

# プログラミング教育必修化世代を対象とした プログラミング理解度判定テストの構築

勝間田 仁<sup>†</sup> 加藤 利康<sup>‡</sup> 中村 一博<sup>†</sup>

日本工業大学先進工学部情報メディア工学科<sup>†</sup>

日本工業大学共通教育学群<sup>‡</sup>

## 1. はじめに

2020年度から順次施行される文部科学省が定める学習指導要領では、小学校から高等学校の各段階でプログラミング教育が必修化される[1]. 大学でのプログラミング教育においても、必修化されたプログラミング教育を受けた世代に対して教育について検討する必要がある。

著者らが所属する大学の情報系学科では、プログラミングの学習経験と論理的な思考を確認する小テストを実施することで、大学1年次に行うプログラミングの授業のクラス編成を行っている。プログラミング教育必修化世代を対象とした場合、高校までのプログラミングに関する知識や技術の習熟度合をより細かく把握することで、学生に適したクラス編成や授業の実践が可能となり、プログラミング学習の効果が期待できる。

本研究では、高校の「情報の科学」の教科書を対象に、プログラミングに関する学習項目を調査し、その結果をもとに試案したプログラミング理解度判定テストの検証結果について示す。

## 2. 高校のプログラミング学習内容の調査

高校では、2022年度から現行の情報科目「社会と情報」、「情報の科学」の学習内容を再編し、「情報I」が共通必修履修科目として新設される。現状では、「情報I」の教科書は刊行されていないため、アルゴリズムとプログラムに関連した学習内容を含む「情報の科学」について調査を行った。表1に、「情報の科学」を刊行する出版社5社の教科書6冊を対象とした調査結果を示す。調査結果から、「入出力（変数）」、「演算（式）」、「順次構造」、「分岐構造」、「反復構造」、「配列」、「反復構造（入れ子）」、「基本アルゴリズム」をプログラミングの理解度を把握するための学習項目とし、それぞれの理解度の指針を表2の通り定めた。

Development of guidelines and quizzes for measurement of programming ability in programming-education compulsory age

<sup>†</sup>Masashi Katsumata, <sup>‡</sup>Toshiyasu Kato, <sup>†</sup>Kazuhiro Namura · Nippon Institute of Technology

## 3. プログラミング理解度判定テストの試案

2章で選定した8つの学習項目ごとに理解度を把握するための問題を作成した。作問した問題は、Moodleの小テスト機能を利用して提示することにし、自動採点を行うため多肢選択式で問う形式とした。問題作成では、フローチャートを利用してプログラムの流れを表現し、プログラミング言語に依存しない形式で問題を作成するようにした。教科書を対象とした学習内容調査で、教科書Fがフローチャートを使用していなかったが、他の全教科書がフローチャートを使っていたため、フローチャートを利用することにした。

作問した問題は、全20問であり、「入出力（変数）」、「演算（式）」、「順次構造」、「分岐構造」、「反復構造」、「配列」の学習項目は、各3問の18問、「反復構造（二重ループ）」、「基本アルゴリズム」は各1問とした。

## 4. 理解度判定テストの検証

### 4.1 理解度判定テストの実施

理解度判定テストの構築は、今回が最初の試みであるため、問題の難易度、試験に要する時間、テスト結果の活用について検証するため、大学1年生13名を対象に理解度判定テストを実施した。13名のうち、10名は情報系の学生、3名は非情報系の学生である。情報系の学生は、1年春学期のプログラミングの必修科目を履修済みで、現在（秋学期）もプログラミング必修科目を履修中である。一方、非情報系の学生は、システム系2名、電気工学系1名である。現在、システム系の学生はプログラミングの科目を履修している。電気工学系の学生は大学でのプログラミングの科目は履修していないが、高校在学時にプログラミングの学習経験がある。理解度判定テストの実施に際して、各自のノートPCを利用することとし、メモ用紙へのメモ書きを可能とした。テストの時間制限は30分に設定した。

表1. 教科書「情報の科学」におけるプログラミング学習内容の調査結果

教科書	該当ページ数	言語	アルゴリズム	フローチャート	順次・分岐・反復	変数	配列	関数	基本アルゴリズム
A	12	VBA, ドリトル	○	○	○	○	-	-	選択ソート
B	19	JavaScript	○	○	○	○	○	○	バブルソート
C	25	JavaScript, VBA	○	○	○	○	○	-	逐次探索 二分探索 選択ソート
D	14	VBA	○	○	○	○	○	-	逐次探索 二分探索 バブルソート
E	8	VBA	○	○	○	○	○	-	逐次探索
F	6	VBA	○	-	-	-	-	-	逐次探索 二分探索 バブルソート

\*平成28年度に検定済の教科書を対象に調査した。 \*\*教科書DとEは同一の出版社から刊行された教科書である。

表2. 学習項目に対する理解度の指針

学習項目	学習項目に対する理解度の指針
入出力(変数)	変数への値の代入, 変数の値の出力が理解できる
演算(式)	変数を使った式が理解でき, 式の値を変数に代入できる
順次構造	プログラムを手続きの順番に理解できる
分岐構造	与えられた条件による分岐処理の手順を理解できる
反復構造	繰り返す条件のもとで行う処理の手順を理解できる
配列	同じデータ型の値を配列として扱うことが理解でき, 反復構造の処理で配列を利用することができる
反復構造(二重ループ)	二重ループの反復構造で行う処理の手順が理解できる
基本アルゴリズム	基本アルゴリズム(ソート, 探索)の特徴を理解し説明できる

#### 4.2 理解度判定テストの結果と考察

表3に理解度判定テストの各問題に対する情報系, 非情報系の学生の正答率を示す。情報系の学生の平均正答数は18問, 非情報系の学生の平均正答数は14問となった。平均の解答時間は, 21分30秒となった。一部の問題を除いて, 情報系の学生の正答率が非情報系の学生の正答率より高い結果となった。情報系学生は, 理解度判定テストの範囲について既に学習済みであり, 学習項目の理解度を把握することを目的とした問題の難易度としては, 妥当な結果と判断できる。情報系と非情報系の学生とでプログラミング学習歴の差があるが, フローチャートで表現された問題内容を理解できることが確認できた。また, 反復構造について問う問題14, 15, 19において, 情報系の学生の正答率が非情報系の学生より低い結果となった。解答結果から, 情報系の学生の同一の学生がこの3問について誤答しており, この学生が反復構造に関する知識が不十分という判断をすることができる。

次に, 理解度判定テストの終了後に行ったアンケートで得られた主な意見を示す。

- ・目立って難しい問題はなかったと思う。
- ・学習項目の理解度は判別できると思うが, どれほど理解しているかを調べるには難しい問題があっても良い。
- ・問題を解き終わった後の見直しの表示を見や

表3. プログラミング理解度判定テストの結果

問題種別	学習内容カテゴリ	問題番号	正答率	
			情報系学科	非情報系学科
基本	入出力(変数)	1	100%	100%
		2	100%	100%
		3	100%	33%
	演算(式)	4	85%	83%
		5	85%	33%
		6	85%	33%
	順次構造	7	100%	67%
		8	100%	100%
		9	80%	67%
	分岐構造	10	90%	67%
		11	100%	100%
		12	90%	67%
	反復構造	13	70%	33%
		14	90%	100%
		15	90%	100%
	配列	16	80%	67%
		17	100%	67%
		18	90%	67%
応用	反復構造(二重ループ)	19	70%	100%
	アルゴリズム	20	73%	44%

すくして欲しい。

- ・問題内容の用語の統一をして欲しい(ループ処理)

#### 5. おわりに

必修化されるプログラミング教育を受けた大学生のためのプログラミング理解度判定テストの試案について示した。今後は, 理解度判定テストの改良と理解度判定テスト結果の他の授業への活用についても検討していく。

#### 参考文献

- [1] 兼宗進: 学校教育でのプログラミング必修化と情報専門家への期待, 情報処理学会論文誌教育とコンピュータ, Vol15, No.2 pp.9-16(2019).