

研究論文

# 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

井手 広康<sup>1,a)</sup>

受付日 2021年9月20日, 再受付日 2022年1月5日/2022年2月21日,  
採録日 2022年6月25日

**概要:** 令和4年度より高等学校では「情報I」が始まり, プログラミングの分野では, 基本的に教科書に記載されたプログラミング言語を用いて授業を行うことになる. また, 令和7年度からの大学入学共通テストに「情報」が導入されることから, 授業で扱うプログラミング言語と共通テスト「情報」に出題されるDNCLとの違いを意識しながらプログラミング教育を実施する必要がある. そこで本研究では, DNCLと「情報I」の教科書に使用されているPython, JavaScript, VBA, Scratchとの相違点について考察するとともに, 試作問題/サンプル問題と教科書に記載されているプログラミング分野の内容について比較を行った. その結果, DNCLと各プログラミング言語には, 変数の宣言, ブロックの区切り, オブジェクトの結合, 配列/リストの作成と操作, for文の表記などに大きな違いがあること, すべての教科書が試作問題/サンプル問題に出題された内容を網羅できているわけではないことが分かった.

**キーワード:** 情報I, 教科書, プログラミング, 大学入学共通テスト, DNCL

## Comparison and Consideration of Programming Fields in Information Study I Textbooks

HIROYASU IDE<sup>1,a)</sup>

Received: September 20, 2021, Revised: January 5, 2022/February 21, 2022,  
Accepted: June 25, 2022

**Abstract:** A subject “Information Study I” will start in high school from April 2022, and in the field of programming, classes will be basically conducted using the programming language described in the textbook. In addition, it needs that programming education will be conducted while being aware of the difference between the programming language used in class and DNCL since “Information Study” will be introduced into the Common Test for University Admission from January 2025. Therefore, in this study, we will consider the differences between DNCL and Python, JavaScript, VBA, and Scratch used in the textbook of “Information Study I”, and compared the prototype test/sample test with the contents of the programming field described in textbooks. As a result, it turned out that there are major differences between DNCL and each programming language in variable declaration, block delimiter, object concatenation, array/list creation and operation, for statement notation, etc., and not all textbooks can cover the contents of the prototype test and sample test.

**Keywords:** information study I, textbook, programming, the common test for university admissions, DNCL

### 1. はじめに

平成30年告示高等学校学習指導要領 [1] (以下, 「新学習指導要領」と表記) において, 教科「情報」は選択必修科目であった「社会と情報」と「情報の科学」が必修科目

<sup>1</sup> 愛知県立小牧高等学校  
Aichi Prefectural Komaki High School, Komaki, Aichi 485-0041, Japan

<sup>a)</sup> k619154u@gmail.com

「情報 I」に統合された。これにともない、「情報デザイン」や「プログラミング」、「データの活用」などの単元が必修となった。一方、教科「情報」の入試化を巡っては、令和 3 年 3 月 24 日に大学入試センターが公開した「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」[2]において、大学入学共通テスト（以下、「共通テスト」と表記）に「情報」を出題する指針が示された。さらに、令和 3 年 7 月 30 日に大学入試センターが公開した「令和 7 年度大学入学選抜に係る大学入学共通テスト実施大綱の予告」[3]において、共通テストに「情報」が導入されることが正式に決定した。

共通テストに出題される「情報」の問題に関しては、令和 2 年 11 月に大学入学共通テスト「情報」試作問題（検討用イメージ）[4]（以下、「試作問題」と表記）が大学入試センターより関係各所に通達された。続けて翌年の令和 3 年 3 月には、大学入学共通テスト「情報」サンプル問題 [5]（以下、「サンプル問題」と表記）が同センターより公開された。これら試作問題/サンプル問題におけるプログラミングに関する問いでは、Python の表記に似た疑似言語が使用されているが、令和 4 年 3 月現在、大学入学共通テスト「情報」に使用される疑似言語の仕様書は公開されていない。ただし、試作問題/サンプル問題に使用されている疑似言語は「情報関係基礎」に出題されている共通テスト手順記述標準言語（DNCL）[6]の表記に近いことから、本稿では両者をともに「DNCL」と表記する。

プログラミング言語の選定のための参考資料として、これまでにいくつかの情報が各所から提供されている。まず、令和元年 5 月に文部科学省が公開した高等学校情報科「情報 I」教員研修用教材 [7]があげられる。この教員研修用教材では、Python, JavaScript, VBA, ドリトル, Swift の 5 つのプログラミング言語を使用して、プログラミング分野の研修内容が記載されている（ただし、Python 以外の 4 つのプログラミング言語は、教員研修用教材の「本編」ではなく「別冊」[8]に記載がある）。なお、令和 2 年 6 月に公開された高等学校情報科「情報 II」教員研修用教材 [9]では、上記のような「別冊」は存在せず、「本編」のプログラミング言語には Python のみを使用された。次に、前述した試作問題/サンプル問題があげられる。試作問題/サンプル問題のプログラミング言語には DNCL が使用されているものの、問題の傾向を把握することで、授業で使用するプログラミング言語のイメージを持つことができた。最後に、令和 4 年 4 月に発行された「情報 I」の教科書があげられる。「情報 I」では、6 社の教科書会社より計 13 種（うち、2 種は採択時には 1 冊として扱われる）の教科書が発行され、これらの教科書には Python, JavaScript, VBA, Scratch の 4 つのプログラミング言語が使用されている。実際に授業で使用される教科書を比較することで、

具体的なプログラミング言語の選定もあわせて行うことができた。

令和 4 年度より「情報 I」が始まり、プログラミングの分野では、基本的に教科書に記載されたプログラミング言語を用いて授業を行うことになる。また、令和 7 年度からの共通テストに「情報」が導入されることから、各プログラミング言語と共通テスト「情報」に出題される DNCL との違いを意識しながらプログラミング教育を実施する必要がある。そこで本研究では、DNCL と「情報 I」の教科書に使用されている Python, JavaScript, VBA, Scratch との相違点について考察するとともに、試作問題/サンプル問題と教科書に記載されているプログラミング分野の内容を比較し、これらの結果をふまえて各プログラミング言語を授業で扱う際の留意点について述べる。なお、サンプル問題の表紙にも明記されているように、サンプル問題は「情報 I」の教科書の検定中に作成されたものであるため、教科書との照合はされてはならず、サンプル問題と教科書が相互に対応していないのは明白である（勿論、サンプル問題以前に作成された試作問題に関しても同様である）。また、後述するように、「情報 I」では複数の教科書を発行している教科書会社が大半を占めており、これは生徒の多様性や幅広い学力層に対応するための工夫や配慮だと考えられる。本研究は、決して教科書やプログラミング言語の良し悪しを決めたり、その優劣を比較したりするものではないという点について明確に断っておきたい。

## 2. 教科書目録と各教科書の特徴

文部科学省が公開している令和 3 年度の高等学校教科書目録 [10]によると、現行の学習指導要領において、各教科書会社が発行している共通教科「情報」の教科書は、表 1 に示すように「社会と情報」が計 13 種、「情報の科学」が計 8 種となっている。一方で、令和 4 年度の高等学校教科書目録 [11]によると、新学習指導要領において各教科書会社が発行を予定している「情報 I」の教科書は計 13 種となっている。ここで教科書の種類だけに着目すると、「社会と

表 1 共通教科「情報」における教科書の比較

Table 1 Composition of textbooks of “Information Study”.

教科書会社	令和 3 年度		令和 4 年度
	社会と情報	情報の科学	情報 I
A 社	2	1	2
B 社	2	2	4
C 社	1	0	1
D 社	3	2	2
E 社	4	2	3
F 社	1	1	1
計	13	8	13

情報」と「情報 I」は同じ 13 種であるが、「社会と情報」には新訂版や改訂版を含んでいる。そのため、「情報 I」では「社会と情報」や「情報の科学」と比較して多種多様な教科書が発行されていることが分かる。

以下、2.1 節～2.13 節に、プログラミング分野に関して各教科書を比較した際の特徴を述べる。教科書を比較する際に着目した点は、「単元（まとめ）の構成」、「採用しているプログラミング言語」、「プログラミングの題材と演習の流れ」、「発展的なプログラミングの内容」の 4 観点である。なお、教科書の採択において公平性を期すため、以下、各教科書会社を「A 社」「B 社」…、教科書名を「教科書 1」「教科書 2」…と表記する。

### 2.1 A 社—教科書 1

A 社の教科書 1 は全 5 章から構成され、このうち 3 章「コンピュータを活用する」がプログラミング分野に該当する。プログラミング言語には Python と Scratch が併記されている。プログラムの演習では、「Hello, world」の表示からはじまり、変数や制御構造について学習していく。その後、発展的なプログラムとして、乱数を使用したさいころやじゃんけん、円の面積を求める関数などを扱っている。なお、3 章以外にも、5 章「活動して提案する」において、Python を用いて簡易的な AI を作成したり、micro:bit を使用して開閉センサを作成する実習が記載されている。

### 2.2 A 社—教科書 2

A 社の教科書 2 は全 5 章から構成され、このうち 3 章「プログラミング」がプログラミング分野に該当する。プログラミング言語には Python と JavaScript が併記されている。プログラムの演習では、変数や代入、表示からはじまり、繰り返しや条件分岐、リスト、合計や平均の計算について学習していく。その後、発展的な内容として、乱数を使用したじゃんけんや疑似サイコロ、円の面積を求める関数などを扱っている。なお、3 章以外にも、5 章「問題解決」において、Python を用いて簡易的な AI や素数を求めるプログラムを作成したり、micro:bit を使用して開閉センサを作成する実習に加えて、ドリトルや Scratch を使用した問題解決に関する実習も取り入れられている。

### 2.3 B 社—教科書 1

B 社の教科書 1 は全 6 章から構成され、このうち第 6 章「プログラミング」がプログラミング分野に該当する。プログラミング言語には Python が使用されている。プログラムの演習では、順次構造、選択構造、反復構造からはじまり、配列、関数について学習していく。その後、発展的な内容として、線形/二分探索、交換/選択ソート、計算量などを扱っている。さらに高度な内容として、オブジェクト指向プログラミング、オープンデータの活用、動的シミュ

レーション、計測・制御などが記載されている。

### 2.4 B 社—教科書 2

B 社の教科書 2 は、2.3 節の教科書 1 とプログラミング言語が異なるだけで、プログラムの内容は同じである（ただし、プログラミング言語に依存する表現や用語の差異はある）。なお、プログラミング言語には JavaScript が使用されている。

### 2.5 B 社—教科書 3

B 社の教科書 3 は全 6 章から構成され、このうち第 6 章「アルゴリズムとプログラミング」がプログラミング分野に該当する。プログラミング言語には VBA が使用されている。プログラムの演習では、2 つのセルの値の足し算からはじまり、約数の判定、九九の計算、四捨五入の処理を通して、変数や制御文、配列、関数などについて学習していく。その後、発展的な内容として、線形/二分探索、交換ソート、計算量などを扱っている。

### 2.6 B 社—教科書 4

B 社の教科書 4 は全 6 章（序章を除く）から構成され、このうち 4 章「アルゴリズムとプログラム」がプログラミング分野に該当する。プログラミング言語には Scratch が使用されている。プログラムの演習では、画像を 3 回繰り返し表示するものからはじまり、合計金額の計算や棒グラフの表示を通して、変数や代入、配列/リスト、関数、グラフの描画などについて学習していく。その後、発展的な内容として、交換ソート、じゃんけんゲームの実習が記載されている。また、章末の StepUp!において、線形/二分探索に関するアルゴリズムの記載がある。

### 2.7 C 社—教科書 1

C 社の教科書 1 は全 5 章（最後の章は「終章」と表記）から構成され、このうち第 3 章「モデル化とシミュレーション、プログラミング～情報や情報技術を活用しよう～」がプログラミング分野に該当する。プログラミング言語には VBA が使用されている。プログラムの演習では、整数の和を求める計算からはじまり、偶数/奇数の判定、3 つの整数の合計処理を通して、基本構造（順次構造/分岐構造/反復構造）について学習していく。その後、発展的な内容として、モンテカルロ法による円周率の計算、交換/選択ソート、線形/二分探索の実習が記載されている（ただし、教科書にはソートと探索に関するプログラムは記載されていない）。

### 2.8 D 社—教科書 1

D 社の教科書 1 は全 4 編から構成され、このうち第 3 編「コンピュータとプログラミング」がプログラミング分野に

該当する。プログラミング言語には Python, JavaScript, VBA が併記されている。プログラムの演習では、「hello」の表示からはじまり、三角形の面積の計算、変数の大小の比較、連続する整数の表示を通して、変数や代入、配列、関数などについて学習していく。その後、発展的な内容として、「モデル化とシミュレーション」の単元でいくつかのシミュレーションのプログラムが記載されている。なお、第3編以外では巻末実習において、平方根の近似値計算（プログラムは VBA による記載）、フィボナッチ数列（プログラムは Python による記載）の実習が記載されている。

## 2.9 D 社—教科書 2

D 社の教科書 2 は全 4 編から構成され、このうち第 3 編「コンピュータとプログラミング」がプログラミング分野に該当する。プログラミング言語には Python, JavaScript, VBA が併記されている。プログラムの演習では、「hello」の表示からはじまり、三角形の面積の計算、変数の大小の比較、連続する整数の表示を通して、変数や代入、配列、関数などについて学習していく。その後、発展的な内容として、平方根の近似値計算（プログラムは VBA による記載）の実習が記載されている。

## 2.10 E 社—教科書 1

E 社の教科書 1 は全 4 章から構成され、このうち第 3 章「コンピュータとプログラミング」がプログラミング分野に該当する。プログラミング言語には Python が使用されている。プログラムの演習では、文字列の検索、素数判定という例題をもとに、変数やデータ型、演算、関数、配列などについて学習していく。その後、発展的な内容として、乱数による数値の振り分け、二次元配列、交換ソートを学習する。また、同章の「モデル化とシミュレーション」の単元では、人口の予測や売り上げの最大化（線形計画法）、待ち行列に関するプログラムが記載されている。なお、第 3 章以外にも、第 2 章「コミュニケーションと情報デザイン」において、JavaScript を用いた動的な Web ページのプログラムが記載されている（ただし、第 3 章のプログラミング分野に JavaScript の記載はない）。

## 2.11 E 社—教科書 2

E 社の教科書 2 は全 4 章から構成され、このうち第 3 章「コンピュータとプログラミング」がプログラミング分野に該当する。プログラミング言語には Scratch が使用されている。プログラムの演習では、入力した文字列を変数に格納して表示するものからはじまり、繰り返し処理、リスト、関数について学習していく。また、概念のみではあるが、発展的な内容として、構造化プログラミングとオブジェクト指向プログラミングに関する記載がある。なお、本書と 2.12 節の教科書 3 はブックインブック形式（本書に教科書

3 を挟み込む形）となっており、教科書の採択時には 1 冊として扱われる。

## 2.12 E 社—教科書 3

E 社の教科書 3 は 10 個の section から構成され、このうち section 6「プログラミング実習」がプログラミング分野に該当する。プログラミング言語には Scratch が使用されている。本書にはさまざまな実習課題が掲載されており、2.11 節の教科書 2 と併用してこれらの実習に取り組む流れとなっている。プログラムの演習では、スプライト（ねこのキャラクタ）を動かしたり、「あっちむいてホイ！」（乱数を使用したアニメーション）、「暗記ゲーム」（分岐/反復を用いるプログラム）の記載がある。

## 2.13 F 社—教科書 1

F 社の教科書 1 は全 4 章から構成され、このうち第 3 章「コンピュータとプログラミング」がプログラミング分野に該当する。プログラミング言語には VBA が使用されている。プログラムの演習では、電子さいころの作成、くじの表示やあたり判定を通して、変数や乱数、制御構造（順次/選択/反復）、サブルーチンなどについて学習していく。その後、発展的な内容として、素数判定やエラトステネスのふるい、章末実習では、画像を動かすプログラムに関する実習が掲載されている。

## 3. DNCL と各言語との主な相違点

令和 7 年からの共通テスト「情報」に出題されるプログラミング問題に関して、過去に大学入試センターから公開された試作問題/サンプル問題が参考になる。次に、両者のプログラミング問題の概要について説明する。

まず、試作問題は全 8 問から構成され、このうち第 5 問がプログラミング問題に該当する。第 5 問は、シフト暗号（シーザー暗号）による暗号解読をテーマにした問題であり、頻度分析プログラムを作成して暗号文の解読を行っている。第 5 問におけるプログラムのまともりは大きく 2 つあり、1 つは「出現頻度を求めるプログラム」、もう 1 つは「暗号文を復号するプログラム」（図 1 左）である。問題文は T さん、M さん、先生の 3 人による会話形式で進められ、図 1 のようにプログラムの空欄に入る適切なコードを解答群から選択する形式となっている。

次に、サンプル問題は全 3 問から構成され、このうち第 2 問がプログラミング問題に該当する。第 2 問は、比例代表選挙の当選者を決定する仕組みに関して、各政党に配分する議席数（当選者数）を決めるプログラムを作成する内容となっている。第 2 問におけるプログラムのまともりは大きく 2 つあり、1 つは「得票に比例した各政党の当選者数を求めるプログラム」（図 2 右）、もう 1 つは「各政党の当選者数を求めるプログラム」である。問題文は M さん、

問 3 次の会話文の空欄 **ス** ~ **チ** に当てはまる内容を、後の解答群のうちから一つずつ選べ。

**Tさん**: 暗号文を一文ずつ復号して表示するプログラムができたよ (図 7)。  
**Mさん**: なるほど、復号も右にシフトで考えているんだね。実行してみたら読み取れる英文になったの?

```

(01) Angoubun = ["p", "y", "e", "b", ... (省略) ... "k", "b", "d", "x", "."]
(02) 配列変数 Hirabun を初期化する
(03) hukugousuu = 26 - サシ
(04) i を 0 から 要素数 (Angoubun) - 1 まで 1 ずつ増やしながら:
(05)     bangou = 差分 (ケ)
(06)     もし bangou != -1 ならば:
(07)         もし ス <= 25 ならば:
(08)             Hirabun[i] = 文字 (ス)
(09)         そうでなければ:
(10)             Hirabun[i] = 文字 (セ)
(11)         そうでなければ:
(12)             Hirabun[i] = ソ
(13) 表示する (Hirabun)
    
```

図 7 暗号文を復号するプログラム

**【関数の説明】**

文字 (値) ... 番号の値に対するアルファベットの文字を返す。  
 値が 0 以上 25 以下でなければ「アルファベットでない」を返す  
 例: 文字 (4) は「e」を、文字 (23) は「x」を返す  
 文字 (-1) や文字 (27) は「アルファベットでない」を返す

**Tさん**: うん、復号したらこんな英文が表示されたよ。正しい英単語に変換されているみたいだから推測は当たっていたね。

four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal, now we are engaged in a great civil war, testing whether that nation, or any nation ... (省略) ... last full measure of devotion - that we here highly resolve that these dead shall not have died in vain - that this nation, under god, shall have a new birth of freedom - and that government of the people, by the people, for the people, shall not perish from the earth.

**Mさん**: これって有名なリンカーンのゲティスバーグ演説じゃない。ほら最後のところ有名なフレーズだよな。

**Tさん**: 先生、課題ができました。元の英文はリンカーンのゲティスバーグ演説ですね。プログラムで文字の出現頻度を調べて、シフトされた文字数を推測しました。復号はこのプログラムで変換してみました。

**先生**: よくできたね、素晴らしい! このプログラムはもっと簡単にできるね。この (07) ~ (10) の※部分は工夫すれば 1 行にまとめられるよ。ヒントは余りを求める算術演算子 % を使うんだ。

**Tさん**: えっ、1 行ですか? ... 分かった!

```
Hirabun[i] = 文字 (タ) チ
```

とすればもっと簡潔にできたんだ。

**先生**: 素晴らしい!

**ス** ~ **ソ** の解答群

01 bangou+hukugousuu	02 bangou
03 hukugousuu	04 bangou+hukugousuu-26
05 bangou+hukugousuu-25	06 hukugousuu-26
07 Angoubun[i]	08 Hirabun[i]
09 Angoubun[i+hukugousuu]	

**タ** の解答群

01 bangou+hukugousuu	02 (bangou+hukugousuu)
03 i+hukugousuu	04 (i+hukugousuu)
05 hukugousuu+26	06 (hukugousuu+26)

**チ** の解答群

01 25	02 26	03 bangou	04 hukugousuu
-------	-------	-----------	---------------

図 1 大学入学共通テスト「情報」試作問題 (第 5 問—問 3)

Fig. 1 The Common Test for University Admissions “Information Study” prototype test (Question 5-3).

Kさん、先生の3人による会話形式で進められ、図2のようにプログラムの空欄に入る適切なコードを解答群から選択する形式となっている。

ここで、サンプル問題の第2問—問1で使用されているプログラム (図2右上のプログラムの空欄に解答を埋めたもの) を図3(a)に示す。また、図3(a)のDNCLで書かれたプログラムについて、「情報I」の教科書に使用されているPython, JavaScript, VBA, Scratchで書き直したものを図3(b)~(e)にそれぞれ示す。以下、3.1節~3.4節において、DNCLと各プログラミング言語との主な相違点について述べる。ただし、プログラミング教育ではプログラミングの考え方/概念を学習することが重要であるため、ここではプログラミング言語の文法のみに着目した相違点に関する言及にとどめる。

### 3.1 DNCLとPythonとの主な相違点

図3(a)のDNCLで書かれたプログラムをPythonで書き直したプログラムが図3(b)である。図3(b)のように、両者はコードを1対1で対応させることができる。DNCLとPythonとの主な相違点として次の点があげられる。

#### • for文の表記

DNCLではfor文を「Aを0から3まで1ずつ増やしながら繰り返す:」と表記するところを、Pythonでは「for A in range(4):」や「for A in range(0, 4):」と表記する。このように、Pythonでは終了値に1を加算した値をrange内に表記する必要がある。

### 3.2 DNCLとJavaScriptとの主な相違点

図3(a)のDNCLで書かれたプログラムをJavaScriptで書き直したプログラムが図3(c)である。図3(c)のように、両者はコードをおおむね1対1で対応させることができる。DNCLとJavaScriptとの主な相違点として次の点があげられる。

#### • 変数の宣言

DNCLでは変数の宣言が必要ないが、JavaScriptでは変数を宣言するためにvar/let/constのいずれかを使用する必要がある。

#### • ステートメント末尾のセミコロン

JavaScriptではステートメント末尾に「;」の記載が必要であるが、DNCLではこれが不要な。

**第2問** 次の文章を読み、後の問い(問1~3)に答えよ。

Mさんは、18歳になって選挙権が得られたのを機に、比例代表選挙の当選者を決定する仕組みに興味を持った。そこで各政党に配分する議席数(当選者数)を決める方法を、友人のKさんとプログラムを用いて検討してみることにした。

**問1** 次の文章の空欄「ア」~「ウ」に入れる最も適当なものを、後の解答群のうちから一つずつ選べ。同じものを繰り返し選んでもよい。

Mさん：表1に、最近行われた選挙結果のうち、ある地域のブロックについて、各政党の得票数を書いてみたよ。

	A党	B党	C党	D党
得票数	1200	660	1440	180

Kさん：今回の議席数は6人だったね。得票の総数を議席数で割ると580人なので、これを基準得票数と呼ぶのがいいかな。平均して1議席が何票分の重みがあるかを表す数ということで。そうすると、各政党の得票数が何議席分に相当するかわは、各政党の得票数をこの基準得票数で割れば求められるね。

Mさん：その考え方に沿って政党ごとの当選者数を計算するプログラムを書いてみよう。まず、プログラムの中で扱うデータを図1と図2にまとめてみたよ。配列Tomeiには各政党の党名を、配列Tokuhyoには各政党の得票数を格納することにしよう。政党の数は4つなので、各配列の添字は0から3だね。

i	0	1	2	3
Tomei	A党	B党	C党	D党

i	0	1	2	3
Tokuhyo	1200	660	1440	180

Mさん：では、これらのデータを使って、各政党の当選者数を求める図3のプログラムを書いてみよう。実行したら図4の結果が表示されたよ。

```

(01) Tomei = ["A党", "B党", "C党", "D党"]
(02) Tokuhyo = [1200, 660, 1440, 180]
(03) sousuu = 0
(04) giseki = 6
(05) mを0から「ア」まで1ずつ増やしながら繰り返す：
(06)   sousuu = sousuu + Tokuhyo[m]
(07)   kizyunsuu = sousuu / giseki
(08)   表示する("基準得票数：", kizyunsuu)
(09)   表示する("比例配分")
(10) mを0から「ア」まで1ずつ増やしながら繰り返す：
(11)   表示する(Tomei[m], "：", 「イ」 / 「ウ」)
    
```

図3 得票に比例した各政党の当選者数を求めるプログラム

基準得票数：580  
 比例配分  
 A党：2.068966  
 B党：1.137931  
 C党：2.482759  
 D党：0.310345

図4 各政党の当選者数の表示

「ア」~「ウ」の解答群

① 0   ② 1   ③ 2   ④ 3   ⑤ 4   ⑥ 5   ⑦ Tomei[m]  
 ⑧ Tokuhyo[m]   ⑨ sousuu   ⑩ giseki   ⑪ kizyunsuu

図2 大学入学共通テスト「情報」サンプル問題(第2問—問1)  
 Fig. 2 The Common Test for University Admissions "Information Study" sample test (Question 2-1).

・ブロックの区切り

DNCLでは「:」とインデントを使用してブロックを区切るが、JavaScriptではブロックの区切りに「{ }」を使用する。

・オブジェクトの結合

DNCLではオブジェクトの結合に「,」を使用するが、JavaScriptでは「+」を使用する。

・インクリメントの表記

DNCLではインクリメントを「A = A + 1」と表記するが、JavaScriptでは「A++」と表記することが通例である。ただし、JavaScriptでも「A = A + 1」と表記できる。

・for文の表記

DNCLではfor文を「Aを0から3まで1ずつ増やしながら繰り返す:」と表記するところを、JavaScriptでは「for (var A = 0; A < 4; A++)」と表記する。このように、JavaScriptではforの括弧内を「初期化式; 条件式; 変化式」のように表記する必要がある。

3.3 DNCLとVBAとの主な相違点

図3(a)のDNCLで書かれたプログラムをVBAで書き

直したプログラムが図3(d)である。図3(d)のように、両者はコードをおおむね1対1で対応させることができる。DNCLとVBAとの主な相違点として次の点があげられる。

・変数の宣言

DNCLでは変数の宣言は必要ないが、VBAでは変数を宣言するためにDim文が必要である。

・ブロックの区切り

DNCLでは「:」とインデントを使用してブロックを区切るが、VBAではブロックの区切りに「Next」や「End」を使用する。

・オブジェクトの結合

DNCLではオブジェクトの結合に「,」を使用するが、VBAでは「&」を使用する。

・配列の作成と操作

DNCLでは配列を作成するために関数を使用する必要はないが、VBAでは複数の値をまとめて配列に格納する場合、Array関数を使用する必要がある。また、DNCLでは配列の操作に「Test[3]」のように表記するが、VBAでは括弧が異なり「Test(3)」と表記する。

```
(01) Tomei = ["A党", "B党", "C党", "D党"]
(02) Tokuhyo = [1200, 660, 1440, 180]
(03) sousuu = 0
(04) giseki = 6
(05) mを0から3まで1ずつ増やしながら繰り返す:
(06) | sousuu = sousuu + Tokuhyo[m]
(07) kizyunsuu = sousuu / giseki
(08) 表示する("基準得票数:", kizyunsuu)
(09) 表示する("比例配分")
(10) mを0から3まで1ずつ増やしながら繰り返す:
(11) | 表示する(Tomei[m], ":", Tokuhyo[m] / kizyunsuu)
```

(a) DNCL で書かれたプログラム

```
Tomei = ['A党', 'B党', 'C党', 'D党']
Tokuhyo = [1200, 660, 1440, 180]
sousuu = 0
giseki = 6
for m in range(4):
    sousuu = sousuu + Tokuhyo[m]
kizyunsuu = sousuu / giseki
print('基準得票数:', kizyunsuu)
print('比例配分')
for m in range(4):
    print(Tomei[m], ':', Tokuhyo[m] / kizyunsuu)
```

(b) Python で書き直したプログラム

```
var Tomei = ["A党", "B党", "C党", "D党"];
var Tokuhyo = [1200, 660, 1440, 180];
var sousuu = 0;
var giseki = 6;
for (var m = 0; m < 4; m++){
    sousuu = sousuu + Tokuhyo[m];
}
var kizyunsuu = sousuu / giseki;
alert("基準得票数:" + kizyunsuu);
alert("比例配分");
for (var m = 0; m < 4; m++){
    alert(Tomei[m] + " : " + Tokuhyo[m] / kizyunsuu);
}
```

(c) JavaScript で書き直したプログラム

```
Sub サンプル問題()
Dim Tomei As Variant, Tokuhyo As Variant
Dim sousuu As Integer, giseki As Integer, m As Integer
Dim kizyunsuu As Double
Tomei = Array("A党", "B党", "C党", "D党")
Tokuhyo = Array(1200, 660, 1440, 180)
sousuu = 0
giseki = 6
For m = 0 To 3
    sousuu = sousuu + Tokuhyo(m)
Next m
kizyunsuu = sousuu / giseki
MsgBox "基準得票数:" & kizyunsuu
MsgBox "比例配分"
For m = 0 To 3
    MsgBox Tomei(m) & " : " & Tokuhyo(m) / kizyunsuu
Next m
End Sub
```

(d) VBA で書き直したプログラム

The Scratch script starts with a 'when green flag clicked' event. It initializes variables: 'Tomei' (a list containing 'A党', 'B党', 'C党', 'D党'), 'Tokuhyo' (a list containing 1200, 660, 1440, 180), 'sousuu' (0), and 'giseki' (6). It then enters a loop that repeats 4 times. Inside the loop, it adds the value of 'Tokuhyo' at index 'm' to 'sousuu'. After the loop, it calculates 'kizyunsuu' as 'sousuu' divided by 'giseki'. It then shows two speech bubbles: '基準得票数:' followed by 'kizyunsuu', and '比例配分'. The loop is repeated again, but this time it shows a speech bubble with 'Tomei' at index 'm', a colon, and 'Tokuhyo' at index 'm' divided by 'kizyunsuu'. Finally, it increments 'm' and repeats the loop.

(e) Scratch で書き直したプログラム

図 3 4つのプログラミング言語を使用して書き直したプログラムの比較

Fig. 3 Comparison of programs rewritten in four programming languages.

### 3.4 DNCL と Scratch との主な相違点

図 3(a) の DNCL で書かれたプログラムを Scratch で書き直したプログラムが図 3(e) である。Scratch ではブロックを組み合わせてプログラムを作成するため、図 3(e) のように両者はコードを 1 対 1 で対応させることは難しい。DNCL と Scratch の主な相違点として次の点があげられる。

#### ・文字列の表記

DNCL では文字列を「`""`」で囲う必要があるが、Scratch ではこれが必要ない。

#### ・オブジェクトの結合

DNCL ではオブジェクトの結合に「`,`」を使用するが、Scratch では「`●と●`」のブロックを使用する。

#### ・リストの作成と操作

DNCL では複数の要素を配列に代入する場合、1 行でまとめて記述することができる。一方、Scratch ではリストに追加する要素の数だけブロックが必要であり、加えてプログラム実行時にリスト内を初期化するためのブロックがプログラムの先頭に必要となる。また、DNCL では添字は 0 から始まるが、Scratch の添字は 1 から始まる点でも異

なる。

#### ・インクリメントの表記

DNCL ではインクリメントを「 $A = A + 1$ 」と表記するが、Scratch では「 $A$  を 1 ずつ変える」というブロックを使用することが通例である。ただし、Scratch でも「 $A$  を  $A + 1$  にする」というブロックを使用して同様の処理を表記することもできる。

#### ・if 文の表記

DNCL では if 文の「偽の場合」の処理を「そうでなければ:」の後に記述し、Scratch では「でなければ」の後に記述する。ただし、Scratch には、複数の条件式を記載するブロック（たとえば、Python の「elif」に該当するもの）がないため、Scratch でこれを実装したい場合、if 文のブロックを入れ子にして使用する必要がある。

#### ・for 文の表記

DNCL では for 文を「 $A$  を 0 から 3 まで 1 ずつ増やしながら繰り返す:」のように表記するが、Scratch では「4 回繰り返す」という表記になり、繰り返しのブロック内で変数はインクリメントされない。そのため、Scratch では繰り返しのブロック内で「 $A$  を 1 ずつ変える」などのブロックを使用する必要がある。

#### ・while 文の表記

DNCL では while 文を「 $0 < A$  の間繰り返す:」のように表記するが、Scratch にはこのような while 文に該当するブロックは用意されていない。代わりに、Scratch には until 文に該当する「●まで繰り返す」というブロックがあるため、これを使用して「 $0 \geq A$  まで繰り返す」と条件を反転して表記する必要がある。

#### ・比較演算子の表記

DNCL では比較演算子に「 $>=$ 」と「 $<=$ 」を使用することができるが、Scratch ではこれらのブロックは用意されていない。そのため Scratch では、たとえば「 $A \geq 0$ 」と表記したい場合、「 $A > 0$ 」「かつ」「 $A = 0$ 」というように論理演算子のブロックを使用して結合したり、条件式を「 $A < 0$ 」として真/偽の処理を逆にする必要がある。

## 4. 試作問題/サンプル問題と教科書の比較

試作問題の第 5 問およびサンプル問題の第 2 問に出題されたプログラムには、表 2 の (1)~(13) に示す 13 個のプログラミングに関する要素（以下、「プログラミングの基本要素」と表記）が使用されていた。ここで、「情報 I」の各教科書におけるプログラミングの基本要素の記載の有無を表 2 の「プログラミングの基本要素」欄に示す。表 2 における各記号の意味は次のとおりである。

○：該当の要素に関する解説あるいはプログラムの記載があるもの。

△：該当の要素の一部に関する解説あるいはプログラムの記載があるもの。

×：該当の要素に関する解説やプログラムの記載がないもの。

以下、4.1 節~4.13 節において、プログラミングの基本要素に対して各教科書の内容を比較する。なお、E 社の教科書 2 と教科書 3 はブックインブック形式での採択となるため、表 2 では 1 冊に合算した結果を表記している。

### 4.1 変数

すべての教科書に変数に関する解説あるいはプログラムの記載があった。なお、試作問題/サンプル問題でも多くの変数が使用されている。

### 4.2 インクリメント

すべての教科書にインクリメントに関する解説あるいはプログラムの記載があった。なお、試作問題/サンプル問題でもインクリメントが使用されている。

### 4.3 メッセージ

メッセージとは、画面にテキストを出力するプログラムを意味しており、DNCL では「表示する ()」、Python では「print()」、JavaScript では「alert()」や「console.log()」、VBA では「MsgBox()」、Scratch では「●と言う」のブロックが該当する。なお、試作問題/サンプル問題でも多くのメッセージが使用されている。一部、VBA を採用している 705 の教科書は、MsgBox() の代わりに Cells プロパティを使用して値をセルに出力しているため「×」としている。

### 4.4 異なるデータ型の結合

異なるデータ型の結合とは、たとえば数値と文字列などの値の結合を意味しており、DNCL では「,」、Python では「,」や「+」、JavaScript では「+」、VBA では「&」、Scratch では「●と●」のブロックが該当する。なお、サンプル問題において異なるデータ型の結合がメッセージの引数で使用されている。一部、VBA を採用している教科書では、異なるデータ型の結合に関する解説あるいはプログラムの記載がないため「×」としている。

### 4.5 算術演算子

算術演算子には、足し算、引き算、掛け算、割り算、剰余算などが該当する。なお、試作問題には四則演算子に加えて剰余算を表す「%」が使用されているため、剰余算の記載がなかった教科書は「△」としている。

### 4.6 比較演算子

比較演算子には、「 $>$ 」（より大きい）、「 $<$ 」（より小さい/未満）、「 $\geq$ 」（以上）、「 $\leq$ 」（以下）、「 $=$ 」や「 $==$ 」（等価）、「 $!=$ 」や「 $<>$ 」（不等価）が該当する。なお、試作問題では「 $\leq$ 」と「 $!=$ 」、サンプル問題では「 $=$ 」と「 $>$ 」が



表 2 情報 I の教科書におけるプログラミング分野の比較  
Table 2 Comparison of programming fields in Information Study I textbooks.

教科書会社	教科書	プログラミング言語				プログラミングの基本要素												
		Python	JavaScript	VBA	Scratch	(1) 変数	(2) インクリメント	(3) メッセージ	(4) 異なるデータ型の結合	(5) 算術演算子	(6) 比較演算子	(7) 論理演算子	(8) 条件分岐 (if 文)	(9) 繰り返し (for 文)	(10) 繰り返し (while 文)	(11) 入れ子 (ネスト)	(12) 配列 / リスト	(13) 関数
A 社	教科書 1	●			●	○	○	○	○	○	○	×	○	○	○	○	○	○
	教科書 2	●	●			○	○	○	○	○	○	×	○	○	○	○	○	○
B 社	教科書 1	●				○	○	○	○	○	○	○	○	○	○	○	○	○
	教科書 2		●			○	○	○	○	○	○	○	○	○	○	○	○	○
	教科書 3			●		○	○	×	×	○	○	×	○	○	○	○	○	○
	教科書 4				●	○	○	○	○	○	△	△	○	○	△	○	○	○
C 社	教科書 1			●		○	○	○	×	○	○	×	○	○	○	○	○	×
D 社	教科書 1	●	●	●		○	○	○	×	△	○	×	○	×	○	○	○	○
	教科書 2	●	●	●		○	○	○	×	△	○	×	○	×	○	×	○	○
E 社	教科書 1	●				○	○	○	○	○	○	○	○	○	○	○	○	○
	教科書 2					○	○	○	○	△	△	×	○	○	△	○	○	○
	教科書 3				●	○	○	○	○	△	△	×	○	○	△	○	○	○
F 社	教科書 1			●		○	○	○	○	○	○	○	○	○	△	○	○	×

使用されており、一部の比較演算子の記載がなかった教科書は「△」としている。

#### 4.7 論理演算子

論理演算子には、論理積「and」、論理和「or」、否定「not」などが含まれる。なお、サンプル問題では選択肢においてこれらの論理演算子が使用されており、一部の論理演算子の記載がなかった教科書を「△」としている。

#### 4.8 条件分岐 (if 文)

すべての教科書に条件分岐 (偽の場合の処理を含む) に

関する解説あるいはプログラムの記載があった。なお、サンプル問題では偽の場合の処理の記載がなかったが、試作問題では「そうでなければ:」の記載があった。

#### 4.9 繰り返し (for 文)

繰り返し (for 文) とは、「指定回数繰り返す処理」のことであり、DNCL では「<変数> を <開始値> から <終了値> まで <増減値> ずつ増やしながら:」が該当する。なお、試作問題/サンプル問題でも多くの繰り返し (for 文) が使用されている。この指定回数繰り返す for 文の代わりに、条件を満たす間繰り返す while 文のみを使用している

教科書を「×」としている。

#### 4.10 繰り返し (while 文)

繰り返し (while 文) とは、「条件を満たす間繰り返す処理」のことであり、DNCL では「<条件式>の間繰り返す:」が該当する。なお、サンプル問題では繰り返し (while 文) が使用されている。一部の教科書は、while 文ではなく until 文 (条件を満たすまで繰り返す処理) が使用されているため「△」としている。

#### 4.11 入れ子 (ネスト)

入れ子 (ネスト) とは、for 文の中に for 文、for 文の中に if 文、while 文の中に if 文がある場合などの処理を意味する。なお、試作問題では「for 文—if 文—if 文」、サンプル問題では「while 文—for 文—if 文」という三重の入れ子となって使用されている。

#### 4.12 配列/リスト

すべての教科書に配列/リストに関する解説あるいはプログラムの記載があった (JavaScript と VBA は「配列」、Python と Scratch は「リスト」の表記)。なお、試作問題/サンプル問題でも配列/リストが使用されている。

#### 4.13 関数

関数には「組み込み関数」と「ユーザ定義関数」の 2 種類があるが、ここでの関数は特に後者を意味しており、Python では「def」、JavaScript では「function」、VBA では「Function」、Scratch では「ブロックの定義」がこれに該当する。なお、試作問題では配列の要素数を返す「要素数」、文字の位置の差分を返す「差分」、サンプル問題では小数点以下を切り捨てる「切り捨てる」という関数が使用されていた。一部、VBA を採用している教科書は Function プロシージャによる関数に関する解説あるいはプログラムの記載がないため「×」としている。

## 5. おわりに

本研究では、DNCL と「情報 I」の教科書に使用されている Python, JavaScript, VBA, Scratch との相違点について考察するとともに、試作問題/サンプル問題と教科書に記載されているプログラミング分野の内容について比較した。以下、これらの結果について考察する。

まず、DNCL と 4 つのプログラミング言語との相違点 (文法のみに着目) について考察した結果、変数の宣言、ブロックの区切り、オブジェクトの結合、配列/リストの作成と操作、for 文の表記などに大きな違いがあることが分かった。このうち for 文は、図 3 に示したように DNCL と 4 つのプログラミング言語で表記方法が大きく異なるため、特に注意したい。そのため、授業者は、このような DNCL

と授業で扱うプログラミング言語との相違点について意識するとともに、必要に応じて説明や演習を加えながら授業を進めることが大切であると考え。

次に、試作問題/サンプル問題と教科書に記載されているプログラミング分野の内容について比較した結果、すべての教科書が試作問題/サンプル問題に出題された内容を網羅できているわけではないことが分かった。しかし、共通テスト「情報」の対策として、少なくとも試作問題/サンプル問題に出題された内容を理解しておくことは、一般的に考えて必須となるだろう。そのため、授業者は、表 2 に示したような教科書に記載がない項目について意識するとともに、必要に応じて説明や演習を加えながら授業を進めることが大切であると考え。

本研究では、試作問題/サンプル問題を対象に各教科書の比較と考察を行ったため、共通テストへの対応という観点では十分な分析はできていない。そのため、令和 7 年度から始まる共通テスト「情報」の動向によっては、本稿で示したこと以外にも、授業で扱うプログラミング言語によって対策が必要になる可能性がある。ただし、現在のところ共通テストを想定して作成された問題は試作問題/サンプル問題だけであり、当分の間はセンター試験/共通テストの「数学②」の枠に設定されている科目「情報関係基礎」の過去問 [12] が大いに参考になると考えられる。今後は、試作問題/サンプル問題に加えて「情報関係基礎」の問題も参考にしながら、共通テストに必要なプログラミングに関する知識および技能について考察を深めていきたい。

本研究の一部は、日本情報科教育学会第 14 回全国大会、日本産業技術教育学会第 64 回全国大会、情報処理学会情報教育シンポジウム 2021、情報処理学会高校教科「情報」シンポジウム 2021 秋において発表している。

謝辞 本研究は JSPS 科研費 JP21H03962 の助成を受けたものです。

## 参考文献

- [1] 文部科学省：高等学校学習指導要領 (平成 30 年告示), 入手先 (<https://www.mext.go.jp/content/1384661.6.1.3.pdf>) (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 (令和 3 年 3 月 24 日), 入手先 ([https://www.dnc.ac.jp/kyotsu/shiken\\_jouhou/r7ikou.html](https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html)) (参照 2021-06-20)。
- [3] 大学入試センター：令和 7 年度大学入学者選抜に係る大学入学共通テスト実施大綱の予告, 入手先 ([https://www.mext.go.jp/content/20210729-mxt\\_daigakuc02-000005144.2.pdf](https://www.mext.go.jp/content/20210729-mxt_daigakuc02-000005144.2.pdf)) (参照 2021-07-30)。
- [4] 大学入試センター：大学入学共通テスト「情報」試作問題 (検討用イメージ), 入手先 (<https://www.ipsj.or.jp/education/9faeag0000012a50-att/sanko2.pdf>) (参照 2021-06-20)。
- [5] 大学入試センター：大学入学共通テスト「情報」サンプル問題, 入手先 (<https://www.dnc.ac.jp/albums/abm.php>)

- ?f=abm00040365.pdf&n=サンプル問題 (情報)\_修正版.pdf) (参照 2021-06-20).
- [6] 大学入試センター：共通テスト手順記述標準言語 (DNCL) の説明 (2021 年 1 月), 入手先 (<https://www.dnc.ac.jp/albums/abm00040701.pdf>) (参照 2022-01-05).
- [7] 文部科学省：高等学校情報科「情報 I」教員研修用教材 (本編), 入手先 ([https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1416756.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm)) (参照 2021-06-20).
- [8] 文部科学省：高等学校情報科「情報 I」教員研修用教材 (第 3 章 他プログラミング言語版), 入手先 ([https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1421808.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1421808.htm)) (参照 2021-06-20).
- [9] 文部科学省：高等学校情報科「情報 II」教員研修用教材 (本編), 入手先 ([https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/mext\\_00742.html](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_00742.html)) (参照 2021-06-20).
- [10] 文部科学省：高等学校用教科書目録 (令和 3 年度使用), pp.44-45, 入手先 ([https://www.mext.go.jp/content/20200430\\_mxt\\_kouhou02...mext\\_00001\\_03.pdf](https://www.mext.go.jp/content/20200430_mxt_kouhou02...mext_00001_03.pdf)) (参照 2021-06-20).
- [11] 文部科学省：高等学校用教科書目録 (令和 4 年度使用), p.23, 入手先 ([https://www.mext.go.jp/content/20210604-mxt\\_kyokasyo02-000014470\\_4.pdf](https://www.mext.go.jp/content/20210604-mxt_kyokasyo02-000014470_4.pdf)) (参照 2021-06-20).
- [12] 情報処理学会情報入試委員会：情報関係基礎 アーカイブ, 入手先 (<https://sites.google.com/a/ipsj.or.jp/ipsjnn/resources/JHK>) (参照 2021-06-20).



井手 広康 (正会員)

愛知県立小牧高等学校教諭。愛知県立大学大学院情報科学研究科博士後期課程修了, 博士 (情報科学)。情報教育やゲーム情報学に関する研究に従事。本会初等中等教育委員会幹事, 本会ジュニア会員活性化委員会委員, 本会会誌

編集委員会 (教育分野/EWG) 委員, 本会論文誌教育とコンピュータ編集委員会編集委員, 本会コンピュータと教育研究会運営委員, 日本産業技術教育学会理事, 日本情報科教育学会理事, 情報オリンピック日本委員会ジュニア部会委員等。