

# プログラミング学習の専門的な概念理解における相互評価の 効果

東海林航<sup>\*1</sup>, 伊藤恵<sup>\*1</sup>

<sup>\*1</sup> 公立はこだて未来大学

## Effects of Peer Review on Understanding of Professional Concepts in Programming Learning

Wataru Toukairin<sup>\*1</sup> Kei Ito<sup>\*1</sup>

<sup>\*1</sup> Future University Hakodate

相互評価は様々な学習について効果があることが知られている。そして、プログラミング学習においても相互評価は行われている。しかし、そのほとんどがソースコードについて相互評価しており、配列や if 文, for 文などの専門的な概念についてはほとんど見受けられない。そこで、ソースコードの相互評価と同等の効果が得られるという仮説をたて、専門的な概念に対する相互評価を行い、その効果を調査する。本稿ではまず、プログラミング学習における相互評価の現状について述べる。加えて、専門的な概念に対する相互評価を行った予備実験の結果を示す。最後に、予備実験の結果から本実験に向けた検討と現在の準備状況について述べる。

キーワード：プログラミング学習, 相互評価, 理解度, 専門的な概念, お互い様効果

### 1 はじめに

プログラミング学習において、変数や配列などの専門的な概念を覚えることが多く、プログラミングスキルの向上には専門的な概念の正しい理解が重要である。

本研究では専門的な概念を if, switch, for, while, 変数, 配列などのプログラミングで使われる概念と定義する。また、専門的な概念を学習することはコーディング前に行われることが多く、個人の作業となることが多い。そのため、専門的な概念を理解する際に勘違いや思い込みをしてしまう場合がある。勘違いや思い込みをしたままでは、コーディングを十分に行うことはできない。ここでの問題は個人の主観のみで理解をしているため勘違いや思い込みに気づくのが難しいことである。したがって、他人の理解を見て、自身の理解と比較することが重要である。このような手法として相互評価というものがある。相互評価を行うことで他者と個人の理解を比較し、専門的な概念の理解を深

めることができる。個人の理解を他者と共有することは思考の外化につながり、他者の理解から類似点と異なっている点を比較することで内省を促す。外化と内省が理解度に与える効果については、清水他<sup>(1)</sup>の研究によると理解度に効果があると示唆されている。

相互評価は様々な学習に対して効果があることが知られており、プログラミング学習における相互評価も例外ではない。生田目<sup>(2)</sup>では学生同士の評価によってプログラミング、フローチャートの理解度が上がったと述べている。以上のように、プログラミング学習における相互評価の多くがコードに対しての評価であり、専門的な概念に対しての評価はほとんど見当たらない。また、コーディングの際も専門的な概念について理解していなくても仕様通りに動いてしまうこともある。だが、多くの場合は専門的な概念を理解していなければ思った通りの結果を得ることはできず、プログラミング学習の観点からも正しく理解していないことは望ましくない。このように、専門的な概念の理解が不十分

であってもコードが書けてしまったときコードに対しての相互評価は意味がない。また、専門的な概念の理解が上手く出来ず、コードが書けない場合には相互評価に参加することはできないという問題がある。

本研究では、専門的な概念に対しての相互評価がコードに対しての相互評価と同等の効果が得られるという仮説をもとに専門的な概念に対しての相互評価を行い、その効果について述べる。

## 2 専門的な概念理解における相互評価

本研究はプログラミング学習における相互評価のほとんどがコードに対しての相互評価である現状を踏まえて、あまり行われていない専門的な概念に対する相互評価の効果を調査する。具体的には専門的な概念に対しての相互評価を行い、評価の内容と相互評価の感想や意見から相互評価の効果を分析することで仮説について検討する。

本研究ではコードに対する相互評価の代わりに、Java のプログラムコードを提示し、提示されたコードに関しての設問を出題する。そして、設問への解答を相互評価の対象とする。

評価の方法として、点数とコメントの両方を採用した。その理由として点数付けは一目で評価の良し悪しを確認することができる。しかし、点数だけではその理由を知ることができない。また、コメントは意見を詳細に伝えることができるが、コメントだけでは一目で良し悪しを確認することはできない。さらに、評価内容としてコメントがあることで相互評価の満足度が上がると考える。なぜなら、コメントは意見を詳細に伝えることができるため被評価者が自身への指摘などを把握することができ、改善することができるからである。評価者が必ず正しいとは限らないため、被評価者が相互評価の意見を鵜呑みにすることは避けたい。そのため、良し悪しを一目で確認することができる点数と意見を詳細に伝えることができるコメントを組み合わせることとした。なお、相互評価については、評価者と評価先に関係性がある場合に評価が甘くなる「お互い様効果」<sup>(3)</sup>があることが知られているため、評価者と評価先が分からないように匿名とした。

## 3 予備実験

プログラミング学習の専門的な概念における相互評価の効果を調べるにあたり、前述した仮説などの項目を確認するために予備実験を行った。確認した項目としては、仮説の妥当性、準備、問題の難易度、評価項目の妥当性、相互評価の流れである。各項目を確認し、場合によっては修正することで本実験が滞ることなく進むと考える。設問は Word ファイルで配布、解答してもらい、相互評価は Moodle のワークショップ機能を用いて行った。アンケートについては Google フォームで作成し、実施した。

### 3.1 被験者

この実験では Java のプログラムコードを提示し、提示されたコードに対する設問に回答する。したがって、被験者は Java についてある程度の知識があることが求められる。以上より、今回は著者ら所属大学の 2 年次に行われる Java の講義「情報処理演習 I」を履修した学生とした。

### 3.2 予備実験の流れ

予備実験は以下の流れで行う。

1. プログラムコードとそれに対する設問を提示し、被験者は提示されたコードに対する設問に解答する。
2. 解答を他の被験者が評価し、評価後に解答した被験者が評価内容を確認する。このとき、評価対象の条件として本人以外の解答をランダムに割り当てる。
3. 相互評価の効果を調査するためにアンケートを行う。

なお、設問内容やアンケートの内容は後述する。

### 3.3 設問内容

設問内容としては、if、switch、for、while、関数のコードを作成した。また、各コードに対して最終的に出力される結果や処理の流れ等を解答する設問を用意した。各コードは著者ら所属大学のプログラミングの講義で使用されている教材<sup>(4)</sup>を参考に Java で作成した。設問の対象を上記のように選択した理由は、予備実験であるため各専門的な概念について広く効果を調べたい

```

if文
boolean tenki = true;
double temp = 26.5;
if(tenki == true && temp >= 26.5){
    System.out.println(“洗濯をします”);
    System.out.println(“散歩に行きます”);
}else if(tenki == false){
    System.out.println(“DVDを見ます”);
}else{
    System.out.println(“お家でゲームをします”)
}

```

上記のコードについて以下の質問に答えてください。

1.1. 最終的な結果は何が出力されますか。

1.2. if(tenki == true && temp >= 26.5)では何が行われているか説明してください。

図 1: 設問の一部

ことと被験者の得手不得手に考慮するためである。また、変数については変数だけのコードとすると短いので各専門的な概念のコードに組み込み1つの設問として出すことにした。設問は専門的な概念が理解できていれば答えられる難易度を意識した。設問例としては、「boolean の変数は何を格納することができますか。」、「if(tenki == true && temp >= 26.5) では何が行われているか説明してください。」などがある。図1が作成した設問の一部である。

### 3.4 評価方法

評価方法として3.3節の各設問の解答に点数付けを行い、その点数の理由をコメントに記述してもらう方法とした。この方法とした理由としては第2章で述べた点数付けとコメントの利点がある。図2が作成した項目の一部である。

### 3.5 アンケートの内容

アンケートを行うことで意見や感想を聞くことができる。そして、意見や感想を聞くことで本実験に向けて考慮すべきことを確認する。また、意見や感想と評価内容との関係などを分析することで専門的な概念に対する相互評価の効果を調査する。アンケートについても点数付けとコメントで回答するものとした。効果を定量的に調べるために点数付けを採用し、今後行う予定である本実験への改善点と効果の内容を詳細に調べるためにコメントを採用した。項目としては、相互評価によって理解度が深まったか、振り返りの機会になったか、相互評価をもっとやってみたいと思うか、相互評価が重要だと感じたか、Java のプログラミングがどのくらい得意か、相互評価についての意見とした。また、被験者の学年と学籍番号から Java の講義の中間

## 相互評価実験

### 評価フォーム

#### アスペクト1

設問の1.1について自分の意見と合っているかどうかを基準に評価をしてください。

評定の下にあるコメントには、その評価とした理由や意見を書いてください。

アスペクト1の評定

選択 ... ▾

アスペクト1へのコメント

#### アスペクト2

設問の1.2について自分の意見と合っているかどうかを基準に評価をしてください。

評定の下にあるコメントには、その評価とした理由や意見を書いてください。

アスペクト2の評定

選択 ... ▾

アスペクト2へのコメント

図2: 相互評価の項目の一部

試験と期末試験の点数を参照する。そして、参照した点数と回答してもらったJavaのプログラミングの得意度からプログラミングスキルを推定する。プログラミングスキルは相互評価の内容やアンケートの内容との関係性を分析するのに用いる。なお、試験の点数の使用は各被験者の同意を得たうえでやっている。図3が作成したアンケートの一部である。

### 3.6 アンケート結果

少数の被験者を集めて行った予備実験のアンケート結果を示す。図4は「相互評価を行ったことで理解が深まったか」に対する点数付けの結果を表すグラフである。また、表1は「相互評価を行ったことで理解が深まったか」に対する点数付けの理由である。「相互評価を行ったことで理解が深まったか」については理解が深まったとは言えない結果となった。なぜなら、4を当てはまる、1を当てはまらないとする4段階の点数付けで2を選んだのが2人、3を選んだのが1人、4を選んだのが1人であり、3や4といった高い評価を付けた人の意見が処理の流れやプログラムの言語化について

であり専門的な概念の理解とは関係ないためである。

図5は「相互評価が振り返りのきっかけになるか」に対する点数付けの結果を表すグラフである。また、表2は「相互評価が振り返りのきっかけになるか」に対する点数付けの理由である。「相互評価が振り返りのきっかけになるか」についてはきっかけになると言える。なぜなら、被験者全員が3か4を選択しており、意見についても肯定的であったためである。また、振り返りのきっかけになる理由としてほとんどの被験者が他の人の解答を見ることを挙げていた。

表3は「相互評価をもっとやってみたいと思うか」に対する点数付けの理由である。「相互評価をもっとやってみたいと思うか」については「自分の考えが深まると思う」という肯定的な意見もあるが、評価相手の理解度への不満や評価の項目についての不満が見られたため、どちらとも言えない結果になった。

表4は「相互評価を重要だと感じるか」に対する点数付けの理由である。「相互評価を行うことが重要だと感じるか」についてはおおむね肯定的な意見が多く、重要だと感じていることが分かった。

## アンケート

今回の実験についてのアンケートです。

\*必須

あなたの学年を教えてください。(例：4年) \*

回答を入力

あなたの学籍番号を教えてください。(例：b0000000) \*

回答を入力

1. 相互評価を行うことで理解度が深まったと感じますか。 \*

	1	2	3	4	
当てはまらない	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	当てはまる

その理由を教えてください。 \*

回答を入力

2. 相互評価が振り返りのきっかけになると感じますか。 \*

	1	2	3	4	
当てはまらない	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	当てはまる

図 3: アンケートの一部

表 1: 「相互評価を行ったことで理解が深まったか」に対する点数付けの理由

学習途中だと理解度が深まると思うが、自分は理解していた内容で、理解度が深まったわけではないから  
処理の流れや細かい説明など評価を経て学ぶものが多かった

とても難しいと感じる問題が少なかったから、新しく何かを理解することは少なかった。  
ただ、説明が難しいような問題もあったので、プログラムの言語化するための学びはあった。

相手の解答を見ているぶんには、相互評価の理解は深まると思うが、相手からの評価を見て、学ぶことは設問に  
対する回答の仕方だと思った。一部では当てはまるけど、一部では当てはまらないかなという印象。

1. 相互評価を行うことで理解度が深まったと感じますか。

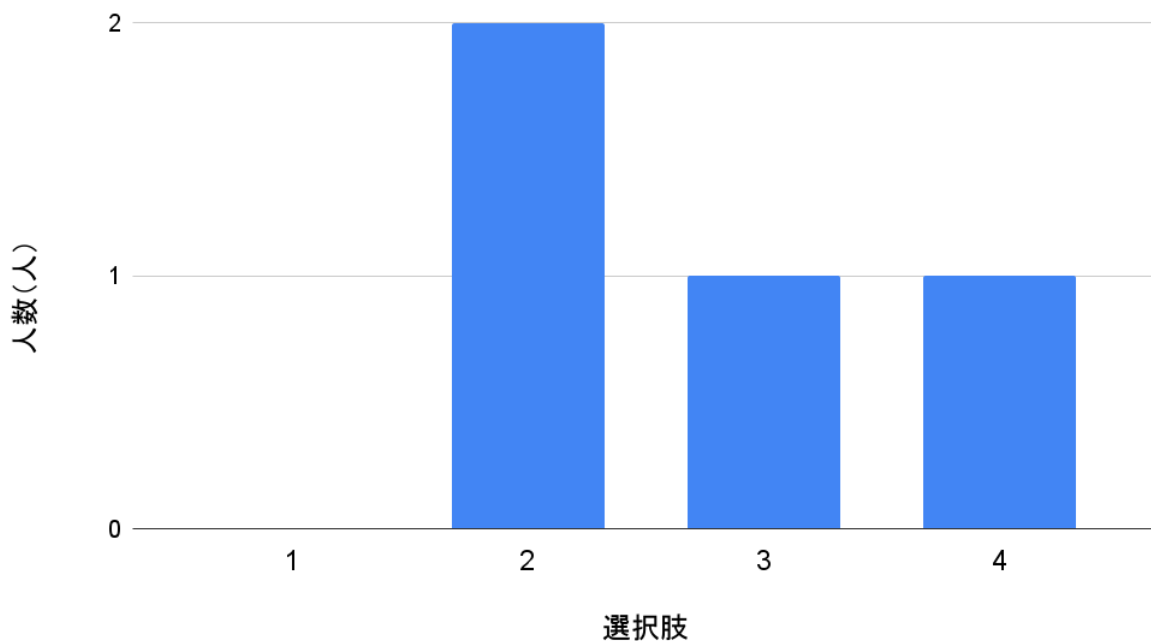


図4: 「相互評価を行ったことで理解が深まったか」に対する点数付けの結果

2. 相互評価が振り返りのきっかけになると感じますか。

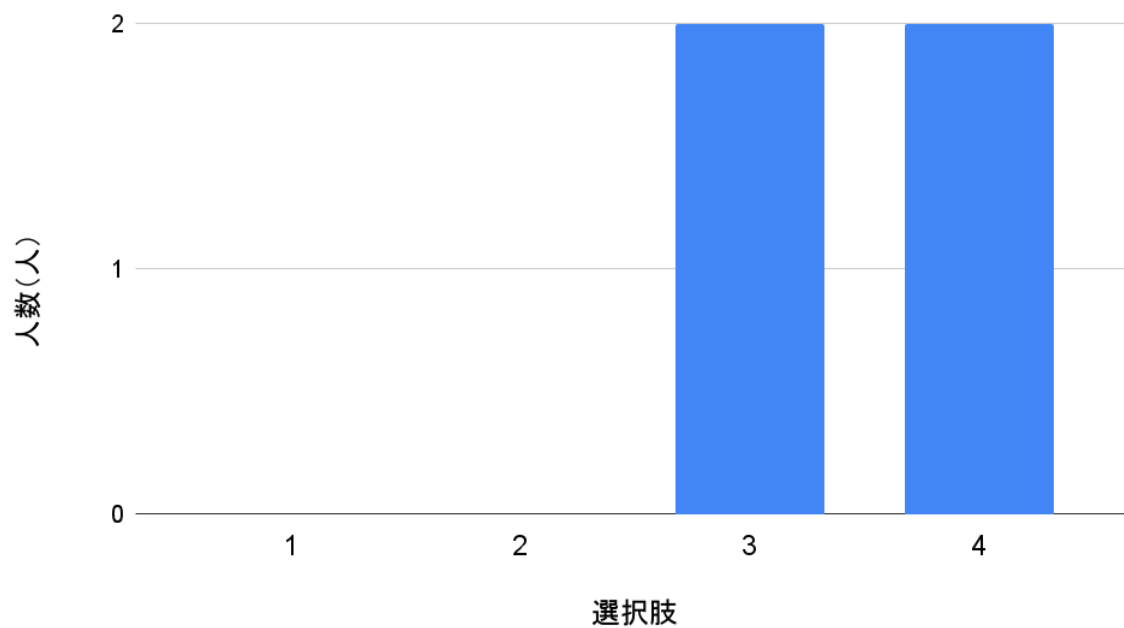


図5: 「相互評価が振り返りのきっかけになるか」に対する点数付けの結果

表2: 「相互評価が振り返りのきっかけになるか」に対する点数付けの理由

---

振り返りとしてはとても良いと思う

---

相互評価の相手と自分の解答を見比べて、学びを深めることができたから

---

他の人の解答を見ることで、自分の解答の間違いに気づくことがあった。

---

相手の解答を見ていて、「あー、自分の解答にこれがたりなかったな〜」と、自分の直す点に気づくきっかけになるから。

---

表3: 「相互評価をもっとやってみたいと思うか」に対する点数付けの理由

---

相手のほうが理解度低いときに意味を成さなくなることがあるように感じたため。(匿名だとより)

さらには、相互評価することが自分の勘違いなどに気づくためならば問題と解答さえあれば気づけるし、理解までもっていくことができる。もしそうでなく理解までもっていくためのものだとしたら一回のやり取りでは足りないと思った。個人的には信頼できる友人とわからないところを話し合っている時間のほうが理解度を深めるための話し合いができると思った。

---

自分の考えが深まると思う

---

振り返りや記述による言語化は、相互評価ならではの気づきがあってよかった。

---

こういう相互評価ならいらなかなあ〜って思った。コードに対する理解よりも、設問に対する回答の仕方(?)を求められている気がして、なんか違うと思った。

---

表4: 「相互評価を行うことが重要だと感じるか」に対する点数付けの理由

---

重要だとは思いますが、人によって合う合わないはありそうだと感じた。

---

人による気がする 問題のレベルが高ければ相互評価がもっと効果でそう

---

上記理由で重要だと感じた。

ただ、「正しく理解する」という面では、テストの実施や教員採点との違いは、今回では自分はあまり感じなかったもので、実施目的にもよると感じた。

---

今回の相互評価は得るものは少なかったから、重要だとはあまり感じなかったけど、自分が理解がたりていないものについての相互評価は得るものは大きいのでは、と思った。

---

### 3.7 実験結果

3.6節の結果から検討が必要な部分を明確にすることができた。まず、設問の難易度については簡単すぎたので難しくした方が相互評価の効果を得やすいという意見を頂いたので、被験者のレベルに応じて設問の難易度を決めていくことを考えている。また、評価する際の指標として「自分の意見と合っているかどうかを基準に評価をしてください。」としていたが、これについては自分の解答にそってやるべきかどうかといった迷いが発生していた。さらに、被験者のほとんどが自身の解答と合致しているかで判断していたため再度、指標について検討する必要があると感じた。実験の流れについては設問を Word ファイルで配布し、Word ファイルで提出してもらったが、Word の環境がない被験者については本来の形式で設問を閲覧できなかったため Moodle 上で設問を閲覧できるようにしたり、事前に Word の環境を持っているか確認するなどの対策が必要である。また、今回は相互評価について 1 回しか行わなかったため、相互評価に慣れていない被験者がいた場合に十分な評価を行うことが出来なかった可能性がある。そして、設問への解答も最初の 1 回だけであり、評価を受けてから被験者自身の解答に反映する機会がなかった。以上を踏まえ、今後は設問への解答と相互評価を複数回行うことを検討していきたい。そして、予備実験で得られた結果や意見を参考に本実験までの準備を進める予定である。

## 4 まとめ

本稿では、まず、専門的な概念を理解することが重要であること理解する際の問題を述べた。次に、相互評価の効果とプログラミング学習における相互評価の現状を述べた。そして、あまり行われていない専門的な概念における相互評価の効果を調査することを示し、調査の具体的な内容について述べた。そして、予備実験とその結果について述べ、結果から予備実験に向けて改善する方法を示した。今後は、予備実験の結果や頂いた意見を踏まえて予備実験の準備をさらに進める予定である。

## 参考文献

- (1) 清水誠, 渡邊文代, 安田修一: “外化と内省が理解に与える効果: 維管束の学習を事例に”, 理科教育学研究, 48(2), pp. 45 - 51 (2009)
- (2) 生田目康子: “ピア・レビューをともなうグループ学習の評価 - 斉型プログラミング授業への適用”, 情報処理学会論文誌, 45(9), pp. 2226-2235 (2004)
- (3) 藤原康宏, 大西仁, 加藤浩: “公平な相互評価のための評価支援システムの開発と評価: 学習成果物を相互評価する場合に評価者の選択で生じる『お互い様効果』”, 日本教育工学会論文誌, 31(2), pp. 125 - 134 (2007)
- (4) 中山清喬, 国本大悟: スッキリわかる Java 入門第 2 版, 株式会社インプレス, (2014)