

論文

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

西田 知博^{1,a)} 原田 章² 中西 通雄³ 松浦 敏雄⁴

受付日 2016年3月24日, 再受付日 2016年8月3日,
採録日 2016年11月3日

概要: 筆者らは、10年にわたり初学者向けプログラミング学習環境であるPENを用いてプログラミングの入門教育を行ってきた。本論文では、情報科学を専門としない学部の全6クラスの情報リテラシ科目のなかで実施した、90分×4回のプログラミング演習について述べる。当初は、キーボードから数字を読み込んで計算結果をディスプレイ画面に出力するような例題から始めて逐次・条件分岐・繰返しを学び、最後に図形を描画する単一のコースウェアを用いていた。その2年後、図形描画の例題から始めるタイプのコースウェアも開発し、クラスごとにどちらかのコースウェアを用いて演習を行ってきた。本論文では、2011年度から4年間の授業におけるアンケートおよび試験成績を統計的に分析した。その結果、図形描画をともなう例題を扱う方が、繰返しのようになつまずきやすい学習内容でも、理解度や楽しさを下げることなく学習できていることが分かった。

キーワード: プログラミング入門教育, 図形描画プログラミング, プログラミング入門コースウェア

Comparison between Two Types of Courseware for Introductory Programming

TOMOHIRO NISHIDA^{1,a)} AKIRA HARADA² MICHIO NAKANISHI³ TOSHIO MATSUURA⁴

Received: March 24, 2016, Revised: August 3, 2016,
Accepted: November 3, 2016

Abstract: We have been teaching introductory programming for ten years using Programming Environment for Novices, PEN. This paper describes the programming exercise which consists of four sessions of 90-minute lessons in computer literacy. The exercise is performed in six classes, and each class has 45 to 60 non-computer science students. Originally, the courseware for the exercise started with programs that read numbers from the keyboard and write calculated results to the display. The first three lessons of the courseware were ‘traditional’ and devoted to sequential processing, conditional branch and repetition, but were followed by one lesson that focused on drawing circles and boxes using graph functions. Two years later, the authors developed another courseware, in which the first three lessons started with drawing circles and boxes, followed by one lesson of calculation of input values and writing results to the display. Each class has adopted one of the two types of courseware. This paper discusses the four-year experience of the programming exercise from 2011 to 2014. Statistical analysis of the questionnaires and exam scores shows that students using the courseware that started with drawing diagrams seemed to learn in a more receptive manner and more enjoyably than the students who used other courseware.

Keywords: introductory programming education, drawing with code, introductory programming courseware

¹ 大阪学院大学
Osaka Gakuin University, Suita, Osaka 564-8511, Japan
² 追手門学院大学
Otemon Gakuin University, Ibaraki, Osaka 567-8502, Japan
³ 大阪工業大学

Osaka Institute of Technology, Hirakata, Osaka 573-0196, Japan
⁴ 大阪市立大学
Osaka City University, Osaka 558-8585, Japan
a) nishida@ogu.ac.jp

1. はじめに

情報処理学会では、1992年の「大学等における一般情報処理教育の在り方に関する調査研究」の報告をはじめとして、「プログラミング」を学ぶことの重要性を表明している [1], [2], [3]. ここでいうプログラミングの教育は、情報を専門とする学生だけのものではなく、情報を専門としない学部学科も含めたすべての学生に、コンピュータがプログラムによってどのように動作するかを体験的に認識させることが主眼である。つまり、実際にプログラミングを通じてプログラムの誤りによって期待どおりの結果が出ないことなどを体験することで、コンピュータは書かれたプログラムを忠実に実行するという本質を理解できる。しかし、限られた時間でこの本質を学習できるようにするためには、適切な環境を用意することが必要である。

そこで我々は、比較的短い時間でこの目的に沿った学習ができる初学者向けプログラミング環境 PEN (Programming Environment for Novices) [4] を開発した。そして、いくつかの大学および高校において、PEN を活用した授業実践を重ね、教材の開発・改良を行ってきた [5].

PEN はタイプミスによる構文エラーの発生を低減させるための入力支援機能ボタンを備えたプログラムの編集機能に加え、プログラムの一時停止・1行実行が可能な実行制御機能、実行中の各変数の値を表示する機能など、プログラミングの実行の様子を把握しやすくするためのデバッグ機能を備えており、初学者が短時間でプログラミングを学びやすい環境を提供している。プログラムは大学入試センターの科目「情報関係基礎」で用いられている手順記述言語 DNCL [6] を基本とした言語で記述する。DNCL を記述言語とした理由は、日本語をベースとした言語であり、特別な説明なしで入試で用いられているということからも、分かりやすさの点では優れていると考えたためである。DNCL は試験用言語であるため、以下の点を拡張した xDNCL を PEN の記述言語としている。

- (1) 変数や型を意識させるための変数宣言の追加
変数型としては、整数、実数、文字列の3つを用意した。
- (2) 命令の補強
多分岐処理や、繰返しからの脱出、型変換、注釈文などの命令を追加した。
- (3) 組み込み関数の追加
文字列操作関数、数学関数、図形描画関数、ファイル I/O 関数を用意した。

なお、xDNCL の言語マニュアルおよび、図形描画マニュアルは授業資料として配布している [7].

著者らは、2006年度から大阪大学の文学部と人間科学部の各3クラスにおいて、1年生対象の情報リテラシ科目に PEN を用いたプログラミングを組み入れ、10年にわたる

実践を行ってきた。図形描画の例題を学習させることで学生の学習意欲が上がることは経験的には感じられていた。これを客観的に評価するために、2008年に図形描画を中心とした例題から始めるコースウェアと、キーボードから数字を入力して、計算結果をコンソールに出力する例題から始める伝統的なコースウェアとを比較し、学生の学習意欲や理解度に差が生じているかどうかを調査した [8]. しかし、結果として、2つのコースウェアにおける学生の意識および理解度に関する明確な有意差を見出すことはできなかった。この原因として、以下のことが考えられる。

- (1) 2つのコースウェアに難易度の差があった。
- (2) 各クラスの授業の進め方に違いがあった。
- (3) 各例題に費やす時間がクラスによって異なっていた。

そこで、これらの点について考慮し、教材の構成と授業の進め方を見直し、2011年度からコースウェア間の難易度と学習内容を統一し、実施するコースウェアが特定の学部や教員に偏らないように、教員が担当するコースウェアを1~2年ごとに交換するなど工夫して再調査を行ってきた。本論文では2011年度から2014年度まで4年にわたって調査し、アンケートによる学習効果の調査と期末試験による学習達成度の調査を分析した結果について述べる。

2. 関連研究

初学者にプログラミングをどのように教えるべきかについては多くの研究がなされている。文献 [9] では、「カリキュラム」「教授法」「言語選択」「ツール」の4つのカテゴリに分け、これまでの初学者へのプログラミング教育の研究を広範囲に分析している。文献 [10] では、プログラミングの敷居を下げるという視点で、プログラミング言語や環境を分類している。これらはいずれも広範囲なサーベイではあるが、プログラミング学習のコースウェアの違いを比較するような研究は見当たらない。

初学者教育におけるプログラミング環境としては、視覚的な対象物を動かすことによりプログラム学習を進める、Scratch [11]、ドリトル [12]、ビスケット [13] などが広く用いられている。最近では、プログラミング教育活動である Hour of Code のチュートリアル [14] で、親しみやすいキャラクターを使い、学習を進めていくようなコースウェアも提供されている。これらのプログラミング環境で用いられる教材は、プログラムの実行結果を視覚的に確認できるため、初学者がプログラミング課題に取り組みやすく、学習しやすくなることが示唆されている。また、視覚的な教材を開発することに着目した研究としては、岡本らが画面上だけではなく実物を使った「視覚的顕在化」に着目して、マイコンボードを使用したコースウェアの開発とアンケートをもとにした授業の評価を行っている [15]. このように実行結果を視覚的に出力するプログラミング教材の開発や有効性について多くの研究がなされているが、それらと従来の

テキストベースの出力をするプログラミング教材との比較研究は見当たらない。

3. 授業の概要

3.1 対象とするクラス

大阪大学では、全学部において1年次における情報教育科目が開講されており、文学部では1994年以降、人間科学部では1995年以降、情報教育科目「情報活用基礎」が必修科目となっている。本研究では、この2学部の授業における実践を対象とした。

両学部の「情報活用基礎」は1年次前期に開講され、それぞれ受講者を3つのクラスに分け(表1)、各クラスを1名の教員と2名もしくは3名のティーチングアシスタント(TA)で担当している。担当教員は互いに授業の進行状況や講義資料の情報を交換しながら、同じ計算機環境で授業を進めている。プログラミング演習は終盤のテーマとし、両学部とも90分×4回を割り当てている。

3.2 コースウェアの概要

コースウェアは、特徴の異なる2種類を用意した[16]。1つは、キーボードからの入力に対して、計算結果をコンソールに文字列として出力する形式の演習を繰り返すもの(従来型と呼ぶ)であり、もう1つは、描画用のウィンドウに図形を描画する形式の演習を繰り返すもの(図形型と呼ぶ)である。ただし、4回目の授業では、従来型のコースウェアで図形描画の演習を取り上げ、図形型のコースウェアで従来型の演習を取り入れており、一方のコースウェアでしか学べない学習項目はないように配慮している。表1に各クラスでどちらの形式の授業を行ったかを示す。表2にそれぞれのコースウェアで取り上げたすべての例題と練習問題を示す*1。各コースウェアで取り上げた例題・練習問題は、それまでの数年の経験に基づいて、授業担当教員の間で相談して改良を加えてきたものである。特に、両コースウェア間の難易度の差をできるだけ小さくするよう

表1 各年度の受講者数と授業内容

Table 1 Numbers of students in each class.

年度	学部	第1教室	第2教室	第3教室	合計
2011	人間科学部	45 (図)	45 (図)	49 (従)	139
	文学部	59 (図)	64 (従)	59 (従)	182
2012	人間科学部	48 (従)	48 (従)	46 (図)	142
	文学部	56 (従)	69 (図)	56 (図)	181
2013	人間科学部	48 (従)	48 (図)	45 (図)	141
	文学部	60 (従)	63 (従)	59 (図)	182
2014	人間科学部	48 (図)	48 (従)	46 (従)	142
	文学部	60 (図)	62 (図)	60 (従)	182

(従)：従来型, (図)：図形型

*1 例題および練習問題の詳細は、文献[7]の授業資料(従来型)および授業資料(図形型)を参照されたい。

に配慮してきた。ただし、これはあくまで授業担当者の主観に基づいたものである。

表2 2014年度の授業時間配分(単位:分)

Table 2 Lecture time allocation table (2014).

(a) 従来型コースウェア

第1回	予定	文3	人2	人3
準備:プログラミングとは	25	20	22	16
例題 1.1 (入力の3倍)	20	20	20	26
練習 1.1 (長方形の面積)	10	10	11	15
例題 1.2 (3の倍数の判定)	10	10	7	7
例題 1.3 (3の倍数か否かの判定)	05	10	0	8
第2回	予定	文3	人2	人3
宿題の解説	10	10	9	10
練習 1.15 (合否判定・乱数)	15	15	10	13
例題 1.4 (成績判定)	10	10	10	14
練習 1.10 (うろう年)	15	17	11	16
例題 1.5 (数当てゲーム)	15	8	10	12
例題 1.6 (10回繰り返し)	10	9	6	10
第3回	予定	文3	人2	人3
宿題の解説	10	10	12	10
練習 1.15 (繰返・乱数の平均)	10	11	15	12
例題 1.7 (二重ループ)	06	5	10	7
練習 1.23 (□をn行目にn個)	14	13	12	11
例題 1.8 (繰返(2)-乱数)	05	3	5	3
例題 1.9 (実数-温度変換)	15	14	10	10
練習 1.29 (階乗)	15	19	13	20
第4回	予定	文3	人2	人3
宿題の解説	10	5	10	9
例題 1.10 (描画関数の使い方)	12	22	12	18
例題 1.11 (円の水平配置)	13	8	12	13
練習 1.33, 34 (円を並べる(1, 2))	17	13	16	16
例題 1.12 (円の描画・乱数)	08	4	8	8
練習 1.38, 39 (円の描画2・乱数)	10	12	12	12

(b) 図形型コースウェア

第1回	予定	文1	文2	人1
準備:プログラミングとは	25	20	20	25
例題 1.1 (緑の円)	30	17	20	30
練習 1.2 (長方形)	10	18	17	10
第2回	予定	文1	文2	人1
宿題の解説	10	11	12	10
例題 1.2 (3の倍数の判定)	15	15	7	17
例題 1.3 (3の倍数か否かの判定)	05	9	9	10
練習 1.5 (合否判定・乱数)	15	9	16	13
例題 1.4 (成績判定)	10	12	10	7
練習 1.9 (うろう年)	15	17	13	13
第3回	予定	文1	文2	人1
宿題の解説	10	10	7	10
例題 1.5 (半径当て・乱数)	15	18	12	18
例題 1.6 (繰返し・乱数)	10	14	8	12
練習 1.14, 15 (繰返・乱数)	15	12	16	15
例題 1.7 (円の水平配置)	10	10	8	10
第4回	予定	文1	文2	人1
宿題の解説	10	11	6	10
例題 1.8 (二重ループ)	08	13	9	10
練習 1.23 (n行目にn個並べる)	12	13	14	10
例題 1.9 (繰返(2)-乱数)	05	1	6	5
例題 1.10 (実数-座標の平均)	15	23	17	20
練習 1.27 (階乗)	15	18	28	15

3.3 授業の概要

以下、両コースウェアの各回の授業の進め方および実習時間の配分について説明する。

3.3.1 授業の進め方

授業は基本的に例題を解説し、その後その例題に関連した1つもしくは2つの練習問題に取り組ませることを繰り返した。以下に、従来型の例題 1.1 を例に説明する。

例題 1.1 キーボードから1つの整数を入力し、それを3倍した値を出力するプログラムを書いてみよう。

プログラム例：

```

1: 整数 a
2: 整数 b
3: 「整数を入力してください:」を表示する
4: a ← input()
5: b ← a * 3
6: b を表示する
    
```

毎回の授業に先立ち、各担当教員には「授業時間記録シート」[7]を配布している。図 1 にこのシートの一部を示す。

このシートは、各回の授業で実施予定の例題と練習問題に費やす時間の目安と、授業の進め方を簡潔に記している。

図 1 に記されているように、教員はPENの使い方を解説しつつ、受講者と一緒にプログラム例を入力する、次に「一行実行」によりプログラムを動作させつつプログラムの各行について説明する。説明する内容は、文献[7]の「授業資料」の各例題の下に書かれた解説と同じである。

このように、「授業時間記録シート」を用いることで、各クラスの授業の進め方をなるべく統一するように努めた。その後、通常はノーヒントであらかじめ決めておいた練習問題に取り組ませる。しばらく時間をおいて受講者の様子を見ながら、必要に応じてヒントを出している。授業で扱った例題および練習問題は表 2 のものがすべてである。授業資料[7]には、表 2 に掲載されていない例題も多数あるが、これらは練習問題を早くやり終えた受講者のためのオプション問題として用意している。

3.3.2 授業時間配分の調整

過去の授業経験から、担当教員の合意によって、それぞれの例題、練習問題などに割り当てる予定時間を定めている

20分：例題 1.1 (入力の3倍)

- ・PENの使い方を解説しつつ、「入力支援ボタン」を活用しながら、例題 1.1 のプログラムを入力する。(学生と一緒に…)
- ・実行させてみる
- ・一行実行させながら、プログラムの各行を解説する
- ・プログラムを保存する(ファイル名は ex1.01.pen)

図 1 授業時間記録シート (一部)

Fig. 1 Lecture time allocation and recording sheet.

る(表 2 参照)。各担当教員は、この予定時間にできるだけ沿って授業を進めたが、受講者の反応などから、必要に応じて個々の教員の判断で授業の進行速度を調整した。実際に各例題、練習問題に要した時間(2014年度)を表 2 に示している。表から分かるように、どのクラスもほぼ予定どおりの時間で授業を進めることができたと思われる。

例題および練習問題以外に要した時間は以下のとおりである。両コースウェアともに、初回の授業では、最初の25分(予定時間、以下同様)でプログラミング学お意義の説明と実習の準備作業を行っている。2回目から4回目の授業では、最初の10分で、前回の宿題について解説している。また、すべての回で授業の最後の10分程度でアンケートを実施した。上記にあげた時間以外はすべて実習(例題および練習問題)の時間であり、両コースウェア間での実習時間の差はない。

また、第4回を除いて毎回プログラム作成の宿題を課した(文献[7]参照)。宿題に要した時間は各回のアンケートで答えてもらっており、どの回もおおむね90%以上の受講者が30分~1時間以内と回答している。

4. 学習効果の検証

本研究は、コースウェアの違いが学生のプログラミング学習にどのような影響を与えるかについて検討することが目的である。そこで、学習効果について検証するために、各授業回の終了時にWeb上でアンケート調査を行った。また、学習の達成度は期末試験の成績を用いて検証した。本章では、分析対象者とアンケート調査の概要について説明し、学習効果についての分析および考察を行う。

4.1 分析対象者

分析対象となる授業の受講者数は表 1 に示すとおりである。4回の授業でのアンケート結果の変化と、期末試験の成績を分析するために、本研究では4回の授業に出席してその際に行ったすべてのアンケートに答え、かつ、期末試験を受けた学生のみを分析対象とした。表 3 にその人数内訳を示す。

4.2 調査の方法

授業アンケート調査の内容は以下のとおりである。

(1) 前回の宿題に対する達成度の自己評価*2

表 3 分析対象学生の内訳

Table 3 Numbers of participants.

年度	男性	女性	合計
2011	55	114	169
2012	50	79	129
2013	42	76	118
2014	66	109	175

*2 第1回アンケートにはこの項目はない。

表 4 授業に対する理解度の分散分析表

Table 4 ANOVA table: Self-evaluation of the understanding.

要因	df	平方和	平均平方	F
年度	3	33.0	11.01	3.799 *
授業方式	1	.5	.48	.167
年度 * 授業方式	3	31.7	10.56	3.645 *
誤差 1	583	1688.9	2.90	
授業回	3	68.0	22.66	40.408 ***
年度 * 授業回	9	29.0	3.22	5.741 ***
授業方式 * 授業回	3	186.9	62.28	111.049 ***
年度 * 授業方式 * 授業回	9	26.0	2.88	5.144 ***
誤差 2	1749	981.0	.56	

(* : $p < .05$, ** : $p < .01$, *** : $p < .001$)

(2) 授業で扱った個々の例題・練習問題に対する理解度や難易度の自己評価

(3) プログラミング内容に関する理解度の自己評価

(4) 授業に対する理解度や面白さの自己評価

(5) プログラミングに対する楽しさや容易さの自己評価

本研究では、授業を行った4回の回答結果について対応をとる必要があったため、ログイン名の入力を求めた。ただし、入力の強制は行わなかった。質問項目は質問文と5段階から7段階の回答選択肢によって構成されていた。

この調査は、毎回の授業の終了時に行った。回答者はWebページを用いてアンケートに回答する。回答に要する時間は10分程度であった。

4.3 授業に対する理解度

コースウェアの違いが学生に与える影響を評価するために、質問項目の中から、まず、授業に対する理解度について尋ねた項目について分析することとした。この項目の質問文は「今日の授業は理解できましたか?」で、6段階の回答選択肢を用意した。回答選択肢の内容とその数量化は、理解度が高いと判断された場合に値が高くなるよう、「理解できた=6, だいたい理解できた=5, どちらかという理解できた=4, どちらかという理解できなかった=3, あまり理解できなかった=2, 理解できなかった=1」とした。

この回答を従属変数とし、「年度(2011年・2012年・2013年・2014年の4水準)」、「授業方式(従来型授業・図形型授業の2水準)」、「授業回(第1回から第4回までの4水準)」を要因とする3要因分散分析を行った。要因「年度」と「授業方式」は受講生が年度で異なるので「対応なしの要因」、要因「授業回」は同一の受講生の分析であるので「対応ありの要因」であった。表4は分散分析の結果を表したものである。要因「授業方式」以外の主効果・1次交互作用・2次交互作用が有意水準5%もしくは0.1%で有意となった。また、図2は、授業に対する理解度の授業回別平均を授業方式および年度別に表したものである。値が高いほど、理解度が高いことを表している。

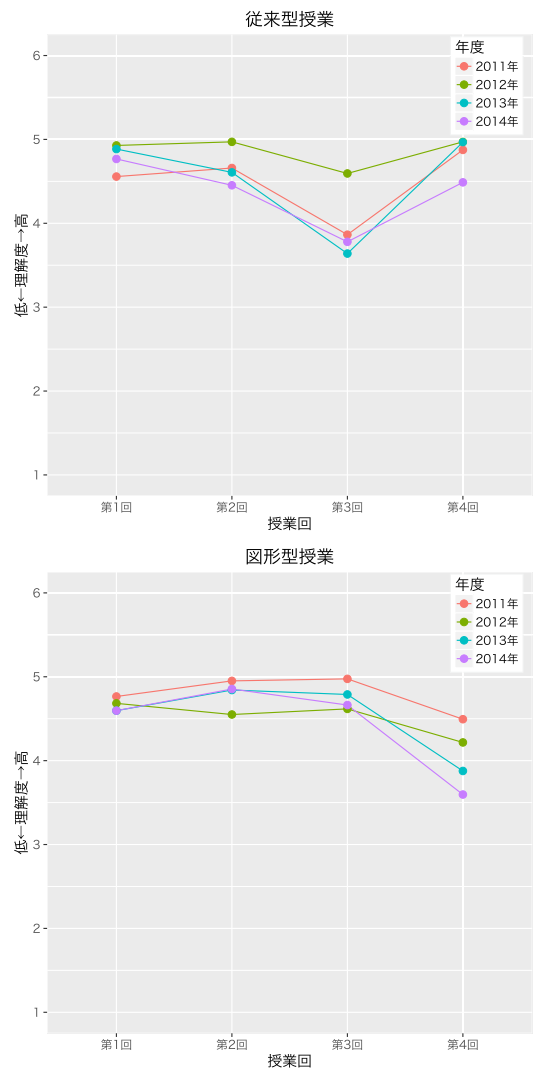


図 2 授業に対する理解度の変化

Fig. 2 Transitions of self-evaluation of the understanding.

表4より、2次交互作用が有意であることから、年度・授業方式・授業回によってさまざまな有意差があることが分かる。そこで、表2から、授業方式によって各授業回の平均値の変動(第1回から第2回, 第2回から第3回, 第3回から第4回の3つ)が有意であるかどうかについて年度ごとに検討した。統計的検定としては対応のある母平均の差の検定を用いたが、有意かどうかを判断する基準となる有意水準は、全体の危険率が5%となるようにBonferroni法を用いて調整した値を用いた*3。以下では、このような方法を「有意水準5%のBonferroni法」と呼ぶこととする。

その結果、従来型授業では、すべての年度で第2回から第3回の変動が有意であり、2012年以外の年度で第3回から第4回の変動が有意であった。図形型授業では、2012年以外の年度で第3回から第4回の変動が有意であった。

*3 この検定での有意水準は、授業方式2水準、年度4水準、各年度で比較する母平均の対が3個であったので、全部で24回の有意性判断を繰り返すことになる。したがって、全体で危険率を5%とするために、1つ1つの検定については、 $5\%/24 = 0.208\dots\%$ を有意水準として有意かどうかの判断を行った。

次に、各年度の従来型授業と図形型授業の平均値を授業回ごとに調べた。検定方法は対応のない母平均の差の検定で、有意性判断については有意水準5%のBonferroni法を用いて行った。その結果、すべての年度において第3回の平均値間に有意差が見られた。また、2011年以外の年度において第4回の平均値間に有意差が見られた。

検定の結果および図2の平均値変動を見ると、従来型授業では第3回の内容が他の回に比べて難しいと感じられたと思われる。一方、図形型授業では第4回の授業内容が他の回に比べて難しいと感じられたようである。従来型授業も図形型授業も、第3回の授業は「繰返し」を扱っていたが、すべての年度において、第3回の授業方式別平均値の間に有意差が見られたことから、2012年を除く他の年度は、図形型授業の方が繰返しについて理解できたと自己評価していることが分かった。これは同じ内容であっても、図形型授業の方が理解してもらいやすい場合があることを示している。

第4回では、従来型授業は図形描画の演習を行い、一方、図形型授業は最後の練習問題としてテキスト問題の演習を行った。図形型授業を受けている学生にとってはそれまでの内容と一線を画するものようにとらえられたために、難しいと感じたのではないと思われる。

4.4 プログラミングに対する楽しさ

コースウェアの違いが学生に与える影響を評価するために、プログラミングに対して楽しさを感じているかについて尋ねた項目についても分析した。この項目の質問文は「プログラミングはあなたにとって楽しいですか?」で、7段階の回答選択肢を用意した。回答選択肢の内容と数量化は、楽しいと判断された場合に値が高くなるよう、「とても楽しい=7、かなり楽しい=6、どちらかという楽しい=5、どちらでもない=4、どちらかという楽しくない=3、あまり楽しくはない=2、まったく楽しくはない=1」とした。

この回答値を従属変数とし、授業に対する理解度と同じ要因で3要因分散分析を行った。表5は分散分析の結果である。要因「授業方式」の主効果が有意水準5%で、要因「授業回」の主効果が有意水準0.1%で有意であった。また、要因「年度」と「授業回」の1次交互作用が有意水準5%で、要因「授業方式」と「授業回」の1次交互作用が有意水準0.1%で有意であった。また、図3は、プログラミングに対する楽しさの授業回別平均を授業方式および年度別に表したものである。値が高いほど、楽しさが高いことを表している。

本研究の目的は、授業方式の差異による影響を調べることにあるので、有意であった1次交互作用のうち、授業方式と授業回の1次交互作用について検討することとした。図4は、各授業回の4年間の平均値を授業方式別に表し

表5 プログラミングに対する楽しさの分散分析表

Table 5 ANOVA table: The degrees of enjoyment to program.

要因	df	平方和	平均平方	F
年度	3	50.0	16.65	2.218
授業方式	1	43.9	43.88	5.847 *
年度 * 授業方式	3	38.0	12.82	1.708
誤差 1	583	4376.0	7.51	
授業回	3	22.3	7.43	12.068 ***
年度 * 授業回	9	11.9	1.32	2.144 *
授業方式 * 授業回	3	51.0	16.99	27.585 ***
年度 * 授業方式 * 授業回	9	6.8	.75	1.224
誤差 2	1749	1077.1	.62	

(* : p < .05, ** : p < .01, *** : p < .001)

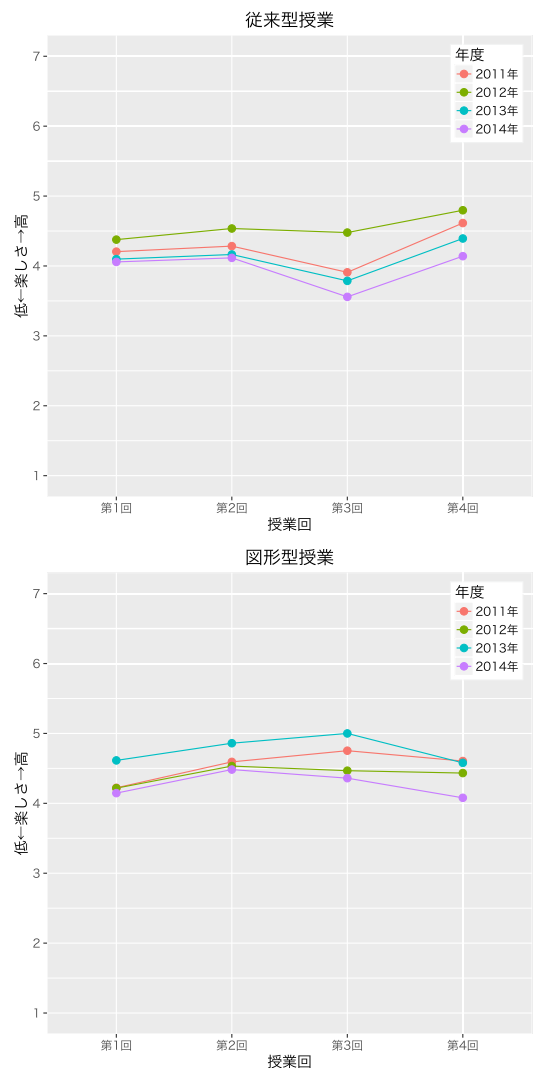


図3 プログラミングに対する楽しさの変化

Fig. 3 Transitions of the degrees of enjoyment to program.

たものである。そこで、各授業回において従来型授業と図形型授業の平均値間に有意差があるかどうかについて調べた。検定方法は対応のない母平均の差の検定で、有意性判断は有意水準5%のBonferroni法を用いて行った。その結果、第2回と第3回において平均値間に有意差が見られた。

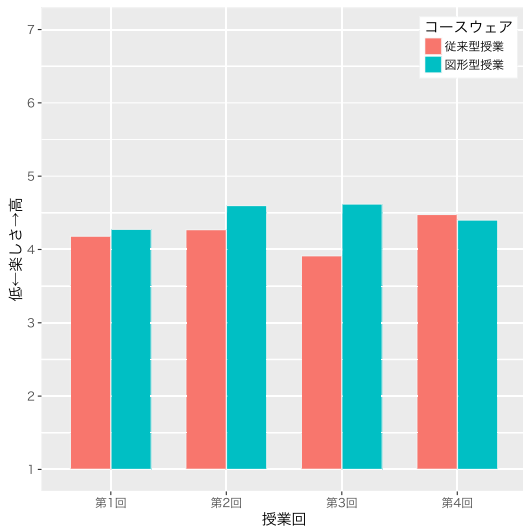


図 4 プログラミングに対する楽しさの平均値 (授業回, 授業方式別)
 Fig. 4 Averages of the degrees of enjoyment to program.

図 3, 図 4 より, 図形型授業では楽しさの平均値が第 2 回で大きく増加し, それが第 3 回でも維持されていることが分かる. 一方, 従来型授業では第 1 回と比べて第 2 回はほぼ同じで, 第 3 回は減少している. これらの回では, 条件分岐と繰返しの制御構造を学ぶため, プログラミングに対する難しさを感じ始め, 楽しさが低くなる可能性が考えられる. しかし, 図形型授業では図 2 に示すとおり理解度はほぼ変化せず, 楽しさが増加しているため, 図形型問題が学生に難しさを感じさせず, 楽しく課題に取り組ませる結果につながっていると考えられる. 一方, 従来型授業では 3 回目で楽しさが減少しているが, 4 回目に図形描画を学ぶことにより楽しさが向上し, 2 回目までと同水準もしくは高い値になっている. また, 図形型授業と比べてもほぼ同じ値となっている. このことより, 文字・数値の出力のみで演習を進め, 繰返しなどの難しい内容で楽しさが減っても, 図形描画の演習を行うことにより, その減少を補っていることが見て取れる.

5. 学習達成度の検証

本章では, 学習達成度の分析を行うために行った期末試験の概要について説明し, その結果の分析と考察を行う.

5.1 期末試験の方法

学習の達成度の測定は授業終了後の試験により行った. この試験は学期の期末試験を兼ねており, プログラミングに関する 3 問以外にも 2~3 の問題を出題している. これらの出題の影響は完全に排除できていないが, プログラミングに関する問題に解答する時間が十分確保されるように配慮した.

プログラミングに関する 3 問について, 年度ごとに問題内容は異なるが, 難易度はほぼ同じと考えており, 以下の

問題 1 以下のプログラムは, 図 A のような図形を描画するものである. ウィンドウの大きさは 450×450 . 点線は説明上の補助線であり 無視しても良い. 点線の間隔は, 50 ピクセルである. 空欄に入れるのに適当な語句・数を解答欄に書け.

```

1: 整数 n
2: gOpenWindow(《あ》)
3: gSetLineWidth(3)
4: n ← 1
5: n < 8 の間,
6: | gSetColor(255, 0, 0) /* 赤 */
7: | gDrawLine(50, 《い》, 《う》, 《え》)
8: | gSetLineColor(0, 0, 255) /* 青 */
9: | gDrawLine(《お》, 50, 《か》, 《き》)
10: | n ← (《く》)
11: を繰り返す
    
```

問題 2 $3+6+9+\dots$ と 3 から順に 3 の倍数を加えていき, その和が 20000 より大きくなったとき, 最後に加えた数を求めるプログラムを書け. また, 最後に加えた数も示せ.

問題 3 400×400 のウィンドウに, 乱数を用いて一様に分布する半径 3 の円を 2000 個描くプログラムを書け. ただし, 円の中心座標に応じて, 図 B のように色分けせよ.

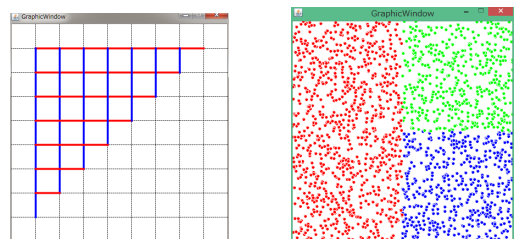


図 A 問題 1 の実行結果 図 B 問題 3 の実行結果

図 5 2014 年度の期末試験問題
 Fig. 5 The term-end examination in 2014.

点は共通している. 問題 1 は一部に空欄を含む図形描画を行うソースプログラムを提示し, その空欄を埋める問題であり, 問題 2 はコンソールにテキストの結果を出力する従来型の問題であり, 問題 3 は図形型の問題である. 問題 2, および, 問題 3 は, いずれも繰返しの中に条件分岐を含む構造のプログラムを作成する問題である. 満点は, 問題 1 (以降「穴埋め問題」と呼ぶ) が 5 点, 問題 2 (以降「テキスト問題」と呼ぶ) が 7 点, 問題 3 (以降「図形問題」と呼ぶ) が 8 点である. 2014 年度に文学部の学生対象の試験で出題した問題を図 5 に示す. 人間科学部に対しては, 数値などを一部変更したほぼ同じ問題を出題した.

試験中, 学生は PC を利用してプログラムの動作を確認することができ, 任意の Web ページを閲覧することもできる. また, 配布した資料を含めて, 任意の資料や書籍の閲覧も許されている. ただし, メールや SNS を用いて他の学生とコミュニケーションをとることは禁止した. プログラミングに関する問題について, 学生のほとんどは PC

表 6 試験結果の分散分析表

Table 6 ANOVA table: The scores of examinations.

(1) 穴埋め問題				
要因	df	平方和	平均平方	F
年度	3	83.7	27.89	12.351 ***
授業方式	1	4.3	4.33	1.961
年度 * 授業方式	3	29.0	9.67	4.281 **
誤差	583	1316.6	2.26	

(2) テキスト問題				
要因	df	平方和	平均平方	F
年度	3	246.6	82.20	11.138 ***
授業方式	1	343.6	343.57	46.556 ***
年度 * 授業方式	3	57.1	19.02	2.577
誤差	583	4302.4	7.38	

(3) 図形問題				
要因	df	平方和	平均平方	F
年度	3	69.9	23.30	3.109 *
授業方式	1	8.4	8.44	1.127
年度 * 授業方式	3	67.5	22.51	3.004 *
誤差	583	4368.6	7.49	

(* : $p < .05$, ** : $p < .01$, *** : $p < .001$)

でプログラムを作成・実行してから解答を記入していた。

5.2 期末試験の結果

表 6 は、各問題の得点を従属変数として、年度（2011年・2012年・2013年・2014年）と授業方式（従来型授業・図形型授業）を要因とする 2 要因分散分析を行った結果である。穴埋め問題（表 6(1)）では、年度の主効果が有意水準 0.1% で、年度と授業方式の交互作用が有意水準 1% で有意となった。テキスト問題（表 6(2)）では、年度と授業方式の主効果がいずれも有意水準 0.1% で有意となった。図形問題（表 6(3)）では、年度の主効果、年度と授業方式の交互作用が有意水準 5% で有意となった。図 6 は、3 つの問題それぞれについて、授業方式別の平均点を年度ごとに表したものである。

穴埋め問題では、交互作用が有意になったことから、年度によって授業方式の違いによる平均の差の傾向が異なると予想される。そこで、授業方式別平均値の違いを年度別に調べた。検定方法是对応のない母平均の差の検定で、有意性判断は有意水準 5% の Bonferroni 法を用いて行った。その結果、2014 年度のみ授業方式による差が見られた。図 6 の傾向から 2014 年度に関しては図形型授業の方が成績が良かったと考えられる。

テキスト問題では、交互作用が有意とはならず、主効果のみが有意となった。このことから、年度に関係なく授業方式の違いによる平均の差が見られた。図 6 の傾向と合わせて考えると、従来型授業の方が図形型授業よりも平均点が高いことが分かる。また、年度の主効果についても有意

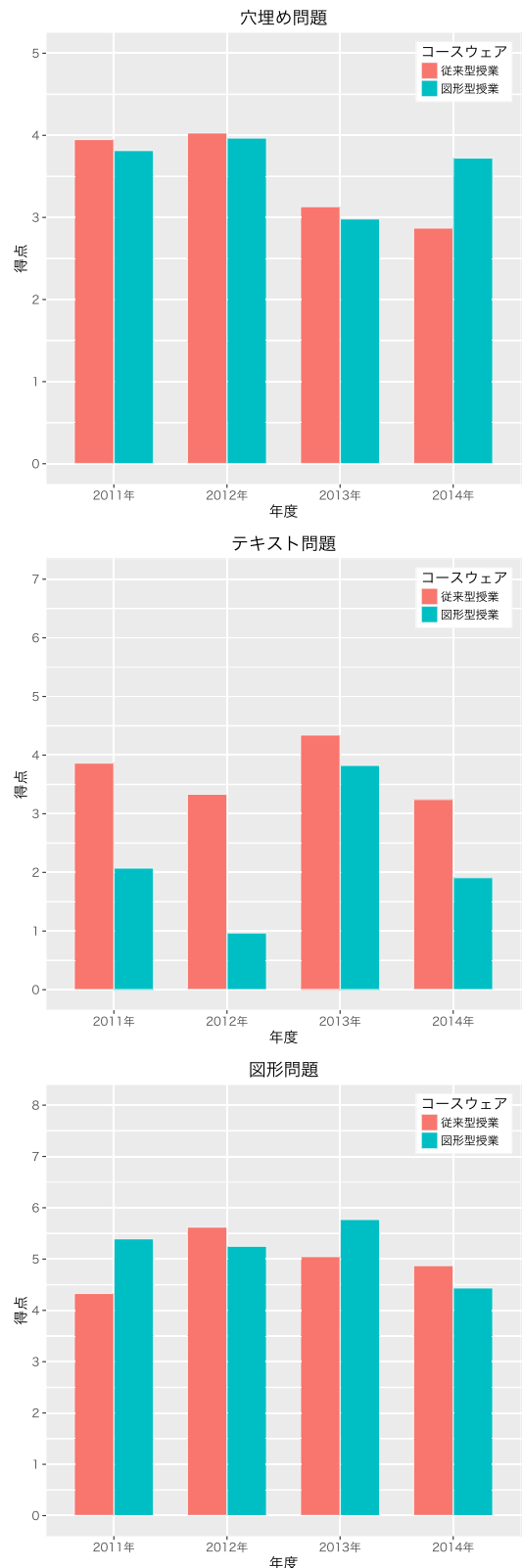


図 6 期末試験の年度・授業方式別の成績比較
Fig. 6 An comparison of the scores of examinations.

であったことから、問題の難易度については、一定ではなかったと考えられる。こうしたことから、テキスト問題については、問題の難易度にかかわらず、従来型授業を受けた学生の方がより高い点数をとる傾向であったといえる。

図形問題では、交互作用が有意になったことから、年度によって授業方式の違いによる平均の差の傾向が異なることが予想される。そこで、授業方式別平均値の違いを年度別に調べた。検定方法は対応のない母平均の差の検定で、有意性判断は有意水準5%のBonferroni法を用いて行った。その結果、いずれの平均値間にも有意な差が見られなかった。以上のことから、図形問題において、図形型授業を受けた学生の方が高い点数をとるといった傾向は見られなかったと考えられる。

5.3 試験結果の考察

本研究での2つのコースウェアでは、プログラミングに関する学習内容が同じになるように配慮している。異なっていたのは、学習を進める際、従来のテキストベース問題を多く扱ったか、図形的な問題を多く扱ったかという点である。このことから、テキスト問題は、従来型授業を受けた学生、穴埋め問題と図形問題は図形型授業を受けた学生にとって解答しやすい問題であったと考えられる。したがって、授業で扱った時間が長いほど学習到達度が高くなるとすれば、解答しやすい問題ほど点数が高くなるはずである。

分析の結果、テキスト問題では授業方式による差が見られたが、穴埋め問題と図形問題では授業方式による差が全体的には見られなかった。図6のテキスト問題結果を見ると、どの年度においても従来型授業の成績が高いことが分かる。このことから、テキスト問題の成績は従来型授業の方が高いと考えられる。先の前提に従うならば、この結果は妥当である。

一方、穴埋め問題と図形問題で授業方式間での差が見られなかったことは、先の前提と矛盾する。これは、従来型授業の第4回目(試験の直近)が図形問題を扱った内容であったためと考えられる。図3からも分かる通り、従来型授業を受けた学生は第4回目の授業後、プログラミングの楽しさが増している。このことから、図形問題に対して主観的には親しみやすいと感じていたと思われる。図2の理解度のアンケート結果も合わせて考えると、これは問題に対する積極性や記憶・知識の定着において良い効果をもたらした。その結果、図形問題において授業方式間に差が見られなかったと思われる。このことが正しければ、テキスト問題においても差が小さくなっていなければならない。しかし、そのような結果は得られなかった。これは、図形型授業では、テキスト問題を扱ったのは第4回目の最後の練習問題だけで時間が短かったことが一因である。また、アンケート(図2, 図3)の結果から分かる通り、学生は従来型授業の場合とは逆の心理的影響を受けていた。したがって、テキスト問題では図形問題のようなことが起こらなかったと考えられる。

以上のことから、本研究の結果は、従来型授業が図形型

授業よりも学習達成度の観点から効果的であるということを示しているのではなく、図形描画を用いることでプログラミング学習が促進される可能性を示唆している。教材が与える心理的影響と学習到達度の関係性をより明確にすることは今後の課題である。

6. まとめ

本研究では、2011~2014年の4年間にわたり、図形描画を主としたコースウェアと、従来型のコースウェアとで学生のモチベーション、理解度、成績に差があるかどうかを調査した。授業内容の変化との対応ではその特徴に差があることが分かり、繰り返し処理の内容を学習する際、従来型授業では、その内容が難しいと評価されているが、図形型授業では同じ内容であっても難しいと評価されなかった。この結果、図形型では繰り返しのようなつまずきやすい学習内容でも理解度や楽しさをほとんど下げることなく学習できていることがうかがえた。一方で、従来型では繰り返しの学習において理解度や楽しさを下げることが分かった。ただしこの授業では、同等の教育内容を提供するために、最終回である4回目に従来型授業で図形描画を、図形型授業で文字列の出力や実数計算を扱っている。その結果、図形型授業では4回目に理解度や楽しさが下がっている。一方で、従来型授業では4回目に理解度や楽しさが上がっている。これらのことより、図形描画を用いた課題は学習者のモチベーションや理解度を上げ、これを主としたコースウェアを提供することは初学者に対する短期間のプログラミング教育に有用であると考えられる。

本論文では、分析の性質上、4回の授業すべてに出席しWebでの調査に回答した受講生が対象であったが、欠席したりアンケートに未回答の学生がいたため、実際の受講生数に比べて分析対象者数が少なかった。今後、調査の方法を改善し、分析対象者数を増やす工夫が必要である。また、どのような条件・状況のときにどちらのコースウェアを用いるのが適切かといった検討を行っていく必要があるだろう。

謝辞 PENの開発を担当していただいた中村亮太氏、授業実践にご協力いただいた大阪大学の外川直子氏、間下以大講師、宮本友介助教、大阪工業大学の安留誠吾准教授に感謝します。本研究はJSPS科研費JP23501163の助成を受けたものです。

参考文献

- [1] 情報処理学会：大学等における一般情報処理教育の在り方に関する調査研究(平成4年度報告書)(オンライン), 入手先(<https://www.ipsj.or.jp/12kyoiku/monbu4-1.html>) (参照 2016-08-03).
- [2] 情報処理学会：大学等における一般情報処理教育の在り方に関する調査研究(平成13年度報告書)(オンライン), 入手先(<http://www.ipsj.or.jp/annai/committee/>)

- education/report3c.pdf) (参照 2016-08-03).
- [3] 情報処理学会 情報処理教育委員会：日本の情報教育・情報処理教育に関する提言 2005 (オンライン), 入手先 <<http://www.ipsj.or.jp/12kyoiku/proposal-20051029.html>> (参照 2016-08-03).
- [4] 初学者向けプログラミング学習環境 PEN Web ページ (オンライン), 入手先 <<http://www.media.osaka-cu.ac.jp/PEN/>> (参照 2016-08-03).
- [5] 西田知博, 原田 章, 中村亮太, 宮本友介, 松浦敏雄：初学者用プログラミング学習環境 PEN の実装と評価, 情報処理学会論文誌, Vol.48, No.8, pp.2736-2747 (2007).
- [6] 独立行政法人大学入試センター：センター試験用手順記述標準言語 (DNCL) の説明 (オンライン), 入手先 <<http://www.dnc.ac.jp/albums/abm.php?f=abm00003020.pdf&n=H23.dncl.pdf>> (参照 2016-08-03).
- [7] 情報活用基礎 プログラミング教育の記録データ - 2011~2014 - (オンライン), 入手先 <<http://www.s.ogu.ac.jp/pen/penData/>> (参照 2016-08-03).
- [8] 安留誠吾, 中西通雄, 景村幸弘：図形描画によるプログラミング入門 - 大阪大学人間科学部・文学部での PEN を用いた実践, 情報教育研究会講演論文集, 2008 年度, pp.531-534 (2008).
- [9] Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M. and Paterson, J.: A survey of literature on the teaching of introductory programming, Working group reports on ITiCSE on Innovation and technology in computer science education, *ACM ITiCSE-WGR '07*, pp.204-223 (2007).
- [10] Kelleher, C. and Pausch, R.: Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers, *ACM Comput. Surv.*, Vol.37, No.2, pp.83-137 (2005).
- [11] 森 秀樹：Scratch を用いた文系大学生向けプログラミング教育, 日本教育工学会論文誌, Vol.34, pp.141-144 (2010).
- [12] 兼宗 進, 御手洗理英, 中谷多哉子, 福井眞吾, 久野 靖：学校教育用オブジェクト指向言語「ドリトル」の設計と実装, 情報処理学会論文誌プログラミング, Vol.42, No.SIG11 (PRO12), pp.78-90 (2001).
- [13] 原田康徳：プログラミング言語ビスケットを用いた基礎としてのプログラミング教育の提案と実践, 情報処理学会デジタルプラクティス, Vol.6, No.2, pp.105-111 (2015).
- [14] Code.org：Hour of Code チュートリアル (オンライン), 入手先 <<https://code.org/learn>> (参照 2016-08-03).
- [15] 岡本雅子, 村上正行, 吉川直人, 喜多 一：「視覚的顕在化」に着目したプログラミング学習教材の開発と評価, 日本教育工学会論文誌, Vol.37, pp.35-45 (2013).
- [16] 西田知博, 原田 章, 中西通雄, 松浦敏雄：プログラミング導入教育におけるコースウェアの違いによる学習効果の比較, 情報処理学会研究報告コンピュータと教育 (CE), Vol.2013-CE-122, No.2 (2013).



西田 知博 (正会員)

1993 年大阪大学大学院基礎工学研究科情報工学分野博士前期課程修了。1996 年同大学情報処理教育センター助手。2000 年から大阪学院大学情報学部講師。2010 年から同大学准教授。プログラミング教育および情報教育に関する研究等に従事。 本会シニア会員。



原田 章 (正会員)

1994 年大阪大学大学院人間科学研究科博士前期課程修了。同年同大学情報処理教育センター助手。同大学人間科学部助手, 甲子園大学人間文化学部講師, 甲子園短期大学助教授・准教授を経て, 2010 年追手門学院大学経営学部准教授。2013 年同大学教授。情報教育におけるコンピュータ不安や習熟度に関する研究等に従事。



中西 通雄 (正会員)

1980 年大阪大学大学院基礎工学研究科情報工学分野博士前期課程修了, 三菱電機入社。1990 年大阪大学基礎工学部助手, 情報処理教育センター助教授を経て, 2002 年より大阪工業大学情報科学部教授。プログラミング教育・技術者倫理教育に関する研究等に従事。



松浦 敏雄 (正会員)

1977 年大阪大学大学院基礎工学研究科情報工学分野博士前期課程修了。1979 年同大学助手。1992 年同大学情報処理教育センター助教授。1995 年大阪市立大学教授。2003 年同大学大学院創造都市研究科教授。分散システムならびに情報教育に関する研究・実践に従事。 本会フェロー。