

指導資料

情報教育 第160号

 鹿児島県総合教育センター
令和4年10月発行

対象
校種 高等学校
特別支援学校



プログラミングによる問題解決を図る授業設計 —大学入学共通テスト「情報」サンプル問題を通して—

- ◆ アルゴリズム*1をプログラムで表現し、更に具体的な状況設定に応じてプログラムを修正することを通して問題解決を図る学習場面を設定することが大事である。
- ◆ プログラミング言語Pythonによるシミュレーション及び課題解決を体験し、情報活用能力及び問題発見・解決能力の育成を図るための教材として、プログラミングに関する大学入学共通テスト「情報」サンプル問題を活用するとよい。

#情報社会の問題解決 #「Python」によるシミュレーション #共通テスト「情報」

1 はじめに

ゲートもロック板もない有料駐車場が増加している。入庫の際にセンサーカメラとAI*2技術により車両ナンバーを自動認識し、出庫の際は車両ナンバーを精算機に入力して精算する。情報端末があればクラウド*3を通じてオンライン決済や駐車状況の取得も行える。利用者の利便性を高めつつ、設備と労働力を削減できる。情報技術の進展を感じるが、実は、高等学校までの情報教育で習得する知識・技能を組み合わせることで十分に実現可能なシステムであると考えられる。

小・中学校のプログラミング教育で活用されている「Scratch*4」の拡張機能「ML2Scratch」を用いることで、AI（機械学習*5）による画像認識が容易にできる（図1）など、プログラミングの開発（学習）環境は充実してきている。AIにより解決できる身近な課題について考え、プログラムを作成する授業を実践している事例もある。高等学校では共通必修科目「情報Ⅰ」（令和4年度開始）等においてプロ



図1 Scratchの画像認識プログラム

※<https://champierre.github.io/ml2scratch/>で拡張機能を追加

プログラミングを全ての高校生が学ぶ。その際、テクノロジーの知識を詰め込むことより、日常や社会の課題を実践的に解決するための考え方を学び、技術の入り口を体験することが大切である。令和7年度大学入学共通テストから「情報」が出題教科として追加され、「情報Ⅰ」の内容が出題範囲となる。そこで今回は、大学入試センターから示されたサンプル問題を教材に、AIやWebサービスの開発に使われるプログラミング言語「Python」によるシミュレーション及び課題解決について紹介する。

2 共通テスト手順記述標準言語（DNCL）

大学入学共通テスト「情報」サンプル問題のプログラミングに関する問題においては、

*1) 問題を解決するための手順や計算方法 *2) 人工知能の略称。人間が行う知的活動をコンピュータプログラムとして実現すること *3) サーバやソフトウェアなどコンピュータに必要な機能を、ネットワークを経由して、サービスの形で提供する利用形態 *4) マサチューセッツ工科大学が無償で公開しているビジュアルプログラミング言語。画面上のブロックをつなぎ合わせてプログラムを作成 *5) データから機械が自動で学習し、データの背景にあるルールやパターンを発見する方法

共通テスト手順記述標準言語 (DNCL) を使用している。DNCLについて、独立行政法人大学入試センターは次のように説明している。

問題の中で使用するプログラム言語は、高等学校の授業で多様なプログラム言語が利用される可能性があることから、公平性を鑑みて、大学入試センター独自の日本語表記の疑似言語としている。これは、高等学校の授業で何らかのプログラム言語を用いて実習した生徒であれば容易に理解できるものである。

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・科目サンプル問題「情報」ねらいから一部抜粋。下線は筆者による。

これまで、職業教育を主とする専門学科及び総合学科で情報に関する科目を履修している共通テスト受験者が数学②の科目として選択できる「情報関係基礎」において、DNCLは用いられてきた。「情報」サンプル問題のDNCL (図2) と「情報関係基礎」のDNCLとの違いは、

```

(01) Tomei = ["A 党", "B 党", "C 党", "D 党"]
(02) Tokuhyo = [1200, 660, 1440, 180]
(03) Tosen = [0, 0, 0, 0]
(04) tosenkei = 0
(05) giseki = 6
(06) m を 0 から [ア] まで1 ずつ増やしながら繰り返す
(07) Hikaku[m] = Tokuhyo[m]
(08) [セ] < giseki の間繰り返す:
(09)     max = 0
(10)     i を 0 から [ア] まで1 ずつ増やしながら繰り返す:
(11)         もし max < Hikaku[i] ならば:
(12)             [ソ]
(13)             maxi = i
(14)     Tosen[maxi] = Tosen[maxi] + 1
(15)     tosenkei = tosenkei + 1
(16)     Hikaku[maxi] = 切り捨て([タ]/[チ])
(17) k を 0 から [ア] まで1 ずつ増やしながら繰り返す:
(18)     [リ] 表示する(Tomei[k], ":", Tosen[k], "名")
    
```

図2 第2問 問3 (大学入学共通テスト「情報」サンプル問題を基に作成)

「情報」サンプル問題のDNCLがPythonを模して作成されていることである。

DNCLで表記されたプログラムは、実際のシステム開発等において使用できるものではないが、DNCLに対応したプログラミング学習環境はいくつかあり、その一つ「PyPEN*6」(図3)ではDNCLで記述したプログラムを実行したり、プログラムをDNCLからPythonに翻訳したり、フローチャートに変換したりすることができる。

しかし、共通テストで出題される形式であるということからDNCLのみを学習することは望ましくない。「情報I」の教科書で採用されている主なプログラミング言語はPython, JavaScript*7, VBA*8, Scratch等である。学校の実態に応じてプログラミング言語を選択し、アルゴリズムを表現する力、プログラムを作成する力、プログラムの不具合を修正したり、効率化したりする力を養うとともに、プログラミングによって問題を解決する力を養うように実習に取り組んでほしい。日本語表記でイメージしやすいDNCLで学習してから他の言語に移行してもよいだろう。他の言語を学習してからPyPEN等を活用すると、DNCLで表記されたコードと学習した言語のコードの対応が容易に把握できるだろう。

アルゴリズムの基本はどの言語でも同じであることから、共通テストの出題内容に言語選択が左右されることはないと考えが、小・中学校において、Scratch等のビジュアルプロ

グラミング言語を中心に学習をしてきた生徒は、DNCLへの対応に時間を要する場合も考えられるので留意してほしい。

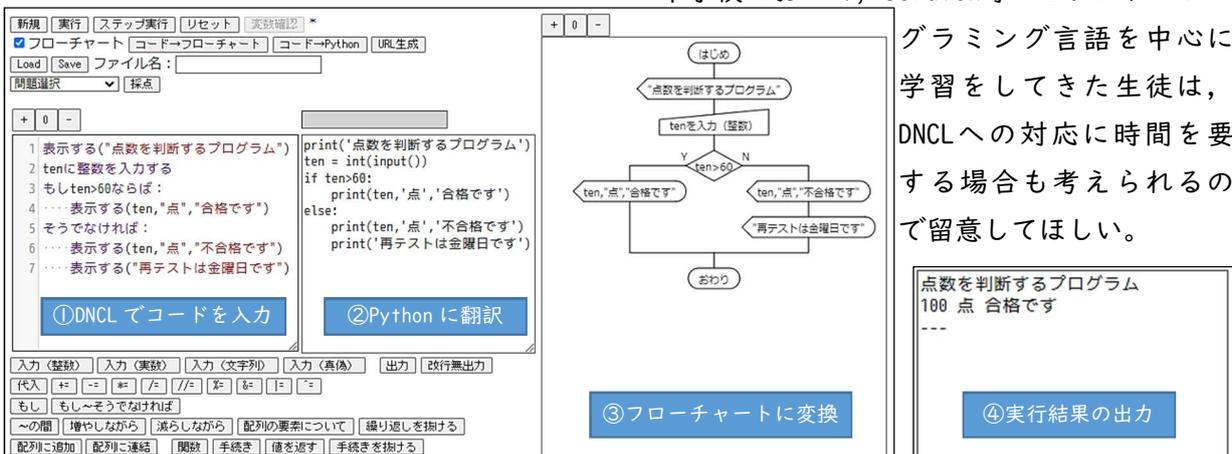


図3 DNCLプログラミング学習環境「PyPEN」の動作画面

3 大学入学共通テスト「情報」サンプル問題のプログラムによる演習

(1) Pythonの学習環境と基本的なコード

Pythonの学習環境は多数あるが、Google Colaboratory*9を紹介する。インストールが不要かつ、すぐにPythonや機械学習・深層学習の環境を整えることができる無料のクラウドIDE（ブラウザ上で動作する統合開発環境）である。本県の公立学校の全ての教員及び児童生徒に付与されている県域アカウント（～@kago.ed.jp）にログインすることで使用でき、作成したプログラムはGoogleドライブに保存されることから容易に他者との共有が可能である。そのため、協働でのプログラム作成や教員が準備した演習データの提供をすることができる。また、基本的なプログラムコードについては、指導資料情報教育第153号「意外と簡単！『Python』－高等学校「情報Ⅰ」でのプログラミングー」を参照してほしい。

(2) サンプル問題のプログラムの構成要素

サンプル問題で出題されたDNCLのプログラ

(3) サンプル問題（第2問）をPythonでコーディング*10

問題の会話文と図表（DNCLプログラムは提示しない）をもとにアルゴリズムやコードを考えさせ、生徒の実態に応じてフローチャートやプログラムの一部を提示してコーディングの支援を行う。

「比例代表選挙の議席配分をするプログラム」

※ 問題の詳細は大学入試センターWebページを参照
https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r71kou.html

① 比例代表選挙での各党への議席の配分数を、得票数に比例して配分した議席数を求めるプログラムを作成する。

【フローチャート】

```

はじめ
Tomei=["A党","B党","C党","D党"]
Tokuhyo=[1200,660,1440,180]
sousuu=0
giseki=6
m:0→3
1ずつ増
sousuu=sousuu+Tokuhyo[m]
kizyunsuu=sousuu/giseki
"基準得票数:",kizyunsuu
"比例配分"
m:0→3
1ずつ増
Tomei[m],":",Tokuhyo[m]/kizyunsuu
おわり
                    
```

【Python プログラム】

```

Tomei = ["A党", "B党", "C党", "D党"]
Tokuhyo = [1200, 660, 1440, 180]
sousuu = 0
giseki = 6
for m in range(0, 3+1, 1):
    sousuu = sousuu + Tokuhyo[m]
kizyunsuu = sousuu / giseki
print("基準得票数:", kizyunsuu)
print("比例配分")
for m in range(0, 3+1, 1):
    print(Tomei[m], ":", Tokuhyo[m] / kizyunsuu)
                    
```

【実行結果】

```

基準得票数: 580.0
比例配分
A党: 2.0689655172413794
B党: 1.1379310344827587
C党: 2.4827586206896552
D党: 0.3103448275862069
                    
```

【プログラムの構成要素】

- 変数 sousuu
- 代入 giseki = 6
- インクリメント tosenkei = tosenkei + 1
- メッセージ 表示する("比例配分")
- 異なるデータ型の結合 表示する("基準得票数:", kizyunsuu)
- 算術演算子 kizyunsuu = sousuu / giseki
- 比較演算子 Koho[i] >= Tosen[i] + 1
- 論理演算子 解答群 ① and ① or ② not
- 条件分岐(if文) もし max < Hikaku[i]ならば:
- 繰り返し(for文)
 - m を 0 から [ア] まで 1 ずつ増やしながらか繰り返す:
- 繰り返し(while文) [セ] < giseki の間繰り返す:
- 入れ子(ネスト)
 - i を 0 から [ア] まで 1 ずつ増やしながらか繰り返す:
 - もし max < Hikaku[i]ならば:
 - [省略]
- 配列/リスト Tokuhyo = [1200, 660, 1440, 180]
- 添字(インデックス) Hikaku[m] = Tokuhyo[m]
- 関数 Hikaku[maxi] = 切り捨て([タ] / [チ])
- 最大値の探索
 - もし max < Hikaku[i]ならば:
 - [ソ]
 - maxi = i

【問われる力】

- ・ 基本的なアルゴリズムとプログラムコードの理解
- ・ アルゴリズムをプログラムで表現する力
- ・ プログラムからアルゴリズムを読み取る力
- ・ 状況に応じてプログラムを修正し、問題解決に向けて考察する力

図4 サンプル問題のプログラムの構成要素と問われる力

*6) ブラウザ上で動作するDNCLプログラミング学習環境 (<https://watayan.net/prog/PyPEN/>) *7) Webサイトやシステムの開発に使われているインタープリタ型プログラミング言語 *8) Visual Basic for Applicationsの略でマクロ（登録した作業を自動的に実行する機能）を作成するためのプログラミング言語 *9) <https://colab.research.google.com/?hl=ja> *10) プログラミング言語を用いてソースコードを作成すること

② ドント方式で議席を配分するアルゴリズムを理解するためトレースする。

【ドント方式】日本では各政党の得票数を1, 2, 3, ...と整数で割った商の大きい順に定められた議席を配分する。

各政党の得票数と整数で割った商

| | A党 | B党 | C党 | D党 |
|--------|-------|------|-------|-----|
| 得票数 | 1200 | 660 | 1440 | 180 |
| 1で割った商 | ②1200 | ④660 | ①1440 | 180 |
| 2で割った商 | ⑤600 | 330 | ③720 | 90 |
| 3で割った商 | 400 | 220 | ⑥480 | 60 |
| 4で割った商 | 300 | 165 | 360 | 45 |

【トレース表の作成】

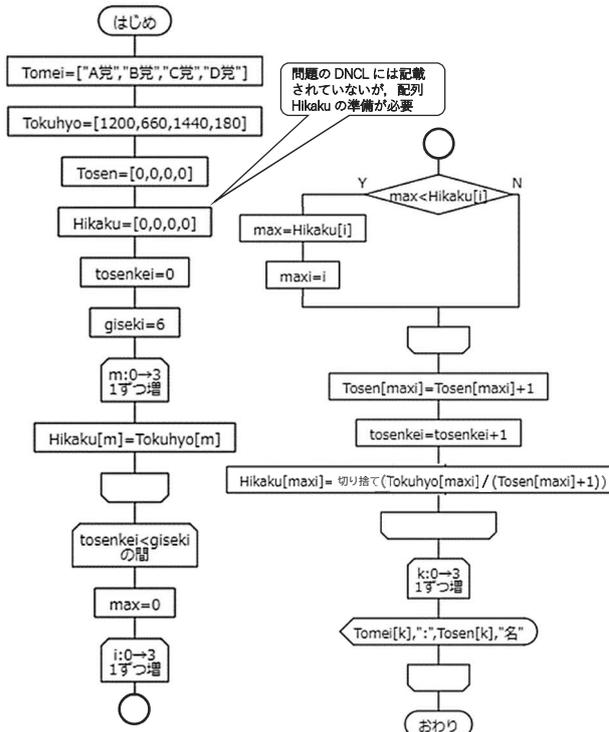
| 手順 | 配列 Hikaku の変化 | | | | 配列 Tosen の変化 | | | | | |
|------------|---------------|-----|------|-----|--------------|---|------|---|---|---|
| | i | 0 | 1 | 2 | 3 | i | 0 | 1 | 2 | 3 |
| 手順1終了時 | 1200 | 660 | 1440 | 180 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1回目の手順3終了時 | 1200 | 660 | 720 | 180 | 0 | 0 | 1 | 0 | 0 | 0 |
| 2回目の手順3終了時 | 600 | 660 | ⑤720 | 180 | 1 | 0 | ②600 | 0 | 0 | 0 |
| 3回目の手順3終了時 | 600 | 660 | ③720 | 180 | 1 | 0 | ④600 | 0 | 0 | 0 |
| 4回目の手順3終了時 | 600 | 330 | ⑥480 | 180 | 1 | 1 | ⑤600 | 0 | 0 | 0 |
| 5回目の手順3終了時 | 400 | 330 | ①480 | 180 | 2 | 1 | ⑥600 | 0 | 0 | 0 |
| 6回目の手順3終了時 | 400 | 330 | ②480 | 180 | 2 | 1 | ③600 | 0 | 0 | 0 |

配列 Hikaku と配列 Tosen の変化

- 手順① 配列 Tokuhyo の各要素の値を配列 Hikaku の初期値として格納する。
 手順② 配列 Hikaku の要素の中で最大の値を調べ、その添字 maxi に対応する配列 Tosen[maxi]に1を加える。
 手順③ Tokuhyo[maxi]を Tosen[maxi]+1で割った商を Hikaku[maxi]に格納する。
 手順④ 手順2と手順3を当選者数の合計が議席数の6になるまで繰り返す。
 手順⑤ 各政党の党名 (配列 Tomei) とその当選者数 (配列 Tosen) を順に表示する。

③ ②のアルゴリズムをもとにプログラムを作成する。

【フローチャート】



【Python プログラム】

```
Tomei = ["A党", "B党", "C党", "D党"]
Tokuhyo = [1200, 660, 1440, 180]
Tosen = [0, 0, 0, 0]
Hikaku = [0, 0, 0, 0]
tosenkei = 0
giseki = 6
for m in range(0, 3+1, 1):
    Hikaku[m] = Tokuhyo[m]
    while tosenkei < giseki:
        max = 0
        for i in range(0, 3+1, 1):
            if max < Hikaku[i]:
                max = Hikaku[i]
                maxi = i
        Tosen[maxi] = Tosen[maxi] + 1
        tosenkei = tosenkei + 1
        Hikaku[maxi] = int(Tokuhyo[maxi] / (Tosen[maxi] + 1))
    for k in range(0, 3+1, 1):
        print(Tomei[k], ":", Tosen[k], "名")
```

【実行結果】

A党 : 2名
 B党 : 1名
 C党 : 3名
 D党 : 0名

【プログラムの問題点と修正】

・ 政党によっては候補者が足りない場合もある。
 ⇒ 各政党の候補者数を格納する配列 Koho を追加、コード修正

```
Koho = [5, 4, 2, 3]
候補者数を格納する配列
(12行目) max < Hikaku[i] and Koho[i] >= Tosen[i] + 1
候補者が不足していないという条件を追加
```

図5 比例代表選挙の議席数を算出するプログラムを作成する学習過程

4 おわりに

令和4年度大学入学共通テスト問題作成方針^{*11)}に「授業において生徒が学習する場面や、社会生活や日常生活の中から課題を発見し解決方法を構想する場面、資料やデータ等を基に考察する場面など、学習の過程を意識した問題の場面設定を重視する」とある。大学入学共通テスト「情報」サンプル問題についても、その方針に沿った内容になっており、情報技術を活用して問題の発見・解決を行う学習活動の場面が設定されている。サンプル問題と、令和2年11月24日に大学入試センターから示

された試作問題^{*12)}では次の内容が出題された。

- ・ サッカーワールドカップのデータ分析
- ・ 東日本大震災から見える情報社会の課題
- ・ 交通渋滞シミュレーション
- ・ プログラミングによる暗号解読
- ・ 二要素認証によるセキュリティ強化
- ・ ネットワークの不具合の原因究明
- ・ Webアクセスログの分析 等

検討を重ねた共通テストの問題は生徒の興味・関心を高め、資質・能力を育む教材となる。実際にプログラムの作成・実行・修正を繰り返し、シミュレーションやデータ分析を重ねながら問題解決に向けて考察する学習場面の設定に活用されることを期待する。

- 参考文献 -
 ○ 文部科学省「高等学校学習指導要領(平成30年告示)解説 情報編」平成30年7月(情報教育研修課 青木 誠)
 ※ 本資料は、UDフォントを使用しています。

*11) 大学入試センターWebページ https://www.dnc.ac.jp/kyotsu/kako_shiken_jouhou/r4/r4.html *12) <https://www.ipsj.or.jp/education/9foeag000012a50-att/sanko2.pdf>