

# 反転型のプログラミング授業における モブプログラミング導入の試み

山川 広人<sup>\*1</sup>, 上野 春毅<sup>\*2</sup>, 小松川 浩<sup>\*1\*2</sup>

<sup>\*1</sup> 公立千歳科学技術大学 情報システム工学科

<sup>\*1</sup> 公立千歳科学技術大学 大学院光科学研究科

## Trial of Mob-Programming in Flipped Classroom

Hiroto YAMAKAWA<sup>\*1</sup>, Haruki UENO<sup>\*2</sup>, Hiroshi KOMATSUGAWA<sup>\*1\*2</sup>

<sup>\*1</sup> Dept. of Information Systems Engineering, Chitose Institute of Science and Technology

<sup>\*2</sup> Faculty of Science and Technology, Chitose Institute of Science and Technology

本研究は反転型のプログラミング授業における知識の補習・実習にモブプログラミングを活用できる可能性の検討を目的とする。反転授業では学生の予習を前提とし、授業では予習した知識の補習・実習を通じた確実な習得をはかる。筆者らの先行研究では CBT で学生の予習度を判定し、これに基づいたグループワークで知識を補習し、個々の実習課題でスキルとしての習得をはかるモデルを提案した。本稿はこのグループワーク（知識の補習部分）にモブプログラミングを導入した授業を試みる。

キーワード: モブプログラミング, プログラミング教育, 反転学習

### 1. はじめに

知識やスキルの習得の積み上げが重要であるプログラミング科目では、基礎的な知識の学習は予習部分に充て、授業内では予習を前提としてより深い知識・スキルの習得を図る補習・実習を行う反転授業<sup>(1)</sup>が有効であろう。筆者らは先行研究で、CBT (Computer Based Testing) で学生の予習度を測定し、これに基づいたグループワークで知識を補習し、個人の実習課題でより深い理解や定着を図る反転授業モデルを提案した上で、学生の習得状況の底上げへの効果や、つまづいた学生の把握による動的な授業進度の調整手法を検討してきた<sup>(2-3)</sup>。しかしながら授業内では、学生の知識の習得状況の違いにより、グループワークに向かう意識や姿勢の差が発生する例、実習にむけ十分な補習ができない学生が生じる例、共有できる知識に限られる例など、課題が残されている。

実務的なシステム開発現場では、近年、モブプログラミング<sup>(4-5)</sup>に注目が集まっている。モブプログラミングはチーム開発手法の一種で、ひとつの開発課題をチ

ーム全員で解決するスタイルをとる。複数人が協議し方針・手段を考えながら進める過程を経ることで、高度な課題にも対応しやすいだけでなく、課題内外の専門知識をチーム内で共有し、チームメンバーを成長させる効果も期待されている。

本研究は、モブプログラミングの知識の共有・チームメンバーの成長効果に着目し、前述の反転学習モデルのグループワークの改善にモブプログラミングの導入が効果を発揮する可能性の検討を目的とする。本稿は特に、反転学習モデルの授業にモブプログラミングの導入を試み、学生のアンケート結果からその可能性を考える。

### 2. ベースとなる反転授業モデル

本稿の試みのベースとなる、先行研究の反転授業モデルの概要と、本稿で取り上げる課題について述べる。

#### 2.1 先行研究の反転授業モデルの概要

この反転授業モデルはプログラミング科目を対象に、ひとつの学習単元を3週分 (Step1.用語と概念の理解、

Step2.知識の活用, Step3.知識の発展, の3Step)の授業で進めることを前提とする。各週の授業は①～⑥のシナリオで進められる。

- ① 学生はeラーニングで、各週の学習単元・Stepにむけた予習を行う
- ② 授業開始時に、学生は予習内容と紐付いたCBTで予習度の判定を行う
- ③ 学生は個人ワークで、その回の実習に求められる具体的な知識を再確認する
- ④ CBTの結果を反映したグループの中で、学生は個人ワークの不明点を確認し、時には教え合う
- ⑤ 学生はPC教室で個人のプログラミング実習を行い、知識のより深い理解・定着にむけて訓練する
- ⑥ 演習後、学生はその日の自分の成果や行動を振り返り、カードにコメントとして記入・提出する

①では、学生は背景知識の解説資料(PDF, 映像など)を使いながらeラーニングシステム内の演習問題を予習する。演習問題は学習単元の3週分(Step1~3)との対応を意識し、表1に示す7段階の難易度によって整備されている。予習の指示では、授業で扱う学習単元およびStepの演習問題を最低限の範囲とする。ただし学生の自己調整を妨げないよう、過去のStepの復や、Stepを先取る予習は制限しない。

②では同システム内のCBT機能④を使い、学生の予習度を判定する。予習度の判定は授業で扱う単位の中で行う<sup>i</sup>。テスト問題は演習問題と同じ内容であり、予習の成果がテスト回答に現れることを狙っている。学生がCBTを受検すると、その1問1問の正誤状況から難易度の異なる問題が出題される。これを繰り返すことで、学生が十分に解けると判定された演習問題群の難易度の値を算出し<sup>ii</sup>、予習度として表示する<sup>iii</sup>。

③では、⑤で必要な知識を問う紙のワークシートに取り組む。ワークシートの内容は予習した知識を使い、語句や仕組みの説明、文法例、プログラミング例などに穴埋めをする形式が主となる。ワークシートを進め

表 1 学習単元内の Step と演習問題の難易度の対応

| 学習単元の Step                 | ラベル | 演習問題の難易度 |
|----------------------------|-----|----------|
| Step1 (1週目の授業)<br>用語と概念の理解 | 基礎  | 1        |
|                            | 標準  | 2        |
| Step2 (2週目の授業)<br>知識の活用    | 基礎  | 3        |
|                            | 標準  | 4        |
|                            | 応用  | 5        |
| Step3 (3週目の授業)<br>知識の展開    | 基礎  | 6        |
|                            | 標準  | 7        |

注) 中間に位置する Step2 は前後のつながりを考慮して難易度を3つにわけている

ることで、実習の達成に用いる知識を再確認し、自身にとって補習を要する箇所を明らかにする狙いがある。

④では、③で明らかになった要補習の箇所を学生どうしのグループワークを通じて解決する。グループ分けには②で判定した予習度を用い、予習度が同程度の学生が偏らないグルーピングを配慮する。ワークシートの答え合わせではなく、ワークシートを元に予習度の異なる学生が自身の互いの予習度や習得状況と向き合い協力しながら、⑤に向けた準備を整えることを狙っている。

⑤では、学生は③・④を経て確認・補習した知識を踏まえ、プログラミング実習に取り組む。ワークシートの例題や、授業のゴールとなる課題を各自でプログラミングし、試行錯誤や達成することで、知識のより深い理解やスキル化が図れることを狙っている。

⑥では、予習や授業を通じた自身の行動・成果をふりかえる。次週の予習や授業に向けた心構えを整える機会を設けることで、学生自身が反転授業モデルのサイクルを継続できるようにすることを狙っている。

## 2.2 本稿で取り上げる課題

本稿は2.1節のシナリオの中で、特に④のグループワークの部分で考えられる3点の課題を対象とする。**グループワークに向かう意識や姿勢の構築の点**：グループワークには、予習度が高い学生から「できる人が一方的に教えるばかりである」「できない・やる気がな

<sup>i</sup> 予習範囲は単元内のStepまで指示するにも関わらず、予習度は単元ごとに判定する理由は、予習段階での学生の自己調整学習を妨げないことにある。つまり、ある単元を扱う3週のうち、早い段階でStep3部分まで予習を進めた学生や、単位の中でつまずきStepの進行に追従できていない学生、およびそれらの度合いを予習度により可視化したい狙いがある。

<sup>ii</sup> 予習度の詳細な算出ロジックは参考文献(5)を参照されたい。

<sup>iii</sup> 例) ある学生に判定された予習度が4であれば、その学生は単元の中で「第2週に必要な知識の標準段階の難易度まで予習できている」という見方ができる。

い学生をどうしてサポートしなくてはいけないのか」という声が、そうではない学生からは「わかりやすく教えてもらえない」「わからないまま進んだ」という声が寄せられることがある。これは予習度が異なる学生がグループ内で協力しあう主旨とは異なる意識・姿勢で取り組んでしまうことのシグナルとも捉えられ、学生の意識や姿勢を整う工夫の余地がある。

**十分な補習ができない学生が生じる点**：学習単元や⑥の実習内容にも大きく影響されるが、特に予習度の分散が大きい週では、実習を完了できない学生が多数発生する例もある<sup>9)</sup>。2.1 節のモデルではグループワークが実習前の準備の要であり、学生達が実習に不足がない段階に到達できているかグループワークの中で相互の理解度を判断できる工夫の余地がある。

**共有できる知識に限りがある点**：グループワークはワークシートの内容に基づいて行われるため、単元や実習で直接的に取り上げるコア部分の知識の補習は期待できるものの、個々が実習段階で用いている細やかなノウハウやテクニックの共有までは行われ難い。実習に向けた準備として効果を十二分に高めるためには、こうした派生部分の共有も促せる工夫の余地がある。

### 3. モブプログラミングとその導入の試み

2.2 節の 3 点の課題は、モブプログラミングが効果的であると言われている点とも共通部分が多い。そこで本研究では、グループワークのフレームワークとしてモブプログラミングの導入を試行し、3 点の課題に向けた効果が発揮されると仮定する。3.1 節にモブプログラミングの概要を、3.2 節では本稿の反転授業にモブプログラミングを試行的に導入する方法を述べる。

#### 3.1 モブプログラミングの概要

モブプログラミング<sup>9)</sup>は近年、システム開発現場で採用されているチーム開発手法の一種であり、ひとつの PC でひとつの開発課題をチーム全員で解決するスタイルをとる。チームメンバーはタイピスト（1 名）とモブ（その他名）の 2 つの役割に分かれる（図 1 に例示）。タイピストは PC の操作や入力を担当する。モブはタイピストに PC の操作や入力の指示をする。タイピストとモブは 5～15 分程度の時間で順番に交代していく。

モブプログラミングを進める上で重要な部分は、ゴールに向けた方針やそれを実現する手段をモブ全員で協議し、合意が得た指示をタイピストに伝えることと、タイピストがモブの指示を忠実に PC やプログラミング上に反映されるように操作を行うことである。これによりモブの指示（つまり、タイピストの操作）はモブ全員の集合知が包含され、高度な課題への対応が期待できるようになる。さらに、副次的な効果としてチームメンバーのスキルアップも期待できる。モブが自身の意見を提案し、モブ間で協調的な質疑や議論が行われることで、意見の背景となる個々の知識・経験・スキルを相互に理解し、共有・学習できるプロセスを兼ねられる。タイピストは細やかな操作方法も含めてモブから様々な指示やフィードバックを受けて手がける。これはモブから共有されるノウハウ・テクニックを実践的にトレーニングするプロセスも兼ねられる。

#### 3.2 反転授業への試行的な導入方法

3.1 節に述べた利点を前提に 2.2 節の課題を振り返ると、グループワークに向かう意識や姿勢の構築には、モブとして協調的な姿勢が求められる仕組みを利用できる。十分な補習ができない学生が生じる点には、タイピストへの指示のために、各個の背景知識を含めて質疑や議論を通じ共有・学習できるプロセスが利用できる。共有できる知識に限りがある点には、実際に PC を操作する環境で細やかな指示やフィードバックを通じたノウハウ・テクニックのトレーニングを兼ねるプロセスが利用できる。本稿ではこの点を意識して、反転授業へモブプログラミングを導入する試みを行う。

##### 3.2.1 モブプログラミングを導入したシナリオの想定

2.1 節の反転授業のサイクルへのモブプログラミン



図 1 モブプログラミングでの活動風景

グ導入を考えると、④ないしはその前段となる③も含めた部分を以下のように変更する案が想定できる。

- ①' 学生は e ラーニングで、各週の学習単元・Step にむけた予習を行う
- ②' 授業開始時に、学生は予習内容と紐付いた CBT で予習度の判定を行う
- ③' 〈変更部分〉：CBT の結果を反映したグループの中で、学生はモブプログラミングで実習に向けた不明点を確認し、知識を補習する
- ④' 学生は PC 教室で個人のプログラミング実習を行い、知識のより深い理解・定着にむけて訓練する
- ⑤' 演習後、学生はその日の自分の成果や行動を振り返り、カードにコメントとして記入・提出する

本稿ではこうしたシナリオを実際に実現できるかを検討する。本稿は試行の第一歩目の段階であるから、①'~⑤'を通しての試行は今後の課題とし、②', ③'の流れを重点的に試行する。

### 3.2.2 モブプログラミング環境の準備

モブプログラミングを行うためには、グループごとにタイピストが操作する 1 台の PC と、画面をモブにむけて表示するディスプレイが必要になる。PC 教室内への用意することは難しいため、グループワークが行える学生ホールのデスク上にノート PC とディスプレイを 1 台ずつ、計 15 セット (15 グループ分) 用意した。ノート PC は PC 教室の実習時と同じソフトウェア (AdoptOpenJDK 11, JetBrains 社 IntelliJ IDEA) が利用できる準備を整えた。

### 3.2.3 モブプログラミングの解説資料

多くの学生にとって、開発現場で用いられるモブプログラミングは聞き慣れない・未体験のものであろう。そこで、モブプログラミングの方法・手順をまとめた解説資料を用意した。解説資料は筆者が作成し、チーム活動の重要性、チーム活動に向かう姿勢 (The Drucker Exercise<sup>(6)</sup>と心理的安全性<sup>(7)</sup>)<sup>iv</sup>、モブプログラミングの方法、モブプログラミングを進める上での注意点の 4 点をまとめた PDF である (図 2 に例示)。

モブプログラミングではモブの協調作業が活動の主となるため、モブ自身が協調を意識して活動する必要がある。チーム活動の重要性と姿勢のページを設けることで、学生に意識・姿勢の徹底を促す狙いがある。

### 3.2.4 モブプログラミング用ワークシート

モブプログラミングを行う上で、実習に向けた準備を整えるためのワークシートが必要となる。従前のワークシートは先述の通り穴埋めの問題形式による知識の確認に重きを置いている。モブプログラミングの導入後においてはより実践的な (実習に近い) 課題で、ノウハウ・テクニックの共有も目指したい。本稿では将来的な実習課題との接続も考慮し、実習課題を達成する手順を順番に解決する内容とした (図 3 に例示)。この形での課題が達成できれば、③'は実習課題の前半部分までをモブプログラミングで行い、④'は実習課題の後半部分や自由度の高い課題を確認するなどといった実用が可能であろう。

## 4. 試行とアンケート評価

3.2 節の準備を整えた上で、授業の 1 回分を用い、モブプログラミング (3.2 節の②'・③') の導入を試みた。対象となった授業は Java プログラミング (学部 2 年生, 出席者 79 名) で、2.1 節の反転授業モデルを適用した科目である。

### 4.1 試行の流れと様子

試行した週はすべての単元の学習を終え科目のまとめ・発展的な学習を行う最終週であったため、学生には予習の指示が行われていない。②'では、過去の学習単元の Step.3 の CBT 結果の中で最も分散が大きい (つまり、学習単元の中で習得が十分にできている知識に差がある可能性が高い) 判定結果を CBT 結果の変わりとして用いた。

③'では、指導教員 (筆者) が 3.2.3~4 項の解説資料とワークシートを用いて学生に流れを解説したあと、②'で 4~5 名前後に分けたグループでのモブプログラ

<sup>iv</sup> The Drucker Exercise : チーム活動に対し「自分は何が得意か」「自分はどのくらい貢献するつもりか」「自分が大切にしたい価値は何か」「自分に期待される成果は何か」の 4 点を共有し、メンバー間の相互理解を促進する手法。

心理的安全性 : チーム活動で、メンバーがリスクを恐れずに活動できるようにするための心構え。本稿では特に「作業を学びの機会と捉える」「自分の誤りを認める」「好奇心をもってたくさん質問する」という手法。

これらは筆者が学生にとって手法がわかりやすい例として判断し、意識や姿勢の構築方法として併せて紹介した。

ミングを開始した(図4に授業風景を例示)。モブプログラミングは80分間、タイピストは8分を基準に交代(タイピストを2回程度体験可能)とし、担当教員がタイムキーパーと交代の指示を行った。開始当初~数回までのタイピスト交代の段階では、モブ役の学生の発言が少ないグループが目立ち、一人の一方的な指示によって進んでしまう姿や、進め方に戸惑う姿が見られた。そこでタイピスト交代の指示の度に、解説にも記載した心理的安全性の徹底や、レポート用紙にイラストを描き説明をする・質問や意見を述べづらひ際は意思表示の合図を取り入れることなどの工夫例もアドバイスし、雰囲気作りを図った。次第にモブどうしの意見交換や質問の元で課題を進められるグループが増え、活発なモブプログラミングが行われた。

## 4.2 アンケート評価と考察

授業終了後に学生へのアンケートを実施した(有効回答者数75名)。設問と回答を表2,3に示す。

Q1~3には約8割の学生が肯定的に回答した。Q1~2は、学生がモブ・タイピストの活動を共に進められた実感の現れであり、授業の様子からも、3.2.3の解説資料や授業中のアドバイスが有効に働いた可能性が高い。Q3の理由の肯定的な記述では、難度の高い課題の解決を通じた理解や、協働を通じた新たな知識・テクニックの取得の様子がみられる。一方で、協議の中で合意が形成できなかった場合や、目新しい知識を得られなかった学生が否定的な記述をしている。モブ間の協議やタイピストとして指示を反映する過程のかみ合い方が、意欲・姿勢の維持や相互の理解度に基づいた活動、派生知識の習得効果に影響することがうかがえる。

一方で、Q4では従前のグループワーク方法を本稿のモブプログラミングに代替することを望まない学生も多数であった。Q4の理由の記述から、ひとつは活動に係る時間の問題、ひとつは学生が従前のグループワークで得ていた効果がモブプログラミングでは兼ねられない例があることに気づく。活動に係る時間の面では、図3の課題の指示のうち、80分で最も進捗があったグループの到達は第5段階であった。モブの協議やタイピストによる反映を都度行うことから、モブプログラミングは開発の速度を上げるものではないが、学生には進度が遅いよう映った可能性がある。従前の

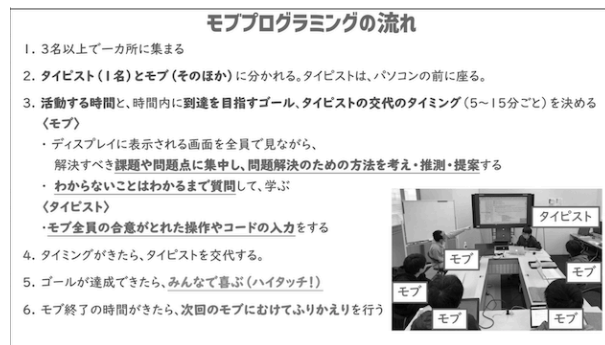
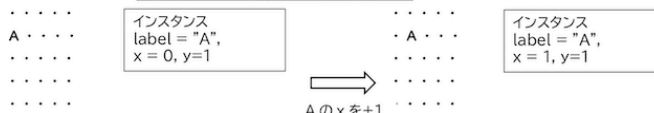


図2 モブプログラミングの解説資料

例:5×5マスのマップに、ゴール:Gと操作対象:Aを配置する。  
G, Aはそれぞれx, y座標のデータを持たせることで、横軸x(0~4)、縦軸y(0~4)のどの位置に表示すればよいか分かる。またx, y座標のデータを変化させる機能を持たせることで、表示位置を移動させることもできる。



上記の例を参考に、以下の順にゲームを実装しなさい。ただし、最後の(条件)は厳守すること。クラス図を作りたいときは、裏面や別紙に記載してよい。

- 5×5マスのマップを用意する。
- 操作対象:Aを配置する。配置場所はランダムとする。
- waszキーの押下にあわせて、操作対象Aがいずれかの方向に1マス移動できるようにする。
- 調査対象Aがマスの端から外側に移動すると、逆端から出現できるようにする。
- 操作対象Aが1マス移動すると、得点が1点加算されるようにする。
- 減点マス:Eを配置する。配置場所はランダムとするが、必ず調査対象A以外の場所に初期配置する。
- 操作対象Aが1マス移動すると、減点マスEも1マス、いずれかの方向に移動する。
- 操作対象AがEと重なった場合はX(エックス)を表示し、得点が1点減点される。
- 操作対象Aが4回移動することに、減点マスEが1つ増える。
- 操作対象Aが20回移動すると、ゲームを終了し得点を表示する。

図3 モブプログラミングで課した課題(抜粋)



図4 モブプログラミングでの授業風景

注) 右上:タイピスト役、その他はモブ役の学生  
モブ役の学生達が協議した指示を、  
タイピスト役の学生がノートPCで反映・表示させる  
グループワークで得ていた効果の面では、学生の手元に残る成果物やそれをを用いる復習のしやすさは、学習単元やStepの中でつまづきを感じている学生にも重要であろう。本稿の試みにあたり、ワークシート内容を大きく改訂した影響も考えられる。この部分を再検討した上で、グループワークにモブプログラミングを導入する際は、従前のグループワークの利点もふまえ、通常時は従来の反転型の授業モデル、単元のまとめや発展的学習にはモブプログラミング、といった使い分

けの必要性が示唆されたと考えている。

## 5. おわりに

先行研究で提案した反転授業モデルでは、グループワークが予習部分の不足と授業のゴールとなる実習部分をつなぐ要である。ここに生じる課題に対し、モブプログラミングを導入する試みを行った。モブプログラミングの効果の発揮を伺わせる結果が得られた一方で、学生がグループワークに求める効果の違いを考慮した導入シナリオの検討が新たな課題となった。

## 謝辞

本研究の一部は JSPS 科研費基盤研究 (C) 17K00492 の助成を受けたものである。

## 参考文献

- (1) 重田勝介: "反転授業 ICT による教育改革の進展", 情報管理, Vol.56, No.10, pp.677-684 (2013)
- (2) Ueno, H., Kato, T., Yoshida, F., Tsukada, N., Tateno, H., Fukamachi, K., Yamakawa, H. and Komatsugawa, H. : "A Model of Flipped Classroom Using an Adaptive Learning System", 25th International Conference on Computers in Education: Work in Progress Posters Proceedings, pp.1-3 (2017)
- (3) 山川広人, 加藤巽, 上野春毅, 小松川浩: "CBT を活用した反転型授業の Java プログラミング授業での実践, 教育システム情報学会研究報告 Vol.33, No.2, pp.57-62 (2018)
- (4) 上野春毅, 光永悠彦, 山川広人, 小松川浩: "段階的な学習目標を持つ反転学習モデルのための適応型学習システムの開発", 教育システム情報学会誌, Vol.37, No.3 (2020 掲載予定)
- (5) Mark Pearl : " Code with the Wisdom of the Crowd: Get Better Together with Mob Programming", Pragmatic Bookshelf (2018)
- (6) Jonathan Rasmusson : "The Agile Samurai: How Agile Masters Deliver Great Software", Pragmatic Bookshelf (2010)
- (7) re:Work - Guide, Tool: Foster psychological safety, <https://rework.withgoogle.com/guides/understanding-team-effectiveness/steps/identify-dynamics-of-effective-teams/> (2020 年 2 月 6 日確認)

表 2 Q1~Q3 の設問と回答人数 (n=75)

| 設問   |  |    |    |
|--|--|----|----|
| Q1. あなたはモブ役として、積極的に発言をできましたか？  |  |    |    |
| Q2. あなたはタイプスト役として、モブ全員の合意に基づいた PC 操作やコード入力ことができましたか？                   |  |    |    |
| Q3. モブプログラミングを通じて、Java やプログラミングについてわからなかった点の解決や、新たな手法に気づきを得ることができましたか？ |  |    |    |
| 回答項目   | Q1   | Q2 | Q3 |
| とてもよくあてはまる   | 29   | 34 | 20 |
| どちらかといえばあてはまる  | 32   | 36 | 47 |
| どちらかといえばあてはまらない  | 13   | 4  | 5  |
| あまりあてはまらない   | 1  | 1  | 3  |
| Q3 の理由例 ※筆者が抜粋し、着目点に下線したもの   |  |    |    |
| 肯定的  | <ul style="list-style-type: none"> <li>● 5×5 マスを作ることに苦戦した。その中で、<u>ArrayList</u> での二次元配列の仕組みについて議論して理解することができた。</li> <li>● チームで動くので脳みそが 5 つありました。そこで答えは 1 つではなく、いろいろな取り組み方が知れて、面白かったため。</li> <li>● <u>Intelij</u> での速いプログラムの書き方の新たな手法を得ることができた</li> </ul> |    |    |
| 否定的  | <ul style="list-style-type: none"> <li>● (前略) 極力モブの指示に従って動くようにしていたが、その時も同意が形成される気配がなく、結果としてモブプログラミングの利点が私のグループでは上手く機能しなかった</li> <li>● 人の <u>コーディングの仕方</u> について色々学ぶことはあったが、特に <u>知らない知識はなかった</u>。</li> </ul>  |    |    |

表 3 Q4 の設問と回答人数 (n=75)

| 設問                                      |   |
|---|---|
| Q4. プログラミングの毎回の授業で、CBT の後に行うグループワーク課題は？ |   |
| 回答項目                                    | Q4  |
| 今回のモブプログラミングの形が良いと強く思う                  | 7   |
| どちらかといえば今回のモブプログラミングの形が良いと思う            | 17  |
| どちらかといえば従来の紙に記入する形が良いと思う                | 33  |
| 従来の紙に記入する形が良いと強く思う                      | 18  |
| Q4 の理由例 ※筆者が抜粋し、着目点に下線したもの              |   |
| Mob が良い理由                               | <ul style="list-style-type: none"> <li>● グループで話し合いできるとともに、<u>実際の動作とも見比べながらできる点</u>、<u>細かな構文の間違いも見つけられる</u>と思う。</li> <li>● <u>実際にプログラムを書いてみてからわかる問題点をグループで検討できるため</u>、<u>従来のグループワークよりも活発で有意義に思えた</u>。</li> </ul> |
| 中立的理由                                   | <ul style="list-style-type: none"> <li>● 従来のやり方のほうが時間という点で個人的にはいいと思ったので、(中略) <u>理解という点でみるなら</u>、<u>モブプログラミングのほうがいい</u>と思った。</li> </ul>  |
| 紙が良い理由                                  | <ul style="list-style-type: none"> <li>● モブプログラミングはメモを取る場面が少なく後で復習することが難しいと感じた。</li> <li>● <u>紙に残すことで復習につながるから</u> (中略) <u>穴埋め形式であると思の手順がわかる</u>のでそれが良いと考える。</li> </ul>   |