

DNCL2（仮称）の学習環境の実装に関する考察

中西 渉^{1,a)}

概要：センター試験および大学入学共通テストの「情報関係基礎」ではプログラミングの問題が出題され、そこでは DNCL というプログラミング言語が使われてきた。筆者はこれらのプログラミングの問題でどのような文法事項が使用されてきたかを調査し、その上で共通テストで用いられることが予想されている新しいプログラミング言語 DNCL2（仮称）とその学習環境について考察した。結果として、筆者が開発している学習環境 PyPEN の実装について多くの示唆が得られた。

キーワード：プログラミング教育、大学入試、DNCL、Python

Consideration on implementation of “DNCL2” learning environment.

NAKANISHI WATARU^{1,a)}

Abstract: Programming questions have been asked in the National Center Test and the Common Test, and the programming language called DNCL has been used there. I investigated what grammatical items have been used in these tests, and considered the new programming language DNCL2 (tentative name), which is expected to be used in the Common Tests, and its learning environment. As a result, I got many suggestions about the implementation of the learning environment PyPEN that I'm developing.

Keywords: Education of programming, University entrance examination, DNCL, Python

1. はじめに

「情報関係基礎」は大学入試センター試験、大学入学共通テスト（以下ではまとめて「センター試験」とする）の数学②で選択できる科目の1つである。受験者は毎年数百人程度であるが、教科「情報」を担当する教員が参考にできる良問が出題されてきた。そして2021年5月、情報処理学会の情報入試委員会によって1997年度以降の問題が公開された[1]。この試験のプログラミングの問題ではDNCLと言われる、日本語ベースの擬似言語が用いられてきた。

2022年度から実施される学習指導要領に対応した入試が行われる2025年度入試では、大学入学共通テストに「情報」が取り入れられることが検討されている[2]。それに先立って、大学入試センターは2020年11月、教育委員会等

に試作問題（検討用イメージ）[3]を示して意見を求めたほか、2021年3月にサンプル問題[4]を公開している。そのプログラミング問題では、日本語ベースではあるがDNCLとは違った擬似言語が用いられており、それは一部関係者の間ではDNCL2と言われている。

本稿はこれらを元に、DNCL2（仮称）に対応したプログラミング実習を行う環境について、行った考察について述べる。第2章ではこれまでに出版されたプログラミング問題で用いられた言語について整理し、第3章ではDNCL、DNCL2、Pythonを比較する。第4章で既に開発されているDNCL2の学習環境について述べ、第5章でそれらについて考察し、特に筆者が開発している学習環境PyPENで行うべき実装について検討する。

2. 「情報関係基礎」プログラミング問題の変遷

センター試験「情報関係基礎」の本試験で出題されたプログラミング問題（2005年まで必答、2006年から表計算

¹ 名古屋高等学校
Nagoya Senior High School
^{a)} watayan@meigaku.ac.jp

との選択)について [1] に基づき調査を行った。センター試験では追試験が実施された年度もあるが、筆者が入手できているのは最近のいくつかしかないので、今回は対象としなかった。

2.1 言語

1997 年は BASIC, COBOL, Pascal で出題がなされていたが、翌年からは日本語による擬似言語が用いられることとなった。1998 年の問題ではブロック終端を「～はここまで」と表記していたが、1999 年にはそれを「～を実行する」「～を繰り返す」で表現するなど DNCL にやや近づいたものになっており、2002 年にはほぼ現在の DNCL になっている。

この擬似言語の仕様については、当初は「読めばわかる」としていたが、2002 年の試験問題評価委員会報告中の問題作成部会見解で仕様を示され、翌年の同文書で DNCL と命名された。その後「DNCL と表計算言語の仕様」は予め公表すべきだとの指摘を受け、2011 年以降の試験のために「センター試験用手順記述標準言語 (DNCL) の説明」[5] が公開され、今に至っている。たとえばそれ以前は 2002 年の仕様に記載されていた後判定のループ (repeat～until) を使った問題が出題されていたが、2011 年度以降は一度しか使われていない。

試作問題やサンプル問題で提示されたのは、日本語ベースであるがそれまでの DNCL と異なり、Python のようにインデントでブロック文を表現する言語であった。本稿ではこれを DNCL2 と呼ぶことにする。

2.2 題材

プログラミングの問題で扱われてきた題材および構文項目は表 1 の通りであった。情報関係基礎の科目が入試に取り入れられた 1997 年度以来、題材がまったく重複していないことは特筆したい。迷路を解く問題は 2002 年度と 2018 年度に取り上げられているが、前者は左手法、後者は袋小路を塞いでいく方法だから全く違うアルゴリズムである。

3. DNCL, DNCL2, Python の比較

DNCL については [5] を参考にすが、DNCL2 はまだ仕様が示されていない。そこで DNCL2 については試作問題およびサンプル問題で使われている文法を参考にす。試作問題とサンプル問題では「《変数》を《値》から…」の行末に「繰り返す」があるかどうかや、インデントの深さの表記方法といった細かい違いはあるが、それほど大きい違いはないものと考えられる。また、Python はバージョン 3 系をもって比較することにする。

比較対象に Python を加えた理由を述べる。学習指導要領では「情報」の授業で使う言語を指定していないが、文

部科学省が教員向けに作成した教員研修用教材 [6] が、「情報 I」については言語別に 5 つの版があるもののそれらは先行して公開された Python 版からの「翻訳」であることや、「情報 II」に至っては明確に Python が中心となっていることなどから、実際には Python が使われることが多いのではないかと現場では噂されていた。試作問題・サンプル問題でその印象はより強くなったものと思われる。DNCL2 も実際 Python を意識した部分が多いと考えられるので、現在仕様が明確でない部分は Python に近いものになることが予想される。

3.1 文字列

DNCL では「」または""で文字列リテラルを表すことになっているが、主に前者が用いられていた (後者は 2018 年の問題などで使われている)。DNCL2 はもっぱら""が使われている。

3.2 出力

DNCL では「《値》を表示する」あるいは「《値》を改行なしで表示する」が用いられている。複数の値を並べて出力するには「と」で結合することができる。

DNCL2 では「表示する ()」が用いられている。改行なしをどのように扱うのかはまだわからない。試作問題では配列の値を「表示する (《配列変数名》)」で表示しているところもあるし、複数の値をコンマ区切りで示すことで並べて表示できるようである。これは Python の print() を模したものと思われる。

3.3 ブロック文

DNCL では条件分岐やループで実行される文のブロックをインデントとブロック終端の「(もし○○ならば...)」を実行する」「(○○の間, ...)」を繰り返す」というような句で表現している。BASIC でいうと IF に対応する END IF や WHILE に対応する WEND のような書き方である。

一方 DNCL2 ではブロックをインデントで表現することは同じなのだが、ブロック終端の句を置かない。これは Python の書き方に近いものと考えられる。

Python では行頭の空白の数によってインデントのレベルが示されているが、印刷や手書きではそれが正確に表現されにくい (特に深いインデントから浅いものに戻るときに、どのレベルなのかがわかりにくい)。そのためか DNCL や DNCL2 では、文字下げ部分には「|」「└」などの記号を入れて、文字下げレベルの数を表している。簡単なプログラムの例を図 1 に示す。

3.4 演算子等の記号

DNCL では代入に「←」の記号を用いてきたが、DNCL2

表 1 プログラミング問題の題材と構文項目
 Table 1 Themes and syntax items used in programming problems

年度	題材	if	else	elif	前条件 ループ	後条件 ループ	for	条件 演算	switch	一次元 配列	二次元 配列	配列 初期化
1997	パターンを描く	○	○				○					
1998	ヒットアンドブロー	○			○			○		○		
1999	配列のマージ	○	○		○					○		
2000	二分探索					○			○	○		
2001	金種計算	○	○		○		○			○		
2002	迷路 (左手法)	○				○			○		○	
2003	文字列探索	○					○	○		○		
2004	小町算	○	○		○		○			○		○
2005	じゃんけん	○	○				○	○		○		
2006	コンウェイのライフゲーム	○	○				○			○	○	○
2007	計算ブロック	○		○		○	○	○			○	
2008	ソート	○				○	○			○		
2009	素因数分解	○			○		○			○		ループ
2010	漢数字への変換	○	○				○			○		
2011	得点集計, 順位	○					○			○		○
2012	分銅 (2 進法, 符号付 3 進法)	○			○		○			○		○
2013	営業時間の最適化	○		○			○			○		
2014	太鼓の楽譜	○	○		○					○		
2015	ボードゲーム	○		○	○			○			○	
2016	荷物の箱詰め	○			○		○	○			○	
2017	三角形の個数	○			○		○	○		○	○	
2018	迷路 (袋小路を塞ぐ)	○	○				○	○			○	
2019	生徒のグループ分け	○				○	○	○		○	○	
2020	宝探しゲーム	○			○		○	○		○		
2021	スゴロク	○	○		○		○	○		○	○	
試作問題	シーザー暗号	○	○				○			○		○
サンプル	ドント方式	○			○		○	○		○		○

※ 「条件演算」は「かつ」「または」「でない」, 「配列初期化」の「ループ」はループによる初期化.

◇ DNCL のプログラム例

```
current ← 2021
もし current-1964=17 ならば：
  「おいおい」を表示する
  を実行する
```

◇ DNCL2 のプログラム例

```
current = 2021
もし current-1964=17 ならば：
  「おいおい」を表示する
```

◇ Python のプログラム例

```
current = 2021
if current-1964==17:
    print("おいおい")
```

図 1 プログラム例

Fig. 1 Example of code

では「=」が用いられている。一方 DNCL では等値演算子として「=」が用いられてきたが、試作問題・サンプル問題ではそれが使われなかったため、DNCL2 で「=」「==」のどちらを用いるつもりであるかは判断できなかった。しかし比較演算子として DNCL で用いてきた「≠」「≤」ではなく「!=」「<=」が DNCL2 では用いられていたことから、Python に合わせた「==」が使われることはあり得ると考えられる。

整数の割り算に関して、DNCL は「÷」を商の整数部分として「/」と区別して扱ってきた（ただしこれが使われるときには、問題文中で説明を加えている）。Python では「//」という演算子があるのだが、DNCL2 ではそれを用いず、商を「切り捨て ()」という関数で整数にしている。

3.5 論理演算

DNCL では論理演算に「かつ」「または」「でない」が用いられるが、DNCL2 では「and」「or」「not」が用いられている。これも Python に合わせたものであろう。

DNCL では「でない」が「かつ」「または」より優先されることなく、左から順に評価されることになっているので、「1=1 かつ 1=2 でない」は「(1=1 かつ 1=2) でない」と解釈されて偽になる。しかし C や Python では論理 NOT、論理 AND、論理 OR の順に優先順位が高いので、同じように書いた論理式が別の評価になることがあり得る。一方 DNCL2 ではこのあたりの事情はまだわからない。しかし DNCL は本来「読めばわかる」ように考えられたものであるから、このような細かい仕様が重要な要素になるような出題は避けられるものと期待する。

3.6 後判定のループ

DNCL にはブロック文の後の句で条件が書けるから後判定のループが存在するが、インデントの前の行でループの条件を表現する DNCL2 や Python ではそれができない。しかし、while True: の中で break することで同等の処理を実現できる。



図 2 「つちのこ」の実行画面

Fig. 2 Screenshot of “tsuchinoko”

2.1 で触れたように、後判定のループは 2019 年度の問題で使われているが、これは容易に前判定のループで書き直すことができるものであった。

4. DNCL2 (仮称) の実行環境

DNCL では、問題のコードを実行して学習できる環境として PEN[7] などが用いられてきた。DNCL2 についても同様に、試作問題やサンプル問題のプログラミング問題のコードが実行できるよう、いくつかの環境が開発されている。

4.1 どんくり

「どんくり」[8] は Web ブラウザ上で DNCL をプログラミング言語として記述して実行できる環境であり、DNCL2 に対応するよう機能拡張が行われた [9]。公開されている実行環境を見つけれなかったため、本稿では [9] を元に比較を行った。

4.2 つちのこ

「つちのこ」[10] は Web ブラウザ上で DNCL2 を実行できる環境である。変数の変化の様子や、各行の実行回数などを表示するビューが用意されているほか、「つちのこ」で作ったプログラムを Python のコードに変換する機能もある。実行画面の例を図 2 に示す。

Python の導入をスムーズに進めるために開発されたものであり、配列だけでなくタプル、辞書、集合にも対応しているほか、入出力の関数も Python に近づけたものになっている。

4.3 Picthon

人型ピクトグラムを題材にしたプログラミングができるピクトグラミング [11] には、Python 版の Picthon があり、DNCL2 をサポートする計画があるという [12]。既にサンプル問題で示されているプログラムは正常に作動するほか、(当然のことではあるが) 人型ピクトグラムを DNCL2 の文法によって動かすこともできる。実行画面の例を図 3 に示す。

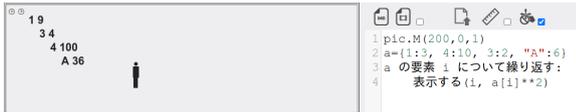


図 3 Picthon の実行画面

Fig. 3 Screenshot of Picthon



図 4 PyPEN の実行画面

Fig. 4 Screenshot of PyPEN

4.4 PyPEN

PyPEN は筆者が開発している、Web ブラウザ上で動作する DNCL2 の学習環境である [13]。コードをフローチャートで表現するほか、Python に変換したコードを出力することもできる。実行画面の例を図 4 に示す。

5. 学習環境に関する考察

DNCL2 の学習環境はさまざまなものが提案されている。上述したものはすべて Web ブラウザ上で実行できるものであるから、インストールなどの面倒な作業をする必要はない。しかし DNCL2 の仕様が明確でないため、それぞれの「方言」が存在している。

ここではその違いを元に考察を行い、特に筆者が開発している PyPEN ではどのように実装を進めるかについて、得られた考えを述べる。

5.1 演算子記号

3.4 で述べたように、DNCL と DNCL2 では代入記号や比較演算子が異なっている。PyPEN では「=」でも「←」でも代入できるし、「=」でも「==」でも値を比べられるようになっているが、これは必ずしも利点とはいえない。同じ記号が別の意味で用いられるのは構文を解釈する上で面倒であるし、筆者は PyPEN を Python に移行するための足がかりと考えているので、揃えられるところは揃えておきたいと考えている。実際他の学習環境は「=」で代入、「==」で比較となっている。

即座に揃えたいと考える一方、たとえば文部科学省の「高等学校『情報』実践事例集」[14]に掲載されている PyPEN を用いた事例についてプログラムを若干書き直す必要が生じるなど、過去の資料がそのまま使えなくなることを懸念する。

5.2 論理演算

Picthon や「つちのこ」では、DNCL2 や Python に合わせた「and」「or」「not」を用いている。一方 PyPEN では「かつ」「または」「でない」となっている。ただし DNCL2 のように左から評価するのではなく、Python のように優先順位をつけているので 3.5 に述べたものと違い「1=1 かつ 1=2 でない」は「1=1 かつ (1=2 でない)」と解釈されて真になる。これについても DNCL2 に合わせたいが、5.1 と同じ点で躊躇している。

5.3 配列の初期化

DNCL2 では [《値》, 《値》, ...] の構文で配列を初期化できるようなので、この構文はどの環境でも用意されている。それ以外にも、「どんくり」は {1,2,3,...100} というような構文を使って等差数列で配列を初期化できるし、「つちのこ」は Python の range に似た区間オブジェクトを作ることができる。

また、DNCL や DNCL2 では「《変数名》のすべての値を《値》にする」「配列《変数名》のすべての要素を《値》で初期化する」「配列変数《変数名》を初期化する」などの構文で配列を初期化することができることになっている。しかし要素数の指定がないので、これをそのまま実装するのは厄介である。「どんくり」の DNCL 版では、値がまだ代入されていない配列の要素にアクセスしたときには指定された初期値を返す（ただし配列の要素の値は変更しない）実装になっている。これは DNCL や DNCL2 で想定されている実装なのだろうと思う反面、Python にはそのまま持ち込めないで、結局大きさを決めた初期化を追加しなくてはならないように思われる。あるいは「どんくり」の配列と同じ振る舞いをするクラスを作っておいて常用する方法もあるかとは思いますが、煩雑な要素が増えることは否めない。

PyPEN では「《個数》個の《式》」で配列を初期化できるようにしているが、これは [0]*100 のように配列に対する演算を強化することで替えられるとも考えている。

5.4 比較演算子の結合

比較演算子の結合はプログラミング言語によって扱いが違う。たとえば C では「5>4>3」は「(5>4)>3」すなわち「1>3」と解釈されて偽になるのだが、この挙動は直感に反すると言われることがある（「3<4<5」なら真ではあるが、数学で 3 < 4 < 5 が真であることとは意味が違う）。一方たとえば PHP ではこのような結合をしないので「5>4>3」は文法エラーとなり、そのような問題は起きない。

しかし Python では「5>4>3」は「5>4 and 4>3」と解釈されて True になる。筆者は寡聞にしてこのような解釈をするプログラミング言語を他に知らないのであるが、DNCL2

ではどうすべきだろうか。

たとえば Picthon は Python と同じ挙動であるが、「つちのこ」は C と同じ挙動である（ただし $5 > 4$ は 1 でなく True なので、 $5 > 4 > 3$ は論理型と数値型の比較と解釈されてエラーになる）。PyPEN は最近まで PHP と同じように結合しない（エラーになる）仕様であったが、今は Python と同じ挙動にしている。

初学者がこの Python の挙動を当たり前と誤ってしまふと、他の言語を使うときに間違ふ原因になることを懸念している。PHP のようにエラーになるならいいが、C のようにエラーにならずに意図しない結果が生じると厄介なバグの原因になりかねない。この手の紛らわしい構文に対して、カッコをつけるよう警告を出せるコンパイラもあるが、たとえば gcc では parentheses warnings なので $(5 > 4) > 3$ のようにカッコをつけると警告が消えてしまふ^{*1}。

5.5 二次元配列の添字

表 1 にあるように二次元配列は何度も出題されている。DNCL では二次元配列の添字は $a[b, c]$ であるが、Python では $a[b][c]$ であるから、DNCL2 もそのようになる可能性がある。

「つちのこ」や Picthon は $a[b][c]$ 形式であり、PyPEN は両方の書き方に対応している。もっとも、対応していることが必ずしも利点でないことは 5.1 で述べた通りである。

5.6 elif

DNCL には「（もし《条件 1》ならば～）を実行し、そうでなくもし《条件 2》ならば」というような elif に相当する構文が定義されており、実際にセンター試験でも何回か使われてきた。「つちのこ」には「そうでなくもし《条件 2》ならば」という構文が実装されている。

PyPEN では elif にあたる構文を用意していないが、それはフローチャートで表現する方法を筆者がうまく思いついていないためである（JIS X 0121 では表現方法が規定されている）。フローチャートからコードを生成することを諦めれば可能かもしれないが、影響範囲が大きいので安易には決断できない。コードを書き換えて else 節の中に if を置くことも考えられるが、インデントが 1 段深くなってしまふので、後述する switch 文の置き換えとしては使いにくいものになってしまう。

センター試験の初期には「《式》の値に応じて、次のいずれかを実行する」という、switch 文のような構文が使われたことがあるが、DNCL の説明にもこれは掲載されていないし、Python にも switch 文は存在しない。これに相当するものは elif で実現できるので、switch 文に相当するものは不要であろう。

^{*1} $5 > 4 > 3$ と $(5 > 4) > 3$ の結果は Python でも違い、後者は C と同じ結果になる。

5.7 for in

Python の for in の構文については、Picthon や「つちのこ」では「《配列》の要素《変数》について繰り返す」という構文で実装されている。図 3 のプログラムはこの構文を使っている。Python の「for 《変数》 in 《配列》」とは語順が逆転しているが、日本語として自然なものにするためには仕方がない。PyPEN では未実装であるが、同じ構文での実装を計画している。

5.8 型

Python ではリストだけでなく、タプル、辞書、集合といった型が使える。ここであげた実行環境の「配列」は要素を追加したりすることができる点でリストに近いものと考えられる。また、Picthon や「つちのこ」ではこれらのリスト以外の型も実装されているので、変数の値交換を「 $a, b = b, a$ 」のように書くこともできる。PyPEN では配列と辞書が実装されているが、タプルや集合は未実装である。実装することは可能だとは思われるが、

もっとも $a, b = b, a$ で変数の値交換ができることについては、Ruby の多重代入や、Perl のリストへの代入などいくつかの言語でも同じような記法で行うことが可能なので、5.4 のような心配はいらぬように思われる。

5.9 in

Python には、リストや集合にある値が含まれているかどうかを判断する in 演算子がある。PyPEN では「《配列》の中に《値》」として実装したが、for in と同じように、Python とは語順が逆転している。一方、Picthon では in をそのまま用いている。

5.10 組み込み関数

試作問題やサンプル問題では「切り捨て」「差分」「要素数」などの関数があるという前提で問題が作られている。これは DNCL にもあったものであり、言語が元々持っているということだけでなく、問題に応じて必要な関数とその都度定義されているというものだと考えられる。

これについてはプログラムに記載されたコードと別にこれらの関数の定義を書けばいいと考えている（そもそもセンター試験の問題は値の初期化部分などが問題のプログラムには書かれていないことが多い）。もっとも PyPEN では識別子を英字に限っているので、日本語文字でも可能になるよう拡張することを検討すべきであろう。

5.11 その他

Python では関数の中などからグローバル変数を参照できるが、その値を変更するためには global 文で指定する必要がある。共通テストでそれが必要となるような出題があるとは考えにくいですが、筆者自身が PyPEN でプログラム

を作っていて時々これがあるといいと感じることがあった。あるいは、型によって値渡しと参照渡しを切り替えればいいのかもしれない。

三項演算子は Python では「《値 1》 if 《条件》 else 《値 2》」というように、英語の語順そのまま実装されている。「つちのこ」はこれと同じ語順で「《値 1》もし《条件》そうでなければ《値 2》」という構文になっているが、日本語としては不自然に感じられる。むしろ C などの三項演算子の語順のように「もし《条件》ならば《値 1》そうでなければ《値 2》」とした方が読みやすいのではないだろうか。

6. おわりに

センター試験および、共通テストの試作問題・サンプル問題を元に、試験で使われるプログラミング言語 DNCL および DNCL2 について考察を行った。DNCL2 は試験のための言語ではあるが、それが紙上だけで終始するものにならないように、何人もが DNCL2 の実行環境を開発しているのだと考えている。DNCL2 を学んだ生徒がプログラミングで自分の思考を拡張できるように、その基礎を作るにふさわしい環境が作られることが望まれる。

その一方で、学ぶ生徒たちが将来使う言語は様々なことから、DNCL2 や Python だけの「常識」やバッド・ノウハウに囚われることなく、問題解決のための細部に専念できるように、安心して使えるものにしていきたい。

今回、複数の環境を比較することによって、PyPEN の実装に有益な多くのヒントやアイデアを得ることができた。今後の実践に役立てることで、些少なながらも成果を返すことができると考えている。しかしあくまでも PyPEN は Python の代替を目指すものではなく、DNCL の「読めばわかる」を具現化することが望まれるものである。細部に拘泥することは実装者としては楽しいことではあるが、当初の目的を見失わないようにしていきたい。

参考文献

- [1] 情報処理学会情報入試委員会：情報関係基礎アーカイブ，(オンライン)，入手先 (<https://sites.google.com/a/ipsj.or.jp/ipsjrn/resources/JHK>) (参照 2021-06-20)。
- [2] 大学入試センター：平成 30 年告示高等学校学習指導要領に対応した令和 7 年度大学入学共通テストからの出題教科・科目について (2021)。
- [3] 情報処理学会：大学入学共通テストへの「情報」の出題について，(オンライン)，入手先 (<https://www.ipsj.or.jp/education/edu202012.html>) (参照 2021-06-23)。
- [4] 大学入試センター：令和 7 年度以降の試験，(オンライン)，入手先 (https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html) (参照 2021-06-23)。
- [5] 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，(オンライン)，入手先

- (<https://www.dnc.ac.jp/albums/abm00004841.pdf>) (参照 2021-06-20)。
- [6] 文部科学省：高等学校情報科教員研修用教材，文部科学省(オンライン)，入手先 (https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416746.htm) (参照 2020-10-02)。
- [7] 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, pp. 2736-2747 (2007)。
- [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，研究報告コンピュータと教育 (CE)，Vol. 2018-CE-147, No. 10 (2018)。
- [9] 本多佑希，漆原宏丞，兼宗 進：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，研究報告コンピュータと教育 (CE)，Vol. 2021-CE-159, No. 2 (2021)。
- [10] 大門 巧：つちのこ，(オンライン)，入手先 (<https://tdaimon.jp/tsuchinoko/>) (参照 2021-06-21)。
- [11] 伊藤一成：ピクトプログラミング—人型ピクトグラムを用いたプログラミング学習環境—，情報教育シンポジウム論文集，Vol. 2017, No. 16, pp. 110-120 (2017)。
- [12] 伊藤一成：[ピクソン] ピクトプログラミング Python 版である Picthon (ピクソン) で，共通テスト「情報」で採用予定の DNCL2 をサポート，(オンライン)，入手先 (<https://pictogramming.org/?p=2786>) (参照 2021-06-21)。
- [13] 中西 渉：プログラミング学習環境 PyPEN の開発，日本情報科教育学会第 13 回研究会報告書，No. 5 (2019)。
- [14] 文部科学省：高等学校「情報」実践事例集，(オンライン)，入手先 (https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_01342.html) (参照 2021-06-26)。