Java, первый семестр

Купоросов Василий, Недиков Константин

4 сентября 2017 г.

Содержание

1. Вступление

Ссылка на презентацию

1.1. Краткая история

Откуда проблемы у джавы? Разберёмся с его проблемами из детства.
В 91 году решили, что `C++` - плохой язык, поэтому решили сделать `oak` - язык, который предполагали использовать для встраиваемых систем(кофеварки, микроволновки и т.д.)
Потом решили запихать в TV приставку, но всё всё равно было медленно и неудобно, на плюсах было больше программистов, было быстрее всё писать и проще со многих сторон.
Тогда решили сосредоточиться на апплетах, тогда ещё не было `JS` и `flash`, и вот это сработало, они создали нишу, люди плевались, корчились, но писали на этом языке эти странные вещи, ибо аналогов особо не было.
В 96 появился `JDK` – Java Development Kit, который содержал в себе компилятор, виртуальную машину и стандартную библиотеку классов
Java запускается на виртуальной машине. Наш код нужно перевести в нормальный вид для VM – java компилятор переводит java код в байткод. Его одно из главных отличий от машинного кода – в нём хранится огромное количество дополнительной информации, которая не нужна для исполнения программы.
В 98 году произошло разделение языка на два, потом ещё немного разделился, но об этом чуть позже.
В 2004 они выпустили пятую версию языка и этот язык наконец-то стал "нормальным", ибо раньше он был очень медленным и проигрывал почти во всём популярным языкам. Он стал удобным, там стало много полезных вещей и появилась внятная программа развития языка.
`Java` обратно совместима, т.е. восьмая версия компилятора запустит то, что написано для первой. Это основной источник проблем этого языка. В `C#` же оттолкнулись от этой проблемы, несмотря на то, что он является идейным наследником.

1.2. Идеи и особенности

1.2.1. Причина изучения `Java`

Почему мы всё же учим этот язык? Схема с виртуальной машиной сейчас работает почти со всеми языками современного мира, знать язык, который её заложил, весьма полезно. Почему же не `C#`? Если вы умеете программировать на джаве – переход на шарп будет достаточно безболезненным. Обратный же переход очень тяжёлый, в основном из-за того, что в `Java` нет многих фишек, про которые все так любят рассказывать в `C#`.
Используется же `Java` в основном в `Android`, немного для серверных программ, desktop приложений почти нет(но вы всё ещё можете написать очередной клон minecraft).

1.2.2. Особенности

Кроссплатформенность: программу на джаве можно запустить на виртуальной машине на любом компьютере, и она будет работать `ТАКЖЕ`, как и на вашем.

JM работает долго, примерно в 10-20 раз медленнее, чем наш любимый C++ на машинном коде. Это происходит из-за того, что во время исполнения нужно интерпретировать байткод в машинный.
Есть Just-In-Time, которая в процессе выполнения переносит программу в машинный код вместе с оптимизациями. Засчёт этого ускорение может быть очень большим, в итоге выходит, что Java может работать в пару раз медленнее, чем C++.
Из-за этой особенности есть понятие, как разогрев JM. Код может выполняться принципиально разное время на первый и десятый запуск программы.

В Java можно не следить за памятью, всё почистит сборщик мусора. Неверно, что если на объект нет ссылок - он будёт удалён. Если, например, два объекта ссылаются друг на друга, но на них ничего не ссылается - они всё равно будут удалены.
Ещё у нас есть безопасность: любой байткод верифицируется, а ещё нельзя залезть в чужую память. Есть встроенный механизм управления правами, которые ограничивают мощь виртуальной машины(которую, кстати, тоже можно запустить с ограниченными правами) для того, чтобы всё портить.
В Java довольно хорошая стандартная библиотека, но в сравнении с Python всё равно всё ничтожно.

1.3. Разновидности

JRE(Java Runtime Environment) - это JDK без компилятора, чтобы пользоваться программами на java.

Редакции:

1. Standard Edition - для ”ПК”
2. Micro Edition - для ”мобильных”(не для андроид)
3. Enterprise Edition - Для серверов
4. Java Card - Для каких-то карточек

1.4. Стиль и синтаксис

Язык Полностью объектно-ориентированный, т.е. все функции должны быть методами класса.
В языке java есть кодовые соглашения:

1. Писать постфиксный ++
2. Использовать camelCase для имён
3. Ставить фигурную скобочку на той же строчке
4. Писать документацию

Далее всё понятно на слайдах, пока предлагается ознакомиться с ними. **TODO**