

# 情報系初年次のプログラミング演習における，コンピュータとの対話を重視したコースデザインと支援システム

長 慎也<sup>1,a)</sup> 山中 脩也<sup>1</sup> 北島 茂樹<sup>1</sup> 今野 貴之<sup>1</sup>

概要：本学の情報学部では，2019年度より新カリキュラムが施行され，1年次の学生に対して新しいプログラミング演習（以下「演習」）を実施した．演習は，「コンピュータと人の対話によるプログラミング実習」「3人のグループでの作問」「グループ間でのクイズの解答とレビュー」というステップで進行する．各ステップを踏むごとに個人またはグループでインタラクションをする範囲を広げていき，相互の意見や気づきを交換することでプログラミングに関する深い知識を獲得することを目指している．これらのインタラクションを支援する目的で，作問システムを開発した．本発表では演習内で使用した教材と作問システムについて紹介する．

## A course design and system which is focused in “Dialogue with Computer” at introductory programming course

CHO SHINYA<sup>1,a)</sup> YAMANAKA NAOYA<sup>1</sup> KITAJIMA SHIGEKI<sup>1</sup> KONNO TAKAYUKI<sup>1</sup>

### 1. はじめに

本学の情報学部では，新しいカリキュラムのもと，初年次のプログラミングの授業が大幅に改定された．この授業における目標は，プログラミングのルール（文法や意味，アルゴリズムの実現のしかた等）を知識として受講者に定着させることとしている．ここでいう「知識」とは，教科書や教員の板書，口頭での説明などで伝達される「情報」と対比されて使われる言葉で，本人なりの解釈（主観）が入った信念でありながら，正しいことが証明されている（客観的な）ものとする．

このようなレベルの知識を獲得させるためには，教員などから情報を読む・説明するなどしてルールを伝えるだけでは不十分であると考えられ，演習を取り入れることが多い．これまでのプログラミング授業では，次のような流れで座学および演習を行うことが多かった．

- 教員が説明をしながら受講者が手を動かして演習
- 教員が課題を出題

- 受講者が課題を解答・提出
- 教員による採点・必要ならば受講者による再提出

しかし，このような課題を解くことを演習に取り入れたとしても，その課題ができなくて，しかも本人から（質問をするなどの）行動を起こすことができなければ，周囲とのやりとりをすることなくわからないまま授業を終えてしまう．

そこで，獲得した知識を他の受講者に説明したり共有したりすることで，自分では獲得できなかった知識を得ることもでき，より深い学習ができるのではないかと考え，次のようなグループワークを取り入れた授業を提案する．

- (1) 説明された内容を受講者が文章などで表現する
- (2) 他の受講者と文章で表現した内容を共有する
- (3) 文章について意見交換する

このグループワークは，知識獲得のモデルである SECI モデル [1] がもとになっている．SECI モデルでは次の4つの過程を踏むといわれており，これを繰り返すことで知識を創造できるという．

- 共同化 人から人へ知識を伝達させる←従来の「授業」の範疇

<sup>1</sup> 明星大学  
Meisei University, Japan  
<sup>a)</sup> cho@eplang.jp

- 表出化 伝達された知識を言葉で表現する←前述グループワークの(1)に対応
- 連結化 自分が得た知識と他の人の知識を複合させることで新しい知識を得る←(2)
- 内面化 知識を自分の技術として体得する←(3)

ただし、表出化の過程で「わかったことを述べましょう」など、抽象的な指示を与えてしまうと、「難しかった」などただの感想を書く受講者が多くなるので、文章を書かせる代わりに、プログラミングに関するクイズを作成する「作問」を活動として採用した。

また、クイズの質をよくするためにグループで作成したクイズを選定させる工程を取り入れた。他のグループにクイズ解答させ、コメントで評価することにより、グループ間の知識の共有もできるような活動を取り入れた。

本発表では、これらの作問・解答を支援するために開発したシステムについて報告する。

## 2. 授業の概要

1で述べたような、グループワークと作問をプログラミングに適用した授業を、次のような流れで施行した。

- (1) 教材による演習 配布された教材および言語処理系を利用して、毎回のテーマについての知識を獲得する。この段階ではグループの各メンバーが個別に行う。
- (2) グループ形成 3人で1つのグループを組む。
- (3) クイズ案作成 テーマに関連したクイズをメンバー1人につき最低1題ずつ作成する。
- (4) クイズ選定 グループ内で、各メンバーが作成したクイズの中からもっとも「よい」と考えられるものを1つ選ぶ。ここでいう「よい」とは「難易度が適切である」という意味であり、誰でも正解できる簡単なクイズや、誰も正解できない難しいクイズを避けて出題させるようにした。
- (5) 解答 他のグループが出題したクイズを1人1題解答する。
- (6) 解答照合・コメント 解答の答え合わせを行い、クイズ自体に対するコメントを共有する。

これらの流れを先述のSECIモデルに対応させると、(1)が共同化、(3)(4)が表出化、(5)連結化が、(6)が内面化に対応している。毎週、この流れを継続的に繰り返すことで、プログラミングの知識を獲得させることを目指している。

また、受講者のモチベーションを保つため、受講者の活動の成果を「星」という形で毎回得点化するようにした。星を獲得する方法は次の2つがある。

- 他グループが作成したクイズに正解すると、個人で星を獲得できる。
- 自分のグループで作成したクイズの正解率によってグループが星を得る。正解率が2/3に近いほど多く得ることができる(詳しい算出方法は後述)。得た星を

グループ内で協議の上メンバーに割り振る。

これらの活動を実施した授業は「プログラミング演習」および「プログラミング序論」という2つの科目にまたがっている。「プログラミング演習」は月曜日と金曜日の週2回行われ、金曜日は上記(1)の演習を行い、月曜日は(2)から(4)の作問を行い、次の時限に「プログラミング序論」が行われ、(5)(6)の解答やコメント付けを行う。なお、「プログラミング演習」は約50人ずつの3クラスに分かれており、「プログラミング序論」は1つのクラスで約150人が受講した。

「プログラミング演習」のシラバスを表1に示す。このうち、「アルゴリズムを学ぶ」という部分ではPythonを用いて演習を行い、後半からはC言語を用いて演習を行っている。

プログラミング環境にはオンラインプログラミング環境のpaiza[2]を使用しているが、後期の授業で本格的にプログラミングを行うことに備えてUNIX環境のシェルスクリプトの実習も取り入れた。また、学期末に行った試験ではBitArrow[3]を使用し、C[4]またはPython[5]でプログラムを作成させ、受講者の作成したプログラムのログをもとに採点を行った。

## 3. 「コンピュータとの対話」に重点を置いた教材

1で述べた通り、プログラミング教育の目標はプログラミングのルールを知識として受講者に定着させることであり、知識とは、本人の信念(主観)と正当性(客観)を兼ね備えたものでなければならないといえる。このような知識を定着させるために、教材の構成を次の段階を短いサイクルで繰り返すものとした。この流れを図1に示す。

- **Phase1** 教材が示したソースコードを受講者がコンピュータに実行させると、コンピュータは処理系の決めたルールに従って結果(出力)を受講者に提示する。
- **Phase2** 受講者がソースコードと出力結果を見て、そこに含まれるルールを推察する。
- **Phase3** 教材がソースコードを空欄のまま、結果のみ提示する。受講者はPhase2で推察したルールを使用して、空欄部分のソースコードを推察する。
- **Phase4** 受講者は推察したソースコードを再びコンピュータに実行させ、結果が提示されたものと同じであれば、推察したルールが正しいという確信を持つ。結果が同じでなければ、コンピュータからの出力(エラーメッセージや提示されたものとは異なる出力結果)を見て、修正を加える。
- **Phase1'** 教材は別のソースコードを提示し、以下同様に繰り返す。

この流れは、論理的推論の三要素である「演繹」「帰納」「仮説形成」をプログラミングに適用したものになってい

表 1 授業内容

回	題目	学習内容
第 01 回	ガイダンス	プログラミング演習・プログラミング序論の受け方
第 02 回	アルゴリズムを学ぶ	「出力」と「順次」と「コンパイルエラー」
第 03 回	アルゴリズムを学ぶ	「反復」(1 回目)と「演算」
第 04 回	アルゴリズムを学ぶ	「変数」
第 05 回	アルゴリズムを学ぶ	「反復」(2 回目)
第 06 回	アルゴリズムを学ぶ	「分岐」
第 07 回	アルゴリズムを学ぶ	「関数」
第 08 回	C 言語を学ぶ	「配列」
第 09 回	C 言語を学ぶ	「UNIX における環境に慣れる」
第 10 回	C 言語を学ぶ	「UNIX における環境に慣れる (2 回目)」
第 11 回	C 言語を学ぶ	「出力」と「コンパイルエラー」と「反復」と「演算」と「変数」
第 12 回	C 言語を学ぶ	「反復」(2 回目)
第 13 回	C 言語を学ぶ	「分岐」
第 14 回	C 言語を学ぶ	「関数」
第 15 回	C 言語を学ぶ	「配列」

る。ソースコードを「前提」、実行結果を「事実」、ルールを「理論」と位置付けると、上の流れは次の3つを順番に行っているものといえる [6].

- 演繹：前提と理論から事実を導き出す
- 帰納：前提と事実から理論を推測する
- 仮説形成：理論と事実から前提を推測する

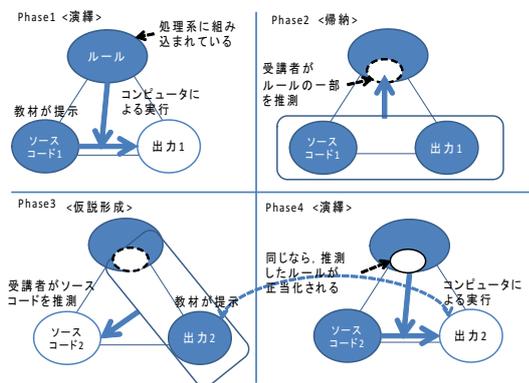


図 1 教材の流れ

このうち、「演繹」にあたる部分、すなわちソースコードを実行して実行結果を得る部分はコンピュータが自動的に実行してくれる部分であり、受講者は「帰納」「仮説形成」を行い、仮設の正しさをコンピュータによる演繹を利用して検証する、ということを繰り返し行うことで、受講者自身が推測したもの（主観）を正当化（客観的なものに変化）させることができる。この過程は、先述した知識の定義と照らし合わせると、知識を獲得する過程に他ならないといえることができる。コンピュータに推論の一部を肩代わりしてもらおう活動は、言い換えるとコンピュータを他者と見なした一種のグループワークととらえることもできる。これを「コンピュータとの対話」と呼び、本授業で用いた教材はこの対話が自然とできるように設計されている。

本授業の第5回の教材を図2に示す。基本的には「提示したソースコードの実行・実行結果の確認」「実行結果からのソースコードの予測と、ソースコードの記述による検証」を交互に繰り返すが、その過程で図1の各Phaseを1回分行うことになる。この図は0503までのプログラムを提示したが、実際には0529までであり、この流れを1回の授業で何度も繰り返すようにしている。この方式を取り入れた教材は小中学生向けのプログラミング体験教室 [7] でも使用されており、受講者は教授者から詳しい説明を受けなくても、自ら学習を進められることが確認されている [8].

#### 4. 作問システムの要件・実装

本節では、作問活動にあたって開発したシステムについて述べる。作問活動をするあたり、グループによる出題までは紙でもできるが、解答者の割り当てと、正誤判定と作問（クイズの良さ）の判定は、受講者が多数であることも考慮して自動化する必要がある。

また、本授業は作問を行う演習授業はPC教室で行うが、解答を行う講義では一般教室のためPCが使えないという制約がある。そこで、解答についてはスマートフォンからできるようにした。

これらの要件を満たす作問・解答システムを実装した。

教員の管理画面を図3に示す。管理画面では、全受講者のグループ、作問したクイズや解答状況などを確認できる。また、図3中の「進捗設定」を用いて、授業、演習が進むに従って教員側で切り替えを行っていく。段階には次の4段階がある。

- クイズを作る 受講者がクイズの作問およびグループ内で出題するクイズの選定を行うことができる段階。「プログラミング演習」が行われている最中は基本的にこれが選択されている。
- みんなのクイズを確認 「プログラミング序論」が開始

<p>0501 次のコードを書いて、実行してみよう。</p> <pre style="border: 1px solid black; padding: 5px;"> 1 for x in range(1,10): 2     print(x, end=" ") </pre>	<p>Phase1          コンピュータによる実行          Phase2          受講者によるルールの推定</p>
<p>0502 0501 のコードの下に、「range(1,10)」を使うコードを追加して、次が出力されるコードを作ろう。</p> <pre style="border: 1px solid black; padding: 5px;"> 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 </pre> <p>(改行には、括弧の中に記載のない「print()」を使おう)</p>	<p>Phase3          受講者によるソースコードの推定          (コード作成)          Phase4          コンピュータによる、受講者が作成したコードを実行</p>
<p>0503 0502 のコードの中の「繰り返し」に着目し、同じ出力がされるコードを4行で作ろう。          (1行目のタイピングミスに注意しよう!)</p> <pre style="border: 1px solid black; padding: 5px;"> 1 for y in range(3): 2     [?] 3     [?] 4     [?] </pre>	<p>Phase1'          コンピュータによる実行          Phase2'          受講者によるルールの推定          :          :</p>

図 2 教材例 (右側は図 1 の各 Phase との対応)

されるときに教員がこちらの段階に切り替える。受講者は他のグループで作ったクイズを見ることができ、解答はできない。模擬試験を行う場合に用いる。

- 個人でクイズを解く 他のグループが作成したクイズの中からランダムに1問が選択され、解答できる状態になる。
- 答え合わせ 解答したクイズの答え合わせおよび他の解答に対してコメントをつけることができる。また自分のグループで出題したクイズの解答状況を確認できる。

受講者はログインした後、グループの作成または参加を行う。グループの代表メンバー1人がグループの作成を行うと、7桁のグループ番号が生成されるので、他のメンバーは番号を参照してグループに参加する。図4は、グループに参加した直後の画面であり、メンバーの名前と、作問へのリンクが含まれる。

[ホーム](#) 【クイズを作る】

## グループ番号：1793255

メンバー：test | test2 | test3 | [グループを抜ける](#)

[新しいクイズを作る](#)

### 自分たちが作成したクイズ

図 4 受講者のグループ作成画面

クイズの作成画面を図5に示す。問題文と正解を登録することができる。解答はスマートフォンで行うことを想定しており、解答の入力を単純化するため、正解には数値およびスペースのみを入力させるようにしてある。例えば、次のような形式での出題が可能である(選択肢は問題文に記載する)。

- (単一・複数) 選択形式 正しいプログラム片や出力結果を選択肢から選択させる、プログラムに関して述べた文章から正しいもの、間違っているものを選ぶ、

## 教員メニュー

### 進捗設定

[最新の情報に更新](#)

- [\\*クイズを作る](#)
- [みんなのクイズを確認](#)
- [個人でクイズを解く](#)
- [答え合わせ](#)

### 一覧

- [グループとメンバーを見る](#)
- [全クイズと解答を見る](#)

### その他

- [設定](#)

[教員連絡用チャット](#)

図 3 教員の管理画面

など

- 並べ替え形式 指定された題意を満たすプログラムを、選択肢のプログラム片を並べ替えることで完成させる。
- 記述形式 プログラムの出力（数値またはその並びのみ）を記述させる。

なお、解答の正誤判定は、解答者が解答した内容がここで設定した正解と同一であるかどうかで自動的に行う（余分なスペース等は除去してから判定する）。また、プログラムを提示して実行結果を予測させる問題において、解答時にはスマートフォンしか使用しないため、問題文に書かれているプログラムを実行することで答えを知られてしまう可能性は低い。

[ホーム](#) 【クイズを作る】

## グループ番号：1793255

メンバー：test | test2 | test3 | [グループを抜ける](#)

### 問題文

次のプログラムの出力を答えよ

```
for i in range(3):  
    print (i*2)
```

### 正解

数値(負数, 小数点つき可)をスペースで区切ったものに限定。

0 2 4

OK

図 5 クイズ作成画面

グループ内のメンバーで作成したクイズは図 6 のように一覧することができる。メンバー間でクイズをレビューし、最終的によいと思ったクイズを選択して出題する。

[ホーム](#) 【クイズを作る】

## グループ番号：1793255

メンバー：test | test2 | test3 | [グループを抜ける](#)

[新しいクイズを作る](#)

### 自分たちが作成したクイズ

[question:1581892](#) - by test2

[question:3904197](#) - by test3 [出題]

[question:6711168](#) - by test

図 6 クイズ選定・出題画面

教員が「みんなのクイズを確認する」という段階を設定すると、受講者は他のグループが作成したクイズをランダムにみることができる。さらに教員が「個人でクイズを解

く」という段階を設定すると、受講者は解答を始めることができる。受講者が解答を始めた時点で、その受講者が解くべきクイズがシステムを割り当て、図 7 のような解答画面が表示される。

このときのクイズの割り当ては次のような手順で行う。これは、どのクイズもほぼ同数の解答者が割り当てられることを考慮して行っている。

- 他のグループが出題した各クイズ（すでに自分に割り当てられたものは除く）について、すでに解答者が割り当てられている人数を集計する。
- 解答者が割り当てられている人数が最小であるクイズ（一般に複数）の中からランダムに1つのクイズを割り当てる。

受講者一人あたりに割り当てられるクイズの数の上限は、教員の管理画面で設定可能である。

[ホーム](#) 【クイズを作る】

## グループ番号：1793255

メンバー：foo | [グループを抜ける](#)

### 問題文

次の出力になるように???に当てはまるものを選択肢から選べ

```
1 3 5 7 9  
for i in range(5):  
    ???  
1: print (i)  
2: print (i*2)  
3: print (i+1)  
4: print (i*2+1)
```

### 正解

数値(負数, 小数点つき可)をスペースで区切ったものに限定。

4

OK

図 7 クイズ解答画面

教員が「答え合わせ」という段階を設定すると、図 8 のように解答したクイズについて自動的に正誤判定を行い、正解であれば「score=1」、正解でなければ「score=0」という表示を行う。

答え合わせの段階中では、自分のグループで出題したクイズではなく、受講者自身が解答したクイズでもないクイズの解答の中からランダムにコメントをすることができる。コメントの内容としては、クイズ自身の評価および、解答に対してのコメント（誤答と判定された解答が別解として認められる場合など）をすることができる。

また、自分のグループで出題したクイズについて、解答の詳細を図 10 のように確認することができる。ここで、正解率 = 正解数/有効解答数(空欄でない解答)の値によって獲得できる星が表示される。

正解率をもとに獲得できる星の数の詳しい計算方法は次の通りになる。

- あるクイズに対する有効解答数を  $n$ 、正解数を  $s$  と

[ホーム](#) 【答え合わせ】

## グループ番号：1793255

メンバー：test | test2 | test3 |

### これまで解いたクイズ

[question:3775781\\_score=1](#)

### 他の人の解答

[他の人の解答にコメントする](#)

[自分が書いたコメントを見る](#)

[自分たちが作成したクイズを確認する](#)

図 8 解答正誤確認画面

[ホーム](#) 【答え合わせ】

## グループ番号：1793255

メンバー：test | test2 | test3 |

### 問題文

次のプログラムの出力を答えよ

```
for i in range(3):
    print(i*2)
```

### 正解

0 2 4

### このクイズに対する解答とコメント一覧

有効解答=3 正解率=2/3 獲得★=3

[獲得★の計算方法...](#)

- [解答=1 2 3 得点=0](#)
- [解答=0 2 4 得点=1](#)
- [解答=0 2 4 得点=1](#)

図 10 クイズの評価画面

[ホーム](#) 【答え合わせ】

## グループ番号：1793255

メンバー：test | test2 | test3 |

[クイズ一覧に戻る](#)

### 問題文

次のプログラムについて、正しく動作するものはどれか選べ

```
1 print("hello"+str(i))
2 print("hello"+int(i))
3 print("hello"+i)
```

### 解答

3

### 正解

1

### 得点

0

出題グループ:group:2187767

解答者:foo

### この解答に対するコメント

日時：2019-10-15 13:10:40 投稿者：test

iに何が入っているかによって正解が変わるのでは？

[コメントを書く...](#)

図 9 コメント画面

する。

- $0/n, 1/n, \dots, n/n$  の各値について「減点幅」を計算する。これらの中で  $2/3$  に最も近い値の減点幅は 0 とし、減点幅が  $d$  であった値の次に  $2/3$  に近い値は減点幅を  $d+1$  とする (同じ近さの値は同じ減点幅とする)。
- このクイズを出題したグループの人数 (通常は 3) か

表 2 作成・出題されたクイズの数

回	作成数	出題数	作成数/出題数
1	162	47	3.4
2	147	36	4.1
3	137	49	2.8
4	133	48	2.8
5	130	49	2.7
6	135	49	2.8
7	126	49	2.6
8	131	49	2.7
9	121	49	2.5
10	113	47	2.4
11	125	48	2.6
12	125	50	2.5
13	116	45	2.6
14	117	48	2.4
計	1818	663	2.7

ら、 $s/n$  の減点幅を減らした値が、このグループに与えられる星の数になる (マイナスなる場合は 0)。

## 5. 作成されたクイズ

この節では、システムにより作成されたクイズの個数、内容などについて述べる。なお、1 回目はシステムを使う練習を行ったため、クイズの内容はプログラミングと関係ないものであった。

各回の受講者によるクイズの作成された数、および出題された数を表 2 に示す。この表によれば、毎回 50 問程度が出題されたことになる。作成数/出題数は最初のうちは 2.7 程度、回を追うごとに 2.4 くらいまで低下するが、おおむね各グループの 3 人のメンバーは 1 問ずつの作問に参加していたと考えられる。

表 3 獲得した星の個数別, クイズの件数

回	0 個	1 個	2 個	3 個	合計
1	16	19	8	3	46
2	17	4	5	10	36
3	20	8	11	10	49
4	18	9	9	12	48
5	20	9	16	4	49
6	20	8	11	10	49
7	31	3	5	10	49
8	23	12	8	6	49
9	39	3	4	3	49
10	33	7	4	3	47
11	27	10	6	5	48
12	37	4	4	5	50
13	27	12	2	4	45
14	29	3	9	7	48
計	357	111	102	92	662

次に, 出題されたクイズについて, そのクイズを作成したグループが獲得した星の個数を表 3 に示す\*1. 全体を通して, 星を獲得できなかったクイズが 357 件, 1 個獲得したクイズが 111 件, 2 個獲得したのは 102 件, 3 個獲得できたのは 92 件であった\*2.

第 7 回以降, 獲得できた星の数が 0 であるものが増え, 3 を獲得できたものが減り始める. これは, 1 人あたりの解答できるクイズの数を増やすように設定しなおしたためである. これについては考察で詳しく述べる.

星 3 個を獲得したクイズの例を図 11 に示す. このクイズは有効解答数 10, 正解数 7 であったが, 多くのクイズがこのクイズのような単一選択形式であったため, 星 3 個を獲得したクイズの中には, 偶然に選んだ選択肢が当たって 2/3 に近い正解率になった可能性もある (記述形式や並び替え形式では偶然当たる可能性は低くなる).

コメントの数を表 4 に示す. 4 回目あたりまでは 1 人当たり 1 個程度のコメントをしていることがわかるが, 回を追うごとにコメントの数は減っていく傾向にあった.

## 6. 考察

今回の授業では, プログラミングのグループワークとして作問の活動を取り入れた. 作成したクイズの質を担保できるように, 正解率をもとに星の数を与えるようにした. この集計を今回作成した作問システムで自動的に行うことにより, 150 人規模の授業であっても, 教員や TA の負担を少なくして運営することができた. さらに, 作問の活動にはグループ内のメンバーが均等に参加でき, 正解率が適正なクイズが多く作成された.

一方, システムや授業の運営にはいくつかの課題が見つ

\*1 有効解答がなかったクイズがあるため, 合計が表 2 と異なる.

\*2 表 3 は, グループのメンバーがすべて 3 人であると仮定して計算しているが, 実際にはメンバーが 3 人未満のグループも存在したため, 獲得できた星の数はより少ない可能性がある.

以下のコードを実行したとき  
 最も多く表示される記号はどれか答えよ

```
def emoji(x):
    print(x,end=" ")

def emoji_table(m,n):
    for y in range(1,n+1):
        for x in range(1,m+1):
            if((y+x)%3==1):
                emoji("★")
            if((y+x)%3==2):
                emoji("○")
            if((y+x)%3==0):
                emoji("◇")
        print()

emoji_table(5,5)
```

1: ★  
 2: ○  
 3: ◇

図 11 星 3 個を獲得したクイズの例

表 4 コメントの数と文字数

回	コメント数	コメントの文字数平均
1	239	17.8
2	195	19.3
3	166	14.4
4	155	14.7
5	113	13.3
6	133	15.4
7	183	16.4
8	123	11.2
9	95	15.4
10	97	13.5
11	82	12.6
12	69	15.2
13	56	12.3
14	47	12.5
総計	1753	15.3

かった.

まず, システムには作成されたクイズの正解率およびそこからもたらされる星の数の計算機能はあるが, 各受講者やグループがこれまでいくつの星を獲得したかを集計する機能が未実装であったため, 別途 Google Form を用いて, 受講者の申請ベースで星の集計した. このため, 獲得すべき星の個数と実際に獲得した個数がずれている可能性がある. 今後, システムに集計機能を追加する必要がある.

また, 当初は毎回 1 人の受講者が 1 つのクイズを解くこ

とを想定しており、出題されたクイズはグループの数、すなわち受講者数の1/3であるので、クイズ1問あたりの解答者は3人である予定であった。しかし、1つのクイズだけを解答させると、解答をすぐ終わってしまう受講者が多くなり、もっとたくさんのクイズを解かせたほうがよい、という意見が担当教員から出たため、7回目以降は1人4題まで解けるように変更した。先述した星を与えるアルゴリズムは、解答者が増えるほどもらえる星の数が減る傾向にある。例えば正解数/有効解答数が1/3の場合と、2/6の場合を比較すると、両者正解率は同じであるにもかかわらず、前者は星が2もらえるのに対して、後者は星が1しかもらえないことになる。このため、獲得できる星が以前より減ってしまった。

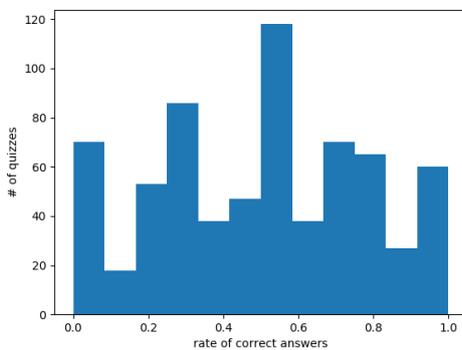


図 12 出題されたクイズの正解率の分布

図 12 に、出題されたクイズと正解率の分布を示す。正解率が極端なもの(0 や 1) はあまり多くはなく、多くのクイズは「誰かは正解し、誰かは間違えている」クイズであったため、難易度は適正なクイズが多かったことが伺える。今後、正解率が同じであれば獲得できる星は同じになるようなアルゴリズムに改定すべきと考えられる。

また、クイズの解答形式が単一選択に偏っているのを、正解を入力する UI を改善して、並び替えや穴埋め式のクイズを簡単に作成できるようにする必要がある。

さらに、クイズの評価に関するコメントによるやり取りが回を追うごとに減ってしまっている。その理由として考えられるのは、コメントによるクイズの改善をする活動がなかったことや、コメント自身のよさを星に反映させる仕組みがなかったことなどが挙げられる。コメントを行う作業は、SECI モデルの表出化の重要な過程であるため、今後は作問に限らず、プログラムを書いている段階から気づいたことを書いたり、共有したりする環境を構築する予定である (BitArrow に機能追加して実装する予定)。

## 7. 関連研究

作問活動は、高度に知的な活動であり、リハーサル・推敲を伴う認知的発達に有益であると言われている [9]。

一方、受講者による作問活動では問題の質を担保することが課題となっており [10]、すでに提案されている既存の作問システム [10] [11] では、教員や他の受講者とのインタラクションを通じて、問題を改善することを支援するようなコメント機能などを活用している。今回開発した作問システムでは、このようなコメント機能に加えて問題の質をある程度自動的に判定する仕組みとして、正答率を利用した自動的なスコア (星) の付与を取り入れた。

また、既存の作問システムでは出題形式としては選択式、記述式などが提案されているが、記述式においては採点を他の受講者に採点の役割を担わせている [12]。この方式では、採点者が正しい採点ができなかった場合、正答率が正しくなくなるため、先述したよう正答率による質の評価に影響が出る可能性がある。本システムでは数値 (の並び) 入力に限定した回答方式を採用したことで、作問できる問題の種類に幅を持たせつつ、スマートフォンなどの入力が難しいデバイスからも解答でき、かつ自動的に採点ができる仕組みを実現した。

さらに、既存の作問システムでは授業時間外での利用ができるよう、すべての機能を自由に利用できるにしている [10] が、本システムでは、「クイズを作る」「個人でクイズを解く」などの段階を教員が設定して、それに対応した機能だけが使えるようにしている。これは、SECI モデルの各段階に対応させた活動を、受講者や教員が意識して行えるようにすることを狙いとしている。

## 8. まとめ

本発表では、今年度から本学で新しく開始された初年次プログラミング授業の概要、およびそこで使用された教材と作問システムについて報告した。

教材は、コンピュータと受講者が「演繹」「帰納」「仮説形成」などの推論を行う活動である「コンピュータとの対話」を進めるように設計されており、作問システムは、受講者同士のグループにより作問と選定 (出題) を行い、出題されたクイズに対する解答者を自動的に割り振ることができる。また、クイズのよさを正答率を用いて自動的に判定し、それを「星」という形で受講者に還元することでクイズの質を担保できるようにした。システムを運用することで、学習者間で自主的に学べるようになり、150 人規模のクラスでの運営に耐える環境であることが確認された。

謝辞 本研究は JSPS 科研費 19K03153 の助成を受けたものです。

## 参考文献

- [1] 野中郁次郎, 竹内弘高: 知識創造企業, 東洋経済新報社 (1996).
- [2] ギノ株式会社: paiza.io, <https://paiza.io/>. 2019 年 5 月 21 日アクセス.

- [3] 大阪電気通信大学, 東京農工大学, 明星大学: Bit Arrow, <https://bitarrow.eplang.jp/>. 2019年5月21日アクセス.
- [4] 長 慎也, 長島和平, 堀越将之, 兼宗 進, 並木美太郎: オンラインプログラミング環境 Bit Arrow を用いた C 言語プログラミングの授業実践, 情報処理学会情報教育シンポジウム論文集, Vol. 2017, No. 17, pp. 121–128 (2017).
- [5] 慎也 長, 和平長島, 広樹間辺, 進 兼宗, 美太郎並木: オンラインプログラミング環境 Bit Arrow における Python 処理系, 情報処理学会情報教育シンポジウム論文集, Vol. 2019, pp. 122–129 (2019).
- [6] 山中脩也, 北島茂樹, 長 慎也, 今野貴之, 武富拓也: 知識創造コミュニティでのプログラミングの知のダイナミクス, 情報処理学会夏のプログラミング・シンポジウム 2019 (2019).
- [7] 明星大学 COPERU プロジェクト: プログラミングパーク”CABA”, <http://coperu.net/caba>. 2019年10月16日アクセス.
- [8] 喜田綾芽, 山中脩也, 北島茂樹, 長 慎也: プログラミングの学びにおける暗黙知を表出させる「コンピュータとの対話」の実践, 情報処理学会夏のプログラミング・シンポジウム 2019 (2019).
- [9] 横山琢郎, 平嶋 宗, 岡本真彦, 竹内 章: 統合レベルでの作問を支援する学習環境の設計・開発と小学校低学年での学習効果, 人工知能学会全国大会論文集, Vol. JSAI05, pp. 102–102 (オンライン), DOI: 10.11517/pjsai.JSAI05.0.102.0 (2005).
- [10] 平井佑樹, 樋山淳雄: 作問に基づく協調学習支援システムとその分散非同期学習環境への適用, 情報処理学会論文誌, Vol. 49, No. 10, pp. 3341–3353 (2008).
- [11] 高木正則, 田中 充, 勅使河原可海: 学生による問題作成およびその相互評価を可能とする協調学習型 WBT システム, 情報処理学会論文誌, Vol. 48, No. 3, pp. 1532–1545 (2007).
- [12] 菅原典子, 織田恵太, 赤池英夫, 角田博保: 集合教育に用いる即応型 e-ラーニングシステム SHoes における組織学習支援, 情報処理学会論文誌, Vol. 48, No. 8, pp. 2791–2801 (2007).