様　式　C－19、F－19－1、Z－19　（共通）

科学研究費助成事業　　研究成果報告書

令和　３ 年　６ 月 10 日現在

機関番号：１４３０１
研究種目：若手研究(B)
研究期間：2017～2020
課題番号：１７Ｋ１７８２４

研究課題名（和文）プログラミング演習の学習科学的分析と初学者向け学習教材の開発

研究課題名（英文）Development of Learning Materials for Novices and Learning Science Analysis of Learning Processes

研究代表者
岡本　雅子（Okamoto, Masako）

京都大学・高等教育研究開発推進センター・特定講師

研究者番号：５０７３６７８３

交付決定額（研究期間全体）：（直接経費）　　2,700,000円

研究成果の概要（和文）：本研究は、導入段階のプログラミング演習における「理解の促進」と「学習の動機づけの付与」の両立を目的として、学習科学的視点から演習のあり方についてその理論的枠組みを提案するとともに、学習教材やカリキュラムの開発など教育の現場に反映できる程度にまで具体化を試みるものであった。そのために、(1)効果的なプログラミング演習実施のための提案、(2)教材およびカリキュラムの開発を実施し、学校教育等におけるプログラミング演習の１つのモデルケースとして提示した。

研究成果の学術的意義や社会的意義
コンピュータプログラミングの入門教育をめぐっては、様々な研究が試みられている反面、プログラミング演習における「理解の促進」と「学習の動機づけの付与」の両立といった観点においては、ほとんど手つかずの状況にあり、また、体系化された指針も示されていなかった。本研究では、学習科学の視点からプログラミング演習の学習教材およびカリキュラムを検証し、導入過程における演習の実施形態について具体的な指針と事例を示したことにより、字義通り入門教育の実践の場で活用に資する研究として、少なからず日本の学校教育におけるプログラミング学習の効果的実施に貢献するものと研究代表者は考えている。

研究成果の概要（英文）：The purpose of this research was to achieve both "promotion of understanding" and "motivation for learning" in programming courses at the introductory phase. It proposed a theoretical framework for the exercises from a learning science perspective. It attempted to concretize it to the extent that it could be reflected in the field of education, such as the development of learning materials and curriculum. In order to achieve this purpose, (1) proposed effective programming courses, (2) developed teaching materials and curriculum, and presented them as a model case of programming courses in higher education.

研究分野：情報教育

キーワード：教材開発　プログラミング教育　演習　初学者　学習科学

※科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属します。

様　式　C－１９、F－１９－1、Z－１９（共通）

１．研究開始当初の背景

(a)プログラミングの学習環境とプログラミング教育

　コンピュータプログラミング教育は半世紀以上にわたって実施されてきたが、とりわけ高等教育機関で C 言語などの高水準言語（記述の抽象度が高いプログラミング言語）を学習する場合については、1980 年代を跨いで、その学習環境を大きく変化させている。大型計算機を使用する環境では、演習を日常的に実施することは難しく、授業は専らテキスト教材の解説を中心に実施されることが一般的であったが、コンピュータの普及と技術革新が進むにつれ、模倣による実践的練習、すなわちプログラミング演習は、とくに専門教育において、授業の中心的な構成として取り入れられるようになっていった。一方、かつて専門教育として取り扱われていたコンピュータプログラミングは、大学教育では、教養教育などにおいて実施されることが散見されるようになってきたほか、2003 年より高等学校において教科「情報」の導入により選択必修科目となったのに続いて、2012 年から中学校の学習指導要領で技術家庭科に導入されるなど、普通教育として実施されている。このように、（研究開始当初の）プログラミング教育は、幅広い層にその導入過程の教育を実施すること、そして、その実施形態が演習を中心に構成されるように変遷してきており、はじめてプログラミングに触れる機会は、専門教育において目的を達するための手段として学ぶという形態から、プログラミング学習そのものを目的とするように変遷したともいえる。

(b)プログラミング教育改善のための試み

　初学者を対象としたプログラミング学習・教授活動のための支援に関しては、興味を持たせるなどのモチベーション面の改善（王ほか, 2007）を目的とした研究が少なくない。一方で、導入過程における理解そのものの促進を目的とする実践的かつ具体的な取り組みについては、あまり試みられていなかったが、研究代表者自身の取り組みも含め、近年になって報告が増加しており、そうした中で一応の進展が見られるようになった。しかしながら、これらの研究は、それぞれ別の方向で進められており、学習者にとって興味深くわかりやすい教材といった、双方の研究の方向性を両立させるための工夫に係る分析的な試みはほとんど見られない。

２．研究の目的

　プログラミング教育は、現在、普通教育として幅広い層を対象とし、演習を中心に構成された授業として実施されている。演習で用いられるプログラムは、プログラミングを学ぶために設計されたものであって、プログラムそれ自体の目的、例えば、物理学の方程式を解くためであるとか、ウェブサイトを充実させるためであるとかについては、特に考慮されていない。プログラムそれ自体は、何らかの目的を達するための手段であることから、そのプログラムを学ぶことによって何ができるようになるか、そして、それが学習者の興味に沿ったものであるかについては、学習の「動機づけ」として、より重視すべきであると考えており、プログラミングを理解するための分かり易さだけでなく、学習の「動機づけ」にも配慮した教材を開発し検証したい。

　そこで本研究では、導入段階のプログラミング演習における「理解の促進」と「学習の動機づけの付与」の両立を目的として、学習科学的視点から演習のあり方についてその理論的枠組みを提案するとともに、学習教材やカリキュラムの開発など教育の現場に反映できる程度にまで具体化を試みる。

３．研究の方法

　本研究は、導入段階のプログラミング演習における「理解の促進」と「学習の動機づけの付与」の両立を目的として、学習科学的視点から演習のあり方についてその理論的枠組みを提案するとともに、学習教材やカリキュラムの開発など教育の現場に反映できる程度にまで具体化を試みるものである。そのために、(1)効果的なプログラミング演習実施のための提案、(2)教材およびカリキュラムの開発を実施し、学校教育等におけるプログラミング演習の１つのモデルケースとして提示する。

４．研究成果

(1)　初学者向けプログラミング学習教材とカリキュラムの開発（プログラミング演習における帰納的な学習方法の提案）

　コンピュータプログラミング教育における演習は、現在、コードをひとつひとつ学ぶために設計されたサンプルプログラムを入力・実行し、それを確認するという模倣の過程として実施されている（図 1)。こうしたひとつひとつ積み上げながら学ぶ方法（「部分から全体へ」）に対し，最初から完成された

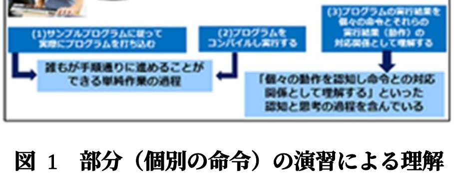

図 1　部分（個別の命令）の演習による理解

サンプルプログラムと実行できる環境を供与して、そのプログラムの特定箇所を改変して、改変箇所の機能を理解するという方法（「全体から部分へ」）もまた、演習の在り方として検討する価値があると考えられる。

これまでのプログラミング教育における演習では、プログラム全体が見えない状態から学習を始めており、学習者は抽象性の高い事例を学ぶため、単純な命令が実行できたというだけでは理解できた実感がわかず、不安なまま演習が進んでいた。加えて、個別のサンプルプログラムを学習するだけでは、学習者は提示された作業を繰り返し実施しているだけで、その時点で何を学んでいるかが理解できずに作業を進めるだけになっていた。これらを改善するために、本研究では、「全体から部分へ」の学習方法について、その有効性について考察するとともに、そうした演習において有効なサンプルプログラムの在り方について検討した。

具体的には、以下の手順で学習者は学習していく。

①意味のあるプログラムの全体像を提示して実行
②一部分ずつ改変して実行
③部分を抜き出した短いプログラムを提示して実行
④部分を「コピペ」して自由に組み合わせオリジナルプログラムを作成

また、使用するサンプルプログラムに求められる要素は以下の通りである。

①学習課題となる内容を過不足なく含んでいること
②実行結果が身近で現象として理解しやすいこと
③学生にとって「意味がある（実用性が高いと思える）」プログラムであること
④サンプルプログラムを改変して実行した際の変化が視覚的に顕在であること

これまで、演繹的な方法と対峙する方法として、帰納的な方法を提案してきたが、演習への導入を考えた場合、どちらか一方を取り入れるという方略のほかに、双方を取り入れることで、それらが、相互に補完的に機能するという場合も想定される。こうしたことから、演習への導入に際しては、それぞれの方法をどの学習課題において、どの程度取り入れるか、あるいは、単独で実施するか双方を取り入れる方がよいのかなど、実践的な側面から検証していく必要がある。

(2) プログラミング演習におけるイメージ図使用の効果について

HTML を学ぶ際、タグと実行結果を同時に例示してその機能を学んでいく方法が一般的である。HTML は、タグがそのまま実行結果として可視化されるため、この方法の有効性は疑うべくもないが、実践上、こうした方法だけでは学習者の理解が進まないケースも散見される。こうしたケースについては、タグと実行結果の具体例を示す方法だけでなく、タグの意味する概念を抽象化したイメージ図も合わせて示すテキストを作成し、それを使用することで改善が可能ではないかと考えた。プログラミング学習におけるサンプルの提示は、具体的であるものの、それゆえに抽象的理解につながらないケースがあることはたびたび指摘されているところであるし、事前に抽象イメージを提示する方法の有効性についても提案されてきた。HTML の学習の場合、タグの支持する具体的内容はそのまま画面上に静止画として顕在化するケースがほとんどであるからして、そうしたイメージ化を必要とするケースは決して多くはないように思われるが、中にはイメージ化が有効な場合も想定される。つまり、異なるタグを用いて同じ画面表示を実現できるケースがあることから考えても、タグの意味する内容によっては顕在化されていないということである。

こうしたことから、本研究では、HTML 演習を対象とし、イメージ図を加えたテキストを用いたケースを事例として紹介するとともに、その際に観察された受講者の反応とその効果について考察した。

イメージ図のないテキストを用いた授業実践では、特にセルの結合を伴う課題において助言を要する受講者が多かっただけでなく、最終的に課題を完成させることができなかった受講生がみられたが、一方、イメージ図を掲載したテキストを用いた授業実践では、受講者のほとんどが独力で課題を完成させたほか、すべての受講者が最終的に課題を完成させている。このように、本授業では、明らかな結果としてイメージ図の効用が確認されている。こうした結果は、イメージ図の効用としてだけではなく、概念を具体的事例として学習するだけでは、不十分であり、抽象化したものとして学習する必要があることも同時に示しているともいえる。

同授業実践は、HTML のごく初歩的な内容を扱ったものであり、学習範囲が限られていたことから、その効用が確認されたのは、ごく一部の学習内容についてであったが、このほかにも、同様の方法が有効なケースがあるものと考えられる。そうしたケースを抽出するとともに、有効なイメージの提示方法についても探っていきたいと考えている。

(3) 「主体的プログラミング学習段階」を対象としたプログラミング演習における学習困難性の考察

プログラミングの導入の授業で到達できる段階は、個々のサンプルプログラムを読み解く基礎が身につく程度である。入門課程の学習の後、自ら目的とするプログラムを設計するには、より実用的なプログラミング能力のトレーニングが必要となる。こうした「応用段階」のプログラミング教育については、授業あるいは演習をどのように実施したらよいかという授業設計のあり方を主な課題として議論されてきたが、一方、学習者側からの視点でどのような困難性がみら

れるかについて議論がなされてこなかった。こうしたことから、本研究では、まず、模倣でなく自らプログラムを構想して記述するプログラミング演習において、学習者が具体的に作業する内容を階層化および明確化し、それらの作業のうち授業内においてどのようなトレーニングが実現できるかについて議論する。そして次に、この議論を踏まえて設計し、実施したプログラミング実践において、学習者にとってどのような困難性がみられたかについて抽出し、その問題点について考察する。

プログラミング初学者は、まず、最初にプログラミングの基本的な要素である変数と制御構造などを学習し、サンプルプログラムを模倣する過程やサンプルプログラムの一部改変などに取り組む。これらは教員から与えられた知識を覚え、また、提示されたプログラムをタイピングして実行し、動作を確認する極めて依存的な学習過程である。この段階を経た学習者は、教授者から使用するコードや目的とするプログラムのイメージやモデルについて参考となる情報を与えられながらも、自らアルゴリズムを考え、それをコーディングするという主体的なプログラミング学習段階へと進む。本研究では、前者を「受動的プログラミング学習段階」、後者を「主体的プログラミング学習段階」と規定した。

「主体的プログラミング学習段階」において授業で提供可能な学習内容を検討し、以下の 4 点の能力向上を目的とした教材とカリキュラムの開発を試みた。

①具体的事象を提示した状況下におけるモデルの構築
②モデルに沿ったアルゴリズムの作成
③使用する複数のコードを予め提示した上で実施するコーディング
④デバッグを主な目的としたテスト

開発した教材とカリキュラムを用いて授業を実施したところ、それぞれの段階におけるつまずき箇所を抽出することができた。その結果、自らプログラムを構想して記述していくことができないという事象であっても、その要因はまちまちであり、各学習階層において別個の困難性として表出していることがわかった。とくにアルゴリズムをチャートとして記述できているにもかかわらずコーディングの際に抜け落ちていた学習者が見られた。こうしたコーディング段階の誤りの原因が、単に当該コードの理解不足であるならば、当該コードを知識として、再度、記憶すればよいのであるが、アルゴリズムをもとにコーディングする際に本実践において明らかにできなかった想定外の要因があるとも考えられ、これについては今後の研究課題としたい。自らプログラムを構想し、それをプログラミング言語に落としていく作業は、上流から下流へとつながる一連の作業であるため、上流のどこでつまずいたとしても、それに続く下流の作業に進むことはできない。

上流から下流までのそれぞれの作業は、どれをとっても自立的にプログラミングする際には必要とされる能力であるが、その能力を涵養しようとする授業あるいは演習において、その一つが欠けるからといって、以降の学習に手を付けられないようでは学習者の不利益が大きいし、授業として成立しない。本実践でも明らかになったように、上流から下流までの各階層につまずき箇所があるのだから、続く階層に移行する際には、前階層の成功が次階層の成功の前提にならないように、つまり、どの階層からでも作業が可能であるように適切にフォローしながら授業を進めていく必要があると考える。

<引用文献>

①王 文涌、池田 満、李 峰栄、プログラミング教育における動機づけ教授方法の提案と評価、日本教育工学会論文誌、31、349-357、2007

５．主な発表論文等

〔雑誌論文〕　計0件

〔学会発表〕　計3件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 岡本雅子
２．発表標題 Webプログラミング演習におけるイメージ図使用の効果について
３．学会等名 情報処理学会第81回全国大会
４．発表年 2019年

１．発表者名 岡本雅子
２．発表標題 プログラミング演習における帰納的な学習方法の提案
３．学会等名 日本教育工学会大会講演論文集
４．発表年 2017年

１．発表者名 岡本雅子
２．発表標題 「主体的プログラミング学習段階」を対象としたプログラミング演習における学習困難性の考察
３．学会等名 大学ICT推進協議会年次大会
４．発表年 2020年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

－

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関