

論文

# PenFlowchart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

中西 渉<sup>1,a)</sup> 辰己 丈夫<sup>2,b)</sup> 西田 知博<sup>3,c)</sup>

受付日 2015年3月4日，再受付日 2015年5月12日，  
採録日 2015年9月5日

**概要：**プログラミング教育では，テキストでソースコードを記述するテキスト型のプログラミング環境だけでなく，画面上の部品を組み合わせてプログラムを作成する環境も使われている．本研究では，テキスト型の初心者用プログラミング学習環境であるPENで実行できるソースコードを，ドラッグ&ドロップで作ったフローチャートから生成するPenFlowchartを開発し，これを用いて高校生を対象にプログラミング教育を行った．その結果を，PENだけで学習を行っていたときのものと比較して検討を行ったところ，授業の進行（特に序盤の演習）がスムーズになった．また，期末試験におけるプログラミングとフローチャートの変換を行う問題についても点数の向上が見られ，その傾向は成績下位者において顕著であった．

**キーワード：**プログラミング学習，フローチャート

## Evaluation of the Effectiveness of Programming Education with Flowcharts Using PenFlowchart

WATARU NAKANISHI<sup>1,a)</sup> TAKEO TATSUMI<sup>2,b)</sup> TOMOHIRO NISHIDA<sup>3,c)</sup>

Received: March 4, 2015, Revised: May 12, 2015,  
Accepted: September 5, 2015

**Abstract:** Programming environments used in programming education may be text-based or use visual objects. In this study, we developed and evaluated PenFlowchart, which generates codes for PEN (a text-based programming environment for novices) by making flowcharts using objects manipulated by a mouse being used for programming education for high school students. The results of learning using PenFlowchart were compared with results using PEN alone. Also, we found that students' grades are improved (particularly in the case of poorly performing students) on the reference questions in term-end examinations, which ask conversion from text-based programs to flowcharts or vice versa.

**Keywords:** programming learning, flowchart

### 1. はじめに

近年の情報処理技術の発達にともない，初等中等教育に

においても，情報処理や情報そのものの取扱いについて，児童・生徒にどのように学ばせるかについての議論が活発になっている．情報処理学会は2005年の提言 [1] において「小学校・中学校・高等学校それぞれの発達段階に応じて適切な『手順的な自動処理』の体験を持たせる」ことを提案した．日本経済再生本部も日本再興戦略 [2] において，世界最高水準のIT社会を実現するためのプランの中で，ITを活用した21世紀型スキル習得のために「義務教育段階からのプログラミング教育等のIT教育を推進する」としている．

<sup>1</sup> 名古屋高等学校  
Nagoya Senior High School, Nagoya, Aichi 461-8676, Japan  
<sup>2</sup> 放送大学  
The Open University of Japan, Chiba 261-8586, Japan  
<sup>3</sup> 大阪学院大学  
Osaka Gakuin University, Suita, Osaka 564-8511, Japan  
a) watayan@meigaku.ac.jp  
b) ttmtko@gmail.com  
c) nishida@ogu.ac.jp

情報処理学会が2006年に行った教育用プログラミング言語に関するワークショップ [3] では十数個のプログラミング言語が紹介された。その中にはテキストによるプログラミング以外に、Scratch や Squeak に代表されるようなパーツの組合せでプログラミングができるビジュアル型プログラミング環境も含まれている。

そこで、ビジュアル型プログラミング環境での学習が、テキスト型のプログラミングに対してどう作用するかを考察するのが本研究の目的である。

筆者らは、最近の教育用プログラミングに関する研究成果をもとにして、テキスト型のプログラミング学習環境である PEN [4] で実行できるソースコードをフローチャートから生成するプログラミング環境 PenFlowchart を作成して、授業を行った。またその結果を、我々の実践以前に行われた、テキストのみによるプログラミングだけで授業を行ったときと比較し、その成果について検討した。

以下、2章ではこの分野の先行研究について述べ、3章では PenFlowchart の概要について述べる。また4章では、PenFlowchart を用いた授業実践の内容と結果について述べ、5章で考察、6章で本論文のまとめを述べる。

## 2. 先行研究

ここでは、本研究を進めるにあたって参考にした先行研究について述べる。

### 2.1 PEN と xDNCL

PEN は大阪学院大学情報学部西田研究室と大阪市立大学大学院創造都市研究科松浦研究室の共同プロジェクトとして開発されている、初学者向けのプログラミング学習環境である [5]。PEN で用いられているプログラミング言語 xDNCL は大学入試センター試験「情報関係基礎」の出題で用いられている言語 DNCL を拡張したものである。日本語がベースになっていることから読んで理解することは容易である反面、入力は面倒である（日本語変換や、表記の「ゆれ」、句読点の有無などに気をつけなくてはならない）。そこで、構文の入力が素早く正確にできるように入力支援ボタンやインデントの自動挿入が実装されている（図 1）。また、実行中の変数の値表示や、実行速度の調整、1行ずつの実行など、動作確認やデバッグに適した機能も充実している。

中村らによる報告 [6] では、入力支援ボタンを用いることが（キー入力に習熟した者にとっては）プログラムの入力時間の短縮にはつながらないが、キー入力回数が大幅に減ったことを確認しており、このことがタイプミスによる文法エラーの発生を抑えられる可能性について言及している。また、入力支援ボタンのメリットは入力の手間を省くことよりもむしろ、ブロック構造を意識させることにあると筆者らは考えている。



図 1 PEN の実行画面

Fig. 1 The PEN programming environment.

西田らの研究 [4] では、プログラミング学習において PEN を用いた場合、JavaScript よりも理解度および自己評価が高いという結論を得ている。また、構造化チャートを用いたプログラミング環境がアルゴリズム教育には適していることを認める一方、構造化チャート自体の学習が必要であることや、コードを書くことが「プログラミング教育」には重要であることから、PEN では図による表現でなくコードによる記述を選んだことにも言及している（本論文では「アルゴリズム」という語を「プログラムを記述するときの考え方」の意味で用いる）。

石らは PEN のエディタに手を加えることで、ソースコードのブロック構造の破壊を不可能にする実装について研究した [7]。これはエディタ画面上でブロック構造を保ったパーツを組み合わせてソースコードを構成するというもので、構文エラーを防ぐということについては十分な機能がある。開発動機においては、後述する PenFlowchart とほぼ同じだと考えられるが、実装においてはソースコードの編集を中心に据えているという点で異なっている。

### 2.2 ビジュアル型言語によるプログラミング

フローチャートによるプログラミング環境には Flowgorithm [8] や、阿部らの「アルゴリズム自習システム」[9]、山本の「動くフローチャート」[10] などがある。これらはいずれも、画面上でフローチャートを作成することで、プログラムの実行やトレースを行うことができる。また、プログラミング言語のソースコードの出力も可能である。後述する PenFlowchart ではフローチャートはプログラムの作成環境の1つではあるが、実行は PEN の画面で行うため、フローチャート上で動作の確認をすることはできない。

斐品ら [11] は学習者が特定のプログラミング言語の構文に気を取られすぎないように、構造化チャート PAD によるアルゴリズム作成・実行が可能な学習支援システム JPADet を構築した。演習をプログラミング言語 C で行うクラスと

表 1 フローチャートとプログラムの個数 (旧課程)

Table 1 Numbers of flowcharts and programs in textbooks (previous curriculum).

出版社	発行	フローチャート	プログラム
実教出版	H15	7	4 Visual Basic
実教出版	H17	7	4 Visual Basic
日本文教出版	H15	3	5 DNCL 風
日本文教出版	H17	5	5 DNCL 風
第一学習社	H15	2	1 BASIC
第一学習社	H17	2	1 BASIC
啓林館	H15	5	3 Pascal 風
清水書院	H15	3	7 DNCL 風
オーム	H15	0	12 DNCL 風

JPADet で行うクラスを比較し、後者の方が授業を早く進行できたという結果を得ている。

ビジュアル型言語とテキスト記述型言語の併用については松澤らの研究 [12] がある。BlockEditor はビジュアル型言語としてプログラムを作成することによって、テキスト記述型言語である Java のソースコードを生成するプログラミング環境である (Java のソースコードからビジュアル型に変換することもできる)。同研究ではプログラミングの授業を Java を基本にしつつ BlockEditor も併用できる環境で行い、BlockEditor の利用率の推移について調査している。80%の学生が徐々に BlockEditor の利用率を減少させて Java の環境に移行していることから、GUI やビジュアル型言語がテキスト記述型言語の学習につながっていることを示している。

### 2.3 プログラミングか、アルゴリズムか

辰己が 2006 年に発表した調査研究 [13] において、高等学校「情報 B」の文部科学省検定教科書の「コンピュータにおける情報の処理」「情報の表し方と処理手順の工夫」に現れるフローチャートとプログラムの個数は表 1 ののとおり状況であることが示されている。

表中のフローチャートの数 (単なる部品説明は除いている) は、アルゴリズムの説明の度合いととらえることができる。DNCL 風の言語を使用している教科書ではプログラムの数がフローチャートの数以上であり、前述した PEN によって DNCL で書かれたプログラムは実行可能であるが、教科書の記述ではこれを直接実行することは想定していない。こう考えると表 1 からはすべての教科書において実行可能なプログラムの記述がフローチャートの記述よりも少ないことが読み取れる。この点に関して辰己は、「アルゴリズムのみ」の学習では、「実際に動作する」という実感を得にくくなってしまふことから、(特に初等中等教育においては) アルゴリズム学習よりもプログラミングを優先すべきであると主張している。

ただし新課程の「情報の科学」の教科書では表 2 のよう

表 2 フローチャートとプログラムの個数 (現行課程)

Table 2 Numbers of flowcharts and programs in textbooks (current curriculum).

出版社	発行	フローチャート	プログラム
実教出版	H25	13	17 Visual Basic
実教出版 (最新)	H25	7	8 Visual Basic
日本文教出版	H27	13	12 JavaScript
東京書籍	H25	2	8 JavaScript
			4 ドリトル
			2 DNCL 風
数研出版	H25	3	1 Visual Basic

に、プログラムの個数がフローチャートの個数を上回る教科書が多く、上記した状況は改善されていると考えられる。

## 3. PEN と PenFlowchart

ここでは、PEN を授業で用いたときに生徒が陥りやすい間違いと、それを解決するために開発した PenFlowchart の概要について述べる。

### 3.1 PEN で起こりやすいエラー

前章ですでに述べたとおり、PEN では入力支援ボタンを有効に用いることで、プログラム入力の手数を大幅に削減できる。しかし、プログラム入力画面が普通のテキストエディタと同じ構成・使用方法となっていることから、実際に教室で使用してみると、生徒の中には、構造についての注意をすることなくエディタを操作し、結果として、そのブロック構造を破壊してエラーを起こしてしまう者がいた。たとえば、

もし  $a=0$  ならば  
| 「ゼロ」を表示する  
を実行する

というようなプログラムを、

もし  $a=0$  ならば  
「ゼロ」を表示する

と書きなおしてしまう。プログラミングに必要な構造を理解できている者であれば、最初の例に IF~ENDIF のようなブロック構造を見出すことができるため、構造を壊すことはない。しかし、初学者である生徒の中には、日本語としての自然さを意識し、「もし~ならば『~を表示する』を実行する」よりも「もし~ならば、~を表示する」の方が自然であると考えて「を実行する」を削除してしまい、その結果としてエラーを生じさせる。またこれは、生じたエラーへの対処に気をとられることによって、構造の理解が阻害されるという悪循環につながる。

このような場合、PEN は実行時に構文エラーがあることをメッセージとして表示するが、それはけっして初学者に

とって分かりやすいものではない。上の例では「エラー：○行の○文字目あたりを見てください」と表示されるが、それを見て「を実行する」が抜けていることが原因であると気づく者は、そもそもこの行を削除したりしないと考えられるからである。

### 3.2 PenFlowchart の開発

プログラミングの授業を円滑に行うためには、上述したエラーを生じさせないようにすることが重要である。そこで、中西は、このようなエラーを生じさせないようにしながら、プログラミングの授業を円滑に実行できる環境を開発することとし、画面上でフローチャートを作ると、自動的に PEN 用のプログラムを生成する PenFlowchart を 2011 年に開発した [14]。開発にあたっては、PAD などの構造化チャートを用いることも検討したが、2.3 節で述べたように高校の情報の教科書ではそれらは使われていない。教科書にない形式のチャートを用いると、そのために新たな学習が必要になる。高校の教科書で用いられているプログラムのための図式はフローチャートに限られていることから、フローチャートによる実装を行うことにした。

PenFlowchart の実行画面を図 2 に示す。上部のパーツを下のフローチャートにドラッグ&ドロップし、パラメータなどを記述することでプログラムを作成する。パーツのドラッグ&ドロップは、PEN で入力支援ボタンを押すことに対応している。生成されたプログラムは同時に起動される PEN の画面に自動的に入力され、そのまま実行やデバッグを行うことができる。フローチャートが忌避される理由の 1 つとして、線が自由に引けてしまうので流れが分かりにくくなることがあるが、PenFlowchart では構造を保ったまま部品を配置することしかできないので、そのような問題は発生しない。作成されたフローチャートは PNG または EPS フォーマットの画像として出力することができる。

2012 年には、PEN のプログラムからフローチャートを生成できる機能を付加した。ただし、PEN では実装されている「～の値に応じて」「そうでなくもし」の構文 (switch や else if に相当する) や、サブプロシージャには対応していない。また、フローチャートを修正したときはただちにプログラムに反映されるが、プログラムを修正したときは変換ボタンを押すことでフローチャートに反映するようにしている。これは、修正中のプログラムは構文エラーの状態にあることが多いからである。

アルゴリズムの表現であるフローチャートによってプログラミングができることから、フローチャートの学習が「動作の実感をともなう」ものとなる。これは、2.3 節で辰己が主張した「アルゴリズムのみでは動作の実感を得にくい」ことを改善したものであるともいえる。

PenFlowchart は、Java を利用して開発した。プログラムおよびソースコードはライセンスは GPL とし、<http://>

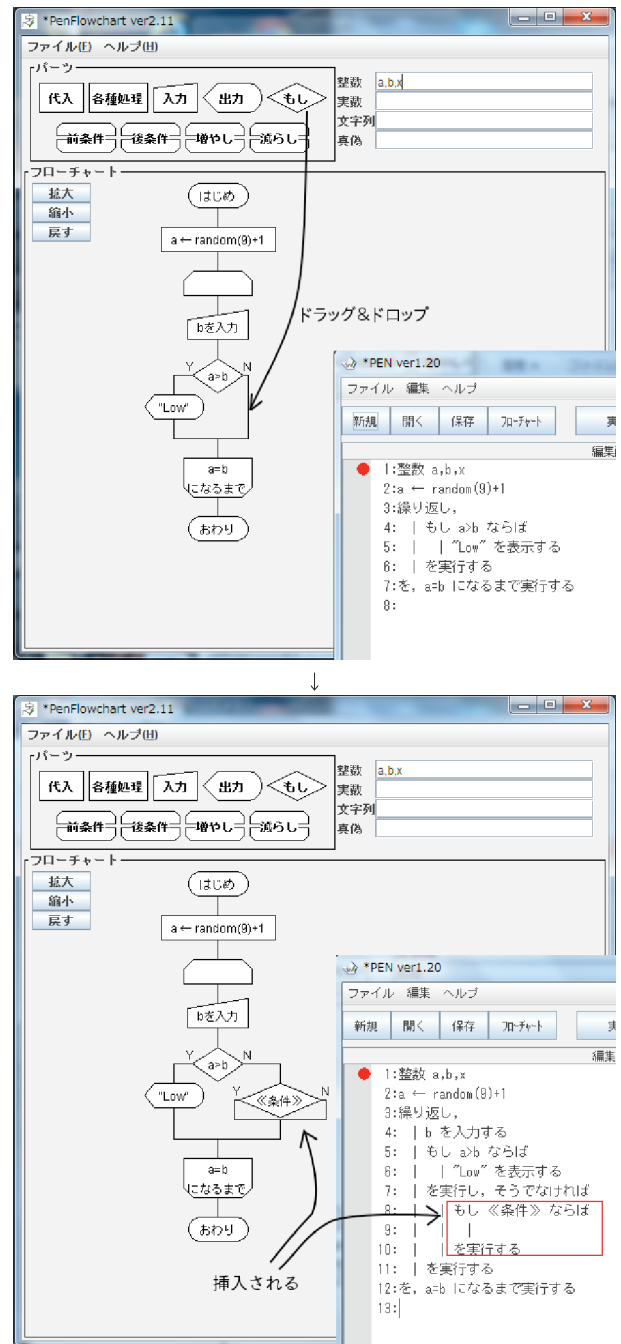


図 2 PenFlowchart の実行画面

Fig. 2 The PenFlowchart programming environment.

[watanayan.net/prog/](http://watanayan.net/prog/) で配布をしている。

## 4. 授業実践の内容と結果

ここでは 2 つの高等学校での実践をもとに、PenFlowchart の効果を評価する。

### 4.1 大阪学院大学高等学校における実践

著者の 1 人は、2007 年度から高大連携授業として、大阪学院大学高等学校情報コースの 3 年生を対象に 1 年間のプログラミングの授業を担当している。プログラミング環境は PEN を使い、グラフィカルな例題、課題を中心と

表 3 5 回目までの授業内容とプログラム数

Table 3 Course contents and numbers of programs (Lessons 1-5).

(1) プログラミング環境の導入, 例題プログラム (BMI 算出) を用いた環境操作法の習得, 簡単なプログラムの作成		
例題	操作法の習得	1
	文字列の表示	1
	変数と演算	1
演習	文字列の表示	1
	変数と演算	1
(2) 逐次処理		
例題	円描画	1: 描画
	変数値の交換	1
演習	円描画	2: 描画
	変数値の交換	1
(3) 条件分岐 1		
例題	1 分岐	1
	2 分岐	1: 描画
	大小比較	1
	論理演算	1
演習	1 分岐	2
	2 分岐	1: 描画
	大小比較	1: 非描画, 1: 描画
	論理演算	2: 非描画, 1: 描画
(4) 条件分岐 2, 繰返し		
例題	多分岐	1
	for 型繰返し	1: 非描画, 1: 描画
	while 型繰返し	1
演習	多分岐	1: 非描画, 1: 描画
	for 型繰返し	2: 非描画, 2: 描画
	while 型繰返し	1
(5) 繰返し 2		
例題	2 重繰返し	1: 描画
	繰返しを使った軌跡描画	1: 描画
	アニメーション	2: 描画
演習	2 重繰返し	2: 描画
	軌跡	3: 描画
	アニメーション	3: 描画

して演習を行っている。2012 年度は序盤, PenFlowchart のみを利用させ演習を進めた。テキストは 2012 年度には例題のプログラムにフローチャートを併記したことで, PenFlowchart でグラフィックス関連命令には日本語の関数名を利用しているため, 英語の関数名を日本語の関数名に置き換えたこと以外は同じものを利用している。5 回目までの例題と演習内容は表 3 のとおりで, ここまでで制御構造の基本を学ぶようになっていく。なお, 表 3 では該当の回で提示した例題と演習の内容と数を示し, グラフィカルな例題・演習を使っている場合は「描画」と示し, その数の内訳も示している。生徒には提示した例題を入力し, その後の演習に取り組むように指示した。入力・作成したプログラムのファイルは毎回, 授業終了時に提出させ, その回のうちにできなかったものに関しては, 次回以降で取

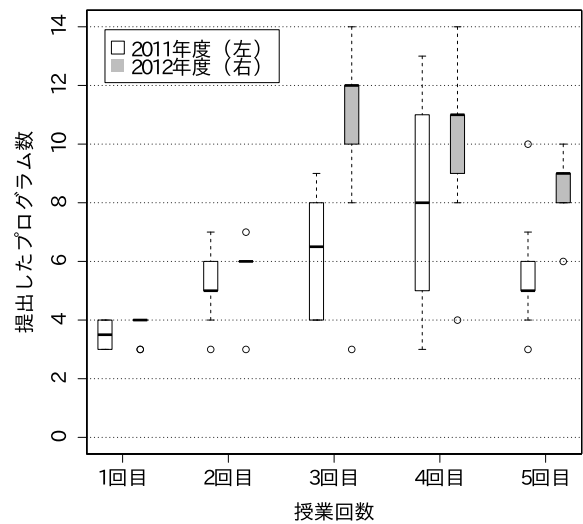


図 3 取り組んだプログラム数

Fig. 3 Trend of number of programs submitted by students.

り組むように指示した。

授業を受けた生徒の数は両年度とも 15 名であった。図 3 は, 5 回目までに欠席や欠席に準じる (課題にまったく取り組まなかった) 回のなかった, 2011 年度の 10 名, 2012 年度の 9 名を対象に各回に作成した例題と演習問題のプログラム数を比較したものである。その結果, 2012 年度の方が作成したプログラムの数は多くなっており, 特に 3 回目以降では大きな差がでていく。人数が少なく, 年度による生徒の差はあるが, 条件分岐というプログラムの構造が現れる回から大きな差が出たことは, PenFlowchart により初学者に対するプログラミングの序盤の演習がスムーズに進むことを示唆している。

#### 4.2 名古屋高等学校における実践

著者の 1 人は名古屋高等学校で情報 B を担当し, 1 年生の 2 学期の授業ではプログラミングとアルゴリズムを扱っている。実習で用いるプログラミング環境として, 2006 年度から PEN を用いている。PenFlowchart は 2011 年度の途中から導入し, 生徒には PEN と PenFlowchart のどちらでも使いたい方を使うよう指示している。この授業では自作プリントを用い, 例題はすべてフローチャートとプログラムを併記している。

PenFlowchart 導入以前 (2009, 2010 年度) と導入以後 (2011, 2012 年度) の定期テスト結果について比較を行った。また, 2011 年度以降の生徒に PEN と PenFlowchart の使い分けについてアンケートを実施した。以下, その結果について述べる。

アンケートの結果は表 4, 表 5 のとおりである。2012 年度の回答は表 4, 表 5 とともに PenFlowchart に偏っているが, 2011 年度は均衡している。これは, 2011 年度の授業開始時には PenFlowchart が作られておらず, 授業用の自作プリントが PEN の使用だけを想定して書かれていた

表 4 どちらをメインで使ったか (人数と比率)

Table 4 Which did students use PEN or PenFlowchart? (number and ratio).

年度	人数		比率 (%)	
	2011	2012	2011	2012
PEN 中心	75	22	28.4	7.8
PenFlowchart 中心	64	172	24.2	60.8
課題によって使い分け	50	33	18.9	11.7
課題の途中で使い分け	3	9	1.1	3.2
忘れた	56	32	21.2	11.3
(無回答)	16	15	6.1	5.3

表 5 一方だけを使うとしたらどちらを選ぶか (人数と比率)

Table 5 Which did students prefer PEN or PenFlowchart? (number and ratio).

年度	人数		比率 (%)	
	2011	2012	2011	2012
PEN	131	32	49.6	11.3
PenFlowchart	115	235	43.6	83.0
(無回答)	18	16	6.8	5.7

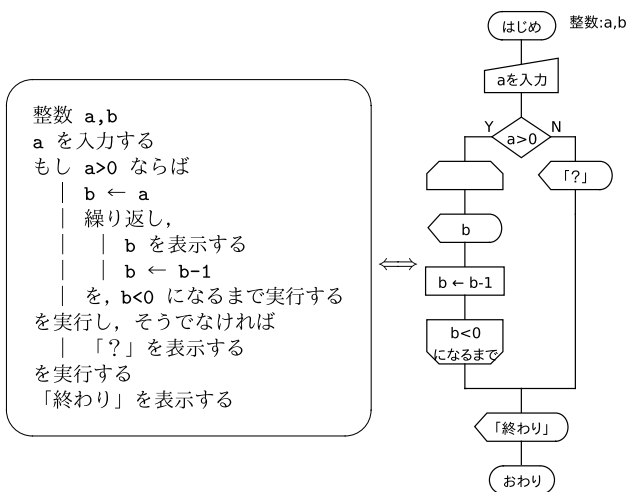


図 4 期末テストの小問の例

Fig. 4 An example of reference question of term-end examination.

こと, また, 同年度は PenFlowchart 開発初期であり, ほぼ毎週のように頻繁にバージョンアップしたことにより PenFlowchart の使用を敬遠する生徒がいたことなどによると推測できる. 2012 年度は最初から両方を使うことを想定して自作プリントを改訂したため, PenFlowchart 中心の生徒が多くなったと考えられる.

年度をまたがった成績を比較するため, 期末テストで毎年出題している小問の成績の比較を行う. これはフローチャートからプログラムへ, またその逆の書き換えの問題であり, 難易度は図 4 の例のようなループと IF-ELSE のネストで統一している. また, 問題には xDNCL の文法を資料として与えて, 構文に使われる最低限の語彙は理解できるように配慮した.

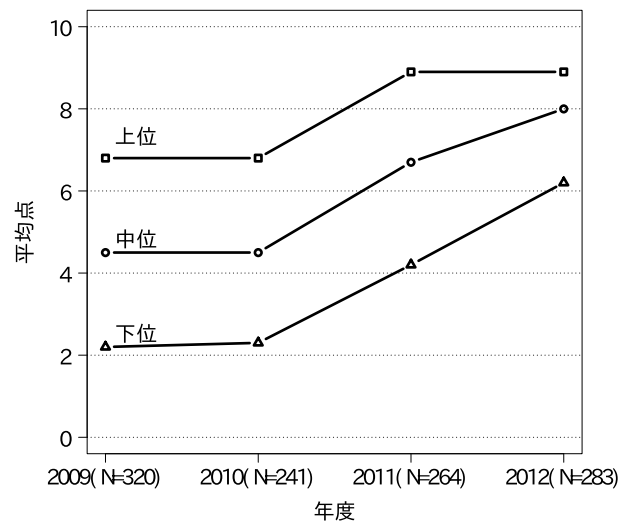


図 5 小問の平均点推移

Fig. 5 Trend of average scores of the reference question.

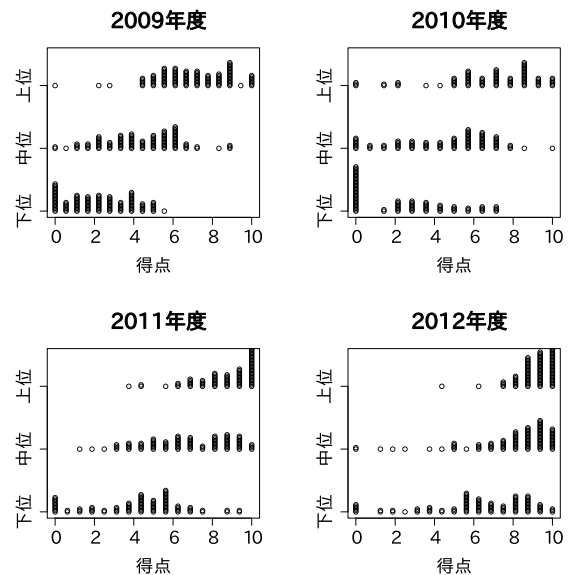


図 6 小問の得点分布推移

Fig. 6 Score distributions of the reference question.

筆者らは, PenFlowchart によりテキストでなく図でプログラミングを考えることは, 低成績者にとって有効であると予想した. そこで得点の比較を行う際に, 同期末テストの点数によっておよそ 1:1:1 の割合で上位・中位・下位のランクに分けて集計を行った. 上記小問の平均点 (10 点満点に換算してある) の推移は図 5 のとおりである. PenFlowchart を導入した 2011 年度以降, 全体的に点数は向上しており, 特に下位生徒には効果的であったと考えられる. 図 6 は上記した小問の得点分布であり, これによれば 0 点 (あるいはそれに近い点数) の生徒が確実に減っていると見られる.

## 5. 考察

4 章に示した実践の結果に基づく考察を述べる.

表 6 表 5 の回答別の期末テストの平均点

Table 6 Average scores of term-end examination (PEN vs. PenFlowchart).

年度	2011	2012
PEN	57.5	64.0
PenFlowchart	56.8	55.2

5.1 課題提出

4.1 節で生徒が作成したプログラム数は、1, 2 回目の授業では大差ないが、条件分岐や繰り返しが含まれる第 3 回目以降での差が大きい。3.1 節で述べたエラーが起こるのは条件分岐や繰り返しが使われるようになってからである。そこで大きい差が得られたことは、フローチャートでプログラミングすることによってそのようなエラーを避けることができたためと考えられる。

名古屋高等学校の授業では、以前は構文エラーのまま課題を提出する生徒が目だったが、PenFlowchart を使い出してから明確に減少した。

5.2 PEN と PenFlowchart の選択

4.2 節のアンケートを行うにあたって筆者らは、PEN と PenFlowchart の両方が使えるのであれば多くの生徒が PenFlowchart を使うだろうということ、また、それでもあえて PEN を選ぶ者は高成績者が多いだろうということを予想していた。

前述したとおり、PenFlowchart を使わせる環境が十分に整わないまま演習が始まったため、2011 年度の生徒では PEN を選ぶ者の方が若干多かったが、2012 年度は多くの者が PenFlowchart を選んだことから、予想が正しければ、PEN を選んだ者には好成績者が多いと考えられる。

表 5 の回答によってどちらを選択するかをグループ分けして期末テストの成績を集計した結果は表 6 のとおりであった。2011 年度にほとんど差がなかったのは 4.2 節で述べたように PenFlowchart が敬遠される傾向があったためと考えられる。2012 年度の平均点は PEN を選んだ者の方が有意に高かった (Welch の検定,  $p = 0.043$ )。

5.3 PenFlowchart 導入の効果の測定方法

前節に示したように PenFlowchart を導入することにより、フローチャートやプログラムの構造を把握する問題については得点の向上が見られた。しかし、これはすでにできあがっている構造を読み取り、変換する力だけに対する評価であって、プログラムを望むとおりにつくり上げる力や、すでに作られたプログラムがどのように動作するかを読み取る力を評価するものではない。名古屋高等学校の定期テストではフローチャートをトレースして実行結果を問うような問題も出題しているが、4.2 節の分析対象の小問との相関を分析したところ、どの年でも相関係数 0.3~0.4

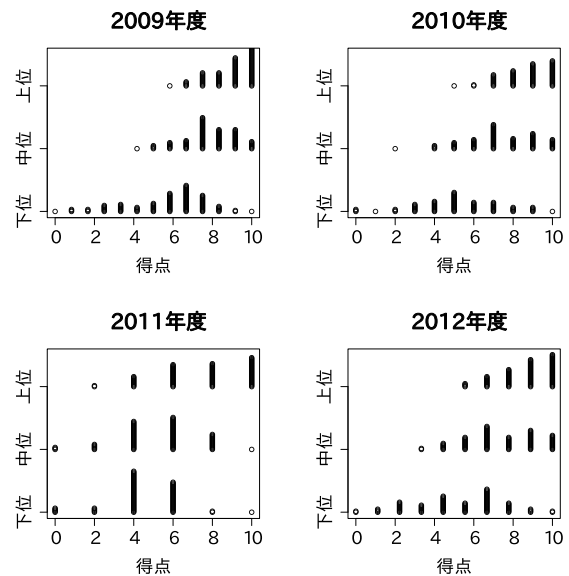


図 7 変数トレース問題の得点分布推移

Fig. 7 Score distributions of the question about trace of variables.

程度の弱い相関しかなかった。得点分布推移は図 7 のとおりであり、PenFlowchart を使う以前と以後で目立った変化は見られなかった。

6. まとめ

本研究では PenFlowchart をプログラミング学習の導入時に用いることによって初学者の学習がスムーズに行われること、フローチャートやプログラムの構造を把握する問題については PenFlowchart 導入以前よりも得点の向上が見られることを検証した。

今後、PEN のみを使っていた過去の定期テストを分析し、プログラムを作る力などの分析を行っていききたい。また、DNCL 以外のプログラミング言語とフローチャートの組合せを考えたプログラミング環境についても考察を進めたい。PenFlowchart の BASIC 版、JavaScript 版はすでに開発しており [15]、これによる実践を今後行っていききたいと考えている。

謝辞 本研究は JSPS 科研費 26350327 の助成を受けたものである。

参考文献

- [1] 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, 情報処理学会 (オンライン), 入手先 (<http://www.ipsj.or.jp/12kyoiku/proposal-20051029.html>) (参照 2014-11-06).
- [2] 日本経済再生本部：日本再興戦略—JAPAN is BACK, 首相官邸 (オンライン), 入手先 (<http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/saikou-jpn.pdf>) (参照 2014-11-06).
- [3] 情報処理学会：教育用プログラミング言語に関するワークショップ 2006 報告集 (2006).
- [4] 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 (2007).

- [5] 中村亮太, 西田知博, 松浦敏雄: 初学者向けプログラミング学習環境 PEN, 大阪学院大学, 大阪市立大学 (オンライン), 入手先 (<http://www.media.osaka-cu.ac.jp/PEN/>) (参照 2014-11-06).
- [6] 中村亮太, 西田知博, 松浦敏雄: プログラミング入門教育用学習環境 PEN, 情報処理学会研究報告コンピュータと教育 (CE), Vol.2005-CE-081, No.104 (2005).
- [7] 石 天翔, 中村亮太, 松浦敏雄: 初学者向けプログラミング学習環境 PEN における入力支援機能の改良, *Journals of Informatics*, Vol.8, No.1 (2011).
- [8] Cook, D.: Flowgorithm (online), available from (<http://www.flowgorithm.org>) (accessed 2015-03-30).
- [9] 阿部清彦, 大山 実, 大井尚一: コンピュータプログラミング学習のためのアルゴリズム自習システム, 工学・工業教育研究講演会講演論文集, Vol.18, pp.140-141 (2006).
- [10] 山本 恒: フローチャートによるアルゴリズムの視覚化と検証システムの開発, 日本情報科教育学会第 4 回全国大会論文集, pp.102-103 (2011).
- [11] 斐品正照, 徳岡健一, 河村一樹: 構造化チャートを用いたアルゴリズム学習支援システム (情報教育), 情報処理学会論文誌, Vol.45, No.10, pp.2454-2467 (2004).
- [12] 松澤芳昭, 酒井三四郎: ビジュアル型言語とテキスト記述型言語の併用によるプログラミング入門教育の試みと成果, 情報処理学会研究報告コンピュータと教育 (CE), Vol.2013-CE-119, No.2 (2013).
- [13] 辰己丈夫: JavaScript 高等学校の教室で準備なしに利用可能なプログラミング言語, 教育用プログラミング言語に関するワークショップ 2006 報告集, pp.6-11 (2006).
- [14] 中西 渉: PenFlowchart の開発, 情報処理学会研究報告コンピュータと教育 (CE), Vol.2012-CE-113, No.13 (2012).
- [15] 中西 渉: PenFlowchart の「他」言語化, 日本情報科教育学会第 1 回研究会報告書, pp.17-20 (2013).



中西 渉 (正会員)

1989 年名古屋大学理学部数学科卒業。同年より名古屋高等学校数学科 (のちに情報科) 教諭。日本情報科教育学会会員。



辰己 丈夫 (正会員)

1993 年早稲田大学大学院理工学研究科修士課程数学専攻修了。2014 年筑波大学大学院ビジネス科学研究科博士後期課程修了。博士 (システムズ・マネジメント)。1993 年早稲田大学情報科学研究教育センター助手, 1999 年神戸大学発達科学部講師, 2003 年東京農工大学総合情報メディアセンター助教授を経て, 2014 年より放送大学教養学部准教授。情報科教育法, 情報倫理, 情報倫理教育, 数学基礎論, 発達障害者教育の ICT 活用に興味を持つ。日本ソフトウェア科学会, 電子情報通信学会, CIEC, 日本情報科教育学会各会員。情報処理学会シニア会員。



西田 知博 (正会員)

1991 年大阪大学基礎工学部情報工学科卒業。同大学大学院基礎工学研究科を経て, 1996 年同大学情報処理教育センター助手。2000 年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に従事。ACM, 電子情報通信学会, 情報科教育学会各会員。情報処理学会シニア会員。