

# オンライン実習における ペアプログラミングの実践

大塚亜未<sup>1</sup> 来住伸子<sup>1</sup>

**概要:** 初年次プログラミング科目の実習を2020年度はオンラインで実施した。前半は、10人程度の学生と1名のティーチング・アシスタント(TA)から成るグループで実習した。後半は、ペアプログラミングを導入し、学生の意欲および学生間のコミュニケーション能力の向上を目指した。後半の途中では、前半の成績と「課題の取り組みタイプ」を配慮し、ペアの組み合わせ変更を複数回行った。ペアプログラミングについての学生のアンケート回答、テスト成績とペア課題の関係などによって、オンライン実習状況におけるペアプログラミングの有効性について検討した。検討の結果、各種オンラインツールの利用とペアの組み合わせなどの配慮により、教室における対面ペアプログラミングより有利な点が観察できた。

**キーワード:** プログラミング教育, オンライン教育, ペアプログラミング

## Practice of pair programming in online workshop for the first-year computer science college students

Ami Otsuka<sup>1</sup> Nobuko Kishi<sup>1</sup>

**Abstract:** We gave an online programming course for the first-year computer science students. It consists of one lecture and one workshop for 27 weeks. The workshop was given in a small group with TA. In the last 9 weeks of the workshop, the students are asked to do the assignments both in solo and in pair-programming. We report on the effects of pair-programming on the students' grade and what considerations are needed to ask the students to do in pair programming at entry-level programming course.

**Keywords:** Programming education, Online education, Pair programming

### 1. はじめに

報告者は、情報科学科1年対象の実習付きプログラミング入門科目を少なくとも5年程度、担当している。2020年度は対面授業ができなかったため、講義も実習もすべてオンラインで実施した。この報告では、オンラインで実施した実習の後半部分で導入したペアプログラミングについて配慮した点とその効果について報告する。

本稿の第2章では、関連研究としてペアプログラミングに関する研究を紹介する。第3章では、該当するプログラミング入門科目の全体像を紹介し、ペアプログラミングを導入した経緯を説明する。第4章では、ペアプログラミングを実施した環境や体制について説明する。第5章ではペアプログラミングの実施によって得られたデータを紹介し、解析結果を考察する。第5章はまとめとして、プログラミング教育におけるペアプログラミングの役割について述べ、

<sup>1</sup> 津田塾大学  
Tsuda University, Tokyo, 187-8577, Japan

今後の改善方法についても考察する。

## 2. 関連研究

ペアプログラミングとは、XPの実践技術の一つとして提案された。当時は二人のプログラマーが一台のコンピュータを使ってプログラム開発を行うこととされ、コンピュータに入力する人をドライバ、指示を与える人をナビゲータと呼ぶ。現在、ソフトウェア運用保守のし易さ、知識や経験の組織内共有というような観点から、実際の開発現場で使用されている[1][2]。

一方、教育的な観点から、アメリカでは入門レベル(CS1レベル)のプログラミング教育で使用されている。ペアプログラミングの使用により、大学でコンピュータサイエンスを専攻する学生が増加した、とくに、女性やマイノリティの間で増加したという報告がある[3]。「プログラミングは一人で行う孤独な作業」「社会やビジネスに貢献できるか分かりにくい作業」というプログラミングについての否定的なイメージを変えることに役立つと考えられる。

しかし、XPと同様、ペアプログラミングは参加者や実施環境に合わせた配慮が必要とされている。たとえば、一台のコンピュータではなく複数のコンピュータを利用し、並行して作業する Distributed Pair Programming[4]、一人で行うプログラミング課題と組み合わせる課題を出す Hybrid Pair Programming[5]、ペアの相性についての調査 [6]などがある。

## 3. ペアプログラミングを導入した経緯

今回の事例の背景として、オンライン化する前の、報告者が行ったプログラミング教育について説明し、オンライン化の一環としてペアプログラミングを導入した経緯について説明する。

対象科目は、プログラミング I という情報科学科 1 年生(以下、受講者)の必修科目である。毎週 1 回講義 1 コマ、演習 1 コマの形で実施し、9 週間で 1 ターム 2 単位を履修する科目である。第 1 ターム、第 3 ターム、第 4 タームの 3 タームで構成し、それぞれプログラミング I(1)、プログラミング I(3)、プログラミング I(4) という異なる科目として成績評価を行っている。3 科目に分かれているが、2 7 週かけて Java によるプログラミングの基礎を学ぶ。2019 年度、2020 年度ともに同じ教科書を使い、ほぼ同じ内容を教えている。

2020 年度はオンライン実施になったため、講義部分は 2019 年度と似た内容を Zoom で提供し、演習部分は、形式を大幅に変更した。2019 年度の実習は、PC 教室を使用し、受講者は、Java の処理系が利用できる環境をすぐに利用できた。2020 年度は、実習は本人の PC 上で行うことになったが学生の PC には Java の処理系がインストールされていな

い。そこで、第 1 タームはクラウド上のプログラム実行環境を利用した。最初の 2 回は、Scratch 3.0 でコンピュータの基本操作に慣れてから、プログラミング学習プラットフォーム track [7]上で Java のプログラムを書いてもらい、採点も track で行った。第 3 タームになってから、自分の PC に Java の処理系をインストールする方法を説明し、自分の PC 上で Java のプログラムを書けるようにした。2019 年度の課題とよく似た課題を出題できるようになった。

第 1 ターム、第 3 タームをオンラインで実施した結果、次のような課題点が見つかった。

- 課題点 1 : 受講者のプログラミング能力の差。

第 1 タームの採点結果から、教えた範囲の知識を十分に理解している学生と、ほとんど理解していない学生に、2019 年度よりも大きく分かれることが観察できた。

- 課題点 2 : 受講者のプログラムに関する表現力の不足。

演習時間は受講者を 10 人程度のグループに分け、TA を付けた。受講者は TA に自由に質問できると説明し、昼休みの時間帯にも TA に質問に答えてもらうようにした。その TA から、受講者に答えるのに時間がかかりすぎる、話が終わらないというような不満がでてくるようになった。受講者が分からない点をはっきり説明できない、または TA が話していることが分からないためと考えられた。

そこで、関係者と相談し、第 4 タームからペアプログラミングを導入することにした。TA が学部生だったころ、数年前まではペアプログラミングを実施していたが、運用上の問題から実施しなくなっていた。2020 年度は、次のような配慮をしたうえで、ペアプログラミングを導入することにした。

### 1. 事前の説明と注意

ペアプログラミングとは何かという説明を事前に加えることに加え、ペアプログラミングのアンチパターンを紹介し、だれもがしやすい間違いを避けるように、ビデオを紹介しながら、注意を促した[8][9]。

### 2. ハイブリッド型

すべての課題をペアプログラミングで取り組むのではなく、ペアで取り組む課題と個人で取り組む課題が半分ずつになるようにした。

### 3. オブジェクト指向の利用

第 4 タームでは Java でオブジェクト指向を教えている段階でもあったので、課題はできるだけクラス単位に分けて出題し、ペアで各クラスを分担して作成するように指示した。以前のペアプログラミングでは、一定時間(20分)たったなら、ドライバとナビゲータを交代するように、と指示していたため、交代のタイミングが守りにくかった。

### 5. アンケートの実施

実習の時間ごとに、Moodle を使用して、ペアプログラミングに関するアンケートを実施した。このアンケートについては、次章で説明する。

上記の配慮をしたうえで、第4タームにペアプログラミングを実施したところ、TAからの不満は減り、2019年度と同じ程度の成績を達成できた。

試験の形式は、2020年度はオンライン形式としたため、2019年度との比較は全ての問題に対してはできない。が、実技試験の課題「決められた時間内に指定された機能のJavaのプログラムを作成しなさい」は、ほぼ同じ内容にしたので、おおよその比較はできると考えている。

## 4. オンライン実習の実施方法

### 4.1 オンライン環境の利用

前述したように、講義、演習とも、Zoomを用いたオンライン形式での授業を実施した。演習では10人程度の学生と1名のティーチング・アシスタント(TA)から成るグループ環境で、Breakout roomで課題に取り組んだ。学生は、課題についてTAへ質問したいときに、画面共有機能を用いて自分のパソコン画面を共有し、TAは、学生から共有された画面上のソースコードや出力結果を見ながら、コメントや補足説明を行う。

ペアプログラミングを実施するときは、さらに、受講者二人で一つのBreakout roomを作成し、受講者2名のペアは自由に話しながら作業をできるようにした。TAは、担当する5組のBreakout roomをオンラインで巡回した。

受講者は会話しながら、それぞれのPC上でプログラムを書き、コンパイルと実行を行う。ただし、プログラムは、Google Drive上の指定した場所にすぐにおき、ペアの相手を利用できるようにさせた。Google Driveに置かれたものをペアプログラミングの成果物とした。また、二人で作業した時間(分数)の記録も提出させた。

### 4.2 アンケートの実施

実習の時間ごとに、Moodleを使用して、ペアプログラミングに関するアンケートを実施した。アンケートでは、前回のペアプログラミングについて円滑に行われたかどうか、アンチパターン説明した問題ある行動があったかについて質問した。また、共同作業に関する好みと、時間をかけることに関する好みについても尋ねた。アンケート項目の詳細は、付録に示す。

### 4.3 試験の実施

第4タームの終わり、第9週にはオンラインテストを行った。従来は筆記試験で行っていた、プログラムを読む問題は、Moodle上の多肢選択問題にし、用語の説明問題は、Moodle上の空白記入問題に変更した。そのため、2019年度との比較は全ての問題に対してはできない。

が、実技試験の課題「決められた時間内に指定された機能

のJavaのプログラムを作成しなさい」は、ほぼ同じ3個の課題を使用した。以下のデータ解析で、Write1、Write2、Write3と書いてあるのは、この3個の課題である。

## 5. データ解析

### 5.1 2019年度と2020年度の成績比較

2019年度と2020年度の実技試験3課題の、スコアを比較した。各試験の平均値を図1に示す。

2019年度と対面授業と2020年度のオンライン授業による大きな差はなかったことがわかる。

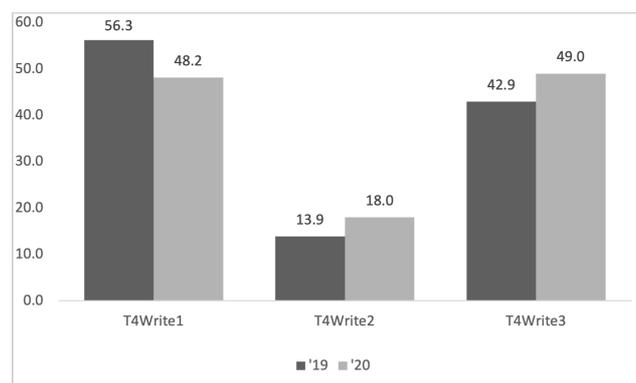


図1 プログラミング実技課題の点数比較

### 5.2 ペアプログラミングに対する個人の満足度の推移

ペアプログラミングの満足度に関する質問項目「ペアでスムーズに課題を進められましたか?」について、「スムーズだった」「まあまあスムーズだった」「あまりスムーズでなかった」「スムーズでなかった」の4段階評価により回答を求めた。各段階の回答割合の推移は(図2)のような結果となった。

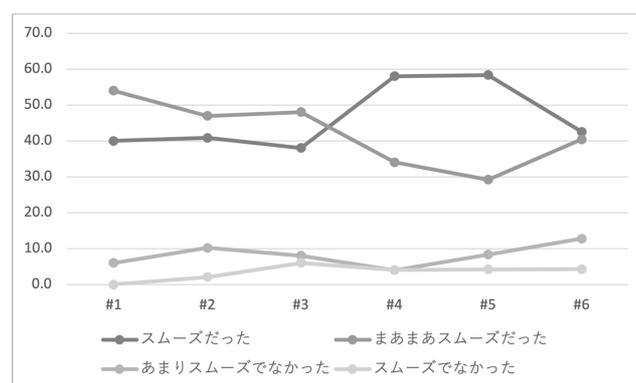


図2 ペアプログラミングの満足度の推移

次に、課題の取り組みタイプや試験のスコアが満足度に影響しているかを調査するため、重回帰分析を実施した。

まず、各回について、満足度を目的変数、作業の取り組みタイプ、各試験のスコアを説明変数として、重回帰分析を実

施した。各回のうち、「3,4 回目の満足度のペア平均値」は（決定係数：0.607641302324056 となった。

また、同じペアごとの満足度の平均値（satis\_q1q2\_ave, satis\_q3q4\_ave, satis\_q5q6\_ave), すべての回の満足度の平均値（satis\_all\_ave) を目的変数とし, n 回目の課題の取り組み形式（qn\_type01), n 回目の課題の取り組みペース（qn\_type02), 各試験のスコア（Word, Read, Write1, Write2, Write3) を説明変数として, 重回帰分析を実施した（表 4）。

それぞれの重回帰分析のうち、「3,4 回目の満足度の平均値」は決定係数：0.607641302324056 となり, 表 3 のような標準回帰係数になった。（決定係数：0.607641302324056）, 「すべての回の満足度の平均値」は, 決定係数：0.596807456363165 になり, 標準回帰係数について, 表 4 のようになった。

（表 3）重回帰分析結果  
目的関数：3, 4 回目の満足度の平均値

説明変数	標準回帰係数
3 回目：作業の取り組み形式 (q3_type01)	-0.169573
3 回目：作業の取り組みペース (q3_type02)	0.701474
4 回目：作業の取り組み形式 (q4_type01)	0.252445
4 回目：作業の取り組みペース (q4_type02)	-0.494759
Reading 試験の得点 (Read)	0.196929
Word 試験の得点 (Word)	0.320504
Writing 試験(1)の得点 (Write1)	0.165086
Writing 試験(2)の得点 (Write2)	-0.237724
Writing 試験(3)の得点 (Write3)	0.043374
第 3 ターム Writing 試験の得点 (T3Write)	-0.067148

（表 4）重回帰分析結果  
目的関数：全 6 回の満足度の平均値

説明変数	標準偏回帰係数
1 回目：課題の取り組み形式 (q1_type01)	0.292067
1 回目：課題の取り組みペース (q1_type02)	0.029563

2 回目：作業の取り組み形式 (q2_type01)	-0.165551
2 回目：作業の取り組みペース (q2_type02)	-0.094955
3 回目：作業の取り組み形式 (q3_type01)	0.015972
3 回目：作業の取り組みペース (q3_type02)	0.555357
4 回目：課題の取り組み形式 (q4_type01)	0.048116
4 回目：課題の取り組みペース (q4_type02)	-0.388908
5 回目：課題の取り組み形式 (q5_type01)	-0.207460
5 回目：課題の取り組みペース (q5_type02)	0.276445
6 回目：課題の取り組み形式 (q6_type01)	0.076573
6 回目：課題の取り組みペース (q6_type02)	-0.238379
Reading 試験の得点 (Read)	0.062013
Word 試験の得点 (Word)	0.186537
Writing 試験(1)の得点 (Write1)	0.024072
Writing 試験(2)の得点 (Write2)	-0.083889
Writing 試験(3)の得点 (Write3)	0.012768
第 3 ターム Writing 試験の得点 (T3Write)	0.032565

これらの結果から, ペアプログラミングに対する満足度は, 作業の好み（一人でしたいかどうか, 時間をかけたかどうか）や, 成績で全部は説明できないが, 一定の関係があることが分かる。

## 6. おわりに

ペアプログラミングをオンライン実習に導入し, プログラミングのコミュニケーション能力の向上や, オンラインツールの利用機会の増加を目指した。プログラムを書く試験の成績は, 対面型授業の場合とほぼ変わらない結果を得た。

ペアの組み合わせについて配慮したため, ペアプログラミングに不満足という回答は少なかった。これは, 作業に関する好み（タイプ）と成績に配慮した組み合わせの成果であ

ることを期待したが、重回帰分析では、ペアプログラミングの満足度と、成績やタイプと強い関係があるとは必ずしも言えない結果であった。

今後、ソフトウェア開発作業自体が、オンラインまたはハイブリッドのチーム作業になる可能性が高い。オンライン環境でのペアプログラミングに影響を与える要素を詳しく分析し、ペアプログラミングを入門レベルのプログラミング教育に積極的にとりいれていきたい。

## 参考文献

- [1]Laurie A. Williams and Robert R. Kessler. 2000. All I really need to know about pair programming I learned in kindergarten. *Commun. ACM* 43, 5 (May 2000), 108–114. DOI:<https://doi.org/10.1145/332833.332848>
- [2]Laurie Williams and Robert Kessler. 2002. Pair Programming Illuminated. Addison-Wesley Longman Publishing Co., Inc., USA. (邦訳 ローリー ウィリアムズ, ロバート ケスラー ペアプログラミング—エンジニアとしての指南書. ピアソンエデュケーション, 2003, 279p.)
- [3]Linda L. Werner, Brian Hanks, and Charlie McDowell. 2004. Pair-programming helps female computer science students. *J. Educ. Resour. Comput.* 4, 1 (March 2004), 4–es. DOI:<https://doi.org/10.1145/1060071.1060075>
- [4]Brian Hanks. 2005. Student performance in CS1 with distributed pair programming. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05)*. Association for Computing Machinery, New York, NY, USA, 316–320. DOI:<https://doi.org/10.1145/1067445.1067532>
- [5]Hans Yuan and Yingjun Cao. 2019. Hybrid Pair Programming - A Promising Alternative to Standard Pair Programming. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 1046–1052. DOI:<https://doi.org/10.1145/3287324.3287352>
- [6]Neha Katira, Laurie Williams, Eric Wiebe, Carol Miller, Suzanne Balik, and Ed Gehringer. 2004. On understanding compatibility of student pair programmers. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education (SIGCSE '04)*. Association for Computing Machinery, New York, NY, USA, 7–11. DOI:<https://doi.org/10.1145/971300.971307>
- [7]<https://givery.co.jp/services/track/>
- [8]Siddharth, “Pair Programming Antipatterns: XPerience”, <https://sidstechcafe.com/pair-programming-antipatterns-xperience-792fe0112aa1>
- [9]Digvijay Gunjal, “Pair Programming Anti Patterns”, <https://www.youtube.com/watch?v=McZ131y0OYU>

## 付録

### 付録 A.1

#### プログラミングの取り組み方に関する質問

1. 演習時間中に取り組む課題は  
[誰かと相談しながら進めたい派/個人で進めたい派]
2. 課題は  
[できるだけ早く終わらせたい派/じっくり時間をかけて取り組みたい派]

#### ペアプログラミングの満足度に関する質問

3. ペアでスムーズに課題を進められましたか?  
[スムーズだった/まあまあスムーズだった/あまりスムーズでなかった/スムーズでなかった]

#### ペアプロで避けるべきことに関する質問

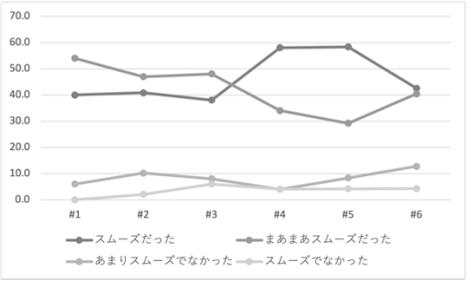
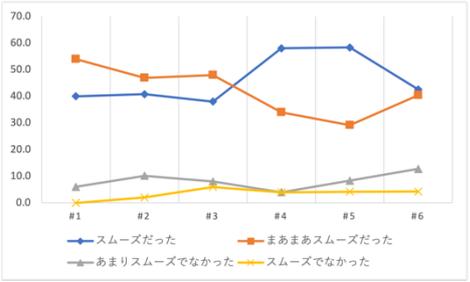
4. 「ペアプログラミングで避ける8つのこと」に基づいて、先週のペアプログラミングの取り組みを振り返ってみましょう。
  - Keyboard Hogging キーボードの独り占め. 二人ともドライバーになろうとしてキーボードを取り合うこと. [避けられた/避けられなかった]
  - Keyboard Hogging に関するコメント (任意)
  - Pair Affinity ペア相手の選り好み. 誰とペアになるかについて, こだわること. [避けられた/避けられなかった]
  - Pair Affinity に関するコメント (任意)
  - Distraction 注意散漫. プログラミングに集中せず, 他のことに注意を向けること. [避けられた/避けられなかった]
  - Distraction に関するコメント (任意)
  - Dictating Pair 主張するペア. Driver が Navigator の意見に反論すること. [避けられた/避けられなかった]
  - Dictating Pair に関するコメント (任意)
  - Pair Absenteeism 欠席するペア. Pair のどちらかが席を外し, パートナーを一人にすること. [避けられた/避けられなかった]
  - Pair Absenteeism に関するコメント (任意)
  - Passive Pairing 受け身のペア. Pair の一人が意見を言わず, 残りの一人が Navigator と Driver の二役をすること. [避けられた/避けられなかった]
  - Passive Pairing に関するコメント (任意)
  - Partners in Crime 罪深いペア. プログラムが完成していないのに, 二人でプログラミングをやめること. [避けられた/避けられなかった]
  - Partners in Crime に関するコメント (任意)
  - Philosophical Pair 哲学的なペア. プログラミングと関係ないことを話すこと. [避けられた/避けられなかった]
  - Philosophical Pair に関するコメント (任意)

#### 自由記述

5. ペアプログラミングをしてみて, よかったと感じた点 (任意)
6. ペアプログラミングをしてみて, 難しいと感じた点 (任意)
7. 反省や要望など (任意)

## 正誤表

下記の箇所に誤りがございました。お詫びして訂正いたします。

訂正箇所	誤	正
3 ページ 右段 2 行目	以下のデータ解析で, Write1, Write2, Write3 と書いてあるのは, この 3 個の課題である.	削除
3 ページ 5.1 1 行目	2019 年度と 2020 年度の実技試験 3 課題の, スコアを比較した.各試験の平均値を図 1 に示す. 2019 年度と対面授業と 2020 年度のオンライン授業による大きな差はなかったことがわかる.	2019 年度と 2020 年度の実技試験 3 課題 ( T4Write1, T4Write2, T4Write3) の, スコアを比較した. 2019 年度の対面授業と 2020 年度のオンライン授業による大きな差はなかったことがわかる.
3 ページ 図 2		
4 ページ 1,2 行目	各回のうち, 「3,4 回目の満足度のペア平均値」は ( 決定係数:0.607641302324056 ) となった.	分析の結果, いずれも決定係数は 0.5 未満であった.