

PenFlowchart によるプログラミング導入教育の評価

中西渉^{1,a)} 辰己丈夫^{2,b)} 西田知博^{3,c)}

概要: プログラミング教育においては、テキストでソースコードを記述するものばかりでなく、パーツを組み合わせてプログラムを作成する環境が多く使われている。しかし後者については、初学者の導入段階では適していると考えられるものの、その後の学習につながるかどうかが問題視されている。

本研究では、テキスト型の初心者用プログラミング学習である PEN 用のソースコードをフローチャートで生成する PenFlowchart を開発し、これを用いてプログラミング学習を行なった。その結果をそれ以前のものと比較して検討を行なった結果、授業の進行（特に序盤の演習）がスムーズになったほか、テキスト型のプログラミングの課題についても点数の向上が見られた。

キーワード: プログラミング学習, フローチャート

Evaluation to introductory education of programming using PenFlowchart

Abstract: In programming education, our programming environment are not only text type but also visual type. We assemble programs by combining the parts in visual type environment. And visual type suits beginners, but it is questionable whether it leads to subsequent study.

In this research, we developed 'PenFlowchart', which generates codes for PEN by making flowcharts, and performed programming education. As a result of comparing their results with the past thing, we recognized that advance of the lessons became smooth and the mark of the test of text type programming improved.

Keywords: Learning of programming, Flowchart

1. はじめに

近年の情報処理技術の発達に伴い、初等中等教育においても、情報処理や情報そのものの取り扱いについて、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。また、慶応義塾大学では総合政策学部と環境情報学部が 2016 年度からの入試において共通教科「情報」を出題すると公表した。これは高等学校の新学習指導要領では数学からプログラミングがなくなってしまうことに対応する処置であると説明されている。

一方で、初等中等教育におけるコンピュータプログラミングの学習についての研究は、古くはパパートによる Logo の開発に始まる。我が国では 1980 年代の慶応義塾大学の中西らによる Logo を用いた実践 [3] や、BASIC などの言語を利用した実践があった。

だが、GUI の普及に伴ない、教育用プログラミングの言語や環境には変化が生じるようになった。最近の教育現場では、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プログラミング環境が用いられて

¹ 名古屋高等学校
Nagoya High School

² 早稲田大学
Waseda University

³ 大阪学院大学
Osaka Gakuin University

a) watayan@meigaku.ac.jp

b) ttmtko@gmail.com

c) nishida@ogu.ac.jp

いる。

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元にして、テキスト型のプログラミング学習環境であるPENのソースコードをフローチャートから生成するPenFlowchartと呼ばれるプログラミング環境を作成して、授業を行なった。またその結果を、我々の実践以前に行なわれた、テキストのみによるプログラミングだけで授業を行なったときと比較し、その成果について検討した。

以下、第2章ではこの分野の先行研究について述べ、第3章ではPENとPenFlowchartの概要について述べる。また第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、第5章において考察と本論文のまとめを述べる。

2. 先行研究

2.1 プログラミングか、アルゴリズムか

筆者らが2006年に発表した調査研究[4]では、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」の内容は、表1の通りの状況であることが明らかとなった。

表1 フローチャートとプログラムの個数(1)

出版社	発行	FC	プログラム
実教出版	H15	7	4 VB
実教出版	H17	7	4 VB
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

なお、表1のFCの欄は「掲載されているフローチャートの個数(単なる部品説明は除く)」であり、アルゴリズムの説明の度合と捉えることができる。また、プログラムの欄は掲載されているプログラムの個数と記述言語である。DNCLとは、大学入試センターが情報関係基礎の出題に用いている日本語を用いた処理言語であり、PENを利用すれば、実行可能である。

この表を見ると、フローチャートの個数がプログラムの数より多いのはVB、BASIC、Pascal風を採用した各社であり、逆にプログラムの数の方が多いのがDNCL風日本語擬似プログラミング言語を採用した各社となっていることがわかる。これは、日本語を利用しないプログラムを掲載している会社は、プログラムより「フローチャートの方がわかりやすい」と考えているからであると推測する。一方で、DNCL風言語を利用している会社は、プログラムリストをプログラムそのものというよりも、「アルゴリズムを日本語で記述したもの」と扱っていることが、教科書よ

表2 フローチャートとプログラムの個数(2)

出版社	発行	FC	プログラム
実教出版	H15	7	4 VB
実教出版	H17	7	4 VB
日本文教出版	H15	8	0
日本文教出版	H17	10	0
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	10	0
オーム	H15	12	0

り推測できる。そこで、DNCL風言語で書かれたプログラムの個数をフローチャートの個数として数えれば、表1は表2のように変わり、多くの教科書では、プログラムそのものよりは、アルゴリズムを説明しているということがわかる。

2.2 ビジュアル型言語によるプログラミング

ビジュアル型言語によるプログラミング教育は、単に図を描いたりすることに終わってしまっていて、他の環境に繋がらないのではないかと心配はある。しかし飯作ら[5]ではViscuitを用いた授業を行なうことで、フローチャートによる構造化の3要素の理解度が向上したことが述べられている。Viscuitのプログラミングは図だけで遷移を記述するものであり、これを用いてアルゴリズムの学習をしていることは興味深いと考える。しかしこの研究は、Viscuitを用いることが他の方法よりも有効であるかどうかを示すものではない。

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[6]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ない、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。ただしここで調査されているのはツールの利用率のみであり、BlockEditorを用いたことが理解度の向上に結びついたかについては直接的なデータを得ていない。

2.3 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 [7]。PEN で用いられているプログラミング言語 xDNCL は前述した 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。

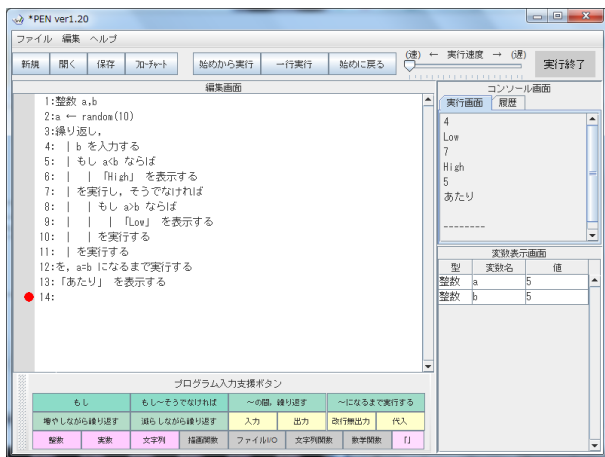


図 1 PEN の実行画面

中村らによる報告 [8] 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繋がらないが、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。しかし、プログラムをワープロ入力のように打ち込むことが高速にできたからといって、アルゴリズムが理解されていたかという点には疑問が残る。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。

西田らの研究 [9] 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。

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の研究もなされている [10]。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 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。

3. PEN と PenFlowchart

前節で既に述べた通り、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、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感じ、「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てしまう。またこれは、生じたエラーへの対処に気を取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。

さらに、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、それは決して初学者にとってわ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」が抜けていることに気づく人は、そもそも初めから削除したりしないため、このエラーメッセージは初学者にとってわかりやすいものではない。

そこで、このようなエラーを生じさせないようにすることが重要であると考え、画面上でフローチャートを作ると、自動的に PEN のプログラムが生成される環境である PenFlowchart を 2011 年に開発した [11]。

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なうことができる。2012 年にはさらに改良を

加え、PEN のプログラムからフローチャートを生成できるようにもした。ただし、構文エラーのあるものは変換できない。また、一部の構文・機能には対応していない。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。

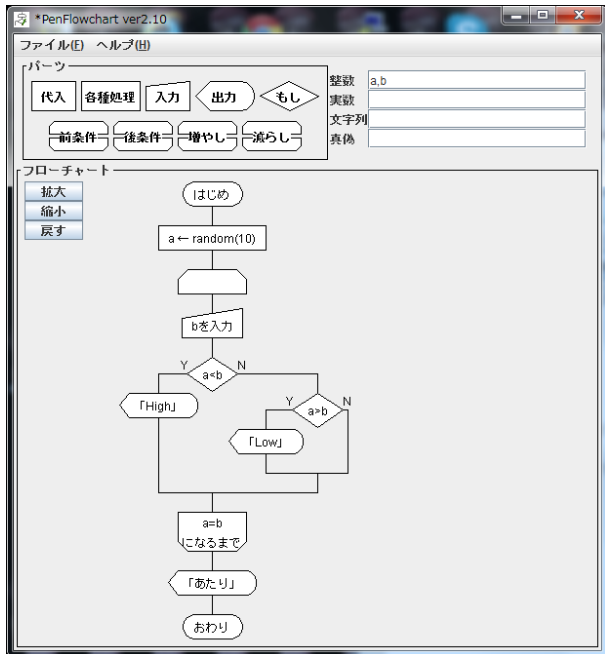


図 2 PenFlowchart の実行画面

なお、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://watayan.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元に、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

著者の一人は、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、課題を中心として演習を行なっている。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ので、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 回目までの演習内容は以下の通りで、ここまでで制御構造の基本を学ぶようになっている。

- (1) プログラミング環境の導入、例題プログラムを用いた使用法の修得、簡単なプログラムの作成

- (2) 逐次処理
- (3) 条件分岐 1
- (4) 条件分岐 2, 繰り返し 1
- (5) 繰り返し 2

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 回目までに欠席や課題未提出のなかった、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の平均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っており、特に 3 回目以降では大きな差がでている。サンプル数が少なく、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。

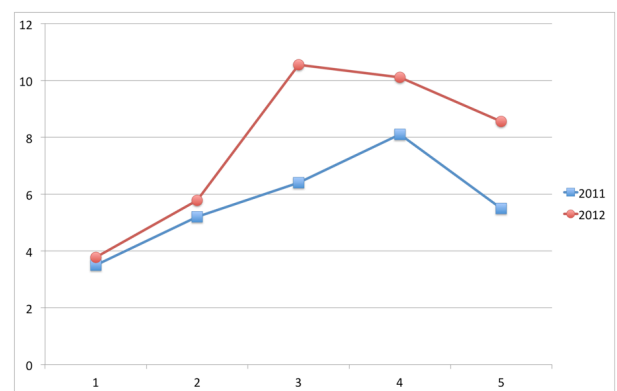


図 3 取り組んだプログラム数の平均

4.2 名古屋高等学校における実践

著者の一人の勤務校である名古屋高等学校では情報 B を履修させており、高 1 の 2 学期はプログラミングとアルゴリズムを中心に授業を行なっている。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。

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。また、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なった。以下、その結果について述べる。

アンケートの結果は表 3, 表 4 の通りである。表 3 で「忘れた」が多いのは、2013 年度になってからアンケートを行なったため、これらの授業を行なった 1, 2 年前のことを思い出せなかったためだと思われる。2012 年度の回答は表 3, 表 4 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 1 時間目の授業が 1 学期の期末テスト後であり、その時点ではまだ

PenFlowchart を作る構想さえなかったため、PEN だけで行なったことや、授業用の自作テキストが PEN の使用だけを想定して書かれていたこと、また、PenFlowchart を導入してからも頻繁にバージョンアップしたことにより PenFlowchart の使用を敬遠したためだと思われる。2012 年度は最初から両方を使うことを想定して、テキストもそれにあわせて改訂したため、PenFlowchart 中心の生徒が多くなったと考えられる。

表 3 どちらをメインで使ったか (人数)

年度	2011	2012
PEN 中心	75	22
PenFlowchart 中心	64	172
課題によって使い分け	50	33
課題の途中で使い分け	3	9
忘れた	56	32
(無回答)	16	15

表 4 一方だけを使うとしたらどちらを選ぶか (人数)

年度	2011	2012
PEN	131	32
PenFlowchart	115	235
(無回答)	18	16

期末テストは年ごとに問題が異なるため、全体の得点を比較することには意味がない。しかし毎回、フローチャートからプログラムと、その逆向きに書き換える問題を出題しているので、この結果を用いて比較を行なう。難易度は図 4 の例にあるように、ループと IF-ELSE のネスト程度のものである。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配慮した。

```

整数 a,b
a を入力する
もし a>0 ならば
  | b ← a
  | 繰り返し,
  |   | b を表示する
  |   | b ← b-1
  | を, b<0 になるまで実行する
を実行し, そうでなければ
  | 「?」を表示する
を実行する
「終わり」を表示する
    
```

図 4 期末テストの題材の例

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なう際に、2 学期の期末テストの点数によっておよそ 3 : 4 : 3 の割合で上位・中位・下位のランクに分けて集計を行なった。上記小問の

平均点 (10 点満点に換算してある) の推移は表 5 の通りである。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 5)。図 6 はこれらの得点分布であり、これによれば 0 点 (あるいはそれに近い) 生徒が確実に減っていると見られる。

表 5 小問の平均点推移

年度	2009	2010	2011	2012
上位	6.9	7.0	9.1	8.9
中位	4.5	4.4	6.7	8.0
下位	2.1	2.3	4.2	6.2
全体	4.5	4.6	6.6	7.7
人数	320	241	264	283

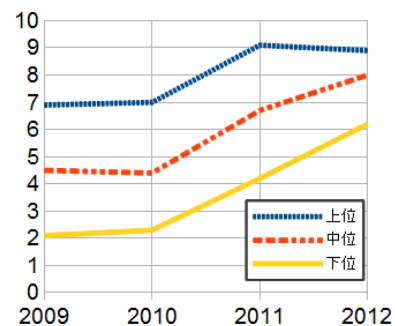


図 5 小問の平均点推移

PenFlowchart を導入することで、課題提出にも変化が現れた。ほぼ毎時間、小さいプログラムを課題とし、生成される PEN のプログラムをメールで提出させていたが、以前は未提出という評価だけは避けようとして、動かないプログラムでも平気で提出する生徒が目についた。その中には構文エラーのために実行できないものもあったが、これは前述したように入力支援ボタンで作られた構文を破壊したのもあれば、友人に画面を見せてもらって書き写すときに書き損じたものも多かった。PenFlowchart から生成されるプログラムではそのような構文エラーは発生しないし、隣の画面の真似をするためには構造を組み立てながらプログラムを作り上げるしかない。そのため、自分で 1 からプログラムを作ることが難しいような生徒でも、一定の学習効果をあげることができたのではないかと考える。また、教員側から見れば、動作が正しいかどうかだけを判定すればいいことになり、定量的には測定していないが、評価の手間が軽減するというメリットもあった。

5. 考察

5.1 PEN と PenFlowchart の選択

4.2 のアンケートを行な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

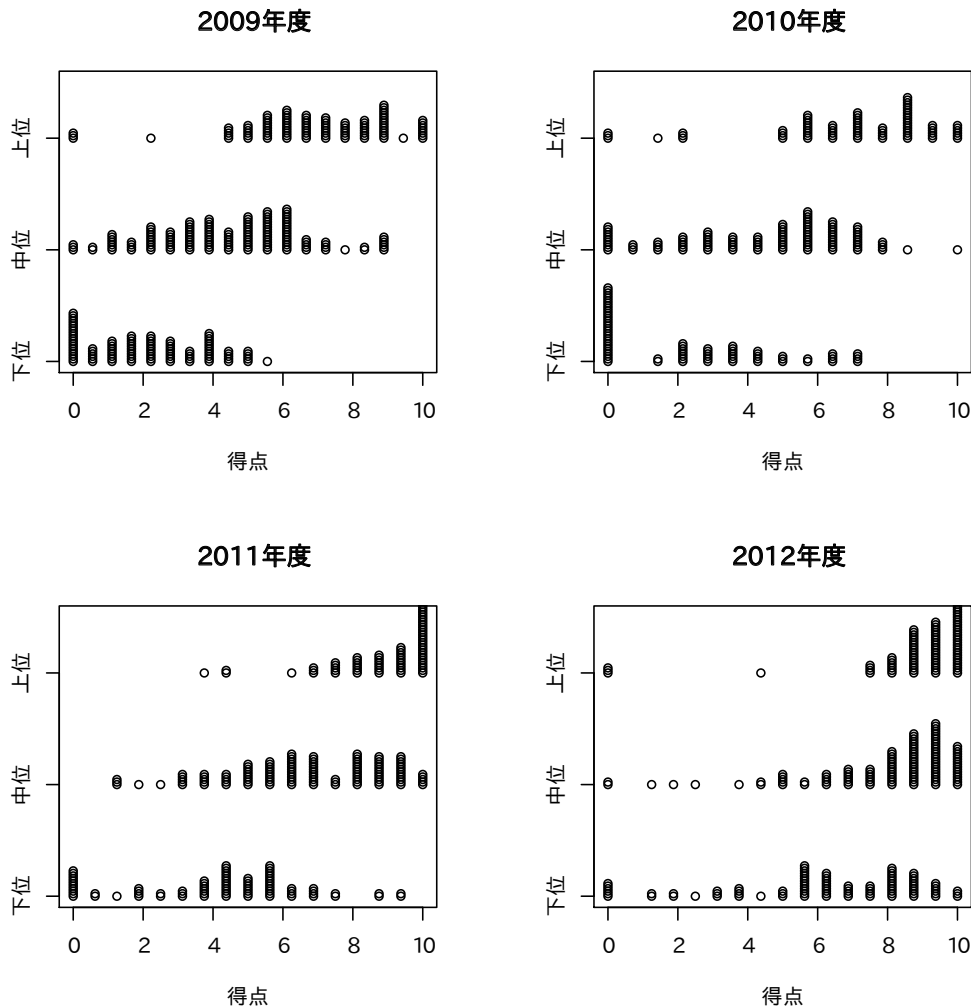


図 6 小問の得点分布推移

PenFlowchart を使うだろうということを予想した。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。

前節で述べたように、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はずである。

しかし、表 4 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集計した分析対象の小問の平均点は表 6 の通りであった。その結果、2011 年度はほとんど差がなかった。また、2012 年度は予想通り PEN を選んだグループの方が平均点は若干高かった。しかし、平均値の差について t 検定を行なったところ p 値は 6.3% であったので、予想を裏付ける有意差はなかった。ただし、得点分布 (図 6) にあるように対象とした小問の得点が高得点側に分布し、頭打ちになっている状態なので、分析方法を精査する必要がある。また、表 3 や表 4 を小問の成績のランク別に集計

表 6 表 4 の回答別の小問の平均点

	年度	
	2011	2012
PEN	6.4	8.3
PenFlowchart	6.8	7.7

しても、目立った差は見られなかった。

5.2 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、これは既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、プログラムを望む通りにつくり上げる力や、既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。定期テストではアルゴリズムの手順を追いかける問題も出題しているため、4.2 の分析対象の小問との相関を分析したが、どの年でも相関係数が 0.3~0.4 程度の弱い相関しかなかった。

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なっていきたい。
また、DNCL 以外のプログラミング言語とフローチャートの
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。

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なわ
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。

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 2013-09-11).
- [2] 日本経済再生本部：日本再興戦略, 首相官邸 (オンライン),
入手先 (<http://www.kantei.go.jp/jp/singi/keizaisaisei/>)
(参照 2013-09-11).
- [3] 高岡詠子：伝説の教育用プログラミング言語 Logo, 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. 21-26 (2006).
- [4] 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).
- [5] 飯作俊文, 飯塚重善, 徳永幸生, 杉山 精：Viscuit を用
いた初学者向けプログラム概念教育カリキュラムの提案,
情報処理学会全国大会講演論文集, Vol. 72, No. 4 (2010).
- [6] 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).
- [7] 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 2013-09-12).
- [8] 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 コンピユ
ータと教育 (CE), Vol. 2005-CE-081, No. 104 (2005).
- [9] 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).
- [10] 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).
- [11] 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).