

Описание

“Это Java памяти много ест. Надо ей ещё добавить” – типичная ~~отмазка~~ резолюция от разработчиков, которым некогда/неохота/тяжело разбираться с истинной причиной очередного `OutOfMemoryError` на production. Повезло, если такое предложение прокатит и не повторится, но, скорее всего, рано или поздно разбираться всё же придётся.

В таких случаях, как правило, нужно сделать **снимок памяти** (heap dump), но с ним не всё так просто: он может много весить, его бывает не легко достать с production, в него не посмотришь “глазами” (он бинарный), а в спец. инструментах данных столько, что можно утонуть.

На этом тренинге вы научитесь снимать дампы памяти разными способами (в зависимости от окружения), ускорять и упрощать их передачу к себе для анализа и, главное, овладеете техниками и инструментами для эффективного обнаружения источников проблем с потреблением памяти: от построения наглядных отчётов до глубокого анализа графа объектов через язык запросов.

Ценность

Знания и навыки, которые предоставляет тренинг, могут оказаться бесценными в ситуациях исчерпания памяти приложением, когда расширение доступной памяти уже невозможно или требует серьёзного обоснования, а поскольку сервис (и бизнес) пристаивает, на долгие инженерные изыскания времени нет.

Аудитория

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

Содержание

Теория

1. Вводная модуля: дампы памяти, их содержимое, источники
2. Связь со сборкой мусора
3. Основы устройства дампов
4. Анализ дампов в Eclipse Memory Analyzing Tool

Практика

1. Общая вводная (бизнес-легенда)
2. Настройка лабораторного стенда
3. Разбор нескольких базовых практических кейсов по схеме:
4. Продвинутые кейсы
5. Выводы и рекомендации