

中学校技術科のプログラミング教育における ビジュアル型プログラミングから テキスト型プログラミングへの移行について

伊藤 伸一・原田 信一

Research on Programming Education Using Visual- and Text- Type Programming Languages
in Junior High School Technical Courses

Shinichi ITOH and Shinichi HARADA

Accepted July 20, 2020

抄録：新学習指導要領により小学校におけるプログラミング教育が必修化され、2020年度から全面実施されている。多くの場合視覚的に学習することができるビジュアルプログラミングが利用されている。ビジュアルプログラミングは、テキスト型プログラミングの文字を入力する文法の学習を行わずに、命令文の実行結果を体験することができる点で有効な方法である。ビジュアルプログラミングの1つであるScratchは、そのシステム開発においてテキスト型プログラム言語が用いられている。Scratchの動作システムを理解するためには、テキスト型プログラム言語の知識が必要である。本論文では、ビジュアル型プログラミングとテキスト型プログラミングの連携を行うために、中学校技術・家庭科技術分野（以下、技術科）において、テキスト型プログラミング教育を行うための方針と問題点について考察する。

検索語：ビジュアル型プログラミング、テキスト型プログラミング、技術科、アルゴリズム

Abstract: Programming education was made compulsory in primary school in 2020, in which the objectives of the class is not to learn programming language but is, rather, to train logical thinking. There are four fields in junior high school technical courses, and in one of these areas, information, and technology, programming education is being done. However, such education mainly uses programming language for the control of an electric - motor car kit or a slit LED sensor. Thus, there are not many lessons learned about the algorithm or the programming language. To bring up the ICT human resources, more lessons on programming education should be done with use of a text- type programming language in junior high school or high school. This study proposes examples of programming education using both visual- and text-types of programing languages in junior high school technical courses.

1. はじめに

今後の社会の中で子どもたちが主体的に生きるためには、情報環境を道具として利用しながら論理的な思考ができる力を備えた人材を育てることが求められている。情報通信ネットワークの発達とともに、コンピュータは計算の道具から情報の収集や発信をする道具に変容し、今や知識は Web から収集する時代になってきている。この情報環境を使いこなすためには、単なる知識だけでは次に進展せず、十分な知識の基で問題を解決するための段階的な論理的組み立てができる知識が必要である。情報環境を利用した問題解決の論理的思考力を育成するために将来の社会を担うべき人材である小学生からプログラミング教育を始める必要がある。

新学習指導要領において小学生コンピュータプログラミング教育の導入が明記された。そこでは、プログラミング言語を学ぶのではなくプログラミング的思考力を養成することを目的としている。Scratch は、小学校のプログラミング教育で最も頻繁に用いられているソフトウェアの 1 つであり、ブロック型のプログラミング言語であるが、模型を遠隔操作したり、チャット機能を実装したりとネットワークや双方向性をも備えている。一方、技術科においてプログラミングは、4 つの内容のうちの「D 情報に関する技術」で必修化されている。2017 年告示の中学校学習指導要領技術・家庭編「D 情報と技術」には、次のように示されている。

- (2) 生活や社会における問題を、ネットワークを利用した双方向性のあるコンテンツのプログラミングによって解決する活動を通して、次の事項を身に付けるよう指導する。
 - ア 情報通信ネットワークの構成と、情報を利用するための基本的な仕組みを理解し、安全・適切なプログラムの制作、動作の確認及びデバッグ等ができること。
 - イ 問題を見いだして課題を設定し、使用するメディアを複合する方法とその効果的な利用方法等を構想して情報処理の手順を具体化するとともに、制作の過程や結果の評価、改善及び修正について考えること。
- (3) 生活や社会における問題を、計測・制御のプログラミングによって解決する活動を通して、次の事項を身に付けることができるよう指導する。
 - ア 計測・制御システムの仕組みを理解し、安全・適切なプログラムの制作、動作の確認及びデバッグ等ができること。
 - イ 問題を見いだして課題を設定し、入出力されるデータの流れを元に計測・制御システムを構想して情報処理の手順を具体化するとともに、制作の過程や結果の評価、改善及び修正について考えること。

従前の学習指導要領では、計測・制御のプログラミングは従来から行われていた内容であり、計測・制御のためのプログラミングの作成を通して、コンピュータを用いた計測・制御の基本的な仕組みを

知り、簡単なプログラミングの作成ができるようにするとともに、情報処理の手順を工夫する能力を育成することをねらいとしている。そして、今回の新学習指導要領ではプログラミングを主たる目的にすることが求められている。ここで問題となるのが、中学校技術科におけるプログラミング教育において、どのプログラミング言語またはソフトウェアを用いるのが適当であろうかという点である。

小学校で多用されていたビジュアル型プログラミングをそのまま用いて、高度な内容にすることが考えられる一つの方法である。プログラミングの問題の例として、数字が書かれた数枚のカードを大きい順または小さい順に並び変えるソートアルゴリズムがある。これを Scratch で実装すると、図 1 のようになるが、この場合は Scratch のブロックは少々複雑になるので、テキスト型プログラミングに比べて必ずしも容易であるとは言えない。¹⁾

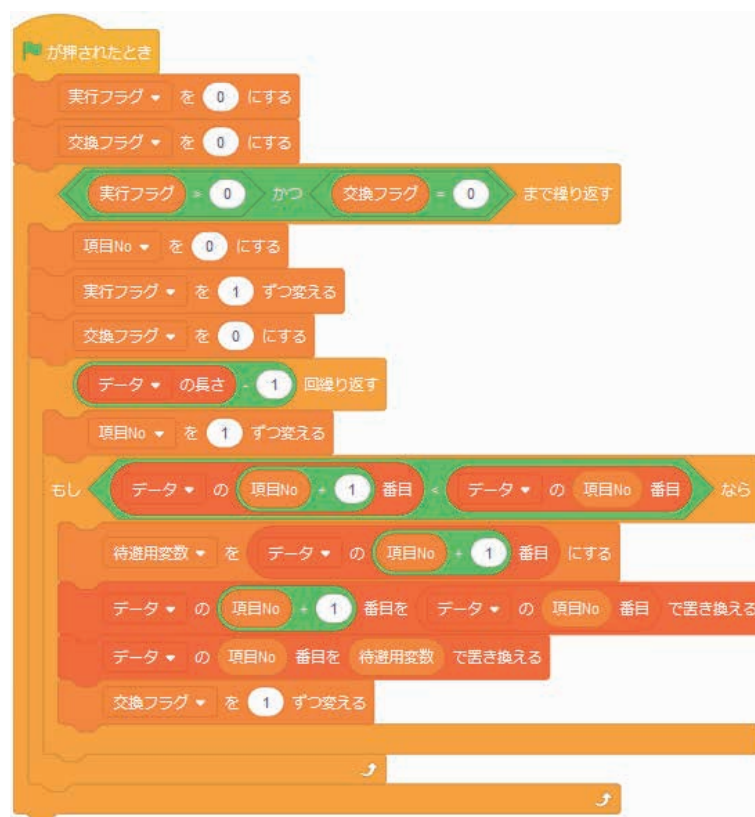


図 1: Scratch で実装したバブルソートアルゴリズム

ビジュアル型プログラミング言語を用いた場合でも、進んだアルゴリズムを実装するには複雑なブロックを作らねばならない。このようにアルゴリズムの複雑化に伴ってプログラムが複雑化する問題を回避するための 1 つの方法は、オブジェクト指向プログラミングの方法を用いることである。これは、あるプログラムを作成するときに、すべての命令を 1 つのファイルに収めるのではなくて、主たるプログラムと複数の目的別に分けられたオブジェクトプログラムのファイルに分けて、お互い連携させながら動作させる方法である。このとき、主たるプログラムからオブジェクトプログラムを呼び出し、目的の動作をさせることができる。先ほどの、ソートプログラムでは並べ替えの方法は別のプログラムにしておき、利用者が用いる主たるプログラムには、並べ替えを行いたい数の書かれたカードの数字だけを示し、その主たるプログラムが並べ替えのプログラムを呼び出す。利用者は複雑なプ

プログラム全体を作らなくても、目的の動作を行うことができる。

ビジュアル型プログラミングの利用については、アルゴリズムの複雑化に伴いその構造を解読するのは視覚的に却って困難さを増すことがある。プログラムの複雑化に伴って、学習者が処理の流れを解読するには、命令語が明確に定義されているテキスト型プログラミングを併用することがより理解を深めると考えられる。また、従来よりの開発のために用いられてきたプログラミング言語の多くはテキスト型であるので、内容によりテキスト型プログラミングで学ぶほうが効率がよく、進んだ学習者にはテキスト型プログラミングの経験をさせる機会を与えることが必要である。このような背景を考慮し、中学校技術科のプログラミング教育においては、テキスト型プログラミングを導入していくことが、学習者のプログラミングに対する理解度や知識を増すと考える。本論文では、ビジュアル型プログラミングを学んだ初学者がそれに引き続いて、テキスト型プログラミングを学習することを促進するための学習計画を考察する。

2. アルゴリズム提示型の中学校技術科のプログラミング教育の計画

技術科のカリキュラム（年間学習計画）では、2～3年次で「D 情報の技術」を実施することが多い。そして、技術科の3間の授業時間数は、1年生35時間、2年生35時間、3年生17.5時間で、合計87.5時間である。ここでは、プログラミングの問題意識を明確化するため、学習の順序を次のようにする。

1. 処理すべき問題とその解き方（アルゴリズム）の説明²⁾
2. Scratch を利用した、1. で示された問題のプログラミング^{1), 3)}
3. Java を利用した、1. で示された問題のプログラミング⁴⁾
4. 時間があれば、次の問題に進み1.に戻る。なければ次の問題に進まず、まとめを行う。

これは、小学校のプログラミング教育が、プログラミング的思考、に慣れることを目的としており、前述したソートなどの論理的な内容を扱わない場合が多いので、中学校段階ではプログラミングの本来の目的である問題を処理するための論理的方法の考えを学ぶことを目的とするために、問題とその解決方法（アルゴリズム）の授業を行った後に、プログラミングにとりかかることとする。プログラミングは、最初に Scratch を用いるが、その理由は小学校で学んだ場合が多いこととリテラルな文法ミスにより、学習が進まず生徒の意欲が薄れてしまうことを回避するためである。次に、テキスト型のプログラミングを Java を用いて学習する。ここでは、Java の文法は最小限にとどめ、プログラムの構造と動作の関連を説明し、テキスト型プログラミングの導入とする。以下、問題のアルゴリズムを説明する前に、整数型の変数の説明を行っておく必要がある。

(1) 1 から整数 N までの和 ($= 1 + 2 + \dots + N$) を求める問題

1) アルゴリズムの説明

1. 和を求める整数の上限値を入力する。(10 ならば、 $N = 10$)
2. 和の変数 s に初期値 0 を代入する。($s = 0$)
3. 足し算の回数を表す変数 i を 1 に設定する。($i = 1$)

4. 和の変数 s に回数の変数 i を加える。($s = s + i$)
5. 回数の変数 (i) が整数の上限 (N) より小さいか、または等しいならば、回数の変数に 1 を加え ($i = i + 1$) た後 4. に戻る。回数の変数 (i) が整数の上限 (N) を超えたら、終了する。このとき得られた s が答えである。

ここで、アルゴリズムの学習を定着させるために、筆記でこのアルゴリズムの計算を行う。1 から $N = 5$ までの和を求めるとする。

- $N = 5$
- $s = 0$
- $i = 1$
- $s = 0 + 1$
- $i = 1 + 1 = 2$
- $s = 1 + 2 = 3$
- $i = 2 + 1 = 3$
- $s = 3 + 3 = 6$
- $i = 3 + 1 = 4$
- $s = 6 + 4 = 10$
- $i = 4 + 1 = 5$
- $s = 10 + 5 = 15$
- $i = 5 + 1 = 6$
- 和の回数が上限を超えたので終了する。答えは 15 である。

2) Scratch によるプログラミング

次に、Scratch を用いた 1 から 10 までの和を求めるプログラムを示す (図 2)。¹⁾

ここでは、変数 N に 1 ずつ増加させることによって加える数を生成し、それを和を表わす変数 $Total$ に 10 回加える繰り返し文を用いている。



図 2: Scratch の 1 から 10 までの和を求めるプログラム

3) Java によるプログラミング

Listing 1:1 から 10 までの整数和を求める Java ソースコード

```
1 import java.util.Scanner;
2 public class SeisuuWa {
3     public static void main(String[] args){
4         int s,n;
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print(" 1 以上の整数を入力してください");
8         n = sc.nextInt();
9         s = 0;
10        for (int i = 1; i <= n; i=i+1) {
11            s = s + i;
12        }
13        System.out.println(" 1 から " + n + " までの和は " + s);
14    }
15 }
```

各行の命令文の説明を行う。

1. インポート文といい、以下で使う数字の入出力の機能(ジャバに用意されているジャバ.ユーティリティパッケージ。パッケージは、機能を持ったクラスの集まり。)の中のスキャナークラス(キーボードからの文字入力を扱う)を呼び出す。
2. クラス名の設定。クラスとはプログラム単位(=メソッド)の集まり(この例では、1つのメインメソッドのみ。)のこと。ここでは、“SeisuuWa”という名前にした。この場合、このクラスのファイル名は、“SeisuuWa.java”として保存しなければならない。
3. このクラスのプログラムの始まる行。メインメソッドの始まり。右端の{に注意。
4. 求めるべき整数和を表す変数s、足し算を行う回数を表す変数iの宣言。intは、2つの変数が整数型であると宣言している。
5. スキャナークラスをオブジェクト化するために、コンストラクタとして宣言する。オブジェクトとは、プログラム機能を部品のようにつなぎ合わせるための部品のこと。コンストラクタとは、クラス内でオブジェクトの型を宣言すること。ここでは、scというコンストラクタをスキャナークラスで定義している。
6. 見やすくするために1行空白行を入れている。
7. 文字や数字を出力するための命令。文字列は、“と”の間にはさんで書く。
8. キーボードから入力された数字を整数として読み取って、変数nに代入。
9. 整数和の初期値を0に設定。
10. 回数を表す変数iに1を代入。iがn以下であるか判断し、それが真ならば10行目の{と12行目の}の中の処理(つまり11行目)に進み、その後にiに1を加える。
11. sにiを加える。
12. for文の終わり。
13. 最後の結果を表示する。文字や数字をつなぐときは、+記号でつなぐ。
14. メインメソッドの終わり。
15. クラスの終わり。

すべての命令文を生徒に入力させるのは困難であるので、9行目、10行目、11行目を重点的に説

明し、入力作業を行うことが効率的である。

(2) 一列に並んだカードを数字が小さい順に並べ替える問題

1) アルゴリズムの説明

例として、左から順に 10, 8, 16, 4, 12 と書かれた 5 枚のカードを小さい順に並べ替える（昇順）バブルソート法を示す（図 3）。

2) Java によるプログラミング

Listing 2: カードの並べ替え（昇順）

```

1  import java.util.Scanner;
2  public class bubble_sort {
3      public static void main(String[] args) {
4          final int maisu = 5;
5
6          int[] cards = new int[maisu];
7          int koukan, nextsaisho;
8          Scanner sc = new Scanner(System.in);
9
10         for (int i = 0; i < maisu; i = i + 1) {
11             System.out.print(i + " 番目のカードの数");
12             cards[i] = sc.nextInt();
13         }
14
15         for (int i = 0; i < maisu - 1; i=i+1) {
16             nextsaisho = i;
17             for (int j = i + 1; j < maisu; j = j + 1){
18                 if ( cards[j] < cards[nextsaisho]) {
19                     nextsaisho = j;
20                 }
21             }
22             koukan = cards[i];
23             cards[i] = cards[nextsaisho];
24             cards[nextsaisho] = koukan;
25         }
26
27         for (int i = 0; i < maisu; i = i + 1) {
28             System.out.println(cards[i]);
29         }
30     }
31 }

```

1. スキャナークラスの呼び出し
2. 本クラスの名前の宣言
3. メインメソッドの開始
4. カードの枚数を 5 枚とする 5.
6. 5 個の要素を持つ整数型の配列 cards の宣言
7. 整数型の変数の宣言。koukan は、交換する前のカードが左から何番目にあるか、nextshousho は、交換した後のカードが交換する前に左から何番目にあるかを表わす。
8. キーボード入力した文字を変数として扱うためのスキャナーオブジェクトの宣言
- 9.
10. 配列の順序を表わす表わす数字 i を一番左（i=0）から一番右（i=4）まで動かす繰返しの開始

11. 順番にカードの数の数の入力促す画面への表示
12. キーボード入力された数字を配列 `cards[0]`, `cards[1]`, $\dots$, `cards[4]` に順に代入
13. 入力を表わす繰り返し文 `for` の終わり 14.
15. カードを左 (`cards[0]`) から順に見ていく
16. とりあえず見たカードを最小であると仮定する。見たカードの左からの位置を `nextsaisho` に代入する。
17. 見たカードの右隣のカードを右端まで順に見る
18. もし、右隣のカードが左のカードより小さいならば、
19. 右側のカードの位置を変数 `nextsaisho` に代入する。そうでなく、右側のカードが左側のカードよりも大きいならば、左のカードの位置が変数 `nextsaisho` に入ったままである。18 行と 19 行の操作を右端まで繰り返す。
20. 大小比較の終わり
21. 繰り返しの終わり
22. 見たカードを変数 `koukan` に代入
23. 見たカードに右側のより小さいを代入する
24. 見たカードを右の元より小さいカードがあった位置に置く
25. カードを左から見ていく繰り返し操作の終わり
- 26.
27. 並び変えたカードの表示。位置の変数 `i` は、0 から 4 まで。
28. カードの値が記録された配列変数 `cards[i]` の表示
29. 表示の繰り返しの終わり
30. メインメソッドの終わり
31. クラスの終わり



図 3: バブルソートアルゴリズム

3. 授業計画

技術科の4つの内容（A：材料と加工の技術，B：生物育成の技術，C：エネルギー変換の技術，D：情報の技術）から，中学校では，「D 情報の技術」は，生徒の発達段階を考慮して2～3年次で学習（22から23時間）することが多い。本稿で示した，アルゴリズムとプログラミングの授業時間の計画を表1に示す。ここで，アルゴリズムにはプログラミングで用いる変数，繰り返し，条件分岐などの文法事項を含む。丁寧に説明することと複雑になりすぎないことの両面を克服しなくてはならない。プログラミングは，早く完成した生徒に対して，和を1からではなく途中からにする場合，偶数だけにする場合など類似のプログラムで解ける問題を用意する。バブルソートに対しては，降順にする場合を追加する。

表1：「D 情報の技術」授業計画（2・3 学年）

テーマ	学習内容	授業時間数
整数和	アルゴリズム	3
	Scratch	3
	Java	4
バブルソート	アルゴリズム	3
	Scratch	4
	Java	5
計		22

4. まとめと今後の課題

本研究では，代表的なアルゴリズムをとりあげてビジュアル型，テキスト型両プログラミングにより，それぞれのスキルを身に着けることを目的とし，授業内容を提案した。予想される困難として，アルゴリズムの考えになじまない，興味がわからない，Scratch の特徴である動作やマルチメディアや模型の制御でないことから意欲が出ない，などが考えられる。これらは，それぞれアルゴリズムには変数の箱に値を入れるアニメーションをつくって説明する，Scratch のプログラミングをあらかじめ用意し最も重要な部分だけつくるようにする，Java も同様にソースプログラムを用意し，アルゴリズム上重要な命令文だけ入力する，などの支援が考えられる。できるだけ興味をもってもらえるように，これとは直接関係のない Scratch や Java の完成品を提示し，生徒は動作だけを確認する，などプログラミングに悩まされて終わることのないような工夫を用意すべきだろう。今後は，今回行わなかった現代プログラミングにおいて重要なオブジェクト化などのプログラミング技法について教育題材を提示する。

中学校での計測・制御のプログラミングは従来から行われていた内容であり，技術科の担当教員にとっては違和感がないと思われるが，小学校でプログラミング的思考の教育が行われてくると，中学校では興味関心を保持した学習が難しくなる。そのために，中学校ではプログラミングが生活の中でどのように役立っているかをより深く理解できるように教える必要がある。小学校でも移動ロボット

教材を使った学習が行われる実践も行われており、単なる延長としての移動ロボット学習は中学校では成立しなくなり、小学校での処理の流れの学習を踏まえて、中学校では更なる授業改善の工夫が求められることを注意する必要がある。小学校プログラミングから中学校技術科「D 情報の技術」への「連携」について、学習指導要領でも教科間の連携がより進められる児童生徒の経験を踏まえながら、取り組みたい課題や環境および発達段階に応じ、段階的に適切な選択をしていく必要がある。もちろん、小・中学校間でのスムーズな接続を考えると、使用する言語やツールについて、発展性を考慮しつつも、共通化が図られることが望ましいと考えられる。⁵⁾⁻⁷⁾

今後は、本授業計画を学校現場において授業実践と評価を行い、教材としての可能性を研究していく。

謝辞

本論文は、本学の教育研究改革・改善プロジェクト経費の補助を受けている。プロジェクト共同研究者の理学科高嶋隆一教授は、有意義な助言をいただいた。

参考文献

- 1) <https://ja.scratch-wiki.info/wiki/>
- 2) (株) アンク (2019 年)『アルゴリズムの絵本』翔泳社
- 3) 北村愛実 (2019 年)『Scratch で楽しむプログラミングの教科書』SB クリエイティブ
- 4) 飯塚泰樹, 大森康朝, 松本哲志, 木村功, 大西建輔 (2019 年)『Java で入門はじめてのプログラミング』森北出版
- 5) 山本朋弘, 薮田拳美 (2016 年)「小学校でのプログラミング学習における中学校技術科教員との共同指導による段階的な課題設定の一考察」『日本教育工学会論文誌』, 40 (3), pp.175-185
- 6) 文部科学省 (2017 年)『中学校学習指導要領解説 技術・家庭編』
- 7) 一般社団法人 日本産業技術教育学会 [編],『小・中・高等学校でのプログラミング教育実践問題解決を目的とした論理的思考力の育成』, 九州大学出版会