

センター試験のプログラミング言語を用いた プログラミング教育

中西渉

watayan@meigaku.ac.jp
名古屋高等学校

2019年12月26日

1 準備

Web ブラウザを起動して

<https://watayan.net/misc/ipsj2019/>
にアクセスしてください。

DNCL について

センター試験「情報関係基礎」で使われるプログラミング言語
 もともとは擬似言語
 → PEN という実行環境

The screenshot shows the PEN (Pseudo-Extended Notation) environment. The main window displays the following code:

```

1:/* 1回目の乱数より2回目の乱数が多い */
2:/* が少ないかを当てるゲーム */
3:
4:整数 i
5:整数 rand
6:整数 ans
7:
8:i ← 0
9:
10:i = 3 の間、
11:  rand ← random(12) + 1
12:
13:  「ハイ&ローゲーム」を表示する
14:  rand と 「よりハイ?ロー?」を表示する
15:
16:  「ハイ:0/ロー:1/終了:3」を入力を表示する
17:  i ← input()
18:
19:  もし i = 0 または i = 1 ならば
20:    ans ← random(12) + 1
21:    ans と 「出ました」を表示する
22:    もし ans > rand かつ i = 0 ならば
23:      「アタリ」を表示する
24:      i を実行する
25:    もし ans < rand かつ i = 0 ならば
  
```

The console window on the right shows the following output:

```

実行画面 | 履歴
-----|-----
ランダムに生成した |
ハイ:0/ロー:1/終了:3 |
を入力 |
1 |
12が出ました |
ハズレ |
-+-+-+ |
ハイ&ローゲーム |
9よりハイ?ロー? |
ハイ:0/ロー:1/終了:3 |
を入力 |
  
```

At the bottom, there is a table for variable declarations:

型	変数名	値
整数	i	1
整数	rand	0
整数	ans	12

Below the code, there is a table for program flow control:

プログラム入力受付ボタン		一の値、繰り返す	
もし	もし〜そうでなければ	そうでなくもし〜	一の値、繰り返す
繰り返しながら繰り返す	繰り返しながら繰り返す	入力	実行停止力
実行	実行	実行	実行

PEN にフローチャートを追加 → PenFlowchart

PenFlowchart ver2.17

ファイル ヘルプ

パーツ

代入 各種処理 入力 出力 条件 繰り返し 終了

変数 a,b
関数 関数

フローチャート

戻次
繰り返し
戻す

```

    graph TD
      Start([はじめ]) --> A[a ← 0]
      A --> B[b ← random(8)+1]
      B --> C{「1から9の数字を当ててください」}
      C --> D[aを入力]
      D --> E{a > b}
      E -- Y --> F{「大きい」}
      E -- N --> G{「小さい」}
      E -- Y --> H[a ← b]
      E -- N --> I[a ← b]
      F --> J[a ← b]
      G --> J[a ← b]
      H --> J[a ← b]
      I --> J[a ← b]
      J --> K[a ← b  
になるまで]
      K --> C
  
```

1: 整数 a,b
2: a ← 0
3: b ← random(8)+1
4: 「1から9の数字を当ててください」を表示する
5: 繰り返し
6: | a を入力する
7: | もし a > b ならば
8: | | 「大きい」を表示する
9: | を実行し、そうでなければ
10: | | もし a < b ならば
11: | | | 「小さい」を表示する
12: | | | を実行する
13: | を実行する
14: | を、a = b になるまで実行する
15: 「あたり」を表示する
16:

実行履歴 (道) 実行終了

コンソール表示

1から9の数字を当ててください
5
大きい
1
小さい
3
小さい
4
あたり

変数表示画面

型	変数名	値
整数	a	4
整数	b	4

プログラム入力支援ボタン

もし	もし〜そでなければ	〜の繰り返し	〜になるまで実行する
繰り返しの繰り返し	繰り返しの繰り返し	入力	実行された 代入
関数	実行	文字列	数値関数
		ファイル名	文字列関数
			数学関数
			;

PenFlowchart を Web ブラウザ上で → WaPEN (Web-aided PEN)

新規 実行 ステップ実行 リセット フローチャート コード-フローチャート

Upload ファイルを選択 選択されていません Download ファイル名:

問題選択 採点

```

1 整数 a,b
2 a=0
3 b=random(8)+1
4 「1から9の数字を当ててください」を表示する
5 繰り返し,
6 | aを入力する
7 | aを表示する
8 | もしa>bならば
9 | | 「大きい」を表示する
10 | | を実行し, そうでなければ
11 | | もしa<bならば
12 | | | 「小さい」を表示する
13 | | | を実行する
14 | | を実行する
15 | を, a=bになるまで実行する
16 「あたり」を表示する

```

+ -

整数 実数 文字列

入力 出力 改行無出力 代入

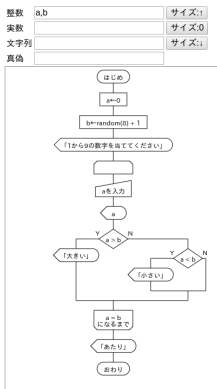
もし もし~そうでなければ

~の間, 繰り返す ~になるまで実行する 増やしながらか減らしながら

関数 手続き

サンプル1 サンプル2 サンプル3 サンプル4 サンプル5 サンプル6 サンプル7 サンプル8

[マニュアル](#)



WaPEN のいいところ

- インストール不要 (Java, Flash 不要)
- 自分のサーバに置けば...
 - サンプルプログラムを変更できる
 - 細かい手直しができる

「サンプル 1」～「サンプル 8」を呼び出して実行してみてください
（「サンプル 9」「サンプル 10」は後で使います）

プログラムを読みながら、何をしているか読み取ってください

2 xDNCL の文法

プリントを参照してください

3 練習

ガチャのシミュレーションを作ってみよう

- 景品は 10 種類
- すべて等確率
- 全種類揃うまでに何回引くか

WaPEN の準備

「新規」ボタンで空っぽにする
コードでもフローチャートでも好きな方で（できればコードで）

注意

ワープロを打つように 1 文字ずつ打ち込むのはダメ

(01)

整数 kaisu,ransu,keihin[10],nuke,n

変数の宣言

コード 「整数」ボタンを押して、変数を書き並べる

フローチャート 「整数」の空欄に、変数を書き並べる

(02)~(03)

```
kaisu ← 0
```

```
keihin ← [0,0,0,0,0,0,0,0,0,0]
```

代入

コード 「代入」ボタン→変数と値を書き込む

フローチャート 縦棒で右クリック→「代入」を選ぶ

代入記号で右クリック→「編集」→変数と値を書き込む

(03) の 0 は 10 個です。

(04)~(15)

繰り返し、

|

を、`nuke=0` になるまで実行する

繰り返し

コード 「~になるまで実行する」 ボタン

フローチャート 「ループ」 → 「後条件」

文字下げの「|」は（だいたい）自動的に挿入される

(05)~(09)

```
kaisu ← kaisu+1  
ransu ← random(9)+1  
keihin[ransu] ← keihin[ransu]+1  
kaisu と keihin を表示する  
nuke ← 0
```

代入と出力（繰り返しの中で）

コード 「代入」ボタン, 「出力」ボタン
Enter で新しい行

フローチャート 「代入」, 「出力」

(10)~(14)

n を 1 から 10 まで 1 ずつ増やしながら、

|

を繰り返す

(変数を増やしながら) 繰り返し

コード 「増やしながら」 ボタン

フローチャート 「ループ」 → 「増やしながら」

(11)~(13)

もし $keihin[n]=0$ ならば

|

を実行する

分岐

コード 「もし」 ボタン

フローチャート 「分岐」

(12)

nuke ← 1

代入

※どの繰り返しの中でどの繰り返しの外であるかに注意

出来上がったら...

実行して確認

うまくいかなければ修正

改造案 (例)

100 回の平均を求める

レアを設定する

参考

n 種類の景品のガチャをコンプリートするための回数の期待値は

$$n \left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \right)$$

$n = 10$ のときの期待値は 29.28...

プログラムを見直してみる

- (01) 整数 kaisu,ransu,keihin[10],nuke,n
- (02) kaisu ← 0
- (03) keihin ← [0,0,0,0,0,0,0,0,0,0]
- (04) 繰り返し,
- (05) | kaisu ← kaisu+1
- (06) | ransu ← random(9)+1
- (07) | keihin[ransu] ← keihin[ransu]+1
- (08) | kaisu と keihin を表示する
- (09) | nuke ← 0
- (10) | n を 1 から 10 まで 1 ずつ増やししながら,
- (11) | | もし keihin[n]=0 ならば
- (12) | | | nuke ← 1
- (13) | | を実行する
- (14) | を繰り返す
- (15) を, nuke=0 になるまで実行する

4 センター試験の問題を解いてみる

2019年度センター試験「情報関係基礎」第3問
これから解いてみよう

まず 30~31 ページの振り分け方法をよく読んでください

方法 1

- 最初に埋まる 2 グループは 3 人
- 残りは 2 人

くじの順位	1	2	3	4	5	6	7	8	9	10
生徒の名前	A	B	C	D	E	F	G	H	I	J
第 1 希望	春	秋	春	春	春	夏	春	秋	春	春
第 2 希望	秋	春	秋	秋	夏	春	夏	春	秋	夏
第 3 希望	夏	冬	夏	冬	冬	秋	秋	夏	冬	秋
第 4 希望	冬	夏	冬	夏	秋	冬	冬	冬	夏	冬

季節	決まった人数
春	
夏	
秋	
冬	

33 ページのプログラムを作ってみよう 「サンプル 9」の《ケ》～《ス》

実行して正しい結果が得られれば（たぶん）OK
問題文にヒントがある（変数の意味は大事）

- `teiin` はグループ人数の上限
基本的には `syo`
- `g` は上限に達したグループ数
`amari` 個のグループは定員を 1 増やす
→ $g < amari$ の間は `teiin` を `syo+1` にする
- `Gnizu` は各グループに振り分けられた人数

答え合わせ

《ケ》 g<amari

《コ》 Gnizu[koho]

《サ》 Gnizu[koho]

《シ》 owari=1

《ス》 Gnizu[i]

方法 2

- 第 1 希望の多い春と秋は 3 人
- 残りは 2 人

くじの順位	1	2	3	4	5	6	7	8	9	10
生徒の名前	A	B	C	D	E	F	G	H	I	J
第 1 希望	春	秋	春	春	春	夏	春	秋	春	春
第 2 希望	秋	春	秋	秋	夏	春	夏	春	秋	夏
第 3 希望	夏	冬	夏	冬	冬	秋	秋	夏	冬	秋
第 4 希望	冬	夏	冬	夏	秋	冬	冬	冬	夏	冬

季節	定員	決まった人数
春	3	
夏	2	
秋	3	
冬	2	

35 ページのプログラムを作ってみよう
「サンプル 10」の《セ》～《チ》
(《コ》《シ》《ス》は 33 ページと同じ)

実行して正しい結果が得られれば (たぶん) OK

- Kibosu は各グループの第 1 希望の人数
- Gteiin は各グループの上限
→ Kibosu をカウントして、それを用いて Gteiin を決める
 - Kibosu が多い方から amari 個のグループは Gteiin を 1 増やす
 - syo+1 が選択肢にないことに注意

答え合わせ

《セ》 Kibosu[g]+1

《ソ》 Kibosu[j]

《タ》 syo

《チ》 Gteiin[g]+1

《コ》 Gninzu[koho]

《シ》 owari=1

《ス》 Gninzu[i]

最後の解答欄《ツテ》の答えは？

図2の(20)はサンプル10では41行

- 回数を数えるための変数 `kaisu` を追加
- プログラムの最初の方で `kaisu` に 0 を代入しておく
- (↑これですれた) 42行の直後に `kaisu` を 1 増やす
- 最後に `kaisu` を表示する

5 考察

変数の意味は超重要
手作業でやるとしたらどうするかを考える

たとえば方法2では

- ① 各グループの第1希望数 → Kibosu
- ② 各グループの定員 Gteiin を syo で初期化
- ③ 「Kibosu の多い順に Gteiin を 1 増やす」を amari 回行う

この3つがプログラムの3つのループ

6 おわりに

「情報」の教科書で使われている言語... JavaScript, Excel VBA

でも, DNCL もいいものです

- 英語の壁を避けられる
- 他の手続き型言語への橋渡し
- 筆記試験がやりやすい
- 教科書では取り上げられてないけど...
 - テキストなんて自分で作ればいい
 - 「基本」はだいたい同じ

PyPEN なんてものも作ってみました

流行りの Python っぽい感じ

Python への「翻訳」もできる

新規 実行 ステップ実行 リセット フローチャート コードフローチャート コード→Python

Upload ファイルを選択 選択されていません Download ファイル名: _____

問題選択 採点

```

1 a ← 0
2 b ← random(8)+1
3 「1から9の数字を出ててください」を表示する
4 a!←bの間繰り返す:
5   aに整数を入力する
6   aを表示する
7   もしa>bならば:
8     「大きい」を表示する
9   そうでなければ:
10    もしa<bならば:
11      「小さい」を表示する
12    「あたり」を表示する
  
```

+ 0 -

入力 (整数) 入力 (実数) 入力 (文字列) 入力 (真偽) 出力 改行無出力 代入

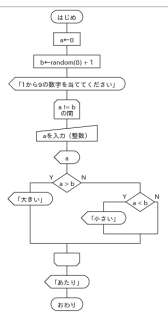
もし もし〜そうでなければ

〜の間, 繰り返し返す 増やししながら 減らしながら 繰り返しを抜ける

関数 手続き

サンプル1 サンプル2 サンプル3 サンプル4 サンプル5 サンプル6 サンプル7 サンプル8

[マニュアル](#)



+ 0 -

「基本」って何だろう

- 処理を分解して考える
 - 処理しやすいように読み替える
- どこまで分解したらいいか
 - 環境に合わせた「公式」まで
 - 理解している「構文」まで

→ どんな言語・環境でも通用する

→ 数学の問題でも同じことをしてる

→ どんな勉強でも通用する

本スライドおよび配布資料（ただしセンター試験の問題・解答は除く）の著作権者は中西渉であり，クリエイティブ・コモンズ 表示 4.0 国際で公開する

