

初学者用ブロックプログラミング環境 wPEN の改良

下村 亮太¹ 中西 通雄² 松浦 敏雄³ 西田 知博⁴ 安留 誠吾⁵ 宮本 友介⁶

概要: 大学入学共通テストの「情報関係基礎」のプログラミングの問題には、従来の大学入試センター試験の DNCL が引き続き用いられた。DNCL を Web ブラウザ上で実行できる環境として、筆者らは wPEN を開発し、2019 年度から大学や高校でのプログラミング初学者教育に用いてきている。wPEN では、画面上で短冊形の命令を組み合わせて作成されたプログラムを、そのまま JavaScript に変換しているために文法チェックや実行制御の限界はあるものの、今回いくつかの改良を行って使いやすくなったので報告する。

RYOTA SHIMOMURA¹ MICHIO NAKANISHI² TOSHIO MATSUURA³ TOMOHIRO NISHIDA⁴
SEIGO YASUTOME⁵ YUSUKE MIYAMOTO⁶

1. まえがき

新しい学習指導要領は、小学校では 2020 年度、中学校では 2021 年度から全面実施され、高校では 2022 年度から年次進行で実施される。小学校では、プログラミング教育が必修となり、コンピュータに意図した処理を行わせるための論理的な思考力「プログラミング的思考」などを育てる^[1]。また、中学校においてプログラミングに関する内容を充実するとともに、高校では科目として「情報 I」と「情報 II」が新設され、「情報 I」は必修科目となる。

情報 I の「コンピュータとプログラミング」の単元では、プログラミングを学習する。新学習指導要領を円滑に実施することを目的として、文部科学省は高校の情報教員向けに、「高等学校情報科『情報 I』教員研修用教材(本編)」を公開している^[2]。研修用教材の第 3 章「コンピュータとプログラミング」では、まず最初に Python 版の資料が公開され、JavaScript 版、VBA 版、ドリトル版、Swift 版の資料も公開されている。

しかし、現状では、現行の学習指導要領でプログラミングを取り扱う「情報の科学」の採択率は 2 割に過ぎず、プログラミングを実習テーマとした教科指導の割合も少ない^{[3],[4]}。そのため、学習しやすいプログラミング環境だけでなく、教員も取り扱いやすいプログラミング学習環境を提

供する必要がある。

初学者に対するプログラミング教育では、ブロック型言語かテキスト型の言語が用いられている。ブロック型のプログラミングは、アルゴリズムを考えて組み立てる基礎を学ぶに適している。一方、C や Java 等の自由に記述することができるテキスト型では、タイピングミスによって英語でエラーが出力されるなどして、初学者の意欲を損なう可能性がある。

そこで、筆者らは高校や大学のプログラミング初学者教育で授業を円滑に進められるようにするため、ブラウザ上で動作するプログラミング学習環境 wPEN を開発し、授業に用いてきた^{[5],[6]}。wPEN では、Scratch などのブロックと同様に、短冊形の命令を組み合わせてプログラムを構成するので、プログラムの構造の誤りやタイプ入力の誤りを減らすことができる。

教員は、使用できる短冊を限定した演習問題を出題するなどして、学習の進度に合わせた環境を提供できる。学習者は、短冊を並べてプログラムを作成し、ブレークポイントを設定したり、表示される変数値をみながら短冊ごとのステップ実行を用いてデバッグできる。

本研究では、ブラウザ上で動作するプログラミング環境 wPEN の改良について述べる。

2. 関連研究

大学入試センターの「情報関係基礎」のプログラミングの問題では、大学入試センター試験用手順記述標準言語 DNCL が用いられてきた^[7]。そして 2021 年 1 月の大学入

¹ 大阪工業大学 情報科学部 情報ネットワーク学科

² 追手門学院大学 経営学部

³ 大和大学 理工学部

⁴ 大阪学院大学 情報学部

⁵ 大阪工業大学 情報科学部

⁶ 大阪大学 大学院人間科学研究科

学共通テストでも DNCL が用いられた。この DNCL の実行環境として最初に作られたのが PEN である [8],[9]。その後、どんくりや XTetra、WaPEN などが作られた。

2.1 PEN

PEN は 2006 年に作成された [8]。手続き的なプログラミングを短時間で容易に習得できるように、構文エラーを生じにくくするための入力支援機能や、プログラムの実行の様子を把握しやすくするためのデバッグ機能（ステップ実行、変数値表示機能、ブレークポイント、実行速度の調整など）が搭載されている。また、学習者がプログラムの実行結果を視覚的に確認できるように図形描画命令が提供されている。

しかし、PEN は Java で開発されたアプリケーションプログラムであるため、新たなアプリケーションの導入が制限されている学校現場では利用しにくく、個人のパソコンへインストールするには敷居が高いという問題点があった [8],[9]。なお、PEN では変数の宣言文を追加するなど、DNCL を拡張して xDNCL としている [10]。

2.2 どんくり

どんくりは 2018 年に作成され、ブラウザ上での DCNL の実行環境を提供している [11]。どんくりは、アルゴリズムの良し悪しの検討が容易に行えるように、プログラムの実行時の繰り返し回数や実行速度などを計測・表示するプロファイラが搭載されている。どんくりでは、キーボードを用いてプログラムの作成・編集を行うため、タイピングミスによるエラーが懸念される。また、デバッグ機能（ステップ実行や変数値表示機能、プログラムカウンタの表示など）が搭載されていない。また、どんくりは DNCL を忠実に実装しているため、変数の宣言文やキーボードからの入力を受け付ける命令はない。

2.3 XTetra

XTetra は、2019 年の U-22 プログラミング・コンテストで表彰された Tetra の改良版である [12],[13]。画面上に入力支援ボタンが配置されており、入力支援ボタンをクリックすると、変数名や値を入力するダイアログボックスが表示される。そのため、学習者は必要最低限のタイピングでプログラムを作成できるが、入力の手間が増える短所もある。また、DNCL プログラムの停止、変数リストと実行ログによるプログラムの追跡を行うことができる。さらに、プログラムにエラーが生じた場合、エラーの発生した行の直下に詳細なエラーメッセージが表示され、プログラミング初学者の学習を円滑に進められるように設計されている。XTetra は PEN や wPEN のようなプログラムカウンタのアイコンの代わりに、次に実行する行の背景色を緑色にし

て表現している。なお、XTetra には変数の宣言はなく、全角の英数字はプログラム実行時に半角に変換される。

2.4 PenFlowchart, WaPEN

PenFlowchart は、フローチャートで描いたプログラムを、DNCL に変換して実行することができる [14],[15]。また、フローチャートから DNCL への変換も可能である。

フローチャートは、プログラムの構造が図示されるため、初学者にとって分かりやすいという利点がある。しかし、一般には、フローチャートは非構造化プログラムも書いてしまうため、教育の場で用いる際には注意が必要である（PenFlowchart では非構造化プログラムが作成できないように配慮されている）。

WaPEN は、PenFlowchart をブラウザ上で動作するようにしたものである [16]。ブラウザの画面にプログラム入力画面、入力支援ボタン、フローチャート、テキスト・グラフィックスの実行画面、サンプルプログラムの読み込みボタンなどを配置し、プログラムの編集が容易な環境を提供している。しかし、WaPEN には変数値表示機能は搭載されていない。

本節で述べた実行環境の機能を比較した（表 1）。本研究では、wPEN の機能のうち★印をつけたところを中心に改良した。この詳細は 4 章で説明する。

表 1 関連研究の比較表

	ステップ 実行	★実行 速度 調整	入力 処理	★関数 定義	★変数 宣言	描画 命令	プロ ファイラ 関数
wPEN	○	×	○	×	無	○	×
PEN	○	○	○	○	選択 可能	○	×
WaPEN	○	×	○	○	選択 可能	○	×
XTetra	○	×	○	○	無	○	×
どんくり	×	×	×	○	無	×	×

3. wPEN の概要

wPEN は、大阪市立大学の松本が CBT(Computer Based Testing) を用いた試験実施環境に組み込むことも視野に入れて、DNCL 用に作成したブラウザ上で動作するプログラミング環境である [17]。

wPEN は、試験問題作成機能を持つ wPEN for Teachers(wPEN-T と略す) と受験及び解答機能を持つ wPEN for Students(wPEN-S と略す) から構成されている。教員は wPEN-T を用いて例題や演習課題を構成する。学習者は wPEN-S を用いて、例題を実行したり、課題に対してプログラムの作成と実行を行う。図 1 は wPEN-S の画面例であり、右側の選択肢欄にはプログラムを構成する要素が短冊として配置されている。学習者は、左側のプログラム編集欄の追加したい場所に短冊をドラッグ&ドロ

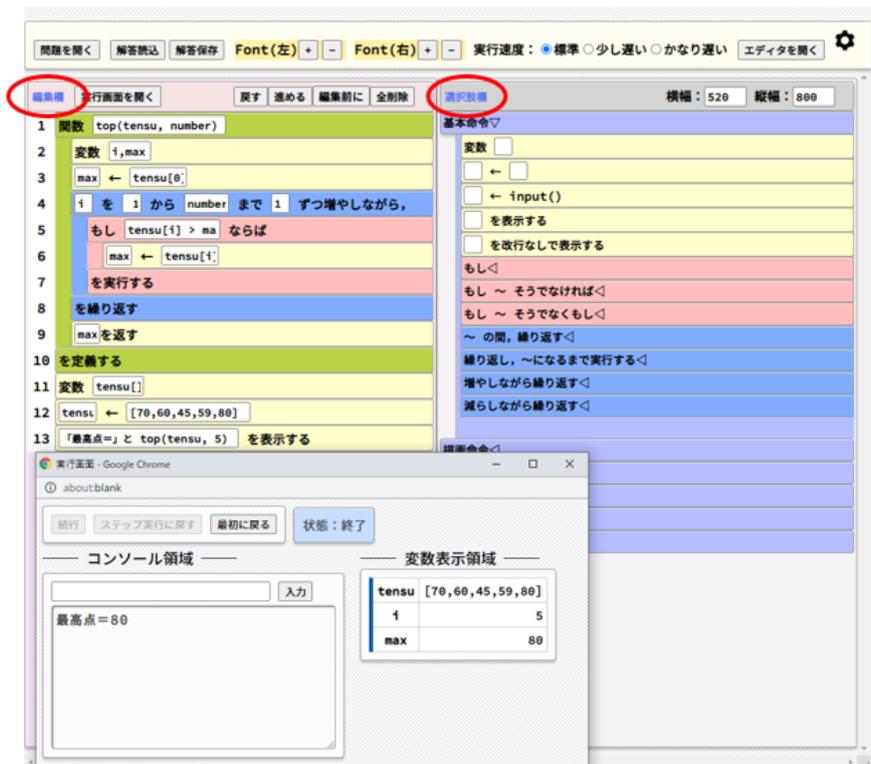


図 1 学習者用 wPEN-S の画面表示例

プすることでプログラムを作成する。左側の編集欄の「実行画面を開く」ボタンを押すと、編集欄に配置されている DNCL プログラムが JavaScript に変換され、実行される。

wPEN は JavaScript で作られており、Web サーバまたはローカルディスク上にファイルを設置して、ブラウザで読み込むことで動作させることができる。

3.1 基本機能

(1) 「問題を開く」ボタン

このボタンをクリックすると、教員が作成した例題や演習課題を読み込むためのファイルダイアログが開く。ファイルの読み込みに成功すると、wPEN-S の画面に例題や演習課題が反映される。

(2) 「解答読込」ボタン

このボタンをクリックすると、学習者が解答した演習課題を読み込むためのファイルダイアログが開く。ファイルの読み込みに成功すると、wPEN-S の画面に学習者の解答が反映される。

(3) 「解答保存」ボタン

このボタンをクリックすると、ファイル名を入力するためのダイアログが開く (図 2)。解答のファイルは、使用しているブラウザに設定されたディレクトリに XML 形式で保存される。

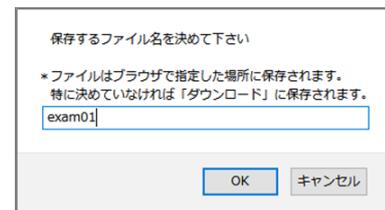


図 2 ファイル名を決めるダイアログ

(4) 「Font (左)」ボタン

このボタンの右側に配置されている「+」ボタンをクリックすると、編集欄の短冊を拡大することができる。また、「-」ボタンをクリックすると編集欄の短冊を縮小することができる。

(5) 「Font (右)」ボタン

このボタンの右側に配置されている「+」ボタンをクリックすると、選択肢欄の短冊を拡大することができる。また、「-」ボタンをクリックすると選択肢欄の短冊を縮小することができる。

(6) 「実行速度」ボタン

プログラムを実行する速度を調整することができる。この詳細は後述する。

(7) 「エディタを開く」ボタン

このボタンをクリックすると、テキストコードと短冊を相互変換するエディタが開く。この詳細は後述する。

(8) 「環境設定画面」ボタン

このボタンをクリックすると、変数の取り扱いを設定

することができる。この詳細は後述する。

4. wPEN の改良内容

本研究では、初学者のプログラミング演習を円滑に進められるように、主として学習者が利用する wPEN-S の問題点の解決を目指し、次の改良を行った。

4.1 使用できる短冊の種類を追加

4.1.1 関数定義用の短冊の追加

新しい高等学校学習指導要領では、「関数を用いてプログラムをいくつかのまとまりに分割してそれぞれの関係を明確にして構造化することなどを扱うことが考えられる。」と書かれている^[19]。しかし、2018年度のwPENには関数を定義する短冊が実装されていない。そこで、wPENに関数定義用の短冊を追加した。

4.1.2 データの可視化機能の追加

wPENはPENから引き継いだグラフィック機能を持っているため、それを使えばグラフを描画することができる。しかし、与えられたデータを可視化するための手段として利用するとき、座標を計算してグラフを描画すると難易度が高いため、wPENに棒グラフや線グラフを簡単に書ける短冊を追加した。具体的には、金利計算や人口の増減、数学や物理などの事象を扱うことを想定している。

4.2 「最初に戻る」ボタンの改良

wPENの実行画面の「最初に戻る」ボタン(図1参照)を押して、再度プログラムの動きを確認したいとき、コンソール領域に出力結果が表示されない問題があった。これは、「最初に戻る」ボタンを押しても、実行画面内に宣言されている変数は削除されていないため、再宣言されると誤認識していたことが原因である。そのため、プログラムの出力結果を確認するとき、実行画面を閉じて、また開くという動作が必要となり手間がかかる。また、プログラム編集欄にプログラムカウンタが表示されないという問題点があった。ステップ実行で確認するときも、同じように実行画面を閉じて、再度開くという動作が必要であった。そこで、実行画面を開きなおす仕組みに変更することで、これらの問題を解決した。

4.3 変数の扱いの設定

4.3.1 変数宣言の有無

wPENはJavaScriptに変換して実行するため、変数を宣言しないでプログラムを作成することができる。教える側の立場の違いによって、変数宣言を必要とするかどうかを設定できるようにした。ただし、JavaScriptでは型を指定した変数宣言はできないため、PENのように型宣言をすることは実現できていない。

4.3.2 変数名の文字種制限

4.3.1で述べたようにJavaScriptで実装しているため、日本語(かな漢字まじり)の変数名も扱える。しかし、変数名にASCIIの英数字文字だけが使える言語(Cなど)へ移行していく場合もあるため、制限できるようにした。

4.3.3 設定画面

4.3.1と4.3.2で述べた機能を設定する画面を追加した(図3)。

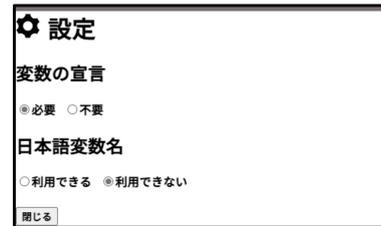


図3 環境設定画面

4.4 実行速度を調整する機能の追加

2018年度のwPENは、デバッグ機能にステップ実行がすでに実現されている。しかし、プログラムの行数が多くなったり、二重ループを利用したプログラムでは、ステップ実行ボタンを何度も押すことになり、デバッグの際に負担がかかる。

そこで、wPENの表示画面上部に実行速度を調整する機能を追加した(図4)。なお、実行速度はCPUやメモリの性能に依存する。



図4 実行速度調整機能

4.5 DNCLの演算子の仕様に対応

wPENで対応していなかったDNCLの全角演算子(+、-、×、÷、%、≠、>、<、≤、≥)を使用できるようにした。

4.6 連結比較演算子の対策



図5 比較演算子を二つ連結したプログラム

初学者は、図5のような比較演算子を2つ用いた条件式を書いてしまう傾向がある。DNCLでは、CやJavaと同様に、このような条件式は許されていない。従来のwPENではチェックできていなかったため、今回エラーとして明示するようにした。

4.7 選択肢欄の制御構文の改良



図 6 一部の短冊を取り出せる機能

図6の「もし～そうでなくもし」ような短冊は、薄赤色の部分をドラッグして、編集欄にドロップすると、続く4つの黄色の短冊が展開される。今回の改良では、この一部、たとえば「を実行し、そうでなくもし□ならば」だけをドラッグ&ドロップできるようにした。この機能は、図7のようなプログラムを変更して、図8のように書き換えたいときに便利である。



図 7 条件分岐のプログラム



図 8 図7を変更したプログラム

4.8 短冊とテキストコードの相互変換

4.8.1 短冊からテキストコードへの変換

教員が wPEN の学習者用指導書あるいは教材サイトを作成するときには、短冊を組み合わせて構成したプログラム全体あるいは断片を画像として取り込んでいた。プログラムを画像で表示すると、条件分岐や繰り返しの制御構造がわかりやすい。しかし短冊の命令をそのまま DNCL のテキストコードとして扱いたい場合もある。そこで、「短冊読み取り」ボタンをクリックすることで、短冊で構成されたプログラムの文字列を、インデントを反映したテキストに変換して取り出せるようにした。また、行番号を自動的につけることもできるようにしたので、プログラムの解説を書くときに活用できる。また、コードに変換された DNCL プログラムをテキストファイルとして保存することができる。

4.8.2 コードエディタ

教員がサンプルプログラムを用意するときには、プログラムを少しだけ変更して実行したいことがある。そこで、テキストコードを編集して、簡易的な短冊の形に変換して実行できるコードエディタ機能を実装した。ここで簡易的な短冊とは、編集欄の短冊の中身を編集できないという意

味である。また、コードエディタは、テキストファイルとして保存しておいた DNCL プログラムを読み込むこともできる。これら2つの機能は主に教員用の機能として作成したが、特に隠し機能とすることはせず、短冊でのプログラミングに慣れてきた学習者でも使えるようにしている。



図 9 短冊読み取り画面

4.9 エラーメッセージ表示の改良

改良前の wPEN では、図10のように文法的に誤ったコードを短冊に書いた場合、実行してもエラーメッセージが全く出力されないことがあった。これでは、学習者が困るので、必ずエラーが表示されるようにした。

以下に、誤りがあるプログラム及びその実行結果をそれぞれ図10、図11に示す。

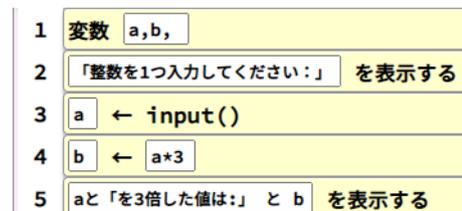


図 10 誤りがあるプログラム



図 11 誤りがあるプログラムの実行画面

5. 評価

5.1 評価方法

2021年2月10日7限目(50分)に、普通科高校のプログラミング未経験の一年生38名にwPENを試用してもらった。具体的には最初の10分間でwPENの操作説明を行った後、約30分間に例題でプログラムの流れを確認し、例題を参考にしながら練習問題のプログラムを2つ作成してもらい、最後の10分でアンケートを実施した。なお、練習問題(3)については授業時間が足りなかったため取り組んでいない。

解説に利用した例題のプログラムを以下に示す。

- (1) 「こんにちは」と「wPENの世界へようこそ!」を2行で出力(表示)するプログラムを見てみよう
- (2) 「今日の日付は」と「2月10日です」を2行で出力(表示)するプログラムを作ってみよう
- (3) キーボードから1つの整数の入力を受け付け、それを表示するプログラムを見てみよう
- (4) キーボードから2つ整数の入力を受け付け、2つの整数の和を変数waに代入して、その結果を出力するプログラムを作成してみよう

練習問題に用意したプログラムを以下に示す。

- (1) キーボードから2つ整数の入力を受け付け、2つの整数を表示するプログラムを作成してみよう
- (2) キーボードから三角形の底辺と高さを入力し、面積を出力するプログラムを作成してみよう。
- (3) キーボードから入力された時間(分)を秒に変換して表示するプログラムを作成してみよう

アンケートは選択形式の質問を4肢択一で5つ、2択で1つ、記述形式の質問を3つ、アンケート用紙を用いて行った。回答に要した時間は10分程度であった。

5.2 評価結果に対する考察

図12にwPENの操作方法に関する評価結果を示す。図13に練習問題の感想を示す。

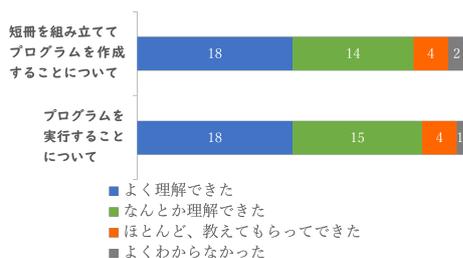


図12 高校生によるアンケート結果

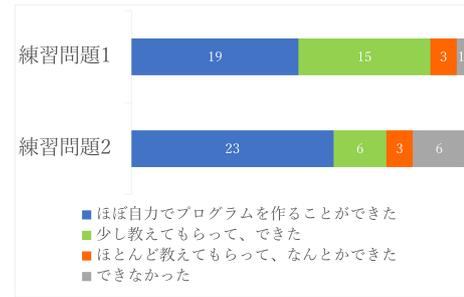


図13 練習問題の感想

高校での試用評価は50分間の1回だけであるが、日本語表記された短冊を組み合わせてプログラムを作成して実行する操作は、理解してもらえた(図12)。このことから、wPENはブロック型の利点が活かされており、プログラミング初学者にとって使いやすい実行環境になっていると考えられる。

また、プログラムを自分の力で作成することについては、むずかしさを感じた生徒も6名いたが、おおむね実行できていた(図13)。練習問題(2)では、代入文を使用するため、難しい問題と想定していたが、例題や練習問題を通してwPENによるプログラミングに慣れが生じたのか、自力でプログラムを作ることができた生徒が多くなった。一方、代入文の使い方を理解できていない生徒も見受けられ、筆者も机間巡視中に質問を受けた。

筆者は初めに学習環境の設定で変数名に全角文字を使用することを禁止するように指導したが、一部の生徒は全角文字で変数を宣言していた。練習問題では、キーボードによる入力処理を取り扱った。机間巡視中、一部の生徒が実行画面の状態が「入力待ち」になる前に、実行画面内の入力ボタンを押して、意図的に実行結果例に合わせる生徒が見受けられた。また、実行時エラーが発生しエラーメッセージが英語で表示されたとき、どうすれば良いのかわからず筆者に対応を求める生徒も数名ほど確認した。

6. 今後の課題

アンケートの自由記述欄や、本稿の著者グループのミーティング時に頂いた質問などから、wPENの改善点が浮上した。

(1) 半角の'(アポストロフィ)が使用できない

評価者に出力文を使ったプログラムを作成してもらった。このとき、半角の'(アポストロフィ)を使用すると、エラーが発生した。これは、xDNCLからJavaScriptへ変換するときに、全角の鉤括弧("「”、””)を半角の'(アポストロフィ)に正規表現で置換されていたことが原因であった。

(2) 全角文字を禁止する設定を標準にする

半角と全角の区別がつかないプログラミング初学者も

多く、つい全角文字を使ってしまうため、wPEN の標準設定として変数名に全角文字の使用を禁止しておくべきだと考える。または、実行画面内の変数表示領域で全角文字が使用されていることがわかるようになるか、wPEN 全体の文字フォントを全角と半角の区別が明確に付くものに変更することも考慮すべきと考える。

(3) 追加した短冊の評価

本研究で関数定義やグラフ描画などの短冊を追加したが、50分という時間の制約から被験者に使用してもらう機会はなかった。今後、追加した短冊の有用性を検証する必要がある。

(4) 実行画面のエラーメッセージの日本語表記

実行画面内にエラーメッセージが出力されるようになったが、英語表記のため、学習者にとってはどの短冊でエラーが生じているのか特定することが難しい。今後の改善として、初学者が間違えやすいエラーを収集し、そのエラーのみ日本語で表示するなどの対応が必要だと考えられる。また、エラーが発生した行番号を表示することも今後の課題である。

7. 結論

本研究では、wPEN に関数定義の短冊を追加した。また、実行速度調整機能を追加し、デバッグ機能を拡張させた他、普通科高校の生徒に評価してもらった。その結果、wPEN をプログラミング学習に利用しやすい実行環境にできた。しかし、授業時間の制約から、基本的な逐次処理だけ行ったため、条件分岐や繰り返し文、追加した関数定義やグラフ描画処理を利用した試用を行えていないため、wPEN を利用したプログラミングの授業を実施して評価を行い、さらに改良を加えていくことが課題である。

謝辞

本研究の一部は、JSPS 科研費 JP20K12112, JP19K12281, JP17K01088 の助成を受けたものです。

参考文献

- [1] 文部科学省: 小学校プログラミング教育の手引き (第二版) https://www.mext.go.jp/content/20200214-mxt_jogai02-000004962.002.pdf (2021/2/15 確認)
- [2] 文部科学省: 高等学校情報科「情報 I」教員研修用教材 (本編), 文部科学省, https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm (2021.1.30 参照)
- [3] 文部科学省: 情報教育に関する資料, 教育課程部会 情報ワーキンググループ (第 1 回) 配布資料 (2015), https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/_icsFiles/afieddfile/2015/11/11/1363276.08.1.pdf (2021.2.16 参照)
- [4] 小松原潤子: 「キミのミライ発見」取材を通して見た高校の情報への歩み ~2012 年に描いた未来図とともに, 情処誌, べた語義 vol.108, pp.1058-1061 (2020.10)
- [5] 大宮大地, 松本嵩大, 松浦敏雄, 中西通雄: 試験問題作成機能と学習及び受験用機能を持つ DNCL プログラミング環境, 情処研報, IPSJ-CE 201903-148 (2019.3)
- [6] 中西通雄, 宮本雄介, 安留誠吾, 西田知博, 松浦敏雄: 短冊形のコードによるプログラミング入門環境, 教育システム情報学会全国大会論文集 pp.47-48 (2020.9)
- [7] 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, <https://www.dnc.ac.jp/albums/abm.php?f=abm00039945.pdf> 令和 2 年度試験 センター試験用手順記述標準言語 (DNCL) の説明.pdf(2021.1.29 参照)
- [8] 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, 情処論, Vol.48, No.8, pp.2736-2747 (2007.8)
- [9] 西田知博, 原田章, 中西通雄, 松浦敏雄: 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, 情処論「教育とコンピュータ」, Vol.3, No.1, pp. 26-35 (2017.2)
- [10] 初学者向けプログラミング学習環境 PEN : xDNCL 言語マニュアル, <https://pen.moe.hm/#!/manual/xdncl.md> (2020.2.16 参照)
- [11] 本多佑希, 兼宗進: ブラウザ上で動作する DNCL 学習環境「どんくり」の開発, 情処研報 2018-CE-147, No.10, pp.1-4 (2018.12)
- [12] 大門巧: Tetra, <https://sites.google.com/view/tetra-dncl> (2021/2/16 確認)
- [13] 大門巧: XTetra, <https://t-daimon.jp/XTetra/> (2021/1/31/確認)
- [14] 中西渉: PenFlowchart の開発, 情処研報 2012-CE-113, No.13, pp.1-7 (2012.2)
- [15] 中西渉, 辰巳丈夫, 西田知博: PenFlowchart によるプログラミング導入教育の評価 情処研報 2013-CE-121, No.9, pp.1-9 (2013.10)
- [16] 中西渉: WaPEN... DNCL の Web ブラウザ上の実行環境におけるフローチャートなどの実装, 情報処理学会, 2018 情報教育シンポジウム, pp.210-214 (2018.8)
- [17] 松本嵩大: プログラム実行可能な CBT システムの設計と実装, 大阪市立大学大学院創造都市研究科修士学位論文 (2018.3)
- [18] 大宮大地: 試験問題作成機能と受験及び解答機能を持つ DNCL プログラミング環境, 大阪工業大学情報科学部卒業論文 (2019.1)
- [19] 文部科学省: 高等学校学習指導要領 (平成 30 年告示) 解説情報編, 文部科学省, https://www.mext.go.jp/content/1407073_11.1.2.pdf (2021/1/31)