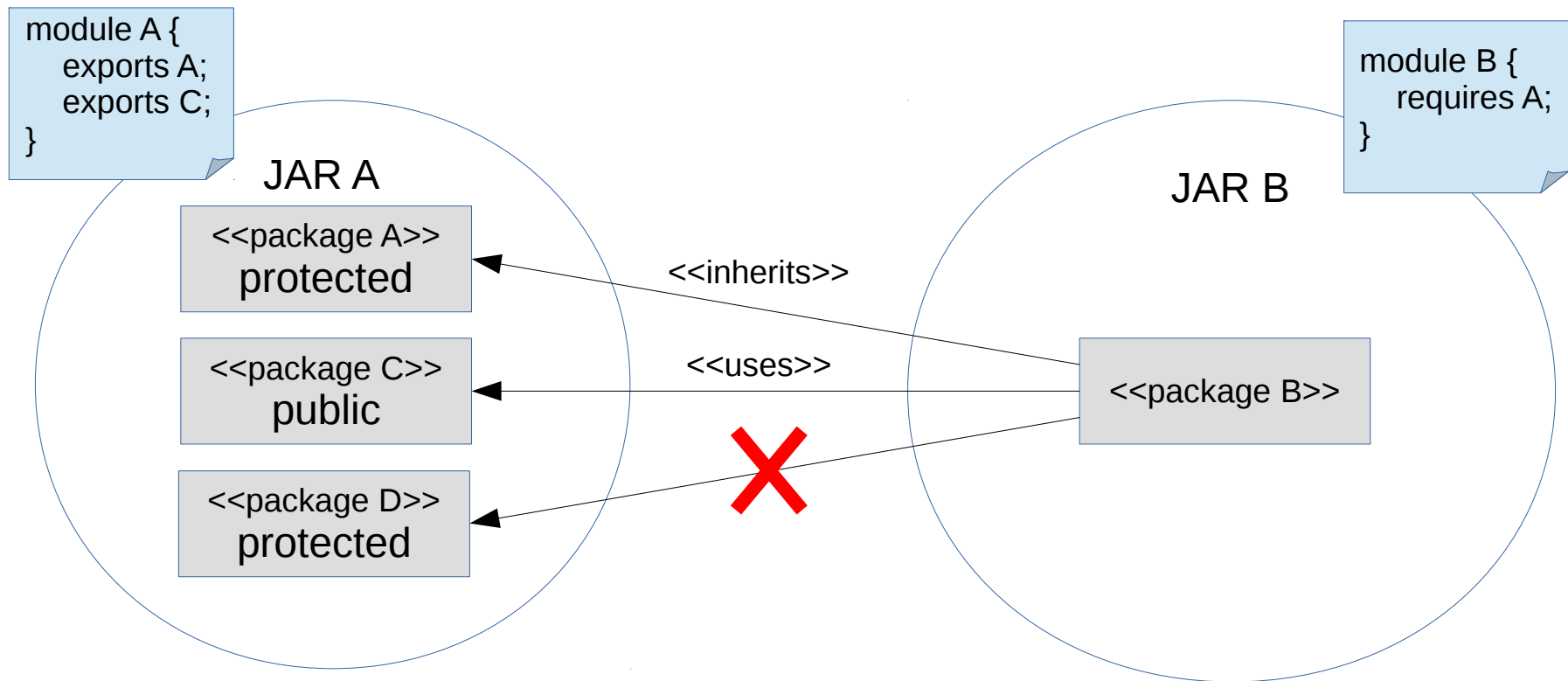
Модульность до Jigsaw

проблема package и private



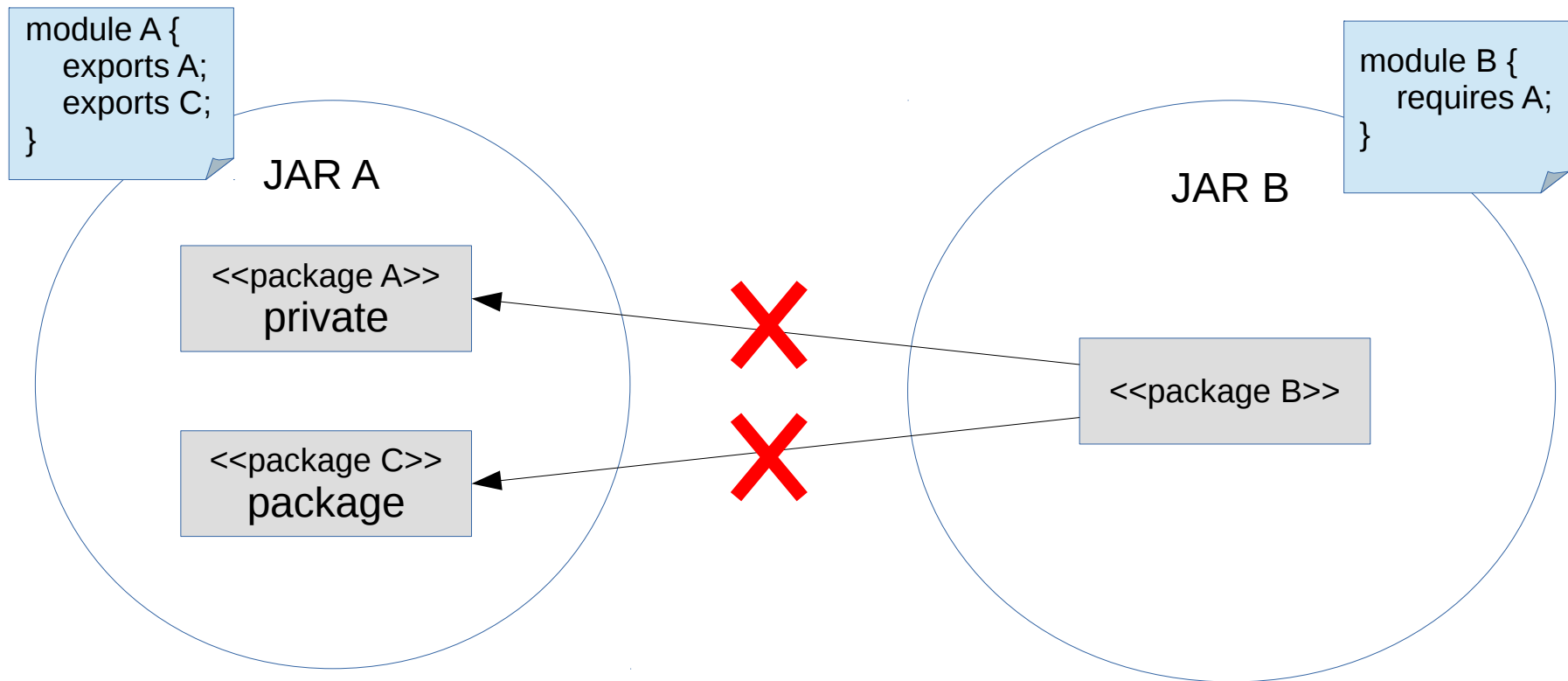
Модульность Jigsaw

нет проблем с `protected` и `public`



Модульность Jigsaw

нет проблем с package и private



Modular Java Application

javac & java

`--class-path → --module-path`

`--source-path → --module-source-path`

`java -jar --class-path <path_to_classes> app.jar →`

`java --module-path <path_to_modules> --module app`

расположение jar

`--class-path → unnamed module`

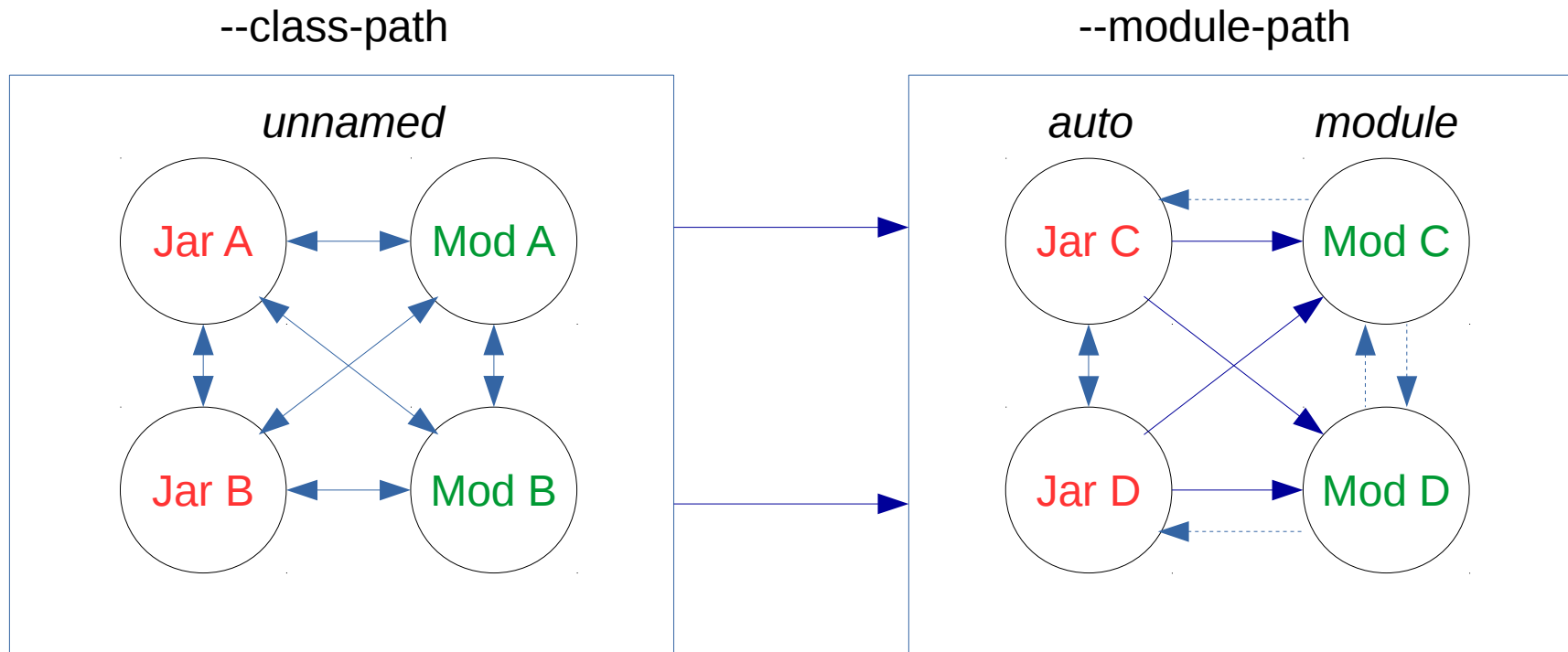
`--module-path → automodule (имя генерируется из названия jar)`

расположение модуля

`--class-path → unnamed module`

`--module-path → module`

Modular Java Application



Плюсы использования Jigsaw

Инкапсуляция уровня модулей

Улучшение архитектуры приложения (запрещены циклы и сплит-пакеты)

Решение проблем с JAR Hell

Более агрессивное разрешение зависимостей внутри и между модулями

Возможность создавать образы приложений

План

Naumen CC PMS

Эволюция Java

Переход на OpenJDK

Проект Jigsaw

Модульный Naumen CC PMS

Миграция приложения на модули

перевод компонентов приложения из `--class-path` на `--module-path`

```
java -jar --class-path  
lib/libA:lib/libB:app/appA:app/appB my.app.App
```

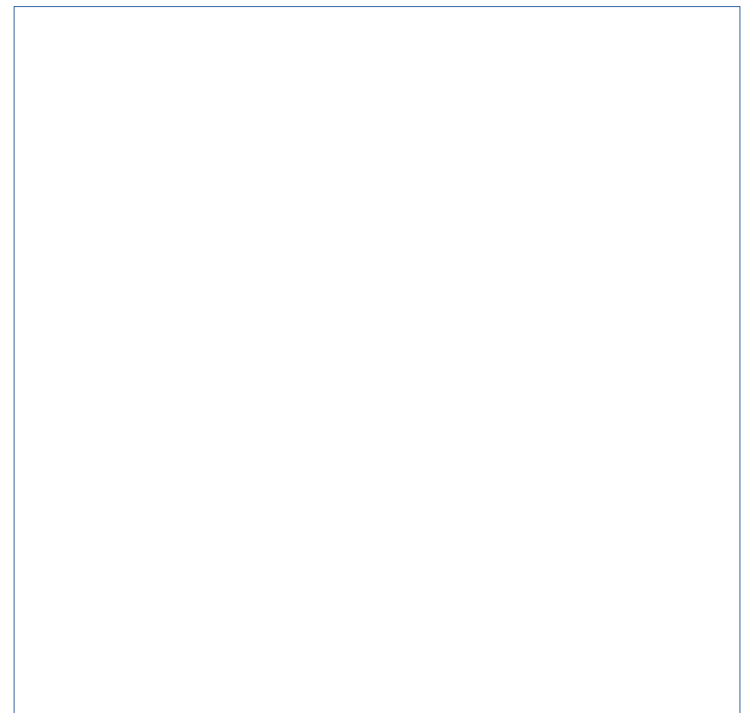
```
java --module-path lib:app --module appA/my.app.App
```

Миграция приложения на модули

--class-path



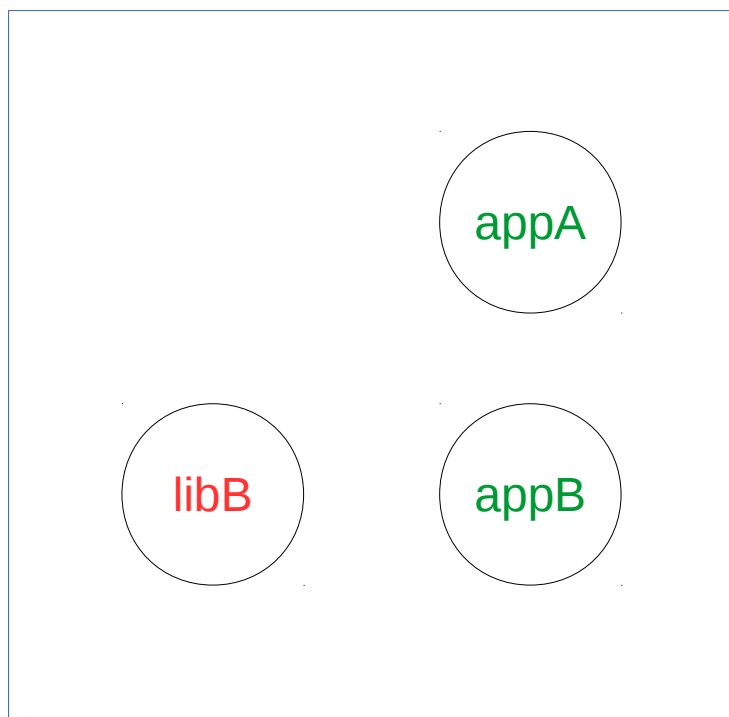
--module-path



Миграция приложения на модули

перенос библиотек из --class-path в --module-path

--class-path



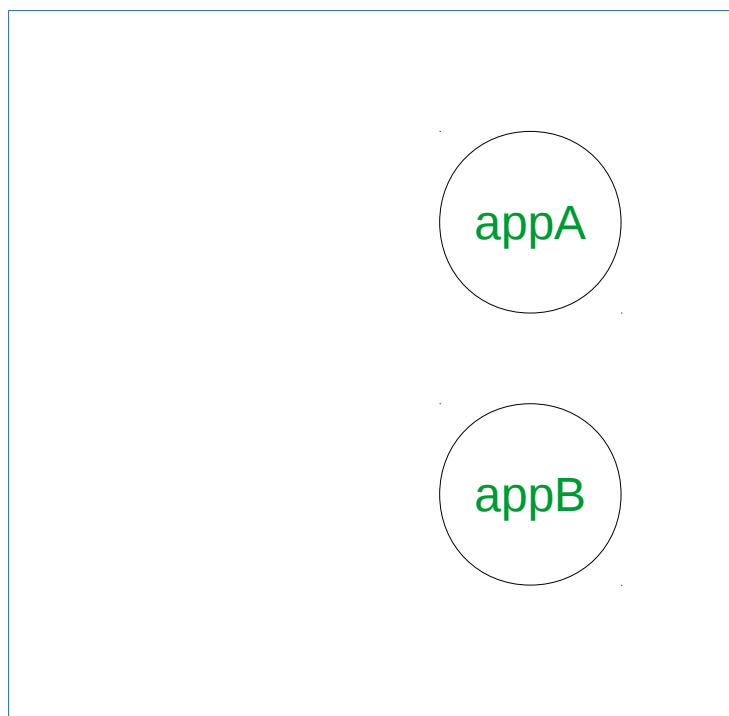
--module-path



Миграция приложения на модули

перенос библиотек из --class-path в --module-path

--class-path



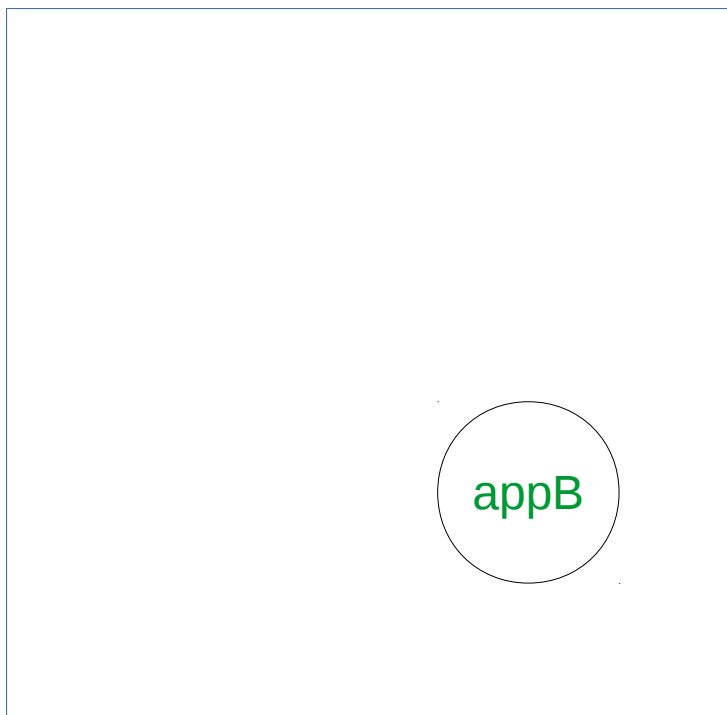
--module-path



Миграция приложения на модули

перенос компонентов приложения из --class-path в --module-path

--class-path



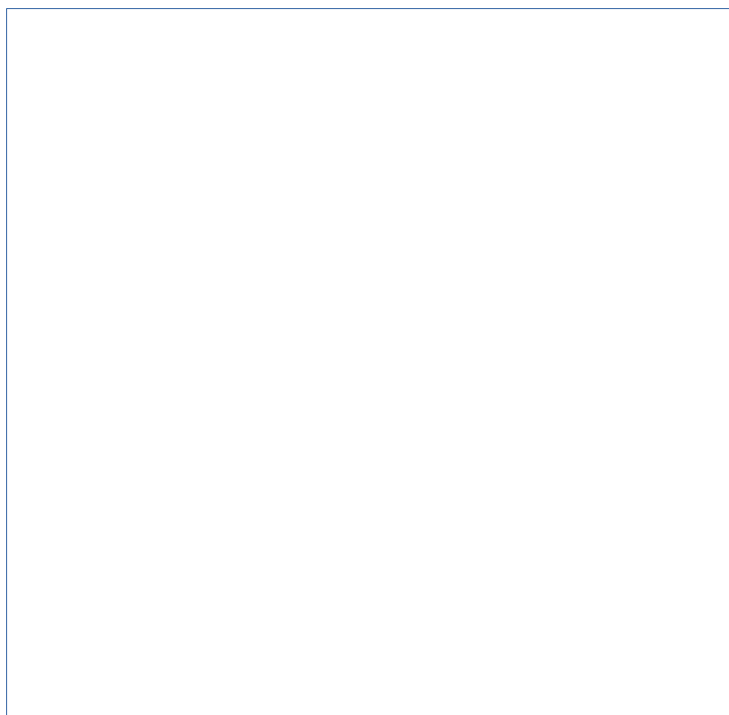
--module-path



Миграция приложения на модули

перенос компонентов приложения из `--class-path` в `--module-path`

`--class-path`



`--module-path`



Две задачи, которые необходимо решить

сборка и запуск модульного приложения в IntelliJ Idea

сборка и запуск модульного приложения с помощью Gradle

Сборка и запуск приложения в IntelliJ Idea

очень слабая поддержка модульности

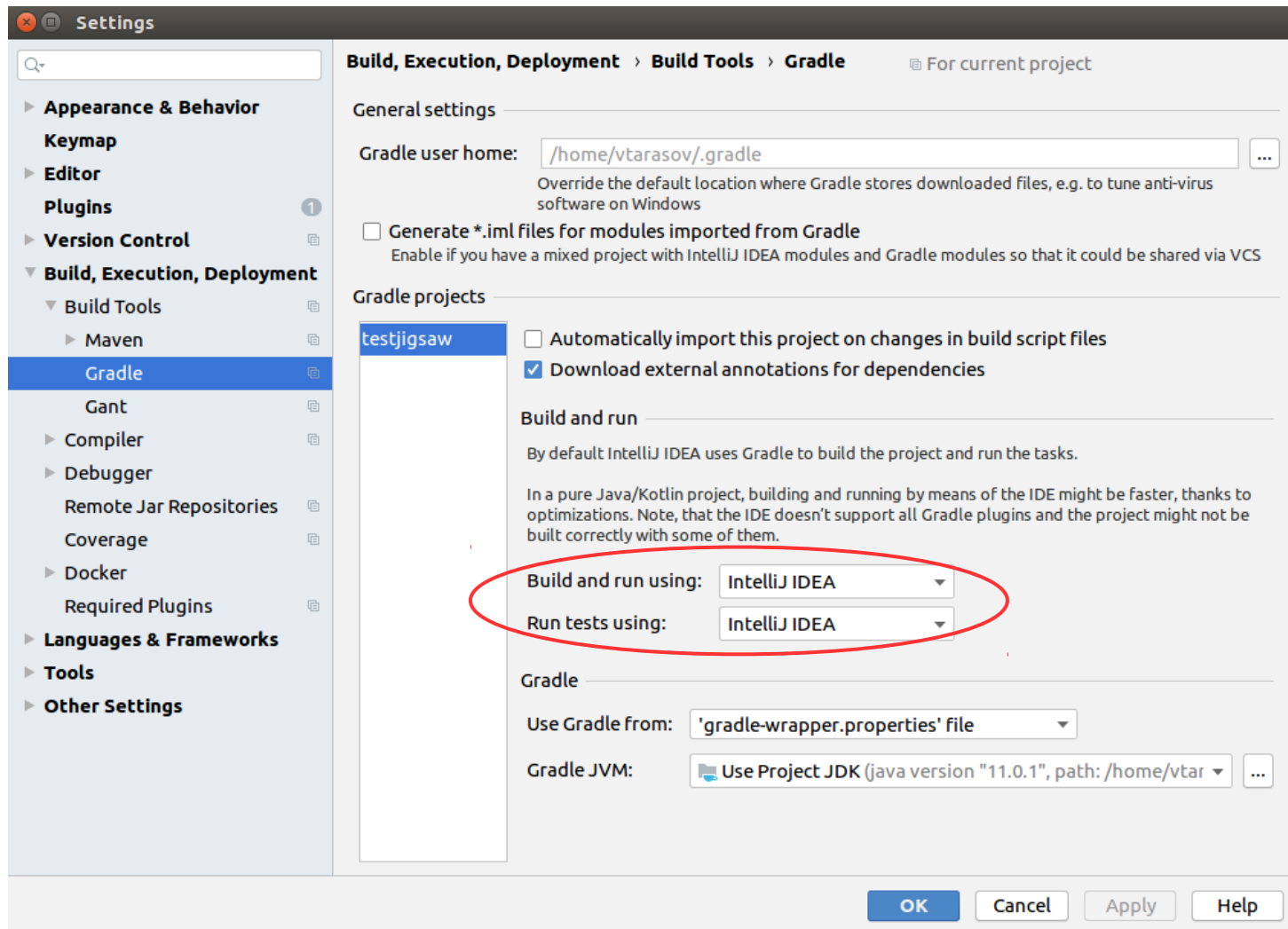
можно запускаться или только на `--class-path`, или только на `--module-path`

если запускаемся из модуля, то все компоненты будут в `--module-path`, иначе в `--class-path`

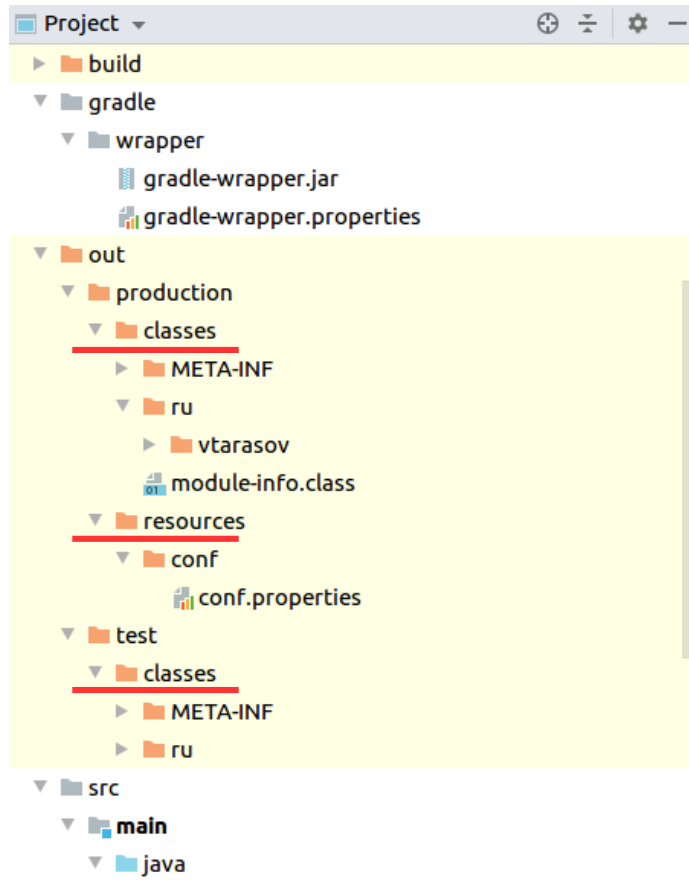
представление ресурсов компонента как отдельного модуля

представление тестов компонента как отдельного модуля

Сборка и запуск в IntelliJ Idea



Сборка и запуск в IntelliJ Idea



3 отдельных модуля

app/out/production/**classes**

app/out/production/**resources**

app/out/test/**classes**

```
java ... -p app/out/production/classes:app/out/production/resources:...  
-m app/ru.vtarasov.App
```

Сборка и запуск в IntelliJ Idea

проблемы

полноценный модуль только `app/out/production/classes`

нет доступа в ресурсам через

`ClassLoader.getResourceAsStream(String)`

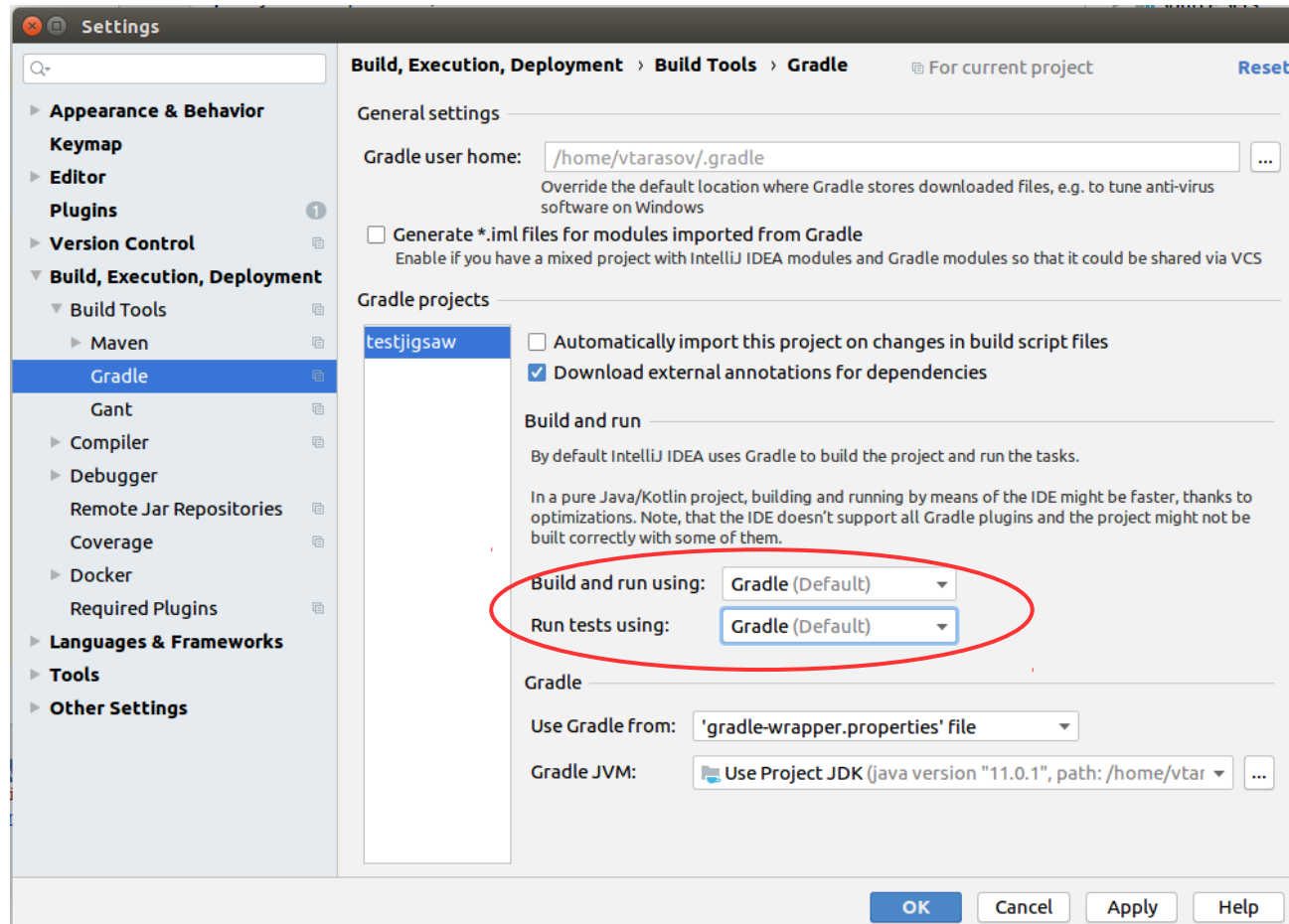
нет возможности запустить модульные тесты на `--module-path`

```
java ... -p app/out/production/classes:app/out/production/resources:...  
-m app/ru.vtarasov.App
```


Сборка и запуск в IntelliJ Idea

решение

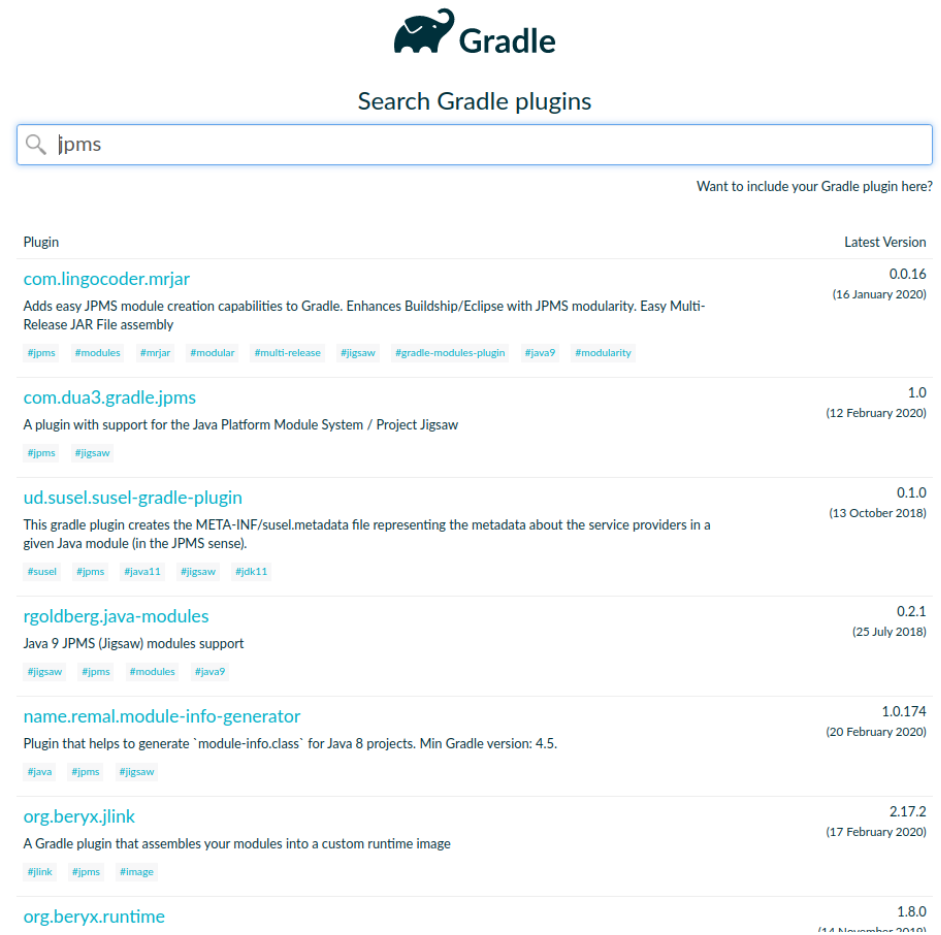
собираться и запускать через Gradle



Сборка и запуск через Gradle

нет поддержки модулей из коробки

существует много плагинов



Gradle

Search Gradle plugins

jpms

Want to include your Gradle plugin here?

Plugin	Latest Version
com.lingocoder.mrjar Adds easy JPMS module creation capabilities to Gradle. Enhances Buildship/Eclipse with JPMS modularity. Easy Multi-Release JAR File assembly #jpms #modules #mrjar #modular #multi-release #jigsaw #gradle-modules-plugin #java9 #modularity	0.0.16 (16 January 2020)
com.dua3.gradle.jpms A plugin with support for the Java Platform Module System / Project Jigsaw #jpms #jigsaw	1.0 (12 February 2020)
ud.susel.susel-gradle-plugin This gradle plugin creates the META-INF/susel.metadata file representing the metadata about the service providers in a given Java module (in the JPMS sense). #susel #jpms #java11 #jigsaw #jdk11	0.1.0 (13 October 2018)
rgoldberg.java-modules Java 9 JPMS (Jigsaw) modules support #jigsaw #jpms #modules #java9	0.2.1 (25 July 2018)
name.remal.module-info-generator Plugin that helps to generate `module-info.class` for Java 8 projects. Min Gradle version: 4.5. #java #jpms #jigsaw	1.0.174 (20 February 2020)
org.beryx.jlink A Gradle plugin that assembles your modules into a custom runtime image #jlink #jpms #image	2.17.2 (17 February 2020)
org.beryx.runtime	1.8.0 (14 November 2019)

Сборка и запуск через Gradle

2 плагина, в которых есть все, что нужно для модульного проекта

```
org.javamodularity.moduleplugin  
rgoldberg.experimental-jigsaw
```



org.javamodularity.moduleplugin

Plugin that makes it easy to work with the Java Platform Module System

<https://github.com/java9-modularity/gradle-modules-plugin>

#java #modules #jpms #modularity

Version 1.6.0 (latest)

Created 01 September 2019.

Plugin that makes it easy to work with the Java Platform Module System



rgoldberg.experimental-jigsaw

Experiment with Java 9 modules before they are officially supported.

<https://github.com/rgoldberg/gradle-java-modules>

#jpms #modules #jigsaw #java9 #java

Version 0.5-SNAPSHOT (latest)

Created 29 August 2019.

Experiment with Java 9 modules before they are officially supported.

Сборка и запуск через Gradle

`org.javamodularity.moduleplugin`

build.gradle

```
plugins {  
    ...  
    id("org.javamodularity.moduleplugin") version "1.6.0"  
}
```

settings.gradle

```
pluginManagement {  
    repositories {  
        ...  
        maven { url = uri("https://plugins.gradle.org/m2/") }  
    }  
}
```

Сборка и запуск через Gradle

`rgoldberg.experimental-jigsaw`

build.gradle

```
plugins {  
    ...  
    id("rgoldberg.experimental-jigsaw") version "0.5-SNAPSHOT"  
}
```

settings.gradle

```
pluginManagement {  
    repositories {  
        ...  
        maven { url = uri("https://plugins.gradle.org/m2/") }  
    }  
}
```

Сборка и запуск через Gradle

плагины делают примерно одно и то же (компиляция и запуск)

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

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

Сборка и запуск через Gradle

плагины делают примерно одно и то же (компиляция и запуск тестов)

```
compileTestJava {
    doFirst {
        options.compilerArgs = [
            '--module-path', classpath.asPath,
            '--add-modules', 'junit',
            '--add-reads', "app=junit",
            '--patch-module', "app=" +
                files(sourceSets.test.java.srcDirs).asPath,
        ]
        classpath = files()
    }
}

test {
    doFirst {
        jvmArgs = [
            '--module-path', classpath.asPath,
            '--add-modules', 'ALL-MODULE-PATH',
            '--add-reads', "app=junit",
            '--patch-module', "app=" +
                files(sourceSets.test.java.outputDir).asPath,
        ]
        classpath = files()
    }
}
```

Сборка и запуск через Gradle

плагины делают примерно одно и то же (процессинг ресурсов)

```
compileJava {
    doFirst {
        options.compilerArgs = [
            '--patch-module', "app=app/build/resources/main",
        ]
        classpath = files()
    }
}

java {
    doFirst {
        jvmArgs = [
            '--patch-module', "app=app/build/resources/main",
        ]
        classpath = files()
    }
}
```


Сборка и запуск через Gradle

запуск приложения

```
java ... --module-path  
...:app/build/classes/java/main:app/build/resources/main ...  
--patch-module app=app/build/resources/main --module  
app/ru.vtarasov.App
```

запуск тестов

```
java ... --module-path  
app/build/classes/java/test:app/build/resources/test:app/build/  
d/classes/java/main:app/build/resources/main:... --patch-  
module  
app=app/build/classes/java/test:app/build/resources/main:app/  
build/resources/test --add-modules ALL-MODULE-PATH --add-  
reads app=junit --add-opens app/ru.vtarasov=junit ...  
worker.org.gradle.process.internal.worker.GradleWorkerMain  
'Gradle Test Executor 1'
```

Сборка и запуск через Gradle

ВЫВОДЫ

через систему плагинов можно собирать, тестировать и запускать модульные приложения

без кастомизаций можно собираться, тестировать и запускаться или только на `--class-path`, или только на `--module-path`

Миграция Naumen CC PMS на модули

легаси приложение с 2005 года

выросло из бандлов OSGi (сейчас на SpringBoot)

МОНОЛИТ

около 60 компонентов (jar)

Миграция Naumen CC PMS на модули

основные проблемы при миграции

сплит-пакеты в сторонних библиотеках

сплит-пакеты в наших компонентах

циклические зависимости

Миграция Naumen CC PMS на МОДУЛИ

основные проблемы при миграции

сплит-пакеты в сторонних библиотеках

сплит-пакеты в наших компонентах

циклические зависимости

Сплит-пакеты в сторонних библиотеках

уже при сборке корневого модуля

```
> 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 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 org.jvnet.staxex reads package org.apache.commons.logging from both jcl.over.slf4j and spring.jcl
```

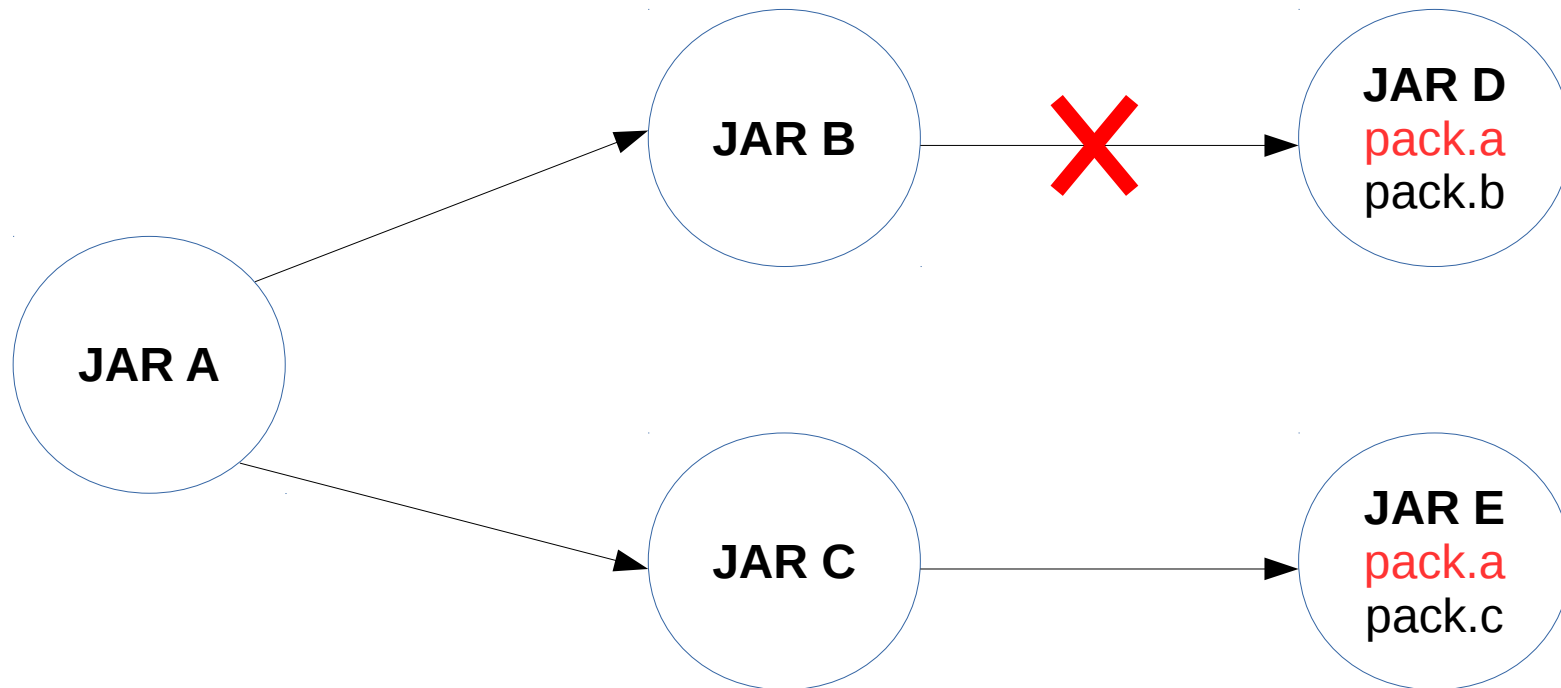
```
error: module org.jvnet.staxex reads package javax.annotation from both java.annotation and tomcat.annotations.api
```

```
error: module org.jvnet.staxex reads package org.w3c.dom.html from both xml.apis and xercesImpl
```

```
...
```

Сплит-пакеты в сторонних библиотеках

решение: исключать из зависимостей

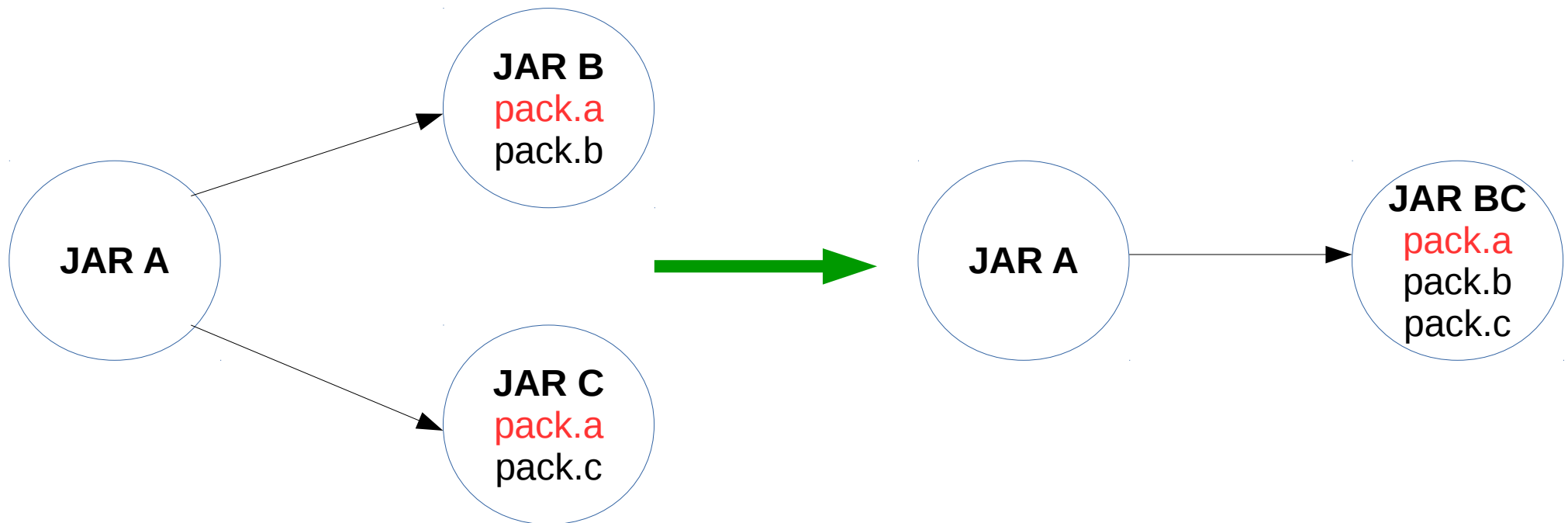


Сплит-пакеты в сторонних библиотеках

```
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")  
}
```


Сплит-пакеты в сторонних библиотеках

если нужны разные классы из одного сплит-пакета: схлапывание двух модулей в один



Сплит-пакеты в сторонних библиотеках

плагин для Gradle

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")
```

Сплит-пакеты в модулях приложения

вопрос

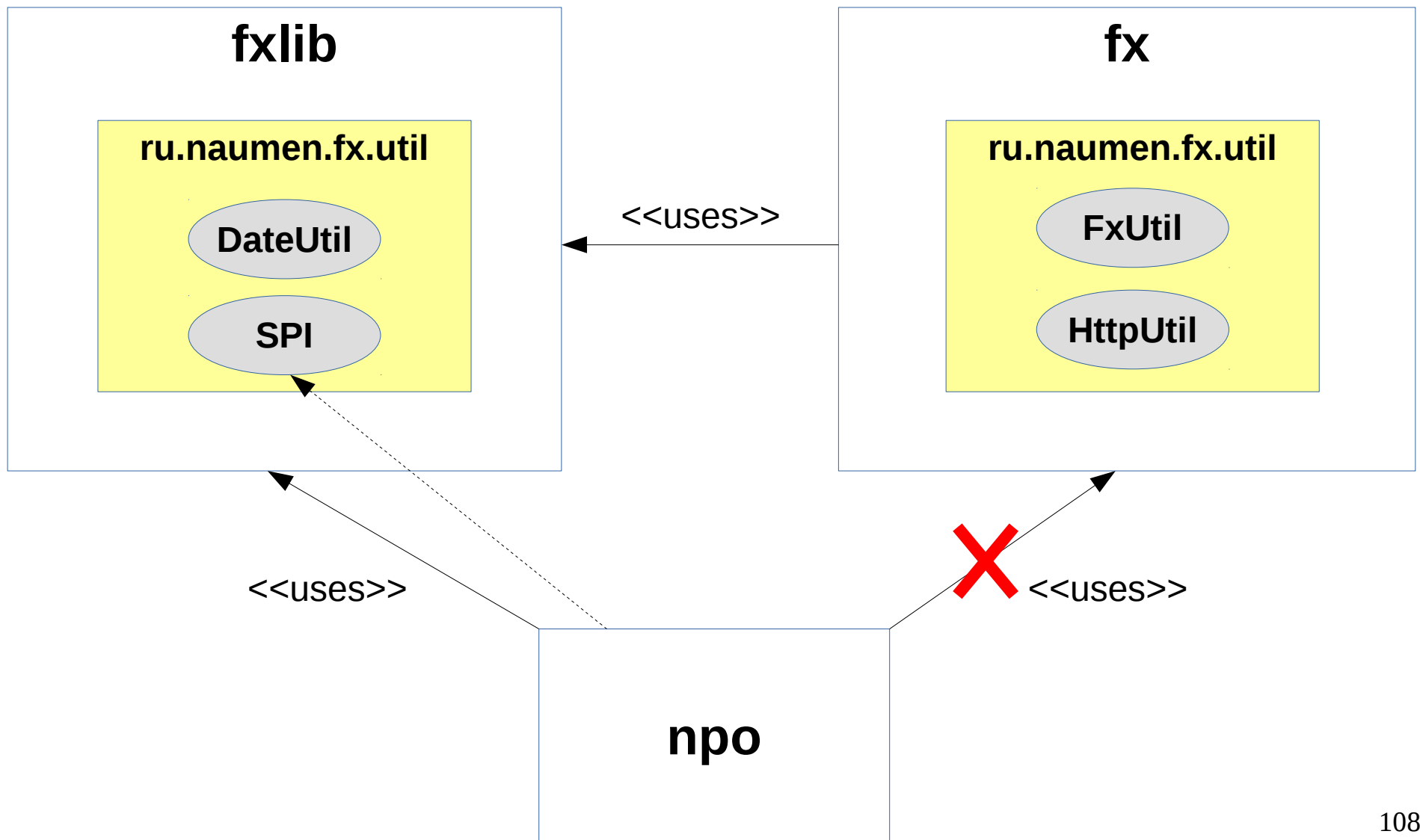
как вообще такое произошло?

решение

перемещение сплит-пакетов в один модуль

переименование сплит-пакета

Сплит-пакеты в модулях приложения



Сплит-пакеты в модулях приложения

нельзя просто так взять и перенести `ru.naumen.fx.util` в `fx` или `fxlib`, потому что сломаются зависимости от `fx` и `fxlib`

нельзя просто так взять и переименовать `ru.naumen.fx.util`, потому что он содержит публичные интерфейсы

Сплит-пакеты в модулях приложения

нельзя просто так взять и перенести `ru.naumen.fx.util` в `fx` или `fxlib`, потому что сломаются зависимости от `fx` и `fxlib`

нельзя просто так взять и переименовать `ru.naumen.fx.util`, потому что он содержит публичные интерфейсы

Миграция Naumen CC PMS на МОДУЛИ

ИТОГИ

перевести на модули возможно, и приложение будет работать

появилось огромное количество трудно читаемого кода, особенно в скриптах сборки

появилось дублирование кода

появились зависимости от тяжелых модулей

вместо улучшения архитектуры получили ее усложнение

Миграция Naumen CC PMS на МОДУЛИ

ИТОГИ

перевести на модули возможно, и приложение будет работать

появилось огромное количество трудно читаемого кода, особенно в скриптах сборки

появилось дублирование кода

появились зависимости от тяжелых модулей

вместо улучшения архитектуры получили ее усложнение

отказались от модульности

План

Naumen CC PMS

Эволюция Java

Переход на OpenJDK

Проект Jigsaw

Модульный Naumen CC PMS

Заключение

оставаться на Oracle JDK 8 нет никакого смысла, если нет коммерческой поддержки

перейти на OpenJDK 8 очень просто, если вы не зависите от внутренней реализации Oracle JDK 8

если хочется новые фишки, то лучше переходить на свежие LTS-версии

перейти на OpenJDK 11 не сложно, если вы не зависите от внутренней реализации Oracle JDK 11

Заключение

использовать модули Jigsaw в большом legacy-проекте не просто
есть проблемы с поддержкой со стороны сторонних библиотек
есть неплохая поддержка со стороны плагинов Gradle
нет хорошей поддержки со стороны Idea

Заключение

использовать модули Jigsaw в большом legacy-проекте не просто
есть проблемы с поддержкой со стороны сторонних библиотек
есть неплохая поддержка со стороны плагинов Gradle
нет хорошей поддержки со стороны Idea
остается ждать лучших времен, когда сообщество будет готово

Спасибо за внимание
Вопросы?