

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

大宮 大地¹ 松本 嵩大² 松浦 敏雄³ 中西 通雄¹

概要: 大学入学共通テストに「情報」を加える意向が示され、その中にプログラミングの試験を入れることも含めて検討されている。この試験ではプログラム作成能力を問うことになるだろうが、商用の特定の言語によらず、日本語表記の疑似言語を使用するのが適切と考えている。Computer Based Testing を試験の前提とすると、プログラムコードを自由にタイピングさせるのではなく、プログラムの断片を組み合わせて解答のプログラムを作成させる方法が採点などの観点から現実的である。筆者らは、この立場に立って、演習問題や試験問題を作成する教員向け機能と、学習・受験できる学習者向け機能を持ち、ブラウザ上で動作するプログラミング環境を DNCL 用に作成した。この機能と構成、および実現方式について述べる。

Programming environment for DNCL with learning and examination functions

DAICHI OMIYA¹ TAKAHIRO MATUMOTO² TOSHIO MATUURA³ MICHIO NAKANISHI¹

1. まえがき

文部科学省は、高大連携の一環として大学入学者選抜改革推進委託事業（平成 28～30 年度）を実施しており [1]、この情報分野を大阪大学が受託し、東京大学、情報処理学会と連携して、「情報学的アプローチによる『情報科』大学入学者選抜における評価手法の研究開発」のテーマで研究を進めている [2]。この情報分野の研究では、Computer Based Testing (CBT) の可能性の検討も行われている。ここでの CBT は、出題者が試験問題としてプログラミングの作成課題を提示し、受験者は CBT を用いて試験中にプログラムを作成・実行することができるものである。本稿で述べる研究は、この CBT を用いた試験実施環境に組み込むことも視野に入れて、プログラミングの試験問題の作問支援機能と、プログラムの実行を含む解答機能を実現することをめざしている。

大学入試として問うべきプログラミングの内容・方法については様々な議論があり、まだその結論が出されていない

いが、筆者らはテキストベースでのプログラムの作成能力を問うことが適切と考えている。しかしながら、テキストエディタを用いて自由に文字列で記述された場合、その採点は困難を極める。これに対して情報処理学会の情報入試研究会が実施している大学入試模擬試験では、容易に採点できるようにするために、プログラムを構成する行を選択肢として用意し、並べ替えてプログラムを構成する問題を採用している [3]。

CBT のシステムでは、出題者の作問作業を支援することも重要である。本稿で述べるシステム（以下、本システム）では、出題者に対してプログラムを構成する部品を作成する機能を提供している。この部品には、プログラムコードの断片を入力する空欄やプルダウンメニューなどを含む。受験者は、試験問題を見ながらプログラムの空欄を埋めたり、選択肢欄にあるプログラムの構成要素を並べ替えてプログラムを作成でき、その場でプログラムを実行することができる。さらに、本システムは、入試に用いるだけでなく、プログラミングの演習に利用することもできる。

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能力を問うものとして使えるが、少し長いプロ

¹ 大阪工業大学 情報科学部 コンピュータ科学科
² オージス総研
³ 大阪市立大学

グラムを書こうとすると操作の手間が増えてしまう。また、ブロック型プログラミングは、実用のプログラミング言語とまったく異なるように見えてしまう短所がある。一方、大学等で行われてきた従来からのプログラミング教育のように、C言語やJavaScriptなどを用いてテキスト形式で自由にプログラムを書かせると、タイピングミスなどでエラーメッセージが英語で出力されるなどして、学習者の意欲を減退させる。また、プログラムを試験答案に書かせると、自動採点が難しいといった問題がある。そこで、ブロック型の長所を維持しつつ、自動採点も考慮し、ある程度の記述の制約を持たせたプログラミングの方法として、短冊型のプログラミングを採用した。

教員は、本システムを用いて、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、試験時に限らず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。本システムを試験に用いる場合は、教員は演習問題と同じように試験問題を設定し、試験受験者は学習時と同じようにプログラムを作成して解答する。なお、本システムでは、演習問題や試験問題を配布する機能や試験の答案を採点する機能を用意していない。この理由は、冒頭に述べたCBTのシステムに組み込む候補として考えているためである。

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疑似言語DNCLが用いられている^[4]。このDNCLの実行環境として最初に作られたのがPENである。その後、WaPEN、PenFlowchart、BlocklyPEN、どんくりなどが作られた。いずれも作問機能や解答機能はない。

2.1 PEN

PENは構文エラーの発生を低減するための入力支援機能や、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（ステップ実行、変数値表示）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。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機能を提供している^[5]、^[6]。

PENはJavaで記述されたアプリケーションプログラムであり、Windows、MacOS、LinuxなどのOS上で稼働するが、Web上では動作しないので、新たなアプリケーションプログラムの導入が制限されている学校現場で利用しにくい、個人のパソコンへインストールする敷居が高いという問題があった。

2.2 WaPEN、PenFlowchart

WaPENは、DNCLのWebブラウザ上での実行環境を提供している^[7]。PENと同様に入力支援やステップ実行機能があるが、変数表示機能はない。

また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。フローチャートは、プログラムの構造が図示されるので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けてしまうので、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つくれぬように配慮されている）。

2.3 どんくり

どんくりは、DNCLのWeb上での実行環境を提供している^[8]。どんくりは、プログラムの実行時の繰り返しの実行回数などのプロファイル情報を提示することで、アルゴリズムの良し悪しの検討が容易に行えるなどの工夫をしている。

どんくりでのプログラムの作成・編集作業は、プログラム編集領域で文字ベースで入力・編集を行う。これは、商用言語のプログラム作成過程と同じであり、学習者は次のステップとして商用言語の学習に移行しやすい。しかし、生徒が小中学校の段階でブロック言語だけを学んできたならば、テキストベースでのプログラミングは敷居が高いのではないと思われる。また、どんくりには、デバッグ機能は用意されていない。

2.4 BlocklyPEN

BlocklyPENは、WaPENをもとにしてその各命令をGoogle Blocklyを用いてビジュアルブロックで表わすように機能拡張したものである^[9]。ブロックにしたことにより、キーボードによるタイピングを極力減らすだけでなく、条件文や繰り返し文の構造をわかりやすく表示できる長所がある。しかし、ステップ実行機能や変数表示機能などのデバッグ用機能がない。

BlocklyPENはさらに改良が続けられており、関数を定義できるようになっている^[10]。

3. wPENの概要

本システムは、教員が使用するアプリケーションプログラムwPEN for Teachers (wPEN-Tと略す)と主として学習者が利用するwPEN for Students (wPEN-Sと略す)から構成されている^[11]。利用場面としては、プログラミングの試験のための利用と、プログラミング学習のための利用の2つの場面を想定している。図1に学習者の画面を示す。

試験の場面では、教員はwPEN-Tを用いて試験問題を作

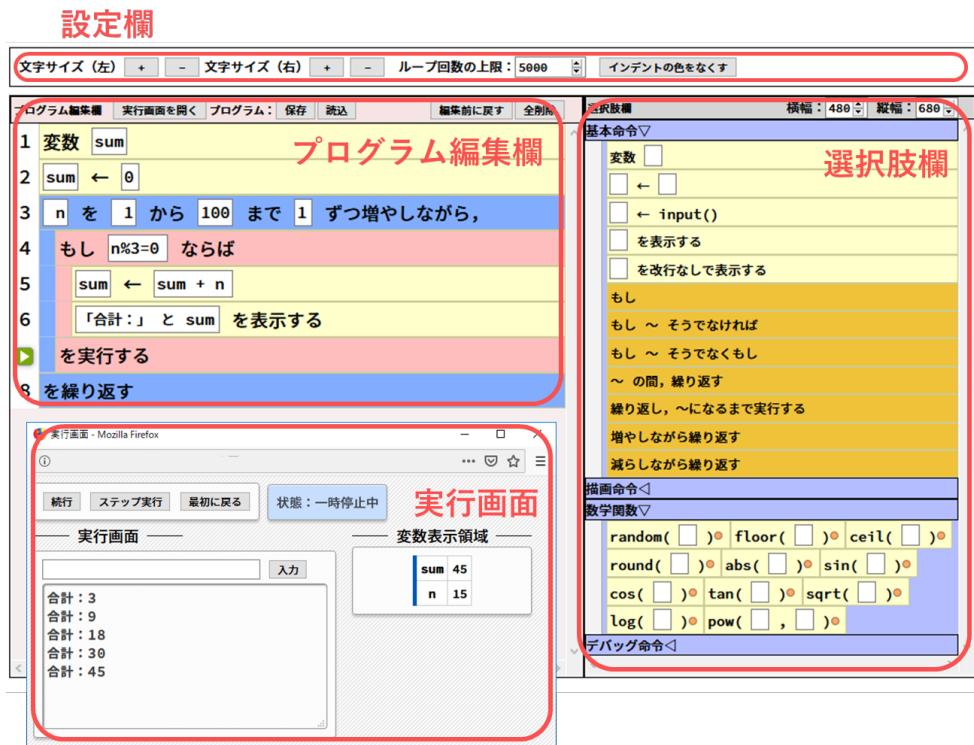


図 1 学習者用 wPEN-S の画面表示例

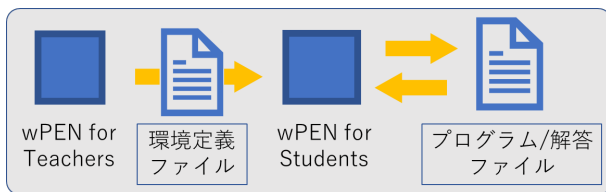


図 2 wPEN の構成

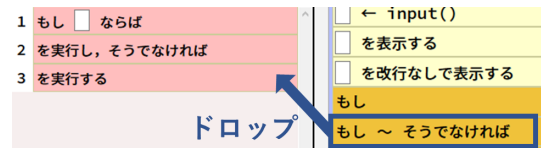


図 3 一度に複数の短冊を追加する短冊

成し、環境定義ファイルを生成する。受験者は、wPEN-S を用いて環境定義ファイルに作成された問題を開いて解答し、解答ファイルを生成する (図 2)。

学習の場面では、教員が wPEN-T を用いてプログラミング学習環境を構成し、それを環境定義ファイルとして生成する。学習者は、wPEN-S を用いて環境定義ファイルに用意された課題に対してプログラムの作成、実行、デバッグを行いながら学習する。学習者が作成したプログラムは、プログラムファイルとして保存され、再度読み込むことができる。

なお、本システムでは、DNCL 言語の学習環境をあらかじめ用意している。

以下、プログラミング学習時の利用、試験時の利用の順で説明する。

4. プログラミング学習時の利用

4.1 学習者用機能

4.1.1 基本操作

学習用として wPEN-S を使用する場合は、設定

欄、プログラム編集欄、選択肢欄から成る (図 1)。

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、短冊をプログラム編集欄の追加したい場所にドラッグ&ドロップすることでプログラムを作成する。また、プログラム編集欄内の短冊は、プログラム編集欄の外にドラッグ&ドロップすることで削除できる。

選択肢欄内のオレンジ色の短冊は、条件文と繰り返し文あり、プログラム編集欄にドロップすると、複数の短冊として展開される。例えば「もし～そうでなければ」の短冊では、図 3 のように 3 つの短冊に展開される。

random() などの数学関数は、代入文や条件式の中だけにドロップできる。このことをオレンジ色で丸形のアイコンを付けることで表現している。

「実行画面を開く」ボタンを押して実行画面を開き、プログラムを実行する。この詳細は後述する。

4.1.2 編集機能

短冊の、切り取り、コピー、貼り付けは右クリックで行える。複数の短冊を一括して扱うこともできる。また、「全削除」ボタンにより、全ての短冊を削除できる。

また、プログラムを大幅に変更した場合などで、プログ

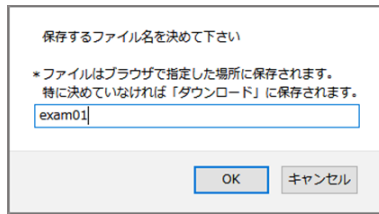


図 4 ファイル名を決めるダイアログ



図 5 短冊のインデントと色付け

ラムを作成する前の状態に戻したい場合は「編集前に戻す」ボタンをクリックすることで、環境定義ファイルを読み込んだ時の状態に戻すことができる。

4.1.3 保存/読み込み機能

「保存」、「読込」ボタンで、作成したプログラムの保存および読み込みができる。

「保存」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 4)。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保存される。

「読込」ボタンをクリックすると、ファイルダイアログが開く。そこでディレクトリを指定してプログラムファイルを選択することで、プログラムを読み込むことができる。

4.1.4 特徴的な機能

(1) 自動インデントと色分け

プログラムの構造を分かりやすくするために、インデントが繰り返し文によるものであれば青系色に、分岐文であれば赤系色に、短冊とインデントを色付けしている。同じ種類の制御構造がネストした場合には、インデントの色が最大 3 段階の濃さで表示される(図 5)。

(2) 選択枝のグループ化

選択枝欄内の三角マーク(▽)がついたタブ(以下、グループタブ)をクリックすることで、グループに所属する短冊の展開と格納を切り替えることができる(図 6)。

(3) 表示関係の変更

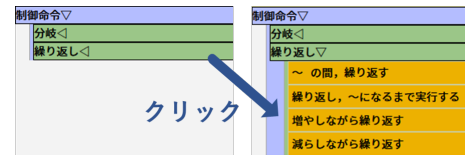


図 6 グループタブ

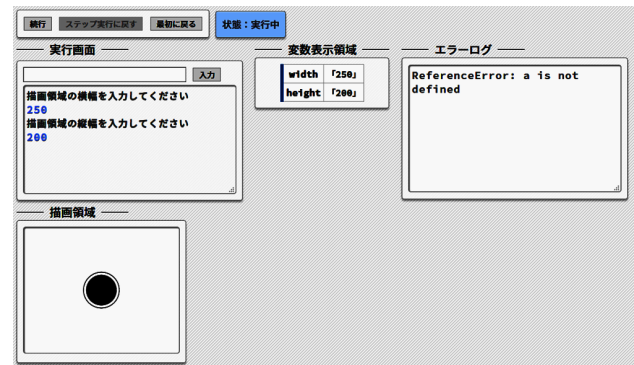


図 7 全ての領域が表示されたプログラム実行画面

選択枝欄上部にある、「横幅」、「縦幅」の値を変更すると、選択枝欄の大きさを変更できる。これにより、他人に説明する際に多くの短冊を一度に見えるようにできる。また、短冊の文字サイズおよび、インデントの色付けの有無の変更機能等がある。

4.1.5 プログラム実行画面

プログラミング編集欄の「実行画面を開く」ボタンを押すと、実行用のポップアップウィンドウが開く。「開始」ボタンでプログラムを連続実行できる。また、「ステップ実行」ボタンで一行ずつ実行でき、「開始」ボタンは「続行」ボタンに変化する。

プログラムからの出力は、実行画面内に表示される。プログラム中で `input()` 命令を実行すると、プログラムは入力終了するまで待機状態になり、学習者が入力した文字は青色で表示される。

変数表示領域には、プログラム中で使用されている変数名と現在の値が表示される。変数の値が変わったタイミングで、変数表示領域の値も変化する。

プログラム実行画面の右上には、プログラムの状態表示欄がある。プログラムの状態は、開始待ち、実行中、入力待ち、一時停止中、終了が存在する。

- 開始待ち : プログラム実行画面が開かれた直後の状態
- 実行中 : プログラムが実行中の状態
- 入力待ち : 入力が確定するまで待機している状態
- 一時停止中 : プログラムを中断して、次に続行もしくはステップ実行が押されるのを待機している状態
- 終了 : プログラムの実行が終了した状態

「最初に戻る」ボタンをクリックすると、プログラムが開始待ち状態になる。

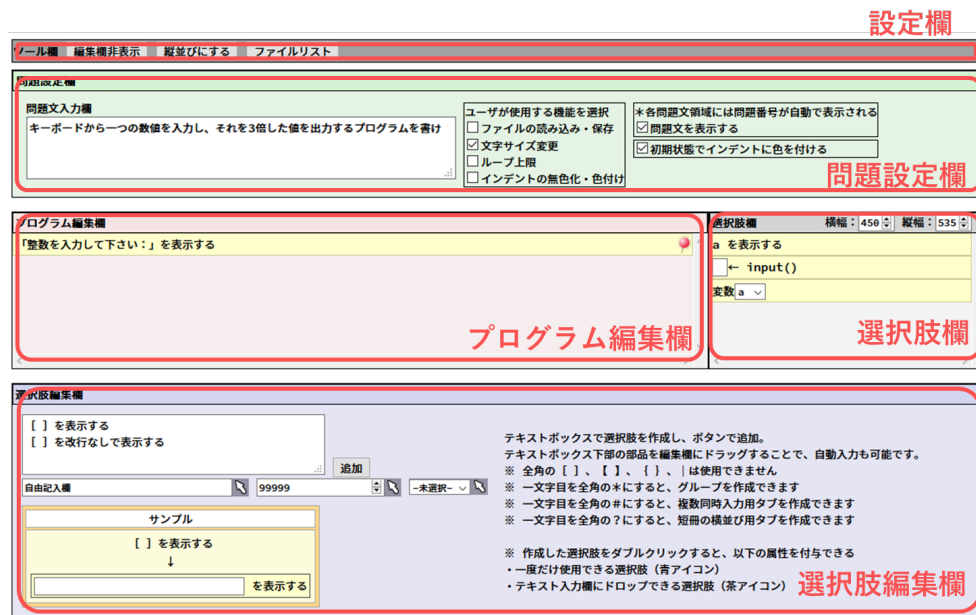


図 8 教員用 wPEN-T の画面表示例

プログラムを開始すると、次に実行する短冊の行番号の位置に緑色の三角のアイコンが表示される（図 1 では 7 行目）。

プログラム実行画面には、前述以外に、次の 2 つの領域がある（図 7）。描画領域は、プログラム中で描画領域を開く命令が実行された場合のみ表示され、エラーログは、プログラム中でエラーが発生した場合にその内容を表示する領域である。

本システムでは、無限ループを含むプログラムを実行した場合への対策として、一定回数以上繰り返しの命令が実行された場合に自動停止する機能を搭載している。繰り返しの回数の上限は、設定欄にある「ループ回数の上限」で変更できる。

4.2 教員用機能

4.2.1 基本操作

初学者には演習問題を与えるだけでなく、見せる短冊の種類を減らして、学びやすくさせるなどの配慮も必要であろう。本システムでは、教員が wPEN-T を用いて初学者のための環境を設定できる。

wPEN-T の画面は、設定欄、問題設定欄、プログラム編集欄、選択肢欄、選択肢編集欄の 5 つの欄からなる（図 8）。

選択肢編集欄では、学習者に提示する選択肢（短冊）を入力し、追加ボタンを押すことで、短冊を作成することができる。選択肢編集欄にある「自由記入欄」と書かれたテキストボックスをテキストエリアにドラッグ&ドロップすると、図 1 の選択肢欄の「変数 □」のような空欄のある短冊を設けることができる。

4.2.2 学習環境の設定

問題設定欄にあるチェックボックスで、ループ回数の上限の変更機能やインデントの色付けの有無の変更機能など、学習の際に学習者に提供する機能を選択できる。また、問題文入力欄に入力した内容が、問題文として表示される。なお、問題文を表示しない設定もできるようにしており、自由課題をプログラミングする環境としても使うことができる。

4.2.3 保存/読み込み機能

設定欄の「ファイルリスト」ボタンをクリックすることで、作成した学習環境の、保存、読み出し、出力ができる。

4.2.4 タブの設定

選択肢編集欄にあるテキストエリアの各行の先頭の文字を特定の文字にすると、特別な機能を持つタブを作成することができる。

- (1) 先頭の文字が全角の*の場合：グループ作成用のタブ（グループタブ）を作成できる。グループタブに短冊を追加することで、その短冊をグループに含むことができる。短冊をグループに含むことで、短冊の種類分けができる。
- (2) 先頭の文字が全角の#の場合：複数の短冊に展開されるタブ（マルチタブ）を作成できる（例えば図 3 の「もし～そうでなければ」）。マルチタブは wPEN-S でオレンジ色の短冊となる。繰り返しや分岐など、複数の短冊を組み合わせる作成処理を、1 つの短冊にまとめることができる。
- (3) 先頭の文字が全角の?の場合：図 1 の数学関数群のように短冊の横並び用のタブ（インライントブ）を作成できる。インライントブに含まれた短冊は、wPEN-S

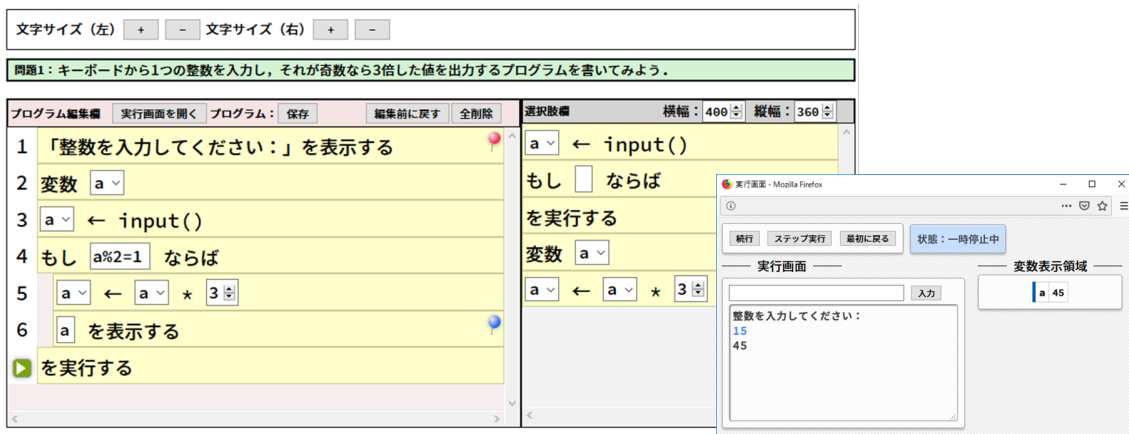


図 9 試験用として使用する際の wPEN-S の画面表示例

では横並びで表示される。内容の似た短冊を横並びにすることで、見栄えをよくできる。

4.2.5 短冊の設定

演習問題を作成するときは、問題文の入力のほかに次の設定をできるようにしている。

(1) 数値入力欄、プルダウンメニュー欄の設定

短冊を作成する際、テキストエリアの下にあるスピンボックスをドラッグ&ドロップすると、短冊中に数値のみ入力することができる空欄を設けることができる。テキストエリアの下にあるプルダウンメニューをドラッグ&ドロップすると、短冊中にプルダウンメニューを追加できる。

(2) 短冊ごとの使用回数制限

選択肢欄の短冊をダブルクリックして、青いピン型のアイコンが付いた短冊にすると、一度しか使用できない短冊を作成できる。通常の短冊と同じく、選択肢欄からプログラム編集欄に追加できるが、一度追加すると選択肢編集欄に表示されなくなるので、この短冊を2回使用することはできない。

(3) 使用すべき短冊の指定

選択肢欄の短冊をプログラム編集欄に移動させると、赤いピン型のアイコンが付き、必ず使用しなければならない短冊にすることができる。これにより、同じ命令をいくつも並べるのではなく、繰り返しや条件文を使うようなプログラムを作らせることができる。

(4) 代入文や条件文の中にだけドロップできる短冊

選択肢欄内の短冊をダブルクリックすると、短冊にオレンジ色の小さな丸いアイコンが付与される。この短冊は、プログラム編集欄には直接ドロップできないが、代入文や条件文の中にだけドロップすることができる。

5. 試験時の利用

試験時のための教員用機能は 4.2 節と同様なのでここでは省略する。学習者用機能は 4.1 節で説明した機能と同様

である。学習者は、wPEN-S で「読込」ボタンから、試験用に作成された環境定義ファイルを選択することで、受験を開始する(図 9)。自分のプログラムを作成したら、「保存」ボタンから解答ファイルを生成する。

6. wPEN の実装

6.1 プログラム実行環境の構成

6.1.1 DNCL から JavaScript への変換

本システムは、JavaScript を利用して構築している。短冊を組み合わせで作成された DNCL プログラムを JavaScript プログラムに変換して、プログラムの実行を行っている。

短冊中に特定のパターンの文字列が存在する場合、あらかじめ決められた JavaScript のソースコードに書き換えられる。パターンの検出には正規表現を用いている。DNCL のプログラムの一行が、JavaScript の一行に変換されるには限らない。

プログラムの変換例を図 10 と図 11 に示す。図 10 の 1 行目の内容は図 11 の 1 から 4 行目に、2 行目は 5 から 7 行目に、8 から 11 行目に変換される。

まず図 11 の、1 行目の `sendLineNumber()` 関数は、これから実行しようとしている行番号を渡して、画面上にその場所を(△印で)示すための関数である。2 行目では、「ステップ実行モード (`stepFlag=true`)」かどうかを調べて、これが真の場合は、`sendStatus()` 関数で、状態表示を”一時停止中”に設定し、そこで、いったん実行を中断する (`yield`)。この間、別のスレッド(後述)で、”続行”ボタンもしくは”ステップ実行”ボタンが押されるのを待ち、このいずれかが押されたとき、`yield` で中断していた処理を再開する。再開後は、状態表示を”実行中”に設定する。3 行目では、

-
- 1 $n < 10$ の間、
 - 2 $n \leftarrow n + 1$
 - 3 を繰り返す
-

図 10 変換前のプログラム (DNCL)

```
1 sendLineNumber(0);
2 if (stepFlag){ sendStatus('状態 : 一時停止');yield;sendStatus('状態 : 実行中');}
3 let timeOutCnt1000 = 0;
4 while((sendLineNumber(1) && checkStep() && (yield ) && afterStepWait()) || n < 10 ){
5   sendLineNumber(2);
6   if (stepFlag){ sendStatus('状態 : 一時停止中');yield;sendStatus('状態 : 実行中');}
7   n = n + 1 ; displayVariable('n',n);
8   sendLineNumber(3);
9   if (stepFlag){ sendStatus('状態 : 一時停止中');yield;sendStatus('状態 : 実行中');}
10  if(++timeOutCnt1000 > 5000) { self.postMessage({'message':'無限ループ','type':1}); sendStatus('状態 : 終了'); return;}
11 }
```

図 11 変換後の JavaScript プログラム

timeOutCnt1000 を 0 に設定している。これは、無限ループの対策として、繰り返し文の実行回数を数えるための変数であり、変数名の末尾にある数値が 1000 から 1001 のように変化することで、繰り返し文ごとにそれぞれ異なる変数名が割り振られる。

4 行目が、本来変換されるべき繰り返し文であるが、元の繰り返しの条件 ($n < 10$) の前に、これから実行しようとしている行番号の設定と、実行状態の設定、ステップ実行であれば「続行」ボタンもしくは「ステップ実行」ボタンが押されるまで待機する処理を行う。checkStep() 関数は、「ステップ実行モード」かどうかを判定し、これが真の場合は、状態表示を「一時停止中」に設定し、true を返す。afterStepWait() 関数は、状態表示を「実行中」に設定し、必ず false を返す。ここまでは、図 10 の 1 行目の変換結果である。

5 行目からは、まず行番号の設定をしている。6 行目では、「ステップ実行モード」かどうかを調べて、これが真の場合は、一旦実行を中断する。7 行目では、図 10 の 2 行目の代入処理を行い、その後、displayVariable() 関数で変数表示領域の変数 n の値を変更している。これが、図 10 の 2 行目の変換結果である。

8 行目からは、図 10 の 3 行目に対応する。10 行目では、繰り返しの実行回数を数える変数の値を一つ増加させ、その値がループ回数の上限に達していれば、無限ループであることを表すポップアップを表示し、状態表示を「終了」して、プログラムを強制的に終了する。postMessage() 関数は引数に取った文字列の内容を含んだ、ポップアップを表示する。11 行目は、繰り返し文を閉じている。

6.1.2 実行画面の構成

wPEN は 2 つのスレッドで動作するように構成している (図 12)。図 1 のプログラム編集欄の「実効画面を開く」をクリックすると実行画面が表示され、作成した JavaScript プログラムの実行用のスレッド (以降、実行スレッド) が作成される。作成した実行スレッドとメインスレッドは、必要な時のみメッセージを送りあうことで、画面表示の変更や、入力された値の受け渡しを行う。

メインスレッドは「開始/続行」ボタンまたは「ステップ実行」ボタンがクリックされるか、テキストボックスに入力されると、実行スレッドへメッセージを送信する。

実行スレッドはメッセージ処理部と、実行部の 2 つに分かれており、メッセージ処理部はメッセージを受信すると、その内容を解析し、解析したメッセージの内容によって異なる動作をする。

(1) 「開始/続行」ボタンが押されたとき

「連続実行モード」に変更してから実行部に処理の開始/再開を要求する

(2) 「ステップ実行」ボタンが押されたとき

「ステップ実行モード」に変更してから実行部に処理の開始/再開を要求する

(3) テキストボックスに入力されたとき

入力された値を実行部に受け渡しつつ、実行部に処理の開始/再開を要求する

実行部は開始/再開の要求を受け取ると、プログラムを次の yield がある部分まで実行し、待機する。このとき、実行画面への出力や、状態表示欄の表示の変更があると、メインスレッドに表示を変更することを要求するメッセージを送信する。

メインスレッドが実行部からメッセージを受け取ると、メッセージの内容を解析し、メッセージの内容に従って、画面の各領域を書き換える。

「最初に戻る」ボタンを押すと、実行スレッドが破棄され、現在のプログラムの内容で新たに実行スレッドが作成される。

実行画面を閉じると、実行スレッドが破棄される。

6.2 環境定義ファイルと解答/プログラムファイルの構造

本システムの環境定義ファイルと、解答/プログラムファイルは同じ構造をしている。図 8 の画面表示から生成される環境定義ファイルを図 13 に示す。

config タグには問題設定欄の情報が、question タグには一つの問題が記されている。question タグを複数記述する

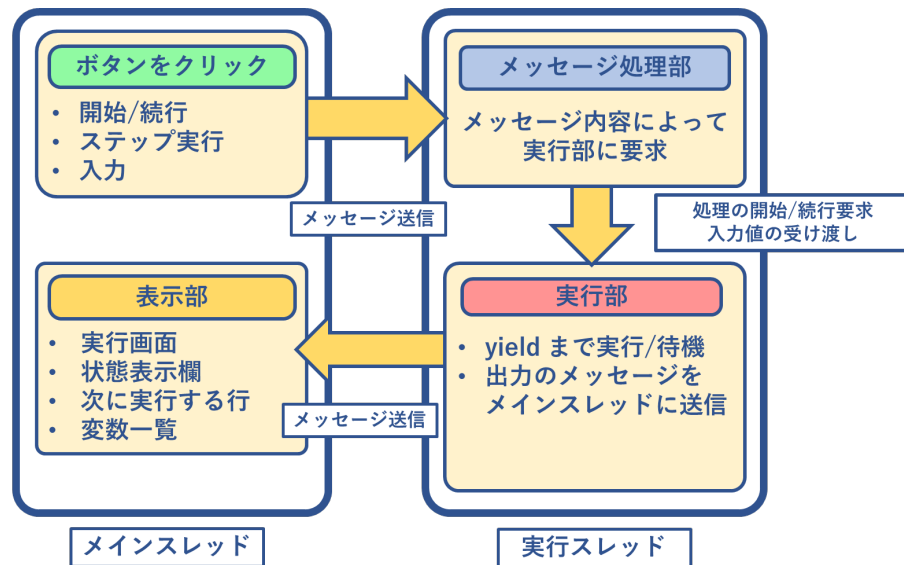


図 12 実行画面の構成

```

1 <?xml version='1.0' encoding='UTF-8'?>
2 <doc>
3   <config colorFlag="false"> .....設定欄の内容
4     <charSize></charSize>
5   </config>
6   <question horizontal='true'>
7     <textArea visible='true'> .....問題文の内容
8       <text>キーボードから一つの整数を入力し、それを 3倍した値を出力するプログラムを書け</text>
9     </textArea>
10    <answerArea height='300'> .....プログラム編集欄の内容
11      <item fixed='true'>「整数を入力してください:」を表示する</item>
12    </answerArea>
13    <itemsArea width='500' height='300'> .....選択肢欄の内容
14      <item unique='true'>a を表示する</item>
15      <item>{text::text} ← input()</item>
16      <item>変数 {pullDown:a,b:pullDown} </item>
17    </itemsArea>
18  </question>
19 </doc>
    
```

図 13 環境定義ファイルの一例

ことにより、複数の問題を持つ環境定義ファイルを作成することができる（現状では一つの問題のみ対応）。

question タグは horizontal 属性を持ち、この値が true の場合はプログラム編集欄と選択肢欄を横並びに表示し、false の場合は縦並びに表示する。textArea 属性は visible 属性を持ち、値が true の場合は問題文を表示し、false の場合は問題文を表示しない（4.2.2 参照）。

answerArea タグと itemsArea タグはそれぞれのサイズを記した属性を持つ。width 属性と height 属性はそれぞれの横幅と縦幅のピクセル数である。answerArea の横幅は itemsArea の width 値によって決まるため、answerArea は width 属性を持たない。

item タグは 1 つの短冊の情報が記されており、短冊の種

類によって 3 種類の属性を持つ。

- fixed 属性：必ず使用しなくてはならない短冊を表す属性。
- unique 属性：一度しか使用できない短冊を表す属性。
- droppable 属性：ほかの短冊の空欄にドロップすることができる短冊を表す属性。

短冊内の空欄、数値入力欄、および選択欄は、{ 種類:値:種類 } の形で表す。種類は、text、number、pullDown の 3 種類があり、text は空欄、number は数値入力欄、pullDown は選択欄を表す。値は text、number の場合は初期値を表し、pullDown の場合は選択肢を表す。なお選択肢は「,」で区切ることで、別の項目であることを表す。{ 種類:値:種類 } の 3 つ目は 1 つ目と同じものを指定しなくてはなら

ない。例えば、図 13 の 15 行目の {text::text} は、「□ ← input()」の左辺の空欄を示す。

7. あとがき

本システムを某高校の情報の先生に評価していただいたところ、以下のような指摘をいただいた。

- 短冊をプログラム編集欄に追加する際にどこに追加されるのかがわかりにくい。移動先の表示を変えるなど、わかりやすくしてほしい。
- 見た目 (look&feel) が古臭い
- Undo/Redo 機能が必要
- 問題作成機能には説明書が必要
- 大問、中間を作れるように、問題のグループ化機能が欲しい

さらに多くの先生方に使用してもらって評価する必要がある。

現在、wPEN-S では、DNCL で記述されたプログラムのみが実行可能である。wPEN 自体が JavaScript で記述されているので、対象とする言語を DNCL の代わりに JavaScript にしたい場合は、対応は容易である。それ以外の言語に対しても、wPEN-T を利用することで、言語を構成する短冊は容易に作成できる。対象とする言語の処理系 (JavaScript への変換) は、JavaScript でコーディングしなければならないが、6.2 節で示したような形で作成できる。

謝辞 本研究の一部は、JSPS 科研費 JP17K01088 の助成を受けている。また、冒頭で述べた大阪大学、東京大学、情報処理学会の研究グループから多くの示唆と助言を頂いたことに感謝する。

参考文献

- [1] 高等教育局大学振興課大学入試室：大学入学者選抜改革推進委託事業, 文部科学省, http://www.mext.go.jp/a_menu/koutou/senbatsu/1397824.htm (2019.1.21 参照).
- [2] 大阪大学大学院情報科学研究科：情報学的アプローチによる「情報科」大学入学者選抜における評価手法の研究開発, <http://www.uarp.ist.osaka-u.ac.jp> (2019.1.21 参照).
- [3] 情報処理学会情報入試研究会：<http://jnsj.jp/?page.id=49> (2019.1.21 参照).
- [4] 大学入試センター：センター試験手順記述標準言語 (DNCL) の説明, <https://www.dnc.ac.jp/albums/abm.php?f=abm00004841.pdf&n=H23.dncl.pdf> (2019.1.21 参照).
- [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, 情処論, Vol.48, No.8, pp.2736-2747 (2007.8).
- [6] 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).
- [7] 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).
- [8] 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).
- [9] 岩崎みのり, 中西通雄：ビジュアルブロック化した DNCL の Web ブラウザ上での実行環境, 教育システム情報学会学生研究発表会, pp.163-264 (2018.2), <http://www.jsise.org/society/presentation/> (2019.1.21 参照).
- [10] 岡本勇紀, 中西通雄：ブラウザ上で動作する DNCL 学習環境 WaPEN と BlocklyPEN の改良, 教育システム情報学会学生研究発表会 (2019.2 発表予定), <http://www.jsise.org/society/presentation/>
- [11] 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).