

実践論文

情報Ⅰにおけるプログラミング言語の選択が 大学入学共通テストの解答に及ぼす影響

井手 広康^{1,a)}

受付日 2022年2月12日, 再受付日 2022年7月19日,
採録日 2022年8月29日

概要: 令和4年度から始まった「情報Ⅰ」の教科書に Python, JavaScript, VBA, Scratch の4つのプログラミング言語が使用されたことから, 多くの高等学校がいずれかのプログラミング言語を使用して授業を行うことになる。本研究では, クラスごとに4つのプログラミング言語を使い分けて同じ授業内容でプログラミング教育を実践し, 最後に大学入学共通テスト「情報」サンプル問題(第2問)を全員に解答させた。サンプル問題の解答結果や事後アンケートの結果から, どのプログラミング言語を授業で使用していても, 大学入学共通テストの解答に大きな影響を及ぼす可能性が低い, 各プログラミング言語が持つ特有の表記や仕様により, 少なからず解答に影響が生じていることが明らかとなった。

キーワード: 情報Ⅰ, プログラミング教育, プログラミング言語, 大学入学共通テスト, DNCL

The Effect of Programming Language Selection in Information Study I on the Answers to the Common Test for University Admissions

HIROYASU IDE^{1,a)}

Received: February 12, 2022, Revised: July 19, 2022,
Accepted: August 29, 2022

Abstract: Since the four programming languages of Python, JavaScript, VBA, and Scratch were used in the textbook of “Information Study I” started from April 2022, all high schools must decide to use one of the programming languages for teaching. In this study, we practiced programming education with the same content by using different programming languages for each class, and finally asked everyone to answer the second question of the Common Test for University Admissions sample problem “Information Study”. From the answer results of the sample problem and the results of the post-questionnaire, it is unlikely that any programming language used in the class will have a significant effect on the answer to the Common Test, but it was clarified that the unique notation and specifications of programming languages have a considerable influence on the answers to the Common Test.

Keywords: information Study I, programming education, programming language, the common test for university admissions, DNCL

1. はじめに

令和4年度より高等学校において年次進行で実施されている平成30年告示高等学校学習指導要領[1]では, 教科「情報」は必修科目「情報Ⅰ」と選択科目「情報Ⅱ」で構

成される。このうち必修科目である「情報Ⅰ」は, 旧学習指導要領における「社会と情報」と「情報の科学」が統合された形となり, 「情報デザイン」や「プログラミング」, 「データの活用」といった単元が必修化された。また, 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」[2]では, 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,

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

^{a)} k619154u@gmail.com

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年度大学入学者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。さらに、令和3年9月29日に大学入試センターが公開した「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（補遺）」[4]では、共通テスト「情報」の試験時間が60分であること、令和3年12月17日に同センターが公開した「令和7年度大学入学者選抜に係る大学入学共通テストにおける得点調整の対象教科・科目について」[5]では、試験科目「情報」の経過措置科目として「旧情報（仮）」が設置され、必要に応じて得点調整がされることが決定した。加えて、令和4年1月28日に国立大学協会が公表した「2024年度以降の国立大学の入学者選抜制度—国立大学協会の基本方針—」[6]では、「2024年度に実施する入学者選抜から、すべての国立大学は、『一般選抜』においては第一次試験として、高等学校等における基礎的教科・科目についての学習の達成度を測るため、原則としてこれまでの『5教科7科目』に『情報』を加えた6教科8科目を課す」としたことで、大学入学希望者の多くが共通テストで「情報」を受験することが予想される。

このような背景を受け、令和4年度より始まった「情報I」では、共通テスト「情報」の受験を意識した授業を行う必要がある。共通テスト「情報」の参考となる資料としては、大学入試センターが令和2年11月24日に関係各所に通達した大学入学共通テスト「情報」試作問題（検討用イメージ）[7]（以下、「試作問題」*1と表記）、令和3年3月24日に公開された大学入試共通テスト「情報」サンプル問題[8]（以下、「サンプル問題」と表記）があげられる。いずれの問題にも「実際の問題セットをイメージしたものや試験時間を考慮したものではない」という旨の注意書きはあるが、共通テスト「情報」をイメージするうえで貴重な資料となっている。試作問題とサンプル問題のプログラミング言語には、Pythonの表記に似た疑似言語が使用されており、サンプル問題とともに公開された「サンプル問題のねらい」[9]には「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」と明記されている。ただし、大学入試センターは「大学入学共通テストで使われるからといって、試験対策のためにDNCLを使って授業が行われることがないように考えている」[10]としている（この「DNCL」は「情報関係基礎」で使用されている疑似言語の名称であり、詳細は2.3節で説明する）。また、実際に授業で扱うプログラミング言語の具体例について、文部科学省や大学入試センターは公言していないが、「情報I」の12種類（2

分冊となっているものは1種類として計上）の教科書にPython, JavaScript, VBA, Scratchの4つのプログラミング言語[11]が使用されたことから、多くの高等学校が上記のいずれかのプログラミング言語を使用して授業を行うことになる。

ただし、試作問題とサンプル問題に使用された疑似言語がPythonの表記に似ていることが影響して、「プログラミング言語はPythonを使用した方が共通テストに有利に働くのではないか」という声や、反対に「Scratchはブロック型プログラミング言語であるため、共通テストでは不利になるのではないか」という声を耳にすることがある。また、特定のプログラミング言語の選択が共通テスト「情報」の解答に及ぼす影響に関して調査した文献は現在のところ見当たらず、上記のような声は推測の域を出ない。

そこで本研究では、クラスごとにPython, JavaScript, VBA, Scratchの4つのプログラミング言語を分けて同じ学習内容でプログラミング教育を実践し、最後の授業でサンプル問題の第2問「プログラミング」を全員に解答させた。本稿では、サンプル問題（第2問）の解答結果や事後アンケートの結果から、「情報I」におけるプログラミング言語の選択が、共通テスト「情報」におけるプログラミング問題の解答に及ぼす影響について考察する。

2. 試作問題とサンプル問題の概要

先に1章で述べたように、共通テスト「情報」の参考となる資料としては、令和2年11月24日に大学入試センターが関係各所に通達した試作問題と、令和3年3月24日に同センターが公開したサンプル問題があげられる。本章では、試作問題およびサンプル問題の概要と、試作問題とサンプル問題のプログラムに使用されている疑似言語について説明する。

2.1 試作問題

試作問題は表1に示すように全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まりは大きく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」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、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して解答する形式となっている。

第5問のプログラムでは、暗号文の文字を配列に格納し、1文字ずつアルファベットの出現頻度を数え上げて、その結果を別の配列に格納する。条件分岐や繰り返し、配列/リスト、関数の基本的な知識はもとより、入れ子（ネスト）による手順（試作問題では「for—if—if」の三重の入れ子が

*1 令和4年11月9日には大学入試センターより大学入学共通テスト試作問題「情報I」が公開されたが、本稿における「試作問題」は令和2年11月24日に通達されたものを指す。

表 1 試作問題の構成

Table 1 Configuration of trial problem.

問題番号	内容
第 1 問	法規や制度, 情報モラルなど
第 2 問	問 1 情報量など
	問 2 動画の仕組みとデータの容量
第 3 問	画像処理
第 4 問	交通渋滞シミュレーション
第 5 問	プログラミングによる暗号解読
第 6 問	二要素認証によるセキュリティ強化
第 7 問	ネットワークの不具合の原因究明
第 8 問	Web アクセスログの分析など

表 2 サンプル問題の構成

Table 2 Configuration of sample problem.

問題番号	内容
第 1 問	問 1 災害時における SNS の使用
	問 2 図によるデータの表現
	問 3 画像のデジタルデータへの変換
	問 4 IP アドレスとネットワークアドレス
第 2 問	各政党の当選者数を求めるプログラム
第 3 問	サッカーチームのデータ分析

使用されている) や, 「%」や「Mod」などの演算子を使用した剰余による配列の操作について理解していなければ解答することは難しい。

2.2 サンプル問題

サンプル問題は表 2 に示すように全 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。ここで, サンプル問題の第 2 問 (一部抜粋) を図 1 に示す。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) を決める方法をプログラム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 つは図 1 左側に示した「得票に比例した各政党の当選者数を求めるプログラム」, もう 1 つは図 1 右側に示した「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, K さん, 先生の 3 人の会話形式で進められ, 試作問題と同様に, 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して解答する形式となっている。

サンプル問題の第 2 問のプログラムでは, 比例代表選挙のドント方式ののっとり, 配列 Hikaku と配列 Tosen を変化させながら, 各政党の当選者数を求めていく。試作問題と同様に, 条件分岐や繰り返し, 配列/リスト, 関数の基本

的な知識はもとより, 入れ子 (ネスト) による手順 (サンプル問題では「while—for—if」の三重の入れ子が使用されている) や, 論理演算子を使用した条件式の組合せについて理解していなければ解答することは難しい。

2.3 共通テスト手順記述標準言語 (DNCL)

試作問題およびサンプル問題では, 図 1 のように Python の表記に似た疑似言語が使用されているが, 令和 4 年 7 月現在, 共通テスト「情報」に使用されるプログラミング言語の仕様書は大学入試センターより公開されていない。ただし, 試作問題およびサンプル問題に使用されている疑似言語は, 大学入試センター試験/大学入学共通テストの「数学②」の科目「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) [12] の表記に近い。さらに, サンプル問題とともに公開された「サンプル問題のねらい」[9] には, 「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」との記述がある。

これらのことから, 令和 7 年度大学入学者選抜から実施される共通テスト「情報」にも, 試作問題とサンプル問題に使用された疑似言語の表記に近いプログラミング言語が使用されると考えられる。ただし, 「情報関係基礎」に使用されている DNCL と, 試作問題とサンプル問題に使用された疑似言語とは表記が大きく異なっているため, 本稿では前者を「DNCL」, 後者を「DNCL v2」と表記する。

3. 授業実践の概要

本研究では, 「情報 I」の 12 種類の教科書に使用されている Python, JavaScript, VBA, Scratch の 4 つのプログラミング言語を, クラスごとに使い分けて同じ授業内容でプログラミング教育を実践し, 最後の授業で大学入学共通テスト「情報」サンプル問題の第 2 問「プログラミング問題」を全員に解答させた。本章では, プログラミングの授業内容と, 授業で使用したプログラミング言語と実行環境, ならびに授業プリントについて説明する。

3.1 プログラミングの授業内容

本実践は, 愛知県の公立高等学校普通科 1 年生 7 クラス 279 名を対象に, 令和 3 年度 2 学期に「社会と情報」において 8 時間 (50 分 × 8 コマ) かけて行ったものである。なお, 生徒らは, 高等学校入学後に初めてプログラミングを授業で学習することになる。また, 入学時は成績 (入試の結果や中学校の評定など) が均等になるようにクラス分けを行っており, 1 学期の「社会と情報」の成績においても, 各クラスに大きな差はなかった。そのため, 本実践の前段階において, クラス間の学力の差はないものとして考える。

ここで, 本実践における 8 時間分のプログラミングの

```
(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

Kさん: 得票数に比例して配分すると小数点のある人数になってしまう。小数点以下の数はどう考えようか。例えば、A党は2.068966だから2人が当選するのかな。

Mさん: なるほど。切り捨てで計算すると、A党は2人、B党は1人、C党は2人、D党は0人になるね。あれ? 当選者数の合計は5人で、6人に足りないよ。

Kさん: 切り捨ての代わりに四捨五入したらどうだろう。

Mさん: そうだね。ただ、この場合ほどの政党も小数点以下が0.5未満だから、切り捨てた場合と変わらない。だからといって小数点以下を切り上げると、当選者数が合計で9人になるから3人も多くなってしまいます。

Kさん: このままでは上手くいかないなあ。先生に聞いてみよう。

基準得票数: 500
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

問3 次の文章の空欄 [セ] ~ [テ] 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。

Mさん: 図9のプログラムを作ってみたよ。商を整数で求めるところは小数点以下を切り捨てる「切り捨て」という関数を使ったよ。

Kさん: 実行したら図10のように正しく政党名と当選者数が得られたね。

```
(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) | 表示する (Tomei[k], ":", [カ] [キ])
```

図9 各政党の当選者数を求めるプログラム

先生: できたようだね。各政党の当選者数は求められたけど、政党によっては候補者が足りない場合もあるから、その場合にも対応してみよう。図11のように各政党の候補者数を格納する配列 koho を追加してみたらどうだろう。例えば、C党の候補者が足りなくなるように設定してみよう。

A党: 2名
B党: 1名
C党: 3名
D党: 0名

図10 各政党の当選者数の表示

図1 大学入学共通テスト「情報」サンプル問題の第2問 (一部抜粋)

Fig. 1 The Common Test for University Admissions “Information Study” sample problem—Question 2—

授業内容を表3に示す。なお、本校の「社会と情報」はチーム・ティーチングの2名体制で実施しており、すべての授業において筆者が教授役であり、補助役の教員(教科「情報」の教員免許はもたない)が座学や実習のサポートを行っている。また、後述するように、クラスによって使用するプログラミング言語は異なるが、すべてのクラスで表3に示した順序で授業を行い、プログラミング言語以外の要因で授業内容に差が生じないように配慮した。以下、1時間目から8時間目までの授業内容について説明する。

1時間目から6時間目までの6回で、変数、条件分岐、繰り返し、配列/リスト、関数について学習する。次の7時間目に、これまでに学習した内容をふまえた総合演習として「数当てゲーム」のプログラムを作成する。最後に8時間目において、サンプル問題の第2問「プログラミング問題」を解答し、その後、事後アンケートに回答する。

ただし、最後の8時間目の授業でサンプル問題(第2問)に解答するため、7時間目までの授業においてサンプル問題(第2問)に出題されている範囲を学習しておく必要がある。先行研究[13]では、試作問題およびサンプル問題のプログラミング分野に出題された内容を、「①変数」、「②代入」、「③インクリメント」、「④メッセージ」、「⑤異なるデータ型の結合」、「⑥算術演算子」、「⑦比較演算子」、「⑧

表3 プログラミングの授業内容

Table 3 Programming lesson content.

時間	授業内容
1 時間目	変数
2 時間目	条件分岐
3 時間目	繰り返し (for 文)
4 時間目	繰り返し (while 文)
5 時間目	配列/リスト
6 時間目	関数
7 時間目	総合演習「数当てゲーム」
8 時間目	サンプル問題 (第2問) の解答

論理演算子」、「⑨条件分岐」、「⑩繰り返し (for 文)」、「⑪繰り返し (while 文)」、「⑫入れ子 (ネスト)」、「⑬配列/リスト」、「⑭添字 (インデックス)」、「⑮関数」の15個に分類している*2。本実践では、これらすべての項目に関する演習を授業に組み入れるように授業計画を立案した。なお、表3に示したプログラミングの授業内容に記載のない②~⑧、⑩、⑬の9つの項目については、7時間目までの授業

*2 その後の先行研究[14]では、プログラミングの基本要素を15個から13個に再分類している。

において適宜演習に加えている。

3.2 プログラミング言語と実行環境

本実践では、表 3 に示した授業内容を、クラスごとに 4 つのプログラミング言語に分けて実践している。使用したプログラミング言語は「情報 I」の 12 種類の教科書に使用されている Python, JavaScript, VBA, Scratch の 4 つである。ここで、使用したプログラミング言語の割り当てと、プログラミング言語の実行環境について表 4 に示す。

表 4 に示したように、プログラミング言語は 1 組と 2 組 (80 名) が Python, 3 組と 4 組 (80 名) が JavaScript, 5 組と 6 組 (79 名) が VBA, 7 組 (40 名) が Scratch を使用している。なお、VBA と Scratch は表 4 に示した実行環境しか選択肢はないが、Python と JavaScript については表 4 に示したものの以外にもさまざまな実行環境があるため、プログラミング言語とあわせて実行環境を教員が選択する必要がある。ここで、先行研究 [15] では、クラスごとに IDLE, Jupyter Notebook (TextFile 形式/Notebook 形式), Jupyter Lab, Google Colaboratory, Spyder, Visual Studio Code の 7 つの実行環境を使い分けて Python によるプログラミング教育を実践した結果、実行環境によって教育効果に大きな差が生じなかったと結論付けている。JavaScript に関する同様の比較研究は存在しないが、本研究では、JavaScript についても Python と同様の傾向がみられると仮定して、実行環境の構築が容易な Google Colaboratory と Google Chrome (デベロッパーツール) をそれぞれ使用している。

3.3 授業プリント

本実践では、表 4 に示したように 4 つのプログラミング言語を使用しているため、プログラミング言語ごとに異なる授業プリント (7 時間×4 言語) を作成した。ただし、プログラミング言語は異なるが、授業プリントのプログラムは表 3 に示した授業内容と同一である。ここで、7 時間目「総合演習」(数当てゲームのプログラム) に使用した、プログラミング言語ごとの授業プリントを図 2 に示す。なお、図 2(a) は Python, 図 2(b) は JavaScript, 図 2(c) は VBA, 図 2(d) は Scratch の授業プリント (一部抜粋) である。

図 2 のように、授業で扱うプログラミング言語によるプログラムを授業プリントの左側、これを DNCL v2 で書き直したプログラムを授業プリントの右側に記載している。なお、令和 3 年度 2 学期の授業実践当時、共通テスト「情報」に使用される DNCL v2 の仕様書は大学入試センターより公開されていなかったため、授業プリントの右側の DNCL v2 は、情報関係基礎 [16] および試作問題/サンプル問題のプログラムを参考に、筆者の推測の元で書き直している (令和 4 年 7 月現在、仕様書は公開されていない)。

表 4 プログラミング言語と実行環境

Table 4 Programming language and execution environment.

クラス	人数	言語	実行環境
1 組	40 名	Python	Google Colaboratory
2 組	40 名		
3 組	40 名	JavaScript	Google Chrome デベロッパーツール
4 組	40 名		
5 組	39 名	VBA	Excel VBE
6 組	40 名		
7 組	40 名	Scratch	オンライン版 Scratch (Scratch 3.0)

このように、本研究では、授業プリントにおいて各プログラミング言語と DNCL v2 との対応をつねに比較できる状態にしており、サンプル問題 (第 2 問) を解く 8 時間目までに「DNCL v2 に特化した演習」は実施していない。なお、生徒は各プログラミング言語よりも併記された DNCL v2 で学習した部分が多くなっていることも考えられるが、本実践では、いずれのプログラミング言語でも実際にプログラムのコードを入力して実行結果を得ているため、DNCL v2 に偏って理解している生徒の割合は低いと推察する。

4. サンプル問題の解答結果に対する考察

サンプル問題の第 2 問は 3 つの問から構成され、問 1 は「ア」~「ウ」の 3 問、問 2 は「エ」~「ス」の 10 問、問 3 は「セ」~「テ」の 6 問、合計 19 問となっている。サンプル問題に配点は記載されていないが、1 問 1 点の 19 点満点と設定して合計点を算出した。ただし、大学入試センターが公開している「サンプル問題『情報』正解表」[17] には、「イ」と「ウ」、「タ」と「チ」、「ツ」と「テ」は「全問正解の場合のみ点を与える」との記載があるため、これらは 2 つが正解の場合のみ 2 点としている。

以下、本章では、サンプル問題の解答結果から、プログラミング言語がサンプル問題の解答に及ぼす影響について考察する。なお、サンプル問題の解答者数は 279 名 (有効回答率 100%) である。

4.1 全体に対する平均点の比較

全体 (全クラス) に対して、各プログラミング言語の平均点 (M) および標準偏差 (SD) を比較した結果を図 3 に示す。図 3 の「合計点」の傾向として、平均点が高い順に VBA の 7.29 点、続いて Scratch の 7.28 点となり、両者にはほとんど差がない結果となっている。さらに Scratch に続いて Python の 6.89 点、最後に JavaScript の 6.33 点となっている。なお、全体 (全クラス) の平均点は 6.90 点であった。このように平均点だけを比較すると、VBA と

```
import random
pc = random.randint(1,10)
while True:
    you = input("数字を入力")
    you = int(you)
    if you > pc:
        print(you,"よりもアよ")
    elif you < pc:
        print(you,"よりもイよ")
    else:
        print("正解!")
        break
```

(a) Python の授業プリント (一部抜粋)

```
pc = Math.floor(Math.random() * 10 + 1);
while (true){
    var you = prompt("数字を入力");
    you = Number(you);
    if (you > pc){
        alert(you + "よりもアよ");
    }else if (you < pc){
        alert(you + "よりもイよ");
    }else{
        alert("正解!");
        break;
    }
}
```

(b) JavaScript の授業プリント (一部抜粋)

```
pc = Int(Rnd * 10 + 1)
Do
    you = InputBox("数字を入力")
    you = Val(you)
    If you > pc Then
        MsgBox you & "よりもアよ"
    ElseIf you < pc Then
        MsgBox you & "よりもイよ"
    Else
        MsgBox "正解!"
    Exit Do
End If
Loop
```

(c) VBA の授業プリント (一部抜粋)

(d) Scratch の授業プリント (一部抜粋)

図 2 7 時間目「総合演習」に使用したプログラミング言語ごとの授業プリント

Fig. 2 Lesson prints for each programming language used for the 7th hour lesson "Comprehensive Exercise".

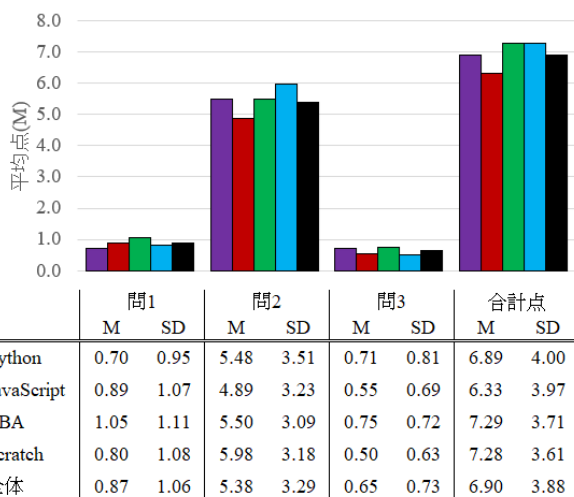


図 3 サンプル問題 (第 2 問) の結果

Fig. 3 Result of the sample problem Q-2.

Scratch が全体の平均点より高く、Python が全体の平均点とほぼ同じ、JavaScript が全体の平均点より低いという結果になっている。

ここで、各プログラミング言語の合計点 (ノンパラメト

リック) に対して Steel-Dwass 検定を行った結果を表 5 に示す。なお、表 5 の値は、すべてのプログラミング言語の組合せに対する p 値を表している。いずれの p 値も有意水準 5% を下回ることがなく、各プログラミング言語の間に有意な差はみられなかった。また、Steel-Dwass 検定を合計点ではなく問 1 から問 3 の各得点に対しても実施したが、同様に有意な差はみられなかった。これらの結果から、授業で扱うプログラミング言語がサンプル問題 (第 2 問) の解答に影響を与える可能性は低いと考える。

4.2 各選択肢に対する正解率の比較

各選択肢「ア」~「テ」の全 19 問に対して、各プログラミング言語の正解率 (%) を比較した結果を表 6 に示す。また、4 つのプログラミング言語の正解率に対してカイ二乗検定 (有意水準 5%) を行った結果、すべての選択肢のうち表 6 の網掛けした 3 カ所の選択肢「ア」、「ケ」、「セ」において有意な差がみられた。以下、有意な差がみられた 3 つの選択肢「ア」、「ケ」、「セ」について考察する。

4.2.1 選択肢「ア」

問 1 の選択肢「ア」は、図 1 左側のプログラムにあるよ

表 5 合計点に対する Steel-Dwass 検定の結果

Table 5 Result of Steel-Dwass test for total points.

	Python	JavaScript	VBA	Scratch
Python	1.000	0.867	0.849	0.900
JavaScript	0.867	1.000	0.312	0.463
VBA	0.849	0.312	1.000	0.900
Scratch	0.900	0.463	0.900	1.000

表 6 各選択肢における正解率の比較

Table 6 Comparison of correct answer rate for each option.

選択肢	正解率 (%)					
	Python	JavaScript	VBA	Scratch	全体	
問 1	ア	30.0	43.8	51.9	40.0	41.6
	イ	36.3	51.3	44.3	37.5	43.0
	ウ	32.5	31.3	34.2	40.0	33.7
問 2	エ	66.3	71.3	83.5	85.0	75.3
	オ	46.3	41.3	45.6	40.0	43.7
	カ	47.5	41.3	44.3	50.0	45.2
	キ	42.5	35.0	44.3	42.5	40.9
	ク	53.8	51.3	51.9	52.5	52.3
	ケ	68.8	66.3	69.6	87.5	71.0
	コ	53.8	38.8	49.4	60.0	49.1
	サ	61.3	55.0	55.7	65.0	58.4
	シ	52.5	43.8	53.2	57.5	50.9
	ス	55.0	45.0	51.9	57.5	51.6
問 3	セ	40.0	31.3	51.9	25.0	38.7
	ソ	18.8	21.3	15.2	20.0	18.6
	タ	15.0	21.3	27.8	25.0	21.9
	チ	7.5	15.0	11.4	15.0	11.8
	ツ	25.0	20.0	21.5	30.0	23.3
	テ	17.5	11.3	12.7	7.5	12.9

うに「得票に比例した各政党の当選者数を求めるプログラム」において、繰り返し (for 文) の終了値を答える問題である。表 6 に示したように、選択肢「ア」では Python が他の正解率と比較して低い結果となっている。ここで、選択肢「ア」を含む DNCL v2 のコードを、各プログラミング言語に書き直したプログラムを表 7 に示す。

変数 m を 0 から 3 まで繰り返し構文でインクリメントする場合、Python では表 7 に示した「range(0, 4)」というように、range の括弧内を「終了値 + 1」と記述しなければならない。なお、JavaScript も条件式に「3」ではなく「4」と表記されているが、「m < 4」のように比較演算子が使用されているため、Python とは異なり「4」が繰り返しの対象にならないことは想像し易いと考えられる。

このように、選択肢「ア」にあてはまる数値は「3」であるが、これを「4」と誤答した割合が Python が最も多

表 7 選択肢「ア」を含むプログラムの比較

Table 7 Comparison of programs including the option "A".

言語	選択肢「ア」を含むプログラム
DNCL v2	m を 0 から 3 まで 1 ずつ増やしながらか繰り返す： └ Hikaku[m] = Tokuhyo[m]
Python	for m in range(0,4): Hikaku[m] = Tokuhyo[m]
JavaScript	for (var m=0; m<4; m++){ Hikaku[m] = Tokuhyo[m]; }
VBA	For m = 0 To 3 Hikaku(m) = Tokuhyo(m) Next m
Scratch	

かったことから、上記のような range の括弧内の書き方が正解率を下げる結果になった可能性があるかと推察する。一方、選択肢「ア」の正解率が最も高かったのは、表 6 にあるように VBA であった。この点については、DNCL v2 の「m を 0 から 3 まで」と VBA の「m = 0 To 3」という表記の対応付けが容易であったことが影響して、VBA の正答率が高くなった可能性があるかと推察する。

4.2.2 選択肢「ケ」

問 2 の選択肢「ケ」は、各政党の当選者数を求める際の配列の変化の過程 (2 回目の手順 3 終了時) において、配列 Tosen [2] の値を答える問題である。表 6 に示したように、選択肢「ケ」では Scratch が他の正解率と比較して最も高い結果となっている。なお、問 2 の選択肢「エ」～「ス」の 10 問はすべて配列の値を答える問題であり、問 2 (10 問) 全体の正解率は Python が 54.8%、JavaScript が 48.9%、VBA が 54.9%、Scratch が 59.8% となっており、こちらも Scratch が最も高い値を示している。Scratch では、他のテキスト型のプログラミング言語とは異なり、図 4 に示したように配列/リストの中身がリアルタイムで表示される仕様になっているため、サンプル問題 (第 2 問) の解答においても配列の変化の様子をイメージすることができ、高い正解率につながったのではないかと推察する。

4.2.3 選択肢「セ」

問 3 の選択肢「セ」は、図 1 右側のプログラムにあるように、各政党の当選者数を求めるプログラムにおいて、繰り返し (while 文) の条件式を答える問題である。表 6 に示したように、選択肢「セ」では VBA が他の正解率と比



図 4 Scratch におけるリストの表示例
Fig. 4 List display example in Scratch.

表 8 選択肢「セ」を含むプログラムの比較

Table 8 Comparison of programs including the option “N”.

言語	選択肢「セ」を含むプログラム
DNCL v2	tosenkei < giseki の間繰り返す:
Python	while tosenkei < giseki:
JavaScript	while (tosenkei < giseki){
VBA	Do While tosenkei < giseki
Scratch	

較して高い結果となっている。ここで、選択肢「セ」を含む DNCL v2 のコードを、各プログラミング言語に書き直したプログラムを表 8 に示す。

表 8 に示したように、各プログラミング言語のプログラムを比較しても、DNCL v2, Python, JavaScript, VBA には特に大きな違いはなく、プログラムの表記が正解率に直接影響したとは考えにくい。一方、Scratch に関しては、条件式の表記が他のプログラミング言語と比較して大きく異なる。これは、Scratch に While 文のブロックが用意されていないため、表 8 のように Until 文を使用して条件式を反転させなければならないことが理由である。このことが影響して、表 6 のように Scratch の正答率が最も低くなった可能性があるかと推察する。

5. 事後アンケート結果に対する考察

7 時間目の授業「総合演習」の最後に、全員に事後アンケートに回答させている。本章では、事後アンケートの項目のうち、サンプル問題（第 2 問）の解答後に感じた難易度と、解答時間の長さに対する回答結果を取り上げて考察する。なお、事後アンケートの回答者数は 279 名（有効回答率 100%）である。また、サンプル問題（第 2 問）の解答

時間に関して、サンプル問題の表紙 [8] には「実際の問題セットをイメージしたものや試験時間を考慮したものでもありません」と記載がある。そのため、本研究では、サンプル問題（第 2 問）の解答時間は余裕を見積もって 30 分に設定している。

5.1 サンプル問題を解いて感じた難易度

サンプル問題（第 2 問）の解答後に感じた難易度を「とても簡単」、「簡単」、「普通」、「難しい」、「とても難しい」の五択で回答させた結果を図 5 に示す。図 5 の全体（全クラス）の傾向として、割合が大きい順に「とても難しい」の 67.0%、「難しい」の 26.9%、「普通」の 4.3%、「とても簡単」の 1.4%、「簡単」の 0.4%となっており、9 割以上の生徒がサンプル問題（第 2 問）に対して「難しい」あるいは「とても難しい」という印象を抱いていることが分かる。ここで、図 5 に示した各プログラミング言語の割合に対してカイ二乗検定（有意水準 5%）を行った結果、いずれにおいても有意な差はみられなかった。そのため、授業で扱うプログラミング言語によって、サンプル問題（第 2 問）を解いて感じた難易度に差は生じないと考えられる。

5.2 サンプル問題の解答時間の長さ

サンプル問題（第 2 問）の解答後に感じた解答時間の長さを「短い」、「ちょうどよい」、「長い」の三択で回答させた結果を図 6 に示す。図 6 の全体（全クラス）の傾向として、割合が大きい順に「ちょうどよい」の 67.0%、「長い」の 17.6%、「短い」の 15.4%となっており、おおむね 7 割前後の生徒が 30 分という解答時間が妥当であるという印象を抱いていることが分かる。ここで、図 6 に示した各プログラミング言語の割合に対してカイ二乗検定（有意水準 5%）を行った結果、Scratch にのみ有意な差がみられた ($p = 0.011$)。つまり、Scratch は他のプログラミング言語と比較して、解答時間が「短い」と回答した割合が多かったといえる。考えられる要因として、事後アンケートの質問項目「授業プリントの左側と右側のプログラムではどちらの方が分かりやすかったか」に対する回答結果を図 7 に示す。

図 7(a)を見ると Python では 52.5%、図 7(b)を見ると JavaScript では 28.7%、図 7(c)を見ると VBA では 31.6%、図 7(d)を見ると Scratch では 85.0%の生徒が「授業で使ったプログラミング言語の方が分かりやすかった」と回答している。このように Scratch では、他のプログラミング言語と比較して「授業で使用したプログラミング言語」と答えた割合が明らかに高い。これらの結果から、Scratch でプログラミングを学習した生徒は、授業でブロック型のプログラムに慣れたことで、サンプル問題（第 2 問）の解答では DNCL v2 を「読む」ことに時間がかかってしまったのではないかと推察する。一方、Scratch はブロック型

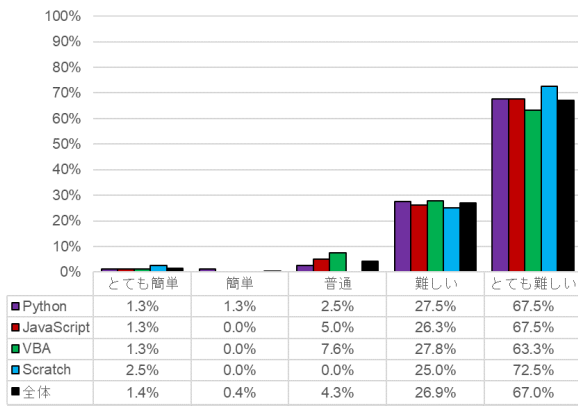


図 5 解答後に感じた問題の難易度

Fig. 5 Difficulty of the question felt after answering.

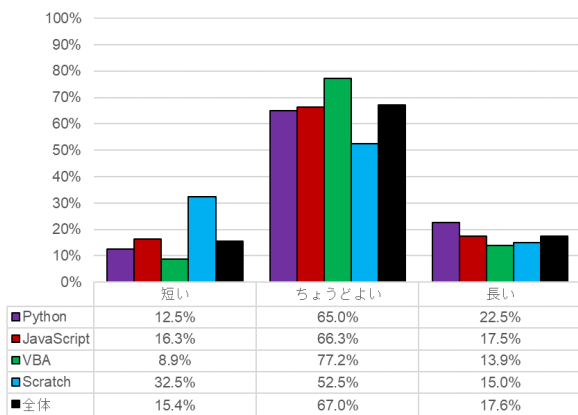


図 6 解答後に感じた解答時間の長さ

Fig. 6 The length of answer time felt after answering.

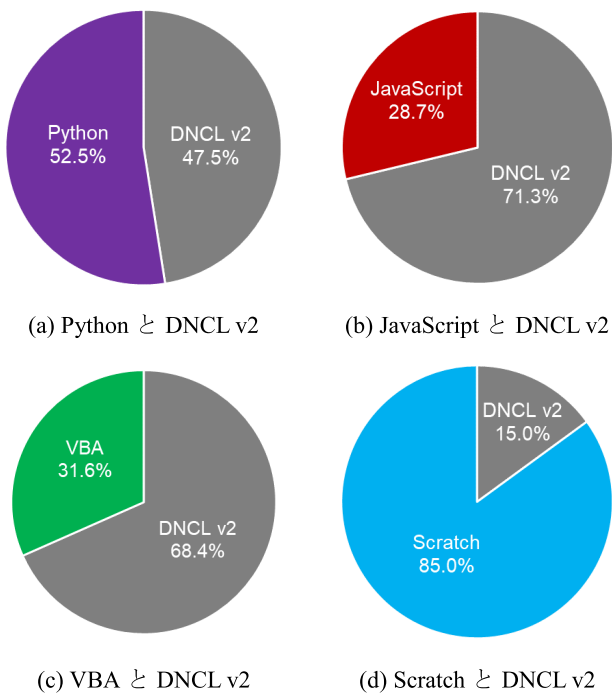


図 7 分かりやすかった方のプログラミング言語

Fig. 7 Easy-to-understand programming language.

かつ日本語表記のプログラミング言語であるため、生徒にとって視覚的にプログラムを理解しやすいというメリットもある。また、4.2.2 項で説明した問 2 の「配列の値を答える問題」のように、プログラムの空欄を埋める問題ではなく、プログラムの構造や意味を問うような問題の場合、Scratch の方が有利であることも推察できる。

6. おわりに

本研究では、クラスごとに Python, JavaScript, VBA, Scratch の 4 つのプログラミング言語を分けて同じ内容でプログラミング教育を実践し、最後の授業でサンプル問題の第 2 問「プログラミング」を全員に解答させた。さらに、サンプル問題 (第 2 問) の解答結果や事後アンケートから、授業で扱うプログラミング言語によってサンプル問題 (第 2 問) の解答に関して差が生じるのかどうか検証した。

まず、サンプル問題 (第 2 問) の解答結果から、各プログラミング言語において多少の平均点の違いはみられたが、Steel-Dwass 検定の結果では各プログラミング言語の間に有意な差はみられなかった。なお、これは第 2 問の合計点 (19 点満点) だけでなく、問 1 (3 点満点)、問 2 (10 点満点)、問 3 (6 点満点) に対する Steel-Dwass 検定の結果でも、同様に有意な差はみられなかった。これらの結果から、少なくともサンプル問題 (第 2 問) に関しては、授業で扱うプログラミング言語によって解答結果に有意な差が生じなかったといえる。ただし、各プログラミング言語の正解率に対するカイ二乗検定の結果、一部の選択肢において有意な差がみられた。その要因として、たとえば、for 文による繰り返しの表記がプログラミング言語で大きく異なることが考えられる。このように、プログラミング言語によって DNCL v2 との表記が大きく異なる部分もあるため、授業ではそのような表記の違いについてもしっかり説明する必要があると考える。

次に、サンプル問題 (第 2 問) の解答後に感じた難易度に関する事後アンケートの結果から、9 割以上の生徒がサンプル問題 (第 2 問) に対して「難しい」あるいは「とても難しい」という印象を抱いていることが分かった。また、プログラミング言語によって回答結果に多少の違いはみられたが、各プログラミング言語の割合に対するカイ二乗検定の結果、有意な差はみられなかった。これらの結果から、少なくともサンプル問題 (第 2 問) に関しては、授業で扱うプログラミング言語によって解答後に感じた難易度に有意な差が生じなかったといえる。

最後に、サンプル問題 (第 2 問) の解答後に感じた解答時間の長さに関する事後アンケートの結果から、おおむね 7 割前後の生徒が 30 分という解答時間が妥当であるという印象を抱いていることが分かった。また、プログラミング言語によって回答結果に多少の違いはみられたが、各プログラミング言語の割合に対するカイ二乗検定の結果、

Python, JavaScript, VBA については有意な差はみられなかった。一方, Scratch にのみ有意な差がみられ, 他のプログラミング言語と比較して解答時間が「短い」と回答した割合が多かったといえる。Scratch でプログラミングを学習した生徒は, 授業でブロック型のプログラムに慣れたことで, サンプル問題 (第 2 問) の解答では DNCL v2 を「読む」ことに時間がかかってしまったのではないかと推察する。ただし, Scratch は生徒にとって視覚的にプログラムを理解しやすく, 問題の内容によっては Scratch の方が解答のうえで有利になることも考えられる。

以上の結果から, 「情報 I」の 12 種類の教科書に使用されている Python, JavaScript, VBA, Scratch の 4 つのプログラミング言語に関して, どのプログラミング言語を授業で使用していても, 少なくともサンプル問題 (第 2 問) を解答するうえで大きな影響を及ぼす可能性は低いと推察する。ただし, 各プログラミング言語が持つ特有の表記や仕様により, 少なからず解答に影響が生じていることも明らかとなった。たとえば, for 文のように, プログラミング言語によって DNCL v2 と表記が大きく異なる部分もあるため, 授業ではそのような表記の違いについてもしっかり説明する必要がある。特に Scratch は, ブロック型プログラミング言語であるため, DNCL v2 と表記が異なる部分が多く, 共通テスト「情報 I」に出題される DNCL v2 を「読む」ための演習が他のプログラミング言語と比較して多く必要であると考えられる。今後は「情報 I」において, プログラミング教育の効果的な授業の展開方法や, 共通テストへの効果的な対策に関して実践および考察を行っていきたい。

本研究の一部は, 日本産業技術教育学会第 64 回全国大会, 第 17 回情報教育合同研究会, 日本情報科教育学会第 13 回フォーラム, 情報処理学会コンピュータと教育研究会第 163 回研究発表会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:平成 30 年告示高等学校学習指導要領, 入手先 (<https://www.mext.go.jp/content/1384661.6.1.3.pdf>) (参照 2022-01-10)。
- [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, 入手先 (https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html) (参照 2022-01-10)。
- [3] 大学入試センター: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, 入手先 (https://www.mext.go.jp/content/20210729-mxt_daigakuc02-000005144.2.pdf) (参照 2022-01-10)。
- [4] 大学入試センター: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 (補遺), 入手先 (https://www.mext.go.jp/content/20210929-mxt_daigakuc02-000005144.1.pdf) (参照 2022-01-10)。
- [5] 大学入試センター:令和 7 年度大学入学選抜に係る大学入学共通テストにおける得点調整の対象教科・科目に

- ついて, 入手先 (https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html) (参照 2022-01-10)。
- [6] 国立大学協会:2024 年度以降の国立大学の入学選抜制度—国立大学協会の基本方針, 入手先 (<https://www.janu.jp/wp/wp-content/uploads/2022/01/20210128-news.001.pdf>) (参照 2022-01-28)。
- [7] 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大学入学共通テストの『情報』の試作問題 (検討用イメージ), 入手先 (<https://www.ipsj.or.jp/education/9faeag0000012a50-att/sanko2.pdf>) (参照 2022-01-10)。
- [8] 大学入試センター:サンプル問題『情報』問題, 入手先 (https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html) (参照 2022-01-10)。
- [9] 大学入試センター:サンプル問題『情報』ねらい, 入手先 (https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html) (参照 2022-01-10)。
- [10] 水野修治:大学入学共通テスト新科目案「情報」について—これまでの経緯と試作問題 (検討イメージ) のねらい, 情報処理学会コンピュータと教育研究会第 158 回研究発表会招待講演, 入手先 (<https://www.wakuwaku-catch.net/kouen210303/>) (参照 2022-01-10)。
- [11] 井手広康:「情報 I」教科書の動向, 情報処理学会高校教科「情報」シンポジウム 2021 秋, 入手先 (<https://www.wakuwaku-catch.net/kouen211101/02/>) (参照 2022-01-10)。
- [12] 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, 入手先 (<https://www.dnc.ac.jp/albums/abm00040701.pdf>) (参照 2022-01-10)。
- [13] 井手広康:大学入学共通テスト「情報」サンプル問題を踏まえた情報 I の教科書におけるプログラミング分野の比較, 情報処理学会情報教育シンポジウム 2021 (SSS2021) 論文集, pp.246-253 (2021)。
- [14] 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, Vol.8, No.3, pp.8-18 (2022)。
- [15] 井手広康:高等学校におけるプログラミング環境の違いによって生じる教育効果に関する比較, 日本産業技術教育学会論文誌, Vol.63, No.2, pp.269-278 (2021)。
- [16] 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/ipsjnu/resources/JHK>) (参照 2022-01-10)。
- [17] 大学入試センター:サンプル問題『情報』正解表, 入手先 (https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html) (参照 2022-01-10)。



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。