

プログラムの内的な振舞いを理解する際のソフトウェア類似性

早瀬 康裕[†]

本論文では、プログラムを理解する手順の観点から、ソースコードの類似性について考察する。理解の手順における類似性の基準として、識別子名に着目した。

1. はじめに

私の主な研究テーマは、ソフトウェアの再利用とソフトウェア保守である。その中で、現在、ソフトウェアを再利用する場合や保守する場合に、開発者が対象プログラムを理解する手順に興味を持っている。ソフトウェアを理解するためには、ソフトウェアの構造や識別子の名前、自然文で書かれた文書やコメントなど、様々な情報が用いられる。ここで、ソフトウェア理解の観点からソフトウェアの類似条件を考えると、同じ手順でソフトウェアの持つ機能や振舞いを理解できることが重要な基準となる。

ソフトウェア開発の経験は、アルバイトとして約5年間と、研究上必要となるプログラムを開発した約7年間の、合計12年間である。そのうち、Javaと関わっていたのは、研究の解析対象およびプログラム開発を行うための言語として、7年間である。また、大学院の授業として、Javaによる開発演習を2年間指導している。

2. アンケート回答

観点Xとして、「プログラムの振舞いを理解する際に、同じ手順で同じように理解できるか」という基準を用いた。

表1に設問への回答を示す。観点Dへの回答は全てyesとした。設問のソフトウェア部品は、全て何らかの観点で類似しており、管理すべきでない理由が無いと考えたためである。

以下、それぞれの設問の、観点A,B,C,Xにおける回答について記述する。

表1 設問への回答

ソース No.	A	B	C	D	X
1	yes	yes	yes	yes	no
2	no	yes	no	yes	yes
3	yes	yes	no	yes	yes
4	yes	yes	no	yes	no
5	yes	yes	yes	yes	no
6	yes	yes	yes	yes	no
7	yes	yes	yes	yes	no
8	no	no	no	yes	no
9	yes	yes	no	yes	yes
10	yes	yes	yes	yes	no
11	yes	yes	no	yes	no

2.1 ソース No.1 についての回答理由

クラス名に全く異なる単語が使われており、開発者がクラスを外部から見たときに、同じ機能を持っていると考える可能性は低い。そのため、理解の手順が異なると考え、観点Xをnoとした。

2.2 ソース No.2 および No.3 についての回答理由

これらのクラス対は、外部から使用可能なメソッドが異なるため、同じように利用することは出来ない。よって、観点Cをnoとした。

2.4 ソース No.4 についての回答理由

観点Cは、IPv6による通信が必要となるソフトウェアでは、4-1を使用できるが4-2は使用できないため、noとした。

観点Xは、実装している機能が異なり、それぞれのコード片の動作を理解するために異なる知識(IPv6とIPv4のループバックアドレス)が必要になるため、noとした。

2.5 ソース No.5 についての回答理由

コード片5-1とコード片5-2で利用している部品は、機能は同じであるが名前やインタフェースが異なる。

[†] 大阪大学
Osaka University

るため、同じ手順で理解することは出来ない。よって、観点 X は no とした。

2.6 ソース No.6 についての回答理由

states フィールドの Vector 型オブジェクトに格納できるオブジェクトが、AccessibleStateSet のみだということが分かるのは 6-2 のみである。よって、観点 X は no とした。

states に AccessibleStateSet 型でないオブジェクトを格納することは不可能ではないが、このクラス的设计意図から外れることは明確である。よって、再利用の観点からは同等であると考え、観点 C は yes とした。

2.7 ソース No.7 についての回答理由

2つのコード片の機能や外部からの使用する際のインタフェースは同一であるため、観点 A~D は yes とした。

一方、理解の観点では、7-2 で行なわれた抽出とメソッドの命名によって処理内容が明確化され、7-1 に比べて容易に理解できるようになっている。よって、観点 X は no とした。

2.8 ソース No.8 についての回答理由

2つのコード片は、処理内容が異なっているため、観点 A~C,X では no とした。

一方、2つのコード片のインタフェースは同一であり、またリストを整理するという目的は共通である。そのため、類似していることを記録して管理する必要は依然として存在すると考えた。

2.9 ソース No.9 についての回答理由

同型のコードであるため、1つに欠陥が見つかった場合には同様の欠陥が潜んでいる可能性が高い。よって観点 B と D は yes とした。一方、異なる型を box 化する役目を持つ2つのコード片は、交換して使用することは出来ないため、観点 C での回答は no である。

2つのコード片で、フィールドやメソッドの役割を比べると、型が異なるという以外の差異はない。よって、観点 X は yes と考えた。この回答は No.1 への回答と矛盾するとも考えられるが、本設問ではどちらの型も整数を表すという共通点があるため、byte と short が整数型の名前であることを知っている平均的な開発者の理解手順に違いが生じることは無いと考えた。

観点 A に対しては同型のコードであるため yes と回答したが、Java の記述能力の限界のため、実際に1つのコードにまとめるのは不可能である。

2.10 ソース No.10 についての回答理由

ソース No.7 と同じ理由により、観点 X のみ no と

した。ただし、No.7 と異なり、適切な単位でメソッド抽出が行なわれておらず、また、抽出したメソッド名も適切でないため、可読性が低下していると考ええる。

2.11 ソース No.11 についての回答理由

クラス名も内部の処理も異なるため、観点 X は no とした。同様の理由で、同じように再利用することは出来ないため、観点 C も no となる。

観点 A は、ソース No.9 と同様である。

3. 議 論

観点 C のように、外部からクラスを利用する観点では、そのクラスが外部に対して同じインタフェースを提供している事が重要となる。つまり、メソッドの集合と、それぞれのメソッドの事前条件と事後条件が等しい必要がある。

一方、観点 A,B,D では、インタフェースよりも、メソッドの中身が類似しているかどうか重要となると考えられる。特に、欠陥を調査すべきかどうかや、リファクタリングの適用を評価する場合には、一部分を除いて同一のコード片が存在することが基準となるであろう。

観点 X は A~D までの基準よりも更に厳しい。具体的には、インタフェースと中身の等価性に加えて、出現する名前やコメント等の、プログラムを実行する際の振舞いに影響を与えない要素も理解の手順に影響を与える。

謝辞 本研究は、文部科学省科学研究費補助金若手研究(B)(課題番号:21700031)の助成を得た。