

一般情報教育におけるプログラミングのスキルの習得度と
プログラミングの考え方の理解度の関係に関する検討

吉田典弘^{†1} 和田裕一^{†2} 邑本俊亮^{†2} 堀田龍也^{†2} 篠澤和久^{†2}

概要: 大学の一般情報教育においてプログラミングの授業を実施し、終了時に文法上誤りなくプログラムを書くスキルの確認をするとともに、受講の前後で、学習したプログラミング言語に依存しない形でプログラミングの考え方の理解度を測る評価問題による調査を実施した。プログラミングのスキルの習得を学習目標としたクラスとプログラミングのスキルの習得が学習目標ではないクラス（非プログラミングクラス）といった学習内容の異なる2つの群に関して先の調査の得点を比較したところ、学習内容の違いと受講前後による有意な差があった。その要因を検討するために、プログラミングスキルの習得度とプログラミングの考え方の理解度の相関関係を調べた結果、授業終了時のプログラミングのスキルの得点が、プログラミングの考え方の理解度に影響を与えていることが示唆された。

キーワード：一般情報教育，プログラミングのスキル，プログラミングの考え方

A Study on the Relationship between Acquisition Level of Programming Skills and
Understanding of Programming Thinking in General Information Education

NORIHIRO YOSHIDA^{†1} YUICHI WADA^{†2} TOSHIAKI MURAMOTO^{†2}
TATSUYA HORITA^{†2} KAZUHISA SHINOZAWA^{†2}

Abstract: We conducted programming classes at the general information education, confirmed the skills of program writing without grammatical errors at the end of the course, and explored the degree of understanding of programming thinking without depending on the programming language learned both before and after the course. We conducted a survey based on the evaluation problem to be measured and compared the results with the scores of a previous survey obtained from two classes that had different syllabi, mainly a programming and a non-programming class. A significant difference in learning content was observed between the survey results, both before and after attending the programming classes. To examine the factors causing this difference, we investigated the correlation between the degree of mastery over the programming language and the degree of understanding of the programming thinking. The results indicated that the final score of programming thinking at the end of the course depends on the understanding of programming skills.

Keywords: General Information Education, Programming Skills, Programming Thinking

1. はじめに

大学における一般情報教育では、プログラミング教育を選択科目として配置している例が多い[1].

こうしたプログラミング教育では、プログラミング言語の文法に対して誤りのないようにプログラムを書くというプログラミングスキルの習得が主要な学習目標の一つとされてきた。また、学習の評価は、授業で使用したプログラミング言語を用いた課題や試験によって行われていることが一般的である[2][3].

一方、総務省による「プログラミング人材育成の在り方に関する調査研究」報告書[4]では、プログラミングに関する教育がもたらす効果として、合理性、論理的思考力の向上等が挙げられており、フローチャートやプログラムの構想の作成とそれらに基づきプログラミングするという過程で、俯瞰的に考える、順序立てて考える、仕組みを考える

などの合理的、論理的な思考が必要となるため、プログラミングを学ぶことで論理的な思考力が向上するとされている。文部科学省の「小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）」[5]では、時代を超えて普遍的に求められる力としての「プログラミング的思考」は、「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけばより意図した活動に近づくのか、といったことを論理的に考えていく力」であると定義されている。これらのことから、これからの時代におけるプログラミング教育に求められるのはプログラミングの考え方を理解することであり、それは小学校から大学の一般情報教育まで共通することである。

上述の政策的な動向を踏まえるならば、これからのプログラミング教育では、順序立てて考えることや手順を踏んで問題を解決すること（磯部、静谷ほか[6]）などといったプログラミングの考え方の理解や、それがもたらす効果を

^{†1} 関西学院大学 Kwansai Gakuin University

^{†2} 東北大学大学院 Graduate School, Tohoku University

どのように評価するかが喫緊の課題となると考えられる。これまでの、授業内で使用した特定のプログラミング言語を用いた試験などによってプログラミングのスキルの習得度が評価されてきたが、プログラミング言語を用いないプログラミング教育が様々な方法で行われており、プログラミングの考え方の理解度については、これまでとは異なる評価方法を検討する必要がある。

河村[7]は、学生がプログラミングを学ぶことで、アルゴリズムをデザインするための論理的な思考力の育成や、解決すべき問題をアルゴリズムに置き換えて定式化する抽象化の習得が期待できることを示唆している。従来のプログラミング教育では、河村の指摘するような能力を評価している例は少ないが、関連する研究には以下のようなものが挙げられる。

大岩[8]は、日本語によるプログラミング教育を実践することで、プログラムの考え方の理解が深まり、その後のプログラミング言語によるプログラミングのスキルが高くなるとしている。また、大場ら[9]は、論理的文章作成思考力とプログラミング力に強い相関がみられたことを報告している。しかし、これらの研究においては、プログラミング言語に依存しない形でのプログラムの考え方の理解度については評価がされていない。他方、宮田ら[10]は、Logoによるプログラミング学習によって育成された問題解決能力がプログラミング以外の他の問題解決場面に転移するかという問題を指導方法との関係から分析し、問題解決のプロセスを重視したアプローチでプログラミングを指導した場合に問題解決能力の転移が起りやすいことを示している。よって、ある学習が別の能力に転移するような関係が、プログラミングのスキルの習得度とプログラミングの考え方の理解度との間にも存在する可能性があると考えられる。

本研究の狙いは、上述のような先行研究を踏まえ、プログラミングの考え方の理解度を、学習したプログラミング言語に依存しない形で評価する方法を確立することである。このために、まずはプログラミングのスキルの習得度と、特定のプログラミング言語に依存しないプログラミングの考え方の理解度の関係を把握することを本研究の目的とした。具体的には、一般情報教育としてのプログラミングの授業を実施し、この授業内で使用したプログラミング言語を用いてプログラムを正確に作成できることを最終課題の成績に基づいて評価し、受講後のプログラミングのスキルとして定義した。また、プログラミングの考え方の理解度を測定するために、学習したプログラミング言語に依存しない形で構造化プログラミングの構成要素の理解度を測る評価問題を実施した。これら2種類の成績の関連性、すなわち、プログラミングのスキルの習得度とプログラミングの考え方の理解度との関係を分析した。学習したプログラミング言語に依存しない形で、プログラミングのスキルとプログラミングの考え方の理解度の関係性を示すことがで

ければ、特定のプログラミング教育に依存しない形でプログラミングの考え方の理解度を評価する手法を確立するための一助となると考えられる。

2. 本研究の枠組み

2.1 観測点と分析方法の説明

本研究で実施した授業と、取得したデータの授業回数との対応や分析方法の方針について図1と表1に示す。図1における観測点とは、授業回数における評価時期を意味する。対象となる授業は、プログラミングのスキルの習得を学習目標としたクラス（以下、プログラミングクラス）とプログラミングのスキルの習得が学習目標ではないクラス（以下、非プログラミングクラス）であった。また、両授業の授業において、図中のAからCの3つに観測点を置いて分析を行った。AからCの点は以下の通りである。

- A点 プログラミングクラスにおいて、受講後のプログラミングのスキルの習得度を、授業における最終課題の得点で確認した点。最終課題の得点は、繰返しの利用、アニメーションの実行時間、アニメーションで動かす絵の完成度、アニメーションで動かす絵の数、配列の利用について、各3点とし、15点満点で採点をした。
- B点 プログラミングクラスと非プログラミングクラスにおいて、受講前のプログラミングの考え方の理解度を確認した点
- C点 プログラミングクラスと非プログラミングクラスにおいて、受講後のプログラミングの考え方の理解度を確認した点

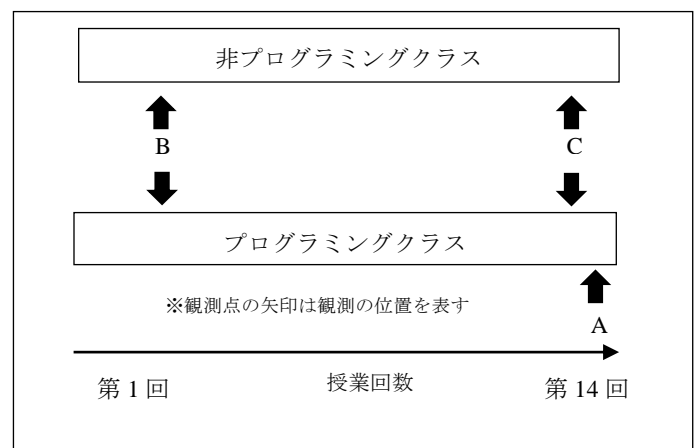


図1 観測点

さらに、これらの観測点での3つの分析方法について説明する。

分析1：プログラミングクラスにおいて、授業で求めるプログラミングのスキルが習得されているかを明

らかにするために、A点における最終課題の得点について分析する。

分析2：プログラミングのスキルの習得のみならず、プログラミングの考え方の理解度が向上しているのかを確かめ、かつ、他の学習内容の授業でもプログラミングの考え方の理解度が向上するのではないかという可能性を明らかにするために、プログラミングクラスと非プログラミングクラスでのB点とC点の評価問題の結果を分析する。

分析3：プログラミングクラスにおいて、受講後のプログラミングのスキルの習得度とプログラミングの考え方の理解度の関係を明らかにするために、A点とC点との相関関係を分析する。

図1の観測点と分析1から分析3の関係を表1に示す。

表1 観測点と分析方法

	図1における観測点	観測対象	使用するデータ
分析1	A点	プログラミングクラス	プログラミングのスキルとしての最終課題の得点
分析2	B点とC点	プログラミングクラス 非プログラミングクラス	プログラミングの考え方の評価問題（事前（B点）、事後（C点））
分析3	A点とC点	プログラミングクラス	プログラミングのスキルとしての最終課題の得点とプログラミングの考え方の評価問題（事後）

2.2 実施した授業

実施した授業内容について以下に述べる。

実験群としたプログラミングクラスは、科目名が「コンピュータ言語（Java言語）」であり、学習目標はアニメーションを作成することで、順次処理、条件処理、繰り返し処理を理解させ、これらに配列を利用することで、プログラムをより効率良く作成できるようになることであった。学習内容として、アニメーションを作成するために必要となるためのことを、Java言語を用いてプログラミング初心者向けとして授業を行った。シラバスを表2-1に示す。授業は総合開発環境「Eclipse」を用いて、プログラムをテキスト入力する形式で進め、順次処理、分岐処理、繰り返し処理を教えた後に、グラフィックスを用い各自で作成した図形を動かすプログラムの作成を最終課題として課した。なお、教科書として書籍[11]を使用した。評価は授業を担当した筆者が、提出されたアニメーションの作品のプログラムのソースコードを5つの観点から確認することで実施した。

統制群とした非プログラミングクラスは、科目名は「コンピュータ基礎」であり、学習目標は、Word, Excel, PowerPointを利用して、他の授業でのレポート作成やゼミ活動、卒業研究で必要となるPCスキルの基礎を身につけることであった。学習内容はMicrosoft Officeの操作に関する初心者向けの内容であり、この授業のシラバスを表2-2に示す。評価としては毎回の授業における課題提出と、最終試験としてWordとExcelによる文書作成を行わせた。この授業ではプログラミングの考え方に関する理解に関する内容、たとえば順次処理、条件処理、繰り返し処理は教えていない。

2.3 プログラミングの考え方の理解度に関する評価問題

前節で示した、学習目標、学習内容、評価が異なる2群に対して、プログラミングの理解度を評価する問題として以下の2問を用いた（付録参照）。この問題を採用した理由は、構造化プログラミングの基本構造である順次、条件分岐、繰り返しの3つの処理を用いて、手続きの順番を並び替え、出題された問題に対して正確な結果を得ることが出来るかを判断させるためである。

問1 順次と条件分岐を問う問題：題材はロボット掃除機の作業（配点4点）

問2 繰り返しを問う問題：題材は1から10の数字を表示（配点6点）

問1は、2017年1月に公表された文部科学省による情報活用能力調査（高等学校）において、「情報の科学的な理解に関する問題」[12]として使用された問題であり、ロボット掃除機の動作を示した要素を置いておき、フローチャートを完成させるものであり、順次と条件分岐について問うものである。文部科学省の調査は高等学校第2学年の4,552名を対象とした正答率が46.2%であること、問題および結果が公表されていることから、この問題を採用した。問1は計4点であるが、順次が2点、条件分岐が2点の配点であった。

問2は、高等学校の共通教科「情報」における「情報の科学」の教科書[13]の中にある例題のプログラムを引用し、それを日本語による表現を用いて自作をした。問題作成においては、プログラミング教育を受けたことがない学生にも問題文の意味が分かりやすい表現となるように配慮し、変数や代入、繰り返しに関する文面を修正している。6点分を全て正解することで、繰り返しを理解していることになる。なお、この問題は、プログラミングの考え方の一つである繰り返しに関して高等学校の教科書の問題をもとに作成したことから、問題内容は妥当と考えられたが、対象学生に対して適切なレベルであるかどうかについて検討するために予備調査を実施した。

表 2-1 プログラミングクラスのシラバス

第1回	ガイダンス, 変数の宣言と使い方, プログラムで計算をさせ, 変数の理解とプログラムによる計算方法を理解させる
第2回	オブジェクトについて, オブジェクトの定義と使い方について, 簡単な例を挙げ, Java におけるオブジェクトの意味を理解させる
第3回	図形を表示する 1 四角形, 様々な色の利用 Java によるグラフィクスにおいて, 図形を描く方法とその図形に色を指定する方法を理解させる
第4回	図形を表示する 2 楕円, 四角形と楕円, 文字列 図形を複数表示される方法を理解させる
第5回	繰り返し, 条件判断 1 図形をたくさん表示, while 文を利用 図形をたくさん表示させることを While 文で書かせ, ある条件で位置をずらして図形を表示するのを IF 文で書かせることで, 繰り返しと条件分岐を理解させる
第6回	繰り返し, 条件判断 2 for 文, if 文 図形をたくさん表示させることを for 文で書かせ, ある条件で位置をずらして図形を表示するのを IF 文で書かせることで, 繰り返しと条件分岐を理解させる
第7回	アニメーションを作る 1 図形をゆっくり表示, 図形を移動 図形を画面の左から右, 右から左に動かすことをここまでの授業で学んだことで実現する. 繰り返しの理解を深める.
第8回	アニメーションを作る 2 図形をいろいろな方向に移動 図形を画面の下から上や斜めに動かすことをここまでの授業で学んだことで実現する. 繰り返しと条件分岐の理解を深める.
第9回	クラス・オブジェクト 1 顔の図形を表示, オブジェクトの利用 オブジェクトを利用することで, 複数の顔を短い行数で書けることを理解させる
第10回	クラス・オブジェクト 2 図形の定義に速度情報, 他の図形を定義 オブジェクトに複数の情報を定義できることを理解させる
第11回	配列 1 前章までの復習, 配列の利用 配列を利用させることで, プログラムを短くすることを理解させ, 分かりやすいプログラムを書かせる
第12回	配列 2 要素を増やしてみる, 要素数を指定しないで多くにの図形を移動 配列の要素を増やし, 配列の利用方法の理解を深める
第13回	作品制作 1 配列を使用して, 各自のテーマでアニメーションを作成
第14回	作品制作 2 配列を使用して, 各自のテーマでアニメーションを作成し提出

表 2-2 非プログラミングクラスのシラバス

第1回	講義概要, 基本操作, 日本語入力, タッチタイピング, 情報倫理 キーボード入力の方法とキーボード入力練習の方法
第2回	電子メール, インターネット, 情報倫理 インターネットからの情報の利用や電子メールの書き方, LMS による情報倫理の方法を理解させる
第3回	Word1(Word 入門) ビジネス文書作成 Word でのフォントの書式設定, 文字位置の設定方法を理解させ, 簡単なビジネス文書を作成させる
第4回	Word2(Word 実践) 表の作成 Word による表作成の方法を理解させ, 表の入ったビジネス文書を作成させる
第5回	Word3(Word 活用 その1) 図の利用 Word による図の挿入方法と図形の作成方法, ワードアートなどの利用方法を理解させ, 見栄えの良い文書を作成させる
第6回	Word4(Word 活用 その2) 長文レポートの作成 Word でのアウトラインやスタイルの利用方法について目次を作成させることで理解させる
第7回	Excel1(Excel 入門) 数式の入力 Excel での計算をセル, 演算記号を利用して行うことと, フィルハンドルによる数式のコピーを理解させる
第8回	Excel2(Excel 実践) 関数の利用 合計 (SUM) や平均 (AVERAGE) などの関数を利用することで, 数式を効率良く利用できることを理解させる
第9回	Excel3(Excel 活用 その1) グラフ作成その1 棒グラフや折れ線グラフなどのグラフを作成する方法と, グラフのタイトルなど, グラフ内の項目を変更する方法を理解させる
第10回	Excel4(Excel 活用 その2) グラフ作成その2 折れ線グラフや複合グラフの作成方法を学びながら, 与えられた問題やデータに相応しいグラフの作成方法を理解させる
第11回	Excel と Word の連携(Word 活用 その3) Excel で作成した表やグラフを Word へ貼り付ける方法について, リンクの貼り付けの方法を理解させる
第12回	PowerPoint1(PowerPoint 入門) プレゼンテーションを行う上でのスライドの作成を, 図や図形の挿入, SmartArt の利用方法などを通して理解させる
第13回	PowerPoint2(PowerPoint 実践) 画面切り替え設定, アニメーションの設定などを通して, スライドにある文字, 図, 図形に効果を加える方法を理解させる
第14回	授業内試験 Word と Excel による文書作成を問題とする

予備調査は、2017年4月と7月に、プログラミングの授業の受講生26名に対し、今回と同様の方法で実施した[14]。その結果、問2の正答率は41.7%と低かったため、問2を解く前の足場かけとなる事前問題(付録参照)を作成した。事前問題の採点結果は、今回の調査データとしては用いないこととした。

2.4 プログラミングの考え方の理解度に関する仮説

本研究では、プログラミングのスキルの習得度とプログラミングの考え方の理解度の関係を明らかにするために、以下のような仮説を立てた。

仮説：プログラミングのスキルを習得することで、プログラミングの考え方である、順次や条件分岐、繰返しと理解度が受講前に比べて受講後で向上する。

3. 方法

前章で示したプログラミングクラスと非プログラミングクラスの授業は、関西学院大学の共通教育センターでの開講科目である。評価対象者はこの科目の受講生で、いずれも非情報系学部・学科に所属している1年生から4年生である。また、両クラスは選択科目であり、受講生は希望者が多数のため抽選で決められる。授業期間は、2017年10月から2018年1月までの授業であり、授業回数は14回であった。

表3に評価対象人数を示す。なお、評価対象人数が少ないのは、両クラスにおいて事前と事後のテストの両方を受けていない、プログラミングクラスについては最終課題を提出していない受講生がいたためである。

表3 評価対象のクラスと人数

クラス	履修人数	評価対象人数
プログラミング	30名	22名
非プログラミング	78名	65名

3.1 分析1

分析1は、プログラミングのスキルの習得度の確認である。そのために、A点におけるプログラミングクラスでの最終課題であるアニメーションを作成させた作品の評価結果(15点満点)の得点分布を示す。これにより、一般的なプログラミングの授業として、プログラミングのスキルを習得していたのかを示す。

3.2 分析2

分析2は、プログラミングの考え方の理解度が向上したことの確認である。そのために、2.3節で示した評価問題を使用し、学習内容(プログラミングクラス、非プログラ

ミングクラス)と受講前後(事前、事後)から検証する。プログラミングクラス、非プログラミングクラスに対して、授業期間の同時期にプログラミングの考え方の理解度の評価問題を解かせる。事前テストの実施はプログラミングの授業の第5回目の冒頭、繰返しと条件分岐に入る前の段階で実施する(B点)。事後テストは、教科書を使用した課題が終了した第13回の授業の冒頭で実施する(C点)。この実施に当たっては、事前テストの正解および各自の点数は公表しない。さらに、事後テストの難易度は事前テストの難易度と同じであるが、若干の内容を変更する(付録参照)。これらについて、問1と問2の得点結果について統計分析を行う。

3.3 分析3

分析3は、受講後のプログラミングのスキルとプログラミングの考え方の理解度の関係の確認である。そのために、プログラミングのスキルのA点とプログラミングの考え方の理解度のC点との相関分析を行う。

4. 結果

4.1 分析1

図1におけるA点で取得し、プログラミングのスキルを測った最終課題の得点分布を図2に示す。平均点は9.4点、標準偏差は3.3点であった。図2により6点の学生が多かったが、作成したプログラムで繰返しと配列は利用しているが、アニメーションで動かす絵が個数に差があるなど、絵の出来栄による得点により差があったためである。評価対象者全員のプログラムのソースを確認したところ、全学生のプログラムにおいて繰返しと配列が利用されていたことにより、本授業において身に付けさせたかったプログラミングのスキルである、繰返しとプログラムを効率良く作成するための配列について習得されていることが確認された。

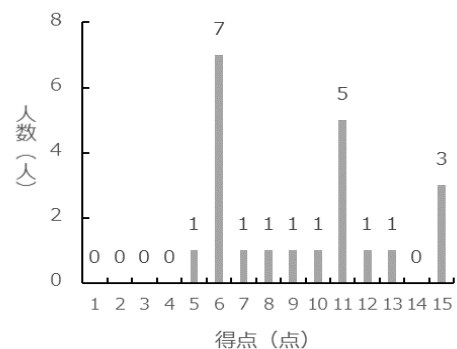


図2 最終課題の得点分布

4.2 分析 2

プログラミングの考え方の理解度に関する得点結果について、問 1 に関しては表 4 に、問 2 に関しては表 5 に示す。以下では、問 1 と問 2 のそれぞれの得点別の分析結果を記述した。まず、各得点の分布に関して Shapiro-Wilk の正規性検定を行ったところ、いずれの得点も正規分布に従っていないことが示された。そこで以降の分析では、ノンパラメトリック手法の一つである Mann-Whitney の U 検定を用いた。具体的には、問 1 と問 2 の得点に関して、受講前（10 月）と受講後（12 月）の各時点において、プログラミングクラスと非プログラミングクラスの得点比較を行った。また、プログラミングクラスと非プログラミングクラスそれぞれについて、受講前後の得点比較を行った。なお、検定の多重性を考慮し、有意性の判定には Bonferroni 法によって調整した有意水準を適用した（調整後の $\alpha = .05/4 = .0125$ ）。

表 4 プログラミングの考え方の理解度の得点結果
(問 1)

	プログラミングクラス		非プログラミングクラス	
	10 月	12 月	10 月	12 月
N (人)	22	22	65	65
Mean (点)	2.6	3.5	2.2	2.3
S.D. (点)	1.4	1.0	1.6	1.6

(問 1 の満点は 4 点)

表 5 プログラミングの考え方の理解度の得点結果
(問 2)

	プログラミングクラス		非プログラミングクラス	
	10 月	12 月	10 月	12 月
N (人)	22	22	65	65
Mean (点)	2.5	3.0	2.3	2.6
S.D. (点)	1.0	1.8	0.6	0.8

(問 2 の満点は 6 点)

4.2.1 問 1 の分析結果

受講前の時点では、プログラミングクラスと非プログラミングクラスの得点に差はみられなかったのに対して、受講後ではプログラミングクラスの得点は非プログラミングクラスと比べて高かった ($p=.002$)。また、非プログラミングクラスでは受講前後の得点に差は見られなかったが、プログラミングクラスでは受講前に比べて受講後の得点が高いことが見出された ($p=.004$)。

4.2.2 問 2 の分析結果

受講前、受講後のいずれの時点においても、プログラミングクラスと非プログラミングクラスの得点に差はみられなかった。また、非プログラミングクラスとプログラミングクラスのいずれにおいても、受講前後の得点に差はみられなかった。

4.3 分析 3

受講後のプログラミングの考え方の理解度とプログラミングのスキルの習得度との順位相関係数を求めたところ問 1 の得点とプログラミングのスキルの習得度との間には $\rho = .47$ ($p < .05$) で中程度の正の相関があった。問 2 の得点とプログラミングのスキルの習得度との相関係数は $\rho = .23$ となり有意な相関はみられなかった。

5. 考察

分析 1 の結果から、プログラミングクラスの学生は受講後の段階で繰り返しや配列の利用といったプログラミングのスキルを習得していることが確認された。分析 2 からは、プログラミングクラスの受講生は非プログラミングクラスの受講生に比べて、問 1 で評価した順次や条件分岐といったプログラミングの考え方の理解度が向上していることが示された。しかし、繰り返し処理の理解度に関しては、プログラミング学習の効果はみられなかった。分析 3 から、プログラミングクラスの受講生におけるプログラミングスキルと順次や条件分岐に関するプログラミングの考え方の理解度に中程度の正の相関があることが認められた。

分析 2 においてプログラミングの考え方に関する理解度の向上が問 1 の得点において認められた。プログラミングクラスの授業内容を振り返ると、プログラミングの考え方の理解度を深める工夫として以下の点が効果的であったと推察される。

- ・プログラミングクラスの第 7 回目の授業の冒頭において、前週までに学習した繰り返しに関する実力確認問題として実施した。すなわち、学習した Java 言語の for 文および While 文を用いて、画面に 1 から 10 を連続して表示させるプログラムを各自に書かせた。ただし、ここではプログラムが正確に動作をしなくても良く、プログラムとしての正解も示さなかった。
- ・その翌週にプログラムの正解を紙で配布し、全ての行について各自で注釈文を書かせ、各行がプログラムとしてのどのような意味を持つかを理解させた。

ここでプログラムへの注釈文を書かせたことで、変数と各行の意味、そして、プログラムが正確に実行されるのにはどのような順番であるかべきかの理解が深まったと考えら

れる。

これらのプログラミングの考え方の理解度を深める工夫は、問1における順次や条件分岐の処理というよりもむしろ問2における繰返しと繰返し内での代入文の理解を深めることを意図した処置であったが、期待に反してプログラミングクラスにおける受講後のプログラミングの考え方の理解度は向上が見られなかった。この理由を探るために、問2の履修生の誤答内容を確認してみると、

- ・繰返しの判定をする命令文 (A) の位置の違い
- ・箱の中身を表示される命令文 (F) の位置の違い

の2つのつまずきにより、全体の解答の順番が大きく違っている様子がみてとれた。このことより、問2のような抽象度の高い日本語による表記であると、繰返しとそれに伴う表示の命令文の位置について正確な理解が出来ていないことが分かった。

分析3からは、最終課題の得点と事後テストの得点には中程度の相関があり、最終課題のアニメーションの出来栄が良い学生は、問1の得点も高いことが分かった。授業担当者の実感として、履修生への毎回の課題では必ず順次処理を利用していることと、アニメーションの出来栄の良い学生が提出したプログラムのソースコードを確認してみると、繰返しの利用時に条件分岐を繰返し内で利用してアニメーションの動きに変化を与えるなど、条件分岐の理解が深い学生が多かったため、問1の得点が向上したためと考えられる。

本研究で用いたプログラミングの考え方の理解度を測る問題は、授業内で使用したJava言語の文法には依存しない問題形式となっていた。今回の授業内容によるプログラミングのスキルの習得では、順次や条件分岐といった処理の理解度が向上することが示された。しかしながら、繰返しに関しては、プログラム言語に依存しない形での理解度の向上を裏付ける知見を得ることはできなかった。繰返しは日常生活でも利用される概念であるが、それをプログラミングで記述した場合に、繰返しの条件設定の命令文の位置はどこが確か、そして繰返しの中のどの部分に命令文をおくと自分の意図した結果を表示ができるかなどが、条件分岐と比べて直感的に判断しにくいのではないかと考えられる。よって、繰返しに関するプログラミングの考え方の理解度を高めるには、繰返しの条件設定の理解だけでなく、繰返し内での表示などの命令文がどの位置であれば適切であるかをプログラムの作成を通して育成することが必要であると考えられる。

以上をまとめると、プログラミングのスキルを習得することで、限定的ではあるものの、学習したプログラミング言語には依存しない形でプログラミングの考え方の理解度が向上する可能性が示唆されたといえる。

6. おわりに

今後の課題としては、プログラミングクラスにおいて、今回の授業とは異なる学習内容や異なるプログラミング言語を使用した授業においても同様な結果が得られるかという点や、プログラミングのスキルとプログラミングの理解度の関係を、本研究とは異なる観測点、例えばプログラミングのスキルに関して受講前の観測を行うことで、より正確なプログラミングのスキルの習得度を測ることと、受講前のプログラミングのスキルと受講前後のプログラミングの考え方の理解度にどのような関係があるのかを検討したい。

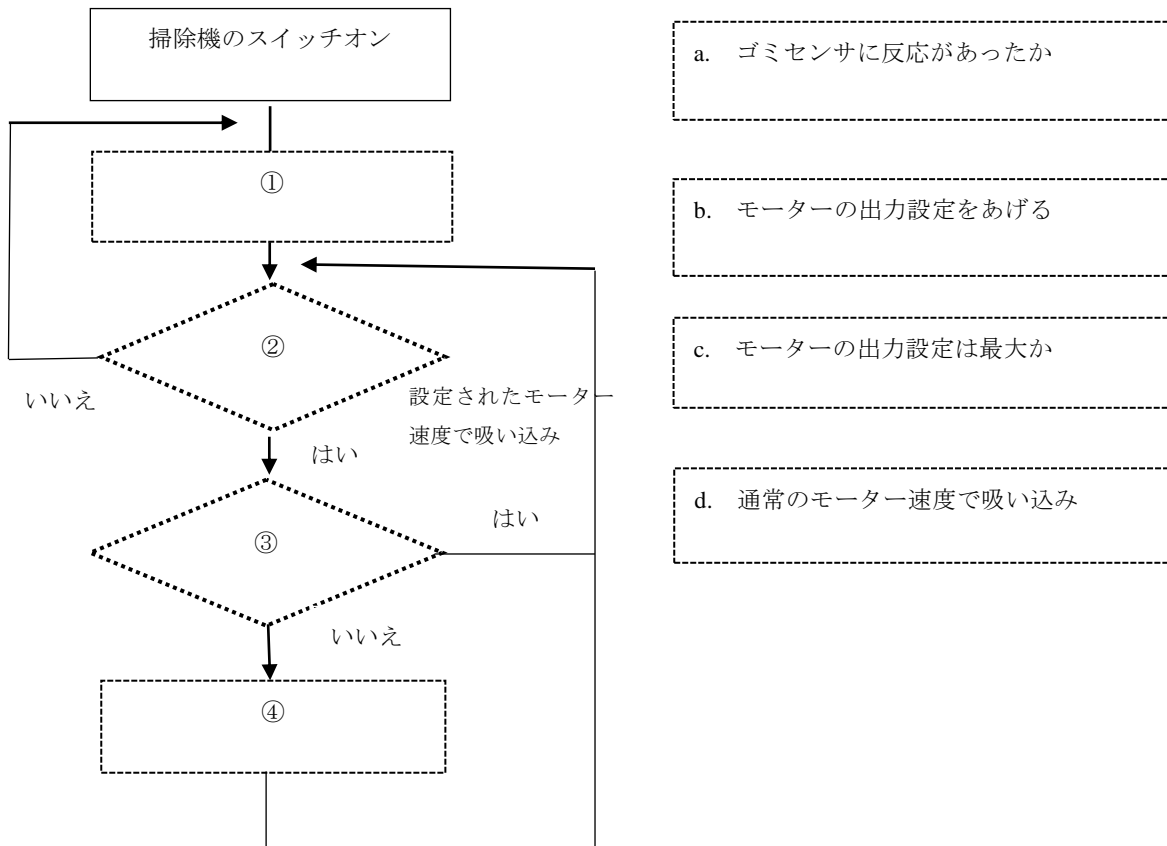
謝辞 本研究を進める上で有益な御助言を頂いた、東北大学大学院情報科学研究科情報リテラシー教育プログラムの代表・窪俊一准教授、副代表・静谷啓樹教授、ならびにメンバーである、徳川直人教授に謹んで感謝の意を表します。また、原稿作成にご協力を頂いた、東北大学大学院情報科学研究科メディア教育論ゼミの皆様にも感謝の意を表します。

参考文献

- [1]高橋尚子. 国内750大学の調査からみえてきた情報学教育の現状(3)一般情報教育編. 情報処理. 2017, Vol.58, No.6 p.526-530.
- [2]西田知博, 原田章, 中西通雄, 松浦敏雄. プログラミング入門教育における図形描画先行型のコースウェアが学習に与える影響. 情報処理学会論文誌教育とコンピュータ. 2017. Vol.3, No.1, p.26-35.
- [3]中西渉, 辰己丈夫, 西田知博. Pen Flowchartを用いたフローチャートによるプログラミング学習の効果に対する評価. 情報処理学会論文誌教育とコンピュータ. 2015. vol.1, No.4, p.75-82.
- [4]“「プログラミング人材育成の在り方に関する調査研究」報告書”. 2015. http://www.soumu.go.jp/main_content/000361430.pdf. (参照 2017-08-31)
- [5]“小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)”. 2016. http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm. (参照 2017-08-31)
- [6]磯部秀治, 小泉英介, 静谷啓樹, 早川美徳. コンピュータシヨナルシンキング. 共立出版. 2016. 204p.
- [7]河村一樹. 一般情報教育におけるプログラミング教育の在り方について. 情報処理学会コンピュータと教育研究会研究報告. 2011, vol.2011-CE108, No.16, p.1-8.
- [8]大岩 元. 識字教育としてのプログラミング. 情報処理学会論文誌教育とコンピュータ. 2015. vol.1, No.2, p1-6.1(2),1-6.
- [9]大場みち子, 伊藤恵, 下郡啓夫, 藤田憲久. 論理的文章作成力とプログラミング力との関係分析. 2018. 情報処理学会論文誌教育とコンピュータ. vol.4.No1.p.8-15.
- [10]宮田仁. プログラミングの教育方法と問題解決能力育成との関連. 教育情報研究. 1997. vol.12, no.4, p.3-13.
- [11]長慎也. 初級Java. 実教出版. 2014. 263p.
- [12]“情報活用能力調査(高等学校)報告書” http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/_icsFiles/fieldfile/2017/01/18/1381046_02_1.pdf. (参照 2017-09-03)
- [13]岡本敏雄編修, 山極隆監修. 最新情報の科学. 実教出版. 2003.
- [14]吉田典弘, 堀田龍也, 篠澤和久. “プログラミング教育における手順的思考力に関する評価方法の分析”, 情報処理学会コンピュータと教育研究会研究報告. 2017. vol.CE141-4, p.1-8.

付録

問1 部屋の掃除をするために、掃除機のスイッチをオンにしました。掃除機の中には、ごみの状況を把握（はあく）して、出力を調整するセンサとコンピュータがあります。掃除機はどのような作業をしているのでしょうか？作業の流れに合うように右のカードを左の①～④に入れましょう。



予備問題

花子さんは、「おはじき」をたくさん持っています。箱に1個から1個ずつ増やし、10個になるまで手元にある「おはじき」を入れる作業をします。以下の作業の手順を並べ替えてください。

- A) 10個になるまで、箱に入れる作業を繰り返す
- B) 箱を用意する
- C) 箱に入れる作業を終わる
- D) 前に入れたおはじきの数に1個追加して箱に入れる
- E) 箱におはじきを1個入れる

問2（事前テスト）

1から10までの数を、連続して表示する手順について、以下の項目を並べ替えてください

（パソコンでプログラムを書き、画面に「1 2 3 4 5 6 7 8 9 10」と表示することをイメージしてください）

- A) 箱 X が 10 以下の間、次の作業を繰り返す
- B) データを入れる、「箱 X」を用意する
- C) 箱 X に 1 を足した値を、箱 X に戻す
- D) 箱 X に 1 を入れる
- E) 繰り返しを終了する
- F) 箱 X の中身を表示する

問2（事後テスト）

画面に「10 20 30 40 50 60 70 80 90 100」と表示するとして実施をした。