

DNCL2（仮称）の学習環境の実装に関する考察

中西渉

watayan@meigaku.ac.jp
名古屋高等学校

2021年8月29日 (SSS2021)

- ① はじめに
- ② 「情報関係基礎」プログラミング問題の変遷
- ③ DNCL, DNCL2, Python の比較
- ④ DNCL2 (仮称) の実行環境
- ⑤ 学習環境に関する考察
- ⑥ おわりに

1 はじめに

情報関係基礎

- センター試験等の数学②での選択科目
- 「情報」担当者が注目
- 1997年度以降の全問題公開（情報入試委員会）
- DNCL という擬似言語が使われてきた

2025年度入試

- 共通テストに「情報」導入が検討中**決定**
- 試作問題，サンプル問題を公開（大学入試センター）
- 従来の DNCL と少し違う擬似言語...DNCL2?

DNCL2（仮称）に対応した学習環境について考察

- ② 過去問の整理
- ③ DNCL, DNCL2, Python の比較
- ④ 学習環境の比較
- ⑤ 考察

→PyPEN で行うべき実装は?

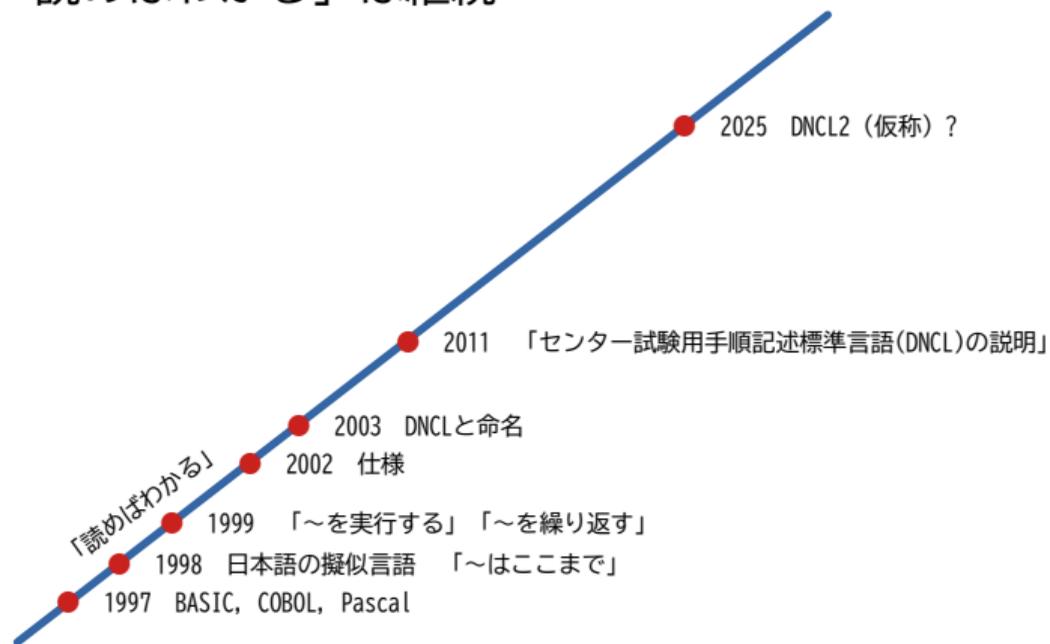
- ① はじめに
- ② 「情報関係基礎」プログラミング問題の変遷
- ③ DNCL, DNCL2, Python の比較
- ④ DNCL2 (仮称) の実行環境
- ⑤ 学習環境に関する考察
- ⑥ おわりに

2 「情報関係基礎」プログラミング問題の変遷

プログラミング問題の比較
(2005年まで必答, 2006年から表計算との選択)
(追試験は除く)

2.1 言語

「読めばわかる」は継続



2.2 題材

題材，構文項目は表 1 参照
題材のカブリがない!

パターン描画	Hit and Blow	配列のマージ
二分探索	金種計算	迷路（左手法）
文字列探索	小町算	じゃんけん
ライフゲーム	計算ブロック	ソート
素因数分解	漢数字に変換	得点集計・順位
分銅（符号付 3 進法）	営業時間最適化	太鼓の楽譜
ボードゲーム	荷物の箱詰め	三角形の個数
迷路（袋小路）	グループ分け	宝探しゲーム
すごろく	シーザー暗号	ドント方式

- ① はじめに
- ② 「情報関係基礎」プログラミング問題の変遷
- ③ DNCL, DNCL2, Python の比較
- ④ DNCL2 (仮称) の実行環境
- ⑤ 学習環境に関する考察
- ⑥ おわりに

3 DNCL, DNCL2, Python の比較

言語の仕様

DNCL 「センター試験用手順記述標準言語 (DNCL) の説明」

DNCL2 不明

試作問題・サンプル問題から推測

Python Version3 系

比較対象に Python を含めた理由

- 文科省「教員研修用教材」は Python 版が先行
 - 他言語版は Python 版の「翻訳」
- DNCL2 が Python の影響を受けているっぽい

3.1 文字列

文字列リテラルの書き方

DNCL

「 または ""

DNCL2

""

Python

"" または ''

3.2 出力

出力

DNCL

《値》を表示する
《値》を改行なしで表示する

DNCL2

表示する (《値》)

Python

print(《値》)

複数の値を表示する場合...

DNCL 「と」で連結

DNCL2 コンマ区切り

Python コンマ区切り

3.3 ブロック文

ブロック文 (例として分岐)

DNCL

```
もし《条件》ならば  
  | 《処理》  
を  
実行し、そうでなければ  
  | 《処理》  
を  
実行する
```

DNCL2

```
もし《条件》ならば：  
  ⊥ 《処理》  
そうでなければ：  
  ⊥ 《処理》
```

Python

```
if 《条件》：  
    《処理》  
else：  
    《処理》
```

3.4 演算子等の記号

代入, 比較, 四則演算

DNCL

代入 ←
 比較 =
 ≠
 ≦
 除算 /と÷を区別

DNCL2

代入 =
 比較 ??
 !=
 <=
 除算 ??

Python

代入 =
 比較 ==
 !=
 <=
 除算 /と//を区別

3.5 論理演算

論理演算

DNCL

かつ
または
でない

左から結合

DNCL2

and
or
not

結合については不明

Python

and
or
not

結合は明確なルールあり

DNCL は「読めばわかる」を目指すはず

3.6 後判定のループ

後判定のループ

DNCL

繰り返し,
| 《処理》
を, 《条件》 になるまで繰り返す

DNCL2

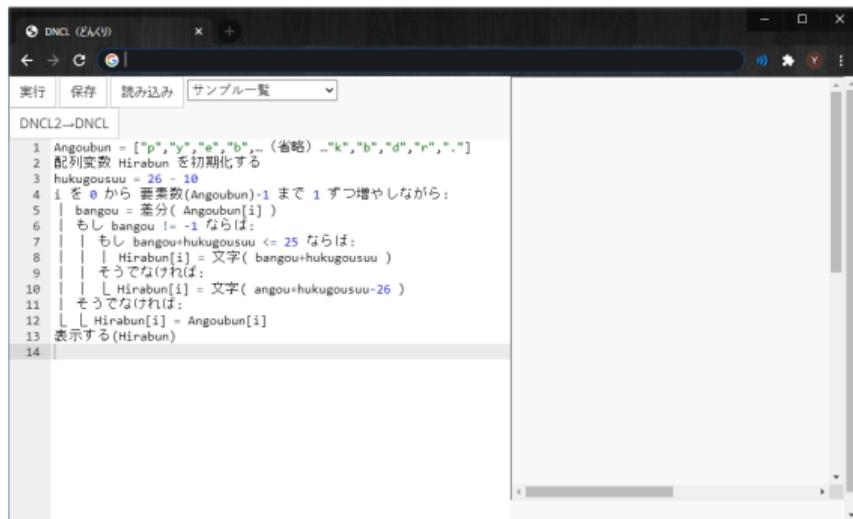
Python

- ① はじめに
- ② 「情報関係基礎」プログラミング問題の変遷
- ③ DNCL, DNCL2, Python の比較
- ④ **DNCL2 (仮称) の実行環境**
- ⑤ 学習環境に関する考察
- ⑥ おわりに

4 DNCL2（仮称）の実行環境

見たことがあるものを列挙

4.1 どんくり



```
DNCL2 (どんくり)
実行 | 保存 | 読み込み | サンプル一覧
DNCL2->DNCL
1 Angoubun = ["p","y","e","b",... (省略) ...,"k","b","d","e","."]
2 配列変数 Hirabun を初期化する
3 hukugousuu = 26 - 10
4 i を 0 から 変数(Angoubun)-1 まで 1 ずつ増やしながら:
5 | bangou = 変分( 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)
14
```

図 4 IDE の画面

(画像は参考文献 [9] より)

4.2 つちのこ

```

実行 ステップ 停止 [off] Python変換
1 Yakuasu = [0] * 101
2 i を 2 から 100 まで 1 ずつ増やしながら繰り返す:
3 もし Yakuasu[i] == 0 ならば:
4     j = i
5     } <= 100 の間繰り返す:
6     Yakuasu[j] = i
7     j = j + 1
8 k = 100
9 k > 1 の間繰り返す:
10 表示する(Yakuasu[k])
11 k = k / Yakuasu[k]
12

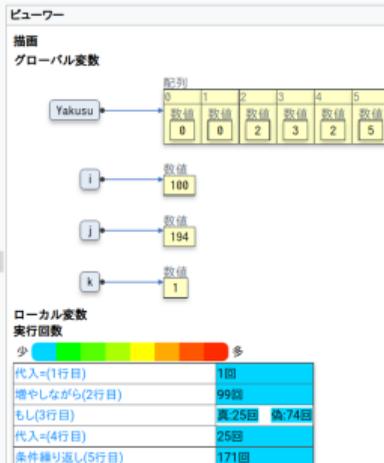
```

コンソール 実行待ち

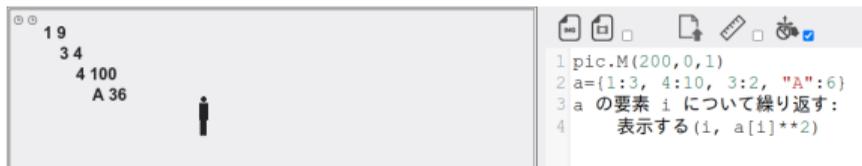
```

5
5
2
2

```



4.3 Picthon



The screenshot shows the Picthon environment. On the left is a terminal window with the following output:

```
19
 34
 4 100
   A 36
```

On the right is a code editor with the following code:

```
1 pic.M(200,0,1)
2 a={1:3, 4:10, 3:2, "A":6}
3 a の要素 i について繰り返す:
4   表示する (i, a[i]**2)
```

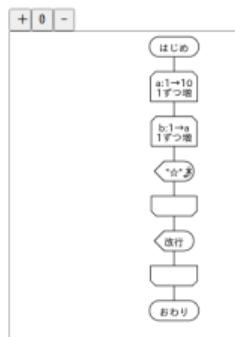
4.4 PyPEN

フローチャート
 コード→フローチャート
 コード→Python
 URL生成

Upload 選択されていません
 [Download](#)

```

1 aを1から10まで1ずつ増やしなが： ☆
2 ……bを1からaまで1ずつ増やしなが： ☆☆
3 ……"☆"を改行なしで表示する ☆☆☆
4 ……改行する ☆☆☆☆☆
5 . ☆☆☆☆☆☆
6 . ☆☆☆☆☆☆☆
  
```



- ① はじめに
- ② 「情報関係基礎」プログラミング問題の変遷
- ③ DNCL, DNCL2, Python の比較
- ④ DNCL2 (仮称) の実行環境
- ⑤ **学習環境に関する考察**
- ⑥ おわりに

5 学習環境に関する考察

Web ブラウザ上で実行できる

- インストール不要
- 方言（細部の違い）がある

各環境を比較

→PyPEN ではどのようにすべきかを考える

5.1 演算子記号

代入は「←」 → 「=」

比較（等値）は「=」 → 「==」

PyPEN も揃えるべきか（他環境は既にそうになっている）

- 揃えたい
- 既にある資料との互換性が...

でも揃えちゃいました

5.2 論理演算

「かつ」「または」「でない」 → 「and」「or」「not」

左から結合 → 不明

PyPEN も揃えるべきか（他環境は既に...）

- 揃えたい
- 既にある資料との互換性が...

でも揃えちゃいました

5.3 配列の初期化

[《値》, 《値》, ...] の構文

他にも

どんくり 1, 2, 3...100 で等差数列で初期化

つちのこ Python の range に似た区間オブジェクト

PyPEN 「《個数》 個の 《値》」

《配列》 * 《整数》 や 《配列》 + 《配列》 も先週実装

「配列 《変数名》 を 《値》 で初期化する」などの構文

↑ 要素数の指定がない...どうやって実装する？

どんくり (DNCL) の配列みたいな振る舞いをするクラスを作る？

5.4 比較演算子の結合

5>4>3 の扱い

C (5>4)>3 → 1>3 → 0

PHP エラー (結合させない)

Python 5>4 and 4>3 → True and True → True

どういう扱いをすべきか

Picthon Python 風

つちのこ C 風 (ただし True>3 となってエラー)

PyPEN PHP 風 → Python 風に変更

しかし Python 風に「慣れて」大丈夫？

他にこんなことする言語ってあるんだろうか...DNCL2 ではどうなるんだろう...

5.5 二次元配列の添字

DNCL `a[b,c]`

Python `a[b][c]`

DNCL2 不明

5.6 elif

DNCL elif にあたる構文あり

DNCL2 不明

つちのこ 「そうでなくもし《条件2》ならば：」

PyPEN フローチャートにしづらくて未実装

フローチャートを諦めるか？

- 「俯瞰するのにほしい」という生徒もいる
- 簡単なものだけ対応すればいいのかも
 - 自作関数・手続きにさえ対応してないし

5.7 for in

Python の「for 《変数》 in 《配列》」

Picthon 《配列》の要素《変数》について繰り返す

つちのこ 《配列》の要素《変数》について繰り返す

PyPEN 未実装 (実装したい)

《配列》の要素《変数》について繰り返す (先週実装しました)

語順が逆なのが玉に瑕

5.8 型

Python のリスト, タプル, 辞書, 集合
Picthon や「つちのこ」では実装

a, b=b, a みたいなこともできる
↑同様のことができる言語はいくつもある
($5 > 4 > 3$ みたいな心配はいらないだろう)

5.9 in

Python の「《値》 in 《配列》」

PyPEN 《配列》の中に《値》

《値》 in 《配列》 (先週実装しました)

Picthon 《値》 in 《配列》

語順を取るか，日本語を取るか...

5.10 組み込み関数

「切り捨て」「差分」「要素数」などの組み込み関数がある前提

↑問題によって違う...あらかじめ用意することはできない

↑問題に合わせてコードを書いて実装すればいい

sum とか sort みたいな関数を予め作っておくべきか

- それらの処理を「自分で書く」必要はあるか
- そういうものは「あるだろう」と感じる嗅覚が必要なのでは?

5.11 その他

- global 文は必要か
- 三項演算子は？

- ① はじめに
- ② 「情報関係基礎」プログラミング問題の変遷
- ③ DNCL, DNCL2, Python の比較
- ④ DNCL2 (仮称) の実行環境
- ⑤ 学習環境に関する考察
- ⑥ おわりに

6 おわりに

試作問題・サンプル問題を元に DNCL・DNCL2 について考察した
DNCL2 の学習環境はいくつもある
→ 紙の上の勉強だけでは終わらない

生徒が触れる言語・環境はさまざま
→ DNCL2 や Python に縛られてはいけない

DNCL・DNCL2 の根幹は「読めばわかる」
その思想を見失わないように PyPEN の実装を進めたい