

Java プログラミング初学者への 単体テスト方法学習支援ツールの適用と考察

上河内 頌之[†] 松浦 佐江子[‡]

芝浦工業大学大学院 工学研究科 電気電子情報工学専攻[†]

芝浦工業大学 システム工学部 電子情報システム学科[‡]

1. はじめに

従来のプログラミング演習ではコードの作成に重点が置かれており、作成したコードをテストする学習がほとんど行われていない。そのため、プログラミング初学者には『テストの方法が分からない』、すなわち『テストをどこから、どのように、どの程度行うのか分からない』という問題点が生じている。著者らは Java プログラミング初学者(以下学習者とする)を対象に一般的なソフトウェア開発工程のテストフェーズの内、最初に行う単体テストフェーズに対するテスト方法学習支援を提案、ツールとして実現した[1]。本稿では Java 言語を 1 ヶ月程度学習した本学 2 年生 25 名に提案する支援を適用した結果を報告する。

2. 単体テスト方法学習支援

2.1 単体テスト方法学習支援モデルと学習目標

図 1 に提案する単体テスト方法学習支援モデル[1]を示す。まず単体テストプロセスを『テストするメソッドの決定・テストデータ作成・テストプログラム作成と実行』の 3 ステップで定義する。単体テストとは、この 3 ステップを図 1 の手順で全てのメソッドに対し繰り返し行って完了する事を学習させる必要がある。各ステップではテスト対象プログラムから取得した情報と指導者が用意する情報をヒントとして与える。指導者は課題毎に全クラスのコンストラクタの引数に設定する値の満たすべき条件、全メソッドに対するテストケース名と期待する結果、解答となるプログラム[1]を用意する。例えばメソッド決定ステップではプログラムから取得したメソッド呼び出し回数や親クラス名等をヒントとして提供し、学習者にメソッド選択を行わせる事で『どこから』テストすべきかを学習させる。また図 1 のプロセスで学習者が誤った選択を行った、並びにテストプログラムを正しく実行し各ステップに戻る際、誤りの原因や各ステップに戻る理由を示す事で学習者を誘導した。

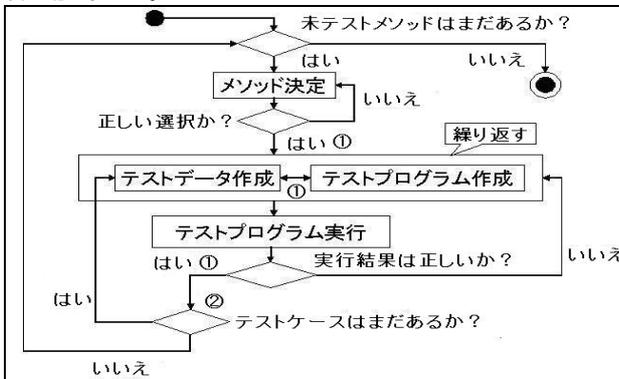


図 1. 単体テスト方法学習支援モデル

Development and Evaluation of a Learning Support Tool for Testing Method of Java Programs

[†] Kamigochi Nobuyuki ; Shibaura Institute of Technology graduate school engineering research course

[‡] Matsuura Saeko ; Shibaura institute of technology

Department of electronic information system

本支援の学習目標は下記の作業を学習者自身で行なえるようになる事である。

- ・テストするメソッドを適切に決定出来る
- ・テストケースに沿ったテストデータが作成出来る
- ・テストプログラムとは何かを理解し、作成出来る

2.2 単体テスト方法学習支援ツール

提案モデルに従い単体テスト方法学習支援ツール[1]を作成した。メソッド選択やデータ作成の作業を集中し易くするため作業スペースとして一箇所にまとめ、ツール中央に配置した。また作業内容、ヒント情報、ソースコードを作業スペースの周りに配置し、それらをステップ毎に 1 画面で表示する事で、各ステップで必要な情報を参照しながらテストが行えると考えた。図 2 にツール画面を示す。

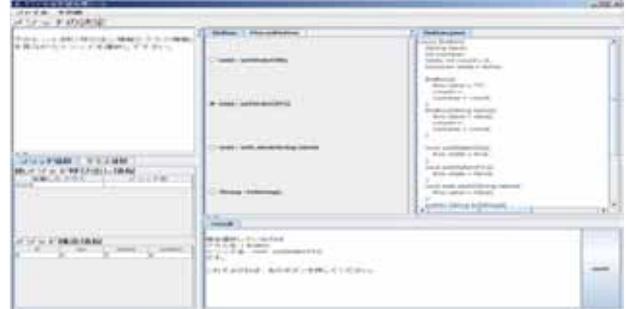


図 2. 学習支援ツール画面

また、ツール化にあたり以下の機能を実装した。

1. 理解度によるヒント量の変更

学習者によって各ステップの内容の理解に差が生じる。本ツールでは理解度に応じて各ステップで示すヒントの量を、ソースコード上で色つきで表示する等変更出来るようにした。その際図 1 中の でステップ終了毎に自身の理解を確認させ、学習者自身で変更させるようにした。

2. テストの進行状況の確認

『どの程度』行えばよいかを把握させるため、テストの終了状況を適時確認出来る事も必要である。本ツールでは図 1 中の 及びメニューから進捗状況を確認出来る。

3. ツール適用実験

3.1 課題並びに指導者の準備

本学のプログラミング演習で用いた 3 つの課題を対象とした。課題 1 は「クラスの定義」を学ぶ課題で、テストデータ、テストプログラムの作り方の学習を目的とした。課題 2 は「合成によるクラスの拡張」を学ぶ課題、課題 3 は「継承によるクラスの拡張」を学ぶ課題である。これらの課題はオブジェクト指向の特徴である複数のクラス定義を行う場合等テストを行うメソッドの選択順序の学習を主な目的とした。表 1 に用いた課題のクラス数、メソッド数、行数を示す。なお、行数の計測には教員が用意する解答のプログラムを用いた。また指導者は 2.1 で述べた情報を各課題に対して準備した。

表 1. 用いた課題のクラス数、メソッド数、行数

| 課題 | クラス数 | メソッド数 | 行数 |
|----|------|-------|----|
| 1 | 1 | 5 | 47 |
| 2 | 2 | 7 | 80 |
| 3 | 2 | 7 | 85 |

3.2 実験概要

本学の Java 言語のプログラミング演習履修者(学習し始めて 1 ヶ月程度の学生)にツールのモニター希望調査を行い、希望者 25 名中の 17 名に対しツールの使用方法の説明を行った。更に、3.1 で述べた課題に対し作成済みの自分のプログラムと 3.1 で指導者が準備した情報を用いてツールを実行する事を指示し、以下の 3 種類のデータ、回答の提出を求めた。この実験は下記 3 つの提出物から 2.1 節の学習目標を達成したかを明らかにする目的で行った。

- ・ **ツール使用ログ:**
選択したメソッド名と正誤、作成したテストデータ、テストプログラム、各ステップ終了時の理解度の記録
- ・ **アンケート:**
理解度による支援内容の変更について、各ステップで行う作業、ツール・実験方法について等、計 31 項目
- ・ **小テスト:**
メソッドのテスト順序の決定、テストケースに沿うテストデータ作成、テストプログラム作成の計 3 問。

3.3 評価

本研究の評価は 3.2 の 3 つの提出物を全て提出した 5 名のデータを用いて、2.1 節で述べた学習目標、ツール、実験方法自体の 3 つに対して行う。学習目標はツール使用ログを課題 1~3 の順に経過を分析し、その後小テスト、アンケートの結果により評価する。またツール、実験方法はアンケート結果、使用ログを用いて評価する。

3.3.1 学習目標に対する評価

[テストするメソッドが適切に決定出来る]

使用ログによると、1 名だけメソッドを誤って選択し続けていたが、残りの学生は最初から常に正しい選択を行っていた。しかし小テストの解答では全員が正解しており、ツールによる学習と小テストの結果の因果関係を結び付けられなかった。またアンケートの「メソッド決定時に着目すべき情報は何か」という質問に対し、3 名の学生がメソッド内で呼び出しているメソッド等ツールで誘導した解答だった。残り 2 名は手当たり次第に選んだ、適当に選んでいたとの回答だった。結果、指導内容を理解していない学生でも正解する小テストを行ったのが問題だったと考える。学習効果を明確にするためにメソッドやクラスを多数定義し順序を答えさせる問題を用いる必要がある。

[テストケースに沿ったテストデータが作成出来る]

使用ログでは、課題 1 のあるテストケースにおいて引数値のデータを整数値で作成するはずが、文字列型のデータとして作る等、妥当でないテストデータを作る学生が 4 名もいた。しかし、課題 3 の時には全員そのような回答もほぼ無く作成出来ていた。また小テストの解答では 4 名正解しており、残りの 1 名も記述方法を間違っているだけで作成したデータは正しかった。この結果から与えられたテストケースに沿ったテストデータが作成出来るようになったと考えられる。またアンケートの「テストケースの選択、テストデータに自オブジェクトのフィールド値が必要な意味は何か」という質問に対し、ツールの使い方だから、指示された通りという回答が多かった。これらから、自らテストケースを選択する意識やテストデータとして何を

作成すればよいかという事についての学習が不足だった。このためテストケースやどの変数のデータが必要かを考えさせ作らせる作業により学習させる必要がある。

[テストプログラムとは何かを理解し、作成出来る]

使用ログでは、課題 1 では全員が assert 構文の使い方を何度も間違えていた。また課題 3 では、2 名は正しく作成出来るようになっていたが、残りの 3 名は間違いを繰り返していた。小テストの結果では正しく作成出来るようになっていた 2 名しか正解せず、間違った回答ではやはり assert 構文を誤って使用していた。この結果から assert 構文に対する理解が浅くなり、指導が不十分であったと考えられる。またアンケートの「assert 構文を理解したか」との質問に対し、全ての学習者がよく、大体分かったと解答しており学習者の意識と結果にズレが生じていた。これらの結果から assert 構文の簡単な例題を繰り返す等確実に理解させる学習が必要である。

3.3.2 ツール、実験方法に対する評価

使用ログではテスト終了時、全員理解度を変更せずに終了していた。アンケートでの理解度による支援内容の変更について質問した所、『いちいち聞いてくるのが鬱陶しい』等の解答があった。またツールに対し『何度も同じテストデータやプログラムを作るのが面倒臭い』『メソッドを決定した後もメソッドの決定画面に戻れたりして欲しい』といった意見があった。更に最初に説明した 17 名中 5 名しかデータが返ってこなかった、実験開始前に十分な予備実験を行ってこなかった等から、「提案したモデルの制限が厳しすぎた事、何度も同じ事を聞いた事から生じたツールの自由度の低さ、使用感の悪さ」から学習意欲が下がってしまったと考えられる。そのためテストデータ作成ステップやテストプログラムを実行後、メソッド選択に戻れるようなモデル変更、ツールインターフェースの変更を行う。その後十分な予備実験を行い、インターフェースを洗練し、作成すべき変数も定義させるテストデータ作成機能等を追加する必要がある。

ここで、テスト駆動開発によりテスト実行能力等プログラミングスキルの向上を促している研究[2]では、学習者が作成したテストデータを用いて指導者が作成したプログラムを実行する等、学習者の作成したテストデータ、テストプログラムの正当性をチェックする手法を提案している。この手法を用いれば作成すべき変数も含めてテストデータを作成するとは何かを学習する事が出来る。

一方、実験開始時にツールの使用方法の説明だけで終了した事も問題だった。メソッドやテストケースを選ぶ理由を説明しておけば、学生の選択や作成に対する意識が働き異なる結果が得られたと考える。

4. おわりに

本稿では、単体テスト方法学習支援ツールを適用し、評価を行った結果、検証データは少ないが学習効果を得られるものがあった。しかし、3.3 節で述べた問題点を改善し、単体テスト学習支援ツールとしての完成度を高め機能を拡張する必要がある。

参考文献

- [1] 上河内頌之, 松浦佐江子: “Java プログラミング初学者に対するテスト方法学習支援ツール”, 電子情報通信学会技術研究報告, ET2006-57, pp37-42, 2006
- [2] 吉田英輔, 角川裕次: “テスト駆動開発に基づくプログラミング学習支援システム: 初心者開発者のためのセルフトレーニングアーキテクチャ”, 電子情報通信学会技術研究報告, S2005-44, pp27-32, 2005