情報処理学会論文誌　教育とコンピュータ　Vol.6　No.2　25–37 (June 2020)

論　文

プログラミング学習支援ツール pgtracer を自学習に活用した授業実践と学習行動の分析

村田 美友紀[1,a)]　嘉藤 直子[2]　掛下 哲郎[3]

受付日 2019年3月11日，再受付日 2019年9月18日，
採録日 2020年2月29日

概要：我々は，初学者を対象としたプログラミング学習支援ツール pgtracer の開発を行ってきた．学習支援ツール pgtracer は Moodle 上で動作し，穴埋め問題を出題・自動採点する．また，学生の学習ログを自動収集し，学生の学習行動や理解度を分析するための機能を提供する．我々は，2016 年度と 2017 年度に開講されたプログラミング科目において，pgtracer を活用して自学習課題を提供する授業実践を行った．2016 年度に行った実践方法について問題点を検討し，2017 年度にそれらの改善を行った．本稿では，pgtracer を用いた授業実践を報告するとともに，2017 年度に収集したデータを分析し，学生のプログラミング学習の行動と理解の関係について明らかにする．分析の結果，トレース表の理解がプログラムの理解に関係していること，学生のプログラム理解度によって解答過程に違いがあること，継続して課題に取り組む学生の方が試験前に集中して学習する学生よりもプログラムをより深く理解していることが分かった．

キーワード：ラーニングアナリティクス，プログラミング，e-learning，穴埋め問題

A Practical Report on Self-Learning and Analysis of Student Activity Utilizing Programming Education Support Tool pgtracer

Miyuki Murata[1,a)]　Naoko Kato[2]　Tetsuro Kakeshita[3]

Received: March 11, 2019, Revised: September 18, 2019,
Accepted: February 29, 2020

Abstract: We have developed a programming education support tool pgtracer for programming beginners. The tool runs under Moodle and provides fill-in-the-blank questions to students. When a student fills the blanks and submit the answer, pgtracer automatically evaluates the student's answer. At the same time, pgtracer automatically collects learning log of the students and provides data analysis functions for the collected log. We assigned homework utilizing pgtracer to the students taking a programming course in 2016 and 2017. Since we found some problems in our 2016 arrangement, we improved the arrangement in 2017. In this paper, we report our practice, and clarify relationship between students' learning behavior and understanding of programming by analyzing the data collected in 2017. As a result, we found that their understanding of the trace table related to their programming skill, and that their answering process to reach to a correct answer depends on their programming skill. Furthermore, we found that the students who continuously did their homework understand programs better than those who only prepared for the examination in a short term.

Keywords: learning analytics, computer programming, e-learning, fill-in-the-blank question

[1] 熊本高等専門学校
National Institute of Technology, Kumamoto College, Yatsushiro, Kumamoto 866–8501, Japan
[2] 有明工業高等専門学校
National Institute of Technology, Ariake College, Omuta, Fukuoka 836–8585, Japan
[3] 佐賀大学
Saga University, Saga 840–8502, Japan
[a)] m-murata@kumamoto-nct.ac.jp

1. はじめに

プログラミングは，高専や大学など工学系の学生にとって重要である．2020 年度より小中学校および高校でのプログラミング教育が順次開始される予定になっており，プログラミング教育の重要性はますます高まっている [1]．

ところが，高専や大学における実際のプログラミングの

授業では，学力や学習意欲が低い学生がしばしば見られる．プログラミング上達のためには，多くのプログラムを作成することが有用だが，授業時間数や教師数などの制約から，授業時間だけでは十分な演習を行うことができない．これを補うためには学生の自学習による取り組みが必要であるが，自主性に任せるだけでは学生は学習しない傾向にあり，また教員は学生の自学習状況を教員が把握できない．課題とする場合にも，採点や提出状況の確認など，教員にかかる負担は大きい．ここで，自学習とは，「学生が学習目標の達成を図るために，授業時間外に行う，教員による直接の教授をともなわない学習」とする．

我々は，Moodle のプラグインとして動作するプログラミング学習支援ツール pgtracer を研究開発してきた [2], [3]．学習支援ツール pgtracer はプログラムとトレース表に対する穴埋め問題を提供する．インターネットが利用できる PC があれば，学生は好きなときに pgtracer を利用してプログラミング学習ができる．また，pgtracer の自動採点機能は教員の採点の負荷を軽減し，データ分析機能は教員による学生の学習状況の把握を支援する．

筆者らは，2016 年度と 2017 年度に熊本高等専門学校（以下，熊本高専）八代キャンパスの授業において pgtracer を活用した自学習課題の提供を行った [4], [5], [6]．本稿では 2016 年度に行った実践方法について問題点を検討し，2017 年度に改善した実践を行った．そこで得られたデータを用いて学習行動の分析を行い，学生のプログラミング学習の行動と理解の関係について検討する．

学習支援ツール pgtracer の分析機能を用いて，学生の学習ログを分析したところ，トレース表の理解の程度（理解度）がプログラムの理解に関係していること，学生のプログラミング理解度によって正答を導く思考の過程に違いがあること，自学習課題に継続して取り組む学生のほうが試験前に集中して学習する学生よりもプログラムの理解度が高いことが分かった．

本稿は以下のように構成されている．2 章では関連研究について述べる．3 章では pgtracer の概要について述べる．4 章では 2016 年度の実践方法について述べ，実践方法の検討を行う．5 章では 2017 年度の実践方法について述べ，6 章では学生の学習行動，トレース表の理解，科目成績との関係について考察する．最後にまとめと今後の課題について述べる．

2. 関連研究

学生の学習活動を分析し，プログラミング学習を支援するための研究は，様々なものが行われている．文献 [7] は，プログラミングエラーログと教材の閲覧ログを分析し，プログラムの構文エラーを発見し，それに対処するための教材を提供する．文献 [8], [9] は，オンライン学習を行う学生の支援を目的に，学生の学習活動を SAS Business Analytics を使用して分析し，適切なテキストやビデオなどの教材を提供する学習環境を構築している．文献 [10] では，ゲームベースの学習分析（GBLA）を用いて，学習分析メカニズムの導入をガイドするための枠組みを提案している．文献 [11] はプログラミング学習者のソースコード編集パターンを系列パターンマイニングすることにより，頻出編集パターンを抽出し，これと学習履歴を分析することで，学習状況推定の自動化を行う．文献 [12] は，プログラミング演習時においてコーディング過程を記録・可視化して教師に提示し，個別指導などの支援につなげるシステム C3PV を提案している．文献 [13] は，プログラムの穴埋め問題を提供し，項目反応理論に基づいた問題の難易度分析と学生の解答分析によって，適切な難易度の問題を学生に提供する方法を提案している．

学習支援ツール pgtracer は，各穴埋めの正誤と解答所要時間，問題の正解率と解答所要時間をログとして保存する．これらのデータについて分析機能を提供している．分析機能を用いることにより，文献 [10], [11], [12] が提供するような学生全体や学生ごとの学習状況の把握が可能である．さらに pgtracer では穴埋めの種類や場所を分析することで，学生が苦手としている項目の推定が可能である．穴埋めの種類としてプログラム中の各種要素（トークン，トークン列，文など）だけでなくトレース表を構成する個別の値も指定できるため，文献 [13] のような既存のプログラミング教育支援ツールと比較しても，より幅広い難易度のプログラミング問題を出題できる．そのため，学生の能力差が大きい場合においても，個々の学生の進度や能力に応じた問題を提供することが可能である．プログラミング習得のためには，文献 [7], [8], [9] が提供するようなテキストやビデオなどの教材による学習だけでなく，多くの問題に取り組むことが必要であり，pgtracer はこれを可能にすることから，学習支援や授業改善に役立てることができる．

3. 学習支援ツール pgtracer の概要

3.1 学習支援ツール pgtracer が出題する問題

学習支援ツール pgtracer はプログラムとトレース表に対する穴埋め問題（**図 1**）を出題する．トレース表はプログラム実行のそれぞれのステップにおける変数と出力の値を示している．穴埋め問題は，プログラム，トレース表，プログラム用マスク，トレース表用マスクの 4 つの XML ファイルから構成される．プログラムとトレース表の分離により，入力値を変更することで，1 つのプログラムから複数のトレース表を作成できる．また，プログラム用マスクを分離することで，1 つのプログラムに対し複数の穴埋めパターンを作成できる．トレース表も同様である．1 組のプログラムとトレース表に対し，どのプログラム用マスクやトレース表用マスクを組み合わせるかで，難易度の異なる複数の問題を作成できる．

3.2　システムの機能

学習支援ツール pgtracer の機能には，穴埋め問題の作成機能，穴埋め問題の出題・採点機能，ログ収集・分析機能がある．

(1)　穴埋め問題作成機能

学習支援ツール pgtracer は，プログラムと入力ファイルを用いてプログラムとトレース表の XML ファイルを作成する．入力ファイルは標準入力値が記述されており，プログラムによっては，不要な場合もある．プログラムとトレース表の穴埋め個所はマウス操作によって設定でき，pgtracer がプログラム用およびトレース表用マスクの XML ファイルを作成する．プログラムに設定できる穴埋めは，トークンまたは文，トレース表に設定できる穴埋めは，変数値，ステップ，変数名である．文献 [3], [13] により穴埋めによって問題の難易度を調整できることが分かっており，穴埋め個所の組合せも活用して問題難易度を調節できる．作成された 4 種類の XML ファイルから構成された穴埋め問題は，問題データベースに保存される．

(2)　穴埋め問題出題・採点機能

データベースに保存された穴埋め問題を一覧表示し，学生が選択した問題を表示する．学習支援ツール pgtracer で設定できる問題オプションは，出題モード，正答の表示/非表示，問題分析機能の表示/非表示，穴埋め選択時の当該ステップの色付けの有無がある．出題モードは，解答を提出したのちに正誤の判定を行う試験モードと，学生が解答を入力するたびに，正解プログラムとの文字列比較によりその正誤を判定する自習モードがある．ここで，正答が正解プログラムで記述された文字列が 1 つだけではない場合，文字列比較だけでは正誤が判断できない．そこで，pgtracer は解答の提出後に穴埋めしたプログラムをコンパイル・実行し，トレース表と比較する．これによって，pgtracer は正解プログラムと一致しない解答であっても正しく正誤を判定できる．

(3)　ログ収集・分析機能

学生が穴埋めを行うたびに，pgtracer は解答，正答，解答所要時間をログとして収集する．学習支援ツール pgtracer は収集したログの分析機能として，学習履歴一覧，学生ごとの分析機能，問題ごとの分析機能，穴埋めごとの分析機能，解答プロセス分析機能を提供する．

ステップ	プログラム
	//キーボードから10個の整数を入力し，その合計と平均を表示する．
	//平均は小数点第2位まで表示する
	# include < stdio.h >
	int main ( ) {
1	int i ; //繰り返しの制御用変数
2	int num , sum = ; //入力した数，合計を格納
3	avg ; //平均を格納
	//整数の入力と合計への加算
4	(i = 0 ; i < 10 ; i ++) {
4.1	scanf ("%d" , &num) ;
4.2	sum num ;
	}
	//平均の計算
5	avg = ( ) sum / 10 ;
	//結果の表示
6	printf ("合計＝%d\n" , sum) ;
7	printf ("平均＝%.2f\n" , avg) ;
8	return 0 ;
	}

ルーチン	ステップ	main i	main num	main sum	main avg	出力
main	1	?				
main	2	?	?	0		
main	3	?	?	0	?	
main	4		?	0	?	
main	4.1		11	0	?	
main	4.2		11	11	?	
main	4		11	11	?	
main	4.1		22	11	?	
main	4.2		22	33	?	
main	4		22	33	?	
main	4.1		33	33	?	
main	4.2		33	66	?	
...中略...						
main	4	9	99	495	?	
main	4.1	9	110	495	?	
main	4.2	9	110	605	?	
main	5		110	605	60.5	
main	6		110	605	60.5	合計 605
main	7		110	605	60.5	平均 60.50
main	8		110	605	60.5	

図 1　pgtracer が出題する問題

Fig. 1　A fill-in-the-blank question provided by pgtracer.

3.3　教育プロセス

図 2 に pgtracer を用いたプログラミング教育プロセスを示す．まず，教員はあらかじめ作成した問題プログラムと入力ファイルをもとに，pgtracer を用いて穴埋め問題を作成する．出題の目的により，自習モードと試験モードを選択できる．学生は，問題一覧から問題を選び解答する．

問題解答後，学生は学習履歴や利用者全体の平均や分布

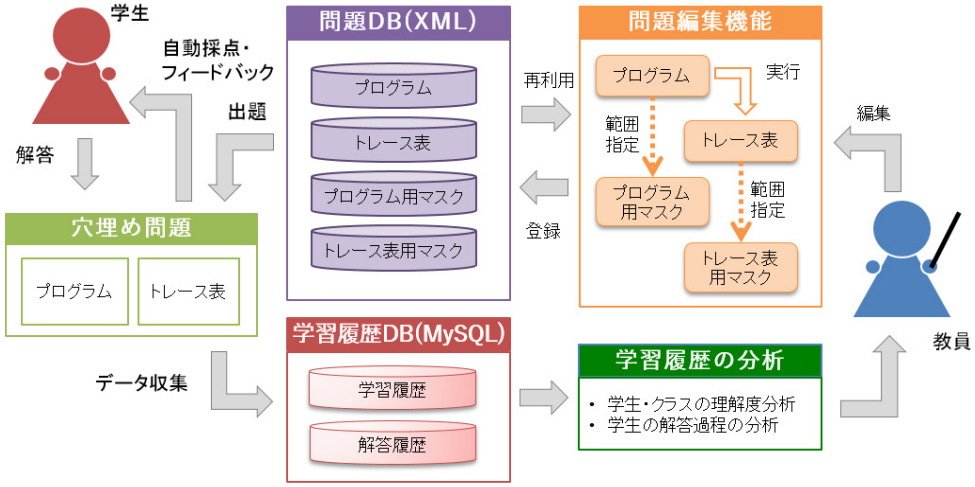

図 2　pgtracer を用いたプログラミング教育プロセス

Fig. 2　Programming education process utilizing pgtracer.

を閲覧でき，自身の学習状況を確認できる．教員は，学生ごと，問題ごと，穴埋めごとに正解率や解答所要時間を確認できる．さらに，穴埋めごとに学生の誤答の種類やその割合について確認できる．また学生の解答手順を再現することができる．教員はこれらの分析機能の結果を学生個人の指導や授業へのフィードバックに利用できる．

4. 2016 年度実践

本章では，熊本高専八代キャンパスにおいて 2016 年度後期に開講された 2 年次開講科目「情報工学基礎」で実施した自学習課題の提供について報告する．被験者である学生は 128 名である．学生は機械知能システム工学科，建築社会デザイン工学科，生物化学システム工学科のいずれかに所属しており，情報工学を専門としていない．

4.1 2016 年度実施科目「基礎情報工学」

「基礎情報工学」の達成目標は以下の 3 項目である．

(1) コンピュータの基礎知識を説明できる．
(2) Visual C++の開発環境を使うことができる．
(3) 基本的なプログラムを作成できる．

授業スケジュールを**表 1** に示す．学習支援ツール pgtracer による自学習課題の提供は後期に行った．評価方法は試験 80%，課題 20%である．課題は通年で 10 個出題され，プログラム，フローチャート，実行結果の提出とともに口頭試問を行った．出題された課題を**表 2** に示す．口頭試問に合格すれば，それぞれ 100 点として評価する．授業は，教員による説明の後，学生がサンプルプログラムの動作確認や演習問題を各自で取り組む方式であった．

試験は後期中間試験，学年末試験の 2 回，紙ベースで実施された．後期中間試験において pgtracer で出題した問題のうち，1 問を部分的に出題した．その配点は後期中間試験全体の 4%であった．

表 1　基礎情報工学の授業スケジュール
Table 1　Lecture plan of fundamental of computer science.

前期		後期	
週	内容	週	内容
1	コンピュータの基本	1	1 次元配列
2	数値の表現	2	1 次元配
3	フローチャート	3	2 次元配列
4	定数，変数，代入	4	2 次元配列
5	printf, scanf	5	ポインタ
6	型と演算	6	ポインタ
7	条件分岐	7	関数
8	前期中間試験	8	後期中間試験
9	条件分岐	9	関数
10	for	10	関数
11	while	11	変数のスコープ
12	do while	12	ファイル
13	break, continue, switch	13	構造体
14	課題実習	14	構造体
15	前期末試験	15	学年末試験

表 2　2016 年度に出題された課題
Table 2　Assignment of the class in 2016.

No	問題
1	身長（整数）と体重（実数）を入力し，標準体重を表示
2	2 つの整数を入力し，大きい数から小さい数を引いた値を表示
3	整数を入力し，「正」，「負」，「0」のいずれかを表示
4	3 つの整数値を入力し，最大値を表示
5	整数を入力し，その値によって表示を変える．（フローチャートは提示）
6	点数（整数）を入力し，評価（A, B, C, D，入力エラー）を表示
7	段数を入力し，＊を使ってその段数のピラミッドを表示
8	入力した文字列の中の大文字，小文字，数字の数を表示
9	整数を入力し，桁数の逆順から表示
10	整数を入力し，その桁数を表示

4.2 自学習課題提供の狙いと問題作成の方針

学習支援ツール pgtracer を活用した自学習課題の提供の狙いとして，学習時間の確保，変数のトレースやプログラムに対する理解の向上，授業内の説明だけでは定着が難しい，変数や関数の名前付けやインデンテーションなどが適切に記述されているプログラムを読む機会の増加がある．これらの狙いを達成するために，問題作成方針を以下のように定めた．

(1) 自習モードとする．
(2) 1 回の課題の問題数は 3 つとする．
(3) 問題は授業の内容に合わせ，いくつかのプログラムは授業で用いたプログラムと似たものとする．
(4) プログラムの穴埋めはトークンまたは文の一部分とする．
(5) トレース表については，前後のトレース表の値がヒントとなるため，問題とする穴埋めの前後のステップも合わせて穴埋めとする．
(6) 変数名，コメント，インデントを適切に記述し，手本となるプログラムを提示する．

方針 (1)，(2) は，継続的な利用を促進し，学習時間の確保を狙っている．2016 年度以前の実験により，pgtracer の自習モードで正誤が即時に判明することが，学生の学習意欲を高めることが分かっている [4]．また，1 回の問題数を制限して，学生の負担感の低減を狙っている．

方針 (3)，(4) より変数のトレースや基本的な文法の使い方を確認する問題とする．学習支援ツール pgtracer は，プログラムとトレース表を併記しており，学習者が穴埋め箇所を選択すると，対応するステップ行が色付け表示されるため，処理の流れの理解を支援できる．また，手本となるプログラムを多く読ませることで，適切な名前付けやインデンテーションがなされたプログラムを読む機会の増加を狙って方針 (6) を定めた．

表 3　2016 年度に pgtracer を用いて提供した自学習課題
Table 3　Questions assigned using pgtracer in 2016 class.

問題	内容
(1)-1	入力した 2 つの整数値の和，差，積，除，剰余を表示
(1)-2	入力した整数値の絶対値と偶数か奇数かを表示
(1)-3	入力した整数の 1 乗から 5 乗までを表示
(2)-1	入力した 10 個の整数値の合計と平均を表示
(2)-2	10 個の整数を入力した順に配列に格納し，逆順に表示
(2)-3	大文字なら小文字，小文字なら大文字に変換して表示
(3)-1	正数が 5 個になるまでに入力された負数の個数を表示
(3)-2	2 つの文字を連結して連結後の文字列を表示
(3)-3	2 次元配列に格納された座標のうち原点から最も遠い座標を表示
(4)-1	2 の 0 乗から 8 乗を配列に格納し，指定された整数乗の答を表示
(4)-2	”end”と入力されるまで入力した文字列の長さを表示
(4)-3	2 次元配列に格納された三角形の辺の長さの合計を表示
(5)-1	(3)-3 と同じプログラムで変数のコメントなし
(5)-2	2 次元配列に格納された 3 つの座標のうち原点から最も遠い点の座標を表示
(5)-3	ポインタを使って変数に代入，表示
(6)-1	(4)-3 と同じプログラムで変数のコメントなし
(6)-2	ポインタを使って，円の半径から面積を求め，表示
(6)-3	ポインタを使って，配列要素を指定し，要素を表示
(7)-1	整数 n を入力して n 回表示. n を引数とする関数を使う.
(7)-2	3 つの座標を入力して原点からの距離を表示．距離を求める関数を使う
(7)-3	関数を使って累乗を計算する
(8)-1	(3)-1 と同じプログラムでコードのコメントなし
(8)-2	名前と得点を入力し，得点に応じてメッセージを表示．メッセージ表示は関数を使う
(8)-3	いくつかの関数の中から適切な関数の呼び出し
(9)-1	階乗を求める関数を使って，順列を求め表示
(9)-2	秒から（時，分，秒）へ変換，（時，分，秒）から秒へ変換する．関数を使って求める.
(9)-3	配列のソート．配列要素の入れ替えは swap 関数を使う
(10)-1	(6)-2 と同じプログラムで変数のコメントなし
(10)-2	(9)-1 と同じプログラムでコードのコメントなし
(10)-3	配列を使った宝探しゲーム．あたりを判定する関数を使う.

2016 年度以前の実験において，変数名が長い，配列要素数が多いなどの場合にトレース表が大きくなり，問題全体を閲覧できないことが分かった．ブラウザの縮小表示機能で，問題全体を表示させることはできるが，学生は pgtracer のバグと認識したり，煩わしさを感じたりと pgtracer に対する信頼性の低下を引き起こし，pgtracer の利用を中断する傾向が見られた．そこで，方針 (6) を遵守しながら，問題全体が閲覧可能な程度に変数名や配列の要素数を決定した．また，1 問を 100 点とし，各穴埋めの配点はすべて同じとした．

自学習課題の問題作成および問題難易度やコメントの確認は，著者らのグループが行った．問題プログラムは授業の内容や進度に合わせたものを採用した．

10 回分の授業について，全 30 問の問題を作成した（**表 3**）．1 回あたりの自学習課題（3 問）を作成するのに要した時間は平均 2 時間であった．内訳は，問題プログラムの作成に 1 時間，プログラムやトレース表のマスク作成に 1 時間であった．

4.3　自学習課題の提供

2016 年度は，最初の 2 週間の授業にて最後の 15 分ずつを使って pgtracer へのユーザ登録および pgtracer の使い方を説明した．しかし，時間内にユーザ登録が終了しない学生やトレース表の概念が理解できない学生がいた．その後は完全な自学習課題とし，授業中に自学習課題を行う時間を設けなかった．また，解答期限も設定しなかった．自学習課題の公開は学内メールで連絡し，授業において自学習課題実施の呼びかけは行っていない．

4.4　アンケート

最後に pgtracer による自学習課題に関するアンケートを実施した．アンケートの項目は，(1) 基礎情報工学の課題に要する 1 週間あたりの平均時間，(2) その他の課題に要する 1 週間あたりの平均時間，(3) 解答した問題数，(4) 問題の難易度，(5) pgtracer はプログラミング学習に有用か，(6) 課題に取り組まなかった理由である．

4.5　実践方法の考察

学生の解答率を分析すると，初回でも 60%に達しておらず，その後解答率は徐々に減少し，7 回以降はほとんどの学生が解答していない．自学習課題を成績に反映していないこと，自学習課題の実施について継続的な呼びかけができなかったことが原因と考えた．

アンケートでは，トレース表に関する項目は設けなかったが，自由記述欄や学生のインタービューから「トレース表の見方が分かりづらかった」,「トレース表の見方が分からなかったのでトレース表の穴埋めは解答していない」などの回答が見られた．また，トレース表の穴埋めの未解答も多く，トレース表が何を意味しているのか，その概念の理解が不足していると推測できた．これは，当初の説明の時間を含め，トレース表を説明する時間が不足していたためと考えた．

以上より，2016 年度の実践方法では解答率が低く，トレース表の概念の理解不足を起因とする未解答が多いなど，pgtracer の解答が学生のプログラミング能力を正しく反映できているとはいえない．そこで，解答率を上げ，解答データの精度を上げるために，2017 年度では，実践方法の改善を行った．

5.　2017 年度の実践

本章では，熊本高専八代キャンパスにおいて 2017 年度後期に開講された 1 年次科目「プログラミング基礎 I」で実施した自学習課題の提供について報告する．被験者は 2016

表 4　プログラミング基礎 I の授業スケジュール
Table 4　Lecture plan of fundamental of programming I.

後期			
週	内容	問題数	備考
1	プログラムの実行	-	pgtracer のユーザ登録
2	定数，変数，代入	2	pgtracer の説明
3	printf，演算子	2	第 1 回アンケート
4	フローチャート	3	
5	scanf, getchar	3	第 1 回小テスト
6	条件分岐（for）	3	課題 1
7	前期中間試験	-	
8	条件分岐	3	第 2 回アンケート
9	switch	2	
10	条件分岐の入れ子	3	
11	for 文	3	第 2 回小テスト
12	while 文，do while 文	3	課題 2
13	break, continue	3	
14	課題実習	2	
15	前期末試験	-	
16	まとめ		第 3 回アンケート

年度と同様の学科に所属する 130 名である．

5.1　2017 年度実施科目「プログラミング基礎 I」

プログラミング基礎 I の達成目標は，以下の 5 項目である．プログラムを理解するためには，変数の動きがトレースできることが重要と考え，達成目標に追加した．

(1) コンピュータ内での数値や文字の表現方法を説明でき，問題に適したデータ構造を設計できる．
(2) C プログラムの作成から実行までの処理ができる．
(3) 変数と代入，標準入出力を用いたプログラムが作成できる．
(4) 条件分岐を含むプログラムを作成でき，変数の動きをトレースできる．
(5) 反復処理を含むプログラムを作成でき，変数の動きをトレースできる．

授業スケジュールを**表 4** に示す．また，課題や小テスト，アンケート，自学習課題の問題数もあわせて示す．本科目を履修する学生は，情報工学を専門としていない．このため，データ型は int 型と double 型，char 型のみを扱うなど内容を絞った．1 回の授業では，3〜4 課題の演習を実施し，ペアワークにより演習内容を相互確認した．

また，トレース表の理解を促進するため，授業において変数の変化を説明する際にトレース表と同様の表を用いた．

評価方法は試験 60%，課題 20%，小テスト 20%である．試験は，本キャンパスで利用可能な e-learning システムを用いた．2 回の試験のそれぞれで，pgtracer を用いた自学習課題のなかから 2 問ずつを選び，問題のキャプチャ画像を用いて出題した．これらの問題の試験全体における割合は 10%である．小テストは，自学習課題のなかから 3 問を選んで，授業中に pgtracer を利用して実施した．

同じ問題を出題することで，自学習課題への取り組みが良い学生は，小テストや試験において高い得点がとれるような仕組みとし，学生の自学習課題へのモチベーションの向上を図るとともに pgtracer の自学習課題への取り組みを科目の成績に反映する．

課題は 2 個出題し，プログラム，フローチャート，実行結果の提出とともに，口頭試問を行った．口頭試問ではプログラムの理解に加え，変数の名前やインデントなどのプログラミングスタイルについても確認を行い，試問に合格すれば 100 点として評価する．出題した課題の内容は以下のとおりである．

(1) 2 つの抵抗値と電圧，接続方法（直列または並列）を入力して，その合成抵抗を求める．接続方法は s または p の文字で指定する．
(2) 5 つの実数を入力し，入力した実数に対応する数の「*」を表示する．ここで，printf(”*”); を用いる．

5.2　問題作成の方針

2016 年度の実践によって，プログラムとトレース表の穴埋めが混在する問題は，学生が難しいと感じることが分かった．このため，2017 年度の実践では，学生の継続した利用のための負荷軽減を狙い，以下の方針も加えた．また，出題の意図を明確にするために，問題ごとに狙いを設定した．

(1) 1 つの問題における穴埋めの数を少なく設定する．
(2) 基本的にプログラムとトレース表のいずれかの穴埋めとする．ただし，応用問題として，両方に穴埋めを設定した問題をいくつか出題する．

トレース表の穴埋めについては，問題としたい変数値の前後にも穴埋めを設定している．これは，前後の変数値がヒントとなることを避けるためである．これらの穴埋めについては，穴埋め数としてカウントしない．

2016 年度と同様に著者のグループによって，問題作成と妥当性の確認を行った．問題プログラムは授業内容に沿ったプログラムを扱った．12 回分の授業について，1 回あたり 2〜3 問で全 32 問の問題を作成した．出題した問題の一覧を**表 5** に示す．また，問題 (5)-1，問題 (12)-1（**図 3**，**図 4**）とその情報（**表 6**，**表 7**）を示す．

5.3　自学習課題の提供

2017 年度は，pgtracer へのユーザ登録および pgtracer の使い方の説明は，それぞれ 5 分増やして 20 分ずつとした．これにより，すべての学生が時間内にユーザ登録を完了できた．また，トレース表の概念を理解できない学生に対応するため，授業内で変数の変化の説明や演習問題にトレース表を用いることで，トレース表の概念の理解向上を狙った．また，授業時間内に pgtracer の問題に取り組む時間を確保できた回があった．

表 5　2017 年度に pgtracer を用いて提供した自学習課題
Table 5　Questions assigned using pgtracer in 2017 class.

問題	内容
(1)-1	関連のない代入文
(1)-2	給与支給額を計算する
(2)-1	関連のない代入文
(2)-2	成績の表示
(3)-1	3 桁の各位の数の合計
(3)-2	面積と円周の長さの計算
(3)-3	○時○分を分，時間に換算
(4)-1	平均を求める
(4)-2	関連のない入力文
(4)-3	文字列の指定した場所の文字を変換
(5)-1	商品の値段から送料込の代金を計算
(5)-2	関連のない if 文
(5)-3	成績を判定する
(6)-1	得点から階級を求める
(6)-2	会員資格，回数から入場料を求める
(6)-3	月から四季を求める（else-if)
(7)-1	四則演算子
(7)-2	月から四季を求める（switch)
(8)-1	関連のない for 文
(8)-2	整数の和
(8)-3	入力した，正と負の数を数える
(9)-1	*を使った図の表示
(9)-2	11 段から 19 段の九九を表示
(9)-3	整数 n について，2^n を求める
(10)-1	while と do-while の違い
(10)-2	0 以下の実数が入力されるまでに入力された実数の和を求める
(10)-3	正の数を 5 個入力するまでに入力した負の数の個数を求める
(11)-1	入力した数の絶対値を求める．0 を入力したら終了
(11)-2	10 個の実数を入力し，正の数の個数と合計を求める
(11)-3	正の整数が入力されたときに素数かを判定する
(12)-1	*を使った図の表示
(12)-2	整数の各桁を足し，1 桁になるまで繰り返す

自学習課題の公開は授業中に連絡し，あわせて自学習課題の実施を呼びかけた．解答期限は次の授業までとした．解答期限内に実施しなかった場合のペナルティーはない．

5.4　アンケート

学期の初期，中間試験後，期末試験後の計 3 回，アンケートを行った．アンケート項目は，(1) pgtracer の使い方の理解度，(2) トレース表の理解度，(3) 問題の難易度，(4) 穴埋めの量，(5) 1 回の自学習課題に要した平均時間，(6) その他の課題に要した 1 週間あたりの平均時間，(7) プログラミングへの興味や意欲などである．トレース表の理解を向上するための取り組みを評価するためにアンケート項目 (2) を追加した．

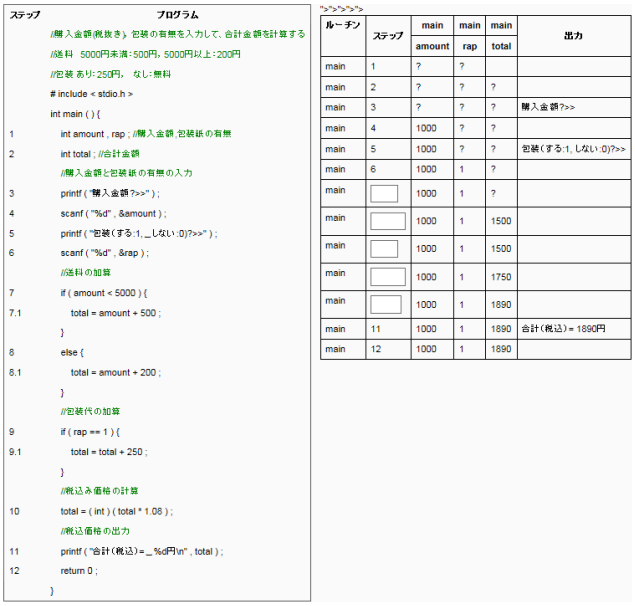

図 3　問題 (5)-1
Fig. 3　Question (5)-1.

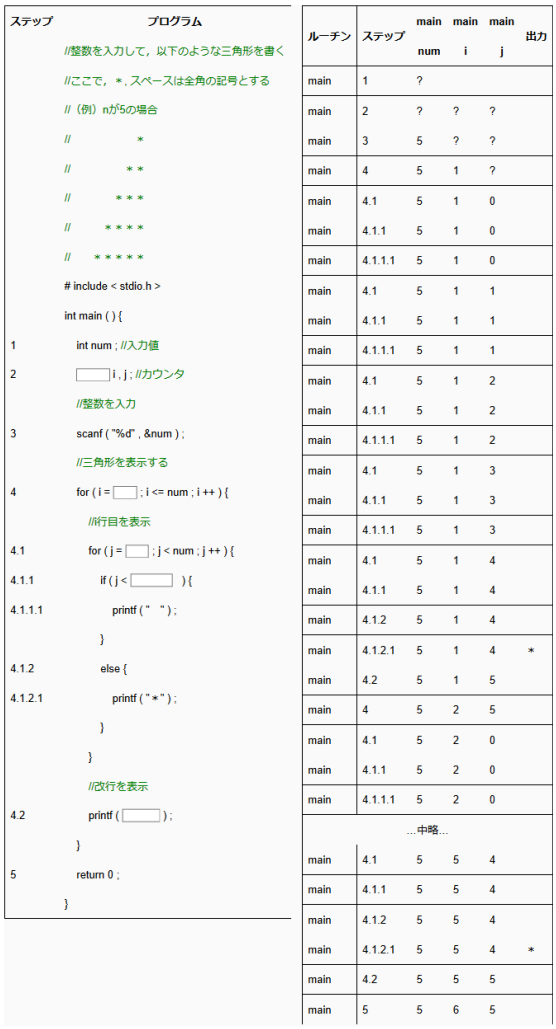

図 4　問題 (12)-1
Fig. 4　Question (12)-1.

表 6　問題 (5)-1 の概要
Table 6　Outline of Question (5)-1.

内容	商品の値段から送料込の代金を計算
狙い	if 文の動作がトレースできる
穴埋めの場所（個数）	トレース表（5）

表 7　問題 (12)-1 の概要
Table 7　Outline of Question (12)-1.

内容	*を使った図の表示
狙い	if 文を含む 2 重ループが書ける
穴埋めの場所（個数）	プログラム（5）

6. 考察

6.1 学生の学習行動に関する考察

図 5 は，各自学習課題の平均解答率を示している．ここで平均解答率とは，1 回の自学習課題で出題された 2 問ないし 3 問の解答率の平均を示す．2016 年度は 10 回，2017 年度は 12 回自学習課題を出題している．2016 年度は全体的に解答率が低い．後期中間試験で pgtracer の問題を範囲としたため，後期中間試験前の自学習課題 (6) までは解答する学生がいたが，それ以降は急激に減少し，すべての問題を解答した学生は 1 名であった．一方，2017 年度は，最も解答率の低いものが最後の自学習課題 (12) で 80%と，徐々に減少する傾向はあるが，ほとんどで 90%を超えている．

図 6 は，2017 年度の延べ解答数を週ごとにまとめたものである．項目軸の日付は，解答状況をチェックした日付である．あわせて試験，小テストの日付を記載している．対象とするクラスが 3 つのため，小テストの日付が 3 つとなっている．グラフを見ると解答数が多くなったのは，2 回の小テストと試験を含む期間であることが分かる．学生が小テストや試験に備えて勉強した様子が分かる．また，小テストや試験の直前の週は取り組みが少ないことが分かる．学生が小テストや試験の直前に勉強をしようと考え，授業後に解答をしなかったと推測できる．

解答履歴を精査すると解答所要時間が 2 分以下で正解率が 0%の解答が多く見つかった．これらの解答が試験直前に実施されていることから，試験に備えて実際には解答せずに，正答を確認していると考えられる．このような解答の割合は，2016 年度の 16.4%から 2017 年度は 5.6%と減少している．また，2017 年度の実践では，授業中においても演習を早く終わらせた学生が，自主的に pgtracer の自学習課題に取り組む様子も観察できた．

以上より，2017 年度の実践で行った科目の成績に自学習課題の取り組みを反映させること，授業内で実施に対する声掛けを継続的に行うといった工夫が，学生の学習意欲を高め，また継続させることにつながり，学生の解答行動が改善されたことが分かる．

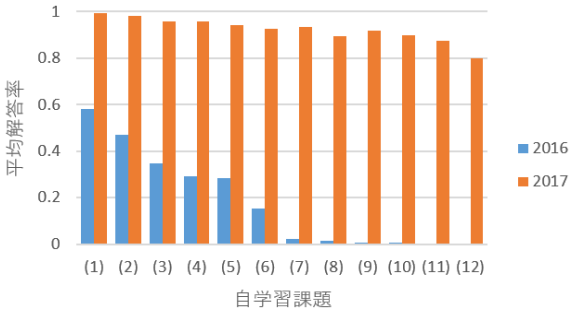

図 5　自学習課題ごとの平均解答率
Fig. 5　Average of the answer ratio of each week.

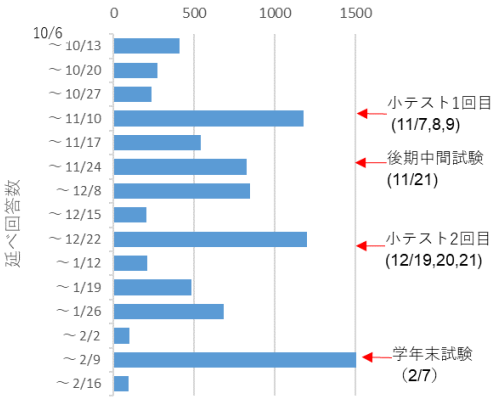

図 6　延べ解答数の推移
Fig. 6　Total number of answers.

表 8　自学習課題の初回と最高得点時の平均正解率
Table 8　Average of right answer ratio of the first attempt and the attempt with the highest score.

	2016 年度	2017 年度
初回解答時平均正解率	62.6	84.6
最高得点時平均正解率	82.2	99.0
解答回数の平均	1.50	1.85

表 8 に自学習課題に出題した問題の初回解答時と最高得点時の正解率および解答回数の平均を示す．2016 年度は解答率が 10%を超えている自学習課題 (1) から自学習課題 (6) までを対象とする．初回解答時の平均正解率に比べ，最高得点時の平均正解率が 2016 年度で 19.6%，2017 年度で 14.4%ほど向上していることが分かる．また，解答回数の平均を見ると 2016 年度は 1.50 回，2017 年度は 1.85 回とほぼ同等の結果が得られた．なお，2016 年度，2017 年度ともに解答回数についての指示は行っていない．このことから，いったん自学習課題に取り組むことができれば，学生は高い正解率を目指して問題の解答を繰り返すことが分かった．つまり，その問題を理解するための行動をとることが分かった．

6.2　トレース表の理解に関する考察

図 7 にアンケート項目 (2) トレース表の理解度についての結果を示している．1 回目のアンケート時でも 80%の学生が理解できたと回答していることから，改善の効果が認められる．授業の開始時に pgtracer の使い方やトレース表の説明時間を増やしたことや，授業の中でトレース表を用いて説明や演習を行うことが，学生のトレース表の理解に役立ったことが分かった．また，授業においてペアワークによって学生同士の教えあいが行われ，教員が特に理解が不足している学生に対応できたことも有効であったと考える．

表 9 にトレース表の理解についてのアンケート結果と定期試験の成績を示す．ここで，選択肢の「しっかりと理解できた」と「まったく理解できていない」が逆の意味を持ち，「ほぼ理解できた」と「あまり理解できていない」も同様に逆の意味を持つことは明確である．そのため，一般的な学生は，「理解できた」があまり高くない理解度を意味すると解釈すると推測できる．よって，各選択肢の意味は明確で，ある程度の対称性も保たれていることから，考察の内容に影響を与えるような誤差は生じないと考えられる．

後期中間では，「まったく理解できていない」と回答した学生の平均点が「あまり理解できない」と回答した学生の平均点を 9 点上回っている．これは「まったく理解できていない」と回答した学生の中に 92 点を取っている学生が 1 名いるためであるが，この学生は pgtracer による問題を 1 問も回答していないため，アンケート回答の信頼度が低いと考えられる．そのため，この学生を除いて以下の考察を行う．この場合，「まったく理解できていない」と回答した学生の平均点は 67.0 となる．よって，トレース表の理解が高いほど，試験成績が高くなることが分かる．

また，試験成績の差が統計的に有意かを調べるために分散が等しくないと仮定した t 検定を行った．ここで，試験成績は，2 回の試験の平均を用いた．アンケート結果については，「しっかりと理解できた」を 5，「まったく理解できていない」を 1 として 5 段階に数値化を行い，定期試験の直後に実施した第 2 回と第 3 回アンケートの平均を用いた．この平均が「理解できた」に相当する 3 以上の学生をトレース表が理解できた学生群，3 未満の学生をトレース表が理解できていない学生群とした．t 検定の結果，両側検定における p 値は 0.03 となり，有意差があるといえる．以上のことから，トレース表の理解とプログラミング能力の間には統計的に有意な関係があると考えられる．

6.3　プログラムの理解度に関する考察

表 10 に，問題解答数および 2 つの試験の成績変化の平均を示す．表 10 より，多くの問題を解答した学生の得点が高いことが分かる．学習支援ツール pgtracer の問題を解答することで，学習時間が増えたことが成績の向上につながったと考える．解答数 0 の学生の得点が高い．該当する学生は 1 名であり，プログラムを理解しているので pgtracer の問題に取り組む必要がなかったと話しているため，授業開始時においてすでに習熟度が高い学生であると推測されることから考察から除外できる．

問題ごとの解答率は 80.0～99.2%の範囲に，初回受験時の平均正解率は 63.4～98.2%の範囲に，所要時間の中央値は 63.5～436.5 秒の範囲に分布している．ほとんどの問題で平均正解率が 80%を超えている．また，アンケートの結果を見ても 3 回すべてで 60%程度の学生が「ちょうどよい」と答えており，学生にとって取り組みやすい問題であったことが分かる．

平均正解率が 70%未満の問題は，問題 (2)-2，問題 (3)-3，問題 (10)-3，問題 (12)-2 であった．これらの問題は，解答所要時間の中央値も大きい．いずれの問題もプログラムの穴埋めをする問題であり，プログラムの穴埋めは学生にとって難しいことが分かる．また，これらの問題のほかに解答所要時間の中央値が 300 秒を超えた問題は，問題 (8)-1

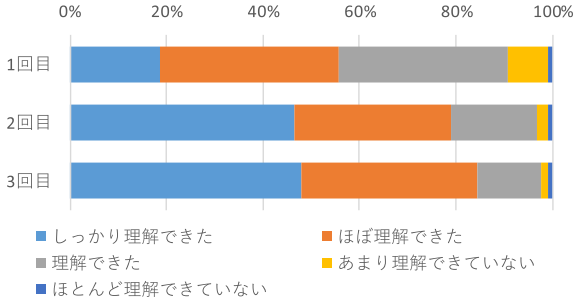

図 7　トレース表の理解度
Fig. 7　Understanding of trace table.

表 9　トレース表の理解と定期試験成績の関係
Table 9　Relationship between understanding of trace table and examination score.

アンケート「トレース表は理解できましたか」	後期中間		学年末	
	人数	平均	人数	平均
しっかりと理解できた	39	80.3	36	75.6
ほぼ理解できた	48	78.3	54	76.0
理解できた	30	79.1	30	68.5
あまり理解できていない	9	67.0	8	64.5
まったく理解できていない	3	76.0	1	51.0

表 10　問題解答数と試験成績の関係
Table 10　Difference of average exam scores.

解答数 x	人数（人）	後期中間（点）	学年末（点）	差（点）
x =32	97	79.85	75.34	-4.51
17<= x <32	25	74.76	67.12	-7.64
1<= x <17	7	67.57	61.14	-6.43
x = 0	1	92.00	82.00	-10.00

表 11　初回解答時の正答率が 80%以下の穴埋めの解答状況
Table 11　Answers at the blanks whose right answer ratio at the first trial is less than 80%.

問題	穴埋め個所（　　内が正答）	初回正答率	平均解答回数	誤答( )内の数字は解答数．記載のないものは解答数 1
(12)-1	if(j< num-i){ for 文の入れ子	44.7%	5.8	未解答(23), num(5), 5(5), 4(3), i(3), 3, num+i, num/10, sum-1,　=num
(3)-3	int hour; 変数定義	74.4%	4.3	未解答(14), int hour(4), 1, 1 時間 30 分, 90, hour, hour=?, int, int<hour>, printf, sum hour
(9)-3	pow = pow * 2; within a for satement	76.3%	5.0	未解答(16), 1024(3), 10, 2*n, 32, 4, j++, n*2, n^2, pow*j
(10)-3	}whike(cnt_pos < 5); do-while 文	76.7%	4.2	未解答(5), cnt_pos==5(3), cnt_pos<6(2), "%d".cnt_neg, break, cnt_pos, cnt_pos>5, cnt_pos>=5, cnt_pos<=5, num<0, num<5, num<8, pos==5
(12)-1	printf("¥n"); for 文の入れ子	76.8%	3.5	未解答(4), ""(3), ¥n(3), ", "", "%d", ":", %, *, 改行

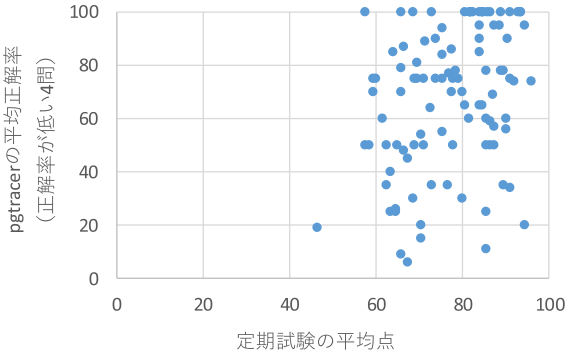

図 8　pgtracer の平均正解率と定期試験の平均点の関係
Fig. 8　Relationship between the averages of pgtracer right answer ratio and examination score.

であった．問題 (8)-1 は，for 文を扱っており，トレース表の穴埋めとなっている．問題 (9)-1 では，2 重ループ文を扱っており，同様にトレース表の穴埋めであるが，これも 283.0 秒と時間がかかっている．これより繰り返し文のトレースに時間を必要としていることが分かる．

図 8 では，正解率が低かった 4 つの問題（問題 (2)-2，問題 (3)-3，問題 (10)-3，問題 (12)-2）について，初回受験時の正解率について定期試験成績との関係を示している．ここで，正解率の平均には，4 つの問題すべてに解答した学生 105 人のデータを用いている．

これらの問題の平均正解率と定期試験成績には，0.32 と弱い正の相関があり平均解答所要時間と定期試験成績には，0.07 とほとんど相関がない．アンケートの自由記述をみると，「友人に教えてもらいながら解いた」との回答があり，理解度が低い学生も高い正解率が得られたと考えられる．また，初回受験日が期限日以降の学生の正解率には 0%が多い．これは，小テストや試験に備えて，問題の解答はせずに，閲覧のみしたと推測できる．

平均解答所要時間について，すべての問題の平均解答所要時間と定期試験成績には −0.13 の負の相関が見られる．これは，成績の良い学生ほど短時間で問題を解き終わっていると考えられる．しかし，難しい問題では成績の良い学生でも時間をかけて問題を解くほうが，正解率が高くなると考えられる．これらの問題のほかに相関係数が正となった問題は問題 (5)-1，問題 (8)-3，問題 (9)-3，問題 (12)-1 であった．問題 (5)-1 を除くとすべての問題で平均正解率が 80%以下であった．

表 11 は，正答率が低い穴埋めを示している．いずれもプログラムに対する穴埋めである．問題 (3)-3 を除いて反復処理に関する問題であった．また最も正解率の低い穴埋めは，2 重 for 文の中にある if 文の条件式を問う問題であった．このことから，反復処理のプログラム作成が学生にとって難しいことが分かった．また問題 (3)-3 については，変数定義部の穴埋めが初出であり，学生が何を解答すればよいのかを混乱した可能性がある．

初回正答率が最も低かった穴埋めを含む問題 (12)-1 について，学生の解答行動を確認した結果，以下の特徴的な解答が見られた．

- 学生 1 は，試験の平均点が 94.5 点と理解度の高い学生である．変数 num や i を利用して何度か入力を繰り返しているが，最後には正解している．
- 学生 2 は，最終的には正解しているものの，最初は定数を入力している．28 回目の入力に長い時間をかけた後，変数 num と i を利用することに気づいている．この学生の試験の平均点は 80.5 点であった．
- 学生 3 は，最初に定数を入力し，次に変数 num を利用することには気づいたものの，i を利用することには気づかずに，正解にたどり着けなかった．この学生の試験の平均点が 72.5 点である．
- 学生 4 は試験の平均点が 66.0 点と理解度の低い学生である．この学生は，まったく正答とはかけ離れた解答を繰り返しており，問題プログラムを理解できていないと考えられる．

これより，学生のプログラミング理解度によって，学生が正解を導く思考の過程に違いがあることが分かる．

表 12　アンケート結果「プログラミングに対する興味や意欲」
Table 12　Student motivation of programming at each examination.

	後期中間			学年末		
アンケートの回答	第 2 回アンケート人数	試験の平均点	期限内解答率の平均	第 3 回アンケート人数	試験の平均点	期限内解答率の平均
とてもある	36	79.9	84.0%	28	76.2	73.0%
ややある	63	79.8	92.1%	60	75.3	77.1%
どちらでもない	25	73.5	82.8%	28	70.9	58.0%
ややない	3	76.0	89.7%	7	60.1	53.1%
ほとんどない	2	64.0	42.3%	6	65.2	67.2%

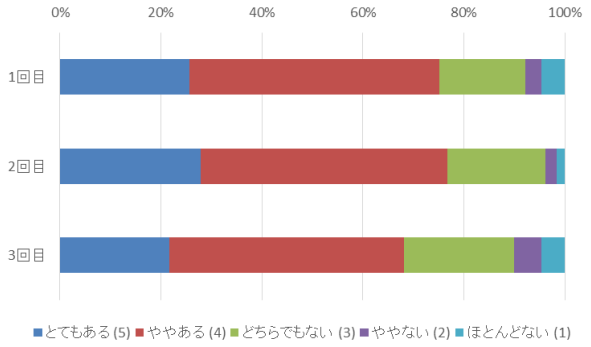

図 9　学生の興味や意欲
Fig. 9　Interest and intrinsic motivation of the students.

6.4　学生の意欲に関する考察

2017 年度の実践では，学生のプログラミング対する興味や意欲との関連を検討するため，アンケート項目にこれを問う設問を追加した．

図 9 にアンケートの結果を示す．「とてもある」または「ややある」と回答した学生の割合は，第 1 回目のアンケートでは 75.2%，2 回目では 76.7%と少し増加し，3 回目では 68.2%と減少している．これよりほとんどの学生は，興味を持ってプログラミングに取り組んでいることが分かる．2 回目までの学習範囲は，選択構造までであるが，3 回目は反復構造が含まれる．3 回目で低下したのは，学生にとって反復構造が難しく感じられたためだと考えられる．

また，**表 12** にプログラミングへの興味と意欲と定期試験成績，自学習課題の実施状況を示す．プログラミングへの興味や意欲については，各試験の直後のアンケートの結果を示している．後期中間試験成績との相関は 0.24，学年末試験成績との相関は 0.27 と弱い正の相関がみられる．後期中間試験では，「とてもある」と回答した学生の平均点が最も高く 79.9 点，「ほとんどない」と回答した学生の平均点は最も低く 64.0 点とその差は 15.9 点であった．期限内解答率をみると，「ほとんどない」と回答した学生の解答率が最も低い．このことから，学生の興味と宿題への取り組み，試験の成績の間には弱い関係があると考えられる．

学年末試験では，「とてもある」と回答した学生の平均点が最も高く 76.2 点，最も低かったのは 60.1 点で「ややない」と回答した学生の平均点であった．期限内解答率をみると，「ややない」と回答した学生の解答率が最も低く，後期中間と学年末試験の差を見ると，第 3 回目で「ややない」と回答した学生の平均点の低下が最も大きかった．この結果は，「ほとんどない」と回答した学生の成績が最も低かった後期中間試験の結果と異なっている．

表 13　第 3 回アンケートで「ややない」以下を回答した学生数と解答状況
Table 13　The number of student answered less than “rather weak” in 3rd questionnaire and answering activity.

	第 3 回回答が「ほとんどない」		第 3 回回答が「ややない」	
アンケート回	1 回目	2 回目	1 回目	2 回目
とてもある	0	1	0	0
ややある	1	2	2	2
どちらでもない	2	2	4	4
ややない	0	0	1	1
ほとんどない	3	1	0	0
期限内解答率	67.2%		53.1%	
平均正解率	68.0%		65.4%	

そこで，第 3 回アンケートで「ややない」，「ほとんどない」と回答した学生について，さらに分析を行った．**表 13** に第 1 回目，2 回目のアンケート結果を示している．第 3 回で「ほとんどない」と回答した学生の半数は，第 1 回目でも「ほとんどない」と回答しており，途中興味が出てきた傾向は見られるが，授業期間を通じて，プログラミングへの興味や意欲が低かったと推測できる．

一方，第 3 回目で「ややない」と回答した学生は，1 回目では，1 人を除き，「どちらでもない」以上を回答しており，授業開始時に比べ，興味や意欲が低下した学生であることが分かる．期限内の解答数の平均を見ると，第 3 回目で「ほとんどない」と回答した学生のほうが上回っており，第 3 回目で「ややない」と回答した学生は，プログラミングへの興味や意欲が低下したことが原因で，取り組みがおろそかになり，成績が低下したと考えられる．このことから，最初から興味や意欲が低い学生よりも，途中から低下していった学生のほうが，成績の低下が大きいことが

分かった．

6.5　学習支援ツール pgtracer の機能に関する考察

学習支援ツール pgtracer は，学生ごと，問題ごとの解答履歴を一覧表示する機能を持つ．2017 年度には pgtracer の分析機能を用いて解答状況をモニターし，解答状況が悪い学生を抽出した．このような学生に対してメールを送るなどの対応を行う予定であったが，授業担当者の多忙により 1 回しか対応ができなかった．このため，解答状況が少ない学生へのメッセージの自動送信機能が望まれる．

2017 年度には解答期限を設けたが，期限内に解答したかどうかは，定期的に解答履歴を取得して確認した．解答期日によるログのフィルタリング機能を追加することで学生の期限内の解答状況を確認できると考える．また，問題の解答期限を設定し，解答が遅れた問題に遅延マーク，新規に出題された問題に New マークを付加することで，学生自身が進捗状況を視覚的に把握できると考える．

また，pgtracer は解答を提出すると正誤と正答が表示される．しかし，正答が表示されるだけでは，学生がなぜ間違ったのか，正答を導く思考の過程に気づくことが難しい．そこで，正答とあわせて解説も表示することで，学生の理解を促進できると考える．

トレース表の表示において，トレース表の横幅が大きい場合に，一部が表示されず，ブラウザの縮小機能を用いて全体を表示している．現在は，変数名や関数名の名前を短くする，配列を小さくするなどして対応しているが，UI による工夫についても検討したい．

問題作成の点からは，作成した XML ファイルに対する削除，名前の変更，サブフォルダなどを用いたファイルの整理機能が望まれる．また，マスクを設定する際に，マスク部分を指定した後，マスク設定ボタンを押下するが，マスク部分とボタンの位置を近づけることで問題作成の効率が向上すると考える．

7.　おわりに

本稿では，pgtracer を自学習に活用したプログラミング科目の実践について報告するとともに，学生の学習行動について検討した．

プログラムの理解に関しては，復習として継続的に取り組むことで，試験直前に一括して取り組むよりも学生の理解が進むことが分かった．また，トレース表の概念の理解が高い学生ほどプログラミングの理解も高いことが分かった．さらに，学生の理解度によって，学生が正答を導く思考の過程に違いがあることが分かった．

学生の学習行動について検討することで，自学習課題の成果を科目の成績に反映することによって，学生の学習行動が向上することが分かった．また，プログラミングに対し，最初から興味や意欲が低い学生よりも途中から興味や意欲が低下した学生の方が成績の低下が大きいことが分かった．

今後の課題としては，収集したログ，特に解答を導出する手順や誤答を分析することで，学生の理解度を客観的に推測するための手法について検討したい．また，2018 年度前期開講科目「プログラミング基礎 II」においても自学習課題の提供を実践している．そこでの解答を確認したところ，プログラムの穴埋めにおいて，本来ならば変数名を解答すべきところで，定数を解答して正解となっている解答を見つけた．このような解答を誤りとするために，複数の入力値を用いて実行結果が一致することを確認するなどの手法を 6.5 節で述べた pgtracer の機能拡張とともに検討したい．

謝辞　本研究は JSPS 科研費 17K01036 の助成を受けたものです．

参考文献

[1] 林文部科学大臣：Society5.0 に向けた人材育成の推進，未来投資会議資料 (2018)，入手先 ⟨http://www.kantei.go.jp/jp/singi/keizaisaisei/miraitoshikaigi/dai16/siryou6.pdf⟩.

[2] 掛下哲郎，柳田　峻，太田康介：穴埋め問題を用いたプログラミング教育支援ツール pgtracer の開発と評価，情報処理学会論文誌：教育とコンピュータ，Vol.2, No.2, pp.20–36 (2016).

[3] Kakeshita, T. and Ohta, K.: Student log analysis functions for web-based programming education support tool pgtracer，情報処理学会論文誌：教育とコンピュータ，Vol.5, No.2, pp.456–468 (2019).

[4] Murata, M. and Kakeshita, T.: Analysis method of student achievement level utilizing web-based programming education support tool pgtracer, *5th International Conference on Learning Technologies and Learning Environment* (*LTLE 2016*), pp.316–321 (2016).

[5] 村田美友紀，嘉藤直子，掛下哲郎：プログラミング学習支援ツール pgtracer を自学習に活用したプログラミング科目の実践報告，情報処理学会情報教育シンポジウム，6-2 (2018).

[6] Kakeshita, T. and Murata, M.: Application of Programming Education Support Tool pgtracer for Homework Assignment, *International Journal of Learning Technologies and Learning Environments*, Vol.1, No.1, pp.40–61 (2018).

[7] Fu, X., Shimada, A., Ogata, A., Taniguti, Y. and Suehiro, D.: Real-time learning analytics for C programming language courses, *ACM International Conference Proceeding Series*, pp.280–288 (2017).

[8] Hering, W., Huppertz, H., Kramer, B.J., et al.: On benefits of interactive online learning in higher distance education: Case study in the context of programming education, *eLmL - International Conference on Mobile, Hybrid, and On-line Learning 2014*, pp.57–62 (2014).

[9] Gotthardt, K., Kramer, B.J., Magenheim, J. and Neugebauer, J.: On benefits of interactive online learning in higher distance education: Repeating a learning analytics project in the context of programming education, *International Journal on Advances in Life Sciences*, Vol.6, No.3–4, pp.350–363 (2014).

[10] Malliarakis, C., Satratzimi, M. and Xinogalos, S.: Inte-

grating learning analytics in an educational MMORPG for computer programming, Proc. *IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014*, pp.233–237 (2014).
[11] 石和田圭，森本康彦，中村勝一ほか：プログラミング演習における学習状況推定のためのソースコード編集過程分析手法の開発，電子情報通信学会技術研究報告，No.116, No.438, pp.75–80 (2017).
[12] 井垣　宏，齊藤　俊，井上亮文ほか：プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案，情報処理学会論文誌，Vol.54, No.1, pp.330–339 (2013).
[13] 榎本　命，宮澤芳光，宮寺庸造，森本康彦：項目反応理論に基づき学習支援を行うプログラミング演習用穴あきワークシートシステムの評価，教育システム情報学会研究報告，Vol.31, No.7, E5-1 (2017).

村田 美友紀　（正会員）

佐賀大学大学院工学系研究科修了．博士（工学）．現在，熊本高等専門学校拠点化プロジェクト系准教授．情報専門教育，データベースに関する研究に従事．情報処理教育シンポジウム SSS2018 優秀発表賞受賞．

嘉藤 直子　（正会員）

佐賀大学大学院工学系研究科修了．博士（工学）．現在，有明工業高等専門学校一般教育科准教授．ソフトウェア工学，情報専門教育に関する研究に従事．1999 年度情報処理学会九州支部奨励賞受賞．

掛下 哲郎　（正会員）

九州大学大学院工学研究科修了．工学博士．現在，佐賀大学理工学部准教授．ソフトウェア工学，データベース，情報専門教育に関する研究に従事．2012 年情報処理学会優秀教育賞受賞．電子情報通信学会，ACM，IEEE-CS 等会員．

情報処理学会論文誌　教育とコンピュータ　Vol.6　No.2　25–37 (June 2020)

正誤表

下記の箇所に誤りがございました．お詫びして訂正いたします．

訂正箇所	誤	正
25 ページ	受付日 2019 年 3 月 11 日	受付日 2019 年 3 月 1 日