JSiSE Research Report
vol.34,no.2(2019-7)

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

竹川 夏実*，仲林 清**
*千葉工業大学大学院，**千葉工業大学

A Learning Method for Object-oriented Programming Focusing on its Advantages

Natsumi Takekawa*，Kiyoshi Nakabayashi**
*Graduate School of Chiba Institute of Technology **Chiba Institute of Technology

オブジェクト指向プログラミング（以下 OOP）はプログラムの機能拡張に優れる手法だが，苦手意識を持つ学習者が多いとされている．OOP を正しく利用するには，まず OOP の利便性を理解した上で，基礎概念と利便性の結びつきを意識することが有効ではないかと考えられる．本研究では，インストラクショナルデザインの第一原理に則った学習を行う．具体的には，例示課題として，OOP と非 OOP を用いたプログラムの機能拡張を行わせ，両者の比較を通じて OOP の利便性を認識させる．次に，応用課題で，例示課題と異なる機能拡張課題を自力で解かせる．最後に統合課題として，OOP の利便性が基礎概念によって生じていることを一般化して説明させる．OOP と非 OOP の比較を行う実験群と，比較を行わない統制群で実験を行い，実験群が利便性と基礎概念の結び付きをより明確に理解していることが示唆された．

キーワード: オブジェクト指向プログラミング，比較学習，ID の第一原理

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[1][2]．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，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, OOP の基礎概念を現実社会の物体に投影させる手法[3]，プログラムの動きを可視化する手法[4]，クラス図を自動生成しプログラム全体の構造を理解させる手法[5], などが提案されている．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．高井らは，会社の業務のシナリオを想定しながら OOP の学習を実施した研究を行い, OOP を利用することのメリット（利便性）を実感させることを狙った[6]．上記のような手法によって，単純にプログラミングに関する事項を知識として取り入れるよりも深いレベルで，対象のプログラミング手法について理解させる狙いがある．しかし，利便性を実感させ，それが OOP の基礎概念と結びついていることを学習者に理解させるということに関しては実施されていなかった．

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．これを達成するため，本研究では以下の 2 点を実施した．まず，インストラクショナルデザイン（以下 ID）の第一原理に則って学習全体の流れを設計した．学習者には，例示課題と応用課題の 2 回のプログラムの機能拡張を行わせる．例示課題では，拡張のための段階的な指示を出し，応用課題では目標だけを与えて自力で拡張を行わせる．さらに，例示課題では，OOP に加えて非 OOP でプログラムの機能拡張を行わせる．これによって，両者の比較を通じて, OOP の利便性を認識させる．今回の報告では，以上 2 点を実施した際の OOP に関する学習効果を調査した．

以下，第 2 章では本研究の目的を，第 3 章では学習目標を述べる. 第 4 章で学習手法, 第 5 章で学習課題，第 6 章で実験結果について述べる．第 7 章で実験結果に対する考察を行い, 第 8 章で今後の方向性を述べる．

2. 目的

本研究では学習者に，OOP の「拡張性」,「保守性」といった利便性を実感させ，その利便性がクラスや継承，多態性などの OOP の基礎概念によって生じることを理解させることを目的とする. 前回の実験[7]では，ショッピングカートのプログラムを段階的に拡張させる課題を行った. この際, 実験群には OOP に加えて，非 OOP との比較をさせた．また，そこから得た知識を一般化し自分の言葉で説明させた.

実験の結果，実験群，統制群ともにプログラム拡張課題を参照しながらであれば, OOP の利便性と基礎概念の関係性を説明できた．しかし，両群とも，OOP の利便性と基礎概念の関係性を十分に一般化して説明することはできなかった．このことから，「OOP の利便性と基礎概念の結びつきを自分の言葉で一般的に説明させる」レベルに理解度を引き上げるには，段階的な拡張の例示のみでは不充分であることがわかった.

そこで今回は，ID の第一原理に則り，例示課題におけるプログラムの段階的な拡張の後に，応用課題を課した．例示課題では，前回と同様，実験群には OOP に加えて非 OOP でプログラムの機能拡張を行わせた. 応用課題では，実験群・統制群とも，例示課題とは異なるプログラムの拡張を自力で行わせた.

3. 学習目標

OOP の利便性が生じる理由と関連する基礎概念を結びつけた説明ができることを学習目標とし，評価基準を設定した．これを表 1 に示す．表 1 の内容は，前回の報告で使用したものをベースにし，さらに詳細な評価を行うため理解度のレベルを 1 段階追加している. レベル 0 は，機能拡張の課題を全て完了したが，拡張性・保守性といった利便性を体感できない程度の理解度を示す．さらに，レベル 1，2 は，OOP の利便性を実感し（レベル 1)，それが基礎概念と関連していることを具体例で(レベル 2) 説明ができるレベルである. レベル 3 は，OOP の利便性のみ一般的に説明ができるレベルである．レベル 4，5 は，OOP の利便性の特徴とそれが基礎概念のどのような作用から生じるかについて，基礎概念を列挙しながら説明できるレベルである．このうち，説明の抽象度が高いものをレベル 4 とする. 例えば,「各種基礎概念によって拡張がしやすくなる」といった記述が当てはまる. OOP の利便性の実現に基礎概念が関係していることは理解できているが，具体的にどの基礎概念がどのように作用するのかが明示していないためである．これに対し，レベル 5 は，利便性実現に関係する基礎概念の名称を挙げながらレベル 4 よりも具体的に説明ができるレベルである.

表 1 評価基準

要素	内容
レベル 0	OOP の利便性が分からない・実感できない.
レベル 1	OOP と非 OOP の違いが体感的にわかる.
レベル 2	OOP が手続き型よりも便利であると感じ，どのような便利さがあるか，例を用いて簡単な説明ができる.
レベル 3	OOP の利便性について一般的に説明できる.
レベル 4	OOP には 2 つの利便性があり，その利便性は基礎概念と結びついているということが理解できており，プログラム中の一部の基礎概念の作用を取り上げ，簡単な説明ができる.
レベル 5	OOP には 2 つの利便性があり，その利便性は基礎概念と結びついているということが理解できており，具体的な基礎概念を挙げて説明ができる.

4. 学習手法

4.1 基本的な方針

本学習手法の特徴は，以下の 2 点である.

(1) OOP と非 OOP によるプログラムの比較学習により，OOP の利便性を実感させること.

(2) ID の第一原理に則り，応用課題で，OOP の利便性と基礎概念の結びつきの理解を深めさせること.

以下の節でそれぞれの説明を行う.

4.2 比較学習

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. OOP の基礎概念を利用しない非 OOP（手続き型）を比較対象にすることで，OOP の利便性を実感させやすくする.

課題のプログラムは OOP と手続き型の両手法で同一のデータ構造を取っているが，処理方法が異なる. 手続き型では，if 文の分岐処理でデータの種別を分別し，計算や出力等の処理を行っていく．つまり，新しい種別のデータを追加していくほど分岐が増加し，ソ

ースコードが複雑化していく．一方 OOP では，クラスを利用してデータの種別を分別し，計算・出力等の処理も同クラス内で行うためソースコードが複雑化しない．これは，クラス・継承・多態性などの OOP の基礎概念がプログラム中で利用されているからである．この結果として「メインプログラムの拡張が不要になり，拡張が限定的になる（拡張性）」，または「エラーを防いだり，あるいは発生しても発見しやすくなる（保守性）」という OOP の利便性が実現される．

4.3 ID の第一原理

OOP の利便性と基礎概念の結びつきは，OOP の基礎概念をただ知識として取り込むだけでは理解することは困難である．そこで本研究では，学習の流れに ID の第一原理を取り入れている．ID の第一原理とは，構成主義心理学に基づいて近年提唱されている ID モデル・理論に共通する 5 つの要素を指す[8][9]．表 2 に，ID の第一原理が持つ 5 つの要素をまとめたものを示す．「1.問題」では実際に起こりうる問題を学習者に提示する．「2.活性化」では，既有知識をもとに問題に取り組ませる．「3.例示」では，具体例を見せる．「4.応用」では，例示で見せた例とは別の例を示し，取り入れた知識を使う練習をする機会を与える．「5.統合」は，全ての課題の振り返りとして，その成果を確認する．

表 2 ID の第一原理の各要素

要素	内容
1. 問題	これから取り組む課題について問題提起を行う．
2. 活性化	持っている知識を動員して課題に取り組む．
3. 例示	具体的な例題を参照する．
4. 応用	応用的な課題に取り組み，今までで得た知識を使う練習をする．
5. 統合	全ての課題を振り返る機会を設ける．

5. 学習課題

今回の実験で用意した学習課題は，例示課題，応用課題，統合課題である．前回から大きく変更した点は，応用課題の追加である．

5.1.1 例示課題

例示課題は，

1. プログラム拡張の実行
2. プログラム拡張の振り返り

という流れで実行した．まず，「1. プログラム拡張の実行」では，ショッピングサイトのプログラムの機能拡張を実施する．このプログラムは，顧客 1 人 1 人が買い物かごを取り，好きな商品を入れ，レジで精算するという一連の処理を行うものである．図 1 に OOP で作成したプログラムのクラス図を示す．プログラムの機能要件は，商品ごとで異なる割引条件で小計計算・情報出力の処理を行うことである．OOP の実装条件は各クラスのメソッドをオーバライドし，クラスごとに独自の機能を持たせることである．課題実施前はサンプルプログラムとして図 1 で示している青色のクラスのみ提示し，その後 CD 等のクラス拡張を 3 段階に分けて実施する．「2. プログラム拡張の振り返り」では，プログラム拡張課題を参照しながら，OOP の利便性と基礎概念の関係性について振り返らせる．そのために，記述問題をいくつか出題し理解度を確認する．この記述問題の一部を，表 3 に示す．問 1～2 では，拡張したプログラムの例から，プログラム中にもたらす利点を意識させながら，OOP の各基礎概念の役割を考えさせる．問 3～5 は，前回の実験で多態性の理解度が低かったことを受け，例示課題の段階から多態性の理解度を向上させるために設けている．問 3，4 は，プログラム拡張課題 1～3 のそれぞれで多態性が実装された箇所と，実現された利点を問う．問 5 では，問 3，4 を受けて，実際に多態性が実現したきっかけとなった基礎概念（継承，オーバライド）を考えさせる．

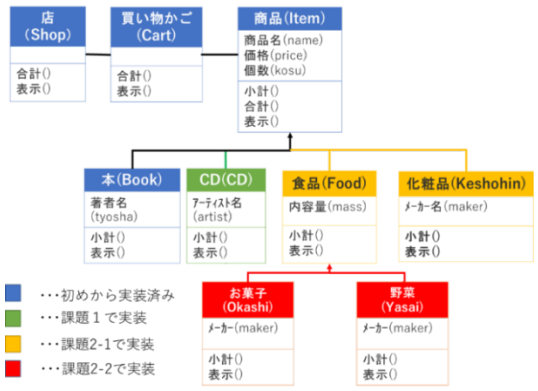

図 1 ショッピングサイト課題のクラス図

表 3 プログラム拡張課題の振り返り（記述問題）

例示課題（記述問題）	
問1)	OOP の各課題で作成した各サブクラス内で，一部の変数が宣言せずに利用できたのはなぜか？
問2)	OOP の各課題で様々なクラスを追加していく際に，小計メソッド shoukei や情報表示メソッド hyouji を再定義せずそのまま使いまわせたのはなぜか．また，そのようなことが実現することで得られるメリットは何か？
問3)	OOP の各プログラム中で，多態性はどのような場面で実現されていたか？
問4)	プログラム内で多態性が実現した際，どのような利点が実現したか？
問5)	基礎課題を通し，多態性が実現するために必要な基礎概念はあったか．ある場合は具体的にどの概念（複数回答可）で，どのように作用したか？

5.1.2 応用課題

応用課題は，

1. 課題全体の俯瞰と多態性に関する考察
2. プログラム拡張の実行
3. プログラム拡張の振返り

という流れで実施した．まず，「1.課題全体の俯瞰と多態性に関する考察」では，応用課題で新たに作成する給与計算のプログラムの概要を説明し，そのプログラム内での多態性の役割について考えさせる．具体的には，クラス図を用いながら給与計算プログラムのデータ構造を俯瞰したあと，多態性について理解を深めるための記述課題を 3 問解いてもらう．記述問題では，まず，プログラム中で多態性が実装されている箇所について予想してもらう（問 1）．そして，なぜその箇所に多態性が生じるのか，継承とオーバライドの働きという観点から考察をさせる（問 2）．最後に，多態性が実現した際のメリットを聞く（問 3），という内容になっている．この 3 問によって「2.プログラム拡張課題」を実施するとき，「クラスの継承とメソッドのオーバライドの働きでクラスごとに異なる処理が行えること」，またこれによって多態性が実現し「メインメソッドの拡張が不要になること」を意識させながらプログラム拡張をさせることを狙う．

次に，「2. プログラム拡張の実行」を実施する．例示課題で実施したプログラム拡張課題と異なる点は，例示課題では段階的に指示を出して拡張をさせているのに対し，応用課題では，プログラムの完成予想として先にクラス図を提示するのみ，つまり拡張の最終的な目標のみ示しているという点である．図 2 に給与計算プログラムのクラス図を示す．プログラム中では，社員の雇用形態や役職によって異なる給与計算・情報出力の処理を行い，給与計算の結果と固定給や労働時間などの情報を表示する．正規社員の各サブクラスでは，給与計算およびボーナス計算のメソッドがオーバライドされ，非正規社員の各サブクラスでは，給与計算メソッドがオーバライドされる．学習者に配布する拡張前のサンプルプログラムは，メインクラスである KyuyoProg クラスと，氏名などの社員全員が共通して持つデータを定義した Shain クラスのみを定義したものとなっている（赤枠内のクラス）．この時点では，社員のデータは，その役職や雇用形態に関わらず全て Shain クラスで処理する仕組みになっている．この状態から，クラス図や入力するデータなどの情報に基づき，プログラムの拡張を行ってもらう．

次に，「3.プログラム拡張の振返り」として，拡張したプログラムに関する記述課題を実施する．表 4 にその内容を示す．この問によって，クラス，継承，オーバライドなどの OOP の基礎概念がプログラム中で作用していること，それらが多態性を実現し，メインメソッドの拡張が不要になる等のメリットを生むことに気づきがあるかどうかを確認する．

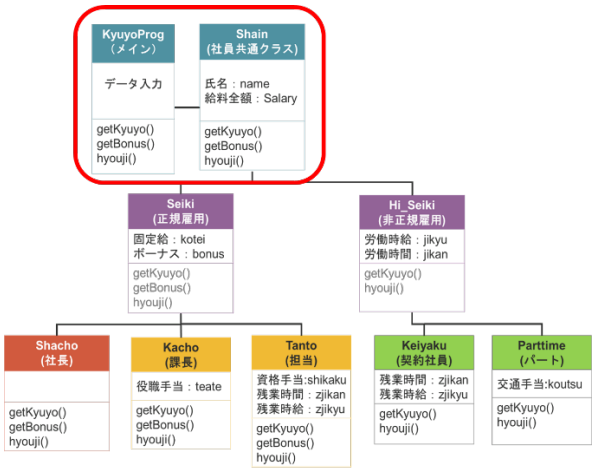

図 2 給与計算プログラムのクラス図

表 4 応用課題における振り返り問題

プログラムを拡張するときに，①どのような基礎概念が作用していたかを説明せよ．また，②どのように作用していたか，それによって③メリットやデメリットは生じたか，④具体的にどんなメリット（デメリット）かを説明せよ．

5.1.1 統合課題

ここでは，特に今回の実験で評価に深く関係する問を抜粋し，表 5 に示す．問 1，2 は，応用課題の給与計算プログラムの拡張に関する問である．類似した問を応用課題で出題しているが（表 3），表 4 で示した問では，多態性，あるいは多態性を実現させるための継承やオーバライドなど，拡張したプログラム内で作用した基礎概念を学習者自身に気づかせるという意図がある．これを踏まえ，統合課題の問 1，2 では，応用課題での学習者の気づきを振り返らせ，多態性についての理解を定着させるといった意図がある．問 4，5 は，これまでの課題を振り返りとして，OOP の利便性と基礎概念の結びつきについて自分の言葉で説明させるまでに理解を落としこませるための問になっている．

表 5 統合課題

問題内容	
問1)	給与計算のプログラム中で，多態性はどのような場面で実現されていたか？
問2)	プログラム内で多態性が実現した際，どのような利点が実現したか？
問4)	なぜ OOP では拡張性が得られるのか？
問5)	なぜ OOP では保守性が得られるのか？

5.2 学習の組み立て

図 3 に学習実験の流れを示す．事前テストでは，OOP の基礎概念に関する問題を出題し，その得点に基づいて，実験群・統制群の 2 群分けを行う．事前テスト終了後，課題テキストを配布し，実験前に各自に読んでもらう．課題テキストとは，OOP の基礎概念や OOP の基本的な考え方，また例示課題で扱うサンプルプログラムの仕組みを簡単に解説した冊子である．例示課題は，ショッピングプログラム拡張課題とその課題に関する記述課題の 2 つを出題している．統制群は OOP による機能拡張を行い，実験群は OOP に加えて「4.2 比較学習」で述べた非 OOP による拡張を行う．応用課題では，5.1.2 で述べたように，課題全体の俯瞰，プログラム拡張，拡張の振返りを行う．統合課題では，今までに行った実習課題のまとめとして，OOP の利便性と基礎概念の関係性について，一般化しながら説明させる．事後アンケートでは，実験で取り扱った OOP の基礎概念や利便性に関する知識や，それらの関係性についてどのくらい理解度が改善したかを学習者に主観的に評価させる．

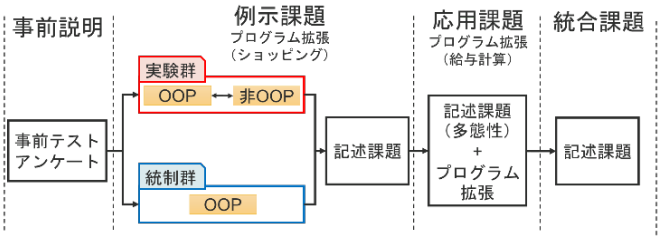

図 3 学習の組み立て

6. 実験結果

6.1 実験概要

対象とした学生は，情報系学科学部生 11 名，大学院生 7 名の計 18 名で，いずれも大学 1 年次に C，2 年次に Java を用いたプログラミング演習を受講している．この 18 名を実験群，統制群の 2 群に分類するために，事前テストを実施した．事前テストでは，基本情報技術者試験の過去問題[10]から OOP に関する問題を 8 問抜粋した．群分けの結果を表 6 に示す．t 検定を実施し，両群の学習者の合計得点に有意な差がないことを確認した．実験期間は 2 週間とし，その中で他人と相談せずに各自で課題に取り組んでもらった．使用言語は学習者が学内講義で扱いが慣れている Java とし，開発環境も同様の理由で eclipse とした．

表 6 事前テストの成績

学習者 / 項目	実験群	統制群
得点の平均（8 点満点中）	5.44	5.00
標準偏差	1.24	1.50
t 検定結果	t(16)=0.69，ns	

6.2 OOP の利便性と基礎概念の関係性の理解度評価

実験終了後，OOP の利便性と基礎概念の結びつきの理解度について，表 1 で示した評価基準に基づいて，統合課題の記述回答を評価した．課題の回答例については 6.5 節で説明する．表 7 に評価結果を示す．レベル 4 以上の理解度に達した学習者は，実験群で 9 名中 6 名，統制群で 9 名中 5 名であった．レベル 4 は OOP の利便性と基礎概念の結びつきについて部分的に説明できる，レベル 5 は具体的な基礎概念を挙げて説明できる，と定義している．両群で OOP の利便性と基礎概念の関係性の理解度に大きな差は見られず，U 検定でも大きな差は見られなかった．

表 7 OOP の利便性と基礎概念の関係性の理解度に関する総合評価

レベル	実験群 (計 9 名)	統制群 (計 9 名)
0	-	-
1	-	-
2	1	2
3	2	2
4	4	2
5	2	3
U 検定結果	ウィルコクソンの W=42.0，ns	

6.3 多態性の具体的な利点の理解の評価

例示課題，統合課題で出題した記述問題の中で，多態性の利点の理解度について評価するための問題を設けた．具体的には，例示課題の問 4（表 3），統合課題の問 2（表 5）が該当する．記述の評価は，多態性の利点について「メインプログラムの拡張が不要・軽減される」など，具体的な記述があれば正解（○/2 点），正解例よりは抽象的であるが的外れではないものは部分点を加算（△/1 点），多態性の利点とは関係のない記述であれば不正解（×/0 点）とした．表 8 に，例示課題，統合課題における両群の得点分布と U 検定の結果を示す．例示課題の平均得点は実験群 1.00 点，統制群 0.44 点で，有意差は無かった．統合課題では，実験群 1.67 点，統制群 0.44 点で，実験群のみ平均点が上昇しており，1%有意水準で有意差があった．この結果から，例示課題の段階でプログラミング拡張を行わせた直後だと，多態性の利点についての記述の質の差は両群であまり変わらないが，応用課題後の統合課題で同様の記述問題を行わせると，実験群の方が統制群よりも記述の質が上がっていることが分かった．

6.4 多態性の理解度向上に関するヒアリング

実験終了後，ヒアリングを行った．表 9 にヒアリング結果を示す．実験を通して多態性の理解度が上がったと感じた学習者が，実験群で 9 名中 7 名，統制群で 9 名中 8 名であった．次に，多態性の理解度が上がったと感じた学習者に，その要因を，役に立ったと感じた順に挙げさせた．要因として提示したのは，実験で使用した課題テキスト，例示課題，応用課題，統合課題の 4 つである．これら 4 つのうち，例示課題と応用課題を上位 3 位以内に挙げた者は，実験群で 7 名中 6 名，統制群では，8 名中 3 名であった．統制群の他の 5 名については，課題テキストを最上位に挙げていた者が 3 名，応用課題のみを挙げていたものが 2 名であった．例示課題と応用課題を上位 3 位以内に挙げた者の人数比に対して母比率の差の検定を行ったところ，$p<0.1$ となり有意傾向が見られた．実験群の学習者は「多態性を理解する要因となった課題は，例示課題と応用課題の両方であり，どちらかの課題ではなく両方の課題に取り組むことで効果が得られる」と実感している傾向にあることが分かった．

表 8 例示課題，統合課題の学習者の得点の分布

学習者 / 評価	例示課題		統合課題	
	実験群 (計 9 人)	統制群 (計 9 人)	実験群 (計 9 人)	統制群 (計 9 人)
○=2 点	3	0	6	1
△=1 点	3	4	3	2
×=0 点	3	5	0	7
平均(点)	1.00	0.44	1.67	0.44
U 検定結果 [1]	ウィルコクソンの W=55.5，ns		ウィルコクソンの W=73.5，p<0.01	

表 9 多態性の理解に関するヒアリング結果

2)「多態性」という OOP の基礎概念について，課題や課題のテキストを通して，多少でも理解度が上がったと感じるか？		
アンケート項目	実験群	統制群
理解度が上がったと感じた	7 名	8 名
例示課題，応用課題(上位 3 位以内)が役に立った	6 名	3 名
母比率の差の検定：基礎・応用を上位 3 位以内に挙げた学習者の比率の差： カイ二乗値=3.61，p=0.057		

6.5 学習者ごとの回答の変化

6.5.1 OOP の利便性と基礎概念の結びつきの理解度

6.1 節で示した OOP の利便性と基礎概念の結びつきの理解度は，例示課題，統合課題の記述問題の回答をもとに評価している．例示課題で出題した記述問題では，OOP の利便性と基礎概念の関係性をショッピングプログラムの拡張課題を例にして回答させ，統合課題では，実施した課題を振り返りながら自分の言葉で回答させた．実験群の学習者 D，統制群の学習者 L の回答例を表 10 に示す．両者の理解度は，最終的にレベル 5 に達している．

まず，例示課題の回答例を説明する．例示課題問 1 では，継承の概念がプログラムに及ぼす働きやそのメ

[1] 本来，2×2 の分散分析を行うべきであるが，ノンパラメトリックな 2×2 の分散分析手法は存在しないため，U 検定で代用した[11][12]．

リットを，ショッピングプログラムの拡張課題を例示しながら説明させる．この問題では学習者 D，L ともにプログラム拡張課題で作成したサブクラスで一部の変数を宣言せずに利用できた理由として,「親クラスで変数が宣言されているため」, といった趣旨の回答が確認できた．そのため，課題の例を使いながら継承がプログラムに及ぼす働きを説明できていると判断し正解とした．この他の問題でも，学習者 D，L は，オーバライドの働きやそれを利用することのメリットについても，プログラム拡張課題の例を挙げながら説明できていた. このことから, 学習者 D, L は, 例示的に OOP の利便性と基礎概念の関係性を理解していると判断し, 表 1 で示したレベル 2 の理解度には達しているとした.

次に，統合課題の回答例を説明する．統合課題問 4 では,「なぜ OOP では拡張性が得られるのか」という問題を出題した．学習者 D，L は拡張性が得られる要因として具体的な基礎概念の名称を挙げながら，OOP で拡張性が生じる要因になる事象を示していることから，理解度レベルを 5 とした.

6.5.2 多態性の具体的な利点に関する理解度の変化

OOP の利便性と基礎概念の関係性には, 多態性の実現が関わっており，これを理解することが OOP の理解度向上に繋がる．そこで，表 4 で示した学習者の理解度をさらに詳しく調べるため，例示課題，統合課題における多態性の利点に関する回答を評価した．評価対象は，表 3 の問 4，表 5 の問 2 の回答である.

具体的な回答例を表 11 に示す. 前節で例に挙げた，実験群の学習者 D には回答に改善が見られ, 統制群の学習者 L には改善が見られなかった．まず，例示課題において学習者 D, L は,「メソッドを一部書き換えるだけでよい」,「同じ処理を記述する必要が無い」といったサブクラスの作成が楽になるという記述をしており，メインクラスが複雑化しない，という多態性が拡張性に及ぼす本質的な利点に関する指摘が無かったため，不正解（×/0 点）とした．しかし，学習者 D は，統合課題では「KyuyoProg.java の記述が簡潔になる」と回答した．これは，メインプログラムの拡張が不要になるという多態性の利点を意味しており，正解（○/2 点）とした．一方で学習者 L は，Shain クラスの構造を真似ればプログラムが作れ，それによって可読性が上がると指摘した．この回答から，学習者 L はプログラム拡張課題を単なる作業としてこなしていた可能性が高い．そもそも多態性の実現に関して気づきがなく，的外れな回答として不正解（×/0 点）とした.

表 10 例示課題，統合課題の回答例

例示課題	問 1）OOP の各課題で作成した各サブクラス内で, **一部の変数が宣言せず**に利用できたのはなぜか？
D（Lv.2）	Food クラスを継承していたため，Food クラスで宣言した変数を利用することができた.
L（Lv.2）	親クラスで宣言されているため.
統合課題	問 4)なぜ OOP では拡張性が得られるのか？
D（Lv.5）	メインクラスをほとんど変更せずに新たなクラスを追加, そこでオーバライドし, そのメソッドの処理を対応したものに変えるだけで済むので, 拡張性が得られている.
L（Lv.5）	継承やオーバライドを用いることにより、サブクラスにおいて拡張したい要素だけを記述すれば良く、これにより記述が容易になるため.

表 11 多態性の利点に関する回答例

例示課題	問 4）プログラム内で多態性が実現した際，どのような利点が実現したか？
学習者 D（×）	syoukei，hyouji メソッドの作成を、メソッド内の計算式や表示内容を一部書き換えるだけで済んだ.
学習者 L（×）	似たプログラムを書くとき，同じ処理を記述する必要がなく，使い回せる.
統合課題	問 2）プログラム内で多態性が実現した際，どのような利点が実現したか？
学習者 D（○）	getkyuyo メソッドを呼び出すだけで各役職に対応した給与計算がなされるため，KyuyoProg.java の記述が簡潔になった.
学習者 L（×）	全てのクラスが Shain クラスと同じ構造をとっていたので，一つのクラスを理解すれば他のクラスの理解も容易．よって可読性が上がることが考えられる.

7. 考察

7.1 実験群と統制群の共通点と相違点

6 章で述べた実験結果から, OOP の利便性と基礎概念の結びつきの理解は，実験群と統制群ともに向上した．具体的には，表 1 で示した評価基準のレベル 4～5 に達した学習者が両群とも過半数以上見られた．相違点はこの利便性と基礎概念の結びつきを理解するための鍵となる「多態性の利点」の理解について，実験群の学習者の方が深い理解が得られたことである．次節から，これらについて詳しく考察する.

7.2 OOP の利便性と基礎概念の関係性の理解度

表 1 のレベル 4～5 に達した学習者は, 前回の実験[7]と比較すると，実験群，統制群ともに増加した．この要因として，今回は，学習の流れに ID の第一原理に則って応用課題を導入し，多態性に関する知識を活用する機会を作ったことが挙げられる. ID の第一原理を学習の流れに則らせる場合は，例示，応用，統合という流れに意味があり，どれか一つの要素を抜いてしまうと学習効果が得られなくなる可能性があることがわかった.

7.3 比較学習の効果

今回の実験では，例示課題で比較学習を行わせたうえで応用課題を実施した実験群で多態性に対する理解度が深まる，という結果が得られた．ヒアリングで，「多態性の理解度が向上したきっかけとなった課題はどれか」という質問を設けたところ，比較学習を行った実験群の学習者のほとんどが，役に立った課題として，例示課題と応用課題の双方を挙げており，統制群の回答とは有意傾向のある差が見られた．学習者の主観的な評価においても，例示課題で比較学習を行ったうえで，応用課題で振り返りをさせるという流れが，OOP の利便性と基礎概念の結びつきの理解を深めるうえで有効であることが示唆された.

8. 今後の方向性

実験の結果と考察から，例示課題で OOP と非 OOP の比較学習を実施したうえで，応用課題で多態性についての学習を行うことが, OOP の利便性と基礎概念の関係性の理解度向上に効果があることがわかった．今後は, OOP の基礎概念の特徴を活用できていないプログラムをリファクタリングによって書き換える課題を導入するなどをし，学習課題の質を高めていく必要がある．また，学習者をさらに増やしたうえで調査を行い，今回よりも明確なデータを取ることも今後の課題である.

参 考 文 献

(1) 情報システムと情報技術辞典編集委員会 : 情報システムのための情報技術辞典，培風館(2006)

(2) アラン・シャロウェイ，ジェームズ・R・トロット : デザインパターンとともに学ぶオブジェクト指向のこころ，株式会社ピアソン桐原(2005)

(3) 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)

(4) 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)

(5) 早川勝，野沢光太郎，松澤芳昭，酒井三四郎 : オブジェクト指向モデリング教育のためのオブジェクト図自動生成システムの設計と評価，情報処理学会論文誌，Vol.54，No.1，pp.66-79(2013)

(6) 高井久美子, 佐々木茂, 渡辺博芳, 荒井正之, 武井恵雄 : 「物語」導入型コンテンツを活用したセルフラーニング型授業―オブジェクト指向プログラミング教育における実践例―，教育システム情報学会誌，Vol.24，No.2，pp.106-116(2007)

(7) 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の改善と評価”， 教育システム情報学会研究報告，Vol.33，No.1，pp39-46(2018)

(8) 野嶋栄一郎，鈴木克明，吉田文 : 人間情報科学と e ラーニング，放送大学教育振興会(2006)

(9) 鈴木克明，市川尚，根本淳子 : インストラクショナルデザインの道具箱 101，北大路書房(2016)

(10) 基本情報技術者試験過去問題（午前）https://www.jitec.ipa.go.jp/1_04hanni_sukiru/_index_mondai.html

(11) 池田郁男 : 統計検定を理解せずに使っている人のためにⅢ， 放送大学教育振興会， Vol.51， No.7，pp.483-495(2013)

(12) 対馬栄輝 : リハビリテーション分野の研究で用いられる統計手法，バイオメカニズム学会誌，Vol.35，No.1，pp.67-75(2011)