

ブロックプログラミングによる アルゴリズムの理解について

佐藤貴子

Understanding Algorithms with Block Programming

Takako SATO

This paper is an introduction to a trial of programming instruction in my class. Some newcomers to programming find the syntax of common programming languages difficult. Also, some students cannot imagine how information mathematics relates to programming. So I introduced block programming to my class and tried to see if it would contribute to the understanding of information mathematics and algorithms. As a result, we were able to obtain certain results.

Keywords: Block Programming, Scratch, programming education, ICT education

1. はじめに

プログラミング教育は近年、小学校・中学校や高等学校などで必修化され、教育用に特化した様々なプログラミング言語が提案されてきている^{[1][2][3]}。自分で自由にプログラミングを行えるようにするためには、プログラミング言語そのものに対する理解に加え、どのような手順で行えば実現できるのかを自力で組み立てられる力が必要となる。また、効率のよいプログラミングのためには、アルゴリズムに対する理解も欠かせない。

メディア情報学科にはプログラミングを学ぶ科目があり、JavaやJavaScript、C#などのプログラミング言語を使った学びが行われている。しかし、本学科で学ぶ学生は大学で初めてPCを使ったプログラミングを行うものも少なくない。そのため、中にはプログラミング言語の文法に悩まされ、プログラムの内容や手順の理解まで心を配れない学生もいた。

そこで、効果的なプログラミング学習法の検討のため、ヒューマノイドロボットPepper^[4]を用いて3・4年生のゼミナールの学生がプログラミング学習を行った取り組みについては既に文献^[5]で報告した。

本研究では対象を1年生にも広げ、昨年度～今年度に行ったブロックプログラミング言語を利用した2つの授業の取り組みについて報告する。

2. これまでの取り組み

筆者が担当する科目のうち、プログラミングそのものを基礎から学ぶ「プログラミング基礎」と、プログラミングに通じる情報科学の基礎を学ぶ「電子メディア論」を例にとり、これまでの取り組みと直面した問題点について述べる。

2.1 「プログラミング基礎」

筆者の担当科目の中で、プログラミングの基礎を学ぶ授業は「プログラミング基礎」である。この科目はJavaというプログラミング言語を用い、基礎的な文法に加えて「変数」「分岐」「ループ」など、Java以外の言語にも通じる基本的なプログラミングの考え方を学ぶ内容であり、履修する学生の多い授業である。2セメ（1年生の秋学期）以降に履修することができる。プログラミングを初めて学ぶ学生を対象とし、基本的な考え方についての教科書や資料を用いた説明ののち、まずはサンプルプログラムなどを参考に自分でプログラムを入力して作成する。

プログラムの入力にはTeraPadなどのテキストエディタを使ってきた（図1）。Javaのプログラムであることを認識してプログラムの一部を着色し、見やすくする機能はあるものの、入力補完機能は使わず、基本的には全て自力で入力する必要がある。

```

SampleCode.java - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
1 import java.io.*;
2
3
4 class SampleCode{
5
6     public static void main(String[] args) throws IOException{
7         {
8             BufferedReader br =
9             new BufferedReader(new InputStreamReader(System.in));
10
11             int sum = 0;
12             System.out.println("商品の値段を入力してください。");
13             System.out.println("全ての商品を入力し終わった方は0を入力してください。");
14
15             String str = br.readLine();
16             int price = Integer.parseInt(str);
17
18             for(;price != 0;){
19                 sum += price;
20                 System.out.println("次の商品の値段を入力してください。");
21                 str = br.readLine();
22                 price = Integer.parseInt(str);
23             }
24
25             System.out.println("合計金額は" + sum + "円です。");
26         }
27     }
28 }
28行: 1行 Java SJS CRLF 挿入

Windows PowerShell
PS C:\> cd "C:\>
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

> javac SampleCode.java
> java SampleCode
商品の値段を入力してください。
全ての商品を入力し終わった方は0を入力してください。
120
次の商品の値段を入力してください。
420
次の商品の値段を入力してください。
315
次の商品の値段を入力してください。
0
合計金額は1940円です。

```

図1 テキストエディタとCUIによるJavaの実習

そのため、そもそもPCでのキーボードを使った英文入力に不慣れなうちは入力やタイプミスの修正に大きく時間を割いてしまい、内容の理解の障害となってしまうことがあった。

また、本来であれば「変数」「配列」などの考え方や、「プログラムの流れをどう組み立てることで目的とする処理をプログラムに行わせることができるか」はプログラミング言語とは切り離して理解できるはずのものである。しかし、処理の組み立てとプログラミング言語の文法とを混ざり合せて学習することで、どこでつまづいているかを認識しづらいという問題があった。

2.2 「電子メディア論」

「電子メディア論」は情報科学の基礎を学ぶ科目であり、こちらも2セメ以降履修できる科目である。数値はもちろんのこと、文字、画像、音声、動画などの様々な情報をどのようにコンピュータの内部でデジタルデータとして表し、どのように処理を行うのかについて学ぶ座学の授業である。集合や論理演算などの情報数学の基本を学んだうえで、最後にプログラミングで処理を行う考え方やフ

ローチャートによるプログラムの流れの図解についても少し触れる内容で実施していた。

アナログデータをデジタルデータに変換する方法は、具体的なデータを例にとり示すことができるため、イメージがしやすい。これに対し、数学的な内容がどう情報処理と関係があるのかイメージがしにくいという問題があった。

そこで、授業終盤に少し触れる程度だったプログラミングで行う処理の説明を丁寧に行ったうえで、机上でも処理の動きを実感できるようにする必要を感じた。

2.3 改善策の検討

文献^[5]では、ヒューマノイドロボットPepperを動かすためのプログラムを、ブロックプログラミングアプリケーションChoregrapheで作成した。ブロックプログラミングのため処理の流れが直観的で把握しやすいこと、処理が進んでいく様子がリアルタイムにロボットの動きとして見えることから、プログラミングの経験がない学生でもプログラムを作成できた。

そこで、プログラミング言語の文法とは切り離して処理の流れを把握できるようにするため、ブロックプログラミングを活用することを考えた。処理の流れの把握だけであればフローチャートで図解するという手も考えられるが、ブロックプログラミングで示すことで実際に動かしてみて結果が想定通りになっているかを確認できることが利点である。

しかし、直観的とは言えブロックプログラミングでも最低限の文法を説明する必要はある。例えばユーザからのキーボード入力を受け付け、内容を受け取る方法などは、プログラミング言語ごとに決まった手続きを踏む必要があり、他のプログラミング言語から一対一対応で変換できるとは限らない。「プログラミング基礎」ではJavaでのプログラミング作成の流れや文法を初めて学ぶことを想定しており、同時に2つの新しい言語を学ぶことは混乱を招きかねない。そこで、今年度はまず「電子メディア論」において、これまでフローチャートのみで説明していたプログラムの処理の流れをブロックプログラミングでも示し、実際に動かしてもらうことを考えた。

使用するブロックプログラミング言語は、Webブラウザ上でプログラムを作成し動かすことができるScratch^[6]を選んだ。Scratchはマサチューセッツ工科大学で開発され、日本語のブロックを組み合わせることでプログラムの作成が行える(図2)。ブロック同士の組み合わせが可能か否かはブロックの形状で示されており、使いこなせば自分で新しいブロックを作成したり、マイコンボード^[7]やLEGO^[8]を動かすためのプログラムを作成したりすることができる。

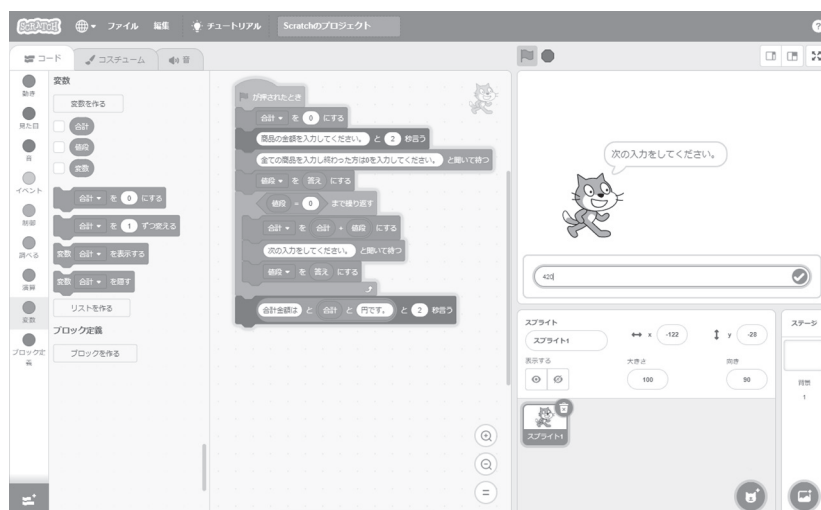


図2 Scratchの操作画面

3. 「電子メディア論」でのとりくみ

授業の前半では数値やアナログデータのデジタルデータへの変換、授業の中盤では集合、命題、論理演算といった情報数学の基礎について学んだ。

あらかじめプログラムの制御構造や変数・配列などのデータ構造、フローチャートによるプログラムの流れの記述方法について座学を行ったうえで、授業3回分を情報処理室での実習により実施した。

- 実施回：第10回～第12回
- 履修人数：49名
- 環境：Windows PC上でWebブラウザを通じて実習

Scratchではプログラムによるキャラクターアニメーションやゲーム作成などを行うことも可能だが、今回はあえて数値や文字の処理に焦点を当てて実習を行った。

3.1 第10回

Scratchの基本的な操作の説明を行った。ブロックのつなげ方や制御構造ごとのブロックの種類を説明したうえで、変数の作り方、ユーザからのキーボード入力の受け付け方についても学び、サンプルプログラム通りのプログラムを作成した(図3)。Scratchのブロックは日本語で表示されることもあり、英単語中心のプログラミング言語と比べると比較的早く使い方を理解できていたように思う。



図3 サンプルプログラム
(処理の内容は図1と同じ)

3.2 第11回

この回は、座学で学んだ内容とプログラミングとを結びつける回と位置付けた。

まずは、フローチャートで示したプログラムを実際にScratchのプログラムとして作成し、途中で変数がどのように変わっていくのかなどを実感してもらった。

次に、座学の講義中に取り上げた集合や命題の例を再度とりあげ、ある集合の人にのみ特定のメッセージを表示させるにはどのようにプログラムを書いたらいいかを考えてもらい、Scratchのプログラムを作成する演習を行った。例えば、生年月日を入力してもらい、年齢を計算して表示するプログラムを作成した。Scratchでは今日が何日であるか(西暦、月、日)を得ることができるが、この場合、今年の誕生日を

既に迎えているか否かにより表示すべき年齢が1歳異なる。そこでまず、今年まだ誕生日を迎えていない人はベン図でどのように表すことができるかを考えた(図4)。次にこれを論理演算式で表した。

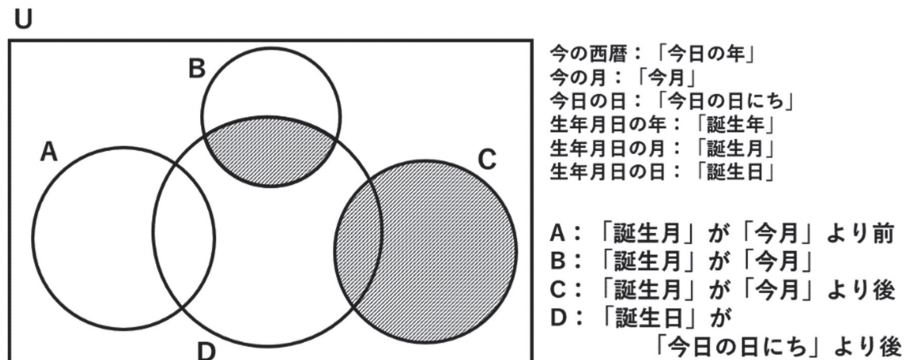


図4 誕生日についてのベン図

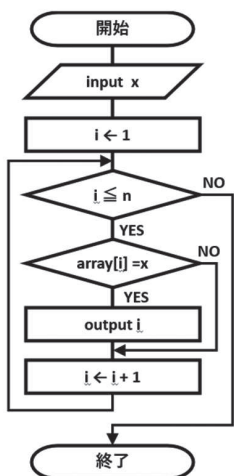
図4の表現に合わせると、今月が誕生月でまだ誕生日が来ていない人の集合は $B \cap D$ 、来月以降が誕生月の人の集合は C とあらわされることから、全体は $(B \cap D) \cup C$ と表すことができる。これを、変数と数式を利用して命題に変え、Scratchでプログラムを作成した(図5)。これにより、情報数学とプログラミングとのつながりを体感してもらった。



図5 Scratchによる年齢を表示させるプログラム

3.3 第12回

リストの作り方を学び、これを配列の代わりに使って探索や並び替えのプログラムを作成した。探索や並び替えは使われる機会が多く、手順によっては対象となる要素の数の多寡により効率に大きく影響を受けることから、様々なアルゴリズムが考案されているよく知られた問題である。机上でアルゴリズムを学ぶ場合には、1手順ごとにどう比較などが行われるかを手計算で試してみる必要があり、間違えると期待した結果が得られない場合もある。そこで、この回ではひとまずプログラムに手順を記述し、自動的に探索や並び替えが行われる様を実体験として感じてもらうこととした。なお、効率の非常によいアルゴリズムはプログラムが複雑になってしまうので、今年度はひとまず線形探索やバブルソートなど、非常に簡単に実装できるアルゴリズムについてフローチャートを示し、プログラミングを行った(図6)。



(a) フローチャート



(b) Scratchのプログラム

図6 線形探索のプログラミング

なお、第12回の授業を受け、座学である第13回の授業では様々な探索・並び替えのアルゴリズムを紹介し、目的が同じプログラムでもアルゴリズムにより効率が大きく異なることを学んだ。

3.4 成果

授業評価アンケートによると、「情報処理の歴史や計算法、しくみなどを、実際にScratchでプログラムを作成することや、小テストでの振り返りを通して、深い学びを得ることができたと思います。」「プログラミングやデータが動く仕組みを学んだ。パソコンでの実習もあり、知識を身につけやすかった。」と実習があったことで理解が深まったというコメントが2件あった（全27件中）。今回は特にプログラミング実習にフォーカスした感想などは集めていなかったため、次年度以降は授業内容をさらにブラッシュアップしながら、学生の実感も聞いていきたいと思う。

4. 「ゼミナールIG」でのとりくみ

文献^[5]の取り組みに引き続いて、今年度の3年生のゼミ生14名を3グループに分け、7回にわたってPepperでのプログラミングを行った。最初の2回はChoregrapheの使い方を学び、3回目以降は「オープンキャンパスで高校生向けに展示する」というテーマで自由に企画を考え、ディスプレイに表示する画像なども自分たちで用意してプログラミングを行ってもらった。

グループ1は相模女子大学をテーマとしたクイズを作成した。オープンキャンパスに来場する高校生は本学に関する情報を知りたいのではないかと、ということから問題を考え、タッチパネルで解答してもらう方式としていた。

グループ2は相模女子大学の食堂のおすすめメニューを提案するプログラムを作成した。ユーザに2～3問程度の質問をし、タッチパネルで回答をしていくと、1品のおすすめメニューを写真付きで紹介するものである。

グループ3は疑似恋愛シミュレーションゲームのようなプログラムを作成した。メディア情報学科に興味を持つ高校生はゲームなどのコンテンツに興味がある生徒が多いであろうという推測から、好みの男性のタイプとシチュエーションを1つずつ選ぶと、そのシチュエーションに合う情景を表示しながらPepperが甘いセリフをしゃべってくれるプログラムである（図7）。

テーマは共通であったが、各グループ個性的な企画を考案し、プログラムの得意・不得意に関係なく試行錯誤を繰り返しながら自由に作成していた。

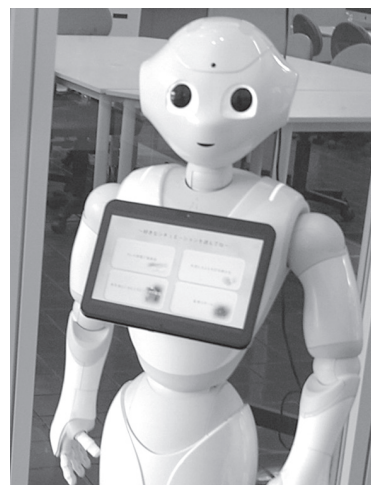


図7 グループ3の作品

5. まとめ

ScratchのプログラミングならびにPepperのプログラミングを通じ、大学生のプログラミング教育におけるブロックプログラミングの活用について取り組みを紹介した。いずれの例でも、ブロックプログラミングを利用することで処理の流れを把握しやすく、意図したとおりにプログラムが動かなかったときに試行錯誤しながら修正を試みるなど、ブロックプログラミングの利点を生かして演習を行うことができたと思う。今後は「プログラミング基礎」においても、処理の流れの理解などに活用する方法を模索したいと考えている。

謝辞

本研究を進めるにあたり、ヒューマノイドロボットPepperの利用について相模女子大学小学部川原田康文校長に多大な協力を頂いた。ここに記して謝意を表す。

参考文献

- [1] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, Yasmin Kafai: Scratch: Programming for All, Communication of the ACM, Vol. 52, No.11, pp60-67, 2009
- [2] 深谷和義, 宮地晶子: 小学生向けプログラミング授業のための「プログラミン」利用の検討, 日本教育工学会論文誌, Vol. 36, pp.9-12, 2012
- [3] Viscuit公式サイト: <https://www.viscuit.com/>
- [4] Softbank社Pepperページ: <https://www.softbank.jp/robot/>
- [5] 今井さやか, 佐藤貴子: 大学生に向けたロボットPepperを用いたプログラミング教育について, メディア情報研究第5号, pp.23-27, 2019年3月
- [6] Scratch公式サイト: <https://scratch.mit.edu/>
- [7] micro:bit日本語公式サイト: <https://microbit.org/ja/>
- [8] レゴエデュケーションプログラミングサポートサイト: <https://legoedu.jp/>