

段階的詳細化能力とプログラミング学習の関係に関する研究

The research of relationship between stepwise refinement ability and how to study programming

木村優那^{*1}
Yuna Kimura

矢入郁子^{*2}
Ikuko Yairi

^{*1} 上智大学大学院理工学研究科理工学専攻情報学領域
Graduate School of Science and Engineering, Sophia University, Japan

Learning programming language becomes more and more important today. It seems to be a rising problem that so many people tend to terminate learning when they are prostrated by a hard feeling against programming during an early period of learning. We focused on the hypothesis that the lack of stepwise refinement ability, which is the technique to take an object and move it from a general perspective to a precise level of detail, makes programming language learning difficult. This paper reports the pair work with eight college students who are pure novice programmer, the individual learning with nine college students who has experience of programming and the relationship between stepwise refinement ability and how to study programming.

1. はじめに

近年、スマートフォンやタブレット等の電子機器を使う人が増え、各種アプリケーションの開発等、今まで以上にプログラムに触れる機会が増えている。一方で、現在の学習方法では学習初期に躓き、プログラミング学習自体を諦めてしまう人が多く、プログラミング学習に対して意欲を感じる人が中々増えないという問題がある[1]。特にプログラミングを行う際、問題を細分化し、要求を具体化する「段階的詳細化」が大きな躓きになっており、文法等を理解していても「段階的詳細化能力」の欠如がプログラミング学習の妨げになっているという指摘がある[2]。

我々はこれまで、高校生30名を対象にグループ学習を行い、会話分析を行った結果、段階的詳細化能力の差によって学習法に違いが出る事を確認した[3]。そこで本稿では、高校生の場合に見られた学習法の差が大学生においても存在しうののかを調べるために、プログラミング初心者、経験者それぞれを対象に、段階的詳細化能力、プログラミング、国語力、問題解決能力各々の調査を実施し、各能力の関係について調べた。

2. 段階的詳細化能力とプログラミング学習

詳細化とは、形式手法において抽象的な記述から具体的な実行プログラムへ検証可能な変換を行うことである。特に、段階を踏んで詳細化することを「段階的詳細化」と呼んでいる。

近年のプログラミング学習の問題点として、学習者が処理の流れを考えずにソースコードを作成してしまうことが挙げられている[4]。これを解消するためにはプログラムのふるまいを大まかな機能に分割し、さらにそれを詳細化していく必要がある[5]。特に本研究では、プログラミング初心者を対象としており、「プログラミング入門段階から、問題の分析、段階的詳細化、アルゴリズム作成を重視した指導を行うべきである」という指摘から、段階的詳細化の重要性に着目した[6]。

3. 新調査の設計

3.1 調査の設計

現在、情報教育分野における実験は、様々な条件の網羅性に欠けているとの指摘がある[7]。そこで我々は、高校生を対象

とした調査(以降「前調査」と略記)会話分析によりわかった「段階的詳細化能力の差によってプログラミング学習法に違いが出る」ということが、大学生にも起こり得るのか調査することにした。

まず1つ目は、プログラミング初心者を対象とした実験である。前調査と同じように段階的詳細化能力の差により協調学習を行うが、グループ構成員の人数により学習法に差が出るのか調べるためにペア学習とした。

2つ目は、プログラミング経験者を対象とした実験である。こちらは個人学習とし、学習法の違いについて調査する。

両実験とも前調査同様、段階的詳細化能力測定、プログラミングの調査を行い、更に国語力、問題解決能力の調査を行う。

3.2 調査票の作成

本研究で使用した段階的詳細化能力調査の用紙は、大きく2つに分けられる。以下に詳細を記す。

1つ目の段階的詳細化能力測定調査では、(1)実生活に基づいた題材、(2)実生活での体験をもとに想像力を働かせる題材を扱う。いずれも問題を詳細化することに重きを置いた調査となっている。(1)は全2問で、参加者間で経験による差が出ないように、大学生であれば誰も1度は経験した事がある「自分の部屋の整理整頓・掃除」、「TOEIC等の資格試験対策(受験日直前までの流れ)」を題材とした。解答の形式は、解答の形式は、各目的に対し、

①目的の流れのうち自身の動作を書く

②①の中から更に詳細に記述する必要がある動作を選び、具体的に書く

を記述するものとした。採点の手法は、①では予め準備した模範解答に基づき、特に重要とされる項目を記述出来ているか5点満点の減点法で行い、②では解答者が選んだ動作それぞれについてどのくらい詳細に記述出来ているかの加点法で行った。(2)は全2問で、企業において早急な対処が求められ段階的詳細化が必要とされる場面である「お客さんや取引先とのやり取りに使用しているPCでメールが受信できなくなり、対処(やり取りしている相手、PCへの対処)が必要な場合」、「取引先との急なミーティングが入り、全て自分で会議設定(参加者、日時の設定や場所の確保)をする場合」を題材とした。解答の形式は、各目的に対し、

①ゴールを設定すること

②必要なタスクをしっかりと分解すること

③優先順位をつけること(必要に応じて場合分け)

連絡先: 木村優那, 上智大学大学院理工学研究科理工学専攻情報学領域, yuna_kimura@yairilab.net

を意識して記述するものとした。採点は、①では予め準備した模範解答に基づき減点法で採点を行い、②では必要なタスクが不足なく分解出来ているか加点法で採点を行い、③では②で行ったタスク分解が正しい順序に則り並べ替えられているか加点法で採点を行った。

2つ目の機能分割調査では、様々なシステムにおける機能分割の重要性に着目した[2][4]。誰も1度は使ったことがあるであろう Google Map やスマートフォンの地図アプリケーションの機能分割を行うものとし、プログラミング能力や経験に左右されない設問にした。解答の形式は、Google Map の機能やインターフェース、機能間の関係性について自由記述とした。採点は、機能やインターフェースの名前1つにつき1点、内容が詳細であれば2点、不足があれば1点とし、基本的には1つの機能やインターフェースにつき満点を3点とした。

学力とプログラミングの関係性について様々な意見があるが、数学は現在プログラミングや情報処理能力との相関はないと言われている[8][9]。そこで、今回は国語力のみ調査を行うことにした。

本調査に使用したのは PISA (Programme for International Student Assessment) で、OECD (Organisation for Economic Cooperation and Development, 経済協力開発機構) による国際的な学習到達度に関する調査である。調査は、自記式で行われ、選択式の問題を中心としながらも、全記述式の問題もある。

PISA における読解力とは「自らの目標を達成し、自らの知識と可能性を発達させ、効果的に社会に参加するために、書かれたテキストを理解し、利用し、熟考する能力」とされている。PISA のように読解力をも超えた応用力まで測ることこそ我々が調査したい「国語力」と考え、本実験では PISA を採用した。

PISA の問題で公開されているもののうち、試験時間が30分であること、問題毎の正答率を考慮し、インフルエンザ、プラン・インターナショナル、贈り物、アマンダの4題を採用した。解答方法、採点方法は公開されているものに則り行われた。以下、表1は問題毎の正答率である。

表1 問題毎の正答率

	第1問	第2問	第3問	第4問	第5問	第6問	第7問
チャド湖	77	52.7	48.8	78.6	58.1		
インフルエンザ	85.8	44.1	38.8	41.3	54.2		
落書き	84.5	42.2	71.1	54.7			
労働力	72	53.3	43.2	80.2	65.3		
プラン・インター	10.9(完全正答4.3)						
警察	77	68.8	50.4	82.4			
ランニングシューズ	85.7	84.9	73.4	77.2			
贈り物	65.3	81.4	42.7	83.9	42.6	50.5	34.2
アマンダ	82.8	62.5	63.9	57.5	75.6		
人事部	89.4	66.7					
新ルール	63.1						

プログラミング学習と関連がある能力として、本研究では問題解決能力を取り上げる[4]。様々な場面で求められる問題解決能力だが、いずれも PDCA サイクルの考えを基に問題解決の構想を出来るかが重要である [10]。

そこで、我々は有名技術系企業の採用試験に用いられる問題解決能力測定に着目した。数ある問題の中でも、参加者の知識量に依存する問題は避け、フェルミ推定、数的推理、一般問題、応用問題の4題とした。フェルミ推定は、現在コンサルティング会社や外資系企業の面接試験で用いられることがある他、欧米では学校教育で科学的な思考力を養成するために用いられることもある。数的推理とは、公務員試験等にも出題されている。

フェルミ推定、数的推理、一般問題は各2題の中から1題を選ぶ形式、応用問題は1題のみ出題し、解答は自由記述とした。採点は、フェルミ推定は、あくまで概算であることから、数値を詳

細に定義出来たかを評価対象とした。数的推理、一般問題、応用問題は模範解答に則り採点を行った。

4. ペア学習実験 調査方法

4.1 調査の流れ

参加者は、ペア学習が円滑に進む程度に交流があり、Excel や HTML を簡単に操作出来るがプログラミングは未経験の大学生が適切である[11][12]。本調査は、常々交流がある同一コミュニティ所属の大学生8名が参加し、以下の流れで実施された。

- (1) 段階的詳細化能力測定調査 45分
- (2) 問題解決能力測定調査 30分
- (3) 国語力測定調査 30分
- (4) ペア毎のプログラミング学習 60分
- (5) 到達度確認調査 50分
- (6) アンケートの実施 10分

4.2 各種能力調査結果

表2に各調査の結果を示す。

表2 各調査結果

参加者	学年	国語力	問題解決能力	段階的詳細化
A1(f)	4年	14	20	70
A2(m)	2年	12	21	53
B1(m)	4年	10	16	46
B2(f)	3年	18	11	45
C1(f)	1年	11	8	39
C2(m)	4年	11	18	39
D1(f)	1年	7	5	33
D2(f)	1年	11	9	31

1年生全員が、段階的詳細化能力が低い結果となった。なお、段階的詳細化能力の結果が上位から2人ずつをペア A~D とした。参加者の頭文字はペアを、カッコの中の m は男性を、f は女性を表している。またペア毎に段階的詳細化能力の得点が高かった参加者から 1,2 と番号を振った。

次に国語力調査結果は、1年生が下位という結果になった。

最後に問題解決能力調査の結果を段階的詳細化能力別ペア毎に見ていくと、ペア毎の平均点が段階的詳細化能力の並びと同じになるという結果であった。全ての能力において性別による差は見られなかった。

4.3 プログラミング学習の様子

ペア毎の学習の様子を以下に示す。

ペア A

唯一、章毎に理解をしながら進めたペアである。つまり、2人が該当の章を読み終わると、理解した方が自主的に相方に説明する学習法であった。最初は A1 が理解して説明する場面が多かったが、後半は A2 が説明する場面も多く見受けられた。

ペア A の Ruby の理解度を示す会話の一例に、変数についての会話がある。直方体の3辺を X, Y, Z とし、その表面積や体積を求める問題についてである。以下、抜粋した会話である。

A1「この変数は、X が10、Y が20、Z が30って決めておいて、計算を、文字を使って簡単に書きたいな」

A2「うん」

A1「変数使うと...数字使ってごちゃごちゃになるのを、X の値を変えたい時に、これ全部10のとこ直さなきゃいけないじゃないですか。でもこれ X の母体のとこだけ直せば、自動的に変わってくれるみたい。なら変数使った方がいいねみたい」

60分間の学習の中で、前半の方がやや会話量は多めであったが、全体を通して会話が長く途切れることはなかった。

ペア B

ペア B は、10分間各々読み進めその後話し合うという風に学習を進めていく学習法であったため、徐々に理解をしていく B1 と、理解出来る箇所を早々読み進める B2 とで速さが異なり、差が開く場面が多数見受けられた。理解を深めた B1 が説明することが多く、教材とは別の例を用いて説明することもあった。

全体を通して見ると、雑談は少なく、独学の時間とペア学習の時間のメリハリをつけて、学習に臨んでいた。また、60分全体を見ると、折り返し地点でやや遅れ気味であったことから、学習のペースを上げた後半の方が、会話量が多い結果となった。

ペア C

ペア C は、2人の中でルールを作らず学習が進められたため各々学習しているものの、2人で共有される場面は他ペアに比べ断然少なかった。特に前半は大半が雑談であり、Ruby に関する話し合いは1度も行われなかった。後半も Ruby に関する話し合いは前半よりは行われたものの、やはり雑談が多かった。

一方で、会話量に着目すると、他ペアはほぼ同程度の会話量もしくは少し増える程度に留まったのに対し、ペア B は後半の会話量が前半の倍近いことが特異である。

ペア D

ペア D も、2人の中でルールを作らずに進められたが、難しい箇所は互いに助け合い学習していた。しかし、最初に相当時間を要したためか、唯一、決められた範囲を終えられなかった。

雑談は少なく、前半後半とも同程度の会話量であった一方で、学習を進めていく中で互いに話し合っても「分からない」で終わることも多く、解決に至らずに次章に進む場面も多かった。

4.4 プログラミング到達度確認調査テスト

昨年度の実験に用いたテストと同じ問題を使用した。プログラミング到達度確認テストの結果を表3に示す。

表3 プログラミング到達度確認調査

参加者	暗記問題	構造問題	合計	平均
A1(f)	34	47	81	83
A2(m)	35	50	85	
B1(m)	23	34	57	56
B2(f)	21	34	55	
C1(f)	21	32	53	50
C2(m)	18	29	47	
D1(m)	20	21	41	50
D2(m)	23	36	59	

ペア A が他ペアに差を開けて、1番2人の平均点が高かった。次に平均点が高かったのはペア B である。一方、ペア C とペア D は平均点が全く同じという結果であった。ペア D は、1人が全体で3番目に高得点であるにも関わらず伸び悩む結果となった。

4.5 分析、考察

SPSS を用いて、プログラミング能力、段階的詳細化能力、国語力、問題解決能力の相関関係について調べた。その結果、プログラミング能力と段階的詳細化能力、プログラミング能力と問題解決能力、段階的詳細化能力と問題解決能力について相関が認められたので詳しく述べる。

学習初期段階のプログラミング能力と段階的詳細化能力の間には、高い正の相関が認められた($r = .797, p < .05$)。よって、

我々が着目した「段階的詳細化能力の欠如こそが、プログラミング学習の妨げになっている」という仮説が成り立つ可能性がある。

学習初期段階のプログラミング能力と問題解決能力の間には、高い正の相関が認められた($r = .727, p < .05$)。故に問題解決能力によりプログラミング学習初期段階の学習に差が出る可能性が示された。

段階的詳細化能力と問題解決能力の間には、高い正の相関が認められた($r = .749, p < .05$)。問題解決能力とは一般的な問題にまつわる計画力や分析力の総称であることから段階的詳細化と相通じるところがあり、相関関係が認められると考えられる。

以上から、段階的詳細化能力、プログラミング能力、問題解決能力はいずれも相互に相関があることが分かった。

前調査と同様、プログラミングに関する単語を会話主題、会話主題が出てきた回数を会話主題出現回数、ペア毎に行われた会話の量を会話量、会話主題出現回数を会話量で割ったものを主題出現率と定義し、図1に示す。

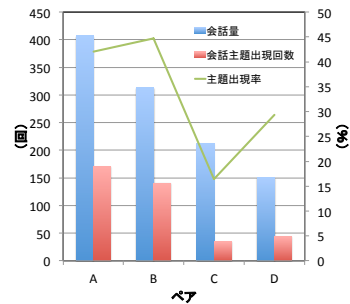


図1 各ペアの会話量、会話主題出現率、会話主題出現率

会話量はペア A が1番多く、続いて B, C, D という結果になった。しかし主題出現率に着目すると、B が1番高く、僅差で A、かなり差が開き D, C と続く結果となった。

5. 個人学習実験 調査方法

5.1 調査の流れ

本調査はプログラミング経験者対象の実験であることから、少なくとも授業でプログラミングに触れたことがある大学生が適切である。情報系学科に所属する大学生、大学院生9名が参加し実施された。参加者9名に関する経歴を表4に示す。

表4 参加者のプログラミングに関する経歴

参加者	年齢	性別	プログラミング歴	プログラミングに触れる頻度	使用したことがある言語
A	21歳	男	3年	授業のみ(今はほとんど触れていない)	C, Java
B	21歳	女	3年	授業のみ(今はほとんど触れていない)	C, Java
C	21歳	女	3年	授業のみ(今はほとんど触れていない)	C, Java
D	21歳	男	4年	毎日	C, C#, Objective-C, Java, JavaScript, PHP, VB
E	23歳	男	5年	今はほとんど触れていない	C, C++, Objective-C, Java, JavaScript
F	24歳	男	14年	週4回くらい	C, C#, C++, Objective-C, Java, JavaScript, VB, Ruby, Peal
G	23歳	男	10年	2年前までは週1~2回だったが、今はほとんど触れていない	C, Objective-C, Java, JavaScript, VB, Ruby
H	24歳	男	6年	週3回	C, C++, Objective-C, Java, JavaScript
I	22歳	女	4年	今はほとんど触れていない	Objective-C, JavaScript, PHP

調査の流れは以下の通りである。

- (1) 段階的詳細化能力測定調査 45分
- (2) 問題解決能力測定調査 30分
- (3) 国語力測定調査 30分
- (4) プログラミング学習 60分
- (5) 到達度確認調査 35分

(6) アンケートの実施 10分

5.2 各種能力調査結果

各種能力調査結果を表5に示す。

表5 各種能力調査結果

参加者	段階的詳細化能力	国語力	問題解決能力	暗記問題	構造問題	合計
A	54	9	15	37	40	77
B	43	9	11	36	42	78
C	37	7	15	32	39	71
D	48	13	7	40	47	87
E	47	11	17	36	22	58
F	43	7	9	45	52	97
G	61	18	12	39	60	99
H	43	11	26	42	60	102
I	50	7	11	42	12	54

段階的詳細化能力調査の結果は、いずれの視点においても大きな差は見られなかった。

国語力調査は、3年生が低く、高得点を上げているのは上級生が多いという結果になった。次に性別という視点で見ると、女子は得点が低く、高得点は男子が多いという結果になった。

問題解決能力調査の結果を年齢、性別という視点で見ると、どちらも大きな差は見られなかった。最後にプログラミング経験年数でいうと、経験年数が特に長い2人が得点が伸び悩んだが、経験年数が短い3人がやや得点が高い結果となった。

プログラミング到達度確認テストは、前調査に用いたテストに、更に暗記問題・構造問題を計4題追加し全20題の調査として実施した。比較的、上級生が高得点をあげ、3年生全員が、得点が伸び悩む結果となった。

5.3 分析、考察

SPSS を用いて、プログラミング能力、段階的詳細化能力、国語力、問題解決能力の相関関係について調べた。その結果、段階的詳細化能力と国語力において強い正の相関が認められた($r = .688, p < .05$)。

次に、学習者の学習法は大きく3つに分類されることがわかった。1つめは、参加者 A,E が該当する「丁寧に読み込むが、ある程度の計画性を持って時間内には終わらせる」学習法である。この学習法の利点は全てを満遍なく学習出来ること、欠点は重要な箇所かけられない可能性があるということである。それ故、この学習法に分類される2人の参加者のプログラミング到達度確認調査の結果が伸び悩んだ可能性が高いと考えられる。2つめは、参加者 B,C,I が該当する「最初から丁寧に読み込む余り、範囲を終えられない」学習法である。利点は、学習した範囲は高得点が見込めるが、欠点は無論学習していない範囲が存在することだ。故に、この学習法に該当する3人のプログラミング到達度確認調査の結果が伸び悩んだと考えられる。3つめは、参加者 D,F,G,H が該当する「最初から最後までをざらりと読み、余った時間で重要な箇所、自分が苦手とする箇所をも再度読む」学習法である。利点は、自分の弱点が強化出来ること、欠点は思わぬ箇所が重要だった場合、対処出来ない可能性があることである。しかし、4人は比較的他の参加者に比べ高得点をあげることが出来た。

以上から、我々が着目した仮説「段階的詳細化能力の欠如こそが、プログラミング学習の妨げになっている」が今回証明出来なかった理由として、メタ知識が考えられる。メタ知識は、知識に関する知識で、「メタ」が「高次の」という意味であることから、通常の認知活動をより高い視点からとらえた認知を意味する。

今回の参加者は情報系学生であることからメタ知識を持っていたことが推察される。特に高得点をあげた参加者は、プログラ

ミングに触れる機会が多く、メタ知識(とりわけ学習法等)を持ち合わせていた可能性が高い。故に、どの箇所をどんな風に学習すればよいか分かっており、効率よく学習に取り組めたと考えられる。

6. 終わりに

本研究において、プログラミング初心者・経験者を対象に段階的詳細化能力、国語力、問題解決能力、プログラミング能力の関係について調査を行った。その結果、高校生、大学生ともに初心者は段階的詳細化能力とプログラミング能力の相関が認められ、かつ段階的詳細化能力別に適した学習法があることがわかった。また、初心者対象実験においてプログラミング能力、問題解決能力、問題解決能力の相関も認められた。なお、国語力とプログラミング能力の相関が認められなかったこと、先行研究において数学力との相関も認められなかったことから、プログラミング能力と学力は関係がないものと分かった。

一方、経験者対象実験の結果、プログラミング能力と段階的詳細化能力の相関が認められず、その原因として今までのプログラミング経験で得たメタ知識である可能性が高いと考えられる。

これらの結果を踏まえ、今後は段階的詳細化能力を伸ばすための手法やシステムの提案、各種能力の相関を踏まえた授業設計の提案等を行っていきたいと考えている。

7. 謝辞

本研究にご協力いただいた皆様に厚く御礼申し上げます。

参考文献

- [1] 岡本雅子,村上正行,吉川直人,喜多一:プログラミング写経型学習過程を対象としたつまずきの分析とテキスト教材の改善 -作業の自立的遂行と作業を介した理解のための支援と工夫-,京都大学高等教育研究第19号(2013).
- [2] 新開純子,炭谷真也:プロセスを重視したプログラミング教育支援システムの開発,日本教育工学会論文誌(2007).
- [3] 木村優那,田村晃弘,矢入郁子:段階的詳細化能力からみた初心者のプログラミング学習の分析(2014).
- [4] 松戸陽子,小山昂紘,原田史子,島川博光:問題文の切り分けによる学習者の機能分割能力の評価,第11回情報科学技術フォーラム(2012).
- [5] 稲葉大祐,原田史子,島川博光:プログラミング教育のための語句選択を用いたプログラミング設計,第8回情報科学技術フォーラム(2009).
- [6] 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(2009).
- [7] 山本三雄,関谷貴之,山口和紀:情報教育における理論と技術に関する研究の調査,情報処理学会論文誌(2009).
- [8] 下郡啓夫,大場みち子:初学習者のプログラミング能力開発のための数学学習,日本教育情報学会(2012).
- [9] 岡本敏雄,松田昇,降矢俊彦:中学生のコンピュータリテラシーと数学の成績との関連性に関する調査研究,日本教育工学雑誌(1995).
- [10] IHARA Masamori:Development of Human Resources Based on Problem Solving Activities(2007).
- [11] 中山晃,高木正則,勅使河原可海:全員参加型の協調学習のための性格を考慮したグループ編成方法の研究,情報処理学会研究報告(2011).
- [12] 西野和典,西端律子,石桁正士:情報教育においてグループ学習を効果的に成立させる形態と条件の検討,日本教育情報学会学会誌(1995).