

試験問題記述言語 DNCL の改定に合わせた 「どんくり」システムの修正と検討

本多 佑希¹ 漆原 宏丞¹ 兼宗 進¹

概要：2025 年度入試からの実施を予定している大学入学共通テストの情報科の試作問題が公開された。センター試験や大学入学共通テストでは独自の仮想言語である「DNCL」が使用されてきたが、試作問題では DNCL 言語の拡張が行われている。拡張に関する言語仕様の説明文書は公開されていないため、本発表では試作問題から言語の拡張を読み取るとともに、DNCL をオンラインで記述して実行できる学習環境「どんくり」に反映することで、DNCL 言語全体の動作を検証し、拡張部分を含めた今後の方向性について検討したい。

1. はじめに

センター試験では、情報関係基礎で問題の題材となるアルゴリズムを受験者に示すために、センター試験手順記述標準言語 (DNCL) という仮想言語が用いられてきた [1] [2]。DNCL は言語仕様や構文がシンプルであるため可読性が高く、処理の流れを理解しやすい利点がある。大学入試センターから公開された、2025 年度入試から実施を予定している大学入学共通テストの情報科の試作問題 [3] では、この DNCL の構文が変更されており、より現代的なプログラミング言語に近い構文となった。本稿では、この試作問題で扱われている DNCL を「DNCL2」、従来の DNCL を単に「DNCL」として記すことにする。

著者らは、DNCL を Web ブラウザ上で実際にプログラミング言語として記述し、実行できる環境「どんくり」の開発を進めている [4-6]。どんくりは DNCL に関数定義や配列定義などの構文を拡張している他、アルゴリズム学習への適用を意識した機能として、定義されている変数を一覧できる機能や、反復や分岐、関数実行が何回行われたかを記録し、実行者にフィードバックする機能を追加している。

今回は、このどんくりに「DNCL2」の構文を追加するとともに、アルゴリズム学習に向けて仕様の拡張を行った。DNCL2 については言語仕様書などが公開されていないため、試作問題から仕様を分析することにした。また、DNCL と DNCL2 を相互に変換することを可能にした。この機能により、センター試験で出題されたプログラムを DNCL2 で確認できる。

本稿では試作問題から読み取った DNCL2 の言語仕様を紹介するとともに、どんくりでの対応とアルゴリズム学習に向けた機能拡張について議論する。

2. DNCL と DNCL2 の比較

2.1 DNCL の概要

DNCL はセンター試験の出題問題のプログラムを記述するために考案された仮想言語である。言語仕様書は Web で公開されている [1] が、センター試験の問題冊子に言語仕様の説明はない。DNCL は構文がシンプルで明快なため、C などのプログラミングを理解している生徒はプログラムを読めることを前提にしている。

2.2 DNCL2 の概要

2025 年度入試から実施を予定している大学入学共通テストの情報科の試作問題 [3] では、DNCL の構文が変更されている。本稿ではこの試作問題で扱われている DNCL を DNCL2 とする。DNCL2 は DNCL よりプログラミング言語の構文に近づいているが、パラダイムとしては DNCL とほぼ変わりなく、シンプルな手続き型言語である。DNCL2 の言語仕様書は公開されていないため、本稿では試作問題から読み解れる範囲の仕様の変更を確認する。試作問題で扱われているプログラムの例を図 1 に示す。

2.3 数値や文字列の扱い

数値に差異は無いが、文字列のリテラルに変更が加えられている。DNCL では文字列を鉤括弧(「」)で表現するが、DNCL2 ではダブルクォート(“”) で表現する。

¹ 大阪電気通信大学
Osaka Electro-Communication University

```

1 Angoubun = ["p","y","e","b",...(省略).. "k","b","d",
  "r","."]
2 配列変数 Hirabun を初期化する
3 hukugousuu = 26 - 10
4 i を 0 から 要素数 (Angoubun)-1 まで 1 ずつ増やしな
  ながら:
5 | bangou = 差分 ( Angoubun[i] )
6 | もし bangou != -1 ならば:
7 | | もし bangou+hukugousuu <= 25 ならば:
8 | | | Hirabun[i] = 文字 ( bangou+hukugousuu )
9 | | そうでなければ:
10 | | | Hirabun[i] = 文字 ( angou+hukugousuu-26 )
11 | | そうでなければ:
12 | | | Hirabun[i] = Angoubun[i]
13 表示する (Hirabun)
  
```

図 1 試作問題のプログラム例

```

(DNCL)
10
「はじめまして」
  
```

```

(DNCL2)
10
"はじめまして"
  
```

2.4 変数の扱い

変数の扱いに差異はなく、宣言時に型の指定が不要で動的に型が決まる。そのため、同一の変数に整数/浮動小数点数/文字列などを代入できる。DNCL では代入の記号には「」が使われていたが、DNCL2 では「=」が使われるようになった。

```

(DNCL) i 3, s 「こんにちは」
(DNCL2) i=3, s="こんにちは"
  
```

複数の値を扱うデータ型として配列が用意されている。DNCL では配列のリテラルは存在せず、以下の 3 パターンで配列が用意されていた。

- インデックス (添字) を指定して 1 つずつ代入する
- 初期値を指定する構文で初期化する
- 本文中で「～という配列が存在する」と説明する

しかし、DNCL2 では配列のリテラルが用意されており、試作問題中では次のように配列が宣言されていた。

```

Angoubun = ["p","y","e","b",...(略)...
"k","b","d","r","."]
  
```

DNCL では配列の添字の初期値は 1 であったが、DNCL2 では 0 から始まり、2 次元以上の配列を参照する場合の構文には変更が加えられている。また、どちらも生成された

```

(DNCL)
i を 1 から 30 まで 1 ずつ増やしなが
...
を繰り返す

もし i>j ならば
...
...
を実行する

(DNCL2)
i を 1 から 30 まで 1 ずつ増やしなが
...

もし i>j ならば:
...
...
  
```

図 2 反復と分岐の構文の比較

要素の初期値を指定する構文が用意されている。

```

(DNCL)
a[0,1] 15
Ary のすべての値を 0 にする
  
```

```

(DNCL2)
a[0][1]=15
配列 Ary のすべての要素に 0 を代入する
  
```

DNCL では構文として定義されていた表示文は、DNCL2 では関数として用意されている。

```

(DNCL) i を表示する
(DNCL2) 表示する (i)
  
```

制御構造は DNCL2 で構文が変更されているだけでなく、インデントを罫線で示す仕様が加えられている。ブロックを「」で示しながら、最後の行は「」とする仕様のようである。制御構造の構文を図 2 に示す。

2.5 比較のまとめ

本章で DNCL と DNCL2 について比較を行った。構文は DNCL からよりプログラミング言語に近づいたように感じる。高等学校の情報の教員研修資料でも採用されている Python に近い。なお、DNCL の構文を独自で拡張し、Python の文法に近づけた PyPEN という環境が存在する [9]。

また、配列の添字が 0 から始まるように変更されたことがわかった。配列の添字を変数 i として各要素を参照するようなプログラムの場合には、DNCL では「1 から要素数

```
(DNCL)
// Ary が定義されているものとする
i を 1 から要素数(Ary)まで 1ずつ増やしなが、
Ary[i]を表示する
を繰り返す

(DNCL2)
Ary = ["A","r","r","a","y"]
i を 1 から要素数(Ary)まで 1ずつ増やしなが、
表示する (Ary)
```

図 3 配列要素を反復で参照するプログラム例

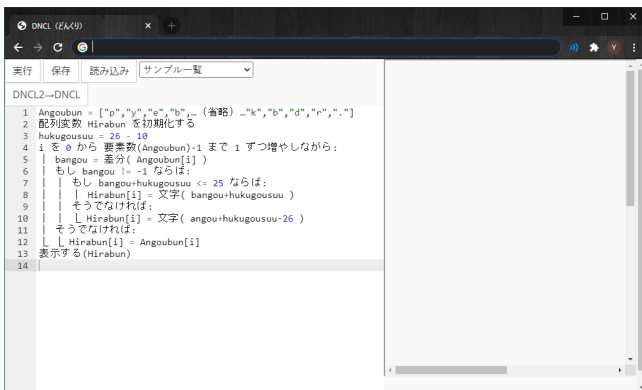


図 4 IDE の画面

まで繰り返す」のに対し、DNCL2では「0から要素数-1まで繰り返す」ことになる。具体的なプログラムを図3に示す。この点からも、DNCL2では現代的なプログラミング言語に近づけることを優先したと読み取れる。

このように、構文を含めて細かい変更はあったが、言語仕様に本質的な大きな変更はなかった。シンプルな手続き型言語である仕様は維持されているため、DNCL2においてもプログラムの理解は容易であることが期待できる。

3. どんくりでの対応

3.1 概要

著者らは、DNCLをWebブラウザ上で実際にプログラミング言語として記述し、実行できる環境「どんくり」の開発を進めている[4-6]。DNCLをプログラミング言語として記述に用いることが学習に効果があることは、PENおよびWaPENで明らかにされている[7,8]。2章で試作問題から読み取った言語仕様をもとに、筆者らが開発を進めているDNCL実行環境「どんくり」上でDNCL2を実行できるように機能拡張を行った。プログラムを記述・実行ができるIDEの画面を図4に示す。

3.2 処理系

どんくりはWebブラウザでDNCLを実行するために、DNCLをJavaScriptに変換(トランスパイル)して実行する。DNCL2でも、同様にJavaScriptに変換することにし

```
//Rieki は本文中で与えられたものである
Rieki {1,-3,2,10,-4,12,8,-4}
saidaiRieki 0

i を 1 から 8 まで 1 ずつ増やしなが、
soRieki 0
j を i から 8 まで 1 ずつ増やしなが、
soRieki soRieki+Rieki[j]
もしsoRieki>saidaiRiekiならば
saidaiRieki soRieki
kaisi i, syuryo j
を実行する
を繰り返す
を繰り返す

「開始時間帯は」とkaisiと「とし、」を表示する
「終了時間帯は」とsyuryoと「とする。」を表示する
「総利益の最大値は」と
saidaiRiekiと「千円である。」を表示する
```

図 5 2013 年出題のプログラム例

た。変換の際には、JavaScriptで開発された構文解析器を使用して解析し、DNCL2のプログラムに対応する構文木を生成、その後に対応するJavaScriptのプログラムを組み立てる。全てがJavaScriptで行われるため、サーバとの通信を不要とし、実行がクライアントの端末内で完結する。学習に使用するツールであるため、教室内で頻繁に通信が行われない点は、通信負荷をかけないという点において利点であると考えている。

3.3 DNCLとDNCL2の相互変換

どんくりでは、DNCLとDNCL2の構文を相互に変換する機能を実装した。プログラムを記述したのち、「DNCL DNCL2」や「DNCL2 DNCL」のボタンを押すことで変換が行える。このように、DNCLで記述された過去の出題問題のプログラムをDNCL2の構文で確認できることは利点がある。

2013年に出题されたプログラムの例を図5に示す。このプログラムを、本機能を使ってDNCL2の構文に変換すると図6が出力される。

4. DNCLの拡張

4.1 ライブラリ関数の整備

「sin」や「cos」_↓「切り捨て」や「四捨五入」などの数学関数や、「入れる」や「取り出す」_↓「入れ替える」などの配列操作関数をライブラリとして提供している。標準で用意している関数の一覧を表1に示す。「確認」は、定義されている変数を一覧で見ることが出来る関数である。この関数をプログラムの随所で実行することで、変数の中身がどのように変化するかを確認することができる。

```

1 Rieki=[1,-3,2,10,-4,12,8,-4]
2 saidaiRieki=0
3
4 i を 1 から 8 まで 1 ずつ増やしなが
5 | soRieki=0
6 | j を i から 8 まで 1 ずつ増やしなが
7 | | soRieki=soRieki+Rieki[j]
8 | | もしsoRieki>saidaiRieki ならば:
9 | | | saidaiRieki=soRieki
10 | | | kaisi=i , syuryo=j
11
12 表示する ("開始時間帯は", kaisi, "とし,")
13 表示する ("終了時間帯は", syuryo, "とする。")
14 表示する ("総利益の最大値は", saidaiRieki, "千円であ
    る。")

```

図 6 図 5 を DNCL2 に変換したもの

表 1 どんぐりで用意したライブラリ関数

日本語名	英語名	日本語名	英語名
	sin	交換	swap
	cos	挿入	insert
	tan	削除	delete
	asin	要素数	length
	acos	取り出す	extract
	atan	追加	push
	atan2	空	empty
	log	空でない	not_empty
	log2	連結	concat
	log10	確認	dump
乱数	random	性能を確認する	profile
平方根	sqrt	文字	charAt
切り捨て	floor		
切り上げ	ceil		
四捨五入	round		
絶対値	abs		
指数	exp		

4.2 プロファイル関数

アルゴリズムの学習などに使うために、実行時の統計情報（プロファイル情報）を表示する関数を用意した。プロファイル情報は、次の情報を自動で収集している。プログラム全体の情報や、ある関数を実行した際の情報を実行者にフィードバックする。

- 実行時間 (msec)
- 繰り返しの実行回数
- 条件分岐の比較回数/真の回数/偽の回数
- 各関数の呼び出し回数

4.3 関数定義構文

DNCL2 をプログラミング言語として記述に用いるにあたって、関数を定義する構文を拡張した。引数を受け取り、戻り値を設定することもできる。定義例を図 7 に示す。

```

1 あいさつ ( ):
2   表示する ("こんにちは")
3
4 二倍( n ):
5   | 結果 = n x 2
6   | 結果 を返す

```

図 7 関数を定義する例

4.4 配列リテラルの拡張

ソーティングアルゴリズムやサーチアルゴリズムを扱う際には、それらのアルゴリズムを適用する配列を作成する必要がある。1 から 100 までの昇順の値を生成する例、2 から 100 までの偶数の値を昇順で生成する例、100 から 1 までの降順の値を生成する例、1 から 100 までのランダムな値を 100 個生成する例を図 8 に示す。

本環境では、これらの配列生成を単純化するために配列リテラルの拡張を行った。1 から 100 までの昇順の値を生成する場合、次のように記述する。

```
Ary1=[1,2,3,...100]
```

2 から 100 までの偶数の値を昇順で生成する場合、次のように記述する。

```
Ary1=[2,4,6,...100]
```

100 から 1 までの降順の値を生成する場合、次のように記述する。

```
Ary1=[100,99,98,...100]
```

1 から 100 までのランダムな値を 100 個生成する場合、次のように記述する。

```
Ary1=[乱数(100),...100]
```

この構文によって、アルゴリズムの学習に入る前に、それを適用する配列を生成する学習の手間を省くことができる。そのため、本来学習すべきアルゴリズムの処理手順によりフォーカスを当てられると考えている。

5. おわりに

大学入試センターから公開された、2025 年度入試からの実施を予定している大学入学共通テストの情報科の試作問題から、DNCL2 の仕様を読み解きながら、DNCL との比較を行った。構文には変更が加えられている部分が多いが、言語のパラダイムなど、仕様に関する大きな変更はなかった。

```

1 Ary1=[]
2 i を 1 から 100 まで 1 ずつ増やしなが
3   追加 (Ary1, i)
4
5 Ary2=[]
6 i を 2 から 100 まで 2 ずつ増やしなが
7   追加 (Ary1, i)
8
9 Ary3=[]
10 i を 100 から 1 まで 1 ずつ減らしなが
11   追加 (Ary1, i)
12
13 Ary3=[]
14 i を 1 から 100 まで 1 ずつ増やしなが
15   追加 (Ary1, 乱数 (100))
    
```

図 8 昇順、降順、ランダムでそれぞれ配列を生成する例

そして調査した結果をもとに、DNCL 実行環境「どんくり」で、DNCL2 の構文に対応した。実際にプログラムを Web ブラウザ上で実行できるようになったと共に、DNCL と DNCL2 を相互に変換する機能を実装した。これにより、DNCL で記述されたセンター試験の過去問題のプログラムを DNCL2 で確認することができる。DNCL2 を実際にプログラミング言語として記述するにあたって、関数定義の拡張を行ったとともに、アルゴリズム学習に向けた機能をいくつか開発した。どんくりで以前から対応していた「宣言されている変数とその値を一覧する機能」や、「プログラム中で使われている反復や分岐の回数などを確認できる機能」を実装した。また、アルゴリズムを適用する対象を簡単に生成するために、配列のリテラルを拡張した。これにより、アルゴリズム学習により活用できるようになったと思われる。

PEN や WaPEN が DNCL をプログラミング言語として記述に用いる学習効果は示されているが、今回独自に拡張した機能については、まだ授業で学習効果を確認できていない。そのため、今後は授業実践などを検討していきたい。

参考文献

[1] 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=abm00004841.pdf&n=H23_dnc1.pdf (2021/2/17 参照).

[2] 兼宗進, 本多佑希: 高等学校「情報 I」の研修資料におけるプログラミングの言語の扱い. 情報処理学会, 情報教育シンポジウム (SSS2019), pp.101-105, 2019.

[3] 大学入試センター: 「情報」試作問題 (検討用イメージ).
<https://www.ipsj.or.jp/education/9faeag0000012a50-att/sanko2.pdf> (2021/2/17 参照).

[4] 兼宗進, 本多佑希: どんくりで学ぶアルゴリズム. 学習情報研究, 2019 年 5 月号, pp.28-29, 2019.

[5] 本多佑希, 兼宗進: DNCL のオンラインプログラミング学習環境「どんくり」の開発. 情報処理学会, 第 81 回全国大

会, 2019.

[6] 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. 情報処理学会, コンピュータと教育研究会 (CE147), 2018.

[7] 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, コンピュータと教育研究会 (CE81), pp.65-71, 2005.

[8] 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, 情報教育シンポジウム (SSS2018), pp.210-214, 2018.

[9] 中西渉: Web ブラウザ上のプログラミング学習環境 PyPEN の改良. 情報処理学会, 情報教育シンポジウム (SSS2020), pp.101-105, 2020.