

プログラミング教育における論理的思考力の育成方法の検討：中学校技術科の情報教育を対象として

著者	那須 健治, 石上 靖芳
雑誌名	静岡大学教育学部研究報告. 教科教育学篇
巻	52
ページ	71-82
発行年	2020-12
出版者	静岡大学大学院教育学領域
URL	http://doi.org/10.14945/00027847

プログラミング教育における論理的思考力の育成方法の検討

—中学校技術科の情報教育を対象として—

Examination of how to develop logical thinking ability in programming education
-For information education in the technical department of junior high school-

那須健治¹, 石上靖芳²

Kenji NASU, Yasuyoshi ISHIGAMI

(令和 2 年 11 月 30 日受理)

ABSTRACT

The purpose of this study was to clarify the procedure for creating a program to develop logical thinking ability through programming education. In order to achieve this purpose, we developed and practiced a three-hour subject during the time of the technical department of junior high school, and verified the procedure for creating a program for each group. As a result, we pointed out the importance of procedures such as understanding of issues, objectives, decomposition, and time series.

1. 問題の所在

2008 年告示の中学校学習指導要領技術・家庭編から「プログラムによる計測・制御」という位置付けでプログラミングによる学習が必修化された。情報処理の手順を考える際に、自分の考えを整理するとともに、より良いアイデアが生み出せるよう、フローチャートなどを適切に用いることとし、情報処理の手順やフローチャート学習の必要性を示唆している。このことから、全ての中学生がプログラミングを体験的に学習すると共に、それらを活用して情報処理の手順を学習することが義務付けられた。

プログラミング教育の重要性として、大志田ら（2017）は、「今後の情報社会においてコンピュータは必要不可欠なものである事を踏まえ、コンピュータは人間の指示通りに処理を行うものである事を学び、ある問題や目的をクリアする、あるいは自身が作りたいものを想像していくために、試行錯誤（トライ&エラー）を繰り返し、手順を組み立てながら「問題解決能力」や「論理的思考力」を身につけていく事を、プログラミングを通して学んでいく」とその意義について述べている。

山本ら（2015, 2016）は、「プログラミング教育の効果は、論理的思考力が身につくことや、自分の考えや感情が発信できる表現力や説得力、新しい価値あるものを生み出す力を身につけることである」と述べている。

そのような中、2016 年 12 月中央教育審議会において「幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について」の答申を受け、平成 29

¹ 教育実践高度専攻

² 教職大学院系列

年告示の学習指導要領から小学校では、子供たちの生活や教科等の学習と関連付けて「プログラミング的思考」などを育む、プログラミング教育が初めて位置付けられた。

2018年3月には文部科学省が「小学校プログラミング教育の手引き(第二版)」を発行し、その中でプログラミング的思考とは、「自分が意図する一連の活動を実現するために、どのような動きの組み合わせが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組み合わせをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と説明し、プログラミング教育を通して、コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けることがねらいの一つであることが示された。

論理的思考力を身に付けるためのプログラムの作成について、プログラミング的思考の定義や例による作成の手順において記載されているが、具体的な実施方法については記載されていない。つまり、論理的思考力を身に付ける具体的な実施方法の詳細は記載されていない。

2017年告示の中学校学習指導要領技術・家庭科では、育成すべき資質・能力の思考力・判断力・表現力は、小学校でのプログラミング学習を踏まえて、プログラミング的思考を発揮して解決策を構想する力、流れ図等に表し、試行等を通じて解決策を具体化する力の育成が挙げられている。しかし、その資質・能力を育成するために必要な論理的思考力を身に付けるプログラムの作成手順について記載している教科書は三社中一社のみであり、簡単に紹介されている程度である。専門外の教諭にとってプログラムの指導展開が難しく、マニュアル通りのプログラム作成となり、プログラミング教育でねらう論理的思考力が身につくとはいえない。

大森ら(2017)は、プログラミング的思考に関する概念として、「抽象化(不要な部分を分離して複雑な事象をわかりやすくすること)」、「パターン認識(類似性とパターンを発見すること)」、「分解(理解と点検をしやすくするために、問題をより小さな小単位に分けていく手順)」、「一般化(合理的・効率的な手順体系にするため、規則化・定型化すること)」、「評価(一連の手順をふりかえり、必要に応じて修正・改善をしたり、次の問題解決に生かせるようにしたりすること)」を挙げている。

そこで、本研究では、筆者が在籍する所属校で行ったプログラム制作の授業において、プログラム作成ができた班(以降は正解班とする)とプログラムの一部に不具合が生じた班(以降は一部正解班とする)、プログラムに問題がある班(以降は不正解班とする)のプログラムの考察および比較することを通して、プログラミング教育を通して論理的思考力を育成するプログラムの作成手順を具体的に明らかにすることを目的とする。

2. 研究方法

2-1 研究対象

調査にあたっては、第一筆者の所属校であるA市立B中学校の2年生217名の7クラスを対象に、全3時間の単元として授業を設定し、授業実践を行った。分析対象データとして、第3時の「障害物で止まり、障害物がないときはライン上を走り続けるプログラムの制作」のパフォーマンス課題である。この実践においては、7クラス中1クラスの各班が制作したアイコン状のブロック図で作成したプログラムを収集し、正解班と一部正解班、不正解班が作成したプログラムを対象とした。なお、班の構成は男女混合の3名もしくは4名で構成され、1クラスあたり8班である。

2-2 題材の概要

題材名を「生活や社会を支える技術」とし、期間は令和元年12月2日から令和2年1月10日まで表1の題材構想で実践を行った。実践で用いた教材はレゴ®マインドストーム®EV3^{※1}を用い、筆者が図1の車体を制作し、第1時から第3時までプログラミング学習を行った。対象生徒のほとんどはプログラミング教育を受けていないため、一般的なプログラミングで使用する英文によるテキスト言語は使用せず、視覚的に理解しやすい一つ一つの処理がアイコン状のブロック図であるテキストではないビジュアル言語^{※2}を用いてプログラムの作成を行った。



図1 教材で使用した車体

表1：本実践の題材構想

時	本時の目標	学習内容
1 ・ 2	レゴを計測・制御する実習を通して、プログラムの制作の手順を理解する。(自動ブレーキとライントレースの実技指導)	①プログラムの流れを理解しよう。 ②障害物を感知したら止まるプログラムをつくろう。 ③ラインに沿って走るプログラムをつくろう。
3	基本的なプログラムを理解した生徒が、センサを2つ用いたプログラムの制作実習を通して、プログラミング的思考を働かせながらプログラムを考えることができる。	①ラインに沿って走り、障害物で止まる。障害物がなくなったらライン上を走り続けるプログラムをつくろう。 ②実用例の紹介。 ③実技を終えて、気付いたことや学んだことを付箋に書こう。

第1時と第2時は、プログラムの基本の処理方法である順次処理・くり返し・条件分岐くり返し・条件分岐型（処理の流れは3-1で説明する）を学習し、基本的な技能を身に付けた。本題材では、回数を指定する条件くり返し型は使用しないため、口頭で説明した。第3時にプログラミング的思考をはたらかせて論理的思考を育成するパフォーマンス課題を構想した。

2-3 分析方法

論理的思考力を育成するプログラムの作成手順を具体的に明らかにするために、生徒たちが作成したプログラムから各班がどのような思考をしていたか可視化する必要がある。レゴのプログラムはアイコン状のブロック図であるため、生徒たちが順次・くり返し・条件くり返し・条件分岐型のどの処理方法を用いてプログラミングをしていたのか、どのように思考していたのか読み取りづらい。そこで、それぞれの処理方法で表したフローチャートを用いて、生徒が作成したプログラムをフローチャートに変換し、そのフローチャートをコード化し、プログラムの合理性や問題点を探る。また、大森ら（2017）によるプログラミング的思考の概念である「抽象化」「パターン認識」「分解」「一般化」「評価」と「小学校プログラミング教育の手引き（第二版）」の作成手順の例から、共通している「分解」、「組み合わせ」、実物とプログラムを比較・改善する「評価」の3つの観点を主とした評価指標であるルーブリックを作成し、フローチャートやコード化、ロボットの動きからルーブリックによる判定を行った。

3. 結果と考察

3-1 プログラムで使用する手順のコード化

ここでは、プログラム作成に必要な順次処理・くり返し・条件分岐くり返し・条件分岐型の4つの型とプログラム作成で用いたレゴのビジュアル言語であるブロック図とそれらに対応したフローチャート、コード記号の設定、コード化についてまとめたものが表2である。

表2：プログラムの型に対応したブロック図、フローチャートおよびコード化

型名	(1)順次処理型	(2)くり返し型	(3)条件分岐くり返し型	(4)条件分岐型
レゴのブロック図				
フローチャート				
コード記号の設定	コード化にあたり、処理一つにつきAを用いる。	コード化にあたり、cをBとし、bはcからもどってくるため「もどるB」とする。	コード化にあたり、ひし形の部分はセンサによる判断を行うため、C(センサ名)とする。「いいえ」の場合、※に処理が流れるため※を「もどるC」とする。「はい」の場合は次の処理に流れる。	Cと同様にセンサによる判断を行うが(3)の型と区別するために、ひし形をD(センサ名)とする。「はい」「いいえ」の次の「処理1」「処理2」までをDの処理とする。
コード化	→A→A	もどるB→B	もどるC→A →C (センサ名)	D (センサ名)

※コード化にあたり、開始と終了はコード化しない。

表1のフローチャートからプログラムの処理の流れを説明する。(1) 順次処理型は、フローチャートにすると、矢印にそって一つ一つの処理が行われる。コード化にあたり、処理一つにつきAを用いる。よってコード化すると「→A→A→」となる。(2) くり返し型は、フローチャートにすると「a→b→c→b→c→b以下くり返し」となる。コード化にあたり、aは矢印、cをBとし、bはcからもどってくるため「もどるB」とする。よってコード化すると「→もどるB→B」となる。(3) 条件分岐くり返し型は、フローチャートにすると、ひし形つまりセンサによる判断を行う。センサが計測した数値とプログラムで設定した基準値の比較を行い、基

準値を満たさない場合は「いいえ」の矢印に流れ※までもどる。基準値を満たした場合は「はい」の矢印に流れ、次の処理へ進む。コード化にあたり、ひし形の部分はセンサによる判断を行うため、C（センサ名）とする。センサによる判断が「いいえ」の場合、※につまり処理がもどるため※を「もどるC」とする。「もどるC」のあとは矢印にそって前の処理と記された処理を行う。この処理は一つの処理のためAとなる。センサによる判断が「はい」の場合は次の処理に流れる。よってコード化すると「もどる C→A→C（センサ名）」となる。（4）条件分岐型は、フローチャートにすると、ひし形つまりセンサによる判断を行う。センサが計測した数値とプログラムで設定した基準値の比較を行い、基準値を満たさない場合は「いいえ」の矢印に流れ、次の処理を行う。基準値を満たした場合は「はい」の矢印に流れ、次の処理へ進む。（3）との違いはもどる流れがないことと「はい」と「いいえ」で別々の処理をすることである。コード化にあたり、（3）と同様にセンサによる判断を行うが、（3）との区別をするために、ここでのひし形は、D（センサ名）とする。「はい」「いいえ」の次の「処理1」「処理2」までがDの処理のため、Aとせず、Dとして扱う。よってコード化すると「D（センサ名）」となる。

3-1-1 コード化の例

コード化までの流れを、図2の（a）を用いて説明する。このプログラムは一つ一つの命令や指示の意味を成すブロック図を組み合わせたものであるため、図2中の（b）のように、一つ一つのブロックを切り離す。切り離した結果、①②③の3つのブロック図で構成され、①は開始、②はくり返し、③は条件分岐を意味する。

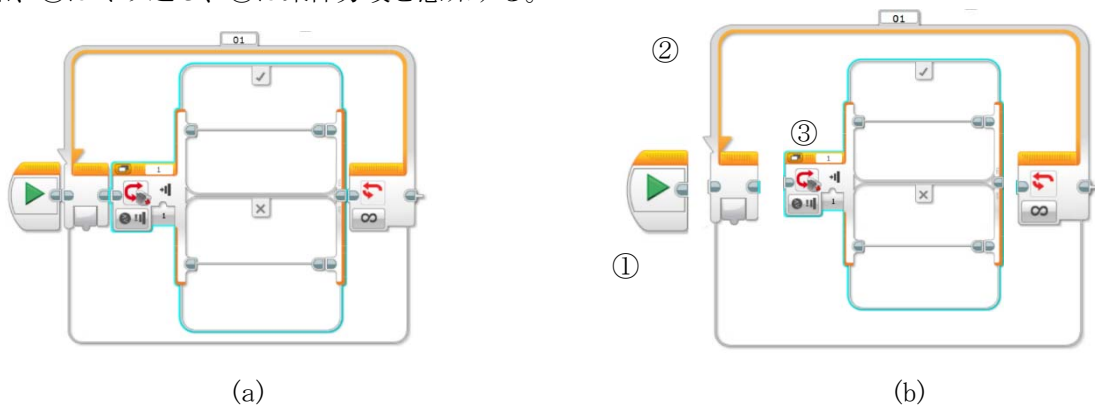


図2：プログラムを構成する一つ一つのブロックを切り離したもの

次に、切り離した一つ一つのブロックに対応するフローチャートに変換し、手順通りに組み合わせてフローチャートを完成させる。開始と終了はコード化しないため①は省く。図2の（b）から②の中に③があるため、図3の（c）（d）のように、表1から②と③をフローチャートにそれぞれ変換し、組み込むと図3の（e）となる。

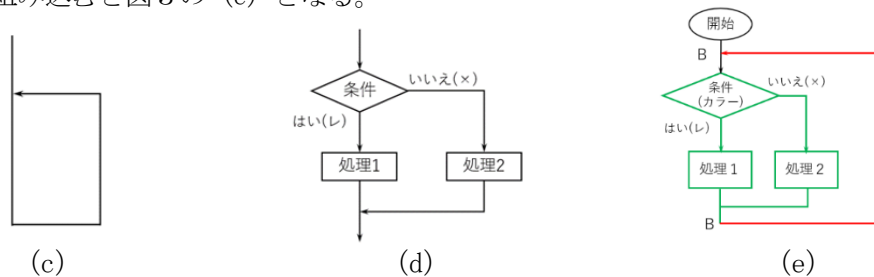


図3：フローチャートが完成するまでの流れ

最後に、図3の(e)をコード化する。図3の流れから(c)の「もどるB→B」の中に、(d)のDが組み込まれるため、コード化すると、「もどるB→D→B」となる。(e)のひし形の(カラー)から最終的に「もどるB→D(カラー)→B」となる。なお、カラーはカラーセンサの略であり、本実践では、カラーセンサの他に超音波センサ(以下超音波とする)も用いた。

3-1-2 ループリックの設定および対象班の判定結果

正解班、一部正解班、不正解班のプログラム作成から作成手順を明確化するための手立てを探るために、プログラムを作成する上で必要な3つの概念である「分解」、「組み合わせ」、実物との比較を行う「評価」の3つの観点をループリックとして設定した(表3)。

表3：プログラム作成に必要な3つの概念「分解」、「組み合わせ」、「評価」のループリック

基準	分解	組み合わせ	評価(実物と比較)
○	必要な動きを一つ一つ細かく分解することができる。	一度組み合わせしてから類似性や関係性を見出し、処理手順を再構成することができる。	実物の動きとプログラムを照らし合わせながらエラーの原因を見つけることができる。
△	必要な動きを一つ一つ分解することができる。	一つ一つの動きを処理の手順ごとに組み合わせることができる。	実物の動きとプログラムを照らそうとするが、エラーの原因を見つけることができない。
×	必要な動きを一つ一つ分解することができない。	処理手順を組み合わせることができず、とりあえず組み合わせている。	実物の動きとプログラムを照らし合わせることができない。原因を見つけることもできない。

次に、対象班が作成したプログラムをフローチャートおよびコード化し、ループリックをもとに評価を行った。正解班は3つの概念がすべてできていた。一部正解班は、「分解」「組み合わせ」は一部でき、「評価」はできていなかった。不正解班は3つの概念すべてができていなかった。各班のコード化をみると、不正解のプログラムはコード化が長くなったり、流れがおかしかったりする。一部正解班のコード化は正解班と似ているが、プログラムの流れは全く異なる。判定した結果を表4にまとめた。

表4：対象班のコード化および判定結果

対象	コード化	分解	組み合わせ	評価
正解班	もどるB→もどるC→D(カラー) →C(超音波)→B	○	○	○
一部正解班	もどるB→D(カラー)→もどるC→A →C(超音波)→B	△	△	×
不正解班①	A→もどるC→A→C(超音波)→	×	×	×
不正解班②	もどるB→D(超音波)→B→D(カラー)→	×	×	×
不正解班③	もどるC→A→C(カラー)→もどるB →もどるb→b→D(超音波)→B	×	×	×
備考：模範解答のコード化 もどるB→D(超音波)→D(カラー)→B				

3-2 対象班のプログラムおよびフローチャート、コード化と考察

3-2-1 パフォーマンス課題の模範解答のプログラム

パフォーマンス課題の模範解答のプログラムとフローチャートおよびコード化を図4に示す。

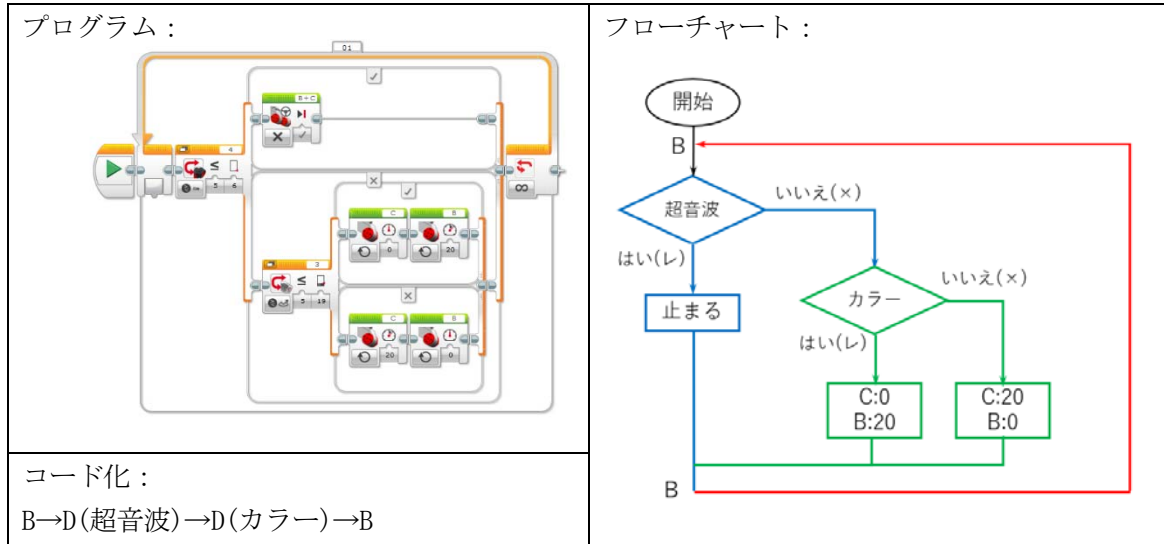


図4：パフォーマンス課題の模範解答のプログラムとフローチャートおよびコード化

パフォーマンス課題を解決する必要な動きとして、①超音波と障害物との距離が6 cm以下になるときは止まる。②6 cmより大きいときはカラーでライン上を走り続ける。そして、③その動きを繰り返していく。の3つであり、これらを組み合わせたのがプログラムになる。フローチャートから、障害物の認識は超音波と障害物との距離で設定し、障害物を認識したときは、「はい(レ)」の状態です「止まる」。障害物を認識しないときは「いいえ(×)」の状態とし、カラーでラインの黒とその周りの白を計測し、その結果で左右のBとCのモータ(数値はパワーを意味し、0は動かないためジグザグ走行をする。)を制御させて走り続けるようにする。なお、この走り続けるプログラムは第2時で行ったプログラムをそのまま引用した。センサが判断した後の処理は一瞬しか行わないため、常にはたらくように「はい(レ)」 「いいえ(×)」の後にもどす「くり返し」が必要になる。

補足として、センサを2つを使用する場合、センサ同士の組み合わせにも違いが生じる。筆者が考えたこのプログラムは2つのセンサを「いいえ(×)」でつなげたフローチャートである。超音波かカラーのどちらかが「はい(レ)」になればくり返されるプログラムであり、この型をOR型と呼ぶ。また、センサ同士を「はい(レ)」でつなげたフローチャートもある。超音波とカラーの両方が「はい(レ)」になればくり返されるプログラムであり、この型をAND型と呼ぶ。

3-3-2 正解班が作成したプログラム

正解班が作成したプログラムとフローチャートおよびコード化を図5に示す。

分解では、授業者が、車に必要な動きの一つである「障害物がないときは走る」を学級全体に投げかけたところ、そのヒントをもとに考え始め、「走るときは超音波が反応しないとき」という必要な動きを考えることができた。

組み合わせでは、動きから超音波の条件分岐繰り返し型を用いて組み合わせることができた。

評価では、エラーの原因を見つけようとレゴの動きとプログラムを照らし合わせながら考えていた。授業最後には、学級全体にプログラムの流れを説明することができた。

このプログラムはセンサ同士を「はい(レ)」でつなげたプログラムのためAND型である。

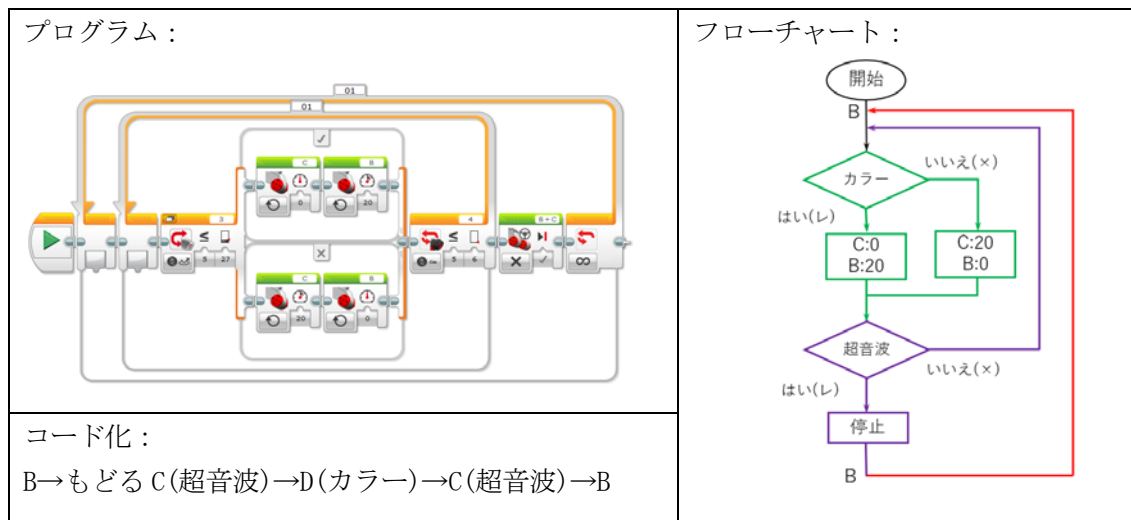


図5：正解班が作成したプログラムとフローチャートおよびコード化

3—3—3 一部正解班が作成したプログラム

一部正解班が作成したプログラムとフローチャートおよびコード化を図6に示す。

分解では、今まで学習してきた障害物で止まるプログラム、線に沿って走り続けるプログラムの2つを融合し、「走って障害物で止まる」プログラムを考えた。フローチャートだとカラーで走り続けるプログラムの後に、超音波による障害物で止まるプログラムである。コンピュータが理解できるところまでは分解できていない。

組み合わせでは、走って障害物で止まるという手順は理解し、走る・止まるのそれぞれの動きを表すプログラムの組み合わせはできている。しかし、具体的な動きを見出すことができないため、そこから先の組み合わせはできていない。

評価では、この班は「走りながら」をイメージしているが、実際のプログラムでは前進して止まるか前進するかの流れになっていることに気付いていない。プログラムの流れを理解していないため、実行後のエラーを見つけることもできなかった。

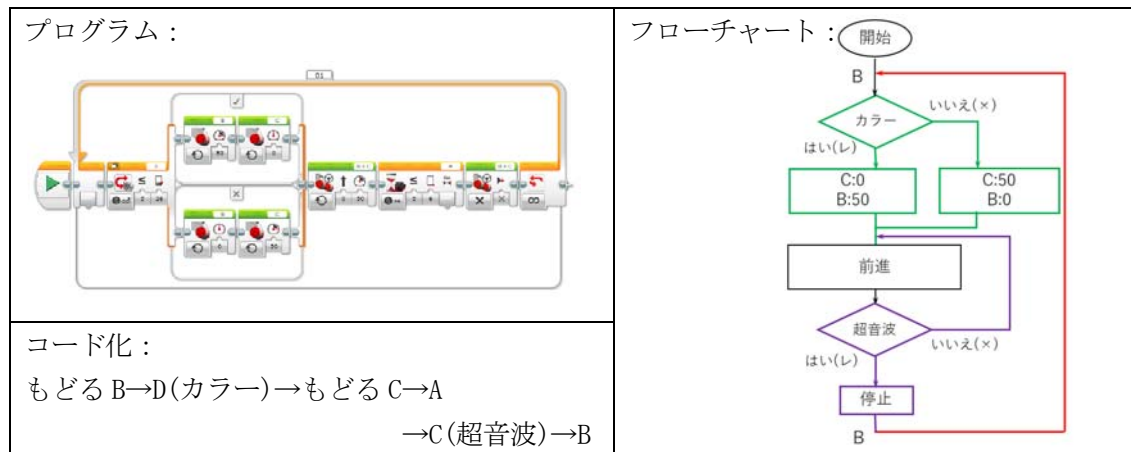


図6：一部正解班が作成したプログラムとフローチャートおよびコード化

3-3-4 不正解班③が作成したプログラム

不正解班③が作成したプログラムとフローチャートおよびコード化を図7に示す。

分解では、フローチャートにすると、使用するセンサは理解しているが、必要な動きを理解していないため、使用しているセンサとプログラムがおかしいことがわかる。

組み合わせでは、とりあえず組み合わせている状態である。またフローチャート真ん中（ひし形の超音波の前）にある「もどる b→b」があり、組み合わせがおかしいことになっている。

評価では、プログラムの流れを理解していないため、原因を見つけることができていない。

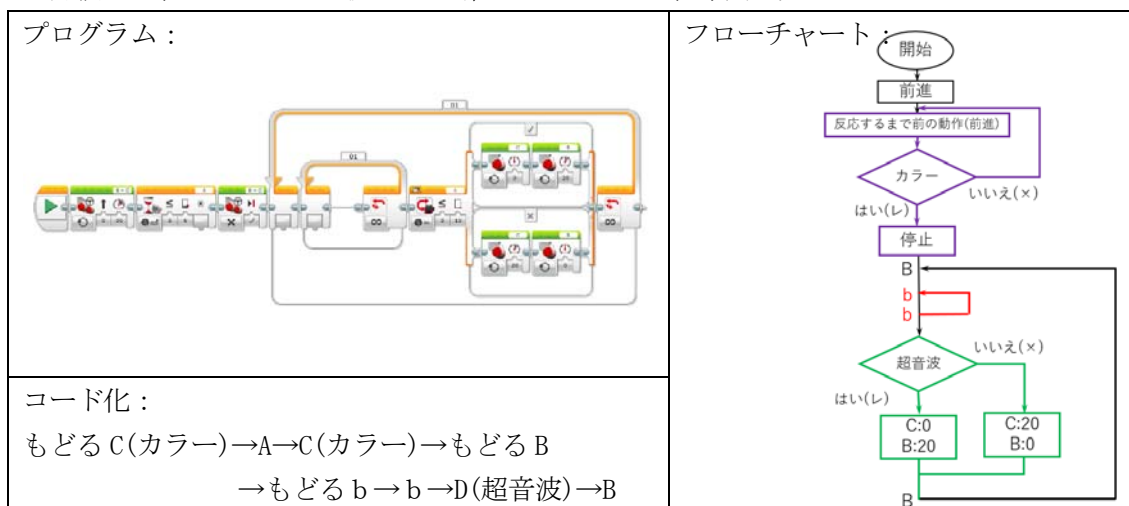


図7：不正解班③が作成したプログラムとフローチャートおよびコード化

4 総合考察

ルーブリックで使用した3つの概念から正解班と不正解班の特徴をまとめる。正解班の特徴として、

- (1) 課題解決に必要な一つ一つの動きを明確化(分解)することができた。
- (2) 明確化できたことにより手順を理解し、組み合わせができた。
- (3) 一つ一つのブロックの動きやプログラムの流れを理解していたので、エラーが起きても原因を見出し、組み合わせを変えることができた。

不正解班の特徴として、

- (1) 使用するセンサは理解していたが、課題解決に必要な一つ一つの動きを明確化できていなかった。
- (2) 動きが明確化できなかったため組み合わせができなかった。
- (3) 一つ一つのブロックの意味理解も曖昧な部分もあり、つなげたときのプログラムの流れを理解していない。そのため、実際の動きとプログラムを照らし合わせるができず、どこに問題点があるのか分からず改善することができなかった。

ここから作成についての手順を考えたとき、まずはプログラムのエラーを見つけるために、プログラムの流れを理解することが大切である。そのために、基本となるプログラムの型の学習時に、処理の流れが明確なフローチャートを用い、一つ一つのプログラム命令言語やブロック図と照らし合わせながら説明することで、プログラムの処理の流れを理解することができるだろう。

プログラム制作に入る前は、制作手順を理解させておく必要もある。その手順として具体的に説明する。はじめに①「課題および目的の理解」である。これは、プログラムを作成する目的を理解していないと、実現したい動きの構想につながらないためである。②「分解」では、①を達成するために必要な動きを一つずつ書き出す。コンピュータが書き出した動きを理解することができるように、必要なセンサやその条件、その結果による動きになっているかの点で確認する。③「時系列」では、②で書き出した動きを処理の手順ごとに整理し、時系列でならべる。④「型の選択」では、時系列で整理した流れから、プログラムに必要な型を選択する。センサを2つ使用するとき、処理の流れによってAND型かOR型になる。⑤「組み合わせ」では、③で整理した一つ一つの処理に合うフローチャートの図記号を用い、④で選択した型になるよう組み合わせ、フローチャートにする。⑥「紙面評価」では、紙面上で一度流れを確認し、無駄やエラーがないかを確認する。⑦「プログラム作成と実行」では、⑥をもとにプログラムを作成し、実行させる。⑧「デバッグ (改善)」では、⑦の実行中にエラーが起きたときは、実行結果とプログラムを照らし合わせながら問題点を発見し、修正を行う。処理の流れや②の分解における必要な動きが適切かどうか確認をする。以上の①から⑧の手順について、まとめたものが表4である。

この手順を踏まえてプログラム制作を行うことで、プログラミングによる論理的思考力を養うことにつながると考えられる。

表4：論理的思考力を育成するプログラムの作成手順

手順	活動内容	詳細
①	課題および目的の理解	プログラムをつくる目的を理解する。
②	分解	どんな動きが必要か、必要な動きを一つずつ書き出す。コンピュータが書き出した動きを理解することができるように、必要なセンサやその条件、その結果による動きになっているかの点で確認する。 例：障害物に反応→反応という言葉は理解できない→「障害物とセンサとの距離が〇cm以下のとき」と変換する。
③	時系列	②で分解した動きを処理の手順ごとに整理し、時系列にする。
④	型の選択	③で整理した流れからパターン(順次、条件くり返し、条件分岐、くり返し)を選択する。センサを使うときのパターンは表1のCの条件分岐繰返し型かDの条件分岐型のどちらかを使用する。センサを2つ使用するとき、C・DかC・C、またはD・Dのパターンとなる。その際の型は、OR型かAND型になる。
⑤	組み合わせ	④のパターンをフローチャートで図式化し、組み合わせる。
⑥	紙面評価	紙面上で流れを確認し、無駄やエラーがないかを見つける。
⑦	プログラム作成と実行	プログラムの制作と実行
⑧	デバッグ (改善)	実行中にエラーが起きたときは、問題点を修正する。場合によっては①に戻る。

5 今後の展望

本実践では、第1時と第2時においてプログラム作成をしながら、レゴの動きと照らし合わせて処理の流れについて説明したが、生徒にとって初めてのプログラミング学習かつ操作であったため、多くの生徒は処理の流れを理解することが難しかった。そのため、第3時のパフォーマンス課題において、プログラムの作成に苦戦するとともに、作成できたとしてもプログラムのエラーを見つけることができず改善できなかつたと考える。プログラムの作成についての手立てを第3時の後半で紹介することで、学習の活性化が導かれた班もあった。中には、正解した班もあった。

基本となるプログラム作成時に、本実践から得られた作成手順をもとに制作し、プログラムの処理の流れもおさえる学習を題材の中に位置づけることが重要である。次の実践では、プログラムの作成手順やフローチャートの学習を行い、計測・制御の技術における問題発見・解決学習でのプログラムの制作を通して、論理的思考を育成することができたか検証していく。

【謝辞】本研究を推進するにあたり、全面的な支援をいただいた富士市立富士南中学校の職員の皆様に、この場を借りて厚く御礼を申し上げます。

【附記】本研究は、令和元年～3年度科学研究補助金基盤研究(C)（課題番号 19K02728 研究代表者 石上靖芳）を受けての研究成果の一部です。

[註]

- ※1 ブロック玩具のレゴが実社会につながるロボットプログラミング教材として開発したものの。
- ※2 プログラムをテキストで記述するのではなく、視覚的なオブジェクトでプログラミングする言語である。

参考文献

- ・文部科学省(2008)：「中学校学習指導要領(平成20年告示)解説 技術・家庭編」, 開隆堂
- ・文部科学省(2016)：「幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について(答申)」
- ・文部科学省(2017)：「中学校学習指導要領(平成29年告示)解説 技術・家庭編」, 開隆堂
- ・文部科学省(2017)：「小学校学習指導要領(平成29年告示)解説」, 開隆堂
- ・文部科学省(2018)：「小学校プログラミング教育の手引き(第二版)」
- ・大志田 憲・昇高茂樹(2017)：「小中学校で実施されるプログラミング教育の意義と課題—本学における小中学生および教員向け講習会を事例として—」, 岩手県立大学宮古短期大学部研究紀要, 第28巻
- ・山本利一・本郷健・本村猛能・斎藤美・水井克洋・石田祐輝(2015)：「初等中等教育におけるプログラミング教育の必要性—プログラミング教育の教育的意義—」, 日本教育情報学会第31回論文集
- ・山本利一・本郷健・本村猛能・永井克昇(2016)：「初等中等教育におけるプログラミング教育の教育的意義の考察」教育情報研究, 第32

卷

第2号

- ・大森康正・磯部征尊・上野朝大・尾崎裕介・山崎貞登(2017)
:「小学校プログラミング教育の発達段階に沿った学習到達目標とカリキュラム・マネジメント」, 上越教育大学研究紀要, 第37巻第1号
- ・中学校技術・家庭技術分野教科書, 開隆堂 P226