

ブロックプログラミングによるプログラミング的思考の涵養*

— 本学における実践を中心として —

礒 本 光 広**

Cultivating Programmatic Thinking through Block Programming:
Focusing on Practice at the NWU

Mitsuhiro ISOMOTO**

要旨 高校において数学を積極的に学習しておらず、プログラミング学習機会もほとんどなかった文系大学生にたいして難易度の低いブロックプログラミングを実践することは、数学的思考を理解するとともにプログラミング的思考の基礎を学習するうえで非常に有用であった。またその後のテキストプログラミング学習に軟着陸でき、Python学習の基礎にもつなげることができた。これらの学習方法は、今後の本学におけるデータサイエンス教育の道筋の一端となることが期待される。

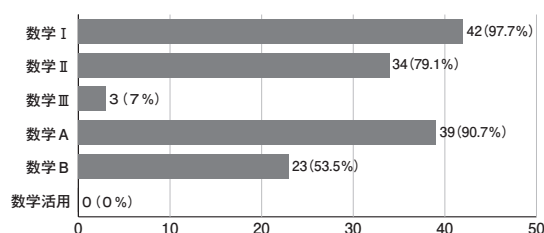
キーワード：ブロックプログラミング、Python、プログル、プログラミング的思考、数学的思考

1 数学およびプログラミングの履修状況

1-1 現状把握

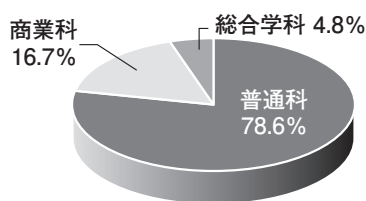
データサイエンス教育の必要性が叫ばれ、大学においても実施する必要に迫られている。そこで1年生の授業において数学とプログラミングの履修についてのアンケートを実施した結果が図表1および図表2である (N=42)。なお、トータルビジネス科および総合ビジネス科は商業科に加えてある。また数学の履修については履修科目をすべて回答する方式によった。

図表2 高校時の数学の履修状況



(出典) 本論文筆者作成。

図表1 出身学科別割合



(出典) 本論文筆者作成。

本学経済政策学科の学生ということで高校時代には文系コースだったことが推察され、数学I、数学A、数学IIを履修している学生が大半であることが読み取れる。一方で数学IIIの履修は3名、数学活用は0名という結果であった。また情報系の授業における学習は、普通科高校のみならず商業高校においてもMs-WordやMs-Excelの利用に

* Received September 26, 2022

** 鎮西学院大学 現代社会学部 経済政策学科 Faculty of Contemporary Social Studies, Nagasaki Wesleyan University, 1212-1 Nishieida, Isahaya, Nagasaki 854-0082, Japan

とどまっており、JavaおよびVBA¹をしたことがある学生が各1名ずつという悲惨な結果となった。本学学生の大半が高校までにプログラミング経験がないことが明らかとなった。

高校時代に数学を積極的に学習しておらず、プログラミングもおこなっていない文系大学生にたいする教材としてブロックプログラミング (Block Programming) の導入を検討することにした。大学生用の教材ではないことは重々承知しているが、指導方法によっては大学生に有用な教材となるのではないかと考えたからである。そこでプログルならびにプログル情報をもちいて文系大学生にたいして数学的思考とプログラミング的思考を涵養する実践を試みる。

1-2 先行研究

浅井²がブロックプログラミング環境に基づく学習支援環境の構築とその有効性について研究しているほか、武富³と水谷ら⁴が特別支援教育における利用研究をしているが、大学における実践例は非常に少ない。大学においてデータサイエンス教育が必修となる状況もあり、今後増加することが期待される。

1-3 ブロックプログラミングとは

ブロックプログラミングはビジュアルプログラミング (Visual Programming) と呼ばれ、プログラムをテキスト (コード) で記述するのではなく、アイコンや絵など視覚的なオブジェクトや

ブロックの組み合わせで記述するものである。「グラフィカルプログラミング (Graphical Programming)」と呼称されることもある⁵。ブロックプログラミングには、マサチューセッツ工科大学が開発に関与した「Scratch (スクラッチ)」、文部科学省が開発した「プログラミン」、Microsoftが開発した「Minecraftプログラミング入門 (Code.org)」、「Viscuit (ビスケット)」、「MOONBlock (ムーンブロック)」、「Google Blockly (グーグル ブロックリー)」など多くのものがある。今回はPython (パイソン) への接続を考えて「プログル⁶」を利用することとした。

1-4 プログルとは

プログルは、児童が自分たちの力で取り組むことができるドリル型の学習教材であり、課題をクリアしながらステージを進めていくだけで自然とプログラミング的思考が身につくように設計されている⁷。このような設計により、算数学習のドリルとして利用することも可能であるが、プログラミング教育の導入として利用することも可能となっている。たとえば、多角形コース、公倍数コースでは基本的な概念である「順次処理」「反復処理」を身につけることが可能であり、平均値コース、最頻値コース、中央値コースでは、それまでに学習した代表値の求め方を画面上で視覚的に捉え直すことが可能となる。加算をするときにそれぞれの数字を加算するのではなく、作業領域に加算していくなど、人間が計算するときとコン

¹ Visual Basic for Applicationsの略。主にマイクロソフト製のMicrosoft Officeシリーズに搭載されており、Word、Excel、PowerPointなどのソフトウェア上でプログラムを作成・実行することで、それらソフトウェアの機能拡張や自動化を可能とする簡易的なプログラミング言語である。

² 北川 舞、浅井健一「OCaml 初学者の学習調査」『プログラミングおよびプログラミング言語ワークショップ論文集』第23号, 1-16頁, 2021年。

松本晴香、浅井健一「Blockly をベースにした OCaml ビジュアルプログラミングエディタ」『プログラミングおよびプログラミング言語ワークショップ論文集』第21号, 1-15頁, 2019年。

³ 武富志郎「知的障害特別支援学校小学部児童に対するロボット教材を用いたプログラミングの指導に関する研究」『日本教育工学会研究報告集』2022-1号, 70-77頁, 2022年5月。

⁴ 水谷好成、福井恵子、梅津直哉、米谷年法「特別支援教育におけるプログラミング教育を含む情報関連学習の段階的な実践」『宮城教育大学情報活用能力育成機構研究紀要』第1号, 77-88頁, 2021年3月。

⁵ 大塚商会ウェブサイト「ビジュアルプログラミング」『IT用語辞典』、<https://www.otsuka-shokai.co.jp/words/visual-programming.html>, [2022年9月25日閲覧]。

⁶ 「特定非営利活動法人みんなのコード」が提供する無料のブロックプログラミング教材である。小学生、中学生、高校生のコース (プログル情報) があるが、今回は小学生のコース、ならびに高校生のコースを利用した。

⁷ プログルウェブサイト「プログルとは」<https://proguru.jp/>, [2022年9月25日閲覧]。

コンピュータに計算させるときの違い、すなわちプログラミング的思考を養うことが可能となる。

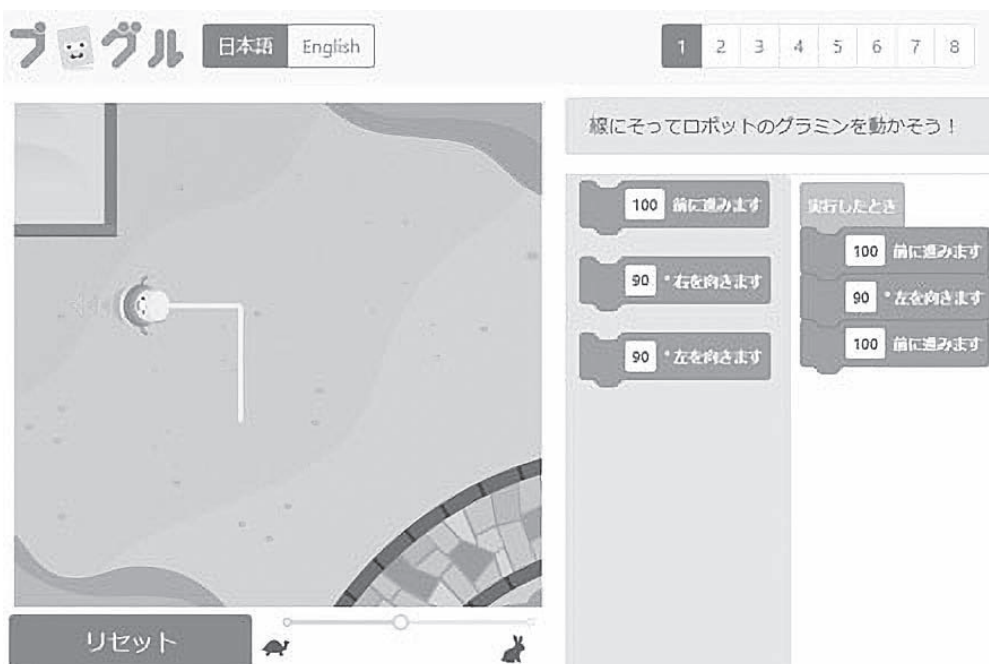
2 プログルの実践

プログルは登録不要であり、パスワード設定等をする必要もないために非常に利用しやすい。最初にこれから学習する内容の説明を読み、それから実践するスタイルである。問題解決の糸口がみつからないときは、ヒントを読むこともできる。

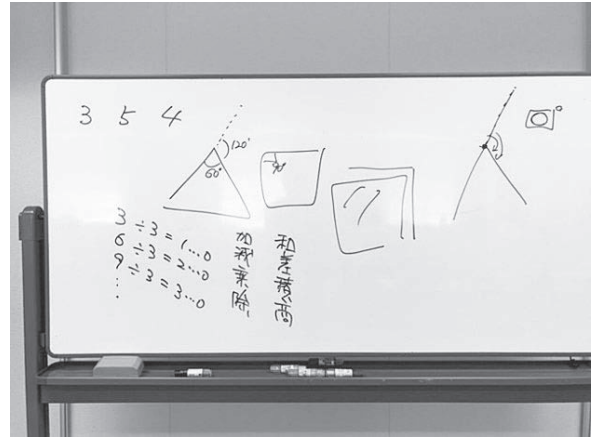
今回は難易度の低い内容をあえて利用することで、数学への抵抗感をなくしプログラミング的思考を涵養することに重点を置いた。そのときに大事なことはまず仮説を立ててから検証することをしっかりと守らせることである。これを怠るとただのブロック遊びになってしまう。



図表3 プログル（多角形コース）の画面



(出典) プログルウェブサイト、<https://proguru.jp/course/turtle/#/1>, [2022年9月25日閲覧]。



2-1 多角形コース

コンピュータは有能で、間違った指示を出してもきちんと動作すると考えている学生が一定数存在する。しかし、そんなことはないことをまず理解させる。小さな子どもと同様に、細かくていねいに指示を出してあげないとできないことも体感させる（図表3）。

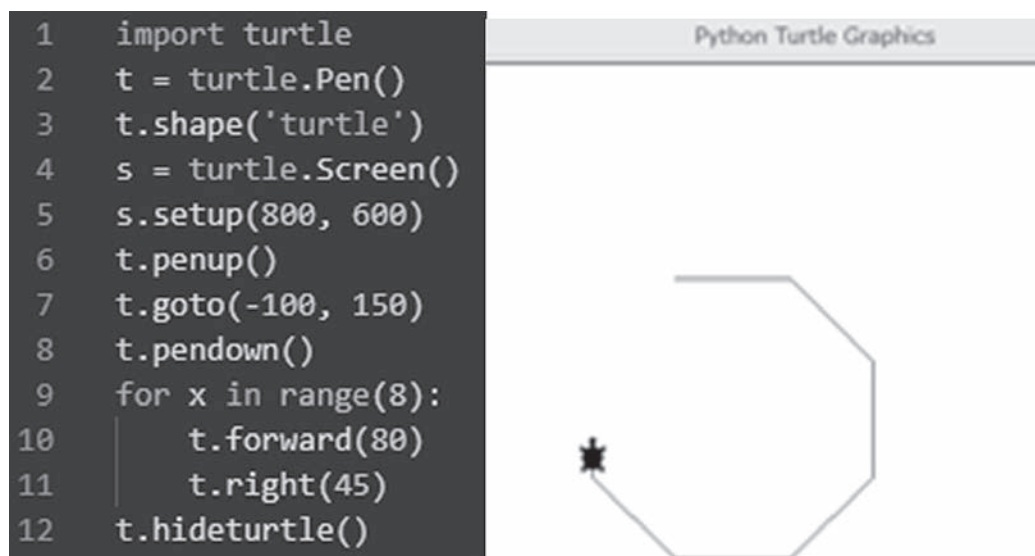
ステージ3において同様の処理を繰り返させ、もう少し簡単な指示で動かないだろうかと感じさせる。そしてつぎのステージで反復処理の図形が登場する。これはC言語やJavaにおいてはwhile文、do文、for文などに相当し、COBOL（コボル）においてはperform文に相当する。ブロックプログラミングをするうえでこの形が反復処理をあらわすということが理解でき、反復処理が便



利だと体感していれば、そして当て推量でブロック遊びをしているのであればテキストプログラミングに容易に移行することが可能となる。

ステージ4では四角形を描かせる問題である。Pythonには標準でTurtle（タートル）と呼ばれるプログラミング学習用のグラフィックライブラリが用意されている⁸（図表4）が、プログルではPythonをインストールせずにTurtleでのグラフィックと同様のプログラミング的思考が体験できる設計となっている。

図表4 PythonによるTurtleのコーディングとその結果



（出典）Pythonをもちいて本論文筆者作成。

2-2 平均値コース

10、40、33の平均を求めることは簡単であろう。しかし代数的に表記することは意外に難しい。一生懸命考えて $(X_1 + X_2 + X_3 + \dots + X_n) \div n$ と表記しても式とはいえない。 $\bar{x} = \sum_{k=1}^n (X_k) \div n$ と表現できればよいのであろうが、文系大学に進学している学生にたいしてそれは無理な注文である

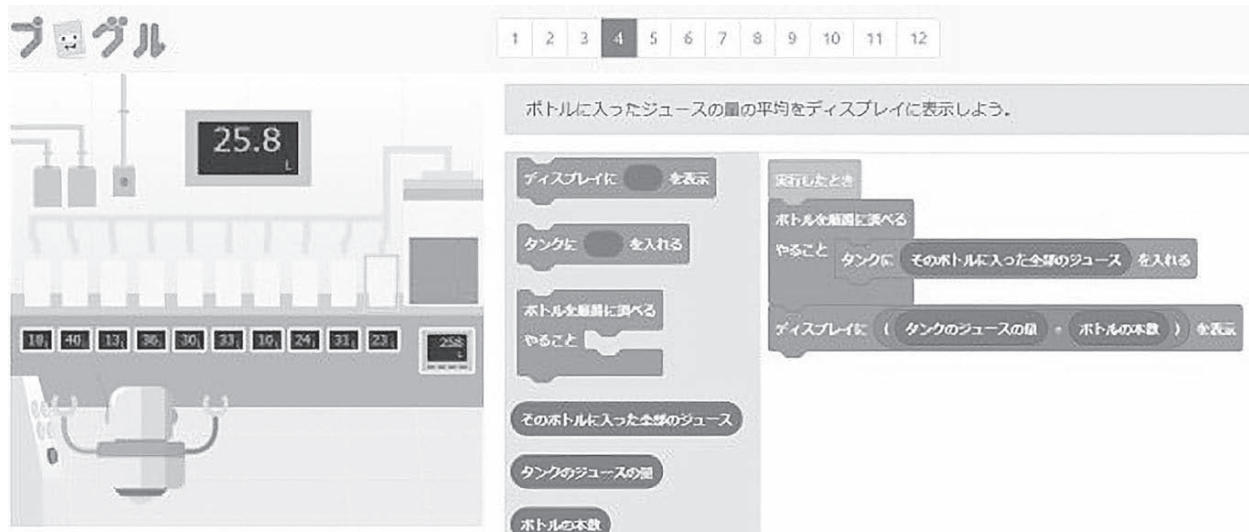
う。しかしプログルを利用することにより段階的に数学的思考、プログラミング的思考を身に付けることが可能であると考え。

ステージ4では平均を求める。図表5および図表6はプログルによるプログラミングとC言語によるプログラミングの一例である。

⁸ タートルグラフィックスは子供向けのプログラミングに入門でよくもちいられ、ウォーリー・フェールゼーグ (Wally Feurzeig)、シーモア・パパート (Seymour Papert)、シンシア・ソロモン (Cynthia Solomon) が1967年に開発したオリジナルのLogoプログラミング言語の一部分である。

Python Software Foundationウェブサイト「turtle: タートルグラフィックス」, <https://docs.python.org/ja/3/library/turtle.html>, [2022年9月25日閲覧]。

図表5 平均を求めるプログラム（プログル）



(出典) プログルをもちいて本論文筆者作成。

図表6 平均を求めるプログラム（C言語）

```
#include <stdio.h>
int main(void){
    int i, sum;
    float avg;

    sum = 0;
    for(i=1; i<=10; i++){
        sum = sum + i;
    }
    avg = sum / 10.0;

    printf("平均 : %.1f\n", avg);
    return 0;
}
```

```
#include <stdio.h>

int main(){
    int i, n v[10];
    float avg;

    printf("n = ");
    scanf("%d",&n);

    int sum=0;
    for(i=0; i<n; i++){
        scanf("%d",&v[i]);
        sum += v[i];
    }
    avg = sum / n;

    printf("平均: %f\n ", ave);
    return 0;
}
```

(出典) 本論文筆者作成。

配列を利用した場合とそうでない場合を挙げてみたが、どちらにしても「すべての数の合計」と「個数」が必要なことがわかる。指導者にプログラミングについての素養があれば、for文やwhile文をもちいてプログラミングしなくても、またこのような低学年向けの教材であっても、プログラミング思考を学習することは十分に可能である。

2-3 公倍数コース

「こんにちは」と表示させるプログラムは一般的に「Hello World（ハローワールド）」と呼ば

れ、伝統的にプログラミング言語をプログラム初心者で紹介するために使われる。大抵のプログラミング言語の入門書では、このプログラムを動作させることを最初の例題としているため目に触れる機会も多い。実際には簡単なコーディングで実行結果が確認できるため、プログラム言語の動作確認にもちいられるという一面もある。¥nは改行をしておいたほうがいろいろと汎用性が高いのでいつもつけているだけで、1行しか表示しない場合にはなくても構わない。図表7はプログルおよびC言語によるプログラミングの比較である。

図表 7 Hello Worldの比較

実行したとき

こんにちは と言います

```
#include <stdio.h>

main()
{
    printf("こんにちは\n");
}
```

(出典) 本論文筆者作成。

ステージ4において1から10までいわせる問題が出題されている。for文で何度も反復処理させながら、「今の数をいう」という考え方がプログ

ラミング的思考であり、それに慣れることが肝要である。図表8はプログルおよびC言語によるプログラミングの比較である。

図表 8 反復処理文の比較

実行したとき

1から 10 まで

順に数を調べる

やること 今の数を言います

```
#include <stdio.h>
int main(void){
    for(int i = 1; i <= 10; i++){
        printf("%d\n", i);
    }
}
```

(出典) 本論文筆者作成。

ステージ6も考えさせる問題である。3の倍数を聞くと回答できる学生も、3の倍数を求めるために必要なのは3で割った余りなのか、3で割った商なのか、3をかけた積なのか、そして3をたした和なのかを聞かれると面食らってしまう。コ

ンピュータに指示を出すためには、指示を細かく分解する必要がある。そして表現方法を変えるためには、物事をきちんと理解している必要がある(図表9)。

図表 9 3の倍数のプログラミング

(出典) プログルをもちいて本論文筆者作成。

蛇足ながら、和差積商という用語が使われているので、調べてみると現在は小学校4年で学習することになっている⁹。加減乗除は知っている、これらは意外と覚えていないことが多いので確認することが大事である。

ステージ14もまた考えさせる問題である。3の倍数、5の倍数、3と5の公倍数の順序を正しく回答しないと正答とはならない。理解しているかどうかを確かめるためには非常によい問題といえるであろう。

2-4 プログルの実践を振り返って

今回、ほとんどの学生が初めてプログラミングをしたと回答しているが、楽しかったととらえている学生が大半であったことは非常に評価できる。どのような学習にも共通することであるが、最初に嫌悪感を抱かせてしまうとそれを覆すのは容易なことではない。そういう意味では非常によい滑り出しである。

また、処理させることを細かく指示を出したり、どの順番で指示を出したり、またどうすれば実現できるのかを考えるのが難しかったと回答した学生も多かった。これは要素分解して手順を考えることすなわち論理的思考のことをいっているので、このような回答をしている学生がきちんと考えながら実践をおこなっていたことを意味する。論理的思考で実現方法を考えたうえで最善手を選ぶのがプログラミング的思考といえるので、今回の実践は成功したといえるのではないだろうか。

またできない、わからない体験をした学生も、実際にできなかった原因は自分が指示をよく読んでいない、もしくは理解していないことであると体感出来たようである。できなかった場合にも、なぜできなかったのか、どうすればできるようになるのかを体感できたことは非常に有意義である。

反省点をあげると、ひとつめに「間違えたときに何度も『60度右を向く』と指示を出した」と回答した学生がいたことである。間違えたときにはまず何が間違えたのかをもう少し考えさせるよう

な指示が必要であった。もし正答してもきちんと考えたうえでの結果でなければあまり意味はない。ふたつめに「数学がわからないとプログラミングはできない」と回答した学生がいたことである。今回の教材は数学を学習するための教材であることをきちんと説明しきれていなかったことの証左である。数学は確かにプログラミングには大事なことであるが、高度なものでなければその限りではない。きちんと説明すべきことであった。

3 プログル情報の実践

3-1 Pythonとプログル情報

データサイエンスにおいてもちいられるプログラミング言語としてPythonやRの人気の高い。ここではもっとも人気のあるPythonについて説明する。Pythonはライブラリが豊富なことが利点のひとつであるが、それらを事前にインストールするとなると環境設定に時間がかかるし、学生にインストールさせようとするセキュリティその他の問題が起きる。その点、プログル情報ではそれらのことにわずらわされずに使用できることが利点としてあげられる。

プログル情報は、高等学校情報科「情報I」で使うことを想定して作成された無償のドリル型のプログラミング教材である。身近なWebサービスを題材にしてテキスト言語が初めての人にもわかりやすくPythonを学べるように作られている¹⁰。プログル情報はプログルと異なり事前登録やパスワード設定が必要となっているため、利用するのに若干敷居が高い。本学においては学生が半角/全角もしくは大文字/小文字をきちんと理解できていなかったために、パスワード設定作業において非常に手間取った。しかしそのアクティビティにより高校時代までのコンピュータの習熟度具合が把握できたともいえる。

⁹ 履修年度によっては中学生で学習した学生もいる。

文部科学省『小学校学習指導要領（平成29年告示）解説算数編』2019年7月。

文部科学省『小学校学習指導要領（平成20年告示）解説算数編』2008年6月。

文部科学省『小学校学習指導要領（平成10年告示）解説算数編』1998年12月。

¹⁰ プログル情報ウェブサイト「プログル情報とは」<https://high.proguru.jp/>, [2022年9月25日閲覧]。

3-2 チャットボット

まずチャットボット¹¹を連想させるロボットと会話するプログラムを作成する。最初に説明文が

あり、それを理解して進めていくのはプログラムの同様の様である。解答に困ったらヒントが準備されているのも同様である。

図表10 Hello World



(出典) プログラム情報をもちいて本論文筆者作成。

レッスン1-1はブロックプログラミング（プログラムの公倍数コースレッスン4、図表7）でも登場したHello Worldである（図表10）。プログラムとよく似た課題をもちい、ブロックプログラミング同様にテキストプログラミングがおこなえるためにスムーズな移行が可能となった。そしてつぎは四則演算をおこないその結果を画面に表示するプログラムである。プログラミングの世界では「`tashizan = 5 + 3`」とは「`5 + 3`」の答えを「`tashizan`」という変数に代入しなさいという意味であり、数学のように「右辺と左辺が等しい」という意味ではないことに留意させる。プログラミングでは当然のことだが、初学者は困惑する箇所である。最後にまたブロックプログラミングで実行した反復処理（図表8）をテキストプログラミングでおこなう。同様の問題であるので理解が

深まると思われる。

レッスン1-2では反復処理とリストについて学習する。「リストの添え字は0から始まるのがコンピュータくんの特徴です」と説明したが、どれくらい学生が理解したのかは把握できていない。しかしいろいろな数字を入力し、試行錯誤しながら確認している様子がうかがえる。リストをもちいた反復処理を終えた後には、`random`モジュールが登場する。このモジュールにはランダムな数字を返したり、リストをランダムに操作したりする関数が含まれ、これがあれば運勢やおみくじなどが簡単にできますね、と説明するとわかったような顔をする学生が散見される。モジュールをインポートすることでPythonは短いプログラムでさまざまなことをすることが可能になる。

¹¹ 「チャット」と「ロボット」を組み合わせた造語で「ChatBOT」と表記されることもある。テキストや音声を通じて、自動的に会話するプログラムのこと。「チャット」は、テキストを使いネット上でやりとりをすること、「ボット」は、人がコンピュータを使っておこなっていた作業を、ロボットが自動的に実行するプログラムのことを指す。

大塚商会ウェブサイト「ボットチャット」『IT用語辞典』、<https://mypage.otsuka-shokai.co.jp/contents/business-oyakudachi/words/chatbot.html>、[2022年9月25日閲覧]。

レッスン1-3では、プログルでは登場しなかった「選択（条件分岐）処理」が登場する。「雨が降ったら傘をもち、そうでなければ傘をもちませんね」と説明したが、進度が速いためか学生たちの集中力が途切れがちである。今回は時間が足りないため、問題を端折りながら先き急ぐこととする。

3-3 WebAPI

API連携が容易にできることがPythonのよい点のひとつであり、それを簡単に実現できるのがプログル情報のよさである。API（Application Programming Interface）とは、ソフトウェア、プログラム、Webサービスといった異なるアプリケーション同士を繋ぐインターフェースのことを指す。そしてAPI連携とはAPIという手段を利用して、アプリケーションの機能とデータを第三者のアプリケーションに連携することを指す。

たとえば、「ポケモンGO」という2016年7月に発売されたゲームアプリがあるが、利用者の位置情報を活用することにより、現実世界そのものを舞台としてプレイするという当時画期的なゲームであった。利用者がいる場所や天気、時刻によって、ストーリーが変化し、画面中に雨が降る

仕組みについて不思議に思ったものだった。これが今思えばAPI連携であり、スマホ内の位置情報と天気予報サイトのサーバと連携させて、ゲーム内画面に雨を降らせていたのであった。これらはスマホやパソコンにおいてGoogleやYahoo等のポータルサイトの天気予報をみたときに雨が降っているのと同じ理屈である。

お天気APIへのパラメータ設定は、RequestsというサードパーティのHTTPライブラリをインポートすることによっておこなっている。このようなライブラリを使用することもPythonの利点のひとつであるが、インストールせずに体感できることもプログル情報のよさである。細かいプログラムの内容は理解できないまでも、おおよその意味を説明すれば、学生はなんとなく理解したような気になるものである。その後、場所コードを8410に書き換えさせると東京都の天気予報から長崎県南部の天気予報へと一瞬で変更された（図表11）。たったこれだけのことで学生は喜んでいる。今回はダイレクトにプログラムを打ち変えたが、実際にはinput関数を利用して入力させれば、誰にたいしても利用可能となると説明すると何人かの学生はうなずいている。

図表11 お天気API

ステージ2

お天気APIにパラメータを設定して、「東京都 東京」の天気を取得し、チャット画面に表示しよう。

ヒントを見る

確認する

The screenshot shows a programming interface with two main panels. On the left is a 'チャット画面' (Chat Screen) displaying a JSON object: `{ "prefecture": "長崎県", "area": "南部", "description": "くもり時々 晴れ", "rain": { "time": "00-03", "prob": 40 } }`. On the right is a code editor with two tabs: '実行時プログラム' (Running Program) containing `print()` and '送信時プログラム' (Send Program) containing Python code that imports 'requests', sets a URL to 'https://high.proguru.jp/api/weather', defines parameters for 'code: 8410', and uses `requests.get()` to fetch and print the response.

（出典）プログル情報をもちいて本論文筆者作成。

このプログラムではURLをみればわかるように、プログル内にあるデータをもちいて作成されている。実際の気象庁のデータを地域コード一覧

JSONデータによって確認してみると図表12のようになる。

図表12 気象庁の地域コード一覧（一部）

```
{
  "4120000": {"name": "武雄市", "enName": "Takeo City", "kana": "たけおのし", "parent": "410013"},
  "4120800": {"name": "嬉野市", "enName": "Ureshino City", "kana": "うれしのし", "parent": "410014"},
  "4121000": {"name": "りちょう", "parent": "410012"},
  "4134100": {"name": "基山町", "enName": "Kiyama Town", "kana": "きやまちょう", "parent": "410012"},
  "4138700": {"name": "玄海町", "enName": "Genkai Town", "kana": "げんかい", "parent": "410013"},
  "4142400": {"name": "江北町", "enName": "Kohoku Town", "kana": "あおまちちょう", "parent": "410013"},
  "4144100": {"name": "太良町", "enName": "Tara Town", "kana": "たらちょう", "parent": "410014"},
  "4220100": {"name": "長崎市", "enName": "Sasebo City (Uku Area)", "kana": "させぼし", "parent": "420022"},
  "4220200": {"name": "佐世保市 (宇久地域)", "enName": "Sasebo City (Uku Area)", "kana": "させぼし", "parent": "420022"},
  "4220400": {"name": "諫早市", "enName": "Isahaya City", "kana": "いさはやし", "parent": "420013"},
  "4220500": {"name": "諫早市", "enName": "Isahaya City", "kana": "いさはやし", "parent": "420013"},
  "4220800": {"name": "松浦市", "enName": "Matsura City", "kana": "まつうらし", "parent": "420021"},
  "4220901": {"name": "いし", "parent": "420021"},
  "4221000": {"name": "佐世保市 (宇久地域)", "enName": "Sasebo City (Uku Area)", "kana": "させぼし", "parent": "420022"},
  "4221300": {"name": "雲仙市", "enName": "Unzen City", "kana": "うんぜんし", "parent": "420011"},
  "4221400": {"name": "雲仙市", "enName": "Unzen City", "kana": "うんぜんし", "parent": "420011"},
  "4230800": {"name": "時津町", "enName": "Togitsu Town", "kana": "ときつちょう", "parent": "420022"},
  "4232300": {"name": "波佐見町", "enName": "Hasami Town", "kana": "はさみ", "parent": "420022"},
  "4241100": {"name": "新上五島町", "enName": "Shinkamizoto", "kana": "さざちょう", "parent": "420022"},
  "430015": {"name": "八代市", "enName": "Yatsushiro City", "kana": "やつしろうし", "parent": "430015"},
  "4320300": {"name": "水保市", "enName": "Minamata City", "kana": "みなまたし", "parent": "430032"},
  "4320600": {"name": "菊池市", "enName": "Kikuchi City", "kana": "きくちし", "parent": "430012"},
  "4321100": {"name": "菊池市", "enName": "Kikuchi City", "kana": "きくちし", "parent": "430012"},
  "4321300": {"name": "宇城市", "enName": "Uki City", "kana": "うきし", "parent": "430015"},
  "4321400": {"name": "阿蘇市", "enName": "Asu City", "kana": "あそし", "parent": "430012"},
  "4321600": {"name": "合志市", "enName": "Koshi City", "kana": "こうしし", "parent": "430012"},
  "4334600": {"name": "美郷市", "enName": "Mihoko City", "kana": "みこうし", "parent": "430012"},
  "4336700": {"name": "南関町", "enName": "Nankan Town", "kana": "なんかんまち", "parent": "430013"},
  "4336800": {"name": "南関町", "enName": "Nankan Town", "kana": "なんかんまち", "parent": "430013"},
  "4340300": {"name": "大津町", "enName": "Ozu Town", "kana": "おおつまち", "parent": "430012"},
  "4340400": {"name": "大津町", "enName": "Ozu Town", "kana": "おおつまち", "parent": "430012"},
  "4342400": {"name": "小国町", "enName": "Oguni Town", "kana": "おぐにまち", "parent": "430020"},
  "4342500": {"name": "小国町", "enName": "Oguni Town", "kana": "おぐにまち", "parent": "430020"},
  "4343200": {"name": "西原村", "enName": "Nishihara Village", "kana": "にしはらむら", "parent": "430020"}
}
```

(出典) 気象庁ウェブサイト「地域コード一覧 (JSONデータ)」国土交通省,
<https://www.jma.go.jp/bosai/common/const/area.json>, [2022年9月25日閲覧]。

図表12では、諫早市は420013、長崎県は420000
 となっており、JSONコードは必ず6桁であり、
 requestモジュールでインポートしてきたデータ

は気象庁のデータそのままでないことがみてとれ
 る。気象庁のデータが図表13であるが、これも同
 様である。

図表13 気象庁データ（一部）

```
{
  "publishingOffice": "長崎地方気象台",
  "reportDatetime": "2022-06-22T11:00:00+09:00",
  "timeSeries": [
    {
      "timeDefines": [
        "2022-06-22T11:00:00+09:00", "2022-06-23T00:00:00+09:00", "2022-06-24T00:00:00+09:00"
      ],
      "areas": [
        {
          "area": {
            "name": "南部",
            "code": "420010"
          },
          "weatherCodes": ["201", "210", "203"],
          "weathers": ["くもり", "曇り", "曇り"],
          "winds": ["南西の風", "南の風", "南の風"],
          "waves": ["1メートル", "1メートル", "2メートル", "2.5メートル", "3メートル"]
        },
        {
          "area": {
            "name": "北8部",
            "code": "420020"
          },
          "weatherCodes": ["201", "210", "203"],
          "weathers": ["くもり", "曇り", "曇り"],
          "winds": ["南西の風", "南の風", "南の風"],
          "waves": ["1メートル", "1メートル", "2.5メートル", "3メートル"]
        },
        {
          "area": {
            "name": "5島",
            "code": "420030"
          },
          "weatherCodes": ["100", "200", "203"],
          "weathers": ["晴れ", "くもり", "曇り"],
          "winds": ["南の風", "南の風", "南の風"],
          "waves": ["1メートル", "1メートル", "3メートル"]
        },
        {
          "area": {
            "name": "対馬",
            "code": "420040"
          },
          "weatherCodes": ["111", "210", "203"],
          "weathers": ["晴れ", "夕方からくもり", "夕方からくもり"],
          "winds": ["南の風", "南の風", "南の風"],
          "waves": ["1メートル", "1メートル", "3メートル"]
        }
      ],
      "timeDefines": [
        "2022-06-22T12:00:00+09:00", "2022-06-22T18:00:00+09:00", "2022-06-23T00:00:00+09:00", "2022-06-23T06:00:00+09:00", "2022-06-23T12:00:00+09:00", "2022-06-23T18:00:00+09:00"
      ],
      "areas": [
        {
          "area": {
            "name": "北部",
            "code": "420020"
          },
          "pops": [{"10", "20", "30", "30", "30", "30"}],
          "area": {
            "name": "対馬",
            "code": "420040"
          },
          "pops": [{"10", "20", "20", "30", "30", "30"}]
        },
        {
          "area": {
            "name": "5島",
            "code": "420030"
          },
          "pops": [{"10", "20", "20", "30", "30", "30"}]
        }
      ],
      "timeDefines": [
        "2022-06-22T09:00:00+09:00", "2022-06-22T10:00:00+09:00", "2022-06-23T00:00:00+09:00", "2022-06-23T09:00:00+09:00"
      ],
      "areas": [
        {
          "area": {
            "name": "長崎",
            "code": "84496"
          },
          "temps": [{"27", "27", "24", "28}],
          "area": {
            "name": "長崎",
            "code": "84072"
          },
          "temps": [{"29", "29", "23", "28"}]
        },
        {
          "area": {
            "name": "福江",
            "code": "84536"
          },
          "temps": [{"27", "27", "23", "27"}]
        }
      ],
      "publishingOffice": "長崎地方気象台",
      "reportDatetime": "2022-06-22T11:00:00+09:00",
      "timeSeries": [
        {
          "timeDefines": [
            "2022-06-23T00:00:00+09:00", "2022-06-24T00:00:00+09:00", "2022-06-25T00:00:00+09:00", "2022-06-26T00:00:00+09:00", "2022-06-27T00:00:00+09:00", "2022-06-28T00:00:00+09:00", "2022-06-29T00:00:00+09:00", "2022-06-30T00:00:00+09:00"
          ],
          "areas": [
            {
              "area": {
                "name": "長崎県",
                "code": "420000"
              },
              "weatherCodes": ["210", "203", "203", "202", "201", "101", "101"],
              "pops": [{"80", "70", "50", "30", "20", "20"}],
              "reliabilities": [{"B", "C", "A", "A", "A"}],
              "timeDefines": [
                "2022-06-23T00:00:00+09:00", "2022-06-24T00:00:00+09:00", "2022-06-25T00:00:00+09:00", "2022-06-26T00:00:00+09:00", "2022-06-27T00:00:00+09:00", "2022-06-28T00:00:00+09:00", "2022-06-29T00:00:00+09:00", "2022-06-30T00:00:00+09:00"
              ],
              "areas": [
                {
                  "area": {
                    "name": "長崎",
                    "code": "84496"
                  },
                  "tempsMin": [{"24", "24", "24", "24", "25", "24"},
                    "tempsMinUpper": [{"26", "26", "26", "27", "26"},
                    "tempsMax": [{"23", "23", "23", "23", "23", "27", "29", "30", "31"},
                    "tempsMaxUpper": [{"30", "29", "31", "32", "32", "32"},
                    "tempsMaxLower": [{"27", "26", "27", "28", "28", "29"}]
                  },
                  "tempsAverage": [
                    {
                      "area": {
                        "name": "長崎",
                        "code": "84496"
                      },
                      "min": "21.8",
                      "max": "27.1"
                    }
                  ],
                  "precipAverage": [
                    {
                      "area": {
                        "name": "長崎",
                        "code": "84496"
                      },
                      "min": "67.9",
                      "max": "127.8"
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

(出典) 気象庁ウェブサイト「長崎の天気」国土交通省,
https://www.jma.go.jp/bosai/forecast/data/overview_forecast/130000.json, [2022年9月25日閲覧]。

3-4 プログラム情報の実践を振り返って

Pythonプログラミングについての感想を読むと、難しかったと回答している学生が25人 (N=35)¹²いたが、そのうちプログラムが英語で書かれていることに言及している学生が12人であっ

た。プログラミングの世界では英語が標準語といってもよく、関数、モジュール、ライブラリなどの内容はすべて英語といってもよい状況である。プログラミングが好きならば別であるが、英語がひとつの障壁になるだろう。しかし難しかった

¹² 統計学の世界ではNとnは厳密に使い分けられており、母集団の大きさを表すときは大文字のN、その母集団から得た標本の大きさを表すときは小文字のnをもちいる。

たと回答しながら「よく読んでみると簡単だった」と記載している学生も多く、見た瞬間に思考回路を閉じたか閉じなかったかが、できたかできなかったかの分水嶺となっているようだ。やってみれば、最初は難しく感じてでもできるようになるという手ごたえを感じた学生が大半であった。そういう意味でもブロックプログラミングを導入したことは非常に有効であった。

お天気APIについては、たくさんのプログラムを入力すればいろいろな情報が入手できるとは思っていただろうが、単純なプログラムの入力、自分たちが住んでいる地域の天気の情報が入手できることに驚いた学生が多かった。またゲームやスマートフォンの天気予報などに連携していることを知り、身近なものだと感じた学生も多かった。そしてPythonライブラリの便利さの一端について回答している学生も散見された。しかし1時限で終わらせるには無理な分量をこなしたため、消化不良な面は否めない。体験利用としてではなくプログラミング教材として利用するのであれば、隔週で座学の時間を設けるなどの工夫が必要であろう。

全体を通しては、自分たちが利用しているアプリは誰かがプログラミングして作成したものなどの気付きを記載している学生がいるなど、身近なところとの接点を見つけられたことがよかった。そのうえプログラミングに興味を持ち、アプリ制作もやってみたいとの意見さえもあった。また日頃の授業で、「どのようにしたら相手が理解してくれるのか」「だいたい理解していることと、完璧に理解していることは大きく違う」ことをしばしば口にする。そのことをもちだし「日頃先生がやっていることと今日の授業がつながっている気がしました」と記載してくれた学生がいたのは嬉しい限りである。日頃の人間関係も、多文化理解も、そしてプログラミングも「相手にわかりやすく伝える」ことが共通項であることをあらためて感じた。

4 結論と課題

プログルにおいて数学の基礎教育とプログラミング基礎教育が同時にできることはすでに述べたが、説明を英語にすることで英語教育をすることも可能である。しっかりとした学力を身に付けさせるには簡単なところから始めて完璧にさせることが非常に重要である。難しいものを教材にして70点を目指す教育では行き詰まりが見えている。

そのあたりを考える必要がある。

まずプログルにおいて数学的思考について取り組ませた。算数とは計算することが主であるが数学、特に代数学では抽象的な観点から理論的に物事を考えることが大事である。算数の教材であっても、ものごとを論理的に考えたり相手にわかりやすく説明したりすることは、数学的思考やプログラミング的思考につながるものである。平均を求めることはできても平均を求める式を代数的に表現することは意外に難しい。この数学的思考ができればプログラミング的思考をすることは容易であり、言語学習につなげることは簡単である。またブロックプログラミングとはいえ、ただ悉皆式にあてはめて正答を目指すのではなく、まず仮説を立ててから検証をすることをしっかりと理解させて実践させれば、きちんとした学習と呼べると考える。

つぎにプログル情報であるが、Pythonを採用していることがわたしが選んだ理由である。データサイエンス教育においてPythonの学習ははずしたくなかったからである。Pythonはライブラリが豊富なことが利点のひとつであるが、それらを事前にインストールするとなると環境設定に時間がかかるし、学生にインストールさせようとするとセキュリティその他の問題が起きる。その点、プログル情報ではそれらのことにわずらわされずに使用できることが一番のメリットである。

プログルとプログル情報の両者を利用したことに関しては、ブロックプログラミングでなんとなくプログラミングの雰囲気があったところへ、似たような画面でテキストプログラミングへ軟着陸することが可能となり、非常に効果的であった。そして徐々にテキストの割合を増やしていくこともよいし、あらかじめ入力してあるプログラムを削除することも自由度の高さを感じられた。能力の高い学生や意識の高い学生にたいしては、すべてを入力させることで時間調整をすることが可能となるなど効果的に授業が進められた。

チャットボットやAPI連携も疑似体験できるなどPythonの有用性を簡単に体験するには効果的であった。合計2時限のプログラミング体験であったが、学生の評価はおおむねよかった。はじめてのプログラミング実習において嫌悪感をもたせず楽しい思いをもって終わらせたことは非常に嬉しい限りである。指導者側にプログラミングの素養さえあれば、十分にプログラミング的思考の涵養に効果的な教材である。今後さらなる実践を

進めていきたい。

【参考文献】

北川 舞、浅井健一「OCaml 初学者の学習調査」『プログラミングおよびプログラミング言語ワークショップ論文集』第23号, 1-16頁, 2021年。

武富志郎「知的障害特別支援学校小学部児童に対するロボット教材を用いたプログラミングの指導に関する研究」『日本教育工学会研究報告集』2022-1号, 70-77頁, 2022年5月。

松本晴香、浅井健一「Blockly をベースにしたOCaml ビジュアルプログラミングエディタ」『プログラミングおよびプログラミング言語ワークショップ論文集』第21号, 1-15頁, 2019年。

水谷好成、福井恵子、梅津直哉、米谷年法「特別支援教育におけるプログラミング教育を含む情報関連学習の段階的な実践」『宮城教育大学情報活用能力育成機構研究紀要』第1号, 77-88頁, 2021年3月。

文部科学省『小学校学習指導要領（平成29年告示）解説算数編』2019年7月。

文部科学省『小学校学習指導要領（平成20年告示）解説算数編』2008年6月。

文部科学省『小学校学習指導要領（平成10年告示）解説算数編』1998年12月。

【参考ウェブサイト】

大塚商会ウェブサイト「ビジュアルプログラミング」『IT用語辞典』、<https://www.otsuka-shokai.co.jp/words/visual-programming.html>, [2022年9月25日閲覧]。

大塚商会ウェブサイト「ボットチャット」『IT用語辞典』、<https://mypage.otsuka-shokai.co.jp/contents/busines>, [2022年9月25日閲覧]。

気象庁ウェブサイト「地域コード一覧（JSONデータ）」国土交通省、<https://www.jma.go.jp/bosai/common/const/area.json>, [2022年9月25日閲覧]。

気象庁ウェブサイト「長崎の天気」国土交通省、https://www.jma.go.jp/bosai/forecast/data/overview_forecast/130000.json, [2022年9月25日閲覧]。

プログルウェブサイト「プログルとは」、<https://proguru.jp/>, [2022年9月25日閲覧]。

プログル情報ウェブサイト「プログル情報とは」、<https://high.proguru.jp/>, [2022年9月25日

閲覧]。

Python Software Foundationウェブサイト「turtle：タートルグラフィックス」、<https://docs.python.org/ja/3/library/turtle.html>, [2022年9月25日閲覧]。