情報処理学会論文誌　教育とコンピュータ　Vol.7　No.1　14–24 (Feb. 2021)

論　文

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

吉田 典弘[1,a)]　和田 裕一[2]　邑本 俊亮[2]　堀田 龍也[2]　篠澤 和久[2]

受付日 2018年8月17日，再受付日 2019年7月31日 / 2020年3月21日，
採録日 2020年6月27日

概要：プログラミングのスキルの習得にはプログラミングの基本的な概念（以下，プログラミングの考え方）の理解がともなうと考えられるが，これまでの研究ではこれらの間の明確な関連性は見い出されていない．そこで，大学の一般情報教育においてプログラミングの授業（プログラミングクラス）を実施し，プログラミングのスキルとプログラミングの考え方の習得度の関係について検討した．プログラミングのスキルは授業終了時の最終課題の成績によって評価した．一方，プログラミングの考え方については，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様式でプログラミングの考え方の理解度を測る評価問題を用いた調査を実施した．対照群として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）を設け，2 つのクラスについて受講前後でのプログラミングの考え方の理解度を比較したところ，プログラミングクラスでは向上したことが分かった．また，限定的であるが，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間には正の相関が認められた．これらの結果から，プログラミングのスキルの習得度が，学習したプログラミング言語には依拠しない様式で測定されたプログラミングの考え方の理解力と関連性を持つことが示唆された．

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

Relationship between Acquisition Level of Programming Skills and Understanding of Programming Thinking in General Information Education

Norihiro Yoshida[1,a)]　Yuichi Wada[2]　Toshiaki Muramoto[2]　Tatsuya Horita[2]
Kazuhisa Shinozawa[2]

Received: August 17, 2018, Revised: July 31, 2019/March 21, 2020,
Accepted: June 27, 2020

Abstract: It is reasonable to suppose that the acquisition of programming capability is accompanied by a fundamental understanding of the concepts of programming (hereafter referred to as programming thinking). However, previous studies have not found clear associations between them. We conducted a programming course for general information education to investigate the relation between the acquisition level of specific programming skills the students learned in the course and the development of programming thinking. Before and after the course, we conducted a survey that measured the students' level of programming thinking. Students in the control group did not undergo any classes for mastering their programming skills and students in the experimental group did undergo group for mastering their programming skills. For these two groups, we compared the degree of the development of programming thinking before and after the courses. The results revealed that students' level of programming thinking raised after undergoing the programming course. To examine the factors causing this improvement, we investigated the correlation between the students' achievement of programming skills and the level of programming thinking at the end of the course. We found a limited substantial correlation between these two variables, indicating that the development of programming thinking is related to the acquisition of programming capability.

Keywords: general information education, programming skills, programming thinking

© 2021 Information Processing Society of Japan

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のスキルの習得が主要な学習目標の 1 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，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）」[4] 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，1 つ 1 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．したがって，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るといえる．

このようなプログラミングの考え方の理解については，大学の一般情報教育においても重要な到達目標の 1 つであり，プログラミングの考え方の理解度を測る重要性は大学の初年次を対象としたプログラミング教育にもあてはまると考えられる．これまでの一般情報教育でのプログラミング教育においては，プログラミングのスキルの測定をもって考え方の評価としてきたと思われる．すなわち，スキルが習得されていれば，プログラミングの考え方も身についたと解釈できるものとされてきた．しかしながら，スキルと考え方とは異なるものであり，それらの評価も異なるものであるはずである．また，プログラミングの考え方は，学習したプログラミング言語に依存しない，一般的な概念としてのプログラミングの考え方を問う問題によって評価されるべきであろう．

ところで，プログラミングのスキルと他の能力との関係を調べた研究としては以下のようなものがあげられる．大場ら [5] は，論理的文章作成力とプログラミング力の間に強い相関がみられたことを報告している．宮田ら [6] 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こりやすいことを示している．また，赤堀 [7] はプログラミングの考え方に関する問題と教科（国語，算数，理科，社会）における論理的な思考の問題を大学生に解かせて，これらの結果に相関があることを見出している．これらの研究は，プログラミングのスキルが，論理的文章力や問題解決能力等の抽象的な思考力と関連することを明らかにしているが，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との関連を検討したものではない．よって，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の関係については，いまだ実証的な知見が不足しているといえる．

そこで本研究では，プログラミングの学習によるプログラミングのスキルの習得とプログラミングの考え方の理解度との関連性を調べ，スキルの向上が実際の理解力の向上に裏付けられているのかを検証する．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，最終課題として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することを求めてスキルを測定するとともに，学習したプログラミング言語に依存しない様式で作成された，構造化プログラミングの構成要素である順次，条件分岐，繰返しの理解度を測るテストを，プログラミングの学習の前後で実施する．これらの結果から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することとする．

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の枠組みと結果の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，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を評価した時点を意味する．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，両授業において，図 1 の A から C の 3 つに観測点を置いて，評価問題の得点結果について分析を行った．

A 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，プログラミングクラスが繰返しの課題に入る前の第 5 回の冒頭にプログラミングの考え方の理解度として，順次，条件分岐についてはフローチャート，繰返しについては抽象的な問題を使用して測定した．

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，第 13 回目の授業でプログラミングの考え方の理解度として，順次，条件分岐についてはフローチャート，繰返しについて抽象的な問題を使用して測定した．

[1] 羽衣国際大学
Hagoromo University of International Studies, Sakai, Osaka 592–8344, Japan
[2] 東北大学大学院
Graduate School, Tohoku University, Sendai, Miyagi 980–8579, Japan
[a)] noriyoshida@hagoromo.ac.jp

© 2021 Information Processing Society of Japan

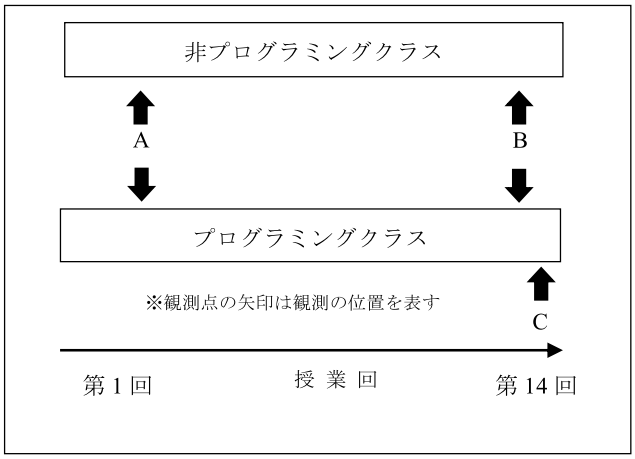


図 1　観測点
Fig. 1　Figure1 Observation points.

表 1　観測点と分析方法
Table 1　Observation points and analysis method.

	図 1 における観測点	観測対象	使用するデータ
分析 1	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
分析 2	A 点と B 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
分析 3	A 点と C 点 B 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

C 点：プログラミングクラスの授業終了後に，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．提出されたプログラムのソースコードについて，プログラムが正確に動作したことを確認し，プログラム中の繰返しの利用回数と配列の使用回数を確認して，その合計数で採点した．

さらに，これらの観測点で測定したデータの分析手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，授業の到達目標であるプログラミングのスキルが習得されていることを明らかにするために，C 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だけでなく，プログラミングの考え方の理解度も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 A 点と B 点の評価問題の結果を比較する．

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との関係を明らかにするために，B 点における理解度得点と C 点におけるスキル得点との相関関係を分析する．同様に受講前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との関係を明らかにするために，A 点における理解度得点と C 点におけるスキル得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．

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，繰返し処理の理解ができ，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．学習内容は Java 言語を用いてアニメーションを作成することであり，プログラミング初心者向けの授業であった．シラバスを**表 2** (a) 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，繰返し処理を教えた後に，グラフィックス関数で自作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 [8] を使用した．評価は，授業を担当した著者のうちの 1 名が学生から提出されたプログラムのソースコードについて，その繰返しの利用回数と配列の数を確認し，その合計数と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の科目名は「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 (b) に示す．評価は，毎回の授業における課題提出状況と，最終試験として課した Word と Excel による文書作成の成果物に基づいて実施した．なお，この授業では，プログラミングの考え方の理解度に関する内容，たとえば順次処理，条件処理，繰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

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）．この問題を正解するには，構造化プログラミングの構成要素である順次，条件分岐，繰返しの 3 つの処理を理解して，手続きの順番を正しく並べ替えることが求められる．

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　繰返しを問う問題：題材は 1 から 10 の数字を表示（配点 5 点）

© 2021 Information Processing Society of Japan

表 2　(a) プログラミングクラスのシラバス．(b) 非プログラミングクラスのシラバス
Table 2　(a) Syllabus of programming course. (b) Syllabus of non-programming course.

(a)

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 図形を複数表示す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

(b)

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

© 2021 Information Processing Society of Japan

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」[9] として使用された問題であった．これは，ロボット掃除機の動作を示した要素をフローチャートの適切な位置に配置してフローチャートを完成させるものであり，プログラミングの考え方のうちの順次と条件分岐の要素を含むものである．文部科学省の調査において高等学校第 2 学年の 4,552 名を対象とした正答率が 46.2 %であり適当な難易度であること，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．

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 [10] に記載されている例題のプログラムを引用し，問題文は自作した．問題作成においては，特定のプログラミング言語で用いられる用語や文法は用いず，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，繰返しに関する文面を修正している．なお，本論文では連得点方式を用いている．隣り合って配列された解答が正しい時に 1 点を与える採点法である [11]．この方法を採用した理由は，正解の位置が完全に一致していなくても，一部分でも連続して正解がある場合には得点を与える方がプログラミングの考え方の理解度として適当であると判断したからである．問 2 は計 5 点である．なお，この問題は，プログラミングの考え方の 1 つである繰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について検討するために予備調査を実施した．

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 [12]．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，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の平均が高く，しかも同程度であったたため，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の理解度が向上する．また，プログラミングのスキルとプログラミングの考え方には相関関係があり，具体的には，プログラミングの授業を受講することで，プログラミングのスキルを習得すると，プログラミングの考え方のうちの順次や条件分岐，繰り返しに関する授業で学習したプログラミング言語に依存しない抽象的な問題に対する得点が向上する．

表 3　評価対象のクラスと人数
Table 3　Target classes and the numbers of students.

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.　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** に履修人数と評価対象人数を示す．なお，評価対象人数が履修人数より少ないのは，両クラスにおいて事前と事後のテストのどちらか一方を受けていない，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．

3.1　分析 1

分析 1 の目的は，受講生がプログラミングのスキルとして繰返しと配列の利用方法を習得していたことを確認することである．そのために，プログラミングクラスにおける C 点での最終課題の評価結果を集計し，得点化した．一般的に「繰返しと配列の使用数が多ければ，よりプログラミングのスキルがある」と判断できる訳ではないが，ここでの得点は作成したプログラムのなかで使用した繰返しの回数と，アニメーションにおける絵を表現する際に使用した配列の数との合計数とした．アニメーションを作成する自由課題なので，多くの絵を使って複雑な動きをさせるプログラムは繰返しの利用回数や配列の数が増える傾向にある．実際の提出物にはそのような傾向があったことから，繰り返しや配列を多く使用して，複雑なアニメーションを作成できればスキルが高いと判断し，繰返しの利用回数や配列の数を評価指標とした．

3.2　分析 2

分析 2 の目的は，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 2 つの評価問題を使用し，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（A 点）．事後テストは，教科書

© 2021 Information Processing Society of Japan

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（B 点）．この実施にあたっては，事前テストの正解および各自の得点は公表しない．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

受講前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を調べるために，C 点で測定したプログラミングのスキルと A 点で測定したプログラミングの考え方の理解度との相関分析，C 点で測定したプログラミングのスキルと B 点で測定したプログラミングの考え方の理解度の相関分析を問 1 と問 2 それぞれについて行う．

4.　結果

4.1　分析 1

図 1 における C 点において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 4.5 点，標準偏差は 3.6 点であった．図 2 により得点が 3 点の学生が多かったことがみてとれる．これは，課題提出の条件として，配列を使用して各自の画像を制作することと，繰返し文を利用して左から右へ移動させること，右から左に移動させることをそれぞれ最低 1 回は使用することしたことによる．図 2 において得点差があるのは，作成したプログラム中で，繰返しを利用している回数に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，22 名中 21 名のプログラムにおいて繰返しと配列が少なくとも 1 回以上利用されていたことから，本授業の習得目標としたプログラミングのスキルである，繰返しと配列について習得されていることが確認された．また，5 点以上の得点であったのは，22 名中 8 名であった．

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 の得点を**表 4** に，問 2 の得点を**表 5** に示す．また，得点分布を**図 3**，**図 4**，**図 5**，**図 6** に示す．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ので，以降の分析ではノンパラメトリック手法を用い，学習内容の得点比較においては Mann Whitney の U 検定，事前事後得点の比較においては Wilcoxon の符号付き順位検定を行った．具体的には，問 1 と問 2 の得点に関して，受講前（A 点）と受講後（B 点）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，各問題において受講前後におけるクラス間の比較と

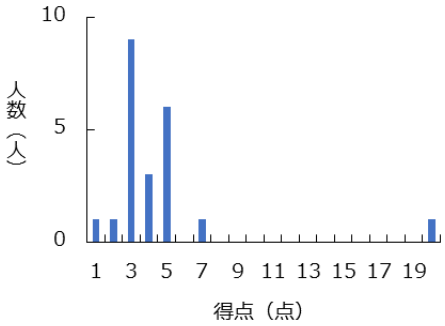


図 2　最終課題の得点分布
Fig. 2　Score distribution of the final task.

表 4　プログラミングの考え方の理解度の得点結果（問 1）
Table 4　Scoring result of Comprehension degree of programming thinking (question1).

	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
Median(点)	2.0	4.0	2.0	2.0
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）
Table 5　Scoring result of comprehension degree of programming thinking (question2).

	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）	1.1	1.9	1.1	1.2
Median(点)	1.0	1.0	1.0	1.0
S.D.（点）	1.0	1.5	0.7	0.9

（問 2 の満点は 5 点）

各クラスにおける受講前後の比較の計 4 回の比較がなされることから，検定の多重性を考慮して有意性の判定には Bonferroni 法によって調整した有意水準を適用した（調整後の $\alpha = .05/4 = .0125$）．

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みられなかったが，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い出された（$p = .004$）．

© 2021 Information Processing Society of Japan

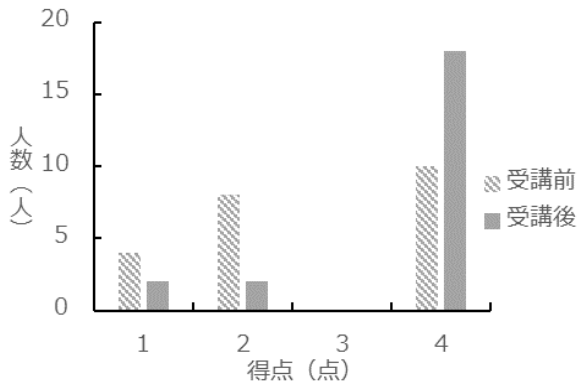


図 3　プログラミングクラスの問 1 の得点分布

Fig. 3　Score distribution of question No.1 in programming class.

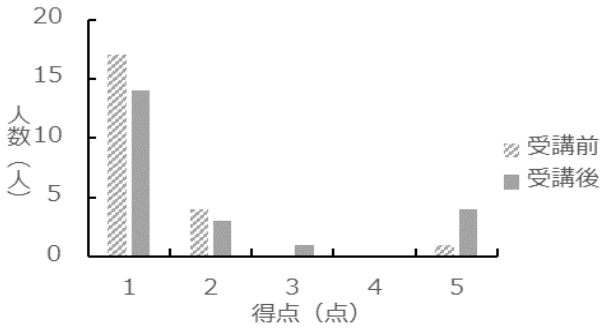


図 4　プログラミングクラスの問 2 の得点分布

Fig. 4　Score distribution of question No.2 in programming class.

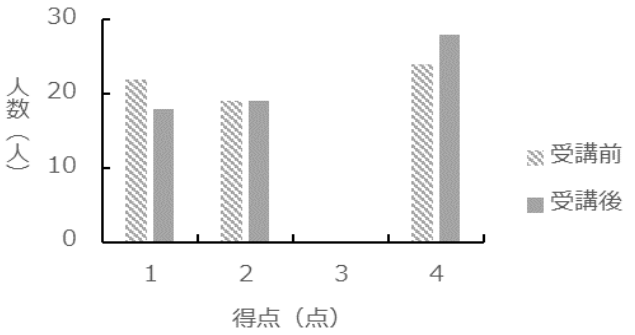


図 5　非プログラミングクラスの問 1 の得点分布

Fig. 5　Score distribution of question No.1 in non-programming class.

4.2.2　問 2 の分析結果

受講前と受講後の時点ともに，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では受講前後の得点に差はみられなかったが，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傾向がみられた（$p = .013$）．

4.3　分析 3

図 7 と**図 8** にプログラミングクラスの受講後の問 1 と問 2 の得点とプログラミングススキルの得点の相関を示す．

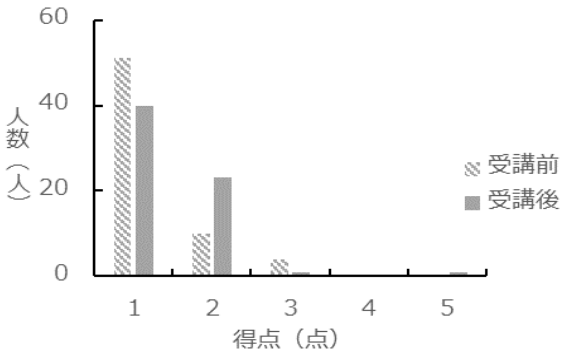


図 6　非プログラミングクラスの問 2 の得点分布

Fig. 6　Score distribution of question No.2 in non-programming class.

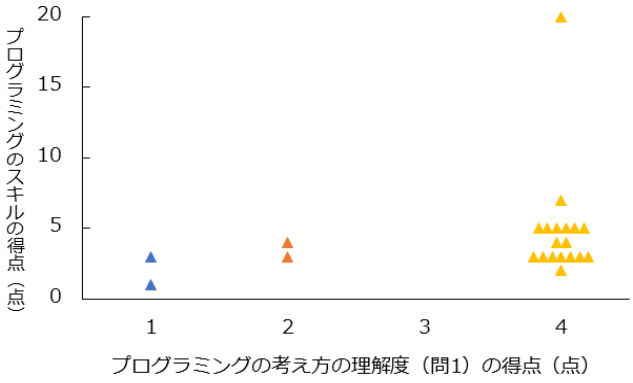


図 7　問 1 の得点とプログラミングスキルの得点の相関

Fig. 7　Correlation between scores of question No.1 and the programming skills.

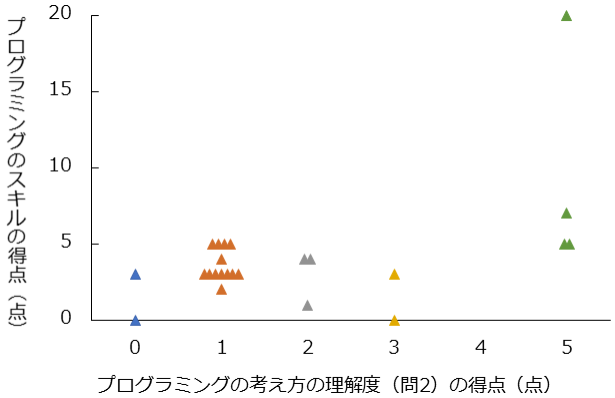


図 8　問 2 の得点とプログラミングスキルの得点の相関

Fig. 8　Correlation between scores of question No.2 and the programming skills.

受講前のプログラミングの考え方の理解度（観測点 A）と受講後のプログラミングのスキルの習得度（観測点 C）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有意な相関はみられなかった（$\rho = .37$，n.s.）．問 2 の得点とプログラミングのスキルの習得度との間には中程度の正の相関があった（$\rho = .44$，$p < .05$）．

また，受講後のプログラミングの考え方の理解度（観測

© 2021 Information Processing Society of Japan

点 B）とプログラミングのスキルの習得度（観測点 C）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有意な相関はみられなかった（$\rho = .37$, n.s.）．問 2 の得点とプログラミングのスキルの習得度との間には中程度の正の相関があった（$\rho = .44$, $p < .05$）．

5. 考察

プログラミングクラスの評価結果から，次の 3 つのことが分かった．

(1) 分析 1 から，このクラスの学生は受講後の段階で繰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．

(2) 分析 2 から，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．問 2 で評価した繰返しの理解度に関しては，クラス間の差はなかったが，プログラミングのクラスの受講前後で理解度が向上している傾向が示された．

(3) 分析 3 から，プログラミングクラスの受講生における受講後のプログラミングスキルに関し，問 1 では受講前のプログラミングの考え方の理解度，受講後のプログラミングの考え方の理解度の双方に相関はみられなかった．問 2 では受講前のプログラミングの考え方の理解度，受講後プログラミングの考え方の理解度の双方に中程度の相関がみられた．

上記の (3) より，問 2 に限定していえば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と関連はみられたが，仮説で述べたようなプログラミングの学習によるプログラミングのスキルの習得とプログラミングの理解度の向上との関係を明確に示すことはできなかった．問 1 と問 2 の結果が異なっているが，問 1 の問題が構造化プログラミングにおける基本構造では比較的に理解しやすい，順次と条件分岐に関する問題であったため異なる結果となったと考えられる．また，問 2 の結果から受講生には受講前から繰返しに関する理解力があったが，この授業が開講されている情報系の共通教育科目として最上位レベルの科目であり，前段階として開講しているコンピュータ関連の授業を履修している学生が多かったことや，履修した学生がプログラミングに対する意欲や動機付けが高かった可能性があるので，受講前に繰返しに関する理解についてある程度の知識を持っていたと考えられる．

これらのことから，プログラミングの学習において理解が難しいとされている構造化プログラミングの基本構造の繰返しについて，プログラミングのスキルの習得度と受講前後のプログラミングの考え方の理解度の関係性を，学習したプログラミング言語に依らない問題によって評価することができた．

一方，上記 (2) のプログラミングクラスのプログラミングの考え方の理解度の得点が向上した理由として，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よく，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，すべ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．ここでプログラムの注釈文を書かせたことで，変数と各行の意味の理解が深まったと考えられる．

ただし，問 2 に関しては学習内容での有意な差が得られなかった．この理由を探るために，問 2 の誤答の内容を精査すると，

- 繰返しの判定をする命令文（A）の位置の間違い
- 箱の中身を表示する命令文（F）の位置の間違い

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．この誤答はプログラミングクラスと非プログラミングクラスの両方のクラスで認められた．また，プログラミングクラスの学生は，分析 1 の結果が示すように，当該授業内で学習したプログラミング言語を用いた同種の問題では繰返しの使い方や意味を十分理解できていたと推察される．これらのことから，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，繰返しを指定する命令文の条件設定の理解だけでなく，その命令文をプログラム内のどの位置に配置するのが適切であるかを判断する能力を育成することが必要であると考えられる．

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 言語の文法には依拠しない問題様式であり，授業内で習得した特定のプログラミング言語の文法に基づいて回答する場合よりも，思考の抽象度が高くなっている．プログラミングのスキルを習得することで，繰返しに関する理解力という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拠しないより一般的な概念としてのプログラミングの考え方の理解度との関係を示すことができた．このような知見を蓄積していくことは，特定のプログラミング言語に依存せずにプログラミングの考え方の理解度の評価する方法を確立する一助となると考えられる．

© 2021 Information Processing Society of Japan

6. おわりに

今後の課題として，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の結果が得られるかを検討することがあげられる．また，今回の調査では受講前のプログラミングのスキルを測定していないため，受講前後におけるプログラミングのスキルの向上の程度とプログラミングの理解度の変化の関連については検討できていなかった．また，プログラミングのスキルの評価指標も課題の内容に合わせたものではないように検討する必要がある．

さらに，より多くの時点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を測定し，両者の関連性が授業を通してどのような変遷をたどるかを詳細に追跡していくことで，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確にしていきたい．

謝辞　本研究を進めるうえで有益な御助言をいただいた，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いただいた，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，Vol.58, No.6, pp.526–530 (2017).
[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，Vol.3, No.1, pp.26–35 (2017).
[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，Vol.1, No.4, pp.75–82 (2015).
[4] 文部科学省：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), 入手先 ⟨http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm⟩（参照 2018-07-31）.
[5] 大場みち子，伊藤　恵，下郡啓夫，薦田憲久：論理的文章作成力とプログラミング力との関係分析，情報処理学会論文誌教育とコンピュータ，Vol.4, No.1, pp.8–15 (2018).
[6] 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，Vol.12, No.4, pp.3–13 (1997).
[7] 赤堀侃司：プログラミングの教育の考え方とすぐに使える教材集，ジャムハウス (2018).
[8] 長　慎也：初級 Java，実教出版 (2014).
[9] 文部科学省：情報活用能力調査（高等学校）報告書，入手先 ⟨http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/_icsFiles/afieldfile/2017/01/18/1381046_02_1.pdf⟩（参照 2017-09-03）.
[10] 山本博樹：主人公の目標構造の教示が幼児による絵画配列に及ぼす効果–継時的理解に及ぼす教示の効果の明確化，読書科学，No.36, pp.41–51 (1992).
[11] 岡本敏雄（編修），山極　隆（監修）：最新情報の科学，実教出版 (2003).
[12] 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，Vol.CE141-4, pp.1–8 (2017).

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．

事前問題

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

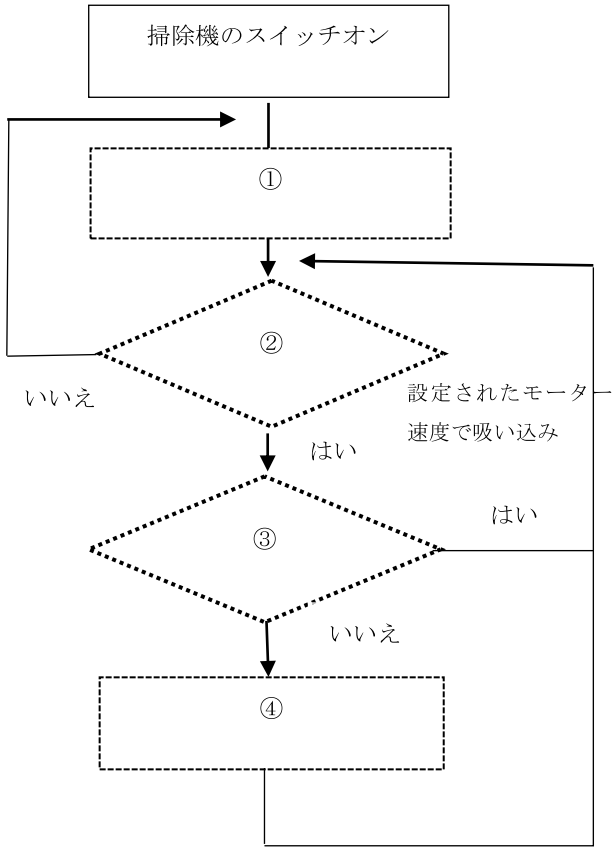
問 2（事後テスト）

10 から 100 までの数を，10 をおきに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「10　20　30　40　50　60　70　80　90　100」と表示することをイメージしてください）．

A) 箱 X が 100 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 10 を足した値を，箱 X に戻す
D) 箱 X に 10 を入れる
E) 繰返しを終了する
F) 箱 X の中身を表示する

© 2021 Information Processing Society of Japan





吉田 典弘　（正会員）

1992 年玉川大学大学院工学研究科修士課程修了．2017 年東北大学大学院情報科学研究科後期博士課程単位取得退学．福島女子短期大学助手，同大学講師，相模女子大学准教授，同大学教授，関西学院大学教授を経て，2019 年 9 月から羽衣国際大学現代社会学部教授．大学における一般情報教育の研究に従事．電子情報通信学会，日本教育工学会，教育システム情報学会等各会員．



邑本 俊亮

1992 年北海道大学大学院博士後期課程単位取得退学．博士（行動科学）．北海道大学文学部助手，北海道教育大学札幌校講師，助教授，東北大学大学院情報科学研究科助教授，准教授，教授を経て，2012 年から東北大学災害科学国際研究所教授．認知心理学，教育心理学の研究に従事．日本心理学会，日本教育心理学会，日本認知心理学会，日本読書学会等各会員．



和田 裕一

1996 年東北大学大学院情報科学研究科博士課程前期修了．1999 年同大学大学院博士課程後期修了．博士（情報科学）．東北大学情報科学研究科助手，同大学講師を経て，2005 年より同大学准教授．認知心理学，コンピュータリテラシーに関する研究に従事．日本心理学会，日本教育心理学会，日本人間工学会等各会員．



堀田 龍也

1986 年東京学芸大学教育学部卒業．2009 年東京工業大学大学院社会理工学研究科修了．博士（工学）．東京都公立小学校教諭，富山大学教育学部助教授，静岡大学情報学部助教授，独立行政法人メディア教育開発センター准教授，文部科学省参与，玉川大学教職大学院教授等を経て，2014 年から東北大学大学院情報科学研究科教授．教育工学，情報教育の研究に従事．日本教育工学会，日本教育情報学会，日本教育メディア学会，AACE 等各会員．

© 2021 Information Processing Society of Japan



篠澤 和久

1983 年東北大学大学院文学研究科博士課程前期修了．1988 年同大学院博士課程後期修了．博士（文学）．東北大学大学院文学研究科助手，山形大学人文学部助教授，東北大学大学院情報科学研究科准教授等を経て，2018 年より同大学院教授．ギリシャ哲学，分析哲学の研究に従事．日本哲学会，日本倫理学会，日本西洋古典学会等各会員．

© 2021 Information Processing Society of Japan