

プログラム内容理解のための図を用いた外化による学習手法

Learning Method by Externalization using Diagrams to Promote Program Understanding

疋田 将之^{*1}, 仲林 清^{*2}

Masashi HIKITA^{*1}, Kiyoshi NAKABAYASHI^{*2}

^{*1} 千葉工業大学情報ネットワーク学科

^{*1}Department of Information Network, Chiba Institute of Technology

^{*2} 千葉工業大学

^{*2}University of Chiba Institute of Technology

Email: s1432126ah@s.chibakoudai.jp

あらまし: プログラミングの理解力向上を目的として学習者がプログラム全体の流れやイメージを図として外化する方法を考えた。いきなり図を描くことは困難なことから構文の意味や役割、必要性和クラス間の関係、繋がりを意識させる支援課題を作成した。研究に使用したプログラミング言語はオブジェクト指向の java である。図を描く実験群と描かない統制群の合計 8 人で比較を行った。両群の理解度に大きな違いはなかった。実験群は、図を外化できた学習者とできなかった学習者に分かれ、外化できた学習者には理解度の向上が見られた。

キーワード: 外化, 図, 理解力向上

1. はじめに

プログラミング学習をする上で重要なのは、その内容を理解することである。しかし、学習者の中にはプログラムの内容を理解したつもりとなっていることがある。この原因として構文 1 つ 1 つを単体で理解し、全体の流れを把握できていないこと、それによって構文の役割をプログラム全体の動作の流れと結び付け考えられていないことが主に挙げられる。

この原因の解消する方法として外化に着目した。外化とは認知プロセスの途中で生み出される処理結果を何らかの形で表すことを示す⁽¹⁾。外化を行うことによって自身が考えていることが明確になり、問題を解いていく上でヒントなる。

そこで本研究では、学習者にプログラム全体の動作の流れを図として外化してもらうことで内容理解の向上を図る。

2. 学習目標

本研究では、オブジェクト指向プログラミング(OOP)の java を学習対象とする。OOP のプログラムを理解するためには、個々の構文だけではなく、クラス間の関係性を理解することが必要となる。OOP の理解に関して表 1 のようなプログラム内容理解レベルを設定した。レベル 1 は、構文 1 つ 1 つの意味や役割、必要性をプログラム全体のイメージや動作の流れと結び付けて考えることができない状態であり、レベル 4 は、構文の意味や役割、必要性和クラス間のやり取りと繋がりがプログラム全体のイメージや動作の流れと結び付けて考えることができ、実行結果がある程度予測できている状態である。本研究では、学習者の理解がレベル 4 に到達することを目標とする。

表 1 プログラム内容理解レベル

レベル 1	一つ一つの構文が何をしているかが分かる
レベル 2	プログラム全体から見た構文の意味、役割、必要性が分かる
レベル 3	クラス間、構文同士の繋がりを理解し、正しく繋げることができる
レベル 4	全体としての構文の集合体を正しく繋げて動かすことができる

3. 学習手法

3.1 内容理解レベルを上げる手法

本研究では 2 章で述べた表 1 のレベルに沿って、学習者の理解レベルを向上させることを目標とする。そのために以下 3 つの手法を適応する。3 つの手法を適応した学習の流れを表 2 に示す。

1 つ目は学習者が考えているプログラムの全体のイメージや動作の流れを図として外化してもらうことである。外化は表 1 のレベル 1 とレベル 3 のときに行う。外化に際しては、特に指示を出さず、学習者のプログラムのイメージ図を描かせる。2 つ目は理解レベルを上げる支援課題である。レベルごとに段階を踏んで出題し、学習者が 2 回目のときに図として外化できる状態にするための支援を行う。3 つ目は学習者の理解度を確認する理解度チェック問題である。実行結果の予測やソースコードの一部を改変したプログラムの実行結果の予測、クラス間でやり取りされているインスタンスや変数の中身を問う問題となっている。これらの問題は平成 26 年基本情報技術者試験春期の java のプログラムを使って作成した。

表 2 学習の流れと目的

学習の流れ	目的
1.事前アンケート	プログラムに対する意識調査
2.ソースコード配布	プログラム内で使われているAPIの説明も配布
3.図として外化1 (レベル1)	プログラムの大まかな流れをつかんでもらう
4.支援課題1 (レベル1~レベル2)	構文の意味や役割, 必要性を意識させる(全7問)
5.支援課題2 (レベル2~レベル3)	構文同士やクラス間の繋がりを意識させる(全7問)
6.図として外化2 (レベル3~レベル4)	頭の中を整理させ, プログラムに対するイメージと動作の流れを理解させる
7.理解度チェック問題 (レベル4)	理解度が向上したか判断する(全15問)
8.事後アンケート	図や問題を解いている時に何を意識していたか, またプログラムに対する意識の変化の調査

3.2 問題の分類分け

理解度チェック問題の各問について表 1 のレベルと比較し, 2 つに分類分けを行った.

分類 1 は, 表 1 のレベル 2~レベル 3 相当の問題で, 部分的なプログラムの動作の流れが分かっているならば答えが予測できる問題である. 分類 2 は表 1 のレベル 4 相当の問題で, プログラムの動作の流れや内容を正しく理解できていないと解くのが難しい問題である.

3.3 評価基準

2 回目以外化した図と, 理解度チェック問題の解答を評価した.

図の評価基準は以下の 2 点である.

- (1) メインクラスを中心として見たときの動作の流れやロジックが正しいかどうか
- (2) メインクラスとサブクラスのやり取りや繋がりが正しいかどうか

理解度チェック問題は分類 1 と分類 2 の正答数を比較した.

4. 実験の構成

図を描く実験群と描かない統制群で比較実験を行った. 統制群の学習の流れは表 2 から図として外化する項目を除いたものとなる. 事前問題を作成し, 実験群と統制群の能力が均等になるようにした. 対象者は情報系大学の学部 4 年生 8 人である.

5. 実験結果

実験を行った結果, 実験群と統制群の正答数に大きな差は見られなかった. また, 実験群は図を外化できた学習者 2 名とできなかった学習者 2 名に分かれ, 正答数に大きな差が出た. 両群で, 成績の良か

った学習者の理解度チェック問題の結果を表 3 に, このうち実験群の C が外化した図を図 1 に示す.

C は分類 1 の問題と比べて分類 2 の正答数が多く, 統制群と比べても大幅に分類 2 の正答数が多かった. これは C の図がプログラムのメインとなるロジックを中心した変数のやり取りを表すような内容が描かれており, このような図を描くことによって理解度が向上し, レベル 4 に到達したのではないかと考える.

一方, D は図として外化できていたが分類 1 の正答数が多かった. これは D の図が C と比べてロジックに関わる内容があまり描かれていなかったため, 理解度が向上しなく, レベル 4 に到達しなかったのではないかと考える.

図として外化できなかった学習者はアンケートの結果から支援課題で意図したことが意識できていなかったことが分かり, 支援課題が上手く機能していなかったのではないかと考える.

表 3 理解度チェック問題の正答数

学習者	実験群		統制群	
	C	D	F	H
分類 1	5	6	7	7
分類 2	6	1	3	2
合計	11	7	10	9

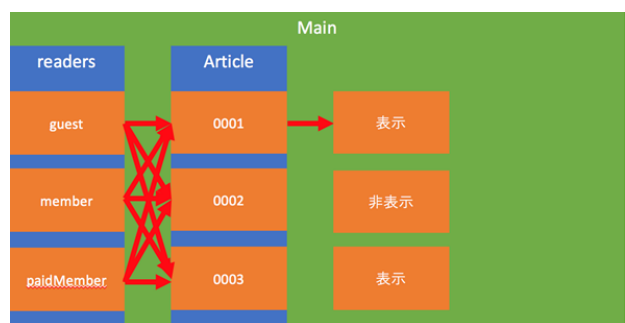


図 1 学習者 C が 2 回目に外化した図

6. 今後の課題

今後の課題として, 図として外化できなかった学習者がでたこと, 図として外化できたが本研究で想定していたレベルまで理解度の向上が見られなかった学習者が出たことからこれらのことについて解決策を考えていく.

参考文献

- (1) 鈴木宏明 教養としての認知科学 東京大学出版会 (2009)
- (2) 宮田仁: “状況論理的アプローチによるプログラミングの指導と認知的徒弟制理論を取り入れたアプローチ”, 教育情報研究, 15 巻, 1 号, P21-31(1999)
- (3) 独立行政法人情報処理推進機構: “問題冊子・配点割合・解答例・採点講評(2014 平成 26 年)” https://www.jitec.ipa.go.jp/1_sukiru/mondai_kaitou_2014_h26.html#26aki