解説

基般

プログラミング学習環境の自作
PenFlowchart によるプログラミング学習

中西　渉
名古屋高等学校

開発に至る経緯

筆者の勤務校である名古屋高等学校は，学年 12 クラス規模の私立男子校である．教科「情報」の授業が始まった当初は，プログラミング教育に適したソフトウェアを模索していた．これは，情報教室の端末を Linux にしたため，Excel の VBA などは使えず，最初は OpenOffice.org Calc のマクロを用いたが，手順が面倒であることなどから生徒の反応が非常に悪かったためである．

筆者は 2006 年に行われた「教育用プログラミング言語に関するワークショップ 2006」に参加し，PEN[1] という初学者向けプログラミング学習環境の存在を知った．PEN の実行画面を**図 -1** に示す．使われているプログラミング言語は，大学入試センター試験の情報関係基礎の出題に用いている手順記述言語 DNCL を拡張したものである．日本語ベースということでそのままでは入力が面倒になるため，構文や命令を簡単に挿入できる入力支援ボタンが用意されている．Java で開発されているので，実行環境の OS を問わないことがありがたかった．さっそく 2006 年度から勤務校の授業は PEN に切り替えた．PEN は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が充実しており，生徒たちがプログラムを作って実行することのハードルは下がった．

しかしプログラムが複雑になるにつれて，エラーに悩む生徒が増えてきた．PEN では本来なら，入力支援ボタンを正しく使っていれば構文エラーは発生しないはずである．しかし彼らは**図 -2** のような正しいプログラムを，わざわざ**図 -3** のように書きなおしてしまうことがある．ほかのプログラミング言語を知っている我々は「もし～を実行する」が「IF ～ END IF」のブロック構造に相当することが直感的に分かるが，生徒は日本語としての自然さを優先してこのようにしてしまうと考えられる．もちろん実行時にはエラーメッセージが表示されるのだが，そのメッセージを見て「『を実行する』を削除したのがいけなかった」と気付くことができる者はそもそもその行を削除したりしないのではないだろうか．

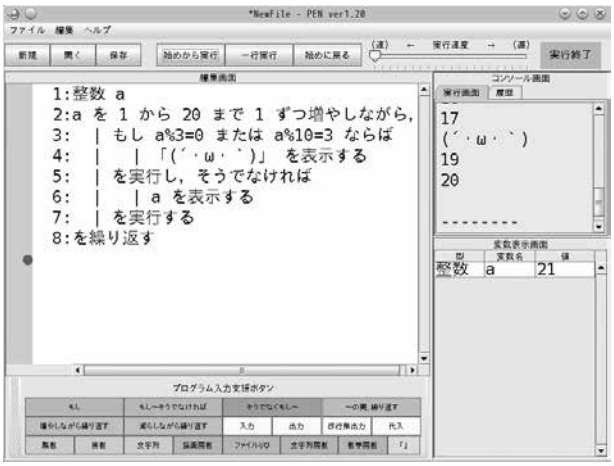

図 -1　PEN の実行画面

そこで構造を保ったプログラムを生成する仕組みがあればこのような問題がなくなると考え，フローチャートを作ることで PEN のプログラムを生成する PenFlowchart を 2011 年に開発した[2]．**図 -4** の

もし a=0 ならば
　|「ゼロ」を表示する
を実行する

図 -2　正しいプログラム

もし a=0 ならば
「ゼロ」を表示する

図 -3　間違ったプログラム

ように，画面上部のパーツをフローチャートにドラッグ & ドロップすることで，PEN 用のコードを生成することができる．実行やデバッグは PEN の機能をそのまま利用する．

コードからフローチャートに書き戻す機能も実装したが，これは松澤芳昭氏らの Block Editor[3] が Java のコードからブロックに変換する機能を持っていることに影響を受けたものである．なお，フローチャートからコードへの変換は自動的に行われるが，逆は明示的にボタンを押さないと変換しない．これは書いている途中のコードを解析しても構文エラーになるからである．

作成したフローチャートは EPS または PNG フォーマットで画像として出力できるため，プリントやスライド，テスト問題を作るときに活用している．

アルゴリズムの学習において，フローチャートが最善のものだと考えているわけではない．PAD のような構造化チャートを使うことも検討したが，高校の教科書はすべてフローチャートを用いているのでそれは適切でないと考えた．また，フローチャートは自由に線がひけるためスパゲティ化してしまうという批判があるが，PenFlowchart ではパーツの加除しかできないので構造は保たれる．

PenFlowchart を用いることによって，前述したようなブロック破壊による構文エラーはなくなり，実習はスムーズに進むようになった．図によるプログラミングを行うことで，フローチャートは分かるがコードが分からない生徒を生むことになるのを危惧したが，期末テストの得点を分析したところ，それ以前よりもフローチャートとコードの関係についての理解がよくできているという結果が得られた[4]．

図 -4　PenFlowchart の実行画面

当初予想していなかった副作用もある．ほぼ毎時間の授業で，課題のプログラムをメールで提出させていたのだが，PEN の頃は隣の生徒の画面を見ながら，入力支援ボタンを使わずに手で書き写す生徒もいた（必然的に構文エラーのまま提出する者も多い）．その理由は，前述したブロック構造を理解しない者が入力支援ボタンを使うと，予想外の形で構文が挿入されてしまうため，かえって面倒を感じることにあるらしい．しかし PenFlowchart でその通りのプログラムを作るには結局フローチャートを自分で完成させなくてはいけない．その結果，教えてもらいながらでも自力でパーツを配置してフローチャートを作成するようになった．PenFlowchart におけるパーツの追加は，PEN における入力支援ボタンに対応しているため，この訓練がブロック構造の意識につながったのではないかと推測する．

いくつかの大学や専門学校でも PenFlowchart を

	DNCL	BASIC
構文	《変数》を入力する	INPUT《変数》
定義	<INPUT :"を入力する"> InputStat() : {} {Ident() <INPUT>}	<INPUT :"INPUT"> InputStat() : {} {<INPUT> Ident()}

図 -5　構文定義ファイル

使用していただいた．学生や教員からの要望を教えていただくことで，機能追加や改善を行った．たとえばすでに作ったプログラムをループに入れて繰り返し実行したいというとき，以前は外側のループの中ですべてを作りなおさなくてはいけなかったので，カット & ペーストでパーツを移動できるようにしたことなどである．

DNCL 以外の言語

筆者の授業は自作プリントを中心に行ってきたので DNCL でも支障なかったが，新課程の情報の教科書で DNCL を使っているものは皆無であるため，PenFlowchart を使うなら教科書は使えないという状況が今後も続くことになる．そこで他言語への移植を試みた．

まず最初に C++ のコードを生成するものを作ったが，授業で使うとなるとコンパイラなどを用意して使い方を教えなくてはいけないので，とても現実的に使えるようなものではなかった．また，コードからフローチャートを生成するには構文の解釈が必要となるが，C++ のパーサを書くのは筆者には困難であるため断念した．

続いて，BASIC 版に着手した．これは，PEN は構文解析部分を JavaCC で作っているので，**図 -5** のように書き換えれば容易に BASIC 版になるからである．その「翻訳」さえ行えば，すでにある DNCL の実行環境を利用することで，プログラムの実行やデバッグを容易に行うことができる．

このようにして BASIC 用の PenFlowchart を作成したが，実はこのような BASIC を用いている

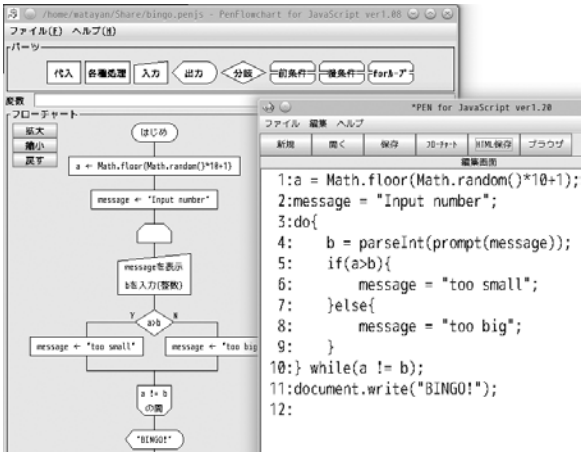

図 -6　PenFlowchart fot JavaScript の実行画面

高校の「情報」の教科書はない．実教出版の「情報の科学」では表計算の VBA を用いているが，セルを配列とみなすという面白いアイディアを採用しているため，教科書のプログラムをこの BASIC 版で実行することはできない．そこで JavaScript 版を作ることにした．これにより JavaScript のプログラムを扱う日本文教出版や東京書籍の教科書に対応できる．

JavaScript の実行環境を内製することは（筆者にとっては）容易ではないので，PEN からプログラム実行に関する機能を取り去り，生成されたプログラムを埋め込んだ HTML ファイルを出力し，ブラウザに読み込ませて実行する機能を追加した．実行画面を**図 -6** に示す．プログラムを書き換えたときには再度 HTML ファイルを出力してブラウザでリロードしなくてはならないなど，余計な手数が増えることで生徒が混乱することを心配したが，今年度（2014 年度）の授業で使ってみた様子では手順として覚えてしまえば大したことではないと捉えているようである．

自由な環境

勤務校では今後の授業で JavaScript 版を使っていくことになるが，開発についてはいろいろ迷いがある．たとえば，フローチャートが足枷になっていると感じられることがある．JavaScript で普通に

用いられる else if などをフローチャートで表現するのは難しいため，正しい構文を使っていてもフローチャートへの変換ができないことがあるのだ．ソースコードを自由に書けるようになった者にとっては，これが不自由な制限に思われることが予想される．高校の授業で教科書通りのプログラムを扱うにはフローチャートが使えるものでいいかもしれないが，それ以上のことを望む場合にはフローチャートを使えないようにしたものを用意すべきであろう．フローチャートを使えなくした環境でも HTML のタグを自動的に埋め込んだり，PEN の入力支援ボタンを使ったりできるという点で，テキストエディタでの作業よりも楽になると考えられる．

フローチャートを使わない環境を作ることは，すでにある機能をただ停止すればいいので容易である．PenFlowchart は GPL（GNU General Public License）で公開しているので，ほかの利用者がそのような変更を加え，それを再配布することも自由である．これは PEN が GPL で公開していたことに由来し，それにより PenFlowchart やほかの言語版を作ることができた．

多くの情報教室で授業支援ツールが用いられているように，授業で使う環境やツールはカスタマイズが必要になることが多い．したがって，そこで用いられるソフトウェアは自由であることが望ましいと考えている．ソースコードが手に入るなら，それをほんの数行変更することで，自分の望む環境が得られることもある．勤務校の情報教室の端末を Linux にしたのは，そのような自由を求めてのことであった．授業で使うソフトウェアをゼロから作り上げることは難しいにしても，既存ソフトウェアのカスタマイズや機能の追加・削除を現場の教員が行うことは不可能ではないし，他人のソースコードを読んで理解するのはいい勉強にもなる．そのように自分で手を動かして不満を解消するような教員が増えることを願っているし，筆者自身もそうありたい．

なお，PenFlowchart とその他言語版は筆者のサイト http://watayan.net/prog/ で配布している．

参考文献

1）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 (Aug. 2007).
2）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, Vol.13 (Jan. 2012).
3）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, Vol.2 (Mar. 2013).
4）中西　渉，辰己丈夫，西田知博：Penflowchart によるプログラミング導入教育の評価，情報処理学会研究報告　コンピュータと教育（CE），Vol.2012-CE-121, Vol.9 (Oct. 2013).

（2014 年 11 月 30 日受付）

中西　渉（正会員） watayan@meigaku.ac.jp
名古屋高等学校教諭．担当教科は情報と数学．コンピュータと教育研究会運営委員．初等中等教育委員．