

Расширение системы вызовов языка Kotlin

Моклев В. В., федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Научный руководитель – Ульяновцев В. И., к.т.н., доцент университета ИТМО

Введение

В современных языках программирования высокого уровня существует абстракция «вызов метода». Под этим понятием подразумевают вызов функции, которая может зависеть от различных условий. Существует множество механизмов вызова методов. Они различаются скоростью работы и гибкостью в использовании. Самыми известными механизмами являются статический и виртуальный вызовы методов. Наряду с ними существуют динамические вызовы, метод «толстых указателей», вызов метода интерфейса и некоторые другие. Среди прочих хочется выделить отдельно особый механизм – механизм «классов типов». По скорости работы он совпадает с виртуальным вызовом (в некоторых случаях даже может работать со скоростью статического вызова), но при этом предоставляет программисту некоторые гибкие возможности использования, которые приближают его к механизму методов интерфейса. Таким образом, реализовав механизм классов типов в языке, возможно ускорить работу скомпилированных программ по сравнению с использованием интерфейсных методов, сохранив его выразительную способность.

Цель работы

Целью данной работы является эффективная реализация механизма классов типов для компилятора языка Kotlin. Эта цель требует решения нескольких задач, таких как:

- разработка описания механизма классов типов для языка Kotlin,
- дополнение системы типов языка Kotlin для использования классов типов,
- доработка алгоритма вывода типов в компиляторе для новой типовой системы,
- разрешение вызовов с использованием механизма классов типов,
- разрешение зависимостей над ограничениями, связанными с классами типов,
- эффективная реализация разработанных алгоритмов в компиляторе.

Описание предлагаемого подхода

В работе рассматриваются основные требования, которые накладываются на механизм классов типов, для написания реальных программ промышленного масштаба. Для этого производится сравнение с существующими механизмами классов типов в других языках программирования, таких, как Scala, Rust, Haskell и некоторых других. На основе этого анализа формируются базовые требования к механизму классов типов. Далее рассматриваются реальные сценарии использования, а также продумывается примерная реализация в компиляторе для того, чтобы наложить некоторые ограничения на механизм классов типов, без которых практическая реализация такого механизма в компиляторе была бы сопряжена с существенными трудностями, например, резким падением производительности.

После формирования всех требований к механизму классов типов для него реализуется прототип. Этот прототип используется для выявления недостатков при помощи реализации нескольких реальных проектов и использованием нового механизма. После этого реализуется окончательный вариант компиляции механизма классов типов.

Результаты

На данный момент проработаны базовые требования и ограничения чна механизм классов типов для использования в языке Kotlin. Для сформулированных требований прототип находится в разработке.

Список литературы

1. Aho A. V., Sethi R., Ullman J. D. Compilers, Principles, Techniques //Addison Wesley. – 1986
2. Hall C. V. et al. Type classes in Haskell //ACM Transactions on Programming Languages and Systems (TOPLAS). – 1996. – Т. 18. – №. 2. – С. 109-138.
3. Sozeau M., Oury N. First-class type classes //International Conference on Theorem Proving in Higher Order Logics. – Springer, Berlin, Heidelberg, 2008. – С. 278-293.
4. Traits [Электронный ресурс] – <https://doc.rust-lang.org/1.12.0/book/traits.html> (Дата обращения: 19.02.2018).