

教育用 SNS を介した相互評価を取り入れた 高校生のプログラミング学習に対する意識調査

High School Students' Attitude toward Learning of Programming with Mutual Evaluation via SNS for Education

菅井 道子* 堀田 龍也** 和田 裕一**
Michiko Sugai* Tatsuya Horita** Yuichi Wada**

東北大学大学院情報科学研究科人間社会情報科学専攻*
東北大学大学院情報科学研究科**
Department of Human-Social Information Sciences,
Graduate School of Information Sciences, Tohoku University*
Graduate School of Information Sciences, Tohoku University**

<あらまし> 本研究では、高等学校の教科「情報」におけるプログラミングの学習の際に、教育用 SNS を介した相互評価を導入した授業実践を行い、相互評価がプログラミングを学習する意識に及ぼす影響を検討した。その結果、教育用 SNS を介した相互評価が、学習者のプログラミングを学習する意識に対して、「楽しさを感じさせる」、「課題遂行の糧となる」、「知識・理解を深める」、「自己効力感を得る」といった好影響を与えることが示唆された。

<キーワード> 中等教育 教育用 SNS プログラミング アクティブ・ラーニング
C 言語

1. はじめに

内閣府（2015）の「世界最先端 IT 国家創造宣言」では、初等・中等教育段階におけるプログラミングに関する教育の充実に努め、IT に対する教育を育むとともに、IT を活用して多様化する課題に創造的に取り組む力を育成することが重要であるとされている。これを受けて、次期学習指導要領改訂では「情報教育」に関する具体的な方向性として、プログラミングや情報セキュリティ等の学習活動の充実に努めるとともに、問題の発見と解決に情報技術を活用するために必要となる科学的な考え方を育成する共通必修履修科目の設置が検討されている（文部科学省 2015a）。

現行の学習指導要領では、中学校の教科「技術・家庭」、および、高等学校の共通教科「情報」の科目「情報の科学」にてプログラミングについての扱いが述べられている。これに関連して、文部科学省（2015b）は、プログラミング指導のための参考資料「プログラミング教育実践ガイド」を発行している。ここでは、児童生徒の発達段階に合わせてさまざまな実践例が紹介されている。例えば、小中学校の実践では VISCUIT

や Scratch などのビジュアルプログラミング言語を用いた実践が、また、高等学校の実践では C 言語や Java, JavaScript などのテキスト記述型のプログラミング言語を用いた実践がそれぞれ取り上げられている。これらの実践例に共通してみられる特徴として、「プログラミングに対して児童生徒に抵抗感を与えないために、モチベーションを継続できるように工夫している」ことや「児童生徒の『プログラミングは難しい』という思い込みを払しょくさせ、自分にもできるという自己効力感を高めさせている」ことが挙げられる。概して、児童生徒にはプログラミングは難しいというイメージがあり、それがプログラミング学習を困難にさせているということがうかがえる。これらのことから、初等・中等教育の授業においてプログラミング学習を設計する上では、プログラミングは難しいと感じさせない工夫が必要であると考えられる。

ところで、今日の社会においては、ICT を活用して協調的問題解決を行うことのできる 21 世紀型スキルを持った人材の育成が重要であると指摘されている（Griffin *et.al.* 2012）。協調的問題解決に利用できる ICT ツールとしては、資

料や画像などのファイルを共有することができ、学校での生徒同士の話し合いなどの場面で使うことができる教育用 SNS がある（文部科学省 2011）。教育用 SNS は、インタラクティブなコミュニケーションツールとして使うことができるため、プログラミングを学習する上では、生徒同士の学び合いやプログラムファイルの共有といった場面で使うことが可能である。

ソーシャルメディアに関しては、高校生の 91%が LINE などのソーシャルメディアを利用し、「友だちとより気軽にコミュニケーションができるようになった」など人間関係に良い変化が生じているとの報告がある（総務省 2015）。もし教育用 SNS を介して生徒同士が楽しく学び合いできるようなプログラミング学習の場が実現できれば、多くの生徒がもつプログラミングは難しいというイメージを少しでも払拭できるのではないだろうか。そこで本研究では、プログラミングを習得する際に教育用 SNS を介した相互評価を導入し、そのことが生徒の学習意識にどのような影響を及ぼすのかについて、授業の事前と事後に調査した質問紙の回答の比較を通じて検討することを目的とした。

2. 方法

2.1. 調査対象

調査は、高等学校の共通教科「情報」の科目「情報の科学」を学んでいる宮城県内の公立高等学校普通科 2 年生 6 クラス 238 名（男：143 名 女：95 名）を対象とした。

2.2. 調査時期

平成 27 年 12 月～平成 28 年 2 月

2.3. 調査内容

調査は、「情報の科学」の授業時間内に、高等学校学習指導要領にある「問題解決と処理手順の自動化」について学ぶ授業の一環として行った。

2.3.1. 授業の流れ

授業の流れを表 1 に示す。

本研究におけるプログラミング授業の実践では、2 つの演習を用意し、図 1 に示す演習の流れの中で小さなステップの積み重ねでプログラミング技術を習得させることを目指した。また、

表 1 授業の流れ

時	授業内容
1 質問紙 演習 1	事前質問紙調査 C 言語プログラムの基礎、 printf 関数、scanf 関数の使い方、 例題による対話プログラムの作成
2 演習 1	コンピュータとユーザーが対話する プログラムの作成
3 演習 1	教育用 SNS を活用した相互評価と プログラムの修正(1/2)
4 演習 1	教育用 SNS を活用した相互評価と プログラムの修正(2/2)
5 演習 2	判断文 (if 文) と計算の仕方 判断文と計算を使った オリジナルプログラムの作成 (1/2)
6 演習 2	判断文と計算を使った オリジナルプログラムの作成 (2/2)
7 演習 2	教育用 SNS を活用した相互評価と プログラムの修正(1/2)
8 演習 2	教育用 SNS を活用した相互評価と プログラムの修正(2/2)
質問紙	事後質問紙調査
9 例題のみ	繰り返し文

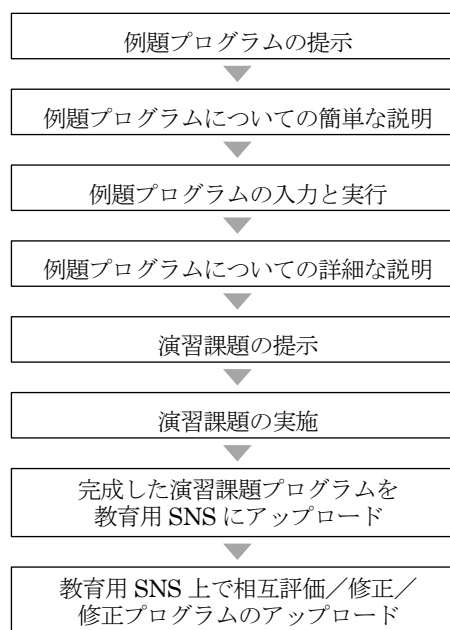


図 1 各演習の流れ

生徒同士の協調的な学び合いを促進するため、教育用 SNS 上で相互評価をさせ、そこでプログラムのバグ（ミス）に自ら気づいたり、または他者から指摘されたりした場合には、修正したプログラムを再度アップロードして評価をもらうことを促した。また、お互いに意見を出し合いプログラムを改良に導くことを推奨した。生徒同士の学び合いをクラス全員で共有するために教育用 SNS を利用したが、口頭での会話を抑制することはしなかった。

質問紙は、授業を始める前と演習2（8時間目）が終了した時点で調査を行った。質問項目等については、2.6.および2.7.にて説明する。

2.4. 教材

教材として、言語マニュアルは教師が作成したオリジナルのものを用いた。題材は、プログラミングの三要素である「順次」、「選択」、「繰り返し」に加え、初学者がつまづきやすいと思われる「プログラミングにおける計算の仕方」の4つを用意した。PDF文書で作成された言語マニュアルは、授業用サーバーの共用フォルダに入れておき、生徒がいつでも見られるようにした。

2.5. 授業環境

授業は、デスクトップ型パソコンが47台（教員用1台 生徒用46台）設置されているコンピュータ室で行った。

事前および事後の質問紙調査は、生徒がインターネット経由で学習支援システム Moodle にアクセスし、「フィードバック」機能による Web-based アンケート方式で回答する方式で実施した。

プログラミング言語には C 言語（Borland C/C++ Compiler 5.5）を用い、プログラムを入力するエディタには統合開発環境の CPad for Borland C++ Compiler Version 2.31 を用いた（いずれも無償でインターネットからダウンロード可）。

相互評価のために教育用 SNS「ednity」を用いた。調査対象校の「情報の科学」の授業において、ednity は日常的に使われており、生徒はその操作には習熟している。生徒は ednity に本

名で登録しており、相互評価は実名で行った。

2.6. 事前質問紙調査

事前質問紙調査では、プログラミングの経験やプログラミングに難しいイメージを持っているか、自分がプログラミングをどの程度できるようになると思うかといった質問項目を用意した（表2）。なお、選択式の回答方法をとる質問項目の説明は後述するため、ここでは割愛する。

2.7. 事後質問紙調査

事後質問紙調査では、プログラミングは難しいと思ったか、楽しいと思ったか、自分がプログラミングをどの程度できるようになったと思うかなどについて問う質問項目を用意した（表3の post01 から post10）。

加えて、教育用 SNS を介したコメントのやりとりがプログラミングを学習する意識に影響を与えたのかを確認するための質問を用意し、無作為で2クラスに対して追加の調査を行った（表3の post11 および post12）。

なお、表3の質問項目のうち、選択式の回答方法をとる質問項目の説明は後述するため、ここでは割愛する。

2.8. 分析対象者

調査対象者のうち、調査方法に不備のあった1クラスを除いた5クラスのうち、事前質問紙調査および事後質問紙調査の両方に回答している186名（男：115名 女：71名）を分析対象者とした。また、事後質問紙調査における追加の質問については2クラス68名（男：40名 女：28名）を分析対象者とした。

表2 事前質問紙調査の質問項目

	質問項目	回答方法
pre01	これまでにプログラミングをしたことがありますか？	選択式 (2件法)
pre02	前の質問で「はい、あります。」と答えた方だけ回答してください。いつ（何歳のとき）、どこで、どのようなプログラミングをしたのか、具体的に記述してください	自由記述
pre03	「プログラム」とはどのようなものだと考えますか？自由に記述してください。	自由記述
pre04	「プログラミング」とはどのようなものだと考えますか？自由に記述してください。	自由記述
pre05	「プログラミング」は難しいというイメージがありますか？	選択式 (2件法)
pre06	前問の回答に関係なく全員に聞きます。プログラミングが難しいとしたならば、どのような点が難しいと思われがちなのでしょうか？	自由記述
pre07	自分の進路（大学等）において、プログラミングは必要だと考えますか？	選択式 (2件法)
pre08	プログラミングを学んだ後、自分はプログラミングがどの程度できるようになると思いますか？	選択式 (5件法)

表3 事後質問紙調査の質問項目

	質問項目	回答方法
post01	「プログラム」とはどのようなものだと理解しましたか？自由に記述してください。	自由記述
post02	「プログラミング」とはどのようなものだと理解しましたか？自由に記述してください。	自由記述
post03	プログラミングは難しいと思いましたか？	選択式 (5件法)
post04	前問で「そう思った」「かなりそう思った」人に聞きます。どのような点で難しいと思ったのでしょうか？	自由記述
post05	プログラミングは楽しいと思いましたか？	選択式 (5件法)
post06	データを扱うための、変数の宣言の仕方や%dの使い方は理解できましたか？	選択式 (5件法)
post07	計算の仕方は理解できましたか？	選択式 (5件法)
post08	if文の使い方は理解できましたか？	選択式 (5件法)
post09	プログラミングをどの程度できるようになったと思いますか？	選択式 (5件法)
post10	プログラミングができるできないにかかわらず、全員に質問です。今後、どのようなプログラムを作りたいですか？	自由記述
post11	教育用 SNS (ednity) で他者からコメントをもらうことが、プログラミングをする上で励みになりましたか？	選択式 (5件法)
post12	教育用 SNS (ednity) で他者のプログラムを実行してコメントを述べることにより、プログラムに関する知識や理解が深まったと思いますか？	選択式 (5件法)

post11, post12 は追加の質問であり、2クラスのみを実施した。

3. 結果

本研究では、事前質問紙調査および事後質問紙調査のうち、選択式の回答をとる質問項目の結果を中心に言及する。なお、表6から表10において、回答の左側の数値は得点を表している。

3.1. プログラミング経験の有無

事前に調査した質問「これまでにプログラミングをしたことがありますか？」については、分析対象者のうち88.2%が未経験者であり、11.8%が経験者であった(表4)。

3.2. プログラミングの難しさ

事前に調査した質問「プログラミングは、難しいというイメージがありますか？」については、未経験者で97.6%、経験者で90.9%が難しいと回答していた(表5)。

事後に調査した質問「プログラミングは難しいと思いましたか？」については、「そう思わなかった」と「まったくそう思わなかった」と回答した者を合わせた割合は、未経験者では29.8%、経験者では36.0%であった。また、その2つに難しいかどうか決めかねている「どちら

表4 プログラミング経験の有無

回答	回答数
0: いいえ, ありません。(未経験者)	164 (88.2)
1: はい, あります。(経験者)	22 (11.2)

下段()内は%

ともいえない」と回答した者を合わせると未経験者では59.7%、経験者では59.1%が難しいと回答していた(表6)。

3.3. プログラミングの楽しさ

事後に調査した質問「プログラミングは楽しいと思いましたか？」については、「そう思った」と「かなりそう思った」と回答した者を合わせて、未経験者では70.7%、経験者では81.8%が楽しいと回答していた(表7)。

3.4. 教育用 SNS についての質問

事後に調査した質問「教育用 SNS で他者からコメントをもらうことが、プログラミングをする上で励みになりましたか？」については、「そう思った」と「かなりそう思った」と回答した者を合わせて、未経験者では65.5%、経験者では70.0%が励みになったと回答していた(表8)。

また、「教育用 SNS で他者のプログラムを実行してコメントを述べることにより、プログラムに関する知識や理解が深まったと思いますか？」については、「そう思った」と「かなりそう思った」と回答した者を合わせて、未経験者では67.2%、経験者では60.0%が知識や理解が深まったと回答していた(表9)。

3.5. 主観評定に基づくプログラミング習得度

事前に調査した質問「プログラミングを学ん

表5 プログラミングは難しいか (事前)

回答	回答数	
	未経験者	経験者
0: いいえ, 難しいというイメージは持っていません。	4 (2.4)	2 (9.1)
1: はい, 難しいというイメージを持っています。	160 (97.6)	20 (90.9)

下段()内は%

表6 プログラミングは難しいか (事後)

回答	回答数	
	未経験者	経験者
1: かなりそう思った	15 (9.1)	1 (4.5)
2: そう思った	51 (31.1)	8 (36.4)
3: どちらともいえない	49 (29.9)	5 (22.7)
4: そう思わなかった	46 (28.0)	8 (36.4)
5: まったくそう思わなかった	3 (1.8)	0 (0.0)

下段()内は%

表7 プログラミングは楽しいか

回答	回答数	
	未経験者	経験者
1: まったくそう思わなかった	2 (1.2)	0 (0.0)
2: そう思わなかった	10 (6.1)	1 (4.5)
3: どちらともいえない	36 (22.0)	3 (13.6)
4: そう思った	86 (52.4)	12 (54.5)
5: かなりそう思った	30 (18.3)	6 (27.3)

下段()内は%

だ後, 自分はプログラミングがどの程度できるようになると思いますか?」については, 一人でプログラムが作れるレベルに相当すると考えられる「マニュアルなどをみながらプログラムを作れる」と「何も見なくても思い通りのプログラムを作れる」を回答した者を合わせると, 未経験者で 53.7%が, 経験者で 68.2%が一人でプログラムを作れるであろうと回答していた (表 10).

また, 事後に調査した質問「プログラミングをどの程度できるようになったと思いますか?」についても, 事前の質問項目と同様に「マニュアルなどをみながらプログラムを作れる」と「何も見なくても思い通りのプログラムを作れる」を回答した者を合わせて, 一人でプログラムが作れるというレベルの回答をした者は, 未経験者で 65.5%, 経験者で 68.1%であった (表 10).

表8 他者からもらうコメントが励みになったか

回答	回答数	
	未経験者	経験者
1: まったくそう思わなかった	1 (1.7)	1 (10.0)
2: そう思わなかった	6 (10.3)	1 (10.0)
3: どちらともいえない	13 (22.4)	1 (10.0)
4: そう思った	34 (58.6)	6 (60.0)
5: かなりそう思った	4 (6.9)	1 (10.0)

下段()内は%

表9 他者へコメントを述べることで学習の理解を深めたか

回答	回答数	
	未経験者	経験者
1: まったくそう思わなかった	1 (1.7)	1 (10.0)
2: そう思わなかった	3 (5.2)	1 (10.0)
3: どちらともいえない	15 (25.9)	2 (20.0)
4: そう思った	33 (56.9)	5 (50.0)
5: かなりそう思った	6 (10.3)	1 (10.0)

下段()内は%

次に, 上述の2つの質問項目において未経験者と経験者の回答の平均値に差があるかを t 検定により確認したところ, どちらの質問項目においても有意差がみられなかった. そのため, 分析対象者全員について, 事前の質問「プログラミングを学んだ後, 自分はプログラミングがどの程度できるようになると思いますか?」の回答を学習前の主観に基づくプログラミングのスキル得点とみなし, また, 事後の質問「プログラミングをどの程度できるようになったと思いますか?」の回答を学習後の主観に基づくプログラミングのスキル得点とみなして, この2つの測定値の平均に差がみられるかを対応ありの t 検定で分析した. その結果, $t(185)=3.31$, $p<.01$, $d=0.29$ で有意であり, プログラミング学習前の学習意識に比べて, プログラミング学習後の学習意識の方が, プログラミングができると感じている者が多いことが示された.

3.6. 相互評価と学習効果との相関関係

教育用 SNS を介した相互評価でコメントをやり取りした感想と, プログラミング学習につ

表 10 主観によるプログラミング習得度

回答	回答数			
	未経験者		経験者	
	事前	事後	事前	事後
1: まったく作れない	13 (7.9)	3 (1.7)	2 (9.1)	1 (4.5)
2: 誰かが作ったプログラムの模倣をする	9 (5.5)	7 (10.3)	2 (9.1)	0 (0.0)
3: 人の助けを借りながらプログラムを作れる	54 (32.9)	43 (22.4)	3 (13.6)	6 (27.3)
4: マニュアルなどを見ながらプログラムを作れる	82 (50.0)	108 (58.6)	13 (59.1)	14 (63.6)
5: 何も見なくても思い通りのプログラムを作れる	6 (3.7)	3 (6.9)	2 (9.1)	1 (4.5)

表 11 教育用 SNS を介した相互評価に関する質問項目と学習後の感想や理解度に関する質問項目との相関

■post11. 教育用 SNS で他者からコメントをもらうことが、プログラミングをする上で励みになりましたか？

	質問項目	相関係数
post06	データを扱うための、変数の宣言や%dの使い方は理解できましたか？	.32**

■post12. 教育用 SNS で他者のプログラムを実行してコメントを述べることにより、プログラムに関する知識や理解が深まったと思いますか？

	質問項目	相関係数
post05	プログラミングは楽しいと思えましたか？	.32**
post06	データを扱うための、変数の宣言や%dの使い方は理解できましたか？	.33**
post08	if文の使い方は理解できましたか？	.34**
post09	プログラミングをどの程度できるようになったと思いますか？	.36**

いての感想および主観評定に基づく理解度との相関関係を、事後の質問項目の回答を点数としたものを変数として分析した。なお、ここでは相関係数.30 以上のものを取り上げる (表 11)。

その結果、「教育用 SNS で他者からコメントをもらうことが、プログラミングをする上で励みになりましたか？」と「データを扱うための、変数の宣言や%dの使い方は理解できましたか？」の得点の間に正の相関がみられた ($r=.32$)。

また、「教育用 SNS で他者のプログラムを実行してコメントを述べることにより、プログラムに関する知識や理解が深まったと思いますか？」と次の4つの質問項目の得点の間に正の相関がみられた。

- (1)プログラミングは楽しいと思えましたか？ ($r=.32$)
- (2)データを扱うための、変数の宣言や%dの使い方は理解できましたか？ ($r=.33$)
- (3)if文の使い方は理解できましたか？ ($r=.34$)
- (4)プログラミングをどの程度理解できるようになったと思いますか？ ($r=.36$)

4. 考察

本研究では、高校生にプログラミングを学ばせる際、生徒が作成したプログラムを教育用 SNS にアップロードして相互評価をさせる授業実践を行い、プログラミングについての学習意識に関する質問紙調査を行った。

分析にあたってはプログラミングの未経験者と経験者に分けて分析したが、各質問項目において未経験者と経験者の回答の平均値に差があるかを t 検定により確認したところ、どの質問項目においても有意差がみられなかったため、以下では主として未経験者の結果をもとに考察する。

4.1. プログラミングは難しいというイメージの払拭

本研究における対象者は、授業開始前には、プログラミング経験に関係なく、90%以上の者がプログラミングは難しいというイメージを持っていた。これは、「ほとんどの生徒はプログラミングの経験がなく、自分でプログラムが作れ

るのだということを想像するものは少ない」とする谷川（2014）の言及と合致する。演習2終了時の質問紙調査では、プログラミングは難しいかとの質問に対して「そう思わなかった」と「まったくそう思わなかった」と回答した者を合わせて29.8%、難しいかどうか決めかねている「どちらともいえない」と回答した者を合わせると59.7%が難しいと感じていた。これに対して、プログラミングは楽しいかとの質問には、「そう思った」と「かなりそう思った」と回答した者を合わせて70.7%であり、プログラミング学習に対する難しいイメージを払拭できている様子がうかがえる。それが、教育用 SNS を活用した相互評価によるものかは、次の2つの質問の得点の相関係数をみることにより検討できる。「教育用 SNS で他者のプログラムを実行してコメントを述べることにより、プログラムに関する知識や理解が深まったと思いますか？」と「プログラミングは楽しいと思えましたか？」の得点の間に正の相関がみられたことから、教育用 SNS を介した相互評価でコメントを述べるのが、楽しく学ぶことに好影響を与えていることが推察できる。

4.2. 教育用 SNS で他者から受け取るコメントが学習者に与える影響

「教育用 SNS で他者からコメントをもらうことが、プログラミングをする上で励みになりましたか？」の質問には、「そう思った」と「かなりそう思った」と回答した者を合わせると、6割以上の者が他者からのコメントに励まされてプログラミングの課題を遂行することができたことがうかがえる。

4.3. 教育用 SNS で他者に対して述べたコメントが学習者に与える影響

「教育用 SNS で他者のプログラムを実行してコメントを述べることにより、プログラムに関する知識や理解が深まったと思いますか？」の質問には、「そう思った」と「かなりそう思った」の回答を合わせると、6割以上の者が他者にコメントを述べることで知識や理解を深めていった様子がうかがえる。またそれは、質問項目「教育用 SNS で他者のプログラムを実行してコメントを述べることにより、プログラムに

関する知識や理解が深まったと思いますか？」と「プログラミングは楽しいと思えましたか？」の得点の相関について分析した結果からもうかがえる。自分が理解していないと他者にコメントを述べることは難しいため、教師の説明をよく聞くようになったり、教材をよく調べるようになったりしたものと推察できる。

4.4. 主観評定に基づくプログラミング習得度

事前・事後におけるプログラミング習得度に関する質問の回答を、学習前後の主観に基づくプログラミングのスキル得点とみなして t 検定で分析した結果からは、事後の質問紙調査においてプログラミングができていると感じている生徒が6割以上いたといえる。これまで述べてきた考察と合わせて考えると、教育用 SNS を介した相互評価によりプログラミングができるという自己効力感を高めることができたと考えられる。ただし、本研究では質問紙調査における主観的な習得度を分析しているため、実際にプログラミングのスキルを習得できたかまでは言及できない。

演習2でオリジナルのプログラムを作る際には、教材の他にインターネット上にある C 言語マニュアルを活用して教師が教えていないプログラムを作成した生徒も少なからずいた。これは、文部科学省（2015）のプログラミングガイドで報告されている「児童生徒はある程度自分でコンピュータを動かすことができるようになると、自分自身で発展・改良するといった工夫をする様子が見られる」といった知見と一致する。

4.5. 本研究における授業を実践する上で考慮されるべき課題

4.5.1. 授業全体に時間がかかる

プログラミングを習得させるには、例題を提示し、その説明をして知識を獲得させ、さらに演習をさせることにより技術を習得させるというプロセスが必要である。また、本研究では、オリジナルのプログラムを生徒に作成させたが、生徒自らがアイデアを出してオリジナルのプログラムを完成させる過程には文献を調べたりする時間も必要である。総じて、プログラミング学習は多くの時間を要するものである。しかし、

「情報の科学」はプログラミング以外にも習得させる知識が多い。本研究の授業の流れ(表1)の9時間目にある「繰り返し文」は、例題の提示にとどまり、生徒への演習はさせていない。本研究の調査対象とした演習と同様に、生徒にオリジナルのプログラムを作成させ、教育用SNSを介して相互評価をさせるのであれば、少なくともあと3時間、全体で12時間の計画を立てる必要がある。

4.5.2. プログラミングに精通した教師が必要

生徒にオリジナルのプログラムを作成させるためには、生徒が解決できないプログラミング上の問題に対応できるだけの力を持った教師が必要である。文部科学省(2013)が報告しているように、プログラミングを指導できる教師が少ないことが、プログラミングを授業に導入する上で問題となっている。本研究における授業を実践する場合には、教師がプログラミングに精通していることが前提となる。

4.5.3. 相互評価に時間がかかる

パソコンで教育用SNSを介した相互評価をする上ではキーボードによるタイピングが必須である。タイピングには時間がかかるため、単位時間内に一人の生徒がコメントを付けられる相手の数は多くはない。本研究では、相互評価の際に評価する相手をあらかじめ指定していたが、SNSの性質上、ディスプレイ上には最新のコメントから順に表示されるため、その評価対象者を捜し出すのにも時間を要した。そのため、相互評価にも時間がかかることを考慮した授業設計が求められる。

5. まとめと今後の課題

本研究では、高等学校の教科「情報」におけるプログラミングの学習の際に、教育用SNSを介した相互評価がプログラミングに対する生徒の学習意識に及ぼす影響を検討した。

その結果、教育用SNSを介した相互評価が生徒のプログラミングに対する学習意識に、「難しさを払拭して楽しさを感じさせる」、「課題遂行の糧となる」、「知識・理解を深める」、「プログラミングができるという自己効力感を得る」といった好影響を及ぼすことが示唆された。

研究方法における今後の課題としては、質問紙調査の質問項目についての見直しや、教育用

SNSでのコメントの内容分析などが挙げられる。授業実践における今後の課題としては、限られた時間の中でプログラミングを習得させるために授業デザインを改善すること、教育用SNSを介した相互評価を効率的に行う方法を検討することが挙げられる。

参考文献

Griffin, P., McGaw, B., and Care, E. (2012)

Assessment and Teaching of 21st Century Skills. Springer, Netherlands

文部科学省(2011) 教育の情報化ビジョン。

http://www.mext.go.jp/b_menu/houdou/23/04/_icsFiles/afiedfile/2011/04/28/1305484_01_1.pdf (参照日 2016.01.10)

文部科学省(2013) プログラミング学習にかかわる動向について。

https://www.kantei.go.jp/jp/singi/it2/senmon_bunka/jinzai/dai7/siryou5.pdf (参照日 2016.04.09)

文部科学省(2015a) 教育課程企画特別部会 論点整理。

http://www.mext.go.jp/component/b_menu/shingi/toushin/_icsFiles/afiedfile/2015/12/11/1361110.pdf (参照日 2016. 04. 01)

文部科学省(2015b) プログラミング教育実践ガイド。一般社団法人ラーン・フォー・ジャパン

http://jouhouka.mext.go.jp/school/programming_zirei/ (参照日 2016. 04. 01)

内閣府(2015) 世界最先端IT国家創造宣言。

<https://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20150630/siryou1.pdf> (参照日 2016.03.29)

総務省(2015) 高校生のスマートフォン・アプリ利用とネット依存傾向に関する調査報告書。

http://www.soumu.go.jp/main_content/000302914.pdf (参照日 2016.04.10)

谷川佳隆(2014) ペタ語義: なぜプログラミング教育が必要なのか。情報処理,55(10), 1147-1147.