

高校生の能動的なプログラミング学習を促す 授業の設計と効果の検討 ーグループ学習と教育用SNSによる相互評価の導入ー

菅井 道子^{1,2,a)} 堀田 龍也^{2,b)} 和田 裕一^{2,c)}

概要: 本研究では、高等学校教科「情報」におけるプログラミング学習の際に、グループ活動により学習者が能動的にプログラミング演習を行うことにより、計算論的思考を育成する授業を設計した。具体的には、生徒はC言語プログラムの基礎を教師から学んだ後、演習時には生徒同士が教え合い・学び合う授業である。生徒自らが課題設定をしたプログラム作成の演習時には、作成したプログラムの相互評価と修正を繰り返すことにより完成度を高めさせた。相互評価の場面では、教育用SNSを介した相互評価と、席を移動しながら相互評価を行う方法の比較をした。本稿では、授業の前後および演習後に得た質問紙調査の回答を分析し、授業の効果と課題について検討した。

Investigating Efficacy of Lesson Design to Encourage High School Students' Active Learning of Programming: Introduction of Group Learning and Mutual Evaluation Using SNS for Educational Use

MICHIKO SUGAI^{1,2,a)} TATSUYA HORITA^{2,b)} YUICHI WADA^{2,c)}

1. はじめに

内閣府の「世界最先端IT国家創造宣言 [1]」では、世の中の様々な課題解決に必要な論理的思考力や創造性、情報活用能力などの汎用的な力を育成する必要があるとされている。これを受けて初等中等教育の次期学習指導要領の改訂が進められており [2]、2020年以降は小学校から高等学校までの全校種にてプログラミング教育が実施されることとなる。

高等学校においては、現行の学習指導要領 [3] のもと

で、教科「情報」の科目「情報の科学」でプログラミングの指導が行われている。文部科学省はプログラミング指導のための参考資料である「プログラミング教育実践ガイド [4]」を発行しており、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事例が取り上げられている。

研究者らによる高等学校での実践としては、高岡ら [5] がグループ学習により高校生にWebアプリケーションの作成をさせて実用的プログラミングスキルの習得に好影響を与えていると報告している事例や、兼宗ら [6] がデータベース操作を取り入れたプログラミング授業の実践が高校生に関心を持たせて積極的に取り組める内容であったと報告している事例がある。

学習者同士の学び合いをプログラミング授業に取り入れた研究としては、生田目 [7] の大学生を対象としてその有用性を報告している事例や、吉田ら [8] の大学生を対象と

¹ 宮城県仙台第三高等学校
Miyagiken Sendai Third High School, Sendai, Miyagi, 983-0824, Japan

² 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University, Sendai, Miyagi, 980-8579, Japan

a) m.sugai@cog.is.tohoku.ac.jp

b) horita@media.is.tohoku.ac.jp

c) yuwada@cog.is.tohoku.ac.jp

して教員が「教えない」ことにより学生同士が主体的に学び合ってプログラムを完成させたことを報告している事例がある。このほか、インタラクティブなコミュニケーションツールである教育用 SNS をプログラミング学習に活用し、生徒同士の学び合いやプログラムファイルの共有場面で利用した菅井ら [9] の研究がある。菅井らは、高校生の C 言語プログラミング学習の際に教育用 SNS を介した相互評価を取り入れ、学習者の意識調査を行った。その結果から、学習者に「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たことを報告している。いずれの研究の授業も、書く・話す・発表するなどの活動への関与と、そこで生じる認知プロセスの外化を伴う能動的な学習 [10] であるアクティブラーニングの学習形態の一つといえる。

これらの先行研究をもとに考えると、高等学校でのプログラミング授業として学習者同士の学び合いと教育用 SNS を介した相互評価を取り入れたアクティブラーニング型の授業を実施することは、プログラミング学習を通して計算論的思考 (Computational Thinking) を育成することを目標の一つとする高等学校情報科での授業の効果を高めるために有用であると考えられる。しかしながら、高校生を対象とした、アクティブラーニング型のプログラミング授業の研究は、高岡ら [5] の研究を除けばまだ少ない。高等学校情報科でプログラミングの指導をする際に参考にできるプログラミング授業の実践例とその効果に関する資料の蓄積が求められている。そこで本研究では、プログラミングを習得する際に、グループ活動を中心とした学習者同士の学び合いと教育用 SNS を介した相互評価を導入した授業を設計・実施し、その授業が生徒の学習意識や態度にどのような影響を及ぼすのかについて、授業の前と授業の中で実施した演習が終わるごとに調査した質問紙の回答をもとに検討した。

2. 設計した授業

本研究で設計した授業のスケジュールを表 1 に示す。授業を実施する上で授業者と生徒が留意することは以下の点である。

(1) 授業者が留意すること

- C 言語の文法は必要最小限のことだけ教える
- 命令と対応するハードウェアを意識させる説明をする
- システムエンジニアの思考を織り交ぜながら説明をする
- 毎回の授業の冒頭では、前回の授業の内容を復習する
- 授業で一度、復習で再度教えたことに関する質問には答えない

(2) 生徒が留意すること

- 誰かに質問する前に、過去の資料を自分で調べる
- 自分の記憶力を過信しない

表 1 授業スケジュール

時間	学習内容
1	「プログラムによる計測・制御」の体験
2	標準入出力
3	演習 1 (作成)
4	演習 1 (相互評価・修正・提出・振り返り)
5	プログラミングの基礎、エラーへの対処法、順次、計算
6	関数の使い方、演習 2 と演習 3 (作成・提出・振り返り)
7	判断 (if~else 構文)、演習 4 (作成・提出・振り返り)
8	入れ替えのアルゴリズム
9	演習 5 (作成)
10	演習 5 (作成・相互評価・修正)
11	演習 5 (相互評価・修正・提出・振り返り)
12	繰り返し (while 構文)
13	繰り返し (for 構文)、演習 6 (作成)
14	演習 6 (作成・提出)、演習 6 および授業全体の振り返り

1 授業時間は 50 分

- 極力自分のグループの中で解決する。自分のグループで解決できない場合は他グループの力を借りる
- 演習が終わっていない人がいないか気配りし、終わっていない人がいたら援助を進んでする
- 演習の振り返りをするときには、自分がしたこと、グループでしたこと、その行為が社会のどの場面と関連していてどのように役に立つのかを意識する

2.1 授業環境

2.1.1 使用したプログラミング言語と実行環境

プログラミング言語は C 言語を使用した。コンパイラは Borland C++ Compiler 5.5*1、開発環境は Cpad for Borland C++ Compiler*2を使用した。

また、プログラムが実際にどのような場面で使われているかを理解させる体験実習の際には、マイコンの Arduino*3および Arduino IDE、温度センサを使用した。

2.1.2 教室の設備および環境

授業を実施するコンピュータ室には、生徒用および教員用のデスクトップパソコンがある。コンピュータ室には有線 LAN が整備されており、生徒は各自のアカウントを使って授業で使用するサーバ (以下、授業用サーバ) にログインして使用している。e-learning システム Moodle Version2 を授業用サーバ上で稼働させており、課題の提出や質問紙調査の回答は Moodle で行っている。有線で接続されているインターネットから教育用 SNS を利用できる。本研究で利用した教育用 SNS は、クラスのページの中で小グループの作成が可能な Edmodo*4である。

2.2 グループ分け

授業前に実施した質問紙調査 (詳細は後述する) より、プログラミング経験のある者が同じグループにならない、プログラミングに興味のない者だけが集まらない、女子が

*1 現在は提供されていないバージョンである。同じ提供元から現在ダウンロードできる無料コンパイラは、次の URL から入手可能である。 <https://www.embarcadero.com/jp/free-tools>

*2 送金義務のないシェアウェア。 <http://cpad.michikusa.jp/>

*3 <https://www.arduino.cc/>

*4 <https://www.edmodo.com/>

1 グループに 1 人だけにならない、といった点に配慮して、4 人 1 組のグループ分けをした。

2.3 学習内容

C 言語プログラミングで学習する内容は、1 時間目の「プログラムによる計測・制御」の体験を除けば、大学や専門学校での C 言語の基礎学習とほとんど変わらない。大学や専門学校での C 言語の基礎学習と異なる点は、必要最小限のことしか教えないことと、プログラミングの導入部分でのスモール・ステップを特に小さく設定したことにある。テキスト代わりのプリントは授業者が作成して PDF 化し（以下、PDF プリントとする）、授業用サーバに入れておいた。2.3.1 以降では、授業設計上、高校生を対象とした授業であることを意識して特に留意した点について述べる。

2.3.1 「プログラムによる計測・制御」の体験

プログラムが身の回りのどのような場面で利用されているかを理解させることを目的として、「プログラムによる計測・制御」の体験実習を行うこととした。生徒が同じ体験を共有することにより、以降の授業において同じ見地で互いに話ができるようにするためでもある。具体的には、文部科学省の「プログラミング教育実践ガイド [4]」に掲載されている「C 言語と電子工作・センシングの基礎学習 [11]」にある、マイコンの Arduino と電子部品の温度センサを用いた電子工作とセンシングの実習を行った。Arduino に温度を計測するプログラムを書き込むことにより、温度センサを介して得られた温度をパソコンのディスプレイに表示する。これにより、例えばエアコンが室温の変化を感知して風を送り出すときにプログラムが関与していることを理解できるようになる。プログラミング実践ガイドでは生徒にプログラムを作成させていたが、本研究の授業ではプログラムの利用場面を理解することが目的であるので、完成したプログラムをあらかじめ配付した。

2.3.2 標準入出力についての学習

初めて作成するプログラムは、何もしないプログラム、つまり、main 関数の枠組みだけを PDF プリントから書き写すこと（いわゆる「写経」）とした。タイピングが苦手であっても、入力するのに多くの時間を必要としない量のコードであり、入力ミスもみつけやすいためである。さらに、生徒の中にはプログラムには動きがあるものという先入観がある者もいるため、実行しても何も起こらないプログラムもあるのだと知ることにより、プログラミングに対する興味や関心を引くこともできる。

その後、標準入出力関数（scanf 関数と printf 関数）の入力を徐々に増やすことにより「コンピュータと対話するプログラム」を完成させることにした。ここでは、整数の入力と文字列の入力の 2 つしか扱わないことにした。また、システムエンジニアが入出力のメッセージを設計するときの思考について教えた。

全員提出の課題である演習 1 として、自分でメッセージを考えて「コンピュータと対話するプログラム」を作成させ、グループ内での相互評価およびクラス内での相互評価をした。相互評価での他者からの意見を受けてプログラムの修正をさせ、完成したプログラムを提出させた。

2.3.3 プログラミングの基礎知識

コンパイルエラーと実行時エラーの違いを教え、エラーの発見および修正の仕方を教えた。また、プログラミング言語の種類と、プログラムの三要素、アルゴリズムについて解説した。

2.3.4 順次処理と四則演算・関数

プログラムの三要素の 1 つ目としての順次と、四則演算・関数の学習では、演算式の読み方と作り方を教えた。

演習課題は、全員提出の課題である演習 2「三平方の定理により三角形の斜辺の長さを求めるプログラム」と、演習 2 のプログラミングを早く終えた人向けに任意提出の演習 3「ヘロンの公式による三角形の面積を求めるプログラム」とした。

2.3.5 判断処理

プログラムの三要素の 2 つ目として判断を教えた。if～else 構文により、入力された整数値が奇数か偶数かを判定するプログラムを例示した。全員提出課題の演習 4 として、「数の大小判定プログラム」を作らせた。

入れ替えのアルゴリズムを教えた後、全員提出課題の演習 5 をさせた。作成条件は、順次・判断処理に加えて計算も必ずするプログラムを自分で発想・設計して作ることにした。グループでの相互評価の後、クラスでの相互評価をした。相互評価での他者からの意見を受けてプログラムの修正をさせ、完成したプログラムを提出させた。

2.3.6 繰り返し処理

プログラムの三要素の 3 つ目として繰り返しを教えた。1 から 10 までの総和を求めるプログラムを、数学で学んだ公式は使わずにこれまでに教わった方法で作るとしたらどのようなプログラムになるかをグループで話し合わせた。足す数が等差数列であることに着目してプログラムを書き直し、それを While 文で書き表した後、さらに for 文に書き直した。

演習課題は、全員提出の課題である演習 6「階乗計算プログラム」とした。

3. 研究の方法

3.1 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）を対象に行った。

3.2 調査時期

平成 28 年 10 月～12 月

3.3 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。授業は、2章で示した授業を実施した。

演習1と演習5の自分で発想・設計したプログラムについて相互評価をする場面で、調査対象者を3つの群に分けた。1つ目は2回の相互評価ともSNSを利用する群（これを「SNS群」と呼ぶ）、2つ目は2回の相互評価とも評価対象者の席まで移動して評価する群（これを「移動群」と呼ぶ）、3つ目は1回目の相互評価を評価対象者の席まで移動して評価をし、2回目の相互評価はSNSを利用する群（これを「MIX群」と呼ぶ）である。

プログラミング授業を実施する前と、授業で課した全員提出課題である演習（演習1, 演習2, 演習4, 演習5, 演習6）を生徒が提出した時点で行った質問紙調査の回答、合わせて6回分をもとに分析した。

4. 結果

4.1 授業前の質問紙調査の結果

授業前に実施した質問紙調査は、調査実施日に出席した227名の回答を分析した。

4.1.1 授業前におけるプログラミングの習得状況

中学校の技術・家庭科で、質問項目として挙げたプログラミングに関連する学習項目を教わった記憶があると答えた者の割合を表2に示す。教わったことが記憶に残っている者が一番多かった学習項目が「プログラム」と「フローチャート」で24.7%、逆に、教わったことが記憶に残っている者が一番少なかった学習項目が「アルゴリズム」で7.5%であった。

表2 中学校で習った記憶があると答えた者の割合

学習項目	割合
プログラム	24.7%
計測・制御	14.1%
ロボットの制御	18.9%
センサ	22.9%
アクチュエータ	12.8%
処理の手順	15.9%
アルゴリズム	7.5%
フローチャート	24.7%

また、コーディングを伴うプログラミング経験の有無についての質問に、ゲームを作る、ロボットや車などを動かすなど、実際にプログラミングをしたことがあると答えた者は、13.7%であった。

4.1.2 プログラミング授業に対する期待感と不安感

プログラミング授業への期待に関する質問について、「5:とてもそう思う」から「1:まったくそう思わない」まで（質問4は「5:とても興味がある」から「1:まったく興味がない」まで）の5件法で答えた回答結果の平均および標準偏差を表3に示す。

表3 プログラミング授業への期待に関する質問に対する回答の平均点と標準偏差

質問項目	M	SD
1. 「プログラミング」は自分にも簡単にできると思いますか?	2.4	1.0
2. 「プログラミング」は楽しいと思いますか?	3.4	1.1
3. 「プログラミング」は難しいと思いますか?	4.3	0.8
4. 「プログラミング」を学ぶことに興味はありますか?	3.8	1.0
5. 今後の自分にとって、プログラミングは必要な技能だと思いますか?	3.7	1.1

4.1.3 プログラミング習得に対する期待感と不安感

「授業後に自分がどの程度プログラミングできるようになると思いますか」との質問の回答を、表4に示す。自分一人でプログラムが作れるようになるだろうと期待している者が選択肢1と2を合わせて61.2%である一方、まったく作れないと考えている者は1.3%であった。

表4 「授業後に自分がどの程度プログラミングできるようになると思いますか」への回答割合

選択肢	人数	割合
1. 先生や友人の力を借りることも、説明書などを見ることもなく、自分一人でプログラムが作れる	15	6.6%
2. 説明書などを見ながら自分一人でプログラムが作れる	124	54.6%
3. 先生や友人の力を借りながらプログラムが作れる	71	31.3%
4. 誰かが作ったプログラムを書き写すことによりプログラムが作れる	14	6.2%
5. まったく作れない	3	1.3%

4.2 プログラミング授業に対する態度の経時変化

全授業に出席し、質問紙調査への回答に不備がなかった者を対象として、授業前に調査した質問紙の回答と全員提出課題である演習後に調査した質問紙の回答、合わせて6回の回答をもとに、「プログラミングは楽しいと思うか」、「プログラミングは難しいと思うか」、「プログラミングに興味はあるか」の3つの態度について、群（3群）×演習回（6回）の分散分析を行った。

4.2.1 「プログラミングは楽しい」という態度の経時変化

授業前に調査した質問「『プログラミング』は楽しいと思いますか?」の回答を楽しさの基準とみなし、各演習後に調査した質問「今回の演習は楽しかったですか?」について、6回の回答の平均値の経時変化を表5に示す。授業前の質問への回答は、「5:とてもそう思う」から「1:まったくそう思わない」の5件法で、演習後の質問への回答は、「5:とても楽しかった」から「1:まったく楽しくなかった」の5件法であった。分散分析の結果、回答の平均点に有意差はみられなかった。3群とも自分で発想・設計したプログラムを作成する演習5の後に最高点となり、4.5付近の値で「4:やや楽しかった」と「5:とても楽しかった」の間であった。

表 5 「プログラミングは楽しいか」という質問に対する回答の平均点と標準偏差

質問項目	M	SD
1. 授業前	3.3	1.0
	3.5	1.1
	3.5	1.1
2. 演習 1 後	4.3	0.6
	4.4	0.5
	4.5	0.7
3. 演習 2 後	3.9	0.9
	4.2	1.0
	3.9	1.0
4. 演習 4 後	4.0	0.8
	4.3	0.7
	4.2	0.6
5. 演習 5 後	4.4	0.7
	4.5	0.7
	4.5	0.7
6. 演習 6 後	3.7	0.9
	3.9	0.9
	3.9	1.0

上段：SNS 群 (n=107)
 中段：移動群 (n=56)
 下段：MIX 群 (n=35)

4.2.2 「プログラミングは難しい」という態度の経時変化

授業前に調査した質問「『プログラミング』は難しいと思いますか？」の回答を難しさの基準とみなし、各演習後に調査した質問「今回の演習は難しかったですか？」について、6回の回答の平均値の経時変化を表6に示す。授業前の質問への回答は、「5:とてもそう思う」から「1:まったくそう思わない」の5件法で、演習後の質問への回答は、「5:とても難しかった」から「1:まったく難しくなかった」の5件法であった。分散分析の結果、回答の平均点に有意差はみられなかった。3群とも繰り返しの演習6の後に最高点となり、4.3前後で「4:やや難しかった」に近い値であった。

表 6 「プログラミングは難しいか」という質問に対する回答の平均点と標準偏差

質問項目	M	SD
1. 授業前	4.3	0.8
	4.2	0.9
	4.2	1.0
2. 演習 1 後	3.5	1.0
	3.5	1.0
	3.3	0.9
3. 演習 2 後	4.3	0.7
	4.1	0.9
	4.1	0.8
4. 演習 4 後	3.6	1.0
	3.6	1.1
	3.5	0.9
5. 演習 5 後	4.0	0.8
	3.8	0.9
	3.9	0.6
6. 演習 6 後	4.4	0.9
	4.3	0.7
	4.2	0.6

上段：SNS 群 (n=107)
 中段：移動群 (n=56)
 下段：MIX 群 (n=35)

4.2.3 プログラミングへの興味の経時変化

授業前に調査した質問「『プログラミング』を学ぶことに興味はありますか？」の回答を興味の基準とみなし、各演習後に調査した質問「今回の演習は興味を持ってましたか？」

について、6回の回答の平均値の経時変化を表7に示す。授業前の質問への回答は、「5:とても興味がある」から「1:まったく興味がない」の5件法で、演習後の質問への回答は、「5:とても興味を持てた」から「1:まったく興味を持てなかった」の5件法であった。分散分析の結果、回答の平均点に有意差はみられなかった。3群ともプログラミング授業全体を通して平均点は、「4:やや興味を持てた」前後の値であった。

表 7 「プログラミングへの興味はあるか」という質問に対する回答の平均点と標準偏差

質問項目	M	SD
1. 授業前	3.6	1.0
	4.1	0.8
	4.0	1.0
2. 演習 1 後	4.2	0.7
	4.4	0.6
	4.5	0.7
3. 演習 2 後	4.1	0.8
	4.2	0.9
	4.1	0.9
4. 演習 4 後	4.1	0.9
	4.4	0.7
	4.1	0.6
5. 演習 5 後	4.3	0.7
	4.5	0.7
	4.3	0.7
6. 演習 6 後	3.8	0.8
	4.1	0.8
	4.0	0.8

上段：SNS 群 (n=107)
 中段：移動群 (n=56)
 下段：MIX 群 (n=35)

4.2.4 授業終了時のプログラミングに対する態度

最後の演習6を提出した時点で尋ねた質問についての回答結果の平均および標準偏差を表8に示す。質問1と質問2の回答は「5:難しいの方が断然強かった」から「1:楽しいの方が断然強かった」までの5件法、質問3の回答は「5:とてもあった」から「1:まったくなかった」までの5件法であった。一元配置分散分析の結果、群間の有意差はなかった。プログラミングをしているときは、SNS群は3.7で「4:難しいの方がやや強かった」に近く、移動群は3.5、MIX群は3.0で「3:どちらともいえない」に近い値であった。プログラムが完成したときは、3群とも「2:楽しいの方がやや強かった」前後の数値であった。また、プログラムがうまく動いてくれたときの達成感は、3群とも4.5以上で「5:とてもあった」に近い数値となった。

4.3 相互評価に対する態度の経時変化

相互評価を行った演習1と演習5の後で調査した3つの質問の回答の平均値について表9に示す。回答は、「5:とてもそう思う」から「1:まったくそう思わない」までの5件法であった。群(3群)×演習回(2回)の分散分析の結果、有意差はみられなかった。

演習5の後で調査した相互評価についての質問の回答の平均値について表10に示す。回答は、「5:とてもそう思う」

表 8 授業終了時に尋ねた質問に対する回答の平均点と標準偏差

質問項目	M	SD
1. プログラミングをしているときに 難しさと楽しさではどちらの 気持ちが強かったですか	3.7	1.1
2. プログラムが完成してうまく動いてくれたときに、 難しさと楽しさではどちらの 気持ちが強かったですか	2.1	1.1
3. プログラムが完成してうまく動いてくれたときに、 達成感がありましたか	4.5	0.6
	4.6	0.7
	4.7	0.6

上段：SNS 群 (n=117)
中段：移動群 (n=71)
下段：MIX 群 (n=36)

表 9 他の人に自分のプログラムを相互評価してもらうこと
についての質問への回答の平均点と標準偏差

態度	演習 1 後		演習 5 後	
	M	SD	M	SD
1. 他の人に自分のプログラムを 評価してもらうのは 嬉しかったですか。	4.1	0.7	4.3	0.7
2. 他の人に自分のプログラムを 評価してもらうのは 恥ずかしかったですか。	3.1	1.1	2.8	1.2
3. 他の人に自分のプログラムを 評価してもらうのは 辛かったですか。	2.1	1.0	2.0	1.0
	1.9	0.9	2.0	1.0
	2.0	0.9	2.1	1.2

上段：SNS 群 (n=115)
中段：移動群 (n=64)
下段：MIX 群 (n=36)

表 10 相互評価についての質問に対する回答の平均点と標準偏差

質問項目	M	SD
1. 相互評価は楽しかったですか	4.2	0.8
	4.5	0.7
	4.2	0.9
2. 相互評価をすることは、 プログラミングを習得する ために有益だと思いますか	4.6	0.6
	4.6	0.8
	4.4	0.7
3. 相互評価をすることは、 プログラムの完成度を高める ために有益だと思いますか	4.7	0.5
	4.7	0.5
	4.6	0.6

上段：SNS 群 (n=117)
中段：移動群 (n=71)
下段：MIX 群 (n=36)

から「1:まったくそう思わない」までの 5 件法であった。

演習 5 の後で実施した生徒の自由記述による演習の振り返りから得た、相互評価に関する感想の例を挙げる。

- 「他の人に評価してもらって正直嬉しかった。これからも頑張ろうと思える時間だった。」
- 「プログラムを完成させることが目標より、よいものにしていこうという考えに変わった。」
- 「他の人のものを見ると、自分になかったアイデアを持っていて、考えが深まった。」
- 「他の人にプログラムを試してもらうことで気づくことができた修正点もあったので、いろいろな人に試してもらえてよかったです。」
- 「自分のプログラムを評価されるのは若干恥ずかしいが、それでも何かを教えてもらえたり、アドバイスをもらえるのは嬉しかった。」

4.4 グループ学習での教え合い・学び合い

最後の演習である演習 6 を提出した時点で尋ねた「他者

に質問をして教えてもらいながら演習を完成させることに抵抗はありましたか。」との質問の平均点と標準偏差を表 11 に示す。回答は、「5:とても抵抗があった」から「1:まったく抵抗がなかった」までの 5 件法であった。一元配置分散分析の結果、群間の有意差はなかった。3 群とも平均点が「2:あまり抵抗がなかった」前後の値であった。

表 11 教え合いに抵抗はあったかとの質問に対する回答の平均点と標準偏差

質問項目	M	SD
1. 教え合いに抵抗はあったか	2.1	1.0
	1.9	0.8
	2.1	0.9

上段：SNS 群 (n=117)
中段：移動群 (n=71)
下段：MIX 群 (n=36)

演習 5 で自分で発想・設計したプログラムを作成した際に、プログラムのアイデアが浮かばないときに、誰かとのことについて会話したかどうかを尋ねた質問の回答を度数分布表として表 12 に示す。選択肢は、「2:はい、グループの人と会話しました」、「1:はい、グループ外の人と会話しました」、「0:いいえ、誰とも会話しませんでした」で、誰かと会話をしたときには影響が大きかった方で答えさせた。3 群とも 80%以上の者がアイデアを得るために誰かと会話をしていた。

表 12 「プログラムのアイデアを出すときに誰かと会話したか」という質問に対する回答の度数分布

群	選択肢			合計 (人)
	2	1	0	
SNS	65 (70.7%)	15 (16.3%)	12 (13.0%)	92
移動	39 (69.6%)	11 (19.6%)	6 (10.7%)	56
MIX	19 (67.9%)	4 (14.3%)	5 (17.9%)	28

各演習の後で実施した生徒の自由記述による演習の振り返りから得た、グループ学習での教え合い・学び合いに関する感想の例を挙げる。

- 「一人のプログラムがなかなか実行できなくてみんなで頑張って間違いを見つけて最終的に実行できてとてもよかった。」
- 「難しいと思ったけど作っているうちに楽しいと思うようになった。同じグループの人に自分はこうやりたいがどういう風にすれば良いかということアドバイスをもらい、自分が作りたいものを作ることができた。また、グループの人が困っているとき、どこがどう間違っているのかをグループみんなで考え、解決できた。」
- 「初めは自分の力だけで完成させる気でいたがどうしてもわからない部分があったのでグループの人に助けもらった。将来自分の力だけでできることというの

はあまり多くないと思う。だからこそほかの人に聞いて教えてもらいそれを咀嚼して次に同じ失敗をしないようにすることが大切だと思った。そうすることで全体として迅速に物事が進むと感じた。」このプログラミングの授業は自分には出来ないことを自覚してグループのメンバーと協力していくことでプログラミングのほかにも他人との協調性が養われていると思う。さらに自分のどこが悪かったのか探すという作業も重要だと思った。」

4.5 C 言語によるプログラミングの学習内容についての理解度と感想

各演習後に質問した C 言語によるプログラミングの学習内容の理解度について、表 13 に示す。回答は、「5:とても理解できた」から「1:まったく理解できなかった」までの 5 件法であった。群 (3 群) × 演習回 (2 回) の分散分析の結果、有意差はみられなかった。どの項目においても平均点が 4 前後であることと標準偏差の値から判断すると、プログラミング授業で学習した内容は、3 群とも「4:やや理解できていた」と推察される。

表 13 学習内容についての理解度に関する質問に対する回答の平均点と標準偏差

学習内容	M	SD
1. 標準入力 (scanf)	4.1	0.6
	4.1	0.7
	4.0	0.6
2. 標準出力 (printf)	4.1	0.7
	4.1	0.7
	4.0	0.7
3. 計算式の作り方	3.9	0.7
	4.0	0.7
	3.9	0.7
4. 関数の使い方	3.8	0.8
	3.8	0.8
	3.8	0.6
5. 判断 (if~else 構文)	4.2	0.8
	4.3	0.7
	4.2	0.5
6. 繰り返し (while 構文)	3.8	0.8
	3.9	0.8
	3.8	0.7
7. 繰り返し (for 構文)	3.7	0.9
	3.9	0.9
	3.8	1.0

上段: SNS 群 (n=117)
中段: 移動群 (n=80)
下段: MIX 群 (n=37)

5. 考察

本研究では、高校生にプログラミングを学ばせる際、アクティブラーニング型の授業に基づいたグループ学習を行うことと、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ことを提案し、席を移動しながら相互評価をする群との比較から、プログラミングについての態度に関する質問紙調査の結果を分析した。

生徒の主観による学習内容の理解度は、どの群において

も「やや理解できていた」で高い方であった。プログラミング学習は、難しいがやりがいがあって楽しく、完成した後の達成感を感じていることが、調査の結果から示された。

本研究の目的は、グループ学習と教育用 SNS を導入した相互評価の教育効果を検討することであった。以下にそれらの効果について考察する。

5.1 グループ学習による効果

最後の演習後に実施した質問紙調査の回答を分析した結果からは、生徒同士が教え合いながら学習することへの抵抗はあまりなかったことがうかがえた。授業者の内省によると、授業当初は他者と話すことが苦手な生徒が、グループ内で解決できそうな質問であっても授業者に聞くということが頻繁に起きていた。授業者は、そのような生徒には最初は柔軟に対応しつつ、グループでの教え合い・学び合いを推進するために、徐々に「授業で一度教え、次の授業で復習したことで、2 回説明したことはもう教えない」というルールを徹底していった。その結果、生徒は、分からなかったり、聞き漏らしたりしたことは他者に聞けばよいことから、自分たちで解決できる問題が多いことに気づき、既に教わったことを授業者に質問する回数が徐々に減った。それにより、授業者は既に教えたこと以外の質問をしたい生徒への対応がより多くできるようになったと内省している。生徒の授業後の口頭での感想からは、クラスメートの説明のスキルが上がっているとの意見も得られている。また、欠席しても、グループのメンバーが説明をしてくれることにより、欠席した生徒も前の時間の演習の遅れをすぐに取り戻せていたことが、生徒の自由記述の回答には多数見られていた。加えて、授業者の内省からは、プログラミング授業の回を重ねるにつれて、生徒が話をよく聞くようになったという。話をよく聞くことにより、教え合い・学び合いも活発さを増すことになることから、生徒・授業者ともに授業が楽しくなるという効果もあるであろう。

自分でアイデアを出し、プログラムを作成する演習 5 では、プログラムのアイデアが浮かばない時に、他者と会話をすることにより発想のヒントを得てプログラムの作成をしていた者が 80% 以上いたことが示された。他者からの学びがあったために、自分も他者のようにプログラミングができるようになりたい、もしくは、他者のように教えられるようになりたい、との自由記述の回答も多く見られたことから、生徒の態度にも好影響を与えられることが示唆された。

以上のことからグループ学習はプログラミングを学ぶ上で効果があったといえる。

5.2 教育用 SNS による相互評価の効果

授業前および各演習後に得た質問紙調査の回答を分析した結果からは、教育用 SNS による相互評価と、席を移動し

ながら行う相互評価では生徒の態度に明確な差は認められなかった。どちらの方法であっても、相互評価は楽しく、プログラミングを習得する上でも、自分の作成したプログラムの完成度を高めるためにも有効であると感じていることがみとれた。また、生徒の自由記述にも、どちらの方法であっても、相互評価の有効性を示す記述がみられていることから、プログラミング授業の効果を高めるには相互評価を取り入れることが有効であるとの肯定的な意見が得られた。

相互評価の後に他者から指摘を受けてプログラムを修正した後、再度他者にチェックをしてもらうという場面では、自席にしながらそれらの作業ができる教育用 SNS の方が有利であるように思われるが、本研究ではその点についての質問紙調査をしていないため、今後の検討課題である。

5.3 アクティブラーニング型授業を取り入れる上での留意点

「授業で一度教え、次の授業で復習したことで、2回説明したことはもう教えない」というルールを遵守することにより、生徒たちに教え合い・学び合いを促すことになり、生徒たちは授業者が新しいことを説明するとき話をよく聞くようになり、一度教えたことに対する質問が減った。しかしながら、生徒の中には他者と話すことが苦手な者もいるので、柔軟に対応する必要がある。

また、教え合い・学び合いを重視するために必要最小限のことしか教えない授業は、もっと高度なことを知りたいと思っているが、自分たちの力ではそこにうまく到達できない生徒の学びの機会を奪っている可能性もある。本研究の授業では、教えたこと以外にやりたいことがあるけれども、それをプログラムで実現する方法がわからなかったら授業者に聞くようにと生徒に予め伝えており、そのような質問をしてきた生徒も多数いた。しかし生徒の中には聞きたくても授業者に聞くことができない消極的な生徒もいるはずである。そのような生徒のことも考慮し、教えたこと以外のことについての質問があった場合には、クラスで共有し、さらに全クラスでも共有するといった配慮が授業者には求められる。

そのような点に留意し、本研究で実践したグループ学習と教育用 SNS を相互評価に導入した授業を行うと、プログラミング授業の効果が増すものと思われる。

6. おわりに

本研究では、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、アクティブラーニング型授業に基づいたグループ学習と、教育用 SNS による相互評価を取り入れた授業を実施し、生徒の学習意識に及ぼす影響を検討した。

その結果、グループ学習を取り入れることにより教え合

いに抵抗感なく、協調的にプログラミングを学ぶことができたことが示された。また相互評価については、教育用 SNS での相互評価でも、席を移動しながらの相互評価でも楽しく行うことができ、プログラミングを習得したり、作成したプログラムの完成度を高めることに有益であることが示された。

今後は、提出されたプログラムや、相互評価で生徒が書いた意見・感想などを内容分析するとともに、群間でパフォーマンスに差があるかどうかを検討することが必要である。

参考文献

- [1] 内閣府：世界最先端 IT 国家創造宣言，内閣府（オンライン），入手先（<http://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20160520/siryoul.pdf>）（参照 2017-01-16）。
- [2] 文部科学省：幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について（答申）（中教審第 197 号），文部科学省（オンライン），入手先（http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/1380731.htm）（参照 2017-01-16）。
- [3] 文部科学省：現行学習指導要領（本文、解説、資料等），文部科学省（オンライン），入手先（http://www.mext.go.jp/a_menu/shotou/newcs/youryou/index.htm）（参照 2017-01-16）。
- [4] 文部科学省：プログラミング教育実践ガイド，文部科学省（オンライン），入手先（http://johouka.mext.go.jp/school/programming_zirei/）（参照 2017-01-16）。
- [5] 高岡詠子，山内崇裕，滑川敬章：高等学校における実用的プログラミングの教育実践，情報処理学会論文誌 教育とコンピュータ，Vol. 2, No. 2, pp. 37-52 (2016)。
- [6] 兼宗進，白井詩紗香，竹中一平，長瀧寛之，鳥袋舞子，田邊則彦：データベースを利用した高校におけるプログラミング授業の実践と評価，情報処理学会研究報告，Vol. 2016-CE-134, No. 22 (2016)。
- [7] 生田目康子：ピア・レビューをとま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，Vol. 45, No. 9, pp. 2226-2235 (2004)。
- [8] 吉田葵，伊藤一成，阿部和広：ものづくり体験を通じたプログラミング授業の設計と評価，情報処理学会研究報告，Vol. 2016-CE-134, No. 12 (2016)。
- [9] 菅井道子，堀田龍也，和田裕一：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，日本教育工学会研究報告集，Vol. JSET16-2, pp. 155-162 (2016)。
- [10] 溝上慎一：アクティブラーニングと教授学習パラダイムの転換，東信堂，東京 (2014)。
- [11] 菅井道子：プログラミング教育実践ガイド実践事例 C 言語と電子工作・センシングの基礎学習，文部科学省（オンライン），入手先（<http://johouka.mext.go.jp/school/pdf/koukou.01.pdf>）（参照 2017-01-16）。