

ステップアップ学習を用いた初学者向け Java 単体テスト学習支援

Learning support for Java unit test for initial scholars using step-up learning

片桐 健伍^{*1}, 酒井 三四郎^{*2}
Kengo KATAGIRI^{*1}, Sanshiro SAKAI^{*2}

^{*1, *2} 静岡大学情報学部

^{*1, *2}Faculty of Informatics, Shizuoka University

Email: katagiri@sakailab.info

あらまし：初学者にとって Java 言語を学習するにあたって例外処理や仕様にあったメソッドの実装は難しい。これらは、いずれも単体テストによって期待する結果と実際に出力される結果を結びつけることで改善を期待できる。これを確認するために学習支援ツール「JUCES」を開発し、実験した。結果は、JUCES を使用しなかった場合との有意差は確認できなかったが、一定の効果をえられることが分かった。

キーワード：Java, JUnit, プログラミング学習支援, ステップアップ学習

1. はじめに

初学者が Java 言語を用いてプログラミングを学習するにあたって課題がある。例えば、例外処理や仕様に合うメソッドの実装が難しいという問題が考えられる。これらは、いずれも単体テストなど期待する結果と実際に出力される結果を結びつける手段があれば改善することができる。しかし、初学者にとって単体テストを実装するのは容易ではない。そこで、これを支援し Java プログラミング学習の手助けとなるようなツールを開発した。

2. 先行研究

本研究に対する先行研究として、「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」⁽¹⁾が挙げられる。上河内・松浦の研究では、Java プログラミング初学者に対するテスト学習支援を行うツールについて研究されている。

上河内・松浦の研究では、示されたヒントに従ってテンプレートを編集することでテストコードを実装していた。本研究では、戻り値や期待する結果をツールのテキストボックスに入力することでコードを自動生成する。また、上河内・松浦の研究では、各ステップの支援内容を学習者自身が設定・変更できるようにツールが実装されていた。本研究では、理解度をツールが判断して支援内容を変更する。

3. 提案ツール

本研究では、1 節で挙げた問題点を解決し、初学者の Java プログラミング学習を支援するためにツール「JUCES」を開発した。今回開発した JUCES は次のような機能を持つ。

- テスト対象メソッドの設定、テストケースの設定、テスト結果の出力を GUI で表示
- 設定したテストケースから JUnit コードを自動で生成
- 最終的に初歩的な JUnit コードを自力で実装できるように、徐々に支援する内容を変更するステップアップ型の学習を実装

JUCES は GUI でコードを作成するため、JUnit コードを自力でコーディングする技能が身につかない可能性が考えられる。これに対処するために、徐々に自力での JUnit コードの実装を学習するための支援を理解度に合わせて調整していく「ステップアップ学習」を採用する。初期段階においては説明を多く盛り込むことで実装方法や結果の見方、JUnit の有用性を分かりやすく学習できるようにする。JUCES が学習者の理解が進んでいると判断すると、自力でのテストコード実装を行うように促したり、実際の出力からどのように必要な情報を得るか学習できるように支援内容を変更していく。

JUCES を使用することで例外処理や仕様に合うメソッドの実装ができるようになることを期待している。JUCES によって単体テストを行いながらこれらを実装し、理解の一助になることを期待している。

JUCES の実行例を図 1 に示す。



図 1 JUCES の実行例

JUCES の画面は 3 つのパーツで構成されており、①はテスト対象コードの表示部、②はテストケース設定部、③は実行結果などの出力部である。テストコード表示部には読み込まれたテスト対象のコードが、テストケース設計部は支援度によって表示されるものが異なる。これによってステップアップ学習が行えるようになっている。

4. 評価実験

4.1 仮説

本実験では、JUCES がない状態での学習よりも JUCES を使用した学習のほうが、学習効果が大きい

という仮説の検証を行う。

4.2 実験方法

被験者は学部4年生4人で、被験者を次の通り2群に分けた。

- 実験群：JUCESによってJUnitを学習する。
- 統制群：テキストを読んでコンパイラとJUnitを使ってJUnitを学習する。

本実験は4年生に対して行った。JUCESは、初学者が使用することを想定して開発を行ったが本実験においてはJavaのコードをある程度自分で実装することが求められるため、対象をこのように設定した。また、実験で行ったアンケートからJUnitに関しては被験者全員が「使ったことはない」と解答したため、本実験の被験者として問題ないと判断した。

まず事前テストを行う。その後、各群に分かれて学習を行う。学習教材として、共通のプログラムを使用する。このプログラムを用いて、実験群はJUCESを使って学習し、統制群はプログラムを元に作成したテキストを用いて学習する。学習後に、各群共に事後テストを行う。事前テストと事後テストの難易度はほぼ同じになるように作成している。事後テスト終了後にアンケートを実施し、JUCES・テキストそれぞれの分かりやすさや、JUnitに対する個人の理解度を解答してもらう。実験結果やアンケートから仮説の検証を行い、JUCESの有効性を評価する。

4.3 実験結果と考察

実験結果を表1、表2に示す。

表1 実験結果(テスト全体の平均点:点)

	事前テスト	事後テスト
実験群	59.5	69.0
統制群	34.0	79.5

表2 実験結果(問題ごとの平均点:点)

問題	事前テスト			事後テスト		
	1	2	3	1	2	3
実験群	15.0	17.0	27.5	17.5	15.0	36.5
統制群	6.0	9.5	18.5	29.5	20.0	30.0

結果としていずれの群も事前テストより事後テストのほうが平均点が向上した。

また、各被験者の事前テストと事後テストの点数の差についてt検定を行ったが、有意な差は見られなかった。これについては、被験者が少ないことが理由として考えられる。一方で各テスト中で確認した項目のうち、テスト結果出力を読んでそのテストの成否状況を把握する第3問については、実験群のほうが平均点が高かった。これについては、JUCESの理解度に応じた表示によってより学習効果が高まったのではないかと考えられる。このことから、今回開発したJUCESにおいて一部の項目では一定の学習効果があったのではないかと考えられる。

実験後に被験者に行ったアンケートの「今回の実験を通してJUnitは理解できたか」という問いに対しては、実験群の被験者がいずれも理解できたと解答したのに対して、統制群の被験者は1人のみが理解できたと解答した。これは、JUCESが初学者にとって必要な情報のみに絞って表示したことで、より理解しやすく学習も進んだのではないかと考えた。

また、実験群の被験者はいずれもステップアップ学習による支援内容の変更に遭遇した。このときの、JUCESを使用した被験者の各段階ごとのコンパイル成功率の遷移を図2、図3に示す。

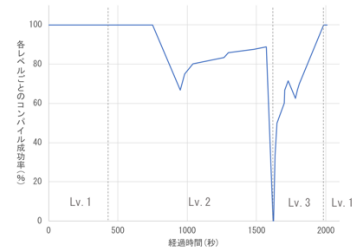


図2 被験者1のコンパイルごとの成功率の遷移

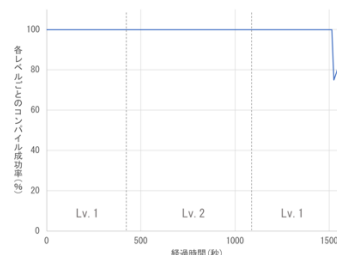


図3 被験者2のコンパイルごとの成功率の遷移

段階が遷移した直後などでメソッドの命名エラーなど小さな問題は発生したもののテストの進行が困難になるような大きな問題は見られなかった。これらのことから、被験者の理解状況に沿ってJUCESが支援内容を変更できていたのではないかと考えた。また、テスト結果の出力方法の違いなどそれぞれの段階固有の機能による学習も行っていたことからステップアップ学習にも一定の効果があるのではないかと考えた。

5. おわりに

本研究では、Java単体テスト学習支援としてステップアップ型の学習を用いたJUCESを開発し、実験を行った。JUCESには明確に有効性があるとはいえなかったが、一定の効果はあることが分かった。今後はJUCESの機能向上や実験内容の見直しによって、さらに有効性の高いツールの提案を行いたいと考えている。

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, pp.37-42 (2006)