

# オブジェクト指向プログラミングの利便性に着目した 学習手法における理解状況の評価

## Evaluation of Learners' Comprehension Status in A Learning Method for Object-oriented Programming Focusing on its Advantages

竹川 夏実<sup>\*1</sup>, 仲林 清<sup>\*2</sup>

Natsumi Takekawa<sup>\*1</sup>, Kiyoshi Nakabayashi<sup>\*2</sup>

<sup>\*1</sup> 千葉工業大学院

<sup>\*1</sup> Graduate School of Chiba Institute of Technology

<sup>\*2</sup> 千葉工業大学

<sup>\*2</sup> Chiba Institute of Technology

Email: s1332096TH@s.chibakoudai.jp

あらまし：オブジェクト指向プログラミング（以下 OOP）は、機能拡張を前提とするプログラムを開発する際に真価を発揮する。しかし、OOP の基礎概念が抽象的であるという理由で苦手意識を持つプログラミング学習者は少なくない。本研究ではこの問題を解決するため、学習者に手続き型プログラミングと OOP を比較させ、OOP の利便性と基礎概念の関係性を考えさせるという学習手法を提案し、学習者の OOP に対する理解度の改善を図った。

### 1. はじめに

オブジェクト指向プログラミング（以下 OOP）は、機能拡張を前提とするプログラムを開発することに優れた手法である。OOP を正しく利用するには、「クラス」、「継承」、「多態性」などの OOP の基礎概念を理解する必要がある。しかし、現状ではこの基礎概念が抽象的で理解が困難であるため、OOP そのものに苦手意識を持つ学習者が多数存在する。

この問題を解決するための指導方法には、OOP の基礎概念を現実社会の物体に投影させる手法<sup>(1)</sup>、プログラムの動きを可視化する手法<sup>(2-3)</sup>、などがある。これらの手法では、OOP の基礎概念の振る舞いを理解させることはできるが、基礎概念の必要性やそれによって生じる利点を理解させることは困難である。

そこで本研究では、OOP の利便性である「拡張性」、「保守性」を実感させ、これらが OOP の基礎概念によって生じていることを理解させる、というアプローチをとる<sup>(3)</sup>。学習者の OOP の基礎概念に関する理解度を人に説明ができるレベルまで引き上げることを目的とする。また、OOP の利便性を実感させるため、OOP と手続き型で同じ課題を解かせて、プログラミングの手間や複雑さを比較させる。

### 2. 学習目標

学習者に理解してもらいたい OOP の基礎概念と利便性の関係性を図 1 に示す。本研究で着目した基礎概念は「クラス」、「継承」、「多態性」の 3 点である。次に、基礎概念と利便性の関係性をプログラミングの観点から説明する。親クラスで定義したメソッドを、継承先の子クラスでオーバーライドすることで多態性が実現する。この多態性により、プログラムの書き換えの手間を省く「拡張性」と、拡張する際に生じるバグやデバッグの手間が減少する「保守性」が得られる。

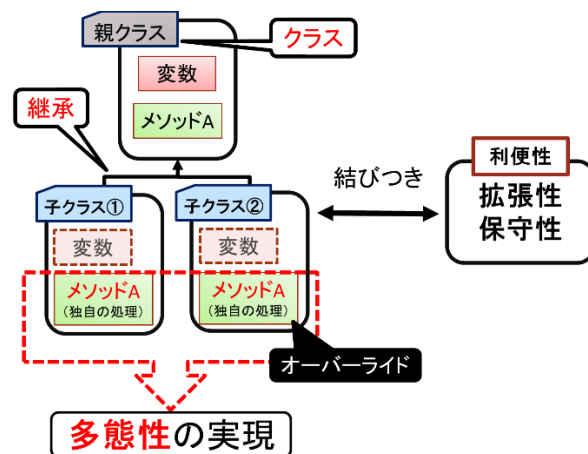


図 1 OOP の基礎概念と利便性の関係性

### 3. 学習手法

本研究では、学習者に手続き型と OOP でプログラムを比較させるという手法をとる。学習者には、OOP と手続き型の両手法で作成した同一のデータ構造を持つプログラムを提示し、拡張させる。その後、それぞれのプログラムにどのような差異が生じたかを確認してもらう。手続き型では、データ分別のために if 文による分岐処理を行うためソースコードが複雑化するが、OOP ではクラス概念を利用してデータ分別をするため、ソースコードが複雑化しないという利点が生じる。

学習課題はショッピングカートの実装である。プログラムの機能要件は 2 点ある。1 点目は、「商品種別ごとの割引率を変えること」である。2 点目は、「割引計算を商品種別ごとのクラスのメソッドのオーバーライドで実現させること」である。この 2 点を実現する際に、OOP と手続き型とでどのような差異が出るかを確認してもらう。

#### 4. 実験結果

対象とした学生は、情報系学科三年生5名である。学習課題は3段階構成であり、各課題でOOPと手続き型の両手法で、データ構造が同一のプログラムを拡張させたあと、記述問題を出題した。学習者のOOPの基礎概念の理解度を調査するため、実験の事前事後にアンケートを行った。また、課題評価はプログラムソースの拡張と記述問題に分けて行った。

前回の研究報告<sup>(3)</sup>では学習者全員の総合的な理解度に着目し、評価を行った。今回は総合的に得点の高かった学習者Cと低かった学習者Dの、ソース拡張と記述問題の回答例をもとに学習者ごとの傾向を評価する。

##### 4.1 比較学習の評価と基礎概念の理解度の改善

表1に、手続き型との比較学習が有意義であったかどうかを評価させた結果を示す。学習者C、Dは両者ともに4点以上の高評価をしていた。表2に、OOPの各基礎概念の理解度に関して、実験の事前事後でアンケートを実施し学習者に自己評価させた結果を示す。学習者Cはどの項目も事前事後で理解度の変化はなかったと回答した。学習者Dは、継承とオーバーライドの概念に関して、事前より事後のほうが理解度が低下したと回答した。

##### 4.2 ソースコードの評価

最後の課題で、学習者Dのソース拡張に誤答が見られた。図2に正解した学習者Cの回答例、誤答した学習者Dの例を示す。学習者DはOOPのソースで継承の手続きを行う際、継承元のクラスの指定を誤り、模範と異なるデータ構造のプログラムを作成していた。具体的には、食品クラス Food を継承するはずの菓子クラス Okashi が、商品クラス Item を継承していたという誤りである。手続き型のソースに関しても、同様の箇所で同じ誤りがあった。

表1 比較に関するアンケート結果 (5件法)

| 手続き型との比較        | C | D |
|-----------------|---|---|
| 比較することで理解は深まったか | 5 | 4 |
| 手続き型との違いは理解できたか | 5 | 4 |

表2 実験前後の学習者の理解度変化 (5件法)

| 学習者           | C  |    | D  |    |
|---------------|----|----|----|----|
|               | 事前 | 事後 | 事前 | 事後 |
| 学習項目          |    |    |    |    |
| クラス           | 3  | 3  | 3  | 3  |
| 継承            | 3  | 3  | 3  | 2  |
| 多態性           | 1  | 1  | 1  | 1  |
| インスタンス        | 3  | 3  | 3  | 4  |
| メソッド          | 3  | 3  | 3  | 4  |
| スーパークラス・サブクラス | 3  | 3  | 2  | 2  |
| オーバーライド       | 3  | 3  | 2  | 1  |



図2 学習者2人のソースの回答例

##### 4.3 記述問題の評価

表3に学習者C、Dの回答例を示す。表3の問は、保守性の特徴と、保守性が何によって生じるかについて記述させたものである。学習者C、Dともに「デバッグがしやすくなる」という保守性の特徴は気づきを得られている。しかし、保守性を実現する概念についての回答を促したところ、学習者Dが不正解となった。このことから、学習者Cは、OOPの基礎概念と利便性の関係性が理解できていると考えられる。学習者Dは利便性の特徴を捉えるのみであり、基礎概念との関係性は意識できていなかった。

表3 学習者の解答例 (記述問題)

|                                                                                             |
|---------------------------------------------------------------------------------------------|
| (問1)保守性が OOP で得られる理由は、デバッグのしやすさが手続き型と OOP でどのように違うからか？                                      |
| (学習者 C)手続き型は追加する部分が多く、追加部分がわかりにくい。デバッグがしにくい。OOP 型は追加する部分が手続き型に比べて少なく、追加部分がわかりやすい。デバッグがしやすい。 |
| (学習者 D)変更箇所が少なくなることでチェックする箇所も少なくなるので OOP の方が優れている                                           |
| (問2)それはどのような概念が OOP にあるから実装されるのか？                                                           |
| (学習者 C)オーバーライド・継承・多態性                                                                       |
| (学習者 D)クラスとインスタンス                                                                           |

#### 5. 考察と今後の課題

今回着目した学習者C、Dは、手続き型との比較学習に意義があるとみなした。しかし、OOPの基礎概念や基礎概念と利便性の関係性の理解度を測ると、学習者Dはほとんど改善されず、理解度が低下した項目も見られた。このことから、比較学習を行うのみでは、理解度に改善がみられる学習者は一部に限られることがわかった。これをふまえ、今後の課題としては、出題する課題の見直しが挙げられる。特にOOPの基礎概念と利便性の関係性を気づかせるために、学習者にどのように働きかけるかをポイントに、課題構成の再検討、難易度調整を行う。

##### 参考文献

- (1) 中鉢直宏, 伊藤一成: “オブジェクト指向プログラミング教育における LEGO を用いた体験型課題の試み”, 情報処理学会研究報告, Vol.2014-CE-124, No.8, pp.1-6 (2014)
- (2) 三浦元喜, 杉原太郎: “オブジェクト指向言語におけるクラス定義の意味とオブジェクトの振舞いを理解するためのワークベンチ”, 情報教育シンポジウム, pp.43-49(2011)
- (3) 石川裕季子, 松澤芳昭, 酒井三四郎: ” オブジェクト指向言語におけるポリモーフィズムの概念を理解するためのワークベンチ”, 教育システム情報学会誌, Vol.31, No.2, pp.208-213 (2014)
- (4) 竹川夏実, 仲林清: “オブジェクト指向プログラミングの利便性に着目した学習手法”, 教育システム情報学会研究報告, Vol.32, No.1, pp.21-28 (2017)