

Java メソッドの純粋性の変化と不具合の混入および除去に関する調査 A Study on Relationship between Changing Purity of Java Methods and Inducing/Fixing Bugs

小倉 直徒[†] 楊 嘉晨[†] 堀田 圭佑[†] 肥後 芳樹[†] 楠本 真二[†]
Naoto Ogura Jiachen Yang Keisuke Hotta Yoshiki Higo Shinji Kusumoto

1. まえがき

関数型言語において関数の純粋性は重要な特性の 1 つである。純粋な関数をオブジェクト指向型言語のメソッドに適用したものを“純粋なメソッド”とよぶ。メソッド内外の状態を変化させない純粋なメソッドは、並列実行による競合といった問題が起こらないため並列で実行されるメソッドの実装において有用である。また、equals メソッドや hashCode メソッド、getter メソッドなど呼び出しによってインスタンスのメンバ変数が変化しないと開発者が推測しているメソッドもある。

本研究では純粋なメソッドが純粋でなくなることに起因する不具合を検証するため、2 つのオープンソースソフトウェア(OSS)のリポジトリを分析した。リポジトリに格納された各リビジョンの全メソッドの純粋性の推定を行い、コミットによる純粋性の変化を検出した。全メソッドのうち純粋でないメソッドに変化したものを対象に、開発履歴における全ての修正内容を目視で判断することで、メソッドが純粋でなくなることによる不具合への影響を評価した。

2. メソッドの純粋性

2.1 副作用

関数型言語において、ある関数の実行前後で関数内外の状態に変化が生じる場合、副作用を持つという。グローバル変数の値の変更や入出力処理を行った場合、その関数は関数外部の状態に変化を与えるため副作用を持つ。また関数内で定義された静的変数を変更する場合も、関数内部の状態に変化を与えるため副作用を持つ。一方、値渡しされた変数の値を変更する場合や、グローバル変数から値を取得するのみの場合、状態に変化を与えないので副作用は持たない。

2.2 純粋なメソッド

副作用を持たないメソッドは、純粋なメソッドである[1]。純粋なメソッドの実行結果は引数として与えられた変数や、メンバ変数、静的変数のみに依存し、またメソッド内外の状態に変化を与えない。

純粋なメソッドは他のメソッドの実行結果に影響を及ぼさないため、並列で実行されるメソッドの実装において有用である。

3. 調査の目的

メソッドの中には純粋であることが望ましいものがある。しかし、これらのメソッドの実装を意図せず修正してしまい、メソッドの純粋性が変化する場合があり、不具合となってソフトウェアの品質に影響を及ぼす可能性がある。

本研究では、ソフトウェアの開発における純粋性の変化を調査することで、純粋性の変化と不具合の混入の関係を明らかにする。本研究の調査項目は以下の 2 点である。

RQ1 メソッドの純粋性が変化することがあるのか

メソッドの純粋性が変化する修正が行われることがあるのかを調査する。変化することが確認できた場合、加えてメソッドの純粋性の変化の回数と傾向について調査する。

RQ2 どのようなソースコードの修正によって、メソッドの純粋性が変化したのか

メソッドの純粋性が変化した際に行われた修正内容を分類する。また修正と同時に不具合が混入したのかを確認し、純粋性の変化が不具合に及ぼす影響を調査する。

4. 調査手法

4.1 RQ1 に対する調査手法

以下の 3 つのステップによって、開発履歴におけるメソッドの純粋性の変化を検出する。

ステップ1 Java バイトコードからメソッドごとの純粋性を推定

Purano[1]は Java バイトコードを解析し、メソッドの純粋性を推定する。自動ビルドツールを用いて各リビジョンをビルドして得たバイトコードを Purano に入力し、メソッド純粋性の情報を得る。

ステップ2 メソッドの系譜を検出

コミットの前後でメソッドのシグネチャが変化した場合でも、同一のメソッドであることを特定する必要がある。メソッドが実装されてから削除されるまで追跡したものをメソッドの系譜と呼ぶ。メソッドの系譜の検出には ECTEC[2]を用いる。

ステップ3 純粋性が変化したメソッドの抽出

ステップ 1 とステップ 2 の出力を用いて、各系譜の全リビジョンに純粋性の情報を付与する。各系譜において純粋性が変化したコミットおよび系譜を特定する。

4.2 RQ2 に対する調査手法

全てのメソッドの系譜のうち、あるコミットの前後においてメソッドの修正が行われ、かつその修正によって純粋でなくなったメソッドの系譜を抽出する。このメソッドの系譜全体の修正を調べることで、純粋性を失ったコミットで行われた修正内容について判断する。

4.3 調査対象

本研究では、jEdit と JFreeChart の 2 つの OSS を対象に調査を行った。それぞれのソフトウェアの概要を表 1 に示す。“リビジョン数”は調査対象のファイルが追加または削除、修正されたリビジョンの数を表し、“行数”は終了リビジョンにおける調査対象のソースコードの行数を表す。

[†] 大阪大学大学院情報科学研究科

Graduate School of Information Science and Technology,
Osaka University

これらはいずれも Java で実装されたソフトウェアであり、Subversion を用いて管理されている。Ant または Maven を用いてソフトウェアを自動でビルドすることができる。

5. 調査結果

5.1 RQ1: メソッドの純粋性は変化するのか

2 つのソフトウェアについて、メソッドの純粋性が変化した系譜の割合を表 2 に示す。純粋性の変化は存在した。

表 2 で示した変化した系譜のうち、メソッドが純粋になった系譜の数と純粋でなくなった系譜の数、および 2 回以上変化した系譜の数を表 3 に示す。“2 回以上変化”とは、1 つの系譜の純粋性が 2 回以上変化したことを表しており、純粋への変化と純粋でなくなる変化のどちらも含む。

メソッドの純粋性の変化が、そのメソッドのソースコードの修正を伴うのか調べた。純粋性の変化のうち、純粋性の変化したメソッドとソースコードの修正が同時に行われた回数を表 4 に示す。純粋性が変化する多くのメソッドのソースコードには修正が加えられていないことが分かった。

5.2 RQ2: メソッドの純粋性が変化した際、同時にどのようなソースコード上の修正が行われたのか

メソッドの純粋性が変化した際に行われたソースコードの修正内容を参照し、どのような修正が加えられたのかを目視で判断する。メソッドの純粋性が変化した理由を調査した結果を表 5 に示す。不具合の混入となる修正はほとんど発見されなかった。

jEdit のリビジョン 19698 において HyperSearchResult.java の 75 行目に標準エラー出力が追加された。出力処理の追加のためにメソッドが純粋でなくなったが、その後のコミットにおいてこの行は削除されており、著者らはこの修正内容を不具合の混入と判断した。

6. 結果の妥当性について

6.1 調査手法

メソッドの系譜に用いた ECTEC により、メソッドのシグチャ修正やファイル移動の検出を自動化しているが、検出に誤りが混入している可能性がある。

修正内容の判断は、いずれも著者らによる目視によって行われた。著者らは対象としたソフトウェアの開発者ではないが、今回の調査では、誤りが混入する確率を下げるため、複数人によって確認している。

6.2 実験対象

今回の実験は、2 つの OSS のみを対象としている。この実験の一般性を向上させるためには、他のソフトウェアやライブラリに対して調査を行う必要がある。

7. あとがき

本研究では 2 つの OSS におけるメソッド純粋性の変化を調査し、開発中に純粋性が変化するメソッドがあることを示した。また、純粋性の変化するメソッドの多くは修正を伴わないことを示した。純粋性を失うような不具合の混入が行われていることを見つけた。

謝辞

本研究は、日本学術振興会科学研究費補助基盤 研究(S) (課題番号: 25220003) の助成を得て行われた。

参考文献

- [1] Jiachen Yang, Keisuke Hotta, Yoshiki Higo, Shinji Kusumoto, “Revealing purity and side effects on functions for reusing java libraries”, In 14th International Conference on Software Reuse, Vol. LNCS 8919, pp. 314–329.
- [2] Yoshiki Higo, Keisuke Hotta, Shinji Kusumoto. “Enhancement of crd-based clone tracking”, In Proc. of the 13th International Workshop on Principles of Software Evolution (IWPSE2013), pp. 28–37, Aug. 2013.

表 1: 調査対象

ソフトウェア	調査対象	開始リビジョン (日付)	終了リビジョン (日付)	リビジョン数	行数
jEdit	/jEdit/trunk/	3,789 (2001/9/2)	22,016 (2012/8/17)	5,302	183,093
JFreeChart	/branches/ jfreechart-1.0.x-branch/	1 (2007/6/19)	2,527 (2013/1/28)	1,606	323,497

表 2: メソッド純粋性が変化した系譜の数

ソフトウェア	調査した系譜	変化した系譜	割合
jEdit	6,271	331	0.0527
JFreeChart	7,790	55	0.0070

表 3: メソッド純粋性の変化の傾向

ソフトウェア	1 回変化		2 回以上変化
	純粋になった	純粋でなくなった	
jEdit	59	115	157
JFreeChart	37	18	0

表 4: メソッド純粋性の変化とソースの修正

ソフトウェア	修正を伴う	修正を伴わない
jEdit	258	443
JFreeChart	18	37

表 5: 純粋でなくなった修正の内容

ソフトウ エア	機能追加	不具合 修正	リファクタ リング	不具合混入
jEdit	68	34	9	1
JFreeChart	5	5	0	0