

プログラミング学習尺度（試行版）の作成

富永敦子^{*1}, 内山芳紀^{*1}

^{*1} 公立はこだて未来大学

Development of a Scale for Programming Learning

Atsuko Tominaga^{*1}, Yoshiki Uchiyama^{*1}

^{*1} Future University Hakodate

本研究では、大学生がプログラミングを学習する際にどのような能力が必要とされているかを測定するための質問紙を作成した。まず自由記述アンケートの回答データを KJ 法により分析し、その結果をもとに 46 項目の質問紙を作成・実施した。探索的因子分析および確認的因子分析の結果、2 つの下位尺度（知識・スキル下位尺度、学習行動下位尺度）からなるプログラミング学習尺度（試行版）を作成した。知識・スキル下位尺度は「構想・設計」「文法知識」「エラーメッセージ理解」の 3 因子、学習行動下位尺度は「積極性」「友人リソース」「Web リソース」の 3 因子からなる。本尺度の信頼性および妥当性を α 係数、確認的因子分析の適合度指標により検討し、概ね妥当であると判断した。

キーワード: プログラミング学習, 質問紙, 尺度開発

1. はじめに

プログラミング教育では、プログラム言語の文法やプログラム書法を理解する能力、アルゴリズムを組み立てる能力の育成が重視されている⁽¹⁾⁽²⁾。しかしながら、初学者の中には、初めて目にするプログラム言語の文法や書法に戸惑い、十分に理解できず、アルゴリズムを組み立てるところまで行き着かない者も多い。大学生初学者の実習における誤りを分析したところ、スペルミスなどの単純な誤り（51.6%）、文法の誤り（23.3%）、プログラムの組み立て方の誤り（25.2%）の 3 種類が抽出された。単純なスペルミスでも、出力されたエラーメッセージの意味がわからないため、解決できないというケースもあった⁽³⁾。また、授業の課題として作成されたプログラムは正常に動作しているため、内容を理解しているように見えるが、実はサンプルコードの寄せ集めで、作成者自身は十分に理解していない場合も指摘されている⁽⁴⁾。プログラミング学習者が何をどのくらい理解しているのかを把握しにくくなっている。

一方、学習全般においては、知識やスキルの習得だけでなく、それらを習得するための方法や、学習する

際の態度も重視されている。例えば、Gagne は、学習成果として言語情報、知的技能、認知的方略、態度、運動の 5 つを挙げている（学習成果の 5 分類）⁽⁵⁾。プログラミング学習では、プログラム言語の文法は言語情報、そのプログラム言語を用いてプログラムを組む能力は知的技能に相当する。そして、プログラム言語の文法やプログラムの組み立て方を効果的・効率的に学ぶための方法が認知的方略であり、あきらめずに学習し続ける姿勢は態度に相当する。いずれもプログラミングができるようになるには必要な要因といえる。

そこで、本研究では、プログラミングを学習する際にどのような能力が必要とされているのかについて、知識やスキルだけでなく、学習方略や態度も含めて測定する質問紙を作成・実施し、その信頼性・妥当性を確認することを目的とする。

2. 予備調査

2.1 方法

質問項目を作成するために、まずプログラミング学習に関する予備調査を 2016 年 7 月に行った。対象者は公立 X 大学情報系学部 1 年生 80 人であった。この

大学では、1年生前期からプログラミングの必修科目があり、回答者はすべてプログラミングを学習中である。

質問項目は「1.プログラムを作成したとき、あなたはどのようなところでつまりましたか。または、どのようなところが苦手ですか」「2.プログラムがうまく作成できないとき、または正しく動作しないとき、あなたはどうしますか」「3.プログラムを作成するために必要なこと(知識、スキルなど)は何だと思いますか」「4.プログラムの作成について、あなたはどのように学習しますか」の4項目であった。回答方法はすべて自由記述で文字数の制限はせず、大学のLMSを用いて行った。

2.2 結果

有効回答は70人(有効回答率87.5%, 全員1年生)であった。1つの質問に対して複数のトピックが含まれている場合はトピックごとに切片化した。その結果、432トピックが抽出された。これらのトピックをKJ法により分類した結果、7グループ「プログラミングの知識」「プログラム作成時の工夫」「エラーの理解・対処」「プログラミング以外の知識」「高次能力」「学習

方法」「意識」に整理された(表1参照)。

3. 本調査

3.1 方法

質問項目を作成するにあたり、KJ法の結果である7グループを、さらに2つの下位尺度「知識・スキル」と「学習行動」に分類した。

1つ目の「知識・スキル」には、「プログラミングの知識」「プログラム作成時の工夫」「エラーの理解・対処」「プログラミング以外の知識」「高次能力」が分類された。これらのグループのトピックを参考に、プログラムを組む際に必要となる知識やスキルに関する質問項目26項目を作成した。

2つ目の「学習行動」には、「学習方法」「意識」が分類された。これらのグループには、プログラミングを学習する際の方法や、プログラミング学習にどのような意識で取り組むかを具体的に記したトピックが多く含まれていた。学習者にとってプログラミングに関する知識やスキルは当然重要であるが、なかなか理解できない内容、あるいは達成できない課題に取り組むには、効果的な学習方法や学習に対する意識も重要である。そこで、これらのグループのトピックをもとに、

表1 KJ法による自由記述の分析結果

下位尺度	グループ	トピック
知識・スキル	プログラミングの知識	関数の仕組み・使い方がわからない(23), for文・while文の仕組み・使い方がわからない(19), 言語の知識を覚えることが重要(8), 基本的な文法知識が必要(8), アルゴリズムについての知識(8), 配列の仕組み・使い方がわからない(7), 変数や関数を使う場面がわからない(7)
	プログラム作成時の工夫	プログラムを作成する作業順序がわからない(9), プログラムが実行される順序を理解することが重要(7), 他の関数や方法を試してみる(7), 作ろうとしているプログラムに必要な機能を整理する(6), プログラムの構造を紙に書き出す(6), 綺麗なインデントで書く(3), プログラミングに行き詰ったとき一度時間を置く(3)
	エラーの理解・対処	プログラムを少しずつ変更し, エラーの原因を探る(16), エラーの内容を理解する(14), エラーが出る前までプログラムを戻す(14)
	プログラミング以外の知識	数学の公式を理解できること(11), リファレンスやエラー文を読むための英語の知識(6)
	高次能力	発想力・想像力が必要(19), 設計・デザインをする能力が必要(7)
学習行動	学習方法	友人や先輩に助言を求める(71), 参考書などを参考にする(66), さまざまなプログラムの開発を経験することが重要(24), サンプルコードなどを参考にする(22), 演習問題・例題を解く(4), 講義スライドを読み返す(7)
	意識	忍耐力が必要(16), 自分が作りたいプログラムを作る(12), 興味を持つことが大事(7)

()はトピック数, トピック数の多い順に表記

プログラミング学習の方法や意識に関する質問項目
20項目を作成した。

回答は、「知識・スキル」「学習行動」とともに「非常
にあてはまらない、かなりあてはまらない、少しあて

表2 知識・スキル下位尺度 探索的因子分析結果

質問項目	因子1	因子2	因子3	
1 仕様書を読んでプログラムを作成するとき、ソースコードの完成形を想像できる	.919	-.072	-.070	
38 類似のソースコードがなくても、一からソースコードを記述できる	.844	-.045	-.057	
34 何かプログラムを作りたいと思ったとき、作りたいプログラムのアイデアを思いつく	.831	-.064	.001	
27 ソースコードを読むことで、大まかな動作を想像できる	.793	-.080	.023	
8 プログラムが思ったとおりに動作しないとき、別のやり方を思いつく	.757	.116	-.190	
15 順を追って論理的にソースコードを記述できる	.704	.082	.018	
47 他者のソースコードやサンプルコードを説明できる	.666	-.042	-.039	
22 プログラムが正しく動作しないとき、すぐに修正箇所を探し出せる	.477	.084	.103	
4 ソースコードを見直すと、間違いに気づくことができる	.469	.071	.111	
54 機能が似ている命令文の違いを説明できる	.461	.214	.070	
44 仕様書に書かれている文章を理解できる	.461	.023	.177	
35 関数を使うべき場所で正しく使える	.404	.249	-.001	
39 ライブラリ関数の仕組みを説明できる	.386	.110	.175	
45 ライブラリ関数を使うべき場所で正しく使える	.343	.195	.225	
7 プログラミングで使われる数学の公式や理論を説明できる	-.235	.860	-.092	
28 関数の仕組みを説明できる	.182	.628	-.062	
21 数学の公式や理論をプログラムで表現できる	.045	.622	.133	
9 命令文の仕組みを説明できる	.187	.550	-.054	
16 配列の仕組みを説明できる	.194	.545	-.073	
2 「文末に『:』を記述する」「文字列は『"』、文字は『'』で囲む」などの、プログラミング言語の記述ルールを説明できる	.180	.487	.116	
53 エラーメッセージの英語の内容を理解できる	-.064	-.011	.864	
18 エラーメッセージが表示されたとき、そのエラーメッセージの内容を理解できる	.038	-.091	.851	
	因子寄与	9.54	1.60	1.20
	因子寄与率 (%)	43.38	7.29	5.46
	累積因子寄与率 (%)	43.38	50.67	56.14

表3 学習行動下位尺度 探索的因子分析結果

質問項目	因子1	因子2	因子3	
32 難しいプログラミングにも挑戦する	.837	.096	-.145	
20 プログラミングを積極的に勉強している	.752	.009	.025	
52 プログラミングを学習するとき、自分の作りたいプログラムを作る	.685	-.081	.090	
24 プログラムをうまく作成できないときは、別の方法を試す	.558	-.094	.116	
31 プログラミングを学習するとき、友人や先輩と協力する	.052	1.010	-.040	
25 プログラムが正しく動かないときは、友人や先輩に相談する	-.077	.691	.112	
37 プログラミングを学習するとき、Web上にある情報やリファレンスを利用する	.014	.014	.833	
42 プログラムが正しく動かないとき、Web上にある情報やリファレンスを利用する	.004	.063	.686	
	因子寄与	2.95	1.70	1.24
	因子寄与率 (%)	36.86	21.20	15.45
	累積因子寄与率 (%)	36.86	58.06	73.51

はまらない、どちらでもない、少しあてはまる、かなりあてはまる、非常にあてはまる」の 7 件法とし、1～7 点に得点化することとした。

フェイスシートでは、性別、年齢、学年だけでなく、プログラミングが得意かどうか、質問項目と同じ 7 件法で尋ねた。この得点をプログラミング得意度とする。本調査は、2016 年 12 月、公立 X 大学情報系学部の大学生 200 人を対象に実施した。

3.2 結果

3.2.1 因子の抽出（探索的因子分析）

有効回答者は 132 人（男性 107 人、女性 25 人；1 年生 118 人、2 年生 12 人、3 年生 2 人；平均年齢 18.81；有効回答率 66.00%）であった。

「知識・スキル」下位尺度と「学習行動」下位尺度のそれぞれについて、探索的因子分析（重みなし最小二乗法、プロマックス回転）を行い、因子を抽出した。使用ソフトは SPSS23.0（IBM）であった。

その結果、「知識・スキル」下位尺度（26 項目）では、スクリープロットの急落から 3 因子が抽出された。3 因子を仮定し、負荷量が.25 未満の項目および多重負荷の項目を除外しながら因子分析を繰り返した結果、3 因子 22 項目が得られた（表 2 参照）。「学習行動」下位尺度（20 項目）もスクリープロットの急落から 3 因子が抽出された。同様の方法で因子分析を行った結果、3 因子 8 項目が抽出された（表 3 参照）。

3.2.2 因子構造の適合性（確認的因子分析）

探索的因子分析で抽出した因子の因子構造の適合

表 4 知識・スキル下位尺度 確認的因子分析結果

因子	質問項目	標準化係数	R ²
構想・設計	1 仕様書を読んでプログラムを作成するとき、ソースコードの完成形を想像できる	0.82	0.67
	34 何かプログラムを作りたいと思ったとき、作りたいプログラムのアイデアを思いつく	0.80	0.64
	38 類似のソースコードがなくても、一からソースコードを記述できる	0.79	0.63
	27 ソースコードを読むことで、大まかな動作を想像できる	0.76	0.57
	8 プログラムが思ったとおりに動作しないとき、別のやり方を思いつく	0.75	0.57
	15 順を追って論理的にソースコードを記述できる	0.75	0.56
	47 他者のソースコードやサンプルコードを説明できる	0.64	0.40
文法知識	28 関数の仕組みを説明できる	0.76	0.57
	2 「文末に『:』を記述する」「文字列は『"』、文字は『'』で囲む」などの、プログラミング言語の記述ルールを説明できる	0.71	0.50
	16 配列の仕組みを説明できる	0.70	0.49
	9 命令文の仕組みを説明できる	0.68	0.46
	21 数学の公式や理論をプログラムで表現できる	0.66	0.43
エラーメッセージ理解	18 エラーメッセージが表示されたとき、そのエラーメッセージの内容を理解できる	0.88	0.78
	53 エラーメッセージの英語の内容を理解できる	0.82	0.68

構想・設計 $\alpha=0.904$, 文法知識 $\alpha=0.826$, エラーメッセージ理解 $\alpha=0.841$
 カイ 2 乗= 92.44, 自由度= 74, GFI=0.907, AGFI=0.868, CFI=0.980, RMSEA=0.044

表 5 学習行動下位尺度 確認的因子分析結果

因子	質問項目	標準化係数	R ²
積極性	20 プログラミングを積極的に勉強している	0.77	0.59
	52 プログラミングを学習するとき、自分の作りたいプログラムを作る	0.75	0.56
	32 難しいプログラミングにも挑戦する	0.72	0.52
	24 プログラムをうまく作成できないときは、別の方法を試す	0.64	0.41
友人リソース	25 プログラムが正しく動かないときは、友人や先輩に相談する	0.88	0.77
	31 プログラミングを学習するとき、友人や先輩と協力する	0.80	0.64
Web リソース	37 プログラミングを学習するとき、Web 上にある情報やリファレンスを利用する	0.86	0.74
	42 プログラムが正しく動かないとき、Web 上にある情報やリファレンスを利用する	0.67	0.45

積極性 $\alpha=0.808$, 友人リソース $\alpha=0.823$, Web リソース $\alpha=0.730$
 カイ 2 乗= 31.15, 自由度= 20, GFI=0.946, AGFI=0.903, CFI=0.966, RMSEA=0.065

性を検証するために、「知識・スキル」下位尺度と「学習行動」下位尺度のそれぞれについて確認的因子分析を行った。計算はAmos23.0 (IBM) による。

①「知識・スキル」下位尺度

探索的因子分析の結果、「知識・スキル」下位尺度は、第1因子が14項目、第2因子が6項目、第3因子が2項目であった。項目数に差があったため、第1因子のみ負荷量が特に高い7項目を分析対象とした。3因子15項目を分析した結果、GFI=0.907、AGFI=0.868、CFI=0.980、RMSEA=0.044であった。 α 係数は第1因子0.901、第2因子0.826、第3因子0.841であった(表4参照)。

第1因子(7項目)は「仕様書を読んでプログラムを作成するとき、ソースコードの完成形を想像できる」「何かプログラムを作りたいと思ったとき、作りたいプログラムのアイデアを思いつく」「類似のソースコードがなくても、一からソースコードを記述できる」などから構成された。これらの質問項目は、プログラム全体を考え、設計するスキルを示していることから「構想・設計」と命名した。第2因子(5項目)は「関数の仕組みを説明できる」「配列の仕組みを説明できる」など、プログラミング言語に関する質問項目から構成されたことから「文法知識」と命名した。第3因子は「エラーメッセージが表示されたとき、そのエラーメッセージの内容を理解できる」「エラーメッセージの英語の内容を理解できる」の2項目により構成されたことから「エラーメッセージ理解」と命名した。

②「学習行動」下位尺度

「学習行動」下位尺度は、各因子の項目数に大きな

差がなかったため、3因子8項目を分析した。その結果、GFI=0.946、AGFI=0.903、CFI=0.966、RMSEA=0.065であった。 α 係数は第1因子0.808、第2因子0.823、第3因子0.730であった(表5参照)。

第1因子は「プログラミングを積極的に勉強している」「プログラミングを学習するとき、自分の作りたいプログラムを作る」など4項目から構成された。プログラミング学習に積極的に取り組んでいることを示す文言であるため、「積極性」と命名した。第2因子は「プログラムが正しく動かないときは、友人や先輩に相談する」「プログラミングを学習するとき、友人や先輩と協力する」の2項目から構成された。よって「友人リソース」と命名した。第3因子は「プログラミングを学習するとき、Web上にある情報やリファレンスを利用する」「プログラムが正しく動かないとき、Web上にある情報やリファレンスを利用する」の2項目により構成されたことから「Webリソース」と命名した。

3.2.3 因子間およびプログラミング得意度との相関

「知識・スキル」下位尺度、「学習行動」下位尺度の各因子、およびプログラミング得意度との相関係数は表6のとおりである。相関係数0.7以上の強い正の相関を示したのは、「構想・設計」と「文法知識」($r(132)=.717, p<.01$)、「構想・設計」と「積極性」($r(132)=.830, p<.01$)、「構想・設計」と「プログラミング得意度」($r(132)=.788, p<.01$)、「積極性」と「プログラミング得意度」($r(132)=.707, p<.01$)であった。0.4以上0.7未満の中程度の正の相関を示したのは、「文法知識」と「積極性」($r(132)=.656, p<.01$)、「文法知識」と「プログラミング得意度」($r(132)=.684,$

表6 因子間およびプログラミング得意度との相関

	知識・スキル下位尺度			学習行動下位尺度			プログラミング得意度
	構想・設計	文法知識	エラーメッセージ理解	積極性	友人リソース	Webリソース	
知識・スキル 下位尺度	構想・設計	.717**	.370**	.830**	-.169	.281**	.788**
	文法知識		.331**	.656**	-.099	.270**	.684**
	エラーメッセージ理解			.340**	.042	.132	.275**
学習行動 下位尺度	積極性				-.145	.330**	.707**
	友人リソース					.001	-.185*
	Webリソース						.280**
	プログラミング得意度						

* $p<.05$, ** $p<.01$

$p<.01$)であった。

「友人リソース」以外の因子は、プログラミング得意度と正の相関を示したが、「友人リソース」はプログラミング得意度と弱いながらも負の相関($r(132)=-.185, p<.05$)を示した。

3.2.4 得意度による下位尺度得点の違い

プログラミング得意度の平均は 3.31、標準偏差は 1.55 であった。プログラミング得意度によって、因子の下位尺度得点に差があるかを確認するために、プログラミング得意度の 1~3 点を低群 ($n=73$)、4 点を中群 ($n=23$)、5~7 点を高群 ($n=36$) とし、因子ごと

に 1 要因参加者間分散分析を行った。下位尺度得点は、各因子の下位項目の平均点とした(表 7, 図 1~6 参照)。その結果、1%水準で有意だったのは「構想・設計」($F(2,129)=49.18, p<.01$)、「文法知識」($F(2,129)=31.35, p<.01$)、「エラーメッセージ理解」($F(2,129)=5.49, p<.01$)、「積極性」($F(2,129)=32.89, p<.01$) であった。「Web リソース」は 5%水準で有意差があり ($F(2,129)=4.59, p<.05$)、「友人リソース」は有意傾向を示した ($F(2,129)=2.90, p<.10$)。

各因子の多重比較 (Holm 法) は以下のとおりであった。「構想・設計」「積極性」は、得意度の高群、中

表 7 プログラミング得意度別の下位尺度得点の平均・標準偏差

	知識・スキル下位尺度			学習行動下位尺度		
	構想・設計	文法知識	エラーメッセージ理解	積極性	友人リソース	Web リソース
高群 ($n=36$)	5.07 (0.77)	5.18 (0.84)	4.92 (1.12)	5.14 (1.04)	4.83 (1.49)	5.88 (1.17)
中群 ($n=23$)	4.30 (0.86)	4.70 (0.60)	4.67 (1.32)	4.08 (1.07)	5.63 (1.21)	5.35 (1.35)
低群 ($n=73$)	3.23 (1.01)	3.73 (1.05)	4.08 (1.38)	3.42 (1.01)	5.51 (1.57)	5.12 (1.19)

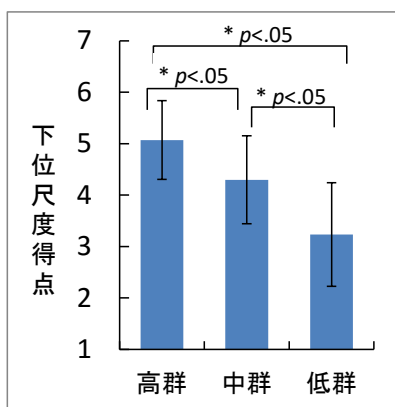


図 1 構想・設計

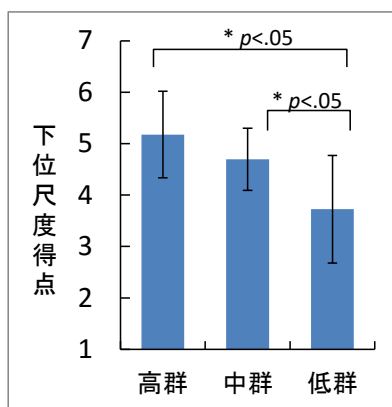


図 2 文法知識

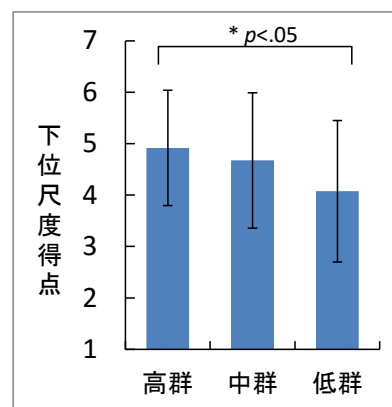


図 3 エラーメッセージ理解

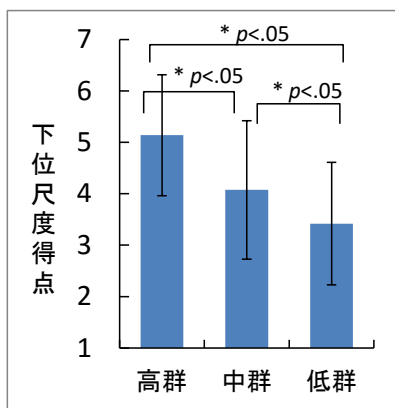


図 4 積極性

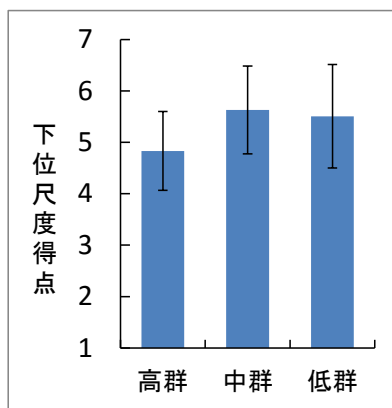


図 5 友人リソース

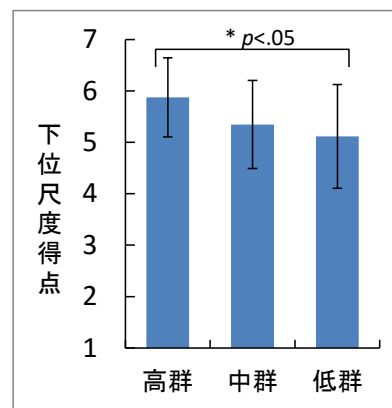


図 6 Web リソース

群, 低群の順に下位尺度得点が有意に高かった ($MSe=0.87, p<.05$; $MSe=1.09, p<.05$). 「文法知識」は, 得意度の高群と中群に有意差はなかったが, 高群・中群は低群よりも有意に高かった ($MSe=0.88, p<.05$). 「エラーメッセージ理解」「Web リソース」は, 高群と中群, 中群と低群には有意差はなかったが, 高群は低群よりも有意に高かった ($MSe=1.73, p<.05$; $MSe=1.51, p<.05$). 「友人リソース」はどの群間にも有意差はなかった.

3.3 考察

3.3.1 プログラミング学習尺度の信頼性および妥当性の検討

α 係数は, 「学習行動」下位尺度の「Web リソース」($\alpha=0.730$) がやや低めであったが, それ以外の「知識・スキル」下位尺度の 3 因子, 「学習行動」下位尺度の 2 因子はいずれも 0.8 以上を示しており, 内的一貫性がみられた.

妥当性は, 確認的因子分析の適合度指標により確認できる. 「知識・スキル」下位尺度の適合度指標 GFI, AGFI, CFI, RMSEA は, 豊田⁶⁾の示す基準値を満たしており, モデルとデータの当てはまりが良く, 妥当なモデルであるといえる.

一方, 「学習行動」下位尺度は, GFI, AGFI, CFI は基準値を十分に満たしたが, RMSEA は 0.065 であり, モデルとして当てはまりが良いとされている 0~0.05 よりやや大きかった. RMSEA はモデルの分布と真のデータの分布との乖離を示す指標であり, 0.10 以上だと当てはまりが悪く, 0.05~0.10 はグレーゾーンとされている⁶⁾. GFI, AGFI, CFI が十分な値であるのに対し, RMSEA がいまひとつである原因としては, 観測変数の少なさが挙げられる⁷⁾. 「学習行動」下位尺度の観測変数, すなわち質問項目は 8 項目であり, 特に第 2 因子「友人リソース」, 第 3 因子「Web リソース」は 2 項目ずつと少ない. GFI, AGFI, CFI は十分な値を示しているため, 概ね妥当なモデルといえるが, 項目数を増やすことにより, モデルをさらに改善できると考えられる.

3.3.2 プログラミング学習尺度による学習者の分析

プログラミング得意度による下位尺度得点の違いを分析した結果によると, 高群は「構想・設計」「文法

知識」「エラーメッセージ理解」「積極性」「Web リソース」が高かった. すなわち, プログラミングを得意とする学習者は, プログラミングに必要な知識・スキルを有し, Web 上の情報やリファレンスを有効に活用しながら積極的に学習を進めていると考えられる.

一方, 中群は「文法知識」「エラーメッセージ理解」「Web リソース」については高群と有意な差はなかったものの, 「構想・設計」「積極性」は高群よりも有意に低かった. このことから, 彼らは関数や配列の仕組みやエラーメッセージの意味などは高群と同じくらい理解しているが, プログラム全体を考え, 設計するスキルには欠けており, プログラミング学習に対する積極性も低いと推測される. 積極性が低いために「構想・設計」のスキルが低いのか, それとも「構想・設計」のスキルが低いためにプログラミングのおもしろさに気づけず, 積極性が高まらないのかは, 今後検討の余地がある.

低群は, 「構想・設計」「文法知識」「エラーメッセージ理解」「積極性」「Web リソース」が高群より有意に低かった. 特に, 「構想・設計」「文法知識」「積極性」は中群よりも有意に低く, プログラミング全般を苦手としていることが因子からもうかがえる. 低群が高群・中群と同じくらい下位尺度得点が高かったのは, 「友人リソース」のみであり, 平均値は 5.51 ($SD=1.57$) であった. 「友人リソース」と他の因子との相関は低いものの負の相関を示していることから, 文法知識に乏しく, プログラムの構想・設計ができず, エラーメッセージも理解できず, Web リソースも活用できない学習者は, 友人や先輩に相談する傾向が高いと考えられる.

友人同士で協力しながら学習することを否定するものではないが, 一方的に答えを教えてもらうだけでは, 知識・スキルを身に付けることは難しく, プログラミング学習に積極的に取り組む姿勢を養うことはできない. 能力に差がある学習者同士による互恵的な学習をどのように進めればよいのかも今後の課題といえる.

4. 結論

本研究では, 大学生がプログラミングを学習する際

にどのような能力を必要としているかを測定するための質問紙「プログラミング学習尺度（試行版）」を作成した。この尺度は、2つの下位尺度「知識・スキル」下位尺度（14項目）、「学習行動」下位尺度（8項目）から構成された。「知識・スキル」下位尺度は「構想・設計」「文法知識」「エラーメッセージ理解」の3因子、「学習行動」下位尺度は「積極性」「友人リソース」「Webリソース」の3因子からなる。本尺度の信頼性および妥当性を α 係数、確認的因子分析の適合度指標により検討し、概ね妥当であると判断した。

本尺度を用い、学習者層を分析したところ、プログラミングを得意とする学習者は、「構想・設計」「文法知識」「エラーメッセージ理解」「積極性」「Webリソース」が高く、順調に学習している様子が見えられた。一方、苦手な学習者はこれらの因子が低く、友人や先輩に相談する傾向が高いことが推測された。

参 考 文 献

- (1) 新開純子, 宮地功: "プログラミング学習支援システムを用いた入門教育の実践", 日本教育工学会論文誌, 33. Suppl, pp.5-8 (2009)
- (2) 平井佑樹, 井上智雄: "ペアプログラミング学習における状態の推定—つまずきの解決の成功と失敗に見られる会話の違い", 情報処理学会論文誌, Vol.53, No.1, pp.72-80 (2012)
- (3) 高本明美, 藤井美知子: "誤り分析にもとづくプログラミング学習の支援", 電子情報通信学会技術研究報告, ET94-32, pp.23-30 (1994)
- (4) 美馬義亮: "公立はこだて未来大学における初年度プログラミング教育", 情報処理, Vol.57, No.4, pp.370-373 (2016)
- (5) Gagne, R. M., Wager, W. W., Golas, K. C., Keller, J. M.: "インストラクショナルデザインの原理", 鈴木克明, 岩崎信 (監訳), 北大路書房, 京都 (2007)
- (6) 豊田秀樹: "共分散構造分析 [Amos 編] —構造方程式モデリング—", 東京図書, 東京 (2007)
- (7) 豊田秀樹: "共分散構造分析 [疑問編] —構造方程式モデリング—", 朝倉書店, 東京 (2003)