プログラミング学習における基礎的概念理解を評価対象とする相互評価と評価後の修正が与える影響の調査

東海林航 [*1]，伊藤恵 [*2]
[*1] 公立はこだて未来大学大学院，[*2] 公立はこだて未来大学

An Investigation of the Impact of Peer Evaluation and Post-evaluation Modification on the Evaluation of Fundamental Conceptual Understanding in Programming Learning

Wataru Tokairin[*1], Kei Ito[*2]
[*1] Graduate School of Future University Hakodate, [*2] Future University Hakodate

There are many studies of peer evaluation in programming learning. In most of them, the target of evaluation is a program, and it is difficult for beginners who cannot create a program to participate. Therefore, we thought that peer evaluation without the need to create programs was necessary. In order to conduct peer evaluation without the need to create a program, we defined the fundamental concepts that are considered to be acquired before creating a program as the evaluation targets. Understanding of fundamental concepts is important in learning programming, because it is necessary to combine multiple fundamental concepts when creating a program. We thought that the understanding of the fundamental concepts could be further deepened by revising the answers to be evaluated after the peer evaluation. In this study, we investigate the effects on learners of peer evaluation and post-evaluation modification of fundamental concepts.

キーワード：プログラミング学習，相互評価，基礎的概念，評価後の修正

1 はじめに

現在，IT 人材が不足していることが知られており，2021 年度に実施された IT 企業に IT 人材の過不足感について尋ねる調査の結果では全体で 8 割強が「やや不足している」「大幅に不足している」と回答している [(1)]．また，IT 人材は今後も不足すると言われており，2030 年度には最大で約 79 万人不足すると予想されている [(2)]．以上より，IT 人材の不足を解消することは重要であり，そのためには IT 人材の育成が必要であると言える．IT 人材の育成の際，プログラミングを学習することは避けて通ることはできない．

プログラミング学習に関する研究は様々行われており，著者らはその中で自分自身の理解と他者の理解を比較，共有できる相互評価に注目した．相互評価で自分自身の理解を共有するために，何らかの方法で自分自身の理解を表現することは外化を促す．また，相互評価で自分自身の理解と他者の理解を比較することは内省を促す. 外化と内省が学習者の理解に与える影響について，清水他 [(3)] は，学習者の理解を促すと述べている．以上より，プログラミング学習に対して相互評価をすることは，学習者のプログラミングに対する理解を促すことができると考えた．

プログラミング学習における相互評価の研究は行われているが，その多くが評価の対象をプログラムとしている．評価対象がプログラムであるとき，学習者がプログラムを満足に作成できないと相互評価に参加することが難しい．加えて，満足に参加できないと相互評価の効果を受けることが難しくなる．そのため，評

価対象がプログラムではない相互評価として，プログラムを作成する前に学習することが多いと考えられるプログラミングの基礎的な内容に基礎的概念と名付け，評価対象とした．以上の内容を踏まえて，著者らは基礎的概念を評価対象とする相互評価に関して複数年にかけて取り組んできた [4]．今回は今までの内容に加えて，相互評価を実施後に評価対象を修正する行為を追加し，さらに学習者の基礎的概念理解を促すことを目指す．相互評価と評価対象の修正については，5.2 節で詳細を述べる．

以上より，本研究では，プログラミング学習における基礎的概念を評価対象とする相互評価と評価対象の修正が学習者に与える影響を調査する．

2　基礎的概念とその理解

本研究では，プログラムを作成する前に学習することが多いと考えられるプログラミングの基礎的な内容を基礎的概念とする．具体的には，for や while などの繰り返し，if や switch などの条件分岐，配列などの概念が挙げられる．基礎的概念は，プログラムを作成する前に学習することが多いことに加えて，主に個人で学習される．そのため，間違って理解した場合には，学習者自身で間違いに気づくことは難しく，間違いに気づかない状態でプログラムを作成することになる．そのような状態では，十分なプログラムを作成することは難しく，何度も試行錯誤する内に偶然プログラムを作成できた場合においても間違った理解に気づくことは難しいと言える．したがって，基礎的概念を正しく理解することはプログラミング学習において重要である．基礎的概念の理解については，図 1 概念とコードがつながること，もしくはコード以外の言語を通して概念とコードがつながることとする．具体例として，for や while に「繰り返し」と名付けることで，何か繰り返す処理を呼び出したい場合に for や while を選択肢として扱うことができることが挙げられる．概念がコードやコード以外の言語とつながっていない場合では，言語化やプログラミングなどの表現する方法がないと言える．以上より，本研究では基礎的概念の理解の目標として，基礎的概念をコードかコード以外の言語で表現できることとする．

3　関連研究

生田目 [5] では，対象授業をプログラミングとした一斉授業にピアレビューを活用している．レビュー対象はフローチャートとソースプログラムおよび実行結果である．その結果として，学習目標の達成に効果があったことを述べており，アンケートの因子分析の結果として 3 因子を抽出している．第 1 因子は，お互いに教えあうことでプログラミングやフローチャートの理解が大幅に向上した (グループ学習の効果) である．第 2 因子は，評価をすることによって，プログラミングの良い具体例を見ることができた (レビュアの利得) である．第 3 因子は，レビューの結果，フローチャートの誤りが発見できた (レビューの効果) である．以上より，ピアレビューはプログラミング学習にとって有効であることが分かった．しかし，フローチャートやソースプログラムをレビュー対象とする場合に基礎的概念を理解していなければ参加が難しくなると考えられる．そのため，プログラミングの初学者においては本研究のような基礎的概念を対象とする方が効果的である．

菅井他 [6] では，高等学校の教科「情報」の科目「情報の科学」でのプログラミング学習の際に，教育用 SNS による相互評価を取り入れた授業を実施している．その結果，プログラミングの授業において教育用 SNS を利用した相互評価は，質問がしやすい雰囲気づくりや生徒が興味を持てる課題設定をすることにより，プログラムの修正と再評価を促すこと，およびプログラムの完成度を高めるのに有効であることを示唆している．この研究の「質問がしやすい雰囲気」を本研究のピアレビューでは，評価先に気を遣わずに評価できる環境と捉え，本研究の相互評価の実施方法を考えた．

藤原他 [7] では，ピアレビューの際に評価した人に評価される際に評価が甘くなるお互い様効果について述べている．お互い様効果の詳細については，評価する相手も評価者を評価する場合は，そうでない場合に比べて評価が甘くなる可能性があるという効果である．この原因について，相手に高い評価をすることで，互恵的に自分にも高い評価をしてもらいたいという期待があるために起きたのではないかと述べている．そのため，本研究では，相互評価を行う際にお互い様効果について考慮することとした．

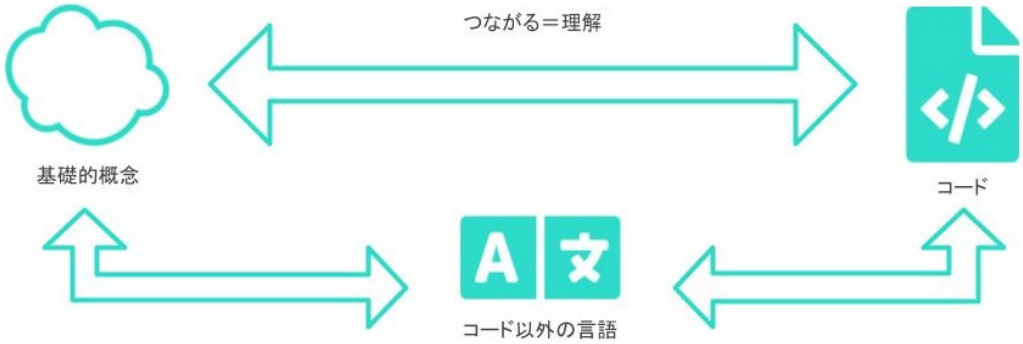

図 1: 基礎的概念の理解のイメージ図

4　基礎的概念を評価対象とする相互評価

この章では，基礎的概念を評価対象とする相互評価の詳細について述べる．

4.1　評価対象

評価対象は基礎的概念の理解であるため，基礎的概念の理解を問う問題への解答を評価対象とした．基礎的概念の理解を問うためには，第 2 章で述べた通り，コードと概念のつながりやコード以外の言語と概念のつながりを問う必要がある．そのため，基礎的概念に対応した短いコードを提示し，そこから内容を読み取る問題とした．

4.2　評価方法

評価方法は，4.1 節で述べた問題 1 問ごとに 5 段階の点数付けと基礎的概念ごとにコメントを項目ごとにつけることとした．5 段階の点数付けは，「正しくない」と思う場合は 1，評価者が「正しい」と思う場合は 5 と評価するように設定した．コメントの項目は，石元他 [8] の研究で使用されている項目を参考にし，「良い点」「改善すべき点」「疑問および反論点」の 3 つに設定した．点数は一目で良いか悪いかを判断でき，コメントは評価者の意見を詳細に伝えることができる．そのため，点数付けとコメントの 2 つを組み合わせ，評価の内容が評価を受けた人 (以下，被評価者) に伝わりやすくなるように工夫した．

4.3　評価相手

第 3 章で述べた菅井他 [6] と藤原他 [7] の研究より，本研究で実施する相互評価は匿名で行う．匿名で行うことで，評価先に気を遣わずに評価できる環境や評価の際に起こる相手を高く評価することで互恵的に自分にも高い評価をしてほしいという期待を減らすことを狙い，公平な評価が行われるようにする．相互評価の実施環境は，著者ら所属大学で使用されている Moodle[1] のワークショップ機能を用いた．また，匿名性を確保するために，実験実施者側で用意した Moodle のダミーアカウントを使用して相互評価に参加させた．

5　実験

この章では，本研究で実施した相互評価実験の詳細について述べる．実験で，基礎的概念を評価対象とする相互評価は以下の手順で行った．

1. コーディングテストへの解答 (評価前のコーディングスキル確認用)
2. 設問への解答
3. 解答した設問を対象に相互評価
4. 被験者が自身への評価を確認後，設問への解答を修正
5. コーディングテストへの解答 (評価後のコーディングスキル確認用)
6. アンケートへの回答

被験者は，著者ら所属大学の学部 1 年次の学生，全 4 名で実施した．以下，全 4 名の被験者を被験者 A，B，C，D とする. 被験者を学部 1 年次の学生とした理由は，必修でプログラミングの演習形式の講義を受けており，その講義でプログラミングを初めて学習する学生が多く，基礎的概念を理解している途中であると予想されるためである．

コーディングテストは，基礎的概念を評価対象とする相互評価がコーディング能力に与える影響を調査す

[1] https://moodle.org/

る目的で実施した．具体的には，相互評価の前後で解答してもらい，その差からコーディング能力に与える影響を調査する．

設問は，4.1 節で述べた通り，基礎的概念ごとに論理的に複雑ではない単一の処理を行うコードを提示し，提示したコードに関する問題に解答する内容である．設問で扱う基礎的概念については，for，while，if，配列とした．図 2 に実際に for についての設問の一部を示す．なお，コーディングテストと設問で扱うプログラミング言語は，被験者の受講しているプログラミングの演習形式の講義で使用されている Processing[2]とした．

アンケート調査は，基礎的概念を評価対象とする相互評価が学習者に与える影響を幅広く調査する目的で実施した．アンケートの内容は，4 段階で回答するように設定した．具体的には，回答者が質問に対して「当てはまらない」と思ったときに 1,「当てはまる」と思ったときに 4 と回答する．また，質問によって複数回答可とするものを数問用意した．

5.1 生成系 AI の活用と評価の割り当て

相互評価の際，参加者の 1 人として生成系 AI である ChatGPT[3]を使用した．ChatGPT を相互評価の参加者として使用するために，設問への解答を生成し，他の被験者の解答を入力したうえで評価を生成した．なお，ChatGPT に初学者が間違える可能性が高いことをプロンプトとして入力し，生成した．生成した評価のコメントについて，なるべく生成したままの状態を使用したが，大きな間違いが見られる場合にコメントを実験実施者側で軽く修正した．修正内容の例としては，「1 の回答が若干間違っています。正しい最終的な出力結果は 12 ですが、回答が 11 になっています。計算をもう一度確認する必要があります。」を「1 の回答が若干間違っています。正しい最終的な出力結果は 18 だと思いますが、回答が 12 になっています。」と修正したことが挙げられる．例で挙げた内容に修正した理由は，評価対象となっている被験者以外の解答が全て出力結果が 18 であり，修正しなかった場合に間違った理解に導く可能性が高いと判断したためである．生成した解答は，他の被験者の解答と同様に評価をしてもらい，生成した評価は相互評価の参加者として提示した．つまり，被験者は ChatGPT で出力した解答，評価とは知らされない状態で相互評価を実施したことになる．生成した解答は 1 つであり，評価は被験者の解答ごとに生成した．そのため，各評価の内容に違いはあるが，評価の数としては 4 つである．

被験者の評価の割り当ては，各被験者自身以外の 3 人への評価に加えて ChatGPT で生成した解答への評価を割り当て，全部で 4 人分の評価をするようにした．

5.2 評価後の修正

以前の実験 [4] では評価後の修正を行っていなかったため，相互評価が基礎的概念に与える影響を調査する方法がコーディングテストとアンケート調査しかなかった．コーディングテストは，コーディングに与える影響を調査する方法であり，基礎的概念の理解に与える影響を詳細に調査方法がアンケート調査しかない．そのため，今回は基礎的概念を評価対象とする相互評価が被験者の基礎的概念の理解に与える影響をコーディング以外の方法で調査するために評価後に設問への解答を修正をさせることとした．そして，修正の数や内容から，相互評価が基礎的概念の理解に与えた影響を考察することを目的としている．具体的には，各被験者が自分自身への評価を確認後に設問への解答を修正させ，修正は各自で必要であると判断した場合のみ行うよう指示した．加えて，修正内容を把握しやすいように修正した部分の文字の色を変更するように指示した．

6 結果

この章では，第 5 章の実験結果について述べる．

6.1 コーディングテスト

相互評価の前後で行ったコーディングテストの正解数 (12 点満点) を表 1 に示す．評価の前後で正解数が大きく増加するなどの変化を見ることはできなかった．

6.2 相互評価

5 段階の評価の平均値を表 2 に示す．なお，自分自身の評価は行っていないため，その部分は空欄になっている．また，評価全体の最頻値は 5 であり，評価のほとんどで最大値である 5 と評価していることが分かる．次に，基礎的概念ごとの評価値の平均を表 3 に示す．表 3 より，全ての基礎的概念において，評価値の平均は

[2] https://processing.org/
[3] https://chat.openai.com/

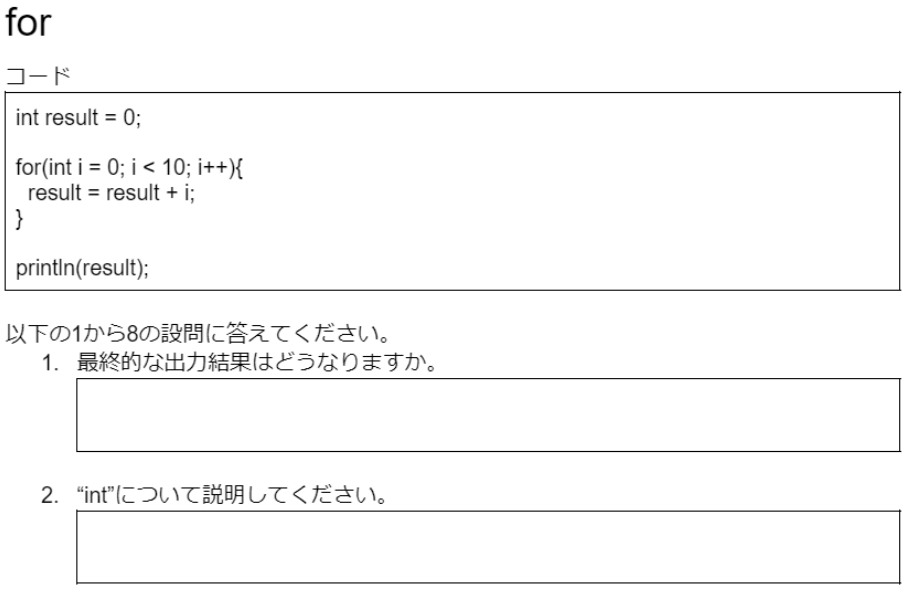

図 2: 設問の一部

表 1: コーディングテストの正解数

被験者	A	B	C	D
評価前	11	11	12	11
評価後	12	11	12	11

4 以上であり，評価値の平均が一番低い基礎的概念は while であった．また，各被験者と ChatGPT がつけた評価値の平均を表 4 に示す．なお，評価値は実験実施者側で評価した値の傾向と似たものであったため，評価値の妥当性は最低限保証されているものとする．つけた点数の平均値が一番低いのは被験者 C であるのに対して，平均値が一番高いのは ChatGPT であった．

評価のコメントについて，各項目で見られた傾向を述べる．まず，良い点では解答や理解が正しいといった内容や記述が詳細であることが記述されていた．次に，改善すべき点では，解答が間違っていることを指摘する内容や記述内容の読みやすさや見やすさを指摘する内容が見られた．最後に，疑問および反論点では，「特になし」という内容の記述が多く見られた．

6.3 評価後の修正

自分自身への評価を確認後に設問への解答を修正した結果について述べる．被験者ごとの修正数について，表 5 に示す. 表 5 より，全ての被験者が何らかの修正を行っていたことが分かる．修正の具体例として，相互評価時の評価コメントの項目である改善すべき点で「３の『array.length :』は回答には必要ない」とコメントされた被験者が修正後に指摘された部分である「array.length :」を削除したことが挙げられる．

6.4 アンケート

アンケート調査の結果から特徴的なものについて述べる．アンケートへの回答は第 5 章でも述べた通り，基本的には「当てはまらない」と思ったときに 1,「当てはまる」と思ったときに 4 と回答する 4 段階を設定した．まず，Processing のプログラミングが得意かどうかについては，平均値の最大値 4 に対して 3 であった．Processing のプログラミングが得意かどうかについて聞いた結果と相互評価で各被験者が受けた評価値の平均に関係性を見出すことはできなかった．相互評価が参加者に与える影響について聞いた質問と回答の平均値を表 6 に示す．表 6 より，相互評価が参加者に与える影響について聞いた質問の回答の平均値は全て 3.25 以上であった．今回の相互評価の評価人数について聞いた質問と被験者ごとの回答を表 7 に示す．表 7 より，今回の相互評価において，各被験者の評価できる数と評価されたい数は同じであった．また，評価項目が妥当であったかを聞く質問の平均値は，3.25 であり，自分自身への評価が妥当であったかを聞く質問は平均値

表 2: 評価結果

		被評価者				
		A	B	C	D	ChatGPT
評価者	A		4.88	4.85	4.35	3.08
	B	4.54		4.69	4.38	3.31
	C	3.85	4.00		4.00	2.42
	D	4.23	4.23	4.62		2.73
	ChatGPT	4.81	5.00	4.92	4.85	

表 3: 基礎的概念ごとの評価値の平均

基礎的概念	for	while	配列	if
評価値の平均	4.19	4.01	4.42	4.18

表 4: 各被験者と ChatGPT がつけた評価値の平均

被験者	A	B	C	D	ChatGPT
評価値の平均	4.29	4.23	3.57	3.95	4.89

表 5: 被験者ごとの修正数

被験者	A	B	C	D
修正数	3	1	4	6

表 6: 相互評価が参加者に与える影響について聞いた質問と回答の平均値

質問	平均値
理解度が深まったかを聞く質問	3.50
振り返りのきっかけになるかを聞く質問	3.50
学習のモチベーションがあがったかを聞く質問	3.25
理解できている部分とそうでない部分を知ることができたかを聞く質問	3.50

が 4 であった．

7 考察

ここからは，第 6 章で述べた結果を基に考察する．

7.1 相互評価

6.2 節より，評価全体の最頻値は 5 であり，評価のほとんどで最大値である 5 と評価していた．また，6.4 節では触れなかったが，設問について広く聞いた複数回答可の質問の結果で「簡単だった」と 4 人中 3 人が回答していた．このことから，評価者が設問の解答ほとんどを正しいと感じており，今回の設問が被験者にとって易しすぎたことが考えられる．また，6.2 節の表 2 より，ChatGPT に対する評価値の平均が他の平均に比べて低かった．これは，設問への解答を生成する際に，初学者として解答することと初学者が間違える可能性が高いことをプロンプトとして入力したが，間違いが明確であることやある解答と他の解答で矛盾が生じるなど，間違いの程度が大きすぎたことが原因だと考えられる．そのため，ChatGPT を相互評価の参加者の 1 人として使用する場合には，初学者の解答や評価を学習させるなどの工夫を通して，解答や評価を初学者のレベルにあわせる必要がある．

コメントについては，おおむね項目に沿った内容が記述されていた．しかし，疑問および反論点については「特になし」といった内容が多く見られた．この原因として，先に述べた評価対象である設問の難易度が易しかったことと，そもそも評価者が疑問や反論点を感じることが少なかったことが挙げられる．基礎的概念の理解において，疑問や反論を持って評価として記述することは思考の外化と内省にとって重要であると考える．そのため，現状では疑問および反論点の項目を変更することはしない．

評価項目や自分自身への評価が妥当であったかについて，6.4 節より，評価項目が妥当であったかを聞く質問の平均値は，3.25 であり，自分自身への評価が妥当であったかを聞く質問は平均値が 4 であった．自分自身への評価については，相互評価を匿名で行ったことが妥当性につながったと考えられる．なぜなら，自分自身への評価が妥当であったかを聞く質問への回答の理由で，ミスや自信がない解答に低い評価がされていたことが記述されていたからである．匿名でない場合，低い評価をすることが難しいため，評価者自身が正しくないと思ったときに低い評価をつけやすい環境を設定できたことは被験者や評価の正確性にとって良かったことが示唆される．

7.2 相互評価が与える影響

6.3 節の表 5 より，評価後に実施した解答の修正において，修正の大小はあるが全ての被験者が修正を行っていた．このことから，今回実施した相互評価が被験者自身の理解に何らかの影響を与えたことが示唆される．また，6.4 節の表 6 より，相互評価が参加者に与える影響について聞いた質問の回答の平均値は全て 3.25 以上であった．平均値が 3.25 であった質問は学習のモチベーションがあがったかを聞いた質問であり，それ以外の質問への回答は被験者全員が 3 以上を回答していた．学習のモチベーションがあがったかを聞いた質問では，1 名だけ 2 と回答しており，他の全ての回答は 3 以上であった．これについて，学習のモチベーションがあがったかを聞いた質問で 2 と回答した被験者は，Processing のプログラミングが得意かを聞く質問で 4 と回答しており，基礎的概念についての理解が高かったためモチベーションがあがらなかったことが考えられる．しかし，他の影響について聞く質問では 3 以上と回答していたため，モチベーション以外の影響は受けていると言える．以上より, 与えた影響が被験者自身の理解や振り返りなどの良い影響であることが示唆される．しかし，相互評価の前後で実施したコーディングテストで正解数が増加するなどの大きな変化を見ることができなかった．これは，コーディングテストの正解数が高かったことから，難易度が易しすぎたことが変化を見ることができなかった原因として考えられる．そのため，コーディングテストの難易度を学習者の基礎的概念の理解の程度にあわせて調整する必要がある．

7.3 相互評価の人数

6.4 節の表 7 より，今回の相互評価において，各被験者の評価できる数と評価されたい数は同じであったことが見られた．このことから，相互評価の人数については自分自身が評価した人数と同じか，それ以上の評価が欲しいことが示唆される．しかし，評価できる人数と評価されたい人数については個人差があり，今回のアンケート調査でも最低が 2 人で最高が 5 人である．

表 7: 評価の人数についてのアンケート結果

被験者	A	B	C	D
何人まで評価できるか聞く質問	5 人	2 人	3 人	5 人
何人に評価されたいか聞く質問	5 人	2 人	3 人	5 人

そのため，相互評価を実施する際のグループ分けについては，設問の難易度と数，参加する学習者のレベルにあわせて調整する必要がある．

8 まとめ

本稿では，まず，基礎的概念とその理解，関連研究について述べた．そして，基礎的概念を評価対象とする相互評価の詳細について述べた．その後，基礎的概念を評価対象とする相互評価を実施した実験について述べた．実験結果から，匿名で実施した基礎的概念を評価対象とする相互評価と解答の修正は学習者に良い影響を与えていることが示唆された．今後は，実験で用いた設問やコーディングテストの難易度を調整し，相互評価が学習者に与える影響がより良くなるように改善していきたい．

参 考 文 献

(1) 独立行政法人情報処理推進機構: “デジタル時代のスキル変革等に関する調査 (2021 年度) 企業調査報告書”, https://www.ipa.go.jp/jinzai/chousa/qv6pgp000000bv6s-att/000097874.pdf (2022)

(2) 経済産業省: “IT 人材の最新動向と将来推計に関する調査結果～ 報告書概要版 ～”, https://www.meti.go.jp/shingikai/economy/daiyoji_sangyo_skill/pdf/001_s02_00.pdf (2016)

(3) 清水誠，渡邉文代，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 (2007)

(4) 東海林航，伊藤恵: “プログラミング学習における基礎的概念の理解を評価対象とした相互評価実験の複数年実施とその考察”, 実践的 IT 教育シンポジウム rePiT 論文集, pp. 16–23 (2023)

(5) 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)

(6) 菅井道子，堀田龍也，和田裕一: “教育用 SNS を高校生の相互評価に導入したプログラミング学習の効果”, 研究報告コンピュータと教育 (CE)，2017-CE-140(2)，pp. 1–6 (2017)

(7) 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)

(8) 石元みさと，末利 容子: “非対面式ピア・レスポンスを取り入れた大学生への小論文指導”, 東京学芸大学国語教育学会研究紀要，14，pp. 12–22 (2018)