

Потенциал глубоко неизменяемых объектов в языке Kotlin

**Старцев Леонид Михайлович, Санкт-Петербургский Национальный
Исследовательский Университет Информационных Технологий, Механики И Оптики**

Научный руководитель — Шалыто А.А., д.т.н., главный научный сотрудник ФИТиП

Введение

В последнее время, в связи с широким распространением функционального программирования и распределённых систем, тема неизменяемых объектов становится всё более популярной. Это связано в первую очередь с тем, что такие объекты сильно упрощают жизнь программиста: владение ими можно спокойно передавать между потоками, не опасаясь гонок данных, можно легко кешировать результаты вычислений на неизменяемых объектах, не нужно производить сложную отладку, пытаюсь понять, где изменилась та или иная переменная.

Если неизменяемость объектов встроена в язык, можно строить более понятные интерфейсы программирования библиотек, когда из одного модификатора изменяемости понятно, может ли клиентский код производить запись в переменную библиотеки или нет.

Цель работы

В данной работе исследуются существующие возможности языка Kotlin для обеспечения глубокой неизменяемости объектов, анализируется литература по этой теме, относящаяся к другим языкам, и предлагаются возможности расширения компилятора Kotlin для улучшения существующего подхода к неизменяемым объектам.

Базовые положения исследования

Различают два типа неизменяемости: ссылочную и глубокую. В первом случае ссылка на объект неизменяема, но сам объект может быть изменён, например, ссылка на список. Такой тип не позволяет замену списка на другой, но позволяет добавление или удаление элемента из списка. Глубокая же неизменяемость позволяет доказать, что неизменяемая ссылка указывает на неизменяемый объект, все ссылки которого так же неизменяемые, и т. д., что даёт больше гарантий программисту.

Компилятор Kotlin и виртуальная машина JVM [1] уже предоставляют возможность помечать некоторые ссылки как неизменяемые. В работе [2] было показано на примере языка Java, что неизменяемые объекты можно ввести как надстройку над языком.

В данной работе рассмотрены различные подходы к обеспечению ссылочной и глубокой неизменяемости объектов в различных языках программирования, как статических, так и динамических [3,4]. Также рассмотрены способы введения глубокой неизменяемости объектов в язык Kotlin. Предлагаемым способом для этого является использование существующего механизма расширения компилятора.

Результаты

В рамках работы были определены места в языке, которые могут быть подвергнуты улучшению с точки зрения неизменяемости объектов, а именно:

- Возможность указывать изменяемость объекта в аргументах функции
- Возможность указывать, изменяет ли метод объекта его экземпляр
- Возможность эффективно копировать и редактировать неизменяемые объекты, если в них нужно вносить глубокие изменения.

Помимо этого, были выдвинуты требования к решению:

- Гарантии проверки во время компиляции
- Гарантии во время выполнения программы
- Полная совместимость с существующим кодом или максимально автоматизируемый способ миграции.

Список литературы

1. Lindholm T. et al. The Java Virtual Machine Specification, Java SE 8 – Addison-Wesley Professional, 2014.
2. Tschantz M. S., Ernst M. D. Javari: Adding Reference Immutability to Java // SIGPLAN Not. - New York, NY, USA, 2005, Окт.
3. Haack C., Poll E. Type-based object immutability with flexible initialization //European Conference on Object-Oriented Programming. – Springer, Berlin, Heidelberg, 2009. – С. 520-545.
4. Leino K. R. M., Müller P., Wallenburg A. Flexible immutability with frozen objects //Working Conference on Verified Software: Theories, Tools, and Experiments. – Springer, Berlin, Heidelberg, 2008. – С. 192-208.