



**Luxoft**  
Компания

рейтинг  
106,62

Профиль

144  
Блог

0  
Вакансии

1,5k  
Подписчики

10 декабря 2015 в 14:38

## Разработка → Шпаргалка Java-программиста 5. Двести пятьдесят русскоязычных обучающих видео докладов и лекций о Java

Разработка веб-сайтов\*, Программирование\*, Java\*, Блог компании Luxoft

Думаю, мало кто будет спорить, что просмотр видео хороших лекций и докладов с конференций это один из самых быстрых и простых способов научиться чему-то новому. Проблема в том, что по Java сложно найти все хорошие видео конференций и доклады по нужной теме. Более того, по названию многих видео с конференций сложно понять, какой именно они теме посвящены.



Поэтому я подготовил данный сборник видео докладов на русском языке с различных конференций ([Joker](#), [JPoint](#), [JavaDays](#), [JEEConf](#), конечно, с [DEV labs](#), которые организывает Luxoft), и, естественно, видео из канала [Luxoft Training Center](#). Всё видео разделено на различные категории и при необходимости добавлено описание.

**Update:** Внимание, актуальная версия со всеми ссылками на видео находится теперь в моем github'e проекте [useful-java-links](#), по [этой ссылке](#).

▼ Общее оглавление 'Шпаргалок'

1. [JPA и Hibernate в вопросах и ответах](#)
2. [Триста пятьдесят самых популярных не мобильных Java opensource проектов на github](#)
3. [Коллекции в Java \(стандартные, guava, apache, trove, gs-collections и другие\)](#)
4. [Java Stream API](#)
5. [Двести пятьдесят русскоязычных обучающих видео докладов и лекций о Java](#)
6. [Список полезных ссылок для Java программиста](#)
7. [Типовые задачи](#)
  - 7.1 [Оптимальный путь преобразования InputStream в строку](#)
  - 7.2 [Самый производительный способ обхода Map'ы, подсчет количества вхождений подстроки](#)
8. [Библиотеки для работы с Json \(Gson, Fastjson, LoganSquare, Jackson, JsonPath и другие\)](#)

Чего НЕ БУДЕТ в данном сборнике:

1. Не будет видео про scala, так как scala это целый отдельный мир, для программистов на scala могу порекомендовать вот этот [сборник](#),
2. Не будет видео на английском языке, я старался выбирать русскоязычное видео, так как далеко не все программисты все-таки владеют английским в совершенстве, ну и англоязычного видео слишком много, поэтому я решил остановится пока только на русскоязычных видео (и небольшом количестве видео на украинском языке), К тому же, уверен, для англоязычного видео по Java в инете и так найдется немало подобных сборников.

Не стесняйтесь писать в комментариях (или мне в личку) о других интересных видео по Java, которое нет в этом списке.

Итак, начнем:

## 1. Вопросы оптимизации и производительности

### 1.1 Производительность и оптимизация

1. Сергей Куксенко — «Квантовые» эффекты производительности,
2. Сергей Куксенко — Сессия вопросов-ответов о производительности Java,
3. Алексей Шипилёв и Сергей Куксенко, Россия — Методология оптимизации производительности,
4. Сергей Куксенко, Oracle — Железные счётчики
5. Алексей Шипилёв, Oracle — Катехизис `java.lang.String` — о реализации строк в Java, способы оптимизации работы со строками и ошибки, которые часто допускают при оптимизации работы со строками,
6. Алексей Шипилёв — Java Benchmarking: как два таймстампа прочитать!
7. Алексей Шипилёв — Java Benchmarking: как два таймстампа записать
8. Алексей Шипилёв и Сергей Куксенко — Оптимизация производительности Java-приложений: теория
9. Алексей Шипилёв и Сергей Куксенко — Оптимизация производительности Java-приложений: детали
10. Алексей Шипилёв и Сергей Куксенко — Оптимизация производительности Java-приложений: ещё детали
11. Олег Бунин, Андрей Паньгин, Алексей Рагозин, Олег Анастасьев — Круглый стол. HighLoad
12. Алексей Шипилёв, Oracle — Сжимай меня полностью
13. Дмитрий Лазаренко — Оптимизация работы JVM в облаке
14. Дмитрий Чуйко — Быстрая загрузка ваших классов
15. Глеб Смирнов — HotSpot, что ты делаешь?! Ахаха, прекрати!
16. Kirill Golodnov — О бесконечно долгой работе дочерних процессов Java-приложения
17. Андрей Афанасьев — Оптимизация производительности распределенных приложений...
18. Руслан Черёмин — Disruptor и прочие фокусы
19. Gleb Smirnov — Side-By-Side Performance Comparisons — что значит «тормозит» и как с этим бороться
20. Nikita Salnikov Tarnovski — First steps in GC tuning — проблемы, связанные с производительностью из-за сборщика мусора

### 1.2 Высоконагруженные приложения (HighLoad)

1. Андрей Паньгин — Особенности разработки высоконагруженного сервера на Java
2. Алексей Шевчук — Подходы к решению проблем производительности высоконагруженных сервисов
3. Роман Елизаров — Миллионы котировок в секунду на чистой Java
4. Олег Анастасьев — За гранью возможного (Keynote) — о проблемах при создания Одноклассников и решения для высоконагруженных сервисов,
5. Андрея Паньгина — Выжимаем из сервера максимум: приемы кеширования и передачи данных на Java — описание приемов создания высоконагруженных систем на `Odnoklassniki.ru`,
6. Alexander Oleynikov — Практический опыт построения API высоконагруженных сервисов — описание создания высоконагруженных систем в Yandex'e,
7. Valerii Moisieienko — How to Manage 10K request per second in HA manner — история разработки высоконагруженного приложения на недорогом железе,

### 1.3 Сборка мусора, утечки памяти и работа с памятью

1. Алексей Рагозин — Сборка мусора, битва алгоритмов
2. Владимир Долженко — GC: друг или враг?
3. Роман Елизаров — Почему GC съедает все моё CPU?
4. Никита Сальников-Тарновский, Plumbr — Где моя память, чувак?!
5. Глеб Смирнов, Plumbr — Тайны — в наших головах, а не в JVM
6. Никита Сальников-Тарновский — Утечки памяти
7. Алексей Шипилёв — О чём молчат Heap Dump-ы
8. Владимир Воскресенский — Микрохирургия HotSpot
9. Александр Макаров — Исследование возможностей и ограничений JVM по оптимизации памяти
10. Андрей Паньгин — Кеширование данных вне Java Heap и работа с разделяемой памятью в Java
11. Владимир Иванов — G1 Garbage Collector

12. [Иван Крылов — StructuredArray и ObjectLayout](#) — как JVM представляет объекты и вопросы производительности представления объектов,
13. [Никита Сальников-Тарновский — Hear, off you go](#)
14. [Nikita Salnikov-Tarnovski — I bet you have a permgen leak in your application](#) — утечки памяти из-за class load'ов

## 2. Инструменты и IDE

### 2.1 IDE и инструменты разработки

1. [Андрей Зайцев, Дмитрий Нестерук, JetBrains](#) — Инструменты командной разработки
2. [Николай Чашников](#) — Поддержка Java 8 в IntelliJ IDEA
3. [Андрей Солнцев](#) — Eclipse vs IntelliJ IDEA
4. [Николай Чашников](#) — IntelliJ IDEA изнутри
5. [Дмитрий Найданов](#) — Play Framework 2.0 в IntelliJ IDEA
6. [Николай Чашников](#) — Компилятор Java и IDE: взаимовыгодное сотрудничество
7. [Андрей Чепцов, Барух Садогурский, Антон Архипов](#) — Круглый стол. Инструменты Java-разработчика
8. [Александр \(Шура\) Ильин](#) — Анализ покрытия с помощью Jcov
9. [Антон Архипов](#) — JRebel — инструмент, который позволяет «на лету» перезагружать классы, измененные с момента развёртывания приложения.,
10. [Барух Садогурский](#) — Разработка модульного софта: добро пожаловать в АД!
11. [Alexey Alexandrov](#) — All inclusive Jeastic PaaS — описание платформы для облачных вычислений
12. [Baruch Sadogurskii](#) — Managing Modular Software for your NuGet, C++ and Java Development
13. [Philip Torchinsky](#) — Интеграция баг-трекера и сервера CI с IntelliJ IDEA
14. [Iadimir Kozhaev](#) — How to write IDE for your DSL in one day
15. [Евгений Борисов](#) — Power of Gradle
16. [Sergey Morenets](#) — Gradle 2. Write once, build everywhere — внимание: видео на **украинском языке**

### 2.2 Отладка, профилирование, мониторинг

1. [Алексей Рагозин](#) — Непрерывное профилирование Java-приложений в ходе эксплуатации
2. [Андрей Паньгин, Одноклассники](#) — Лучший отладчик — сделанный своими руками
3. [Владимир Ситников](#) — Профайлер в каждый дом
4. [Владимир Красильщик](#) — Мониторинг Java-приложения с многопроцессной архитектурой
5. [Виталий Трифанов и Дмитрий Цителов](#) — Динамическое обнаружение гонок в многопоточных Java-программах
6. [Андрей Паньгин](#) — Лучший отладчик — сделанный своими руками
7. [Андрей Паньгин](#) — Аварийный дамп — черный ящик упавшей JVM
8. [Андрей Паньгин](#) — Анализ аварийных дампов JVM, часть 2
9. [Андрей Паньгин](#) — Анализ аварийных дампов JVM, часть 1
10. [Александр Отенко](#) — О чём молчит профайлер
11. [Андрей Паньгин](#) — Аварийный дамп — «черный ящик» JVM
12. [Алексей Рагозин](#) — Диагностические интерфейсы JVM, или Как сделать профайлер своими руками
13. [Владимир Воскресенский](#) — Практическое применение OSS для анализа Java-приложений — Использование Oracle Solaris Studio для анализа высокопроизводительных Java-приложений
14. [Алексей Шипилёв](#) — (The Art of) (Java) Performance Benchmarking II

## 3. Особенности разных языков JVM и версий Java

### 3.1 Groovy и Grails

1. [Барух Садогурский, Евгений Борисов](#) — Эпичные Groovy Puzzlers — Мечь Скобок
2. [Барух Садогурский и Евгений Борисов](#) — Groovy Puzzlers
3. [Барух Садогурский и Евгений Борисов](#) — Groovy Puzzlers — Странное, Непонятное и Мамочки-а-это-что-такое?!
4. [Евгений Борисов](#) — Марлезонский балет по Groovy в трех частях
5. [Барух Садогурский](#) — Метапрограммирование на уровне компилятора в Groovy
6. [Барух Садогурский](#) — Groovy DSL для plugin-ов: расширения человеческим языком
7. [Барух Садогурский](#) — Трансформации Абстрактного Синтаксического Дерева в Груви. ЩИТО?!
8. [Izzet Mustafaiev](#) — Groovy MOPping — принципы построения DSL на groovy
9. [Baruch Sadogursky](#) — Search for the Holy Grail (and test it once found) — какие решения для поиска существуют в Grail и какие решения для тестирования существуют в Grails

### 3.2 Clojure

1. [Николай Рыжиков](#) — Clojure: интерактивная разработка для интерактивного веб

2. [Maurice Naftalin — Closures and Collections: the World After Eight](#)
3. [Dmitriy Morozov — ETL all the things! — использование ETL \(Extract, transform, load\) в Clojure, как можно использовать Spark с Clojure,](#)
4. [Misha Kozik — DSL in Clojure — использование Clojure для написания DSL,](#)

### 3.3 Kotlin

1. [Андрей Бреслав — Type-safe Web with Kotlin](#)
2. [Андрей Бреслав — Project Kotlin](#)
3. [Евгений Геращенко — Kotlin, язык программирования для JVM](#)
4. [— Язык программирования Kotlin](#)

### 3.4 Java Embedded и IoT

1. [Александр Белокрылов, Александр Мироненко, Oracle — Восстание машин: в энтерпрайз с 256 КБ RAM](#)
2. [Белокрылов, Мироненко — Силовая установка с сердцем из Raspberry Pi и мозгом на Java Embedded](#)
3. [Александр Белокрылов и Александр Мироненко — Java Embedded у вас дома](#)
4. [Александр Мироненко и Александр Белокрылов — Java Embedded у вас дома](#)
5. [Лев Прийма — Java SE Embedded 8](#)
6. [Александр Белокрылов и Александр Мироненко — Полноценный Java runtime, 256Kb RAM и 5 литров пива](#)
7. [— Java ME Embedded: проверенные технологии на перспективных рынках](#)
8. [Yakov Fain — RESTful services and OAUTH protocol in IoT](#)

### 3.5 Java EE 7

1. [Markus Eisele — 50 Best Features of Java EE 7](#)

### 3.6 Java 8, лямбды и Stream Api

1. [Сергей Куксенко — Я, лямбда](#)
2. [Сергей Куксенко — JDK 8: Молот лямбд](#)
3. [Сергей Куксенко и Алексей Шипилёв — Через тернии к лямбдам, часть 1](#)
4. [Владимир Иванов — Java 8 и JVM: что нового в HotSpot](#)
5. [Сергей Куксенко — Java 8: Advanced Stream Techniques](#)
6. [Дмитрий Козорез — Java 8 Time API](#)
7. [Владимир Иванов — Java 8: что нового в HotSpot JVM?](#)
8. [Maurice Naftalin — What's New in Java 8](#)
9. [Александр Ильин — Type annotations in Java 8. И почему это хорошо](#)
10. [Сергей Куксенко и Алексей Шипилёв — Через тернии к лямбдам, часть 2](#)
11. [Сергей Куксенко и Алексей Шипилёв — Через тернии к лямбдам, часть 1](#)
12. [Сергей Куксенко — Stream API, часть 2](#)
13. [Сергей Куксенко — Stream API, часть 1](#)
14. [Сергей Куксенко — JDK8: Stream style](#)
15. [Алексей Фёдоров — Компромиссы в Java 8 с примерами и картинками](#)
16. [Александр Белокрылов — Знакомьтесь, Java 8](#)
17. [Дмитрий Козорез — Долгий путь к Лямбдам](#)
18. [Соколов Михаил — Java 8: лямбда выражения и Stream API](#)
19. [Oleg Tsal-Tsalko — Java 8 Date&Time benefits](#)
20. [Oleg Tsal-Tsalko — Java 8 Lambdas Hands-On-Labs](#)
21. [Pavel Grushetzky — Java for hipsters and rocks stars](#)
22. [Oleg Tsal-Tsalko — 10+ new features you ought to know using Java 8](#)

### 3.7 Java 9+ и будущее Java

1. [Иван Крылов, Владимир Иванов, Volker Simonis, Никита Липский — Круглый стол. Будущее Java-платформы](#)
2. [Владимир Иванов — Future of Java: 9 and beyond](#)
3. [Владимир Иванов — Платформа Java: прошлое и будущее](#)
4. [Шура Ильин — Project Jigsaw. Take 2](#)
5. [Александр Белокрылов, Oracle — Java: эволюция неизбежна](#)
6. [Аня Барски — Java Life Story](#)
7. [Olena Syrota, Oleg Tsal-Tsalko — Standard Java JSON binding. When? — описание существующих библиотек работы с JSON, их особенностей и проблем, ожидание стандартного JSON'a в Java 9](#)

## 4. Фреймворки Java

### 4.1 Spring

1. Барух Садогурский, Евгений Борисов — [Spring Puzzlers](#) — Начало
2. Евгений Борисов — [Spring Data? Да, та!](#)
3. Барух Садогурский, Евгений Борисов, Николай Алименков — [Зачем у Spring столько конфигураций?](#)
4. Евгений Борисов — [Spring the Ripper](#)
5. Евгений Борисов — [Spring-потрошитель, часть 2](#)
6. Евгений Борисов — [Spring-потрошитель, часть 1](#)
7. Евгений Борисов — [Spring 4.0: новое поколение](#)
8. Владимир Соськин — [Использование Spring 4 и AngularJS для разработки веб-приложений на базе архитектуры REST](#)
9. Меркурьев Евгений — [Весна идет, весне — дорогу! Новое в Spring Framework](#)
10. Artem Bilan — [Spring and Messaging: The next development generation](#)
11. Vladimir Tsukur — [From CRUD to Hypermedia APIs with Spring](#)
12. Igor Khotin — [App infrastructure for microservices with Spring Cloud 1.0.0](#) — микросервисы, cloud, Spring и DevOps,
13. Sergey Morenets — [Spring Boot. Boot up your development](#) — описание фреймворка Spring Boot, почему данный фреймворк позволяет создавать приложения намного быстрее,

### 4.2 Вебфреймворки (JavaFX, Javascript, AngularJS и т.п.)

1. Сергей Гринёв — [JavaFX: FXML и SceneBuilder](#)
2. Артём Ананьев — [Неформально о JavaFX](#)
3. Дмитрий Кузовец — [FXML и SceneBuilder: отделяем пользовательский интерфейс от логики в JavaFX](#)
4. Павел Нестеренко — [Java + JavaScript: переходы на мобильную сторону](#)
5. Евгений Родин — [Асинхронная обработка данных в web-приложениях на Java](#)
6. Евгений Родин — [Разработка веб-сайтов средствами Java \(на примере простого сайта\) — примеры разработки вебсайта с использованием mysql, tomcat и т.п. технологий](#)
7. Соколов Дмитрий — [Как выбрать библиотеки для современного Web-приложение на Java](#)
8. Владимир Соськин — [Использование AJAX в разработке web-приложений на Java](#)
9. Александр Дворецкий — [Как подготовиться к созданию нагруженных сайтов](#)
10. Владимир Соськин — [Использование Spring 4 и AngularJS для разработки веб-приложений на базе архитектуры REST](#)
11. Victor Polischuk — [Web-application I have always dreamt of](#)
12. Nikita Lipsky — [Delivering Native User Experience in Client-Side Java Applications](#) — нативные клиенты для работы с вебом вместо браузерного веба (от мобильных до десктопных приложений),
13. Olga Semeniuk — [Less JS! Web components for back-end developers](#) — введение в веб компоненты для бекенд разработчиков

### 4.3 Прочие фреймворки и популярные библиотеки

1. Цытович Павел — [Основы разработки приложения на платформе Android](#)
2. Андрей Шевченко — [Генерация больших отчетов с использованием Jasper Reports](#)
3. Владимир Красильщик — [Java-mapping для прагматичных программистов — использования библиотек и фреймворков для mapping'a \(dozer, spring type converter api, namin, scala dsl, transmorph и т.д.\)](#)
4. Andrii Golovei — [Highly available and scalable application for dummies with Storm](#) — Storm фреймворк для машинного обучения, высокопроизводительных вычислений

## 5. Базы данных, поисковые системы и BigData

### 5.1 NoSql и кеши

1. Олег Анастасьев — [За гранью NoSQL: NewSQL на Cassandra](#)
2. Алексей Зиновьев — [Мама, я так хочу Hibernate для NoSQL, ну или что там у вас есть на примете](#)
3. Яков Жданов — [In-memory accelerator for MongoDB](#)
4. Олег Анастасьев — [Класс! ная Cassandra](#)
5. Роман Антипин — [Франкенштейнизация Voldemort, или Key-Value данные в «Одноклассниках»](#)
6. Яков Жданов — [От дисковой архитектуры к In-Memory](#)
7. Алексей Рагозин — [Блеск и нищета распределенных кешей](#)
8. Alexey Tokar — [MongoDB в продакшене — миф или реальность?](#)
9. Artem Orobets — [How do you store your data?](#)
10. Roman Nikitchenko — [Big Data: from mammoth to elephant - transforming legacy with Hadoop](#) — как преобразовать легаси проекты в новые Big Data приложений с современным стеком технологий,

## 5.2 Sql базы данных и JPA

1. Николай Алименков — Босиком по граблям Hibernate
2. Яков Сироткин, Sidenis — Миграция с PL/SQL на Java

## 5.3 Поискковые системы

1. Михаил Хлуднев — В поисках Tommy Hilfiger — о Apache Lucene, свободной библиотеки для высокоскоростного полнотекстового поиска, о разных системах поиска,
2. Alexey Tokar — To find a needle in a haystack — о нечетком поиске в словарях от разработчика Яндекса

## 5.4 BigData, Data Scientist, Машинное обучение

1. Алексей Зиновьев — Java в качестве основного рабочего инструмента Data Scientist
2. Roman Nikitchenko — Apache HBase: crazy dances on the elephant back
3. Serhiy Masyutin — Big Data Analysis — внимание на **украинском языке**
4. Boris Trofimov — Scalding Big Data: Thorns and Roses,
5. Taras Matyashovskyy — Introduction to Real-Time Big Data with Apache Spark — внимание на **украинском языке**
6. Serhiy Masyutin — Big Data analysis approaches in Java world — что такое Big Data, технология Map-Reduce и примеры проектов
7. Kyrylo Holodnov — Scaling a solution of an NP-hard problem with Apache ZooKeeper — о Apache ZooKeeper, как с работать и решать различные задачи,
8. Andrii Golovei — Elastic stack in BigData — использование полнотекстового поиска в BigData используя решения компании Elastic (Elastic Search)

## 6. Особенности языка Java

### 6.1 Сериализации

1. Роман Елизаров — Факты и заблуждения о Java-сериализации
2. Olena Syrota, Oleg Tsal-Tsalko — Standard Java JSON binding. When? — описание существующих библиотек работы с JSON, их особенностей и проблем, ожидание стандартного JSON'a в Java 9

### 6.2 Работа с байткодом, java agent'ы, class loader и компиляции на лету

1. Антон Архипов, ZeroTurnaround — Javassist на службе у Java-разработчика
2. Никита Липский — Ahead-of-time компиляция
3. Антон Архипов — Java-агенты, Instrumentation API и Javassist
4. Антон Архипов — Загрузчики классов в Java: коллекция граблей
5. Владимир Иванов — Динамическая (JIT) компиляция в JVM
6. Дмитрий Завалишин — Конвертер байткода JVM для виртуальной машины Фантом

### 6.3 Асинхронность и многопоточность

1. Барух Садогурский — Как написать асинхронные многопоточные HTTP приложения
2. Алексей Шипилёв — ForkJoinPool в Java 8
3. Николай Алименков — Прикладная многопоточность
4. Дмитрий Чуйко, Oracle — CompletableFuture уже здесь
5. Глеб Смирнов — Расчлени многопоточность
6. Алексей Шипилёв — Прагматика Java Memory Model
7. Глеб Смирнов — Многопоточность Java под капотом
8. Роман Елизаров — Теоретический минимум для понимания Java Memory Model
9. Дмитрий Чуйко — Новинки в java.util.concurrent
10. Алексей Шипилёв — Bulletproof Java Concurrency
11. Сергей Куксенко — Модель памяти
12. Руслан Черёмин — Модель памяти Java: близкие контакты третьей степени
13. Alexey Fyodorov — Atomics, CAS, and Nonblocking Algorithms — вопросы многозадачности, атомики, неблокирующие алгоритмы, CAS и т.п.
14. Mikalai Aliimenkou — Java Concurrency Applied — примеры практических решений про многозадачность

## 6.4 Интеграция

1. Николай Алимиков — Нужен ли нам JMS в мире современных Java-технологий?
2. Яков Файн, Farata Systems — RESTful-сервисы и протокол OAuth в IoT
3. Александр Белоцерковский — Windows Azure + Java = Open Integration
4. Вадим Цеско — Фреймворк Akka и его использование в Яндексе
5. Vladimir Tsukur — Вместе весело шагать с HATEOAS! — разработка Web Api,
6. Максим Юнусов — «Масштабируемое приложение за 15 минут (используем AKKA под Java)
7. Вячеслав Лапин — Слабая связанность Java-сервера и web-интерфейса посредством REST API

## 6.5 Безопасность, Java Security Model и т.д.

1. Михаил Дударев — Побег из песочницы. Уязвимости нулевого дня в Java
2. Михаил Дударев — USS «Enterprise» — сквозь черные дыры безопасности Java Web-контейнеров
3. Михаил Дударев — Опасные типы, или Java Security Model на практике

## 6.6 Виртуальная машина Java и особенности реализации JVM

1. Андрей Паньгин — Java Runtime: повседневные обязанности виртуальной машины Java
2. Олег Плисс — Виртуальные машины. Часть 1
3. Олег Плисс — Виртуальные машины. Часть 2
4. Олег Плисс — Виртуальные машины. Часть 3
5. Олег Плисс — Виртуальные машины. Часть 4
6. Артём Ананьев — Проект OpenJDK
7. Андрей Бреслав — Компромиссы в разработке языков программирования
8. Андрей Бреслав — Компромиссы, или Как проектируются языки программирования
9. Никита Липский и Павел Павлов — Excelsior JET: внутренняя архитектура — описание быстрой JVM реализации Excelsior JET с компиляцией в машинный код,
10. Никита Липский и Павел Павлов — Excelsior JET: обзор возможностей — описание быстрой JVM реализации Excelsior JET с компиляцией в машинный код,
11. Nikita Lipsky — AOT компиляция — статическая компиляция JVM в Excelsior JET, её преимущества и возможности по сравнению с обычной компиляцией,
12. Nikita Lipsky — Java худеет. Уменьшение размера дистрибутива Java приложения без зависимостей — использование статического компилятора Excelsior JET для уменьшения размеров дистрибутивов Java приложений,
13. Никита Липский и Миша Быков — Занимательные истории из жизни техподдержки JVM — о техподдержке JVM Excelsior JET, о Excelsior JET и об особенностях реализации JVM, различия между JVM и проблемы при переходе между разными JVM,
14. Алексей Шипилёв — OpenJDK Frameworks: jmh & jctestress
15. Шура Ильин — OpenJDK Test Base
16. Шура Ильин — JTReg: OpenJDK test harness
17. Шура Ильин — OpenJDK TestFest Intro
18. Alexey Fyodorov — JDK: CPU, PSU, LU, FR – WTF??? — рассказывается о четырех типах обновлений JDK в компании Oracle (Critical Patch Update (CPU), Patch Set Update (PSU), Limited Update (LU) and Feature Release (FR)). Чем отличаются типы обновления, какие изменения возможны в каждом типе обновлений.

## 6.7 Аспектно-ориентированное программирование

1. Николай Гарбузов, Донривер — Аспектно-ориентированное решение классических проблем
2. Игорь Сухоруков. AspectJ Scripting

## 6.8 Рефлексия и Invokedynamic

1. Владимир Иванов — Invokedynamic: роскошь или необходимость?
2. Владимир Иванов — Глубокое погружение в invokedynamic

## 6.9 Работа с нативными библиотеками и кодом (JNI/JNA)

1. Петр Кудрявцев — Эффективная разработка Java приложений с использованием JNI/JNA технологий

## 6.10 Особенности языка Java

1. [Владимир Ситников, NetCracker — Выражаемся регулярно](#) — описание регулярных выражений, их использования и проблем,
2. [Алексей Рагозин — Структуры данных в Java, изобретаем заново](#) — описание реализации коллекций в Java и возможные методы их улучшения,
3. [Светлана Исакова — Простыми словами про вывод типов](#)

## 7. Качество кода и вопросы архитектуры

### 7.1 Рефакторинг, качество кода, методики программирования

1. [Яков Сироткин, Никита Липский, Олег Анастасьев, Филипп Дельгадо — Рефакторинг и технический долг](#)
2. [Антон Кекс — Как нам спасти Java? — Часть 1](#) — о проблемах Java, о том как можно сделать Java лучше, какие привычные подходы программирования в Java на самом деле больше вредны, чем полезны,
3. [Антон Кекс — Как нам спасти Java? — Часть 2](#) — о проблемах Java, о том как можно сделать Java лучше, какие привычные подходы программирования в Java на самом деле больше вредны, чем полезны,
4. [Николай Чашников — Писать код быстрее, ошибаться реже](#)
5. [Николай Чашников — Разработка API в Java-проекте](#)
6. [Николай Алименков — Парадигмы ООП](#)
7. [Михаил Ершов — Разработка совместимого API](#)
8. [Nikolay Chashnikov — Write code faster with fewer errors: is it possible?](#) — как избегать ошибок при написании кода,

### 7.2 Архитектура, бизнес правила, бизнес модель и примеры приложений

1. [Антон Кекс — Архитектура интернет-банка без Enterprise](#)
2. [Константин Кривоустов — Выбор технологий для корпоративного приложения](#)
3. [Александр Тоболь — Платформа для видео сроком в квартал](#)
4. [Антон Кекс — Как я создал desktop-приложение на Java, скачанное 9 миллионов раз](#)
5. [Виктор Полищук — JBoss Drools Expert против грязи](#) — описание системы управления бизнес правилами JBoss Drools
6. [Дмитрий Завалишин и Александр Шлянников — Java-технологии в Digital Zone](#) — примеры реализаций приложений в Digital Zone (использование груви скриптов для гибкой настройки бизнес модели системы приложения такси)
7. [Яков Сироткин — Почему геномный ассемблер нельзя написать за выходные?](#) — история разработки геномного ассемблера
8. [Яков Сироткин — История релизов](#) — продолжение истории разработки геномного ассемблера
9. [Евгений Борисов — eXtreme Application Platform \(XAP\)](#)
10. [Барух Садогурский — Bintray: Масштабируемый сервис в облаке с нуля](#)
11. [Anatoliy Sokolenko — Key Problems of Microservice Architecture](#) — что такое сервисы и микросервисы, как их реализовать на практике, чем это лучше стандартной архитектуры и какие проблемы возникают при такой архитектуре,
12. [Izzet Mustafaiev — Fault tolerance – look, it’s possible!](#)
13. [Oleg Tsal-Tsalko — Distributed systems and scalability rules](#)
14. [Baruch Sadogursky — Everything you wanted to know about async HTTP apps in Java](#) — о написании асинхронного менеджера загрузок с параллельной загрузкой файлов, одного файла, прерывания и паузы для программы репозиторий
15. [Maxim Ivanov — Hey, I’ve seen something like it before!](#) — подходы к принятию архитектурных решений. Хайп (использования модных базвордов — микросервисов, nosql и т.п. ради базвордов) и его последствия для архитекторов.

### 7.3 Тестирование приложений

1. [Глеб Смирнов — Мутационное тестирование, или О чём молчит Code Coverage](#) — описание почему 100% Code Coverage у Unit тестов не гарантирует правильную работу кода, что такое мутационное тестирование и как оно помогает улучшить Unit тесты, описание мутационной системы pitest (<http://pitest.org/>)
2. [Алексей Рагозин — Как и зачем эмулировать распределённый кластер в JVM](#) — описание библиотеки, которая позволяет для задач Unit тестирования эмулировать распределённый кластер на одной ноде
3. [Алексей Рагозин — Сеть — это большая JVM](#) — часть 2 лекции » Как и зачем эмулировать распределённый кластер в JVM "
4. [Mikalai Alimenkou — «TDD for database related code, how is it possible?»](#) — описание TDD методологии
5. [Katya Kameneva — How I lost my QA job](#) — о внедрении TDD технологий
6. [Dmytro Chyzyhkov — Testing Web Apps with Spring Framework](#) — тестирование Spring MVC приложений
7. [Андрей Солнцев — Пацан накодил — пацан протестил!](#)

## 8. Прочее

1. [Алексей Вишенцев и Илья Агошков — Advanced Deployment](#)
2. [Дмитрий Черепанов — Self-contained Application](#)
3. [Сергей Куксенко — Quantum Performance Effects](#)
4. [Кузьма Деретюк — UrsaJ HTTP File Storage](#)

5. Дмитрий Чуйко — Java Mission Control

6. Igor Khotin — The Craft of Metaprogramming on JVM — о метапрограммировании в Java

**Update:** Внимание, актуальная версия со всеми ссылками на видео находится теперь в моем github'e проекте [useful-java-links](#), по этой ссылке.

▼ Общее оглавление 'Шпаргалок'

1. JPA и Hibernate в вопросах и ответах
2. Триста пятьдесят самых популярных не мобильных Java opensource проектов на github
3. Коллекции в Java (стандартные, guava, apache, trove, gs-collections и другие)
4. Java Stream API
5. Двести пятьдесят русскоязычных обучающих видео докладов и лекций о Java
6. Список полезных ссылок для Java программиста
- 7 Типовые задачи
  - 7.1 Оптимальный путь преобразования InputStream в строку
  - 7.2 Самый производительный способ обхода Map'ы, подсчет количества вхождений подстроки
8. Библиотеки для работы с Json (Gson, Fastjson, LoganSquare, Jackson, JsonPath и другие)

java, видео, производительность, оптимизация, высоконагруженные приложения, highload, утечки памяти, сборка мусора, IDE, отладка, профилирование, мониторинг, groovy, grails, Clojure, Kotlin, Gradle, Java Embedded, IoT, Java EE 7, Java 8, лямбды, stream api, Java 9, Spring, JavaFX, NoSql, BigData, Data Scientist

↑ +29 ↓

👁 96,8k ★ 1205



Автор: @vedenin1980



рейтинг  
**Luxoft** 106,62  
Сайт Twitter

## Похожие публикации

+9 [Интеграционное тестирование в Java EE, Адам Бин](#)

👁 17,2k ★ 120 💬 1

+3 [Мастер-класс Адама Бина «Java EE: Архитектура, шаблоны и решения»: отзывы и впечатления участников](#)

👁 5,6k ★ 24 💬 5

+5 [Стратегии интеграции JavaFX](#)

👁 8,4k ★ 35 💬 0

## Комментарии (26)



AlexanderFedorenko 10 декабря 2015 в 15:57 #

+9 ↑ ↓

Когда учишь программирование по видеоурокам



роху 10 декабря 2015 в 22:08 # ↗ ↑

+8 ↑ ↓

Это не уроки, это выступления по теме. Разница как между учебником по физике и конференцией по вопросам свойств планетных систем других звёзд.

 **Terranz** 10 декабря 2015 в 16:34 # -4 ↑ ↓

большое спасибо, положил в копилку полезностей для своих джунов

 **рохu** 10 декабря 2015 в 22:11 # h ↑ +8 ↑ ↓

Можно подумать для вас тут вообще ничего полезного нет.

 **Beholder** 10 декабря 2015 в 21:44 # -2 ↑ ↓

Угу... Не подсчитаете общую продолжительность видео? Наверное, хватит убить время на весь год. No, thanks.

Лучше бы буквами писали. Это, конечно, требует от авторов больших усилий, но намного полезней для обучающихся.

 **рохu** 10 декабря 2015 в 22:07 # h ↑ +6 ↑ ↓

250 видео. Если смотреть по одному в день, то почти на треть меньше года. Ну и, например про юнит-тестирование буквами получилось бы гораздо хуже.

 **vedenin1980** 11 декабря 2015 в 13:15 # h ↑ +3 ↑ ↓

Не подсчитаете общую продолжительность видео

Что-то около 200 часов, то есть на пару недель non-stop просмотра, но... никто же не предлагает смотреть все и сразу (все равно, все технологии Java в совершенстве не знает, наверное, практически никто), плюс большинство видео докладов можно не смотреть, а слушать фоном, занимаясь рутинными делами или во время поездки на работу.

 **рохu** 10 декабря 2015 в 23:14 (комментарий был изменён) # +6 ↑ ↓

Спасибо огромное! Ролики крайне интересные!

На случай, если на хабре есть ещё кто-то, кто думает, что видеофайлы надо хранить на своём компьютере, а не непонятно где в интернете, я сделал [файл со списком ссылок](#). Он человекочитаем, названия и разделы там сохранены. Этот файл можно скормить программе [youtube-dl](#), она всё скачает. Вызывать её нужно из командной строки, примерно так.

```
youtube-dl --encoding UTF-8 -a java_urls.txt
```

encoding нужен, чтобы в названиях файлов не терялись русские буквы, может будет работать и без этого, но мало ли.

 **AlexanderG** 12 декабря 2015 в 21:42 # h ↑ +3 ↑ ↓

видеофайлы надо хранить на своём компьютере

К сожалению, да, видео на ютубе имеют свойство иногда исчезать.

 **aleksandy** 11 декабря 2015 в 06:59 (комментарий был изменён) # +4 ↑ ↓

Не будет видео про scala, так как scala это целый отдельный мир

В таком случае, почему Groovy, Kotlin и Clojure отказано в отдельном мире? Они, в принципе, точно так же имеют свои компиляторы, и генерируемый ими код точно так же бежит под JVM.

~~Дискриминация, однако.~~

 **рохu** 11 декабря 2015 в 12:47 # h ↑ +2 ↑ ↓

Не скажу про Kotlin и Clojure, но вот Groovy сильно совместим с джавой. Наверное как C с C++. Scala она не такая.

 **gurinderu** 11 декабря 2015 в 13:52 # h ↑ 0 ↑ ↓

А в чем scala не такая? bytecode то один.

 **рохu** 11 декабря 2015 в 14:00 # h ↑ 0 ↑ ↓

Синтаксисом. Код на джаве зачастую также является кодом на груви. Как Си код часто является одновременно кодом на C++. А scala, насколько я знаю, всё не так.

 **basnopisets** 12 декабря 2015 в 12:23 # h ↑ 0 ↑ ↓

у Котлина очень высокая степень интероперабельности с Java, что и облегчает переход на него

 **vedenin1980** 11 декабря 2015 в 12:55 # h ↑ +2 ↑ ↓

Скажем так, видео по scala (в отличии от Groovy, Kotlin и Clojure) было слишком много чтобы я смог осилить квалификацию ещё и этого видео. Так что никакой дискриминации, простая практичность.

 **AlexanderG** 12 декабря 2015 в 21:14 # 0 ↑ ↓

Ну что же, теперь посмотрю их и стану супер-специалистом. Осталось найти вторую жизнь на просмотр... Спасибо!

 **23derevo** 13 декабря 2015 в 15:54 (комментарий был изменён) # -1 ↑ ↓

Краткое содержание поста:

1. есть три клевых Youtube-канала у JUG.ru, JEEconf и Java Day Kiev.
2. Ах да, и у Luxoft Training Center тоже, тоже есть свой канал!!!

Классификация, проделанная автором, просто смешна.  
Отбор видео отвратительный.

Спасибо, конечно, что вы включили в этот пост 90% наших видео, включая откровенный шлак (да-да, шлак у нас иногда случался). Я вот только не понимаю, зачем вы включили сюда откровенно говено записанные видео типа вот этого вместо нормально записанного видео этого же доклада на JUG.ru. Ну и зачем вы включили сюда 20 откровенных дублей и повторов.

Короче, дешевая работа.

 **poхи** 13 декабря 2015 в 16:11 # h ↑ +1 ↑ ↓

Дайте пожалуйста ссылку на нормально записанное видео на JUG.ru. И скажите пожалуйста, когда вы говорите мы, какую организацию вы представляете?

 **23derevo** 13 декабря 2015 в 16:27 # h ↑ 0 ↑ ↓

Видео — [вот](#), например.

Мы = [JUG.ru](#)

 **vedenin1980** 13 декабря 2015 в 17:19 (комментарий был изменён) # h ↑ +1 ↑ ↓

Классификация, проделанная автором, просто смешна.

Сделайте другую не смешную хотя бы для своего канала. Скажем в Jeeconf ко всем видео файлам в канале youtube'a есть нормальное описание (пусть и на английском), у JUG только название вроде «Сжимай меня полностью». Вот как догадаться о чем эта лекция? Пересматривать все сотни часов подряд пытаюсь найти нужную? Пусть классификация смешная, но она есть и на неё можно ориентироваться, а не просто «смотрите все видео канала подряд в надежде найти полезное».

вы включили в этот пост 90% наших видео, включая откровенный шлак (да-да, шлак у нас иногда случался).

Простите, но этот **шлак** с вашего **официального** youtube канала. Кто же заставляет выкладывать туда шлак? Честно говоря я надеялся что организаторы конференций отвечают за качества того что выкладывают под своих логотипом, поэтому как раз с JUG'a старался не убирать видео, в отличии от остальных источников.

зачем вы включили сюда откровенно говено записанные видео типа вот этого вместо нормально записанного видео этого же доклада на JUG.ru.

Не вместо, а вместе. Так как если поищите "Андрей Паньгин", то найдете то видео ссылку на которое вы дали ниже. Если откроете и то и другое видео то там даже начало совершенно разное, не говоря уже про вопросы к докладчику.

Ну и зачем вы включили сюда 20 откровенных дублей и повторов.

Затем, что как бы вам не хотелось, JUG не единственная конференция в русскоязычном инете, если доклад называется по другому, был прочитан на другой конференции и докладчику задавали другие вопросы — это совсем другой доклад. Откровенные дубли с одинаковыми докладчиками и названиями, я старался убирать, но не похожие доклады на разных конференциях. Я так понимаю, основная претензия зачем я вообще включал в обзор какое-то видео кроме JUG'a и рассказывал что бывают и другие конференции?

Короче, дешевая работа.

Не буду спорить про фламастеры, сделайте статью с описанием и классификацией видео с вашего канала лучше и подробнее. Я первый вам скажу спасибо. Ну... или хотя бы добавьте нормальные описания в вашем youtube канале, по которым можно было найти нужное видео.

 **vedenin1980** 13 декабря 2015 в 17:26 (комментарий был изменён) # h ↑ +1 ↑ ↓

Да, кстати:

| Классификация, проделанная автором, просто смешна.

Если у вас есть конструктивные замечания, как сделать её лучше — пишите, я исправлю.

| включая откровенный шлак

Опять-таки, ну кто мешает написать мне в личку и попросить удалить видео с канала JUG'a, которые вы сами считаете шлаком?

| есть три клевых Youtube-канала у JUG.ru, JEEconf и Java Day

Знаете больше хороших русскоязычных каналов — напишите, я не нашел, хотя искал.

 **23derevo** 13 декабря 2015 в 19:10 # h ↑ 0 ↑ ↓

Вы просто перечислили все (ну или почти все) видео с канала JUG.ru, добавив по десятку видео с других трех каналов. Если вы считаете, что это какая-то полезная работа — не вопрос, флаг вам в руки.

 **vedenin1980** 13 декабря 2015 в 20:03 # h ↑ +1 ↑ ↓

Хорошо, если настаиваете, я постараюсь максимально заменить видео с канала JUG на аналогичные доклады те же авторов с других конференций. Ваше реакция мне не очень понятна, но дело ваше.

А степень полезности работы я предпочитаю оценивать по количеству пользователей добавивших статью в избранное.

 **23derevo** 13 декабря 2015 в 20:10 # h ↑ +1 ↑ ↓

поймите правильно, мне только на пользу, что 180 из 250 опубликованных вами видео — с нашего канала. Если хотите их заменить на аналогичные — удачи. Останется не 180, а 150, причем в большинстве случаев хуже по качеству. Несколько десятков похожих вы наверняка найдете на канале JEEconf. Так что наших все равно будет больше половины :)

Странное чувство вызвало то, что вы, выложив своими ссылками весь или почти весь наш ютьюб-канал, не потрудились даже посоветоваться как-то с нами.

 **vedenin1980** 14 декабря 2015 в 00:04 # h ↑ +2 ↑ ↓

Ок, я вас понял, подумаю как исправить эту статью. Давайте свяжемся уже в личке и обсудим ваши предложения и замечания. Я всегда готов к любой конструктивной критике и предложениям?

 **ptiss** 21 декабря 2015 в 09:15 # 0 ↑ ↓

Вот ещё по Котлину, там большей частью про сам язык, несмотря на название доклада.

Только зарегистрированные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

## Самое читаемое

Разработка

Сейчас **Сутки** Неделя Месяц

+242 [Как Skype уязвимости чинил](#)

👁 23,3k ★ 98 💬 134

+19 [Улучшение производительности PHP 7](#)

👁 2,4k ★ 25 💬 3

+77 [Здравствуй, дорогой Мегафон](#)

👁 9k ★ 15 💬 79

+10 [Прототип RFC HTTP-кодов состояния для ошибок разработчиков \(диапазон 7XX\)](#)

👁 1,5k ★ 4 💬 6

+12 Компания Google представила набор тестов Wucherproof

👁 1,3k ★ 7 💬 0

## Интересные публикации



**H** Гейзенбаг: Версия 1.0 💬 0

**H** Компания Google представила набор тестов Wucherproof 💬 0

**H** Улучшение производительности PHP 7 💬 3

**CT** Защищенный Dell 💬 4

**H** Преобразование формы представления данных при помощи Excel+PowerQuery 💬 0