

++ 2015 JVM | Никита Липский, Владимир Парфиненко — JVM:  
краткий курс общей анатомии

[https://www.youtube.com/watch?v=JbLCISMRK\\_I&ab\\_channel=JUG.ru](https://www.youtube.com/watch?v=JbLCISMRK_I&ab_channel=JUG.ru)



Докладчики занимаются разработкой JVM.  
Особенность их версии в том, что она имеет статический компилятор, который порождает обычный исполняемый exe файл из java программ.

В начале доклада будет рассказано про классы и байткод, после этого рассказ о том как эти классы обрабатываются JVM, как собирается мусор и тд.

### **Class file**

Байткод джавы имеет четкую спецификацию и байткод написанный для одной виртуальной машины будет корректно работать на любой виртуальной машине.

В программах используемых томкат нет класса main, поскольку он есть в самом томкате.

Для jvm идентификатор класса это имя загрузчика + имя класса

По логике стек должен переполниться, поскольку мы постоянно добавляем значения и должен ожидаться `stackOverflowError`.  
Верификация выполняется до исполнения кода, на этапе его компиляции.

После того как верфикатор отработал, не возможна ситуация когда мы сложим 2 объекта.

Интерпретатор берет последовательно инструкции, по ее названию понимает сколько ей нужно аргументов, берет аргументы, выполняет инструкцию и кладет результат на стек. После этого выполняет следующую инструкцию

Интерпретация работает крайне медленно.

Могут выполнять агрессивную оптимизацию, которая подходит только в текущем случае, во всех остальных она может быть неверна.

Все современные jvm стараются как можно раньше перестать исполнять код и как можно быстрее его скомпилировать.

Размер любого объекта нельзя определить, для каждой виртуальной

