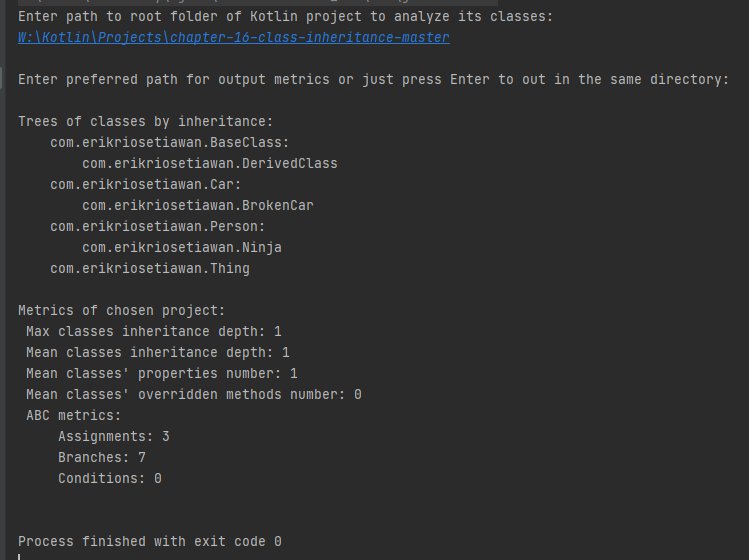
Анализ результатов работы программы, создающей AST для классов проекта Kotlin и предоставляющей метрики для полученных AST, на публичных Github репозиториях, в которых используется Kotlin.

Уточнение: для хранения метрик со средним значением какой-либо величины используется целочисленный тип (Вычисленное среднее значение округляется до целого числа)

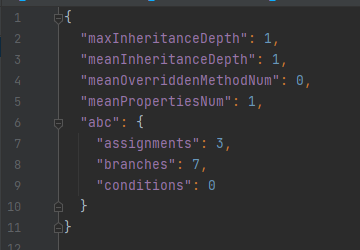
1. <https://github.com/erikrios/chapter-16-class-inheritance>.

Проект Kotlin-проект состоящий из 10 .kt файлов и 3 «линий» наследования: Car -> BrokenCar, BaseClass -> DerivedClass, Person -> Ninja

Результат работы программы:



JSON:



Результат работы адекватный. Для 3 из 4 деревьев классов глубина наследования = 1.

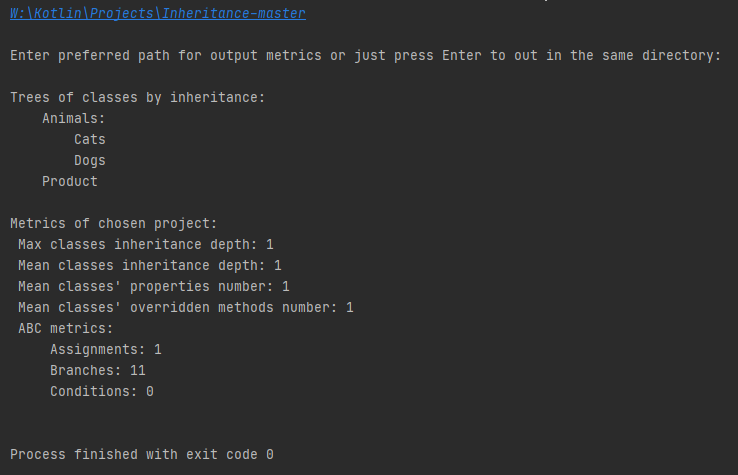
5 свойств для 7 классов в действительности, что соответствует в среднем одну свойству на класс.

Т.к. данный проект не содержит какую-то логику и ветвления, то conditions в ABC-метрике = 0.

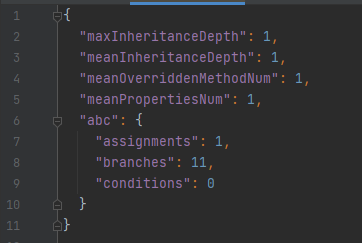
1. Аналогичный проект с более простой структурой: <https://github.com/Prateek-collab/Inheritance>

Проект состоит из 3 классов: суперкласса Animals (с 1 открытым для переопределения методом) и двух дочерних на одном уровне наследования: Cats и Dogs, переопределяющих родительский метод.

Результат работы программы:



JSON:

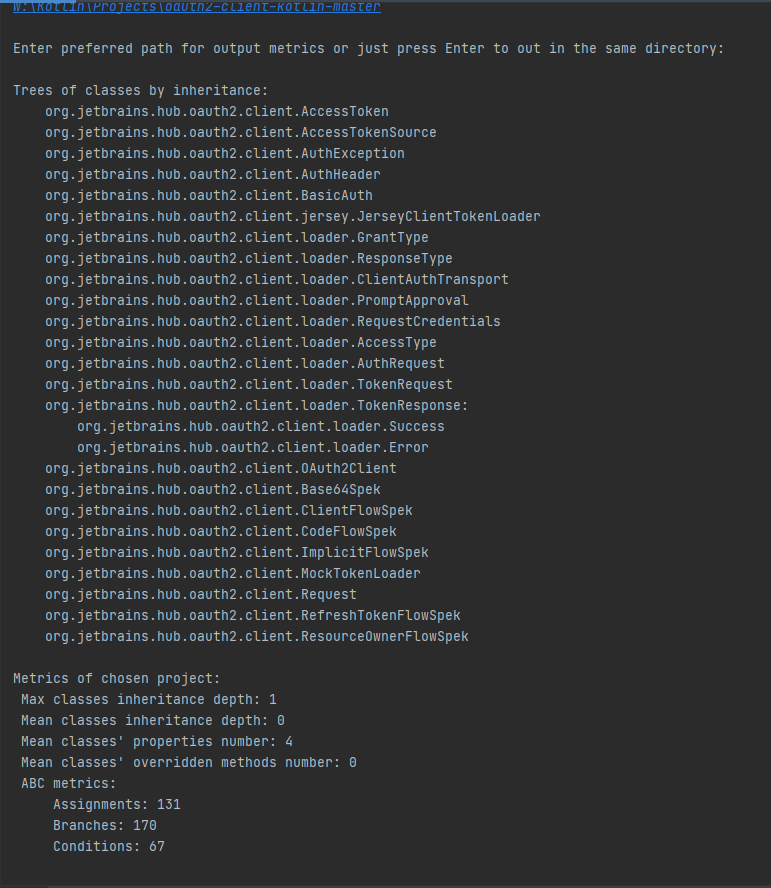


Результат работы весьма адекватный.

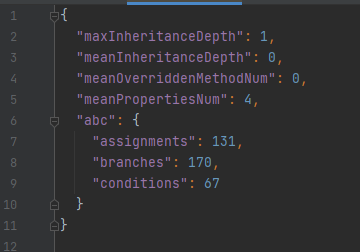
1. “Простой OAuth 2 клиент”: <https://github.com/mazine/oauth2-client-kotlin>

В данном проекте практически отсутствует структура наследования

Результат работы программы:



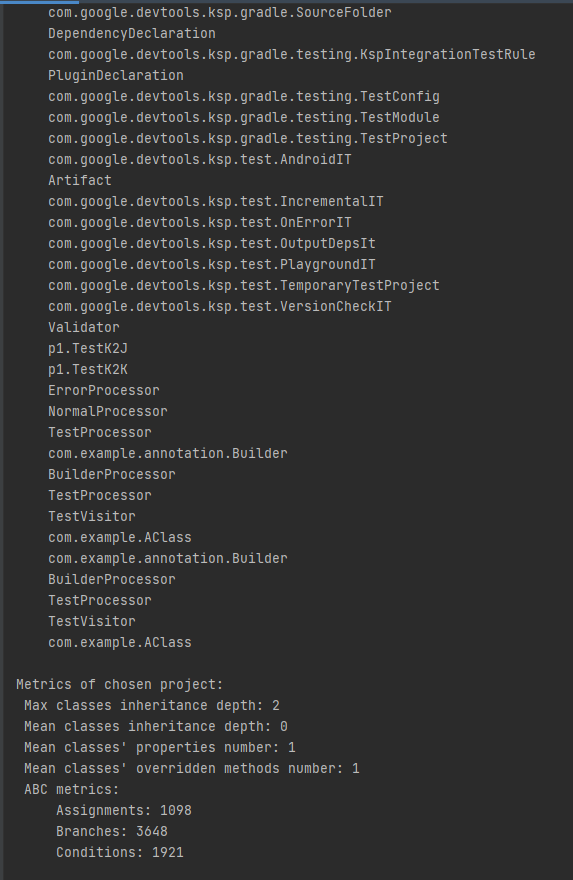
JSON:



Хоть мне и не удосужилось лично посчитать все свойства классов и произвести расчеты ABC-метрики вручную, результат работы программы выглядит адекватным для такого проекта.

1. Проект Google’a, состоящий из 505 классов: https://github.com/google/ksp

Результат работы программы:



Можно заметить, что глубина наследования слишком мала, исходя из дерева.

JSON:

