

# 反転授業で行われる大学のプログラミング教育に対する 上級生によるグループ指導の導入

盛 拓生<sup>1,a)</sup> 渡辺博芳<sup>1,2</sup> 水谷晃三<sup>1</sup> 荒井正之<sup>1</sup> 佐々木茂<sup>1</sup> 古川文人<sup>1,2</sup> 高井久美子<sup>1,2</sup>

**概要:** 大学におけるプログラミング教育の最初の科目において、学生にプログラミングへの興味を持たせ、ゼロからプログラムを書ける力を身につけさせることを目標として授業設計を行った。授業では、Student Assistant(SA)と呼ばれる上級学部生が、複数人の履修学生からなるグループについて、個々の学生のワークシートをそれぞれチェックし、直接学生を指導する体制をとることとした。本研究では、当該科目のSAによるグループ指導体制の構築、実際に行った授業の内容及び結果について検討、評価する。

## An Introduction of Group-Teaching by Senior Bachelor Students to a Programming Course Provided as a Flipped-Classroom in Higher Education

TAKUO MORI<sup>1,a)</sup> HIROYOSHI WATANABE<sup>1,2</sup> KOZO MIZUTANI<sup>1</sup> MASAYUKI ARAI<sup>1</sup>  
SHIGERU SASAKI<sup>1</sup> FUMIHITO FURUKAWA<sup>1,2</sup> KUMIKO TAKAI<sup>1,2</sup>

**Abstract:** We designed a course which is provided as the first course of programming-related education in University and we aim at letting students have an interest in programming and be able to write codes from scratch. In order to realize the above, we introduce Student Assistant(SA) to check worksheets written by students. Moreover, we have SAs to teach students directly. In this report, we show the configuration of the group-teaching by SAs, the contents and results of the course. We also evaluate the practice of this designed course.

### 1. はじめに

本学情報電子工学科(旧ヒューマン情報システム学科)におけるこれまでのプログラミング教育カリキュラムでは、プログラミング科目の課題に対してコピーアンドペーストによると思われるプログラムを提出する学生も少なくなく、その結果4年生の卒研配属の時点でも卒業研究を行うのに十分なプログラミング能力を身につけていない学生もいるような状況であった。

このような状況の下で、本学科では平成28年度入学者

を対象にプログラミング教育関連の科目も含めた大幅なカリキュラムの改定を行うべく検討してきた。

このカリキュラム改定において、本学科入学者に対する最初のプログラミング科目であるプログラミング1について、ゼロからプログラムが書ける、つまり、与えられた問題に対してプログラムの内容を学生自身の考えとしてまとめてプログラミングできる力を身につけさせることを目標として授業設計を行った。本研究では、[10]で報告した授業設計に基づき、実際に授業を行った内容及び結果について検討、評価する。

### 2. 関連研究

近年、プログラミング教育を対象とした反転授業の実践が報告されている[1][2][5]。また、上級生をプログラミング教育の補助員として利用する例も報告されている[1][2][7][8]。

<sup>1</sup> 帝京大学理工学部  
Faculty of Science and Engineering, Teikyo University, Utsunomiya 320-8551, Japan

<sup>2</sup> 帝京大学ラーニングテクノロジー開発室  
Learning Technology Laboratory, Teikyo University, Utsunomiya 320-8551, Japan

a) takuo@ics.teikyo-u.ac.jp

林らは、学部2年生を対象とするプログラミング科目において次の点を工夫した反転授業により、中間層の学生の学力が向上したことを報告している [1][2].

- 穴埋め問題、選択問題からなる Web 上の e ラーニング教材及び動画教材による予習を前提とした講義時間の削減、プログラム実習時間の増加.
- 予習段階での理解確認のための予習後の確認問題、授業開始時の確認テストの実施.
- 通常の出席確認に加えて、TA による課題の確認により出席とみなされる課題確認カードの導入による課題への取り組みの促進.
- 学生のグループ活動によるアクティブラーニングの側面を持つ協調学習の導入.
- 授業外のグループ学習をサポートする、時間外 TA の導入.

我々の授業設計が目指す方向性は、基本的には林らの研究と同じである。異なる点は、学生が必ずしもコンピュータを使用した学習に習熟しているわけではないプログラミングの初学者を対象としている点と、授業内の指導において教員による講義はせず、学生補助員の指導の下、紙で配布される課題に手書きでゼロからプログラミングに取り組むよう設計した点が挙げられる。

### 3. プログラミング関連科目の構成

平成 28 年度入学者を対象とする改定後のカリキュラムにおける 1 年次から 3 年次までのプログラミング関連科目の構成を表 1 に示す。

カリキュラム全体の構成の中で、プログラミング 1 の目標は次の通りである。

- プログラミングの基礎となる知識の修得.
- 基礎的なプログラミングの実践に必要な技術及び概念の修得.
- ゼロからプログラムを書くことができるようになること.

ここで、ゼロからプログラムを書くことができるようになるという目標はプログラミング 1 だけではなく、改定後のカリキュラムを構成するプログラミング関連の科目に共通する目標となっており、3 年次を終えた段階で、実践的なプログラムをゼロから書くことができるようになることを想定している..

そのため、プログラミング 1 では、プログラミングの基本構成要素、概念について、少なくとも文法として理解し、処理手順として利用できるようになることを目指す。

その次の段階として、プログラムの処理手順が箇条書きの日本語で与えられた場合に、習得した文法により学生自身が処理手順をプログラムとして書くことができるようになる、あるいは、プログラムが与えられた場合にその処理内容を自身で読み、説明することができるようになること

を目指した。また、プログラミング 1 の段階では、学生自身がアルゴリズムを考えつくことまでは要求しない。

プログラミング 2 では、プログラミング 1 で学んだ内容に続けてさらにプログラムの基礎を学び、オブジェクト指向プログラミングの基礎までを学ぶ。

プログラミング 2 と同時期に開講されるプログラミング演習 1 では、プログラミング 1 で学んだ基礎的な知識、技術、概念を利用して、簡単なプログラムを作成できるようになることを目指す。演習では、仕様の定義、テストコード、プログラムコードの作成、デバッグ、テストといったプログラム開発の流れを体験的に学ぶ。

これらの科目では、プログラミングの基礎を習得することが目的なので、プログラミング言語として、4.1 節で述べる Processing[3] を使用することとした。

2 年次以降のプログラミング関連の科目では、プログラミング言語として Java あるいはアセンブリ言語を通してさらに高度な内容を学ぶこととなる。

## 4. プログラミング 1 の授業設計

本研究が対象とする科目は 1 年前期に開講されるプログラミング 1 である。大学に入学した 1 年生が大学で受ける最初のプログラミング関連の科目であるプログラミング 1 に対して、これまでのプログラミング教育の問題点を解決するための授業設計を行った。

### 4.1 プログラミング言語の選定

本学科ではプログラミング言語として近年は Java 言語を用いたプログラミング教育を行ってきた。しかし Java 言語を用いてスタンドアロンで動作するプログラムを作成する場合、main メソッド含むクラスを定義しなければプログラムを実行させることができない。このため初学者であっても、クラス、メソッドの概念を十分に学ばないうちに、プログラム中に `class, public static void main(String[] args)` など理解できていない予約語を記述する必要があり、本質的なプログラミング学習に集中できない場合がある。

プログラミング 1 の履修学生の多くはプログラミング初学者であると想定される。そのような想定の下で我々が目的とするプログラミング教育を行うためのプログラミング言語について検討した結果、プログラミング 1 では、Processing[3] を採用することとした。

Processing の場合、コンソールに文字列を表示させるだけのプログラムであれば次のように記述するだけで良い。

```
println("Hello World");
```

また、グラフィカルなプログラムに関しても、例えば、キャンパス上に座標 (0, 0) から (100, 100) を結ぶ直線を描画するのであれば次のように記述するだけで良い。

```
line(0, 0, 100, 100);
```

Processing のプログラムは、実行に際しては内部的に

表 1 プログラミング関連科目の構成  
Table 1 The configuration of programming-related courses

| 1 年生      |                          | 2 年生                         |                             | 3 年生         |              |
|-----------|--------------------------|------------------------------|-----------------------------|--------------|--------------|
| 前期        | 後期                       | 前期                           | 後期                          | 前期           | 後期           |
| プログラミング 1 | プログラミング 2<br>プログラミング演習 1 | 情報科学プログラミング 1<br>プログラミング演習 2 | 情報科学プログラミング 2<br>情報科学基礎実習 1 | システム<br>開発実習 | 情報科学<br>実習 2 |

Java のコードに変換されてから実行されており Java との親和性が高い。

実際に Processing では Java モードが用意されており Java のコードを記述することも可能である。これより、2 年次以降で Java を学習する際の学習コストが下がることが期待できる。

#### 4.2 教科書の選定

Processing をプログラミング言語として採用するに際して、教科書の利用について検討した。

Processing に関連する図書を調査したところ、コンピュータによるアート表現、デザインを目的とするもの、あるいは画像処理、情報の視覚化を目的とするものが大半を占めた。

プログラミング 1 における我々の目的は、Processing そのものの学習ではなく、Processing を通してプログラミングの基礎を学ぶことにある。そのため、我々は Processing 開発者が、コンピュータプログラムの入門書を求める人を対象読者として書いた“Processing をはじめよう (Make: Projects)”[4] を教科書として利用することとし、不足する内容については独自教材で補うこととした。

#### 4.3 反転授業の検討

近年、反転授業が注目されている。本学でもプログラミング教育あるいは情報教育に対する実験的な導入、検討がなされており、その有効性が確認されている [5][6]。これらの結果を踏まえてプログラミング 1 では、次の方針のもとで反転授業を導入することとした。

- イントロダクション、試験以外の授業は原則的に反転授業とする。
- 反転授業では、講義ビデオによる事前学習を義務付ける。
- 講義ビデオの視聴を促し、学生に自身の理解度を把握させるために、“事前課題ワークシート”への記入を義務付ける。
- 授業の終わりに、理解確認のための振り返り課題に取り組ませる。
- 授業中に講義は行わず、学生はグループ単位で学習活動を行う。
- 各グループへの直接的な指導は上級学部生が行う。

#### 4.4 ワークシートの検討

プログラミング 1 では、学生にゼロからプログラムが書ける力を身に付けさせることを目標としている。そのため我々は、紙に印刷した課題をワークシートとして配布し、学生に手書きで解答を記入させ、その内容をスタッフが確認し、解答が正しければ計算機上で実行等を行うこととした。この場合、教員のみで全ての学生のワークシートをチェックすることは困難であると考え、4.5 節で述べる学生補助員を導入することとした。

ワークシートについては、各授業に対して反転授業における講義ビデオを授業前に視聴していることを確認し、自身の理解度を確認するための“事前課題ワークシート”、実際に授業内で取り組むための“授業内課題ワークシート”、授業で学んだ内容を振り返るための“振り返り課題ワークシート”の 3 つを作成することとした。また、授業内課題を済ませた学生に対する課題として、必要に応じてオプション課題を作成することとした。オプション課題は、授業内課題終了後、あるいは授業時間外に学生が取り組むことを想定している。

ワークシート内の課題は、コーディング、コードリーディング、その他に分類し、それぞれ難易度のレベルを表示し、学生が自身の理解度に応じて課題を選択し、取り組むことができるようにした。学生が自身の理解度を把握するための指標としては、4.6 節で説明するルーブリックを利用することとした。

#### 4.5 学生補助員によるグループ指導に関する検討

授業中、学生はワークシート上の課題に対して解答を作成することとなる。その際に、学生は個々の課題について紙で配布されるワークシート上に手書きで解答を作成し、上級学部生が務める学生補助員に解答をチェックしてもらい、解答が正しければ計算機上で実行等を行うことになる。そのため、実際の指導もワークシートを介して学生補助員が直接行うことが有効であると考えた。

その場合の学生補助員の指導方法、配置等については、森田らが実施している SA と呼ばれる学部学生補助員を利用した授業運営、演習教育の例 [7][8] を参考にすることとした。以降、学生補助員を SA(Student Assistant) と呼ぶ。

##### 4.5.1 SA の活動に関する検討

森田は、SA による指導体制を構築する際に教員直下に SA を一律に配置する構成にするべきではないとして、教

員, TA, SA の階層を持つ指導体制を構築している [7]. これは, 非階層的な構成の場合, SA が対応できない場合は全て教員が対応することとなり, 教員側で対応できなくなり, 結果として授業に支障が出るためとしている.

これを踏まえて, プログラミング 1 では 1 回の授業を複数の教員で担当するチームティーチングで行うこととした. 各教員は複数の SA からなる SA チームを指導し, SA チームを構成する個々の SA に対して 10 人以下の学生からなるグループを構成し, このグループ単位で SA の指導の下で学習活動を行うこととした.

各回の授業における指導体制概要を図 1 に示す.

#### 4.6 ルーブリックの検討

本研究に先立ち, 我々はプログラミング教育関連科目のカリキュラム再編に際してカリキュラム全体で利用するためのルーブリックの設計を行なっている [9].

我々が作成したルーブリックの特徴を次に示す.

- 学生が自己評価のために使用する.
- 3 年間の教育カリキュラムで共通に使用する.
- 毎回の授業で日常的に使用する.

### 5. プログラミング 1 の授業構成

#### 5.1 SA への指導体制

プログラミング 1 の開講に先立ち, 主に SA 候補の学生を対象とするプログラミング勉強会を実施した [9]. また, [7], [8] を参考に SA の活動指針となる SA マニュアル, SA 活動を記録するための SA シートを作成した.

##### 5.1.1 SA マニュアル

森田は [7] において, SA がすべきことをまとめた “学生教育補助員 (SA) マニュアル” を作成したことを報告している. 森田のマニュアルは, 実際の指導において多く見受けられた事例を 5 つのケースに分類し, それらのケースにどのように対応すべきかを指導法とコメントとして記述した事例集としてまとめられている. さらに, 各 SA による後輩の SA となる学生へ向けたメッセージが掲載されている.

また, 森田は [7] において, SA を活用する場合の最も重要な点として, “果たすべき役割を可能な限り具体的に指示すると言う\*1 こと”, “補助員との意思の疎通を密に行い, 指導方針や役割分担等を明示化し意思統一を図ること” を挙げている.

我々はこれらを参考にプログラミング 1 の SA に向けたマニュアルを作成した. 我々の作成した SA マニュアルでは, 図 1 に示す授業の指導体制を示し, 授業の概要, SA の役割, SA の活動内容, 学生からの質問への対応, 授業進行の概要, 授業中の各時間帯ですべきことについて, SA と

して何をするか, どのように振る舞うべきかを明示的に, 主に箇条書きでまとめている.

##### 5.1.2 SA シート

森田は [8] において, “集団で指導に当たる場合, SA 達の指導経験の蓄積を共有できるようにすることが有効である” と述べている. これを受けて我々は, SA が自身の担当するグループの各学生について, 事前課題グループ活動への参加状況, 事前課題ワークシートの提出状況, 授業内課題個人活動進捗, 授業内課題グループ活動進捗, 授業全体における個人活動概要, 所見等を記入するための SA シートを作成し, SA マニュアルにおける SA の活動内容の項に, SA シートへの必要項目への記入を明示することとした.

#### 5.2 エリアによるグループの分類

4.5 節で述べた通り, 1 人の SA は複数の学生から構成されるグループを担当することとなる. 学生をグループ分けするにあたって, 事前学習の状況及び出席状況により, グループを教室内の場所により分類することとした. 当初は, 講義ビデオを視聴し事前課題ワークシートに記入している学生からなるグループの場所をエリア A, 事前ビデオは視聴したが事前課題ワークシートに記入していない学生からなるグループの場所をエリア B, 事前学習をしていないあるいは欠席回数が多い学生からなるグループの場所をエリア C とする 3 つのエリアに分類した. しかし 2 回目までの授業で, 事前課題ワークシートが未記入の学生は, 授業内の活動を事前課題に対する個人活動から始める必要があり, C の学生と同じことをする必要があると分かったため, 反転授業が始まった 3 回目以降はエリア A とエリア C の 2 つのエリアに分けて活動を行うこととした. 1 人の SA はエリア A またはエリア C の 1 グループを担当することとした.

#### 5.3 プログラミング 1 の授業計画

プログラミング 1 の授業計画を表 2 に示す.

第 1 回, 第 2 回は講義ビデオの事前視聴は義務付けない講義形式で行い, 第 3 回以降は中間試験を除いて講義ビデオの事前視聴を義務付けた反転授業形式で行うこととした.

#### 5.4 実際の授業の進め方

SA を中心として進められる授業について, その標準的なタイムテーブルを表 3 に示す.

各回の授業は原則的に次のよう進められた.

##### 1. 授業開始前:

- 学生: 授業開始前に教員による事前課題ワークシートへの記入の有無のチェックを受ける. 記入済みの学生はエリア A に座り, 未記入の学生, 前回欠席したものはエリア C に座る.

\*1 ママ

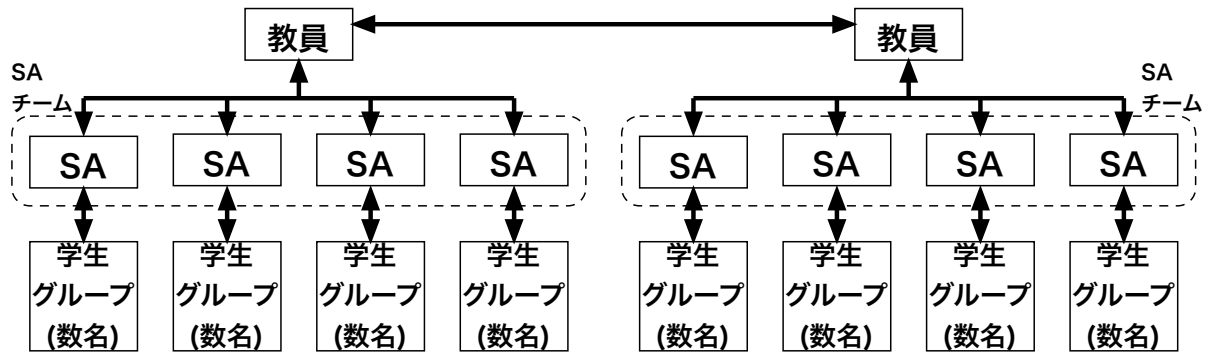


図 1 授業の指導体制

Fig. 1 The teaching system.

表 2 プログラミング 1 の授業計画

Table 2 The program of Programming 1.

| 講義回 | 内容  |
|-----|---|
| 1   | プログラムとは                                   |
| 2   | Processing のインストール<br>Processing 開発環境操作方法 |
| 3   | 簡単な描画プログラム                                |
| 4   | データ型, 演算子, 標準出力                           |
| 5   | 変数  |
| 6   | for 文による繰り返し処理                            |
| 7   | if 文, if-else 文による条件判断処理                  |
| 8   | 中間試験                                      |
| 9   | マウスの情報を使ったプログラム                           |
| 10  | 繰り返し処理と条件判断処理                             |
| 11  | 二重ループと switch 文                           |
| 12  | while 文による繰り返し処理                          |
| 13  | 関数  |
| 14  | 関数とアルゴリズム                                 |
| 15  | まとめと期末試験                                  |

表 3 標準的な授業のタイムテーブル

Table 3 The standard time table of a class.

| 活動内容            | 開始時刻  |       |
|-----------------|-------|-------|
|                 | エリア A | エリア C |
| イントロダクション       | 9:20  |       |
| 事前活動に関する個人活動    |       | 9:30  |
| 事前活動に関するグループ活動  | 9:30  |       |
| 授業内課題に関する個人活動   | 9:40  | 10:15 |
| 授業内活動に関するグループ活動 | 10:00 |       |
| 個人活動            | 10:15 | *     |
| 振り返り課題          | 10:30 |       |
| まとめ             | 10:40 |       |

\* グループ内の進行に応じて設ける.

- SA: 担当エリアの学生を座らせる.

## 2. イントロダクション

- 教員: 学習内容を述べ, 授業開始を宣言する.
- SA: SA は自身がグループ担当であることをグループ

の学生に告げ, グループの学生をグループ名簿シートに記入, 確認する.

## 3. 事前課題に関する活動

- エリア A: 各学生の事前課題の解答をもとにグループメンバとディスカッション, 答え合わせなどを行う.
- エリア C: 個別に事前課題に取り組む.
- SA: エリア A ではファシリテータとしてグループ学習を指導する. 事前課題グループ活動終了後, グループの学生の事前課題ワークシートを回収, スキャンし, 学生へ返却する. エリア C では, 学生ごとに事前課題への取り組みを指導する.

## 4. 授業内課題に関する活動

- エリア A: 各学生が個人活動として課題に取り組んだ後に, グループ活動として, 各学生の解答をもとにグループメンバとディスカッション, 答え合わせなどを行う. 授業内課題を済ませた学生は, オプション課題が出されていればそれに取り組む.
- エリア C: 各学生が個別に授業内課題に取り組む.
- SA: 事前課題に関する活動と同じ.

## 5. 振り返り課題に関する活動

- エリア A, C: 各学生が振り返り課題に取り組む.
- SA: 担当学生の活動内容を SA シートにまとめる.

## 6. まとめに関する活動

- 教員: その日の課題で間違い, 質問が多かった課題に関して簡単な解説を行う.
- SA: 担当グループの振り返り課題シートを回収, スキャンし, 学生へ返却する.

## 6. プログラミング 1 の実施状況と評価

### 6.1 実施状況概要

2016 年度のプログラミング 1 は, 2016 年 4 月 13 日から 2016 年 8 月 1 日までの期間で, 補講 1 回を含んで 15 回開

講された。履修者は情報電子工学科1年生57人、過年度生等が5人の計62人であった。これに対してSAは10人確保することができ、そのうち3年生が7人、2年生が3人であった。

第1回の“プログラミングとは”，第2回の“Processingのインストール，Processing開発環境操作方法”，第8回の中間試験，第15回の補講を除く11回について，講義ビデオの視聴を義務付けた反転授業として実施した。また，講義ビデオは補講を除く14回の授業に対して準備した。補講は，反転授業ではなく演習問題への取り組みを行った。

## 6.2 出席者数，欠席者数及びエリアCの学生数について

図2に，全履修学生数(62人)に対する各回ごとの出席者数，欠席者数及びエリアCの学生数の比率のグラフを示す。図2からわかるように，9回目より前は，欠席者数，エリアCの学生数は全体の10%程度であったが，9回目の中間試験以降は，欠席者数，エリアCの学生数は20%程度で中間試験前からほぼ倍増していることがわかる。

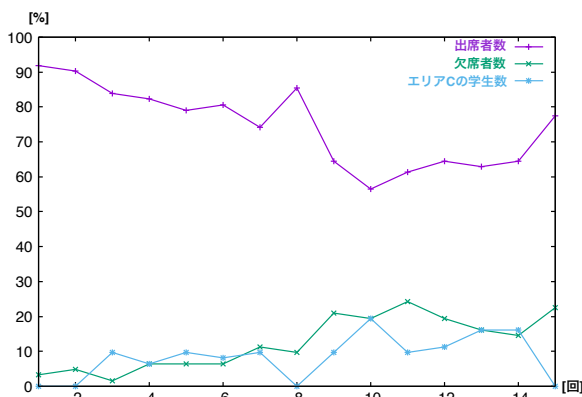


図2 各回毎の出席者数，欠席者数，エリアCの学生数

Fig. 2 The number of attendances, absentees and students in C area per each class.

## 6.3 理解度に対する影響

2015年度のプログラミング1では，2016年度のプログラミング1とほぼ同様な内容を，通常の講義形式でJava言語により扱った。2016年度で追加された内容は，switch文，continue文，break文及びマウスを使ったプログラムである。

それを踏まえて期末試験の平均点により学生の理解度を比較すると，2015年度が57点であるのに対して2016年度は71点で，14点の向上が見られる。扱っている内容，プログラミング言語，試験の内容が同一ではないために単純な比較はできないが，先に述べた通り扱った内容が増えていること，出題者は試験の難易度は大きく変化はないと考えていることから，学生の平均的な理解度は向上していると考えられる。

## 6.4 授業アンケート結果

第15回目の補講時に，LMSのアンケート機能を利用して記名式のアンケート調査を実施した。回答者数は38名であった。

事前学習，講義ビデオに関する質問へのアンケート結果を図3，図4に示す。

反転授業に関しては，80%以上が肯定的に評価をしている。これに対して，ビデオによる講義については肯定的な評価は61%にとどまった。反転授業及びビデオによる講義を肯定的に評価している理由としては，“予習ができるから”，“場所や時間を選ばずに繰り返し視聴できるから”，などが挙げられているのに対して，否定的に評価している理由としては，“講義ビデオの内容が不十分である”，“講義のほうかわからないところを聞けるから”，などが挙げられていた。

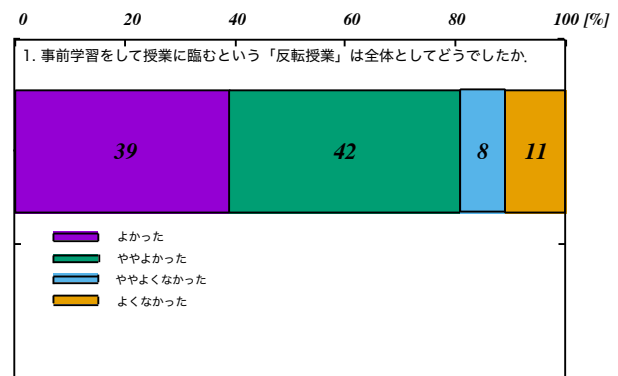


図3 反転授業に関する質問への回答

Fig. 3 Answers for a question on the flipped-classroom.

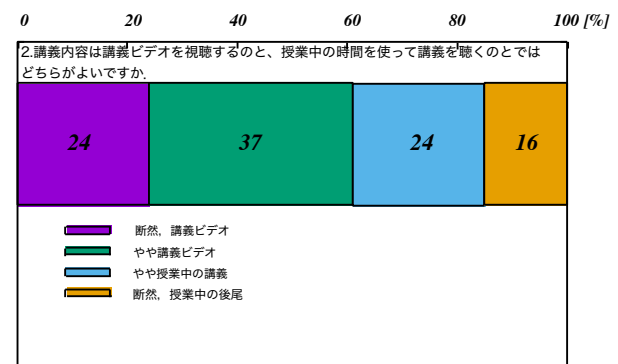


図4 講義ビデオの視聴に関する質問への回答

Fig. 4 Answers for a question on the pre-video-watching.

次に，SAによる指導体制に関する質問へのアンケート結果を図5，図6に示す。

1人のSAが担当するグループの人数については，60%以上の学生が多い，または，やや多いと回答している。また，SAによる説明については80%以上の学生が肯定的に評価している。

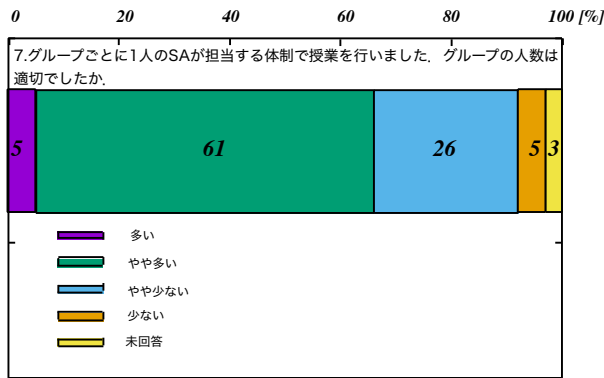


図 5 SA が担当するグループの人数に関する質問への回答

Fig. 5 Answers for a question on the number of students in the charge of each SA.

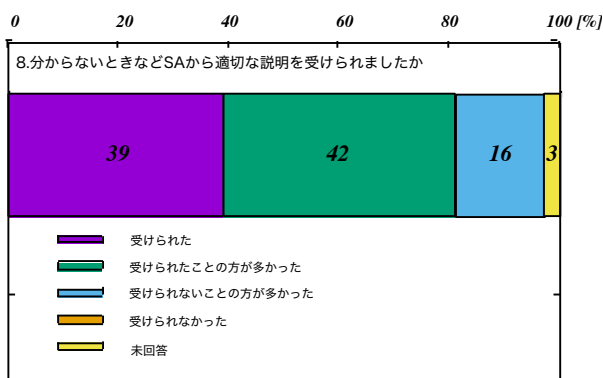


図 6 SA による説明に関する質問への回答

Fig. 6 Answers for a questions on the explanations by each SA.

プログラミング1の授業全体への感想、要望に関する自由筆記の質問に対しては、“課題の解説が欲しい”、“振り返り課題の時間が短い”、などが挙げられていた。

### 6.5 講義ビデオの視聴と成績

全講義ビデオの合計の長さは7時間6分36秒である。また1回の授業あたりの平均のビデオの長さは32分49秒である。これに対して、学生の講義ビデオの合計視聴時間の平均は5時間7分6秒であった。これより、学生が必ずしも講義ビデオを全て視聴しているわけではないことがわかる。

次に、1年生の学生の成績とビデオの視聴時間との相関を求めた。これを表4に示す。なお最終成績は、オプション課題を20%、中間試験を30%、期末試験を50%の比率で評価している。表4より、ビデオの視聴時間と成績の間に正の相関があることがわかる。

### 6.6 SAの活動について

プログラミング1では、SAが学生を直接指導する体制をとった。このような授業の進め方は初めてであったので、

表 4 ビデオ視聴時間と成績の相関係数

Table 4 Correlation coefficients between scores and video access.

|      | 中間試験正答率 | 期末試験正答率 | 最終成績 |
|------|---------|---------|------|
| 相関係数 | 0.30    | 0.42    | 0.54 |

当初は授業前にSAとのプレミーティングを実施し、授業の進め方等について検討したり、コメント等を集めた。授業がある程度スムーズに進むようになり、プレミーティングは6回を待たずして終了となった。その後は、授業開始後にSAによるコメントを集めることとした。

当初は、自身の指導に対する感想等が多かったが、回が進むにつれ、担当する学生の理解の様子、理解できていない点の指摘、授業の進め方、課題の構成に対する意見なども現れてきた。このようにSAが経験を積むことで、SA自身も成長しているように思われる。また、森田は[8]において、SAの志望理由として、“指導を通じて新たに学ぶ事もある”、“指導力を身につけたい”などが上位に位置していたと報告している。

このような観点からSAとしての活動を評価するためにSAに対してアンケートを実施した。実施対象はSA10人で、回答者は7人であった。アンケート項目は次の通りである。

- I. 4月頃の自分と今の自分を比較して、次の力が向上したと思うかどうかについて回答群の1.から5.の該当するものを1つ選んで下さい。
  1. コミュニケーションにおける話す力
  2. コミュニケーションにおける聞く力
  3. 積極的にコミュニケーションをとろうとする態度
  4. グループ活動のファシリテーションを行う力
  5. コミュニケーション力全般
  6. プログラミングの力
  - 回答群: 1. 向上した. 2. 少し向上した. 3. 変化なし. 4. 少し低下した. 5. 低下した.
- II. 自分が成長したと考えられる点、SAをやっている印象深かったこと、SAをやった感想を記述して下さい。
- III. 授業の進め方について改善する方がよいと思うことがあったら、記述して下さい。

このアンケートのI.の各項目に対する回答を図7に示す。全ての質問に対して、80%以上のSAが向上した、または、少し向上したと回答している。

自由筆記の質問に関しては、“相手に応じて説明するスキルが向上した”、“教えることで自分にとっても勉強になる”、“学生に積極的に声かけをすることでコミュニケーション能力が向上した”、“わかりやすく伝えるための工夫をした”、など具体的にどこが向上したかを述べている学

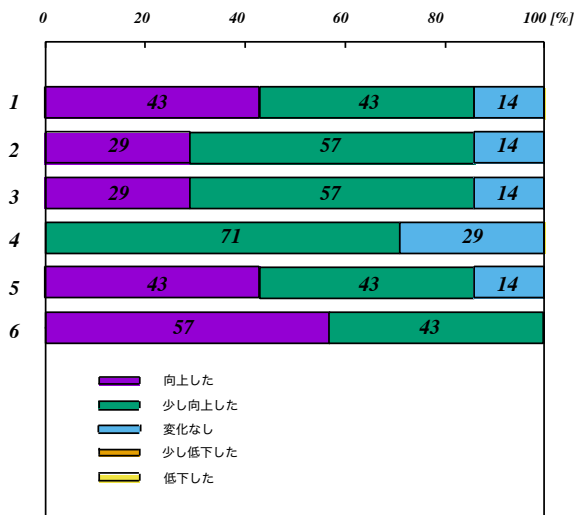


図 7 SA アンケート結果  
Fig. 7 Answers for questions to SAs.

生が多く見られた。

### 6.7 考察

学生の出欠及びエリアCの人数については、中間試験以降欠席者及びエリアCの人数が増大している。欠席者は今回はエリアCに属することとなり、エリアCの学生への指導は、実質的には個人指導のような形になるためSAの負担が増大するため、今後の検討が必要である。

成績から学生の理解度を評価すると、Java言語を用いてほぼ同じ内容を扱った2015年度のプログラミング1と比較して、期末試験の平均点に10点以上の上昇が見られる。これより学生の理解度は向上していると考えられる。

アンケートの回答によると我々が設計した反転授業は概ね肯定的に評価されている。ただし、1人のSAが担当する学生グループの人数については8人程度で実施したが、今後は6人程度にするのが望ましいと考えられる。

講義ビデオの視聴については、6.5節で示した通り、成績と正の相関が見られる。しかし、現状では学生が講義ビデオを全て見ているわけではない。成績の向上を目指すには、学生が講義ビデオを事前に全て見るようにさせる工夫が必要である。

また、授業アンケートの回答に、授業内課題について解答例を希望する学生が多かった。これに関しては、プログラミング1の後継科目であるプログラミング2では、希望する学生には授業時間内でも随時配布する予定である。

事前学習については事前課題ワークシートへの記入の有無の確認を行った。この場合、学生は自身が理解できている点は把握できるが、理解できていない点を把握することは難しいと考えられる。したがって、事前学習については、学生自身に何を理解して、何を理解していないかを認識させ、その上で理解できていない点を授業内に理解するよう

指導するべきであり、そのため工夫が必要である。

SAの活動については、SAとしての活動を補助するためのSAマニュアル、経験の共有などを目的としたSAシートを作成した。これらの活用については後継のプログラミング2などでさらなる検討が必要である。また、SA自身の成長を見るためにアンケート調査を実施した。成長したとの回答が多数を占めた。次年度以降も継続的に調査するべきである。

### 7. おわりに

本稿では、[10]で報告したプログラミング初学者を対象とする授業設計について、実際に授業を行った内容、結果について検討、評価した。

講義ビデオを事前に視聴し、SAによるグループ指導の下、ワークシート上に手書きで課題を作成し、ループリックを自己評価の指標として利用する反転授業として行われた授業に関しては、学生の理解度の向上が見られ、概ね肯定的な評価だった。後継科目であるプログラミング2では、プログラミング1での問題点を改善し、ほぼ同様な形式で授業を行う予定である。

今年度のプログラミング1及び2の実施結果を踏まえて、来年度以降の実施に向けて改善していきたい。

### 参考文献

- [1] 林 康弘, 深町 賢一, 小松川 浩: プログラミング教育における反転授業の実践と評価, 第40回教育システム情報学会全国大会, A2-2, pp.97-98 (2015).
- [2] 林 康弘, 深町 賢一, 小松川 浩: eラーニング利用による反転授業を取り入れたプログラミング教育の実践, 論文誌ICT活用教育方法研究, Vol.16, No.1, pp.19-23 (2013).
- [3] Processing: <http://processing.org>, (2016.09.05).
- [4] Reas C. and Fry B. 著, 船田 巧訳: Processingをはじめよう (Make: Projects), オライリー・ジャパン (2011).
- [5] 水谷晃三, 高井久美子: プログラミング初学者を対象にした動画教材による反転授業の実践と評価, 情報処理学会研究報告 Vol. 2015-CLE-17 No. 34, pp. 1-8(2015).
- [6] 渡辺博芳, 高井久美子: 「情報基礎」におけるビデオ講義を用いた反転授業の評価, 情報処理学会論文誌教育とコンピュータ Vol. 1, No. 4 pp. 64-74(2015).
- [7] 森田 彦: 学生教育補助員を活用した演習教育 - 「プログラミングの場合」 -, 札幌学院大学社会情報第14巻2号 pp. 151-166(2005).
- [8] 森田 彦: SAを活用した授業運営: プログラミング演習の場合, 札幌学院大学社会情報第18巻2号 pp. 117-129(2009).
- [9] 渡辺博芳, 水谷晃三, 盛 拓生, 荒井正之, 佐々木茂, 古川文人, 高井久美子: 大学のプログラミング教育のためのループリックの検討, 情報処理学会研究報告 Vol. 2016-CLE-19, No. 6, pp. 1-9(2016).
- [10] 盛 拓生, 渡辺博芳, 水谷晃三, 荒井正之, 佐々木茂, 古川文人, 高井久美子: 大学のプログラミング教育における反転授業とループリックによる自己評価の導入, 第41回教育システム情報学会全国大会講演論文集 pp. 419-420(2016).