

文理融合型学科のプログラミング科目における反転授業の実践

A practice of a flipped classroom for programming courses
in an interdisciplinary department

尾関 基行

OZEKI Motoyuki

武庫川女子大学 学校教育センター年報

第3号 2018年

文理融合型学科のプログラミング科目における反転授業の実践

A practice of a flipped classroom for programming courses in an interdisciplinary department

尾関基行*

OZEKI, Motoyuki*

キーワード 反転授業, プログラミング科目, 文理融合型学科

1. はじめに

アメリカの労働省によると、アメリカでは 2024 年までにソフトウェア開発およびプログラミングの求人数が 42 万 6900 人に増えると予測されている。日本国内でも 2015 年時点で 17 万人の IT 人材が不足しており、このままいけば 2030 年には 59 万人の IT 人材が不足すると懸念される。そのような社会的背景からプログラミング教育に注力する動きが世界的に活発化しており、イギリス・ハンガリー・ロシア・インド・フィンランドでは初等教育にてプログラミングを含んだ情報教育科目が必修化されている。日本でも、2017 年 3 月に公示された新学習指導要領の中で、初等・中等教育におけるプログラミング教育の強化が図られた。それにより、小学校ではプログラミング教育（プログラミングの思考を養う学習）が必修化され、中学校では技術家庭科においてプログラミングに触れる時間が倍増する。高等学校の新学習指導要領公示は 2018 年であるが、全生徒がプログラミングに触れる機会を得るよう科目「情報」が改変される。

このように初等・中等教育におけるプログラミング教育は国内でも強化されつつあるが、その実施は 2020 年以降であり、その影響が社会に及び始めるのは 10 年ほど先である。それまでの期間を繋ぐのは引き続き大学の役割であるが、その 10 年で IT 人材が大幅に不足していくと予測されているのである。人材が不足している理由は、国内にコンピュータサイエンスを学べる学科が少ないからであり、これを解決するためには、コンピュータサイエンスを主として学ぶ学科（以下、情報系学科）だけでなく、文理融合型の学科でもコンピュータサイエンスを積極的に教育していかなくてはならない。そのような背景もあつてか、オーソドックスな学部・学科構成の多い国立大学でも文理融合型の学部・学科が近年次々と設置されている。IT 人材といっても本格的なプログラミングを必要とする仕事は一部であり、文系の学問を修めつつ、それをコンピュータで処理・管理できる人材を文理融合型学科から輩出していけば、IT 人材不足の問題を軽減することができるだろう。

武庫川女子大学（以下、本学）の情報メディア学科はまさにそのような文理融合型学科であり、その中でも女子大学としては情報系科目を重視したカリキュラムを組んでいる。IT 人材の不足は世界的な問題であるが、日本では特に女性の IT 人材不足が深刻であり、システムエンジニアなどの職業において女性は完全に売り手市場である。共学の情報系学科には女性が少なく（国立大学では 1 割程度）、文理融合型学科でも IT 企業に就職する女子学生は少ない。そういったことから、女子大学の情報系学科や文理融合型学科⁽¹⁾には期待が寄せられており、本学の情報メディア学科にも毎年何社もの IT 企業が採用の相談に訪れている。

しかし、筆者が IT 企業の方へヒアリングしたところでは、システムエンジニアとして雇用した女

* 情報メディア学科講師

性社員には営業やマネジメントといった領域での活躍が主に期待されていることが、その会話のニュアンスから受け取れた。裏を返せば、女性社員にはシステム開発等の主力としてあまり期待されていないということになる。情報系学科を卒業した女性の中にはシステム開発の第一線で活躍している人ももちろんいるが、新卒女性の大きな割合を占める文理融合型学科を卒業した女性にはシステム開発（≒プログラミング）は厳しいと判断されているとすれば、IT人材を輩出しようとする大学として改善すべき点であろう。

本稿では、そうした女子大学の文理融合型学科におけるプログラミング教育について、筆者の担当する科目の実践例を紹介する。詳しくは2章で述べるが、本学のプログラミング教育の問題の一つに「1回の授業時間が短いこと」が挙げられる。筆者の授業では、この問題を解決するために反転授業を取り入れている。ここで紹介する事例は、本学科に限らず、文理融合型学科におけるプログラミング教育に共通したテーマであると考えられることから、本稿に実践報告としてまとめておく。

2. 文理融合型学科のプログラミング科目と反転授業

まず、比較のため、一般的な情報系学科における演習科目について触れておく。筆者は、本学に着任する前に国立大学（京都大学および京都工芸繊維大学）で助教として勤務しており、学生実験やプログラミング演習などの演習科目も担当していた。学生実験はいずれも週3コマ（4.5時間）以上の割当がされており、課題が終わらない場合には延長して居残ることが一般的である。小休憩を挟みながらではあるが、実験によっては1回に6時間以上かかるものもあった。京都工芸繊維大学のほうは情報工学科（正式には「情報工学課程」）であったため、学生実験とはいうものの、そのほとんどの時間をプログラミングに費やすプログラミング科目である。プログラミング演習については京都工芸繊維大学でのみ担当していたが、こちらも1回2コマ（3時間）が割り当てられており、課題が終わらない場合には1時間ほど居残る学生が多かった。筆者の出身大学である筑波大学でも、明確には記憶していないが、ほぼ同様の形態であった。

いずれの大学でも学生実験とプログラミング演習は必修科目であり、4年間、前後期とも実施される（4年次の後期はないこともある）。京都工芸繊維大学の場合、学生実験もプログラミング科目とみなせるので、（学生実験3コマ+プログラミング演習2コマ）×15週×前後期×4年間=600コマ分（900時間）のプログラミング教育を学修することになる。ここでは京都工芸繊維大学の情報工学科の具体例で概算したが、これは一般的な情報系学科のカリキュラムといえる。ここに居残り時間が加わるので、授業時間としてプログラミングに割り当てられる時間は1000時間を超える。さらに、課外時間に取り組む課題も出題され、座学科目でもプログラミングを課題とする授業があるので、その1.5~2倍程度の時間（1500~2000時間）が授業に関係するプログラミングの学修時間といえる。

一方、文理融合型学科には文系の科目もあるので、プログラミング教育に割り当てられるコマ数は半減する。文理融合といっても1対1で配分されているわけではなく、例えば本学情報メディア学科の場合、教員数の割合でいうと理系は1/3である。それでも女子大学の文理融合型学科としては理系の割合が多いほうである。授業時間としてプログラミング（ウェブやデータベースを含む）に割り当てられる時間を本学科で概算すると、4年間で約20科目×15週×1コマ=300コマ（450時間）程度となる。ただし、授業の一部でしかプログラミング等を使用しない科目も多くあり、また、これらは選択科目であることから20科目すべてを履修するわけではない。IT企業に就職していく理系寄りの学生でも平均で100コマ（150時間）程度の時間しかプログラミングに触れていないと筆者はみている。大雑把な概算ではあるが、文理融合型学科の学生は、情報系学科の学生の1/6程度の時間しかプ

プログラミングを学修していないことになる。

問題となるのは4年間の延べコマ数だけではない。プログラミング科目は、通常、1回の授業に2～3コマを割り当ててじっくりと課題に取り組む。講義を聞くだけではなく、自分で実際にプログラムを作成する過程でその技術を身につけていくのがプログラミングの学び方である。実際のところ、プログラミングを習得するのに必要とされる知識はそれほど多くはない。プログラミングの難しさは、その知識を実際の問題に適用していく部分にある。これは座学だけでは身につかず、いくつもの課題を試行錯誤しながら自分の力で問いていく他ない。その時間が少なくとも各回1コマ(1.5時間)は必要であることから、情報系学科のプログラミング科目は1回が2コマ以上の授業となっている。

しかし、文理融合型学科では、理系科目も文系科目と同等に扱われるため、プログラミング科目も原則1回1コマである。1コマでは、プログラミングの実質的な学びのフェーズに行き着く前に授業が終わってしまう。講義と課題を分離し、隔週で交互に実施していく方法も考えられなくはないが、新しく知った知識はできるだけ早く使用しないと学習効率が悪い。また、課題をレポートとして授業外に実施するケースもよくみられるが、プログラミングの学びの中心である課題の時間に学生をサポートする体制を用意しないのでは本末転倒である。

この問題を解決する優れた方策として「反転授業(反転学修)」がある。反転授業とは、講義を映像化して授業前に視聴させ、授業の時間はディスカッションや課題などのアクティブラーニングに充てる方法である。反転授業をプログラミング教育に取り入れれば、1コマすべてを小テスト・演習・課題などに充てることができる。また、次の授業までの間に必ず予習をすることになるので記憶が途切れにくく、プログラミングのような技術的に連続性のある内容を教える場合にも反転授業は向いている。実際、日本の大学における反転授業はプログラミングを含む情報系科目において先行しており、サルマン・カーン²⁾のTED Talksによって反転授業が話題となったすぐ後の2013年頃から北海道大学、千葉科学技術大学、東京工科大学などで実施され始めている。これは、そもそもコンピュータサイエンスやプログラミングの科目においてe-Learningが先行して普及しており、予習のための教材を準備する障壁が低かったことも理由であろう。

以上のことから、2016年度より、筆者も自身の担当するプログラミング科目に反転授業を導入した。プログラミング科目の反転授業については研究や事例がいくつか報告されているが、女子大学の文理融合型学科の試みとしては、おそらく他に報告はなされていないと思われる。

3. 実践事例

3.1 筆者の担当するプログラミング科目

筆者の担当する科目のうち、反転授業を実施しているのは以下の3科目である。それぞれ、主に対象とする学年と、反転授業を実施した年度を括弧内に記す。

- ・プログラミング入門(1年後期, 2016年度)
- ・ウェブコンピューティング(2年前期, 2016年度, 2017年度)
- ・ウェブアプリケーション論(3年前期, 2016年度, 2017年度)

『プログラミング入門』は、科目名のとおり、本学情報メディア学科のプログラミング科目の入門に位置づけられる。ProcessingというC言語の文法に似たプログラミング言語を用いてプログラミングの基礎を学ぶ。1年生の4～5割の学生が履修し、1年次に未履修だった2年生も合わせると100名を超える授業となる。学科の演習室の定員は80名程度であるため、学科のクラス別に二つに分けて開講している。1年生にとって初めての本格的なプログラミング系科目であり、プログラミング的

思考の苦手な学生も多く受講してくることから、特に中盤以降になると理解度の差が広がってくる。

『ウェブコンピューティング』は HTML と CSS を学ぶ、いわゆるウェブの授業である。HTML と CSS は厳密にはプログラミング言語ではないが、文理融合型学科ではプログラミング科目に位置づけられることが多い。この科目は『プログラミング入門』よりもさらに受講生の幅が広く、2年生の7～8割が受講する。文理融合型学科の「文」の部分に当たる学生が半数を超えるため、基本的な内容とはいえ、授業が進むにつれて理解度の差が広がる。本科目も学科のクラス別に二つに分けて開講している。

『ウェブアプリケーション論』は JavaScript の入門に位置づけられる科目であるが、文理融合型学科の学生にとっては比較的高度な内容となる。この科目の受講生はプログラミング系のゼミに所属する学生が中心となり、履修者数は3年生の3～5割である（2016年度は3割、2017年度は5割と大きく変化した）。

以上の3科目のうち、『プログラミング入門』と『ウェブコンピューティング』は2015年度から開講した科目であり、開講初年度（2015年度）は反転授業ではない通常の形式で授業を行っていた。『ウェブアプリケーション論』は2016年度からの開講であり、開講当初から反転授業で実施している。

3.2 事例の共通部分

筆者の実施している反転授業の流れを以下に示す。2016年度の『ウェブアプリケーション論』は少し異なるが、その他の授業ではこの流れに従っている。

課外時間

1. 講義映像の視聴（10～20分の映像×1～2本）
2. A4 ノートの作成（小テストに持込可）

授業時間

1. 小テスト（範囲：前回の課題＋今回の講義映像）
2. 小テストの解説
3. 一斉演習（講義映像の復習）
4. 課題（講義映像の内容を応用したもの）

武庫川女子大学での筆者の1年目（2015年度）の授業経験から、本学の学生は採点されるものに対して強い意欲を示すように感じたため、反転授業にも小テストを取り入れた。小テストは、毎回の授業のはじめに10～15分間で実施する。出題範囲は前回の課題と今回の講義映像であり、課外時間に作成してきたA4ノート（A4用紙1枚、裏も使用可）を持込可とする。このようにすることで、ほぼすべての学生たちが講義映像を事前に視聴してくることがわかった。A4ノートにまとめるために、講義映像を何度も繰り返し視聴する学生も少なくない。

一方、課題については基本的に採点対象としていない。2016年度は提出・未提出のみで採点したが、2017年度はそれも取りやめた。受講生数の関係で採点が難しいことも理由の一つだが、課題はあくまで「質問するためのきっかけ」であり、これを採点対象とすると「正解すること」に焦点が向いてしまう可能性があるためである。また、課題の時間は周りの学生たちと自由に相談して進めてよいことにした。結果、採点対象ではないからといって課題に取り組みないという学生はほとんどおらず、また課題と関係のないことで騒いだりすることもなく、ちょうどよい活気で溢れた時間となっている。

一斉演習とは、講義映像の一部を授業内で復習する時間であり、2017年度の反転授業ではすべての担当科目に取り入れている。プログラミングの学習は座学的な部分と技術的な部分に分けることがで

きるが、その技術的な部分のみを講義映像の内容から抜き出して、一斉演習で扱う。課題の時間と同様、一斉演習も周りの学生と相談しながら進めてもよいとしているが、騒がしくなりすぎることはない。この時間帯も積極的に質問をする時間になっており、特に授業についていけなくなった学生に対しては半ばマンツーマンで助手がサポートに付く。理解度が似通った学生が集団になっていることが多いので、集団のなかの一人にしっかり教えておけば、課題の時間にその学生が周りに教えてくれる。

授業のサポート体制であるが、『プログラミング入門』と『ウェブコンピューティング』には4名の助手がアシスタントについている。『ウェブアプリケーション論』は受講者数が少ないこともあり、2名の助手がついている。一斉演習と課題の間、助手は、学生の質問に速やかに対応すべく教室後部で待機している。また、雑題などをしている学生がいれば注意する。小テストの間はA4ノートを確認したり、不正行為がないか確認してまわる。受講者数の多い『ウェブコンピューティング』でも周りと自由に相談しながら進めるスタイルが成立するのは、4人という数の助手がいるからだと思われる。なお、1教室80名を超えてくると、4名の助手と1名の教員でも質問に対応しきれない時間帯が発生する。

3.3 事例紹介

本節では個別事例を紹介していくが、反転授業化による共通した効果については3.4節に、また反転授業の課題である講義動画の作成については4章にそれぞれ分けてまとめた。ここでは各科目を実施していくなかで明らかになった問題点とその改善の経過について述べる。

2016年度『ウェブアプリケーション論』

『ウェブアプリケーション論』は筆者が反転授業を実施するきっかけとなった授業である。武庫川女子大学に着任した初年度に『ウェブコンピューティング』と『プログラミング入門』を（通常授業として）実施するなかで1回の授業時間が短いという問題を痛感し、それらの応用科目を各回1コマで実施するのは非常に難しいと考えた。この問題を反転授業で解決しようと決め、まずはインターネットのプログラミング学習サイトの動画を参考にして講義動画を作成した。本稿で紹介する3科目では、本科目のみ講義映像が実演形式となっている。実演形式とは、実際にプログラミングを入力・実行していく様子を見せながら解説する動画である（図1左）。

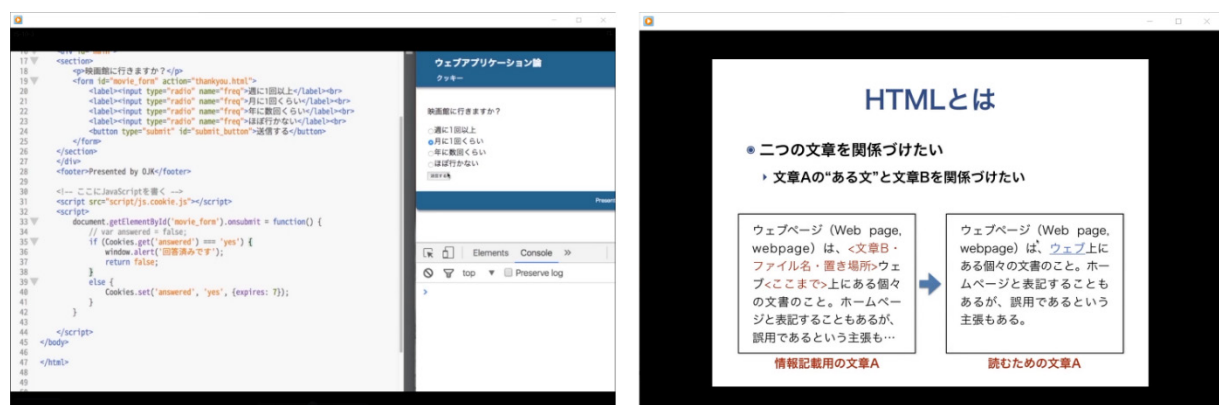


図1 講義映像（左：ウェブアプリケーション論，右：ウェブコンピューティング）

本科目では、課題にできるだけ多くの時間を割くために、授業時間中に一斉演習による講義映像の復習をしなかった。小テストとその解説の後、すぐに課題を提示して取り組んでもらい、わからない

点はそのなかで質問してもらおうよう促した。しかし、前期の後半に実施される授業アンケートにて、「講義映像の内容を授業中にも前で説明してほしい」という意見が複数みられた。これは、本科目の内容が文理融合型学科の学生には難しく、講義映像だけでは課題を解き進められるほどに理解が進まなかったことが原因と考えられる。また、こちらの想定では、講義映像をただ視聴するのではなく、自分でもプログラミングして試してくることを想定していたが、ほとんどの学生は講義映像のなかの知識的な部分を A4 ノートに書き出してくるのみであったことも一因であろう。これは文系学生の典型であると考えられ、ノートを作るよりも実際に手を動かすことを好む理系（情報系）学生とは異なる部分であった。自分自身が典型的な理系であり、また、これまで理系の大学にのみ勤務してきたことから判断を誤った。

2016 年度『ウェブコンピューティング』

『ウェブコンピューティング』は前述の『ウェブアプリケーション論』と同じ 2016 年前期から反転授業を開始したものであるが、『ウェブアプリケーション論』ほど反転授業を実施するという意識はなく、単に講義の一部を映像化して授業本体から切り離す感覚で開始した。2015 年度から開始した『ウェブコンピューティング』では、「ウェブデザイン技能検定 3 級」の内容に沿った座学と、HTML と CSS の基本を習得してウェブサイトを構築するための技術に分けて授業を構成していた。『ウェブコンピューティング』の講義映像はこの座学部分のスライドの解説を映像化したものである（図 1 右）。結果として、本学科の学生たちには『ウェブアプリケーション論』の実演形式の講義映像よりも評判がよかった。この科目の履修生は（筆者の担当科目のなかでは）特に文系寄りの学生が多く、A4 ノートにまとめやすいスライド形式の講義映像は分かりやすかったようである。

座学を授業本体から切り離したため 15～20 分ほど時間の余裕が生まれ、課題の時間が 30～35 分ほど取れるようになった。しかし、同時期に開講していた『ウェブアプリケーション論』の 1 時間以上ある課題時間と比較すると、やはり課題の時間が短く忙しいように感じた。また、授業中の講義（技術的な内容）は小テストに出ないのであまりよく聞いておらず、課題の時間になっても何もできない学生が散見された。以上の反省点から、2017 年度は座学と技術の講義を両方映像化して課外時間に移し、授業冒頭の小テストでその理解度を測るという方法をとることにした。

2016 年度『プログラミング入門』

以上、2016 年度前期の 2 科目について問題点と反省点を挙げてきたが、反転授業は概ね好評であり、課題の時間も拡大できるため、後期の『プログラミング入門』も反転授業へと切り替えた。本格的なプログラミング言語の科目なので実演形式の講義映像にするか迷ったが、1 年生の授業であり、受講生がプログラミング的思考を得意とする学生ばかりではない可能性を考慮してスライド形式の講義映像とした。技術的な部分については授業中の一斉演習のなかで復習した。一斉演習では、筆者が前で打ち込んだプログラムを学生たちに入力して実行してもらったり、一部完成していないプログラムコードを学生たちに自力で完成させてもらったりといった教示方法をとった。

2016 年度前期で明らかになった問題点への改善案を取り入れ、筆者の担当科目における反転授業の大枠の形は本科目ではほぼ固まった。授業アンケートでも反転授業の形式（小テストも含む）自体に対するネガティブなコメントはほとんどなかった。ただ、講義映像のリップノイズ（唇を開くときに鳴る音）が酷いので講義映像を視聴したくないという意見が出された。その意見は 1 件だけであったが、不快に感じる学生は潜在的にはさらにいると判断し、リップノイズの酷い部分はカットしたり、収録

中に気づいた場合は撮り直したりといった対処をとった。なお、この問題が前期に露呈しなかったのは、前期はスマートフォン (iPhone6) やノート PC (MacBook Pro 2013) のマイクを使って音声を収録しており、そもそもの音質が悪かったためである。音量が小さい、言葉が聞き取りにくいという意見を受けて、後期から高性能なマイクに切り替えていたことから別の問題が発生した。

2017年度『ウェブアプリケーション論』

反転授業2年目となった本科目では、前年度収録した講義映像を再利用することができた。ただし、リップノイズを始めとした(昨年度前期の時点では意識していなかった)問題点を修正するため、言い淀み(「え～」や「あの～」といった部分)を手作業で削除し、音量とノイズの調整を専門知識を持った助手に依頼して実施した。また、小テストの後にすぐ課題に入るのではなく、講義映像の内容を一斉演習で復習するようにした。

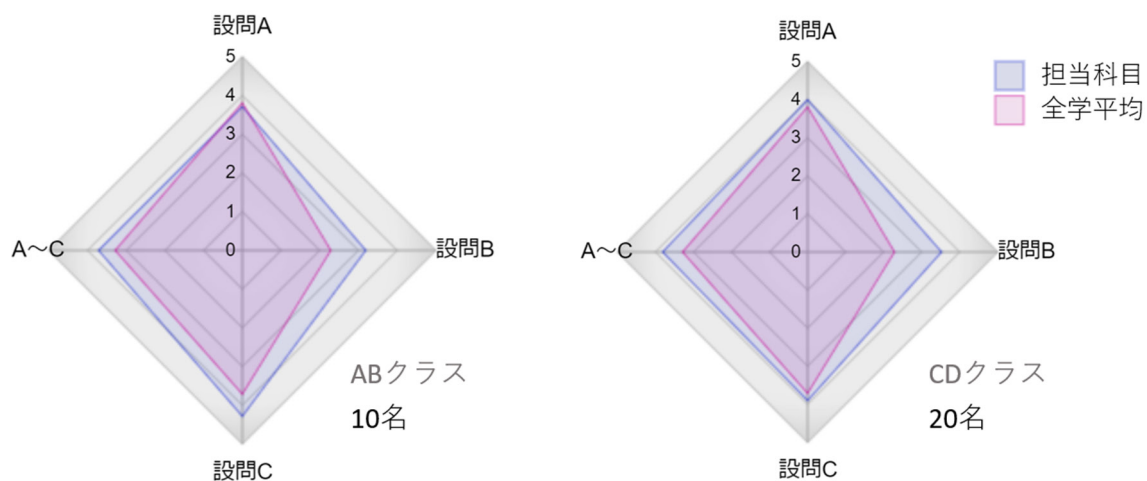
2017年度『ウェブコンピューティング』

本科目でも前年度収録した講義映像を、前述の『ウェブアプリケーション論』と同様の処理を加えて再利用した。また、2016年度の実践報告でも述べたように、技術的な内容についても映像化して、すべての講義を課外時間に移動した。他の担当科目と同様、講義映像の内容は授業内に一斉演習で復習するが、それでも10～15分ほどの時間短縮ができ、課題の時間に40～45分ほど割り当てることができた。これでも十分とはいえないが、プログラミング系の演習科目として最低限の時間は確保できたと考えている。

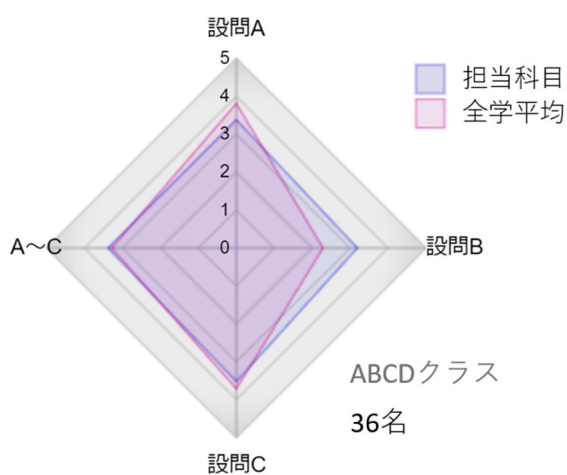
しかし一方で、課外時間にかかる時間(講義映像をA4ノートにまとめる時間)が長くなってしまい、学生によっては1.5時間を越えて2時間近くかかってしまう場合があるとの報告を受けた。大学設置基準では1単位45時間とされており、授業時間が1.5時間の場合、課外学修時間は1.5時間となる。この時間超過の問題については本科目で扱う範囲を削減するしかないが、本科目は全国大学実務教育協会の「ウェブデザイン実務士」の認定科目になっている関係で、すぐには内容を削減することができない。この点については今後のカリキュラム改定なども見据えながら改善していきたい。

なお、前年度(2016年度)の本科目および『プログラミング入門』では、講義映像と併せて、映像内で解説しているスライドのPDFも事前に配布していた。しかし、講義映像を視聴せず、配布されたPDFだけをA4ノートにまとめてくる学生がいるという報告を受けたため、2017年度からPDFの配布を取りやめた。授業アンケートではスライドのPDFを配布してほしいという意見も出ているが、2017年度後期の『プログラミング入門』でも配布しない方針で現在進めている。

ウェブコンピューティング（2016年度前期）



ウェブアプリケーション論（2016年度前期）



設問A

あなたはこの授業を受けてシラバス上で到達目標としていた知識や能力が身についたと思いますか？

設問B

この授業の予習・復習や自己学習に1週間当たり平均してどのくらい勉強しましたか？

設問C

質疑応答、討論、課題の発表など、授業への参加を促すための工夫がなされていましたか？

プログラミング入門（2016年度後期）

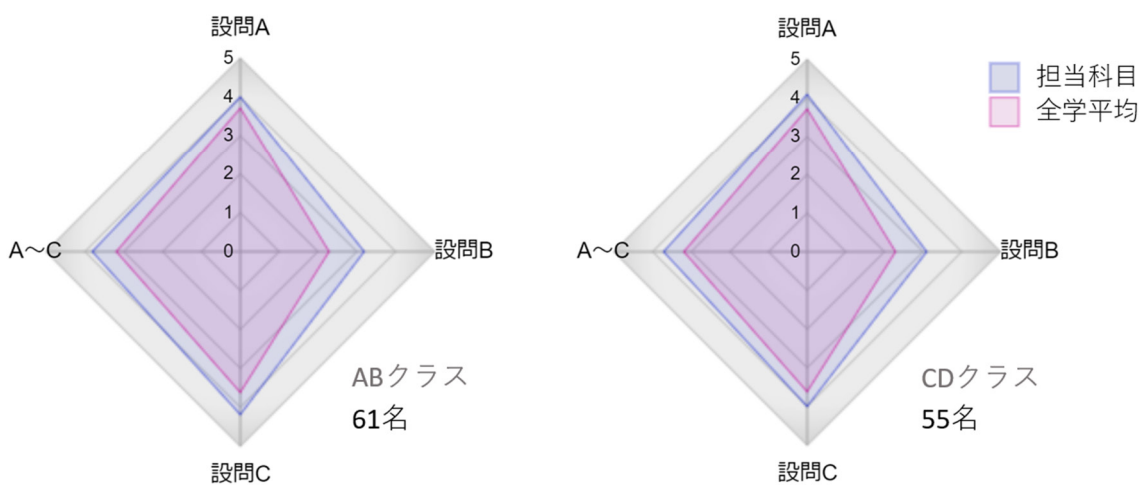
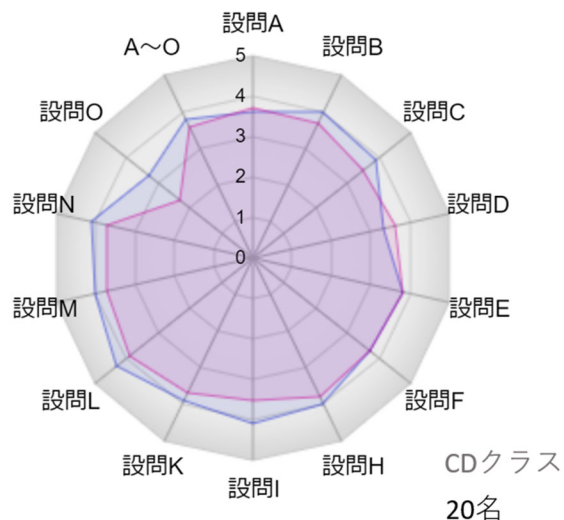
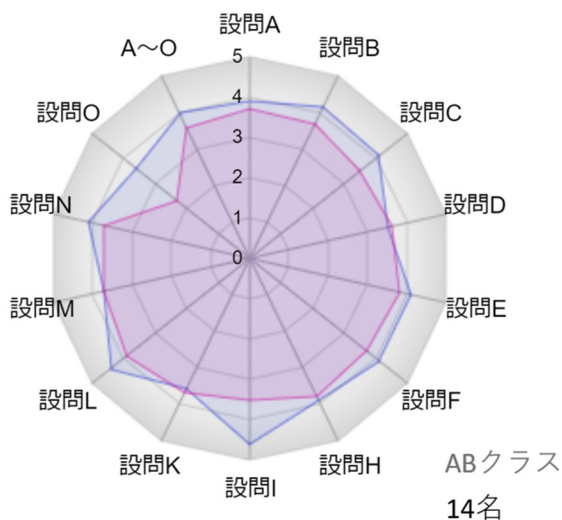


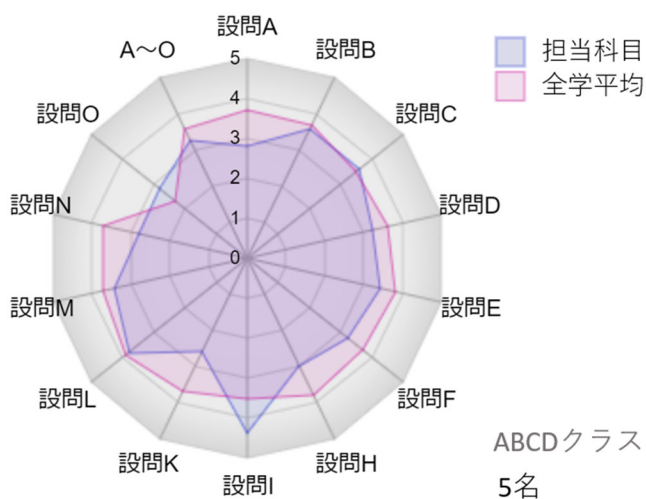
図2 2016年度の授業アンケート結果

ウェブコンピューティング（2017年度前期）

■ 担当科目 ■ 全学平均



ウェブアプリケーション論（2017年度前期）



- 設問I
授業時間以外にも学修する工夫をしていましたか？
- 設問O
予習・復習等に平均して週あたりどのくらい勉強しましたか？
- 設問D
授業の邪魔をする私語・入退室への注意等ができていましたか？
- 設問K
授業の内容をよく理解できましたか？

プログラミング入門
・2015年度（通常授業）
・2016年度（反転授業）
の比較

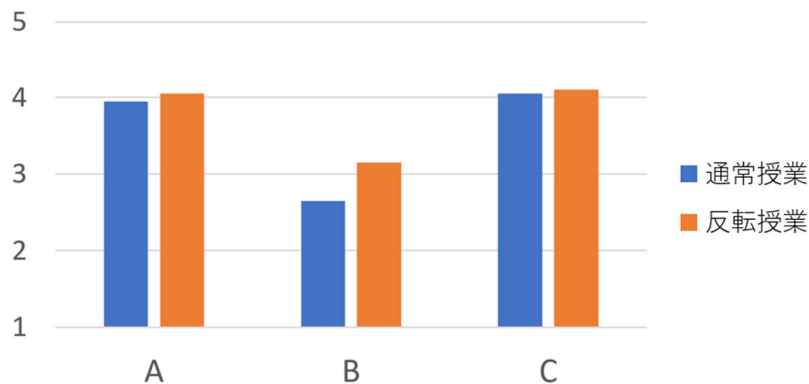


図3 2017年度の授業アンケート結果（上）とプログラミング入門での比較（下）

3.4. 反転授業の効果

武庫川女子大学の用意している授業アンケートの結果を図2と図3上に示す(いずれも本稿末尾に掲載)。2016年度と2017年度でアンケート項目が変わっているが、2016年度の項目B「この授業の予習・復習や自己学習に1週間当たり平均してどのくらい勉強しましたか?」と2017年度の項目I「授業時間以外にも学修する工夫をしていましたか?」およびO「予習・復習等に平均して週あたりどのくらい勉強しましたか?」が結果が大学平均を上回っていることから、反転授業の効果を学生も実感していることがわかる。なお、2017年度の『ウェブアプリケーション論』は、項目I・O以外は全学平均を大幅に下回る結果となっているが、回答者数が5名と少なく、そのうちの1名がほとんどの項目で評価1を入力していたことも一因となっている。その他の結果は概ね全学平均を上回っており、その難易度から評価が低くなりがちなプログラミング科目としては良い評価を得ている。

反転授業の効果を直接比較できるデータとして、反転授業ではなかった2015年度の『プログラミング入門』と、それが反転授業となった2016年度の比較を図3下に示す。結果から、項目B「この授業の予習・復習や自己学習に1週間当たり平均してどのくらい勉強しましたか?」がやや伸びていることが読み取れる。それほど大きな差となっていないのは、2015年度の『プログラミング入門』でも授業冒頭の小テストを実施しており、前回の授業の内容をA4ノートにまとめてくるよう指示していたからである。

なお、『ウェブプログラミング』も2015年度と2016年度で反転授業の有無を切り替えたが、2015年度は最終課題としてウェブサイト制作を指示し、2016年度はそれを実技テストで置き換えたため、直接の比較ができない。ウェブサイト制作を課題にすると課外時間に相当量の作業が必要となるため、項目Bでの差がつかなかった。

4. 講義映像の作成とその課題

反転授業の最大の難点は、講義映像を用意することである。プログラミングの講義映像はJMOOCやドットインストールなどのウェブサービスで無料で提供されるものもあり、それらを利用すれば手軽に実施できる。しかし、筆者が調べた範囲では、動画配信のタイミングが決められていたり、動画の内容や本数に過不足があったりなどの問題があった。大学の15回の講義で使用でき、かつ、文理融合型学科の学生にも理解しやすいコンテンツは、少なくとも2016年3月の時点では提供されていなかった。なお、JMOOCは東京工科大学と連携して反転授業を提供した実績があるが、一教員が望んで実現できるものではない。ゼミ活動など少人数の学生を対象にした場であれば、こういったサービスを活用しつつ、学生を細かくサポートすることも可能である。しかし、何十人という学生が受講する講義科目となると、自前で講義映像を作成したほうが最終的には手間がかからないと判断した。

結論からいうと、講義の内容が3年以上変わらないのであれば、講義映像を自前で用意しても、トータルで見ればコストパフォーマンスに見合う。講義映像を作成する年度は大きなコストがかかるが、以降に述べるノウハウを押さえておけば、2年目からは同じ講義映像が再利用できる。講義映像の質にもよるが、筆者の場合、15分の動画を収録・編集するのに3~4時間かかった。単純計算で講義そのものの時間の12~16倍の作成時間がかかるので、その後2年間の再利用程度ではコストが大きすぎるように感じられるかもしれないが、授業中に講義をしなくてもよいというのは想像するよりも負担が軽減する。プログラミングは変化の大きな分野なので、同じ講義映像は長く使ってもカリキュラムの1クール(4年)が限界であるが、各科目の講義映像の更新年度をずらしておけば、通常科目と同程度のコスト感覚で済むであろう。

講義映像の収録に必要な機材は、映像中に講師が登場しなくてもよいのであれば、PC とマイクだけでよい。講師の登場する講義映像もよくみられるが、カメラが必要になる上、カット編集も難しくなる。マウスポインタを拡大しておけば、ポインティングされている位置は PC 画面のキャプチャ動画だけでも十分に伝わる。筆者は Mac を用いて収録したが、Mac の場合、標準でインストールされている QuickTime Player でデスクトップを動画でキャプチャすることができる。Windows10 でもウィンドウキャプチャであればゲームバーの機能でデスクトップや特定ウィンドウの動画収録が可能である。ただ、ゲームバーは操作がやや独特で理解しにくいいため、画面キャプチャ機能付きの映像編集ソフトウェアを購入するのがよいだろう。

収録には、機材の他、雑音の入らない場所も必要である。多くの教員にとって、これが一番のネックになるかもしれない。教員の居室で収録すると校内放送が入ったり学生の声が入ったりするため、著者の場合は長期休暇中に収録したり、本学マルチメディア館の地下スタジオ（防音室）を使用して収録した。

マイクは良いものを使えば音質は向上するが、同時に細かい音も拾うため、リップノイズ（唇が開くときの音）やブレスノイズ（呼吸音）に気をつけなくてはならない。筆者の場合、2016 年度前期はスマートフォンや PC の内蔵マイクを、後期はヘッドウォーンマイク（頭部装着型マイク）を使用した。2017 年度前期には卓上マイクへと切り替え、場所をスタジオの防音室に移してポップガード（マイクの手前に設置する網のようなもの）を付けた。このように、音質の向上と各種ノイズの混入を軽減するために環境を変化させていったが、外付けマイクを使うことによる音質の向上以外は大きく改善はされなかった。

収録した映像は映像編集ソフトウェアで加工する。2016 年度前期は Mac に標準でインストールされている iMovie を使い、後期の途中から Adobe Premiere Elements を使用した。Premiere Elements に切り替えたのは、オーディオフィルタによってリップノイズやブレスノイズを一括して軽減できることを期待したためであり、あまり効果はなかった。結局、ブレスノイズはそれが含まれる範囲を手動でカットし、リップノイズはその部分を選択的に加工することで対処した。ブレスノイズを一つ一つ削除するのも大変ではあるが、発話の前後（音声がない部分）に入ることが多いので、カット自体は容易である。一方、リップノイズは発話のなかに紛れ込むことも多いので、どうしても処理できない部分が出てくる。2017 年度からは専門的な知識を持つ助手に依頼して、それまで収録した講義映像の音声をすべて調整してもらうことができた。

リップノイズとブレスノイズについては、収録時に注意しておくことが一番大切である。ブレスノイズは口呼吸をできるだけ避けるとともに、定期的に発話と発話の間で意識的に大きく息継ぎをして間を置き、あとからカットしやすいようにする。リップノイズが入らないようにするのは非常に難しいが、唾液の粘性が高まると音が出やすいので、空腹を避け（収録前に何か食べる）、小まめに水分を取りながら収録する。また、収録中にリップノイズが入ってしまったことに気づいたときは、同じセリフを言い直しておく。また、言い間違えたり言い淀んだりした場所をあとからカットするのも手間がかかるので、できるだけ減らす工夫が必要である。スライド形式の場合には、発表ノートに予め話す内容を入力しておき、それを読み上げるのが最終的には時間の短縮化になる。15 分の講義に対して、収録だけで 1 時間以上かかることもあるが、大きくミスした箇所を記憶しているうちに映像編集ソフトウェアで不要な部分を一通りカットする作業までしておいたほうがよい。ブレスノイズや言い淀みのカットは数日後でも効率を落とすことなく編集できる。

以上のように講義映像の収録・編集には非常に手間がかかるが、一度作成しておけば反転授業に限

らず利用できる。例えば、同じ授業をクラス別に複数回実施している場合、講義映像を流して見せることで伝達情報を等しくできる。また、自分の授業を履修していない学生がゼミに配属してきたときにも、講義映像を見ておくよう指示すればよい。

5. おわりに

本稿では、反転授業を取り入れている筆者のプログラミング科目（ウェブ含む）の実践例を報告した。これは、文理融合型学科のプログラミング科目が1回に1コマしかない問題への対策として開始したものである。プログラミング科目は、その性質上、知識を応用して実際に課題に取り組んでいるときに最大の学びがあるため、課題の時間のサポートを最も手厚くする必要がある。1コマの授業では課題に割く時間が短くなるのが問題であったが、反転授業にすることで1コマでも課題の時間を大きくとることができる。実際、1年半の実践経験を通して、文理融合型学科のプログラミング科目に対する反転授業の利点は非常に大きいと実感している。

反転授業の最大の問題は講義映像を用意することである。自前で用意しようとする、機材と環境が必要となり、また映像の収録・編集に通常の講義の10倍以上の時間がかかってしまう。筆者の場合、1回の講義を15～20分と短くしているため、現実的な時間で講義映像を作成することができた。しかし、文系科目のように1コマの講義をそのまま動画にする場合には、軽度の言い淀みや言い直しを含むことを許容するといった省力化が必要である。ただ、女子大学では特に問題になってしまうのがリップノイズとブレスノイズであり、これらを取り除くコストは非常に大きい。インターネットから無料で利用できる日本語の講義映像コンテンツの充実が待たれる。

なお、本稿では文理融合型学科のプログラミング科目のコマ数のみに着目したが、プログラミング教育において、情報系学科と最も差が出るのは研究室活動（ゼミ活動）である。情報系学科のほとんどは各教員（もしくは各講座）で独立した研究室を持っている。研究室に配属された学生には個人のデスクとPCが与えられ、平日・週末関係なく、24時間好きなときに研究することができる。研究室でプログラミングに触れる時間は、授業のそれとは比較にならない。一方、文理融合型学科では、研究室ではなくゼミの形態で研究活動を行っているところがほとんどである。本学情報メディア学科も「〇〇研究室」という呼称を使用しているが、実質的には文系ゼミの形態であり、通常科目と同じくゼミにも開講時間が定められている。学生部屋も複数のゼミで共有しており、デスクやPCも共有であるため、学生がいつでも研究活動できるような環境にない。他大学の文理融合型学科との差別化を図り、抜きん出た存在感を示すには、反転授業だけではなく、研究活動についても情報系学科に準じる環境を揃えることが必要であろう。

注

- (1) 筆者の調べたかぎりでは、国内の女子大学には情報系学部（情報学部や情報工学部など）はなく、お茶の水女子大学と津田塾大学に情報科学科が存在する以外はすべて文理融合型の学科である。
- (2) 世界中の子どもたちの教育のために無料の講義動画をインターネットから利用できるようにするNPO『カーンアカデミー』の設立者