

# Прозрачное взаимодействие программ на языке Kotlin с библиотеками на Objective-C

Удалов Александр, 545 гр.

Научный руководитель: Д. Ю. Булычев

Рецензент: А. А. Бреслав

# Введение

- Взаимодействие JVM и Objective-C
  - переиспользование существующих библиотек
  - UI
- Kotlin
  - компилируется в байткод JVM
  - много общего с Objective-C

# Постановка задачи

- Цель: использование библиотек на Objective-C в коде на Kotlin под JVM
  - загрузка деклараций из заголовков
  - отображение в термины Kotlin
  - генерация JVM-классов

# Описание работы

## 1. Загрузка и отображение

- на вход: заголовочные файлы
- разбор Objective-C деклараций с помощью Clang
- отображение типов в типы Kotlin
- построение иерархии классов и методов

## 2. Kotlin Runtime

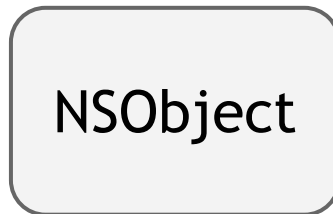
- классы, соответствующие типам Objective-C
- native-код для маршалинга типов и обращения к Objective-C Runtime

## 3. Генерация байткода

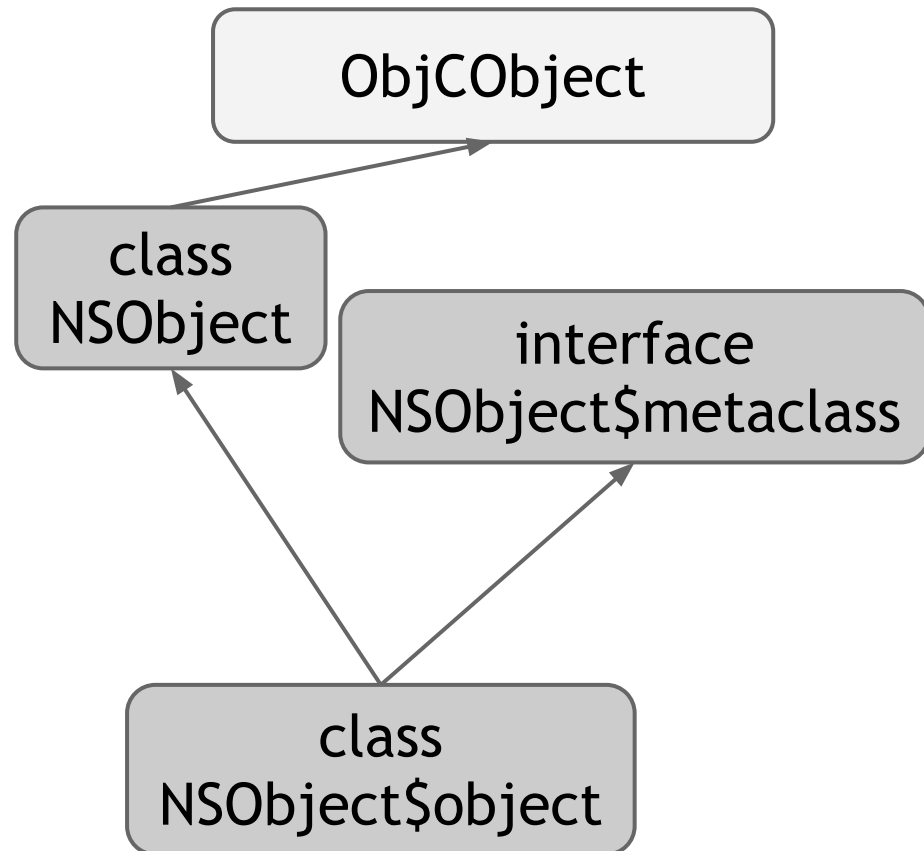
- классы, методы, свойства, константы, ...

# Пример: иерархия

Objective-C



JVM



# Пример: типы

- `int, char, ...` -> тривиально
- Блоки, указатели на функции -> функциональные типы
- `char *` -> `Pointer<Char>`
- `NSObject *` -> `NSObject`
- `id` -> `ObjCObject`
- ...

# Похожие проекты

- JNA

- отображение нативных функций в методы JVM
- не поддерживает Objective-C
- JNAerator

- BridJ

- подобен JNA, но удобнее
- Objective-C в планах

- Rosocoa

- мост между Objective-C и JVM, тоже вручную
- очень молодой проект

# Преимущества

- Автоматический процесс
- Интегрирован в компилятор Kotlin



# Тестирование

- Обширное функциональное тестирование
- Проигрывание аудио с помощью AppKit
- Сравнение с аналогами
  - сопоставимая производительность
  - размер кода значительно меньше

# Результаты

- Модули компилятора Kotlin в байткод JVM
  - загрузка деклараций из заголовков библиотек на Objective-C
  - отображение типов и иерархии в термины Kotlin
  - генерация байткода JVM по построенной модели
- Модуль в Kotlin Runtime, обеспечивающий взаимодействие с Objective-C
- Тестирование реализованной функциональности