

Java-to-Kotlin 変換における柔軟なオプション指示を可能にする仕組みの設計

米田友貴† 川端英之‡ 弘中哲夫‡

広島市立大学情報科学部情報工学科† 広島市立大学大学院情報科学研究科‡

1 はじめに

近年、Android アプリ開発において、既存の Java プログラムを Kotlin プログラムに移行する流れが強まっている。そこで我々は、ユーザが Kotlin 特有の機能を選択的に取り入れ可能な、Java から Kotlin 変換器を開発することで、ユーザ支援を行なっている。しかし、プロトタイプでは、変換の仕様が不十分な点が存在していた。また、変換時のオプション適用範囲を指定することができないなど、柔軟性に欠けていた。

対して本研究では、不十分な変換である、アクセサメソッド変換に関する仕様の再構築を行なった。また、オプション適用にあたって、コメントやファイルから詳細な指示を与える機能の開発を行なった。さらに、適用の成否のサマリーを出力することも可能にした。

2 J2KConverter

2.1 J2KConverter の概要

J2KConverter では、基本変換として Kotlin 特有機能の適用を最低限に留めた変換を提供している。加えて、ユーザが適用したい Kotlin 特有機能をオプションとして指示し、適用することで、より Kotlin らしいコードへの変換を実現する方針を執っている [1]。

入力は、変換対象の Java ソースファイルが入ったディレクトリの絶対パスである。変換結果は、入力ディレクトリの直下に、変換後の Kotlin ソースファイルが入った「outkt」という名前のディレクトリを出力する。

2.2 拡張機能

提供されている拡張機能には、non-null 化とカスタムアクセサ化が存在するが、カスタムアクセサ化に絞って説明する。Kotlin には、クラス内に宣言されたプロパティに対して、アクセサメソッドを暗黙的に作成する機能がある [2]。また、アクセサメソッドに任意の処理を追加したい場合、カスタムアクセサを記述することで実現できる。基本変換では、全てのアクセサメソッドを通常メソッドとして変換する。対して、カスタムアクセサ化では、可能であると判断できるメソッドを、カスタムアクセサとして変換する。図 1 は、カスタムアクセサの例である。

```
class Person{
    var name: String? = null
    get(){
        return field
    }
    set(name: String?){
        field = name!!
    }
}
```

図 1: カスタムアクセサの例

2.3 プロトタイプにおける改善すべき点

プロトタイプでのアクセサメソッド変換は、基本変換では全て通常メソッド変換、カスタムアクセサ化では全てカスタムアクセサ変換となっており、基本的な枠組みの作成に留まっている。そのため、枠組みを活用した新たな設計による、実用性の向上が必要である。

柔軟なオプション指示も課題である。現状では、拡張機能をプログラム全体に適用か否かの択一的方法を採用している。ユーザには、将来的に処理を追加するアクセサメソッドのみのカスタムアクセサ変換というニーズも存在すると考えられる。そのため、変数単位での詳細な指示を与える機能があれば、多様なニーズに対応できる。

また、一度に多くの Java ファイルを変換する場合、拡張機能の適用の成否を 1 つのファイルにまとめて出力する機能がなければ利便性の向上につながらず。

3 改善案の設計と実装

3.1 アクセサメソッド変換の再構築

新たに設計したアクセサメソッド変換の仕様を以下に示す。標準アクセサは、単に値を Get, Set するアクセサメソッドを指し、処理付きアクセサは、任意の処理が追加されたアクセサメソッドを指す。

基本変換

- 標準アクセサが存在 → アクセサメソッド出力なし
- 処理付きアクセサが存在 → 名前を変更し通常メソッド出力。メソッド名の変更をユーザに知らせるため warning 出力

カスタムアクセサ化

- 標準アクセサ → カスタムアクセサ化
- 処理付きアクセサ → カスタムアクセサ化し、処理付きアクセサをカスタムアクセサ化したことを知らせるため warning 出力

Design of a mechanism that enables flexible option instructions in Java-to-Kotlin conversion

Tomoki Yoneda† Hideyuki Kawabata‡ Tetsuo Hironaka‡

†Department of Computer and Network Engineering, Hiroshima City University

‡Graduate School of Information Sciences, Hiroshima City University

3.2 詳細なオプション指示への対応

変数単位で拡張機能指示を与える機能として、次の2つの機能の設計、実装を行なった。

ソースコード内コメントによる局所的指示

コメントによる指示は、Java コード中の変数の直前に「-ca」とコメントを記述することによりカスタムアクセサ化指示を行う機能である。コード中に直接記述するため視覚的にわかりやすい点が特徴である。しかし、多量の変数に対する指示には向いていない。

設定ファイルを用いた大域的な一括指示

ファイルによる指示は、JSON 形式の指示ファイルに変数単位の詳細な指示を記述することで、拡張機能を適用する機能である。JSON 形式とは、データ表現のためのテキストベースの形式である [3]。入力ディレクトリ内に「config.json」という名前で指示ファイルを作成する。1つのファイルにまとめて記述するため管理しやすく、多数のフィールドに対しての指示に適している。

3.3 変換に関する結果や警告の出力

変換に対する警告と拡張機能の適用の成否についてのサマリーを出力する機能を開発した。図2は、指示ファイルと結果ファイルの入出力のイメージ図である。指示ファイルと同様にJSON形式で、outktディレクトリ内に「result.json」という名前で出力する。

表1は基本変換、表2はカスタムアクセサ化に対する結果出力表である。result キーは、カスタムアクセサ化の適用結果が入り、warning キーは、ユーザへの警告文が入る。基本変換では、カスタムアクセサの出力はないため、result キーに結果は出力されない。処理付きアクセサが存在する場合は、warning キーにてメソッド名の変更を警告する。カスタムアクセサ化適用の場合、カスタムアクセサ変換の成否を result キーにて出力する。処理付きアクセサが存在する場合、処理付きアクセサのカスタムアクセサ化を警告する。

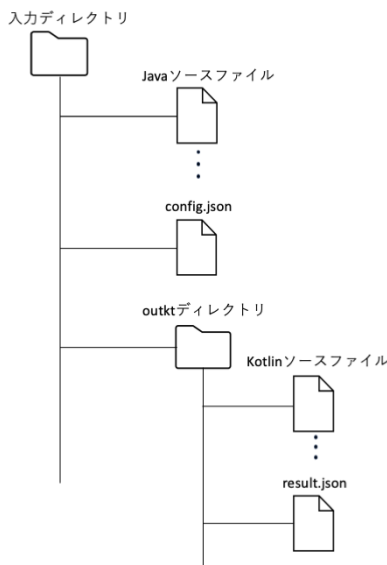


図2: 入出力イメージ

表1: 基本変換における結果出力表

	標準アクセサ	処理付きアクセサ	アクセサメソッドなし
“result”キー	出力なし	出力なし	出力なし
“warning”キー	出力なし	“changed name to ○○ because XX is not NormalAccessor”	出力なし

表2: カスタムアクセサ化における結果出力表

	標準アクセサ	処理付きアクセサ	アクセサメソッドなし
“result”キー	“CustomAccessor Success”	“CustomAccessor Success”	“failed because accessor dose not exist in Java”
“warning”キー	出力なし	“○○ is not Normal Accessor”	出力なし

4 評価

実際にアクセサメソッドが存在する Java コードを変換し、適切に出力がされているかを確認することで評価を行った。評価に使用する Java コードでは、標準アクセサを保持したフィールドと、処理付きアクセサを保持したフィールドをそれぞれ用意した。その Java コードに対して、以下の3つの変換を実行し、Kotlin コード、結果出力ファイルの評価を行った。

1. 基本変換
2. 標準アクセサを持つ変数に対してのみ、コメント指示によるカスタムアクセサ化
3. 処理付きアクセサを持つ変数に対してのみ、ファイル指示によるカスタムアクセサ化

結果、いずれの変換においても、仕様通りの変換が行われることを確認した。結果ファイルに関しては、基本変換、ファイル指示では、Kotlin コードと対応していることを確認した。しかし、コメント指示での result 出力が現状では未対応となっている。

5 まとめ

本稿では、J2KConverter のプロトタイプにおけるアクセサメソッド変換に関して、新たな仕様の設計を行なった。さらに、変数単位の柔軟な拡張機能の適用を可能にし、変換結果と警告を出力する機能を開発することで J2KConverter の利便性を向上させた。

参考文献

- [1] 河野一真, 川端英之, 弘中哲夫. Kotlin 特有の機能を選択的に導入できる Java-to-Kotlin コンバータの設計と実装. 第85回全国大会講演論文集. 2023, vol.1, p303-304
- [2] JetBrains. Kotlin v1.9.22 docs kotlin foundation: Properties. <https://kotlinlang.org/docs/properties.html>. (accessed 2024-01-09).
- [3] Mozilla Corporation. MDN Web Docs:JSON の操作. <https://developer.mozilla.org/ja/docs/Learn/JavaScript/Objects/JSON>. (accessed 2024-01-09)