

# Эффективное профилирование Java приложений

Онлайн курс

## О курсе

Курс посвящен практике использования диагностических инструментов, в первую очередь, профайлеров для решения проблем производительности и стабильности Java приложений.

Данный курс ставит перед собой следующие задачи

- предоставить аудитории методику эффективного применения инструментов профилирования в контексте решения практических задач
- продемонстрировать работу с доступными Open Source профайлерами такими как Visual VM, Java Flight Recorder, а также другими бесплатными инструментами
- научить использованию диагностических инструментов в контейнеризированных и облачных окружениях
- закрепить знания практической работой со специально подготовленными примерами проблемного кода

Курс объединяет лекционные блоки и практические упражнения для лучшего закрепления материала.

Курс разделен на независимые модули для обеспечения гранулярности подачи материала. Отдельные модули могут быть прослушаны индивидуально.

## Ценность курса

На практике, задачи требующие применения профайлера встречаются не часто. По этой причине, у разработчиков, обычно нет возможности научиться эффективно владеть этим инструментом на собственном опыте.

С другой стороны, эффективное использование профайлера в критической ситуации может драматически сократить время решения проблемы (от недель и месяцев до считанных дней).

Данный учебный курс даёт разработчику ключевые знания и практические навыки для эффективного применения профайлера в своей работе.

## Предполагаемая аудитория

Разработчики / архитекторы Java (всех уровней подготовки), Big Data инженеры, SRE (Site Reliability Engineer), инженеры по производительности.

## Тренер

Алексей Рагозин - автор курса и тренер, имеет более 10 лет опыта в области разработки высоконагруженных распределенных систем в различных индустриях. Алексей активный докладчик на российских и зарубежных конференциях в области ПО и open source контрибьютор.

Блог: <http://blog.ragozin.info>

Список докладов на конференциях (видео и слайды): <http://blog.ragozin.info/p/tech-talks.html>

GitHub: <https://github.com/aragozin/>

Контакты: +7 925 499 45 26 email: alexey.ragozin@gmail.com

## Описание модулей

Курс разделён на 3 модуля по 3 занятия каждый. Модули включают следующие темы:

### Модуль 1

- Использование техник профилирования для поиска узких мест в коде (3 занятия)

### Модуль 2

- Java Flight Recorder и Mission Control - глубокое погружения (2 занятия)
- Диагностика из командной строки и инфраструктура JMX (1 занятие)

### Модуль 3

- Диагностика проблем с памятью JVM (2 занятия)
- Многопоточность - проблемы и диагностика (1 занятие)

### Тема “Использование техник профилирования для поиска узких мест в коде”

*Продолжительность: 3 занятия x 3 часа.*

Данный модуль является ключевым с точки зрения тематики профилирования. В рамках модуля, будут раскрыты основные техники профилирования: сэмплирование и трассировка, а так же предложен общий подход для поиска причин проблем производительности в информационных системах.

#### **Занятие 1**

Правильная постановка задачи оптимизации или устранения дефектов производительности. Знакомство с Visual VM. Обзор техник профилирования. Разбор примеров поиска узких мест с помощью Visual VM.

#### **Занятие 2**

Общий подход к поиску проблем производительности. Дерево принятия решений при поиске узких мест в приложениях на JVM платформе. Знакомство с Mission Control, пример сэмплирования с использованием Mission Control. Пример сэмплирование с использованием SJK. Построение флейм графов (flame graph) по различным типам исходных данных. Сравнение результатов разных профайлеров на одном пример. Особенности работы с результатами сэмплирования стеков потоков и синдром смещения оценки при сэмплировании (sampling bias).

#### **Занятие 3**

Обзор техники профилирования трассировкой. Пример трассировки в VisualVM на простом примере. Разбор комплексного примера с использованием сэмплирования и трассировки в VisualVM. Обзор VTrace и пример работы с ним в VisualVM. Заключительная лекция и подведение итогов.

**Требования к ПО:** JDK (11.0.21 или выше), git, Apache Maven (3.6 или выше), Visual VM, Mission Control, Java IDE по желанию.

### Тема “Java Flight Recorder и Mission Control - глубокое погружение”

*Продолжительность: 2 занятия x 3 часа.*

Модуль посвящен углубленному изучению Java Flight Recorder и Mission Control. Рамках модуля будут продемонстрированы и разобраны большинство возможностей этих инструментов.

#### **Занятие 1**

Обзор концепции Java Flight Recorder (JFR). Знакомство с Mission Control и её возможностями. Пример диагностики приложения с использованием JFR / Mission Control. Особенности сэмплирования JFR. Обзор событий JFR в современной JVM.

#### **Занятие 2**

Работа с JFR из командной строки. Удаленное использование JFR с помощью Mission Control. Конфигурация JFR через параметры запуска JVM. Использование JFR в режиме непрерывного профилирования. Программная работа с JFR, пример создания новых событий.

**Требования к ПО:** JDK (11.0.21 или выше), git, Apache Maven (3.6 или выше), Visual VM, Mission Control, Java IDE по желанию.

### **Тема “Диагностика из командной строки и инфраструктура JMX”**

*Продолжительность: 1 занятие x 3 часа.*

Распространенные диагностические инструменты JVM используют ряд низкоуровневых протоколов. Контейнеры и развёртывание в облаках может требовать дополнительной настройке. Так же иногда возникает необходимость провести диагностики используя исключительно консольный доступ к серверному окружению. В данном модуле будут разобраны перечисленные выше сценарии.

#### **Занятие 1**

Архитектура JMX. Пример настройки JVM для удаленного подключения. Обзор возможностей jcmd. Пример аварийной инициализации JMX в Kubernetes с использованием port forwarding. Пример создания MBean в прикладном коде.

**Требования к ПО:** JDK (11.0.21 или выше), git, ssh клиент, Apache Maven (3.6 или выше), Docker (или Podman), Visual VM, Mission Control, Java IDE по желанию.

### **Тема “Диагностика проблем с памятью JVM”**

*Продолжительность: 2 занятие x 3 часа.*

JVM обеспечивает автоматическое управление памятью с помощью механизма сборки мусора (GC). Это в свою очередь создает класс проблем производительности/стабильности, с которыми могут столкнуться разработчики на JVM платформах.

#### **Занятие 1**

Обзор архитектуры памяти JVM и сборки мусора, а также классов возможных проблем. Анализ утечек памяти, пример поиска утечки памяти. Основы анализа дампов памяти в Visual VM и Eclipse Memory Analysis Tool.

#### **Занятие 2**

Ключевые метрики сборщика мусора (GC), анализ работы GC в Mission Control. Сэмплирование аллокации памяти с использованием Java Flight Recorder / Mission Control. Обзор специальных ссылочных типов (WeakReference, SoftReference, Finalizer) в JVM и их эффект на GC. Работа с Native Memory Tracking в JVM

**Требования к ПО:** JDK (11.0.21 или выше), git, Apache Maven (3.6 или выше), Visual VM, Mission Control, Eclipse MAT, Java IDE по желанию.

### **Тема “Многопоточность - проблемы и диагностика”**

*Продолжительность: 1 занятие x 3 часа.*

Курс посвящён диагностике проблем специфических для многопоточных приложений, таких как “клинч потоков” (deadlock) и межпоточная конкуренция (contention).

#### **Занятие 1**

Обзор проблем многопоточных приложений. Пример анализа клинча с использованием дампов потоков. Потоки JVM на линукс платформе и межпоточное взаимодействие. Анализ узких мест межпоточного взаимодействия методом сэмплирования. Инструменты Java Flight Recorder для анализа взаимодействия потоков.

**Требования к ПО:** JDK (11.0.21 или выше), git, Apache Maven (3.6 или выше), Visual VM, Mission Control, Java IDE по желанию.