

# DNCL (ディーエヌシーエール)

## — どんなコードにもエールを (DonNa Code にも L を) —

伊藤一成<sup>1,2</sup>

**概要:** 大学入学共通テストで、2025 年度入試から教科「情報」が導入される。ここではプログラミング分野の問題で扱われる擬似言語仕様として Python 言語に類似した DNCL (発音は「ディーエヌシーエール」) が用いられる予定である。DNCL は、Daigaku Nyushi Center Language の略と言われており、その名の通り大学入学共通テスト用に策定された擬似言語であると認識されている。一方 DNCL の導入により、教育産業の介入も相まって高等学校情報科の授業が入試対策ばかりに向いてしまい、汎用プログラミング言語を使用して自らコーディングする活動が軽視されてしまうのではないかという懸念も根強い。そこで「どんなコードにもエールを」(DonNa Code にも L を) の略語としての DNCL (発音は、ディーエヌシーエール) を提案する。

また、筆者が開発しているピクトグラムコンテンツ作成環境「ピクトグラミング」の派生アプリケーションの一つとして、アプリケーション「DNCL」(発音は、ディーエヌシーエール) を試作したので報告する。DNCL (ディーエヌシーエール) では、DNCL (ディーエヌシーエール) で記述されたコードに加え、Python, JavaScript, ビジュアルブロックで記述されたコードを実行できる機能と共に「どんなコードにもエールを」のコンセプトを踏まえた様々な機能が備わっている。

**キーワード:** 高等学校情報科, 大学入学共通テスト, DNCL, ピクトグラミング, Python, JavaScript, ブロック型

## DNCL (DonNa Code nimo L wo)

KAZUNARI ITO<sup>1,2</sup>

**Abstract:** The subject "Informatics" will be introduced in the Common Test for University Admissions starting with the entrance examination in 2025. The DNCL (Daigaku Nyushi Center Language), which is similar to the Python language, will be adopted as a pseudo-language specification for questions in the programming field. DNCL is widely recognized as a pseudo-language designed for the Common Test for University Admissions. On the other hand, there is a strong concern that the introduction of DNCL, combined with the intervention of the education industry, will cause the informatics classes focus only on entrance exam preparation and neglect activities in which students code by themselves using a general-purpose programming language. This paper first proposes DNCL (pronounced "DNC-ale") as an abbreviation for "DonNa Code nimo L wo."

In addition, this paper also proposes the application DNCL (pronounced "DNC-ale") as one of the derivative applications of the pictogram content creation environment "Pictogramming" developed by the author. The application DNCL can execute DNCL as well as Python, JavaScript and block-typed language. DNCL aims to be an application that incorporates the concept of "DonNa Code nimo L wo."

**Keywords:** Informatics in high school, The Common Test for University Admissions, DNCL, Pictogramming, Python, JavaScript, Block-typed

### 1. はじめに

高等学校では、2022 年から情報科において「情報 I」が必修科目として運用されている。「1.情報社会の問題解決」「2.コミュニケーションと情報デザイン」「3.コンピュータとプログラミング」「4.情報通信ネットワークとデータの活用」が設定されている。「3.コンピュータとプロ

グラミング」において、主に Python, JavaScript, VBA, Scratch の利用が想定されている[1].

さらに、大学入学共通テストでは、2025 年から教科「情報」が導入される予定であるが、ここでは、プログラミング分野の問題で擬似言語仕様として Python 言語をベースとする DNCL が用いられる予定であり、すでにサンプル問題や試作問題が公開されている[2][3]. 従来から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」

<sup>1</sup> 青山学院大学社会情報学部

<sup>2</sup> 青山学院大学ピクトグラム研究所

で使用されてきた DNCL に対して、新たに策定された DNCL は新 DNCL や DNCL2 などと表記されることがあるが、本稿ではこの新しく策定された仕様を DNCL と表記する。

共通テスト「情報」に際しては、授業で学習するプログラミング言語によって有利不利が生じてしまうのではないかとこの心配の声も聞かれている。これに対して井出は、情報 I の教科書で主に扱われている 4 言語に関して、どの言語を用いて授業を履修しても、DNCL で記述されたプログラム問題の成績に大差がなかったことを報告している[4]。

DNCL は、その名の通り、大学入学共通テスト用に策定された擬似言語であると広く認識されている。一方、DNCL の導入により情報科の授業や教育産業の視点が入試対策ばかりに向いて、汎用プログラミング言語を使用して自らコーディングする活動が軽視されてしまうのではないかとこの懸念も根強い。このような、情報教育の観点からの擬似言語と汎用プログラミング言語の関係は、ときに教育界のみならず産業界も巻き込んだ論争の種となる。

また、「擬似言語」なのか「擬似コード」なのかの観点からの議論も必要と考えられる。特に生成系 AI の登場に伴い、日常的かつ即興的に擬似コードを記述する能力の重要性が増大している。これは、高等学校情報科における教育を考える上でも無視できない。従来、高等学校情報科における擬似言語は、受験対策という観点からばかり議論されてきた。大学受験という一時的イベントのためという限定的な観点ではなく、日常的活動の一つとして、コード体系を包括的、多角的な視野から理解し、探求することが強く推奨される。それは新しい AI の発展と共に、擬似コードの理解と利用がますます重要なスキルとなっていく現代社会の教育に対する応答としての意義も持つと考えられる。

このように DNCL をめぐる様々な問題や検討すべき点が残存する中で、それらの問題の改善や、議論の旺盛に寄与する意味を内包した DNCL を略記とする別フレーズを提案することは意義があると考えられる。さらに、そのフレーズが情報科教育の理念に合致する意味内容を含むのであれば、そのフレーズを日常的に学習者が目に触れるようにすることで一定の効果が生じると考えられる。

そこで、本稿では DNCL を略語とする別語意として「どんなコードにもエールを」(「DonNa Code にも L を」、発音は「ディーエヌシーエール」)を提案する。エールというカタカナ単語は和製英語であり、応援や励ましの意味で、日本では広く使われている。これを文字「L」と対応づけた。さらに、そのコンセプトに基づいて Web アプリケーション「DNCL (ディーエヌシーエール)」を試作したので報告する。

以下 2 章で、アプリケーション DNCL (ディーエヌシーエール) を説明する上で前提となる、ピクトグラミング及びその派生アプリケーション群について紹介する。3 章

で、今回試作したアプリケーション DNCL (ディーエヌシーエール) について説明する。4 章で関連研究も踏まえながら考察し、5 章でまとめる。

## 2. ピクトグラミングおよび派生アプリケーション

本章では DNCL (ディーエヌシーエール) を解説する上で前提となる、ピクトグラミングおよび派生アプリケーション[5]について解説する。

### 2.1 概要

ピクトグラムは意味するものの形状を使ってその意味概念を理解させる記号である。ピクトグラムはコンテンツ表現の抽象度の高さから、それを見た人物が自分自身や本人に関わる人物物事など想起させる効果があると言われている。有名な「非常口」ピクトグラムのデザインの策定に関わった太田は” 走る人型を囲む空間が見る人を包む空間とつながって走る人は見る人の投影になる” と述べている[6]。避難中の人を、出口へ向かって走る人型ピクトグラムを見たときに、それを自身に投射し同一視するかに、非常口ピクトグラムのデザインの労力が払われた。この特性は情報教育の諸分野にも活用できるのではないかと考え、開発したのがピクトグラミングである。

ピクトグラミングは、ピクトグラムを作るプロセスの中で、プログラミングの諸概念を体得できるように設計されている。また、ピクトグラミングは、ピクトグラムのデザイン指針に準じたコンテンツを作成できるアプリケーションでもある。

### 2.2 画面説明

ピクトグラミングは、Web アプリケーションでありブラウザ上で動作する。図 1 にスクリーンショットを示す。

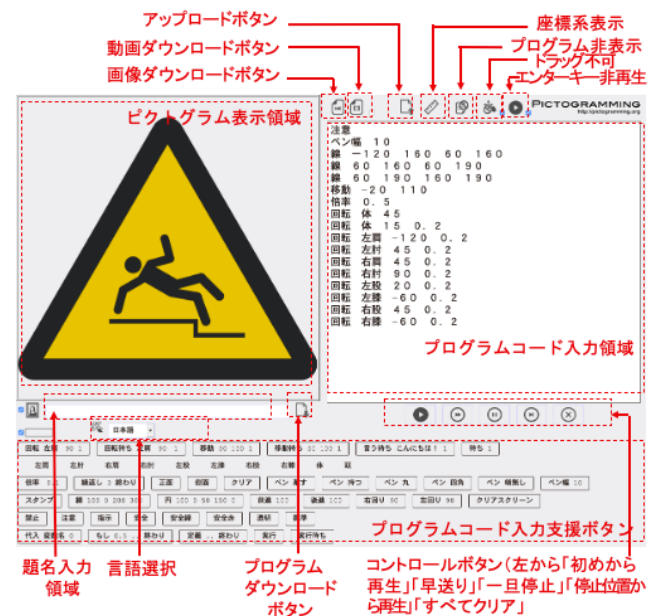


図 1 ピクトグラミングのスクリーンショット

Fig.1 Screenshot of Pictogramming

画面左上部は、プログラムの実行結果を表示する”ピクトグラム表示領域”，右上部はプログラムを入力する”プログラムコード入力領域”，下部はプログラムの入力を支援するボタン群で構成される”プログラムコード入力支援ボタン領域”がある。ピクトグラム表示領域には、初期状態では、人型ピクトグラムが1体表示される。

図1で示したピクトグラミングとほぼ同一のユーザーインターフェースを備える、派生アプリケーションのスクリーンショットの例を図2に示す。

図2(a)に示すのが、ビジュアルブロックを使って記述可能なピクトグラミングの派生環境である Pictoch (ピクトッチ) である。図2(b)に示すのが、Python 言語で記述できる Picthon (ピクソン) である。さらに、JavaScript 言語で記述できる JavaSpict (ジャバスキクト)、Ruby 言語で記述できる Picby (ピクビー) もある。これらはすべて同一インターネットサイト上で公開している。ジャバスキクトやピクビーも図2(b)と同様の画面構成である。これらをまとめてピクトグラミングシリーズと総称する。



(a) ピクトッチ



(b) ピクソン

図2. 派生アプリケーションのスクリーンショット  
Fig.2 Screenshot of derivative applications

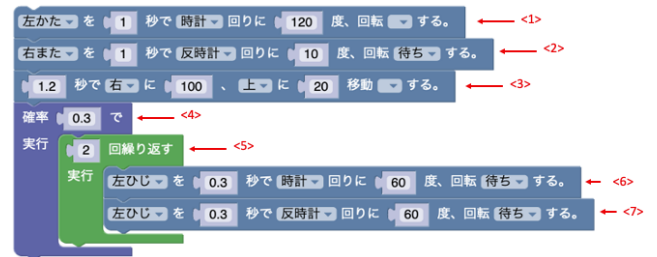
2.3 プログラム記述仕様

ピクトグラミングシリーズでは、大別して次の3種類

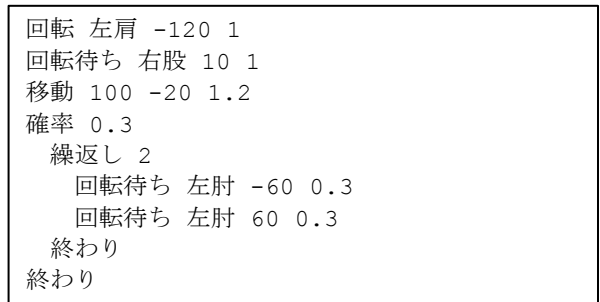
の命令を備えている。

- (1) 人型ピクトグラムの変形及びアニメーションの命令
  - (2) 様々なオブジェクトを描くための図形描画の命令
  - (3) 禁止, 注意等のピクトグラム固有のマーク描画の命令
- またそのほか、変数定義, 繰返し, 条件分岐や関数定義などの命令が用意されている。

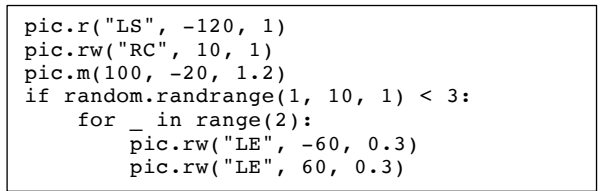
人型ピクトグラムの変形及びアニメーションは、回転と移動の2種類の命令を使い分け、さらに命令の逐次実行と並行実行を組み合わせることで、人型ピクトグラムの多様な動作を可能としている。図3に、人型ピクトグラムが左手を振るプログラム例を示す。



(a) ピクトッチ



(b) ピクトグラミング (日本語)



(c) ピクソン

図3 プログラム例  
Fig.3 Example of programs

図3(a)に示す、ピクトッチのプログラムを例に、各命令と動作について説明する。図3(a)中の<1>で示されたブロックは、身体部位に対する回転の命令である。関節点、回転方向・角度(反時計周りが正)、回転に要する秒数を設定する。

また次のブロックを同時に実行するか、実行が終了するまで、次の命令の実行を開始しないかを「待ち」の有無で選択できる。よって、図3(a)中の<1>と<2>のブロックは並行実行され、<3>で示されたブロックは、<2>のブロックに対応する動作が終了したのち実行される。<3>のブ

ロックは体全体を平行移動する命令である。<4>のブロックは、このブロックに囲まれた内側の命令を一定の確率で実行する。分岐命令に相当する。<5>のブロックは、このブロックに囲まれた内側の命令を指定回数繰り返す。つまり<4>から<7>は、時計回りと反時計回りに左ひじを回転することで 2 回手を振る動作を 30%の確率で実行する動作を記述している。

図 3(a)と同じ出力をする、ピクトグラミングの日本語で書かれたプログラム例を図 3(b)に示す。命令名に続いて複数の引数をスペース区切りで記述する形式となっている。ちなみにピクトグラミングは日本語だけではなく、アルファベットやひらがなによる記述もサポートしている。同様に、図 3(c)にピクソンの例を示す。picという識別子のオブジェクトがあらかじめ定義されており、そのオブジェクトに対するメソッド呼出しの形式で、ピクトタッチやピクトグラミングと同じ命令群を記述できる。ジャバスクピクトやピクビーの場合も同様である。

また、単純図形やセーフティマーク（禁止、注意マーク等）の作画は、ピクトグラム作成において不可欠な工程である。プログラミングの入門講座で、図形描画がテーマの実践も多い。比較的単純な命令セットで扱え、またコードと対応する出力が細かい粒度で視覚化されるのが理由と考えられる。図 1 のスクリーンショットの例で示されているのは、「段差注意」のピクトグラムであるが、ここでは段差のある床を線画で描画しており、注意マークも単一の命令で描画している。

### 3. DNCL (ディーエヌシーエール) の開発

本章では、「どんなコードにもエールを」のコンセプトに基づいて試作した Web アプリケーション「DNCL (ディーエヌシーエール)」について解説する。

#### 3.1 アプリケーション仕様

表 1 に、DNCL (ディーエヌシーエール) と Python 言語の記述対応例を示す。このように両者は類似している。

そこで、DNCL (ディーエヌシーエール) は、ピクソン(図 2 の b, 図 3 の c 参照)を改変する形で開発した。さらにジャバスクピクト及びピクトタッチのコンポーネントも組み入れることで、DNCL (ディーエヌシーエール) は DNCL (ディーエヌシーエール) で記述された擬似コードに加えて Python 言語, JavaScript 言語, ビジュアルブロックで記述されたプログラムも実行可能である。

「DNCL・Python」モード、「JavaScript」モード、「ブロック」モードの 3 モードを有し、それらは画面右下のボタン群を押下することで切り替えられる。「DNCL・Python」モードのスクリーンショットを図 4、「ブロック」モードのスクリーンショットを図 5 に示す。

表 1 Python 言語による記述と対応する DNCL による記述の一例

Table 1 An example of DNCL description and corresponding Python description

	DNCL(ディーエヌシーエール)による記述例	
	対応するPython言語による記述	
1	nyuryoku = 入力する("入力してください")	nyuryoku = input("入力してください")
2	表示する(123)	print(123)
3	i を 0 から 5 まで 1 ずつ増やしなが繰り返す:	for i in range(0, 6, 1):
4	Hairetu の要素 i について繰り返す:	for i in Hairetu:
5	もし i > 0 ならば:	if i > 0:
6	切り捨て(1.5)	int(1.5)
7	関数 kansuu():	def kansuu():

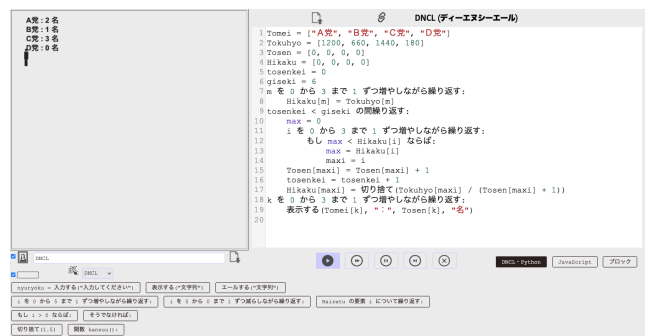


図 4 DNCL (ディーエヌシーエール) 「DNCL・Python」モードの画面構成

Fig.4 Screenshot of "DNCL and Python" mode

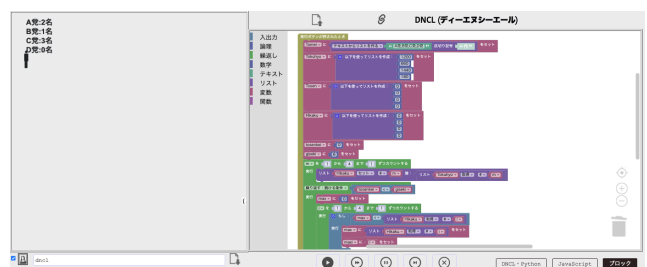


図 5 DNCL (ディーエヌシーエール) 「ブロック」モードの画面構成

Fig.5 Screenshot of DNCL block mode

ピクトグラミングシリーズが想定するピクトグラム作成を主題とするプログラミングと異なり、情報 I で想定されているプログラミングや、大学入力共通テスト「情報」のプログラミング問題では、文字列の入出力が基本となると考えられる。ピクトグラミングシリーズでは、一連の文字列の入出力も可能であり、その機能を使用する。逆にプログラムコード入力支援については、ピクトグラムの作成に

係る入力支援の部分は消去し、文字列の入出力のプログラムを作成することを想定した入力支援に限定して提供する。

ちなみに、文字列の出力は、図4、図5の左上に示すように、人型ピクトグラムがカーソルに擬態化し、文字列を表示する。また文字列の入力は、ダイアログボックス上に表示されるテキストフィールド経由で行う。

### 3.2 エール関数の定義

DNCL（ディーエヌシーエル）では文字列を出力する関数は「表示する」が使用されている。前節で示したように、DNCL（ディーエヌシーエル）でも「表示する」関数を用いて、出力ごとに改行して文字列表示される。DNCL（ディーエヌシーエル）では、新たに文字列を出力する関数として、「エールする」関数を実装した。図6に「表示する」関数と「エールする」関数を使用した場合の出力例を示す。

「エールする」関数を用いた場合は、ピクトグラム表示領域に応援団員に扮した人型ピクトグラムが、効果音と共に三三七拍子による演舞をアニメーションし、エールを送りながら、出力文字列を吹き出しの形式で表示する。

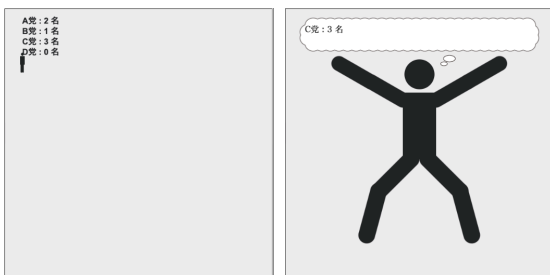


図6 「表示する」関数（左）と「エールする」関数（右）を使用した場合の出力例

Fig.6 Sample output using function "表示する" (left) or function "エールする" (right).

### 3.3 どんなコードにもエール機能

エラーが存在するコードを実行した場合でも、無機質なエラーメッセージを表示するのではなく、「どんなコードにもエールを」という理念に基づいてエールを送りつつ、エラーを提示する機能を実装した。図7にエラー発生時の出力例を示す。

吹き出しの形式で、応援や励ましの表現を文字列で表示する機能を追加した。さらに、「エールする」関数を用いた場合と同様、ピクトグラム表示領域に応援団員を模した人型ピクトグラムが三三七拍子による演舞をアニメーションしエールを送ることで、エラーが発生することが悪いことでないというメッセージを、文字列と身体動作の両方により表現する。

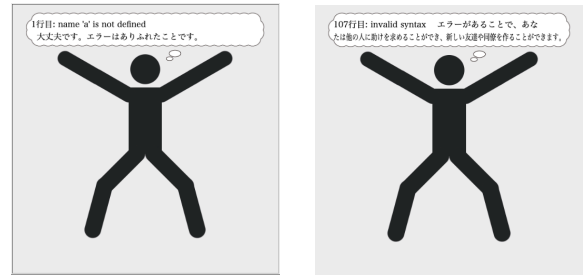


図7 エラー発生時の出力例

Fig.7 Example output when an error occurs

## 4. 考察

### 4.1 DNCL（ディーエヌシーエル）処理系の観点から

DNCL を取り扱ったアプリケーションはこれまでにいくつか提案され、インターネット公開されている。

中西が開発している PyPEN は DNCL の学習環境である [7]。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できる。大門らも、DNCL 処理系「Tetra」[8]を開発し公開している。DNCL の処理系の公開は、DNCL によるプログラミングで完結してしまう作用もあり、検討を要する。実際、擬似コードに固執するあまり、実際に動くコードを書くことなくプログラミング学習を済ましてしまうケースが増える懸念もあるという問題を提議する声は少なくない。高野らも、同様な懸念を踏まえた上で、自然言語によるプログラムの理解のしやすさを活かしたプログラミング言語処理系 Samoyed を提案している [9]。

### 4.2 L（エール）の観点から

テキスト型言語によるつまづきや文法エラーによる学習意欲の減退も検討すべきで、プログラミング教育に関する懸念事項になっている [10]。宮下らは、プログラミングに対するモチベーション向上を意識した言語 HMMML を開発した。プログラミング初学者がミスによりプログラムが動かずにエラーが出る体験を重ねることで、プログラミングに対するモチベーションが低下することを懸念し、この中でコードを超好意的に解釈することでエラーをなくしている [11][12]。

本提案アプリケーションでも、「どんなコードにもエールを」というフレーズに着目して、応援や励ましの観点からモチベーション向上を意識している。

学習支援エージェントの研究に関して、Chase らは、学習者は自分のために学習するよりも仮想的な学習パートナーのためにより多くの努力を投じ、それにより大きな学習の進歩を達成することを明らかにしている。さらに学習過程でのミスは自分ではなく、その仮想的な学習パートナーが犯すという認識が、自身の心理的ストレスを減らす効果があると論じている [13]。

先の太田の言説 [6] に従うとすれば、人の身体的活動や

感覚、知識に基づくポージングやモーションを人型ピクトグラムで直接表現したコンテンツは、身体や自我と同調する活動となる。つまり人型ピクトグラムは、同調的学習と相性の良いコンテンツと考えられ、これがピクトグラミングシリーズの設計理念の一つになっている。DNCL(ディーエヌシーエール)でもこの特徴は継承可能と考えられる。

今回人型ピクトグラムに対して、「応援団員に扮してエールを送る」というロールを初期の設定としたために、本来生じる人型ピクトグラムに対する自己や他者への投射の効果が損なわれている可能性がある。人型ピクトグラムを学習支援エージェントとして、どのように介在させるかは今後の研究課題である。絵のような非言語的情報や、文章を読解することで得られる心的イメージには内容の記憶を促進する効果があるという報告もある[14]。

また、視覚的記号である人型ピクトグラムの特徴として、それが醸し出すユーモアとシュールさが挙げられる。これらの要素を適切に組み入れることは、学習者の関与と動機づけを向上させる可能性を秘めており、その可能性を最大限に引き出すための最善の戦略を探索することは極めて重要であると考えられる。

将来的には、生成系 AI の API 経由で入力プログラムの内容に応じて動的に応援や励ましのメッセージを生成したり、直接実行不可能なプログラムであっても、生成系 AI により実行可能なプログラムに訂正した上で実行したりするなどの機能追加が考えられる。

### 4.3 DNC (どんなコード) の観点から

DNCL(ディーエヌシーエール)との関係で述べれば、「コード」はプログラムコードの意味で捉えがちであるが、「コード」は、多くの意味を持つ単語である。

バイナリコード、文字コード、暗号コード、ピクトグラムに代表される絵記号・図記号など、情報を表現するための様々な符号・記号体系のこともコードという。記号論的コミュニケーションモデルにおいては、情報の送り手と受け手がコミュニケーションを行うための共通の記号体系も「コード」という。「コード」は「倫理規定」や「行動規範」の語意もあり、情報に関する法規やモラル、情報セキュリティの分野と関連性がある。

大西は、探究の段階をアブダクション、演繹、帰納の順に3段階で捉える Peirce の探求段階を、情報科で扱う問題解決方法に対応させる方式を提案している[15]。この中で、「情報デザイン」「プログラミング」「データ活用」の3種類の問題解決方法を扱う理由が不明瞭であることを指摘しており、その結果体系的でない断片的な授業が実施される可能性を懸念している。情報 I で扱う内容が独立した単元群として扱われ、学習者が習得した知識や技術を応用ができないことや、2単位で情報 I の内容すべてを網羅することに注力を置き、動機づけの機会が失われることが懸

念される。

「コード」という観点から情報 I の内容を横断的、統一的に取り扱うことにより、この問題点を改善できる可能性が考えられる。

## 5. おわりに

本稿では、「どんなコードにもエールを」の略語としての DNCL (ディーエヌシーエール) を提案した。また、筆者が開発しているピクトグラムコンテンツ作成環境「ピクトグラミング」及びその派生アプリケーションの開発資産を活用して、アプリケーション「DNCL」(ディーエヌシーエール)を試作した。

今後は、DNCL を略記とする「どんなコードにもエールを」のフレーズの啓蒙及び、アプリケーション DNCL (ディーエヌシーエール) の開発を進めていく。エール手法の個別最適化及びその実装に関しても検討が必要である。その上で授業利用や評価実験を通じて有効性を検証し、改善すべき点を分析、評価する予定である。さらに、アプリケーション DNCL (ディーエヌシーエール) についてもインターネット上で公開予定である。

**謝辞** 本研究は JSPS 科研費 21H03560 の助成を受けたものです。

## 参考文献

- [1] 高等学校情報科「情報 I」教員研修用教材(本編)  
[http://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1416756.htm](http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm) (参照 2023-06-01)
- [2] 大学入学共通テスト「情報」サンプル問題  
[https://www.dnc.ac.jp/albums/abm.php?f=abm00040342.pdf&n=12\\_%E3%82%B5%E3%83%B3%E3%83%97%E3%83%AB%E5%95%8F%E9%A1%8C%E3%80%8E%E6%83%85%E5%A0%B1%E3%80%8F%E3%80%80%E5%95%8F%E9%A1%8C.pdf](https://www.dnc.ac.jp/albums/abm.php?f=abm00040342.pdf&n=12_%E3%82%B5%E3%83%B3%E3%83%97%E3%83%AB%E5%95%8F%E9%A1%8C%E3%80%8E%E6%83%85%E5%A0%B1%E3%80%8F%E3%80%80%E5%95%8F%E9%A1%8C.pdf) (参照 2023-06-01)
- [3] 大学入学共通テスト試作問題「情報 I」  
[https://www.dnc.ac.jp/albums/abm.php?d=395&f=abm00003277.pdf&n=6-2-1\\_%E8%A9%A6%E4%BD%9C%E5%95%8F%E9%A1%8C%E3%80%8E%E6%83%85%E5%A0%B1%E2%85%A0%E3%80%8F%E2%80%BB%E4%BB%A4%E5%92%8C4%E5%B9%B412%E6%9C%8823%E6%97%A5%E4%B8%80%E9%83%A8%E4%BF%AE%E6%AD%A3.pdf](https://www.dnc.ac.jp/albums/abm.php?d=395&f=abm00003277.pdf&n=6-2-1_%E8%A9%A6%E4%BD%9C%E5%95%8F%E9%A1%8C%E3%80%8E%E6%83%85%E5%A0%B1%E2%85%A0%E3%80%8F%E2%80%BB%E4%BB%A4%E5%92%8C4%E5%B9%B412%E6%9C%8823%E6%97%A5%E4%B8%80%E9%83%A8%E4%BF%AE%E6%AD%A3.pdf) (参照 2023-06-01)
- [4] 井手広康: 情報 I の教科書におけるプログラミング分野の比較と考察. 情報処理学会論文誌教育とコンピュータ (TCE), Vol.7, No.3, pp. 8-18 (2022)
- [5] 伊藤一成: 複数のプログラミング言語で記述可能なピクトグラムコンテンツ作成環境の提案と実装, 情報処理学会論文誌 TCE, Vol.7, No.3, pp.1-11 (2021)
- [6] 太田幸夫. ユニバーサル・コミュニケーションデザインの認識と実践 <https://forum8.co.jp/topic/universal107.htm> (参照 2023-06-01)
- [7] 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)
- [8] 大門巧, 大西建輔, 青山浩: XTetra の開発と授業実践による評価. 情報処理学会論文誌 教育とコンピュータ, Vol. 9, No.1, pp. 23-32 (2023)
- [9] 高野志歩, 田村みゆ, 富岡真由, 秋信有花, 倉光君郎: 擬似コードから考える自然言語を活かしたプログラミング言語. 情報教育シンポジウム論文集, pp. 147-151 (2021)
- [10] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen.: "A study of the difficulties of novice programmers." In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05), pp.14-18 (2005)
- [11] 宮下芳明: プログラミングに対するモチベーションを向上させる新言語 HMMMML の開発. 第 51 回プログラミング・シンポジウム予稿集, pp. 57-64 (2010)
- [12] 中橋雅弘, 加藤邦拓, 宮下芳明: 超好意的解釈コンパイラによりかかったプログラマのための激好意的解釈コンパイラ, インタラクション, pp. 191-196 (2013)
- [13] Gambrell, Linda B., and Paula Brooks Jawitz.: "Mental Imagery, Text Illustrations, and Children's Story Comprehension and Recall." Reading Research Quarterly, vol. 28, no. 3, pp. 265-276 (1993)
- [14] Chase, C., Chin, D. B., Oppezzo, M. A. et al.: "Teachable agents and the Protégé effect: Increasing the effort towards Learning", Journal of Science Education and Technology, Vol. 18, No. 4, pp. 334-352 (2009)
- [15] 大西洋: Peirce の探究段階論に基づく「情報 I, II」の構造分析, 日本情報科教育学会誌, Vol.14, No.1, pp.21-28 (2021)