

CAS ガイドラインに基づく中学校・高校段階での プログラミング教材の提案

岩崎 誠^{†1} 掛下 哲郎^{†1}

概要: 2020年から小学校にて新学習指導要領が実施され、順次、中学や高校でも実施される。新学習指導要領では、情報活用能力を言語能力と同様に「学習の基盤となる資質・能力」と位置づけ、中等教育ではプログラミングの基礎等について学習する。しかし、学習指導要領や情報教育の参照基準等を見ても、具体的な教育内容や教育レベルが明確とは言えない。我々はイングランドの Computing At School (CAS) が策定した Computing Progression Pathways (CAS ガイドライン) に基づいて、教育用の mBot ロボットを対象とする小学校プログラミング教材を開発している。本論文ではこの教材を拡張し、アルゴリズム、プログラミングと開発、ハードウェアと処理、データとデータ表現の4つの学習領域について、中学校及び高校段階に相当するレベル 5~8 に対応するプログラミング教材を提案する。本教材は、学習領域とレベルの組み合わせを段階的に変更することで、様々な能力レベルの生徒に対応できる。

キーワード: プログラミング教育, 初中等教育, 手順的思考力, CAS ガイドライン, mBot ロボット

A Proposal of Secondary Level Programming Education Material Based on CAS Computing Progression Pathways

MAKOTO IWASAKI^{†1} TETSURO KAKESHITA^{†1}

Abstract: New curriculum will start at Japanese secondary and high schools in 2021 and 2022 respectively. The curriculum contains education of algorithmic thinking through programming education. However, education content and expected level are still unclear, although Japanese ministry of education published a set of policies regarding curriculum formulation etc. We have developed a toolkit to learn algorithmic thinking using mBot robot. In this paper, we propose programming education material by extending the toolkit for secondary and high school programming education. The material is based on the Computing Progression Pathways developed by Computing at School (CAS Guidelines). We design education material about algorithm, programming & development, hardware & processing having strong relationship with programming education. The material corresponds to the education level 5 to 8 of the CAS guidelines. Education level of our material can be easily controlled by selecting appropriate combination of domain and levels of the materials.

Keywords: Programming Education, Secondary Education, Algorithmic Thinking, CAS Guidelines, mBot Robot

1. はじめに

2020年から小学校にて新学習指導要領が実施され、情報活用能力を言語能力と同様に「学習の基盤となる資質・能力」と位置づけている[1]。中学校においては、技術・家庭科(技術分野)にてプログラミング教育が行われ、現行で必修の「プログラムによる計測・制御」に加え「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」について学ぶことが決まっている。また、高校では「関数を用いたプログラムの構造の整理、性能の改善」や「モデル化とシミュレーション」などについて学ぶ。文部科学省によると、プログラミング教育で育成すべき資質・能力は、知識・技能、思考力・判断力・表現力等、学びに向かう力・人間性等の3つの柱で構成されている。

しかし、中学校ではプログラミング教育を普通教科として単独では実施しておらず、技術・家庭科の技術分野での学習を行うことに留まっている。また「学習指導要領解説」

[2]を参照しても、上記の教科におけるプログラミング教育の内容について例示されているが、教育内容、教育レベル、学習時間等についての具体的な記述は見受けられず、学習教材や、授業計画を作成する際の根拠にするのは難しい。

これに対してイングランドでは、小中学生が学ぶべき情報教育の内容を学習項目ごとに具体的に示した Computing Progression Pathways(以下、CAS ガイドラインと略記する)が Computing At School (CAS) によって策定され、小中学校に広く普及している[3]。

我々は、CAS ガイドラインに基づく小学校プログラミング教材を開発している[4]。CAS ガイドラインで定義されている6つの学習領域から、プログラミング教育との関連が深いアルゴリズム・プログラミングと開発、ハードウェアと制御の3領域を選んで、小学校で標準とされるレベル1~5に対応するプログラミング教材で、プログラミングに関連する学習項目と、指示内容を組み合わせることで、レベルに応じた教育ができる。これにより国際的同等性を確

^{†1} 佐賀大学
Saga University

保できる。また、教育用ロボット mBot (図 1) を活用し、仕様、アルゴリズムおよび scratch プログラムを関連づけて mBot プログラミングを学習できる。これにより、教員や児童・生徒に具体的なプログラム例を示すことができ、algorithmic thinking の指導や系統的なプログラミング教育の実施が期待される。また、アルゴリズム中に含まれる基本操作を、操作対象のデバイス(モーター、各種センサー等)毎に分類することで、様々なアプリケーションを企画する際のヒントとしても活用できる。



図 1 : mBot ロボット

本論文では、開発した小学校プログラミング教材を拡張し、中学校、高校段階に対応した教材を提案する。学習領域をアルゴリズム、プログラミングと開発、ハードウェアと処理の3つに新たにデータとデータ表現を追加し、学習レベルを小学校段階のレベル1から5に新たに中学校、高校段階のレベル5から8を追加した教材である。

以下、2節では、中学校及び高校段階の情報教育について述べる。3節では mBot を用いたプログラミング教育用ツールキットについて紹介する。4節では教材を作成する際に参考にした CAS ガイドラインについて述べる。5節では CAS ガイドラインに基づいた教材の作成方針を提案する。6節では作成した教材の具体例を示す。

2. 中学校及び高校段階の情報教育

2.1 中学校の情報教育に関する学習指導要領

新学習指導要領では、技術・家庭科(技術分野)にてプログラミング教育が行われる[2,5]。技術分野の学習項目は、材料と加工、生物育成、エネルギー変換、情報の4項目である。情報の学習内容は、生活や社会を支える技術、技術による問題の解決、社会の発展と技術の3つに分類される。プログラミングに関連が深い学習内容である技術による問題の解決では、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と、計測・制御のプログラミングについて学ぶ。

2.1.1 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る問題の解決

- 情報通信ネットワークの構成、安全に情報を利用するための仕組み、安全・適切な制作、動作の確認、デバッグ等
 - コンピュータの相互接続方法、ネットワークの構成、サーバやルータ等の働き、パケット通信や

Web での情報表現、記録や管理など

- 課題を解決するためにプログラミング言語を用いて情報処理の手順や構造(順次、分岐、反復等)を入力し、プログラムの編集、保存、動作の確認、デバック等ができること
 - 問題の発見と課題の設定、メディアを複合する方法などの構想と情報処理の手順の具体化、制作の過程や結果の評価、改善及び修正
 - アクティビティ図のような統一モデリング言語等を用いた課題の解決策の構想
 - 解決策の全体構成やアルゴリズムの図表現
- #### 2.1.2 計測・制御のプログラミングによる問題の解決
- 計測・制御システムの仕組み、安全・適切な制作、動作の確認、デバッグ等
 - センサ、コンピュータ、アクチュエータ等の計測・制御システムの要素
 - 計測・制御システムの各要素において異なる電気信号(デジタル・アナログ)を変換し、情報を伝達するためにインターフェースが必要であること
 - 計測・制御システムで一連の情報がプログラムによって処理されていること
 - プログラミング言語を用いて情報処理の手順や構造を入力し、プログラムの編集等ができること
 - 問題の発見と課題の設定、計測・制御システムの構想と情報処理の手順の具体化、制作の過程や結果の評価、改善及び修正
 - アクティビティ図のような統一モデリング言語等を用いた課題の解決策の構想
 - 解決策の全体構成やアルゴリズムの図表現

2.2 高等学校の情報教育に関する学習指導要領

高等学校では、必修科目「情報」を新設し、すべての生徒がプログラミング、ネットワークやデータベースについて学習する[6]。選択科目「情報」では、情報システムや多様なデータの活用、コンテンツの創造について学ぶ。

2.2.1 情報

情報 の学習内容は、情報社会の問題解決、コミュニケーションと情報デザイン、コンピュータとプログラミング、情報通信ネットワークとデータの活用の4つに分類される。

コンピュータとプログラミングでは、コンピュータの仕組みや特徴、情報の内部表現と計算の限界、アルゴリズムの表現方法、プログラミングを用いたコンピュータやネットワークの活用、モデル化、シミュレーションを用いたモデルの評価・改善、情報の特徴とコンピュータの能力との関係についての考察、アルゴリズムの作成・表現及びプログラミングに渡る過程の評価・改善、モデル化やシミュレーションを行い問題の解決策の作成について学ぶ。

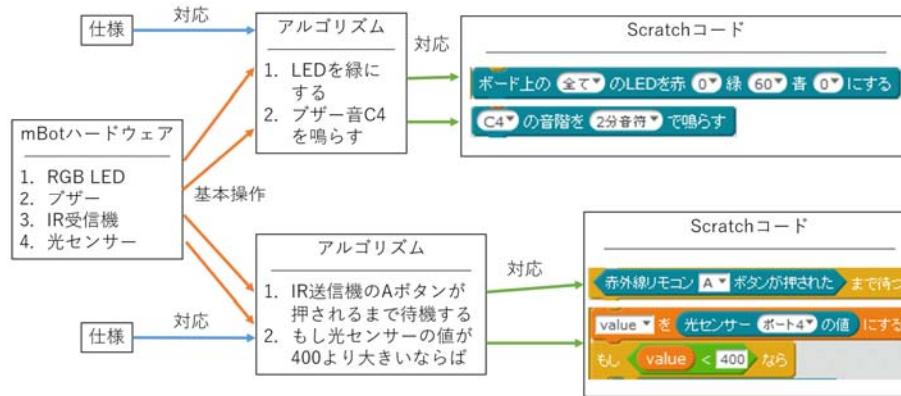


図2：ツールキットの構成

2.2.2 情報

情報の学習内容は、情報社会の進展と情報技術、コミュニケーションとコンテンツ、情報とデータサイエンス、情報システムとプログラミング、情報と情報技術を活用した問題発見・解決の探求の5つに分類される。プログラミングと関連が深い情報とデータサイエンス、情報システムとプログラミングの2つについて内容を以下に示す。

情報とデータサイエンスでは、多様かつ大量のデータの存在やデータ活用の有用性、データサイエンスの役割の理解、データの収集・整理・整形、モデル化とその評価・改善について学ぶ。

情報システムとプログラミングでは、情報システムのデータフローや処理の仕組み、情報セキュリティ、システム設計の表現、ソフトウェア開発プロセス、プロジェクトマネジメント、システム構成要素のプログラミング、社会におけるシステムの役割・影響、システムの機能単位での開発・統合について学ぶ。

2.3 情報教育の参照基準

日本学術会議が策定中の「情報教育の参照基準」[7]では、情報学分野の参照基準や学士課程教育の構築に向けて（答申）、学習指導要領などを基に初等中等段階、高等段階について情報学のどのような学習内容が望まれるかを検討している。また、情報学の知識やジェネリックスキルを情報及びコンピュータの原理、情報の整理と創造などの11分野に分類している。プログラミングに関連が深い「プログラムの活用と構築」分野では、プログラムの理解と活用、プログラムの設計・実装、計画的なプログラミング作業、プログラムの修正、改良の4つの内容が含まれており、それぞれ4つのレベルに分けられている。しかし、CASガイドラインと比較すると内容の具体性不足は否めない。

3. 教育用ロボット mBot を用いたプログラミング教育用ツールキット

3.1 ハードウェア

mBot のメインボードである mCore は、初期状態では、ブザー、IR 送受信機、ボタン、モーターなどのデバイスで構

成される。追加で超音波センサ、ライントレースセンサ、Bluetooth などを利用できる。これにより、CAS ガイドラインの「ハードウェア・処理」をカバーできる。

3.2 ツールキットの概要

我々が提案しているツールキット[4]は仕様、mBot ハードウェア、アルゴリズム、Scratch コードの4つで構成される（図2）。図に示すように、仕様とアルゴリズム、ハードウェアとアルゴリズム、アルゴリズムと Scratch コードは相互に対応付けられている。

これらの要素によってツールキットを構成することにより、以下の利点が得られる。

- 仕様とアルゴリズムを対応付けることにより、目的とそれを実現するための（自然言語による）手段の記述を自然に対応付けることができる。
- アルゴリズムの各ステップと Scratch コードを対応付けることにより、自然言語による手段の記述と、プログラミング言語による記述の対応を具体的に表現できる。
- アルゴリズム中の各ステップは、基本操作に対応しているが、それらの基本操作を操作対象の mBot のデバイス毎にまとめることで、デバイスが提供している機能を自然に表現できる。その例を表1に示す。

表1：モーターと基本操作の対応の例

デバイス	基本操作
モーター	指定した速度で直進する
	指定した速度で後退する
	停止する
	右方向に指定した角度に曲がる
	右方向に指定した速度で曲がる
	右車輪を指定した速度で回転する

本ツールキットは、以下に示す22個のプログラムを含む。これらのプログラムは、授業を行う際の例としても、演習を行う時のテンプレートとしても活用することを意図している。

- Scratch の制御構造等を含む基本コード（1個）
- 図形の描画（4個）

- 入力データを用いたロボットの制御 (6 個)
- 6 本足パーツを活用した歩行ロボット (3 個)
- 各種センサを活用する基本コード (8 個)

我々は mBot に対応して本ツールキットを開発しているが、その他のハードウェアに対応してツールキットを修正することも可能である。

4. CAS ガイドライン

イングランドでは、教員向けの教材の提供や研修を実施している Computing At School (CAS) によって小中学生が学ぶべき情報教育の内容を学習項目ごとに段階的に示した Computing Progression Pathways (以後、CAS ガイドラインと表記する) が策定され、中等学校に普及している[3]。CAS ガイドラインは、学習時の目標設定や、次のステップで習得すべき内容の確認のほか、学習者の習熟度を評価するために活用されている。

CAS ガイドラインの学習項目は、アルゴリズム、プログラミング・開発、データ・データ表現、ハードウェア・処理、コミュニケーション・ネットワーク、情報技術の 6 領域に分類される。また、それぞれの具体的な到達目標は AB (Abstraction): 抽象化, DE (Decomposition): 分割, AL (Algorithmic Thinking): アルゴリズム的思考, EV (Evaluation): 評価, GE (Generalization): 一般化の 5 つの概念に分類される。到達レベルは 8 段階で定義されており、そのうち、小学校ではレベル 1~5、中学校ではレベル 5~7、高校ではレベル 8 が標準とされている。

プログラミングと関連が深いアルゴリズム、プログラミングと開発、ハードウェアと処理、データとデータ表現の 4 領域について、中学校・高校段階に相当するレベル 5~8 の到達目標のうち、本論文で参照する部分を以下に示す。

4.1 アルゴリズム

4.1.1 レベル 5

- 構造化した表記を利用した解決法を説明できる。(AL)(AB)
- 状況の類似点や相違点を特定し、それらを問題解決に利用できる (パターン認識)。(GE)

4.1.2 レベル 6

- いくつかの問題が同じ特徴を共有し、両方の問題を同じアルゴリズムを利用して解決できることを知っている。(AL)(GE)

4.2 プログラミングと開発

4.2.1 レベル 6

- 入れ子の選択文を利用できる。(AL)
- 変数を利用する関数(AL)(AB)
- 順次処理と関数の違いを理解し、適切に利用できる。(AL)(AB)
- 否定演算子を理解し、利用できる。(AL)
- 1 次元配列を利用できる(AB)

- 構文エラーを検出して修正できる。(AL)

4.2.2 レベル 7

- 変数のスコープを理解している。例えば、ローカル変数は関数外からアクセスできない。(AB)(AL)
- 引数の受け渡しを理解し、利用できる。(AB)(GE)(DE)
- 前判定ループ (while) と後判定ループ (repeat-until) の違いを理解し、利用できる。(AL)
- モジュールや関数の単位でエラーの検出や修正ができる。(AB)(DE)(GE)

4.3 ハードウェアと処理

4.3.1 レベル 8

- 低水準のプログラミング言語を使った小規模のプログラムを作ったことがある。(AB)(AL)(DE)(GE)
- ムーアの法則を理解し説明できる。(GE)
- コンピュータによるマルチタスキングを理解し、説明できる。(AB)(AL)(DE)

4.4 データとデータ表現

4.4.1 レベル 6

- ビットパターンを用いた簡単な命令を実行できる (例: 二進数の加算)。(AB)(AL)

5. プログラミング教材の作成方針

本論文では、CAS ガイドラインに基づき、プログラミング教育と関連が深いアルゴリズム、プログラミングと開発、ハードウェア・処理、データとデータ処理の 4 領域について、中学校・高校段階に相当するレベル 5~8 における mBot ツールキットを用いたプログラミング教材を提案する。各レベルの難易度は条件分岐や反復、サブルーチンなどのプログラミングに関する学習項目と、与えたプログラムを修正させる、アルゴリズムからプログラムを実装させるなどの指示内容を変化させることで区別している。表 2 に各学習領域およびレベルにおける学習項目の違いを、表 3 に生徒に対する指示内容の違いを示す。

5.1 学習項目と指示内容に基づく教材の作成方針

学習項目と指示内容は領域 = {アルゴリズム、プログラミングと開発、ハードウェアと処理、データとデータ表現} およびレベル = {1, 2, ..., 8} によって区分されているため、教材の作成方針も、領域およびレベルごとに策定する。また、系統的かつ段階的な教育を行うため、領域 A におけるレベル x の教材は、同じ領域 A におけるレベル x-1 の学習内容を踏まえて構成する。

以上を踏まえた教材の作成方針を図 3 に示す。

学習領域 A におけるレベル x の到達目標は、図の一番上に示すように、学習項目と指示内容に関する 2 つの表で示された領域 A・レベル x の教材が理解または実践できることである。このとき、領域 A・レベル x の教材には、学習項目の表で示した領域 A におけるレベル 1 から x までの項目が含まれる。また、指示内容についても、領域 A にお

表2：学習領域，レベルごとの学習項目

レベル	アルゴリズム	プログラミングと開発	ハードウェアと処理	データとデータ表現
1	単純なアルゴリズム	単純なプログラム	プログラムの必要性	
2	ループ，条件分岐 入れ子ではないループ，条件分岐	算術演算子，条件分岐，ループ	入出力機器	<ul style="list-style-type: none"> 文字，数字 表
3	入力に応じた出力	変数	センサによるデータ収集	<ul style="list-style-type: none"> フラットファイル フィルタ検索
4	引数，変数を使わないサブルーチン		OS，ネットワーク	信頼性
5		<ul style="list-style-type: none"> 高水準言語を用いたプログラミング ループや分岐での論理演算子や式の活用 	<ul style="list-style-type: none"> コンピュータアーキテクチャ フェッチ実行サイクル 	<ul style="list-style-type: none"> 実数 Boolean 型 2進数でのデータ表現等 クエリ言語
6	再帰アルゴリズム	<ul style="list-style-type: none"> 入れ子の選択文 変数を利用する関数 否定演算子 一次元配列 	<ul style="list-style-type: none"> フォン・ノイマンアーキテクチャ アドレス指定可能メモリの関数，命令 	2進数の加算
7		<ul style="list-style-type: none"> 変数のスコープ 引数の受け渡し 前判定，後判定ループの違い 	プロセッサの命令セット	2進数と電気回路
8		<ul style="list-style-type: none"> for 文と while 文の違い 二次元配列 	<ul style="list-style-type: none"> 低水準言語プログラミング ムーアの法則 マルチタスキング 	<ul style="list-style-type: none"> 2進数の減算，16進数との変換 データ圧縮 RDB

表3：学習領域，レベルごとの指示内容

レベル	アルゴリズム	プログラミングと開発	ハードウェアと処理	データとデータ表現
1	具体的な指示を元に修正させる			
2	<ul style="list-style-type: none"> エラー修正 実行結果の予測 フローチャートでの表現 仕様の変更を指示し，与えたアルゴリズム，プログラムを修正できる 			データ形式のプログラムへの活用
3	不完全なアルゴリズム，プログラムを復元させる			単一フィルタ検索
4	与えた仕様からアルゴリズムを設計させる	与えたアルゴリズムに基づいてプログラムを実装させる		<ul style="list-style-type: none"> 複雑な検索（Boolean 型，関係演算子） データや情報の解析，評価
5	<ul style="list-style-type: none"> アルゴリズム作成のガイドラインの理解 類似点や相違点の活用 			表データの問い合わせ
6	アルゴリズムの効率	構文エラーの検出，修正		
7	アルゴリズム作成ガイドラインに基づいてアルゴリズムを設計できる	モジュール単位でのエラー検出，修正		
8	再帰アルゴリズムの設計	再帰アルゴリズムの実装		

るレベル 1 から x までの内容が含まれる。ただし，学習項目と指示内容のいずれかについて，レベル x の項目または指示内容を少なくとも 1 つは含む。

学習領域 A におけるレベル x に到達するための前提は，学習領域 A におけるレベル x-1 に到達していることである。しかし，レベル x の学習項目と指示内容はともに新しい内容なので，それらを組み合わせた教材を用いて新しい内容を教育すると，学生にとって負担になる可能性がある。そこで，1 つ下のレベルの教材に新たな学習項目のみを追加した補助教材 1 および指示内容のみを追加した補助教材 2 を用いて新しい学習項目や指示内容の理解を深めた後に，

それらを組み合わせた教材に取り組ませることで，段階的な教育を行う方針を採用する。なお，補助教材 1 と補助教材 2 のいずれを選択するかは，教育上の効果を考慮して担当教員が決定してよい。

学習項目と指示内容に関する 2 つの表には一部空欄がある。この場合は，補助教材 1 または補助教材 2 のいずれが存在しないため，選択肢は 1 つに限定される。

学習項目の表にも指示内容の表にも，同一の領域・レベルに対して複数の項目が含まれている場合がある。この場合には，すべての項目を少なくとも 1 つの教材が含むように項目を組み合わせる。

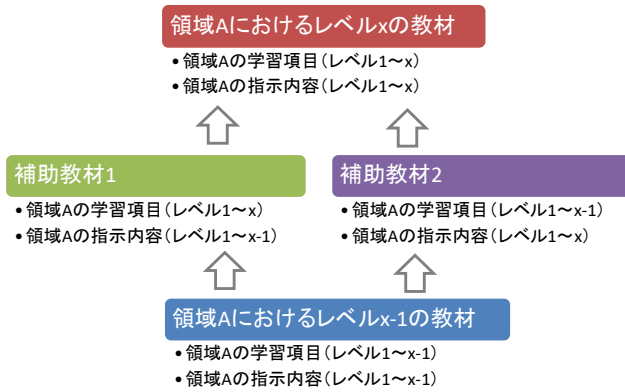


図3：教材の作成方針

5.2 プログラミングと開発(レベル7)における作成方針例

プログラムと開発のレベル7では、3つの学習項目を含むプログラムについて理解し、「モジュール単位でのエラー検出、修正」の指示内容が実践できる段階を目指している。

また、レベル6の教材では、レベル1~6の学習項目(10個)を含むプログラムを理解し、レベル1~6の指示内容(4個)の指示内容が実践できる段階を目指している。

補助教材1は、レベル7の3つの学習項目とレベル6の4個の指示内容から組み合わせで作成する。組み合わせる指示内容については、ひとつ前のレベルであるレベル6の「構文エラーの検出、訂正」を採用した。具体的には、それぞれの学習項目に関連するエラーとして変数の定義(ローカル、グローバル)や引数の定義、ループの終了判定やループ内の処理などにエラーを含むプログラムを用いてエラーの検出や修正をさせる。

補助教材2は、レベル6の10個の学習内容とレベル7の指示内容である「モジュール単位でのエラー検出、修正」から組み合わせで作成する。関数関連の指示内容なので、組み合わせる学習内容はレベル6の「変数を利用する関数」を採用する。具体的には、変数を利用する関数を含むプログラムからメイン関数以外のモジュールを個別に抜き出し、エラーを混入する。エラーを含む個別のモジュールを用いて、モジュール単位でのエラー検出・修正をさせる。

プログラムのレベル7の教材は、補助教材1と2を組み合わせた教材である。具体的には、レベル7の学習項目である変数や引数、ループにエラーを含むプログラムを用いる。補助教材2と同様にエラーを含む個別のモジュールを修正させ、各モジュールを利用するメインルーチンを作成させる。

6. プログラミング教材の具体例

6.1 アルゴリズム(レベル5)

アルゴリズムのレベル5では対応する学習項目がないので、アルゴリズムのレベル4の教材に指示内容のレベル5

の内容である「アルゴリズム作成のガイドラインの理解」と「類似点や相違点の活用」を追加した補助教材2を作成する。

「アルゴリズム作成のガイドラインの理解」の教材では、アルゴリズム作成のガイドライン[8]に基づいて作成されたアルゴリズムを示すことでアルゴリズムの構造を理解させる。アルゴリズム作成のガイドラインでは、ブロック構造、条件分岐、繰り返し、ルーチンだけを用いてアルゴリズムを記述する。条件分岐や繰り返しを記述する際には「

である限り以下の処理を繰り返す。」のように「以下の処理」のフレーズを含み、「以下の処理」の具体的な内容はインデントを下げて具体的に記述する。また、教材に用いるアルゴリズムは、レベル4の学習内容として「ループ」、「条件分岐」、「入力に応じた出力」、「ルーチン」を含むものとする。

「類似点や相違点の活用」の教材では、パンダのSpriteを移動させてピラミッドを描画させる(図4)。ピラミッドを描画する際には、ピラミッドを3つの3角形に分解し、あらかじめ作成した3角形を描画する関数を活用する。



図4：ピラミッドを描画するプログラムの実行結果

3つの3角形は同じ形ということ、3角形を描画するたびに描画の開始点や角度を変更する必要があることを用いて「類似点や相違点の活用」を学習する。サブルーチンに関しては、アルゴリズムのレベル5では引数や変数を用いないので、移動距離は固定する。また、描画する図形が複雑でmBotだと軌跡がわかりにくいので、Spriteを移動させる。Spriteの向きと位置の変化を視覚的にわかりやすくするために、各動作の後に1秒待機する。プログラム全体を図5に、メインルーチンと3角形を描画するサブルーチンを以下に示す。

メインルーチン

1. 緑のフラグがクリックされるまで待機する。
2. 以前書いた線を全て消す。
3. 3角形を描画する。
4. Spriteを移動、向きを調整し、3角形を描画する。
5. Spriteを移動、向きを調整し、3角形を描画する。

3角形を描画するサブルーチン

1. ペンを下ろす。
2. 以下を3回繰り返す。
 - 2.1. 60歩進み、右方向に120度曲がる。

3. ペンを上げる .



図 5: ピラミッドを描画するプログラム

6.2 プログラミングと開発 (レベル 7)

プログラミングのレベル 7 では対応する学習項目, 指示内容をそれぞれレベル 6 の教材に追加した補助教材 1 2 を作成する .

変数のスコープを学ぶ補助教材 1 は, 時間を数えるパンダを複数作成するプログラムである (図 6). このプログラムはパンダ毎のローカル変数と複数のパンダで共有しているグローバル変数を持つ . 最初に作成されたパンダはパンダの総数を数える . クリックするたびにパンダの数が増え, 新たに作成されたパンダは 10 秒経過するまで時間を数える . 時間の変数をローカルからグローバルに変更した場合, 時間を数えている間に新たなパンダを作成すると, 時間が 0 に初期化されてしまい, 正しく数えることができなくなる . ローカルとグローバルをそれぞれ体験させることで変数のスコープについて学習できる . 変数, アルゴリズム, プログラム (図 7) を以下に示す .



図 6: 「時間を数えるパンダ」プログラムの実行結果

変数

- 人数: パンダの総数
- 時間 (ローカル): 個別に数える時間
- 時間 (グローバル): 共通に数える時間

メインルーチン

1. 「人数」を 0 にする .
2. 「人数」が 20 になるまで以下の処理を繰り返す .
 - 2.1. クリックした場合以下の処理を行う .
 - 2.1.1. パンダを作成する .

- 2.1.2. 「人数」を 1 増やし, 表示する .
- 2.1.3. 0.2 秒待つ .

新しいパンダを作成した時に実行するルーチン

1. クリックしたところにパンダを描画する .
2. 「時間」を 0 にする .
3. 以下の処理を 10 回繰り返す .
 - 3.1. 1 秒ごとに「時間」を 1 増やし, 表示する .

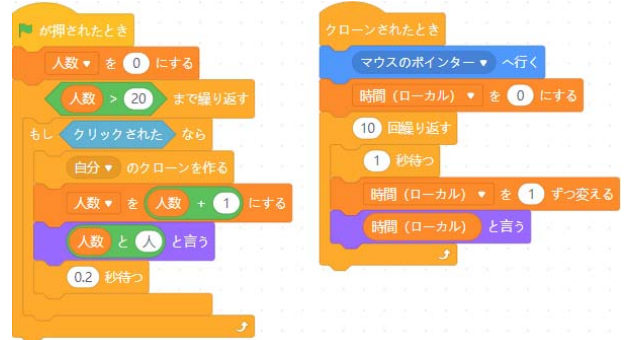


図 7: 変数のスコープを学習するプログラム

引数の受け渡しを学ぶ補助教材 1 では, 引数がある関数を含むプログラムを用いる . 前判定ループと後判定ループの違いについては, Scratch では後判定ループにしか対応していないため, 前判定ループについてはテキストを用いて学習させる .

モジュール単位でのエラー検出, 修正を学ぶ補助教材 2 では, 引数がない関数を含むプログラムを用いる . モジュール単位で分割したプログラムを配布し, 各プログラムを修正させることで学習させる .

6.3 ハードウェアと処理 (レベル 8)

ハードウェアのレベル 8 では対応する指示内容がないので, ハードウェアのレベル 7 の教材に学習項目のレベル 8 の内容である「低水準言語プログラミング」「ムーアの法則」「マルチタスキング」を追加した補助教材 1 を作成する .

低水準言語プログラミングについては, mBot を用いた教材では対応できないので, CASL シミュレータを利用する . CASL は基本情報技術者試験で使用されるアセンブリ言語である . ムーアの法則, マルチタスキングについてはテキストを用いた学習で十分対応できると考えられる .

6.4 データとデータ表現 (レベル 6)

データのレベル 6 では対応する指示内容がないので, データのレベル 5 の教材に学習項目のレベル 6 の内容である「2 進数の加算」を追加した補助教材 1 を作成する .

2 進数の加算については, mBot に接続した 7 セグメントディスプレイに計算結果を表示する教材を用いる . 赤外線リモコンを用いて 2 つの 2 進数を入力し, 桁毎にリストに格納する . 入力した 2 進数を加算し, 結果をディスプレイに表示する . ディスプレイは 4 桁まで表示できるので, 加算する 2 進数は 3 桁までとする . 始めに 1 桁目を加算し, 1 桁目の結果と桁上がりを求める . 2 桁目以降は 2 進数の

各桁と繰り上がりを加算し、各桁の結果と桁上りを求める。計算結果はリスト[1,0,1]に格納しているが、ディスプレイに表示する際にリストを整数(101)に変換する。

変数、アルゴリズム、プログラムを以下に示す。

変数

- 1つ目の2進数：2進数の各桁をリストで保持
- 2つ目の2進数：同上
- 計算結果(リスト)：計算結果の2進数をリストで保持
- 計算結果(整数)：計算結果の2進数を整数で保持
- 1桁の計算結果：各桁の計算結果
- 繰り上がり：各桁の繰り上がり
- 計算する桁：計算する桁の位置
- 2進数の最大桁数：入力した2進数の最大桁数

メインルーチン

1. 2進数を入力する。
2. 1桁目を加算し、「1桁目の計算結果」を「計算結果(リスト)」に追加する。
3. 「計算する桁」を2にする。
4. 「2進数の最大桁数」を求める。
5. («計算する桁」>「2進数の最大桁数」)まで以下の処理を繰り返す。
 - 5.1. 「計算する桁」番目の2進数と「繰り上がり」を加算し、「1桁の計算結果」を「計算結果(リスト)」に追加する。
6. もし「繰り上がり」が1なら、「繰り上がり」を「計算結果(リスト)」に追加する。
7. 計算結果をリストか整数に変換しディスプレイに「計算結果(整数)」を表示する。

2進数を入力する

リモコンの「0」「1」ボタンで2進数を入力し、リストに追加する。「設定」ボタンで入力を確定する。ボタンを入力するたびにブザー音を鳴らす。

2つの1桁2進数を加算する(半加算器)

2進数の1桁目を加算し、計算結果を「1桁の計算結果」と「繰り上がり」に代入する

2つの1桁2進数と繰り上がりを加算する(全加算器)

2進数の各桁(2桁目以降)と繰り上がりを加算し、計算結果を「1桁の計算結果」と「繰り上がり」に代入する

計算結果をリストから変数に変換する

ディスプレイに表示するために計算結果をリストから整数に変換する。

7. おわりに

本研究では、CASが策定したガイドラインを基に、中学校・高校段階に対応するmBotロボットを活用したプログ

ラミング教材を開発した。この教材は仕様、アルゴリズム、プログラム、ハードウェアを関連付けながらアルゴリズム的思考を学習できる。また、プログラミングに関する学習内容と課題内容を段階的にステップアップさせることで、生徒の能力に応じた教材を系統的に提供できる。

今後は、プログラミングを指導する教員からのレビューおよび教材の評価実験を行いたい。また、本教材の趣旨を活かして教育を行うことのできる教員の養成にも取り組みたい。

謝辞 本研究は、JSPS 科研費17K01036の助成を受けたものです。

参考文献

- [1] 小学校学習指導要領(平成29年告示),
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2019/03/18/1413522_001.pdf
- [2] 【技術・家庭編】中学校学習指導要領(平成29年告示)解説,
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2019/03/18/1387018_009.pdf
- [3] CAS Computing Progression Pathways KS1 (Y1) to KS3 (Y9) bytopic Computing At School,
<http://community.computingatschool.org.uk/resources/1692/single>
- [4] 岩崎誠, 掛下哲郎, CASガイドラインに基づく小学校プログラミング教材の提案:mBotロボットとアルゴリズム学習用ツールキットの活用, 情報処理学会 情報教育シンポジウム, 9-16, 2018
- [5] 中学校学習指導要領(平成29年告示),
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2019/03/18/1413522_002.pdf
- [6] 高等学校学習指導要領,
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2018/07/11/1384661_6_1_2.pdf
- [7] 日本学術会議, 情報教育の参照基準(2019.2.23版),
<https://www.edu.cc.uec.ac.jp/~ka002689/9282981/ieduref-0223.pdf>
- [8] 掛下, アルゴリズム作成のガイドライン,
<https://www.cs.is.saga-u.ac.jp/syllabus/GuideLine/algorithm.html>