

中学数学におけるプログラミング教育の提案

- プログラミングの基礎力を育む中学数学の役割 -

吉田 賢史*1・篠田 有史*2・松本 茂樹*3・大脇 巧己*4
Email: k.yoshida@waseda.jp

- *1: 早稲田大学高等学院
- *2: 甲南大学共通教育センター
- *3: 甲南大学知能情報学部
- *4: NPO 法人 Active Learning Association

◎Key Words プログラミング的思考, 学習のスタイル, 学習方略

1. はじめに

小学校では、2020 年度から新学習指導要領が全面实施され、今年度から先行実施されている小学校もある。新学習指導要領では、児童・生徒に次の 3 つの力を育むことが示されている。1 つめは、論理的思考力、2 つめは、プログラムの働きや情報社会が情報技術 (ICT) によって支えられていることに気付くこと、最後に、身近な問題の解決に主体的に取り組む態度を育むことが挙げられている。つまり、コンピュータ等を上手に活用してよりよい社会を築いていこうとする態度などを育むことが求められている。一方、高校の情報科では、必修科目においてプログラミングやシミュレーションによって問題を発見・解決する活動が展開される。中学校においては、技術科の情報の技術においてプログラミングが取り上げられているが、数学科においては、適切な活用に留まっており、プログラミング的思考に関する記述はない。

そこで、本稿では、改訂における新しい項目「プログラミング教育」について、中学の数学科における取組に焦点を当て、ICT 活用能力の基礎となる「プログラミング教育」について整理し、中学の「連立方程式の利用」を題材に、通常授業における「プログラミング的思考」を意識した展開について述べる。また、授業を展開する際に、生徒一人ひとりが得意とする学習スタイルとの関連についても述べる。

2. プログラミング教育

2.1 プログラミングとプログラミング的思考

プログラミングは、「プログラミング言語」や「ソフトウェア・プログラミング」などの言葉が示すように、コンピュータにある目的の処理をさせるために必要な処理手順 (プログラム) を作成することである。それ故、手順を表現する手法として、プログラミング言語が注目され、プログラム言語でプログラムを表現すること (コーディング) に意識が向かいがちである。

しかしながら、小学校段階におけるプログラミング教育の在り方について (議論の取りまとめ)⁽¹⁾において、プログラミング教育は、

子供たちに、コンピュータに意図した処理を行うように指示することができるということを体験させながら、将来どのような職業に就くとしても、時代を超えて普遍的に求められる力としての「プログラミング的思考」などを育成するもの

とされ、プログラミング的思考は、

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合せが必要であり、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、記号の組合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

と定義されている。このように、「コンピュータに意図した処理を行うように指示することができるという体験」には、コンピュータ・プログラミングは必要である。

つまり、プログラミング的思考は、プログラミングの考え方にもとづく論理的思考であり、以下の手順で示される問題解決の思考方法である。

1. 課題を理解する。
2. 課題を解決するための要素を挙げる。
3. 要素を記号で表す。
4. 要素と要素を結ぶ。(シーケンシャル処理)
5. 解決されたかチェックする。

この 5 つの手順の特徴は、作業的な要素を多く含んでいることである。プログラミング的思考において、重要な活動は、第 2 項目と第 5 項目であると考えられる。実際に、IE-School 事業などで見られる実践事例においても、グループワークが取り入れられており、多様な意見を取り入れながら試行錯誤を繰り返し、課題を解決しているケースが多く見られる。このような、プログラミング的思考を取り入れたプログラミング教育では、試行錯誤を容易に体験させることができ、様々な意見を取り入れた実験を次々おこなうことができる。そのため、児童の集中力が継続され、思考を楽しむきっかけになっていると考えられる。

2.2 コンピューテショナル・シンキング (CT : Computational Thinking)

さて、このように小学校で育まれたプログラミング的思考は、中学校の技術科に引き継がれ、

小学校におけるプログラミング教育の成果を生かし、発展させるという視点から、従前からの計測・制御に加えて、双方向性のあるコンテンツに関するプログラミングや、ネットワークやデータを活用して処理するプログラミングも題材として扱うことが考えられる。

と記されている^②。つまり、小学校では、コンピュータ内の処理が中心であるが、中学校では、I/O処理やUIデザインに関する授業が展開されると予測される。

例えば、LEGO® MINDSTORMS® を使用した距離センサーを用いた事例^③では、壁への衝突回避プログラムの作成過程において、パラメータ調整の実験方法にグループによって差異が生じ、完成までの時間も異なった。試行錯誤し、それが直ぐに現象として反映されることから、興味は持続し、体験した生徒全員がゴールできた。

しかしながら、この活動には留意しなければいけない点があると考えている。スタートとゴールが与えられた課題では、論理的な思考が育まれない可能性があり、パラメータを調整するという“当てモノ”で終わってしまう危険性がある。このことは、新井が、

フレームが決まっていると、子どもは教える側が期待しているのとは別の方法で、そのフレームのときだけ発揮できる妙なスキルだけを偏って身につけてしまうのです。

と述べている^④ように、プログラミング教育においても“当てモノ”で解決できるという誤った成功体験にならないように留意しなければならない。このことは、技術科だけでなく、他教科においてもプログラミング的思考を育む上で重要な視点であると考えられる。

しかしながら、数学においては、プログラミング的思考だけでは不十分であり、システム的な思考が必要であると考えている。すなわち、対象を個々の要素に切り分け、相互の関係性を考慮してモデル化をおこない、モデルを定式化できる力の基礎を身につけることは、高等学校情報科におけるプログラミング、データサイエンスの単位につながると考えている。

そこで、プログラミング的思考のコアとなる思考をCTと考え、CTを次のようなステップで考えられる思考、

- CT : 1. (問題解決/課題解決において) 目的を明確化
- CT : 2. 対象の分解 (要素を挙げることが) できる
- CT : 3. 抽象化した要素間の関係性をモデル化できる
 - 構成要素の抽出 (問題・課題の抽象化)
 - 要素間を関連づける (モデル化)
- CT : 4. モデルが正しいか検証・評価できる
(検算・吟味 / シミュレーション)
- CT : 5. モデルを定式化できる
 - 類似のモデルに適用できるように一般化と定義する。

例題 2種類の品物A, Bがある。A 3個とB 1個の重さは880g。
1 A 1個とB 2個の重さは560gである。A, Bそれぞれの1個の重さを求めなさい。

解答 A 1個の重さを x g, B 1個の重さを y g とすると

$$\begin{cases} 3x + y = 880 & \dots\dots ① \\ x + 2y = 560 & \dots\dots ② \end{cases}$$

$$\begin{array}{r} ① \times 2 \quad 6x + 2y = 1760 \\ ② \quad \quad -) \quad x + 2y = 560 \\ \hline \quad \quad \quad 5x \quad = 1200 \\ \quad \quad \quad \quad \quad x = 240 \end{array}$$

$x = 240$ を ① に代入して解くと $y = 160$
 これらは問題に適している。
答 A 1個 240 g, B 1個 160 g

図 1. 連立方程式の利用問題例(加減法記載)^⑤

3. CTと授業展開

3.1 教材

数研出版 体系数学1 (代数編) pp.91-93 を活用し、CTを取り入れた授業の事例を挙げる^⑥。担当クラス62名を対象に授業をおこなった。「連立方程式の利用」では、

1. 未知数を x, y とおく
2. 文章から方程式を2つ つくる
3. 連立方程式を解く
4. 吟味する
5. 解を記述する

というステップで、解答を作成する(図1)。この例題の解答をCTの各ステップと対応づける。

CT:1.問題文から目的を明確化する

このステップでは、問題文を分割し求めたいものを明確にする。例えば、図1の例題の問題文は、

- 2種類の品物A, Bがある。 . . . ①
- A 3個とB 3個の重さは880g, . . . ②
- A 1個とB 2個の重さは560gである。 . . . ③
- A, Bそれぞれ1個の重さを求めなさい。 . . . ④

に分けられる。目的は④に記述されている。

CT:2.問題文から対象を分解する

ここでは、①~④の文章を分解し、問題を解決するための要素を挙げる。各文章から、

- ① から 2種類 / 品物A, B
- ② から A 3個 / B 3個 / 重さは880g
- ③ から A 1個 / B 2個 / 重さは560g
- ④ から A, B / 各1個の重さ / 求める

と分ける。

CT:3.構成要素の抽出と要素間の関連付け

このステップでは、①~④の情報から問題を数式で表現する。数式で表すためには、求めたいものを未知数として文字を用いて表す。

- ④ A を x g, B を y g とする。
- ② A 3個: $3x$ / B 3個: $3y$
文章から, $3x + 3y = 880$. . . ⑤
- ③ A 1個: x / B 2個: $2y$
文章から, $x + 2y = 560$. . . ⑥

CT:4.モデルの検証と評価

⑤, ⑥の連立方程式を解き、解が適切か確かめる。例

題の方程式の解法は、加減法を用いている。処理の手順は、

- 解を求める
- 算出した計算値が⑤, ⑥の方程式を満たすかチェックする。(検算)
- 方程式の解が、問題に適しているかチェックする(吟味)

となる。

3.2 指導の留意点

このCT:1~CT:4のステップは、教員にとって普通の思考の流れであるが、生徒は、別の捉え方をする。つまり、生徒は、CT:2, CT:3のステップを何も考えずに作業として教科書や参考書で与えられたフレームを利用し、数値の当てはめをおこない、数値を導く。CT:4においては、「これらは、問題に適している」は決まり文句としての認識でしかない。これでは、思考を伴わない作業となり、CTを育むことはできない。

そこで、我々は解説に重点を置くのではなく、「問いかけ」に重点を置く。例えば、図1の「解答1行目『Aをx g, Bをy gとする。』は、何故必要なのか?」、あるいは、「『Aをx g, Bをy gとする。』としないと何が不便なのか?」などである。

また、図2のような例題⁶⁾があると、CT:5のステップを意識させることが可能である。図2では、解答の5行目において「①と②を連立方程式として解くと」と記述されており、図1のような加減法の具体的計算は書かれていない。つまり、方程式を解くという作業を別作業(サブルーチン)として扱っている。

4. プログラミングと数学

4.1 類似点

CTと数学の問題の解法を対応づけると、問題解決のためのプログラミングと数学の解法手順は類似点が多く、STEM教育における数学の役割は大きいと考えられる。

図2を例に解答とプログラミングとの対応を考えてみる。

- 1行目は、変数定義 (例えば, float x, y;)
- 2-4行目は、立式 (モデル化)
(e. g. $x+y=600; 5x+3y=2100;$)
- 5行目は、関数のCall
(e. g. $Solve(x+y-600, 5x+3y-2100, (x, y));$)
- 6行目は、関数の戻り値を変数に代入
(e. g. $[x, y]= Solve(x+y-600, 5x+3y-2100, (x, y));$)
- 7行目は、値のチェック
(e. g. $if(x<0 \text{ or } y<0) \{ \dots \} else \{ \dots \};$)

このように、CTによって、問題が定式化されると、プログラムで表現することが可能である。換言すると、プログラムで表現するためには、問題を系統的に捉え、定式化するCTの力が必要である。

計算機に処理をさせるためのプログラム言語は色々あるが、いずれも、目的に対する処理の手順を記述しているに過ぎない。この記述において、

- 手順を計算機に指示すること、

- 同じ処理は1つにまとめ他の処理と共有できるように設計すること、
- 出てきた値が次に使用してよい値になっているか、あるいは、異常値が返ってきていないかチェックすること

は、中学や高校の数学の問題を考えるときの手順と非常に似ている。特に第3項の吟味においては、検算と考えている生徒が多く、「これらは、問題に適している」は何故必要か考えさせ、判断基準はどこにあるのか。単位に注目、未知数の条件へと導く必要がある。その際、方程式は無名式で表されていることに注意を向けるように指導する必要があると考えている。

教科書のように例題があり、類似問題が練習問題で与えられているケースが多いが、例題で、定式化された解答が与えられているケースが多く、生徒は、過去の経験から類似問題を検索し、定式化された例題の解答に数値を当てはめれば、見た目の正解を導くことができる。これを、これを検索型解法と呼ぶことにする(図3)。この解法は、指導によっては、単なるパターンマッチングとなり、CTを阻害するので指導に注意が必要である。

4.2 CTを意識した授業

定期考査などの解答の結果だけでは、パターンマッチングをした結果を記述しているのか、思考した結果を記述しているのかを測ることはできない。CTを育むためには、抽象的なモデルをいかに残させるか、さらに、生徒の思考の過程をどのように残させるかが課題となる。

思考の過程を残した記述は、試験の対策ではなく、日常の活動に取り入れる必要がある。結果だけに意識を向け、解答を真似して解法を覚えるという学習習慣を断ち切らなければならない。自分の考えを記述させるために、指導

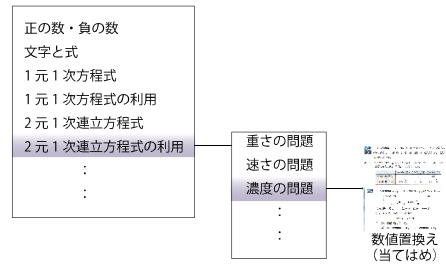


図3. 類似問題検索型解法

例題 10%の食塩水と6%の食塩水を混ぜ合わせて、7%の食塩水を600g作りたい。食塩水は、それぞれ何gずつ混ぜ合わせればよいか答えなさい。

考え方 10%の食塩水をxg, 6%の食塩水をyg混ぜるとすると、食塩水の重さの関係と食塩の重さの関係は、次のようになる。

	10%の食塩水	6%の食塩水	7%の食塩水
食塩水の重さ(g)	x	y	600
食塩の重さ(g)	$x \times \frac{10}{100}$	$y \times \frac{6}{100}$	$600 \times \frac{7}{100}$

解答 10%の食塩水をxg, 6%の食塩水をyg混ぜるとすると

$$\begin{cases} x+y=600 & \dots \text{①} \\ x \times \frac{10}{100} + y \times \frac{6}{100} = 600 \times \frac{7}{100} & \dots \text{②} \end{cases}$$

②を整理すると $5x+3y=2100$ ③

①と③を連立方程式として解くと

$$x=150, y=450$$

これらは問題に適している。

答 10%の食塩水 150g, 6%の食塩水 450g

図2. 連立方程式の利用問題[加減法記載なし]⁶⁾

において留意する点は、解答を正解か不正解かを判定するのではなく、記述された解答のどこが正しくて、どこが矛盾しているか、また、不十分なのかを、教室全体で、生徒同士の意見を交えながら展開する必要がある。

4.3 授業展開の問題点

ところが、一人ひとり異なる頭の中のイメージは異なり、そのイメージの伝わり方もひとり一人異なる⁶⁾ため、言葉だけで共有する事は難しい。例えば、論理的に解を導き説明ができる生徒もいれば、図4に示すように、色々思いつき、整理されないまま答えを導き上手く説明できない生徒もいる。前者の場合は、CTに適した思考であるが、後者はCTを育むためには、図4の下に示すように矢印で関連する要素を結び、結論から順に、問題に遡る手順で解答を作り上げる必要がある。これをネットワーク型解法と呼ぶことにする。

このような思考の差異は、教科書や参考書などの考え方やヒントの与え方にも現れている。例えば、図2の場合は、“考え方”として表が示されているが、参考書などではピーカーの絵を使って考え方を示しているものもある。

演習を重視した授業展開では、検索型解法の方がペーパーテストの得点に繋がりやすく、そのベースとなるデータを正確に記憶した方が効率的である。この解法は、記憶に依存するため、暗記とパターンマッチングが得意な生徒に向いている。それ故、CTについても、パターンマッチングを得意とする生徒に指導する際には、CT:1~CT:4の各ステップの手順を伝えると伝わりやすいと考えている。

一方、ネットワーク型解法が得意な生徒は、ペーパーテストの成績には繋がりにくい。解答に必要な要素が論理的に順序よく思いつくことはなく、バラバラに思いつく傾向が強い。そのため、先ず、思いついた要素を挙げ、次にそれらの関係を矢印で結んでから解答を書いてよいという環境を用意する必要がある。この解法は、検索型解法に比べ多くのステップを必要とするが、CTのステップに似ている。

5. おわりに

検索型解法を得意とする生徒は、過去の経験から問題解決に取組み、ネットワーク型解法の生徒は、過去の経験にとらわれず自由な発想で試行錯誤する。また、解法を観て冗長的な表現や重複を見いだし論理的で効率的な解法に書き換えるのが得意な生徒もいる。さらに、いろいろな意見が出されたとき、その意見のよいところを見いだしまとめるのが得意な生徒もいる。それぞれの長所を活かして、1つの問題に取り組むことで、お互いに苦手なCTステップを補完することができ、問題解決への気づきを促すことが可能であると考えている。

数学は、与えられた問題を系統的に捉え、定式化するCTの力を育む教材が多く存在する。ところが、生徒は、点数偏重の学習環境の中、スピードと正確さという尺度で測られ、定式化されたものを処理することに終始する。さらに、点数を得るために、解答の結論を聞き出し記憶しようとする。これは、プログラムを実行しているPCそのもので、CPUやメモリのスペックを測っているに等しいし、

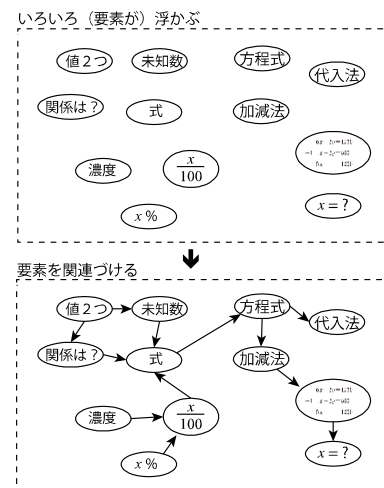


図4. ネットワーク型(図的)解法

AIが学習する過程と同じである。

PCと同じ活動をさせ、AIと同じ学習を経験させることが、プログラミング教育ではない。重要なことは、思考のプロセスであるCTを取り入れた授業であるが、CTの各ステップ全てを最初からこなせる生徒は少ない。対面授業では、検索型解法やネットワーク型解法の是非ではなく、生徒一人ひとりの強み(認知特性)を活かしながら思考を深めることが必要となる。

例えば、知覚認知特性において発想型が優勢の生徒はネットワーク型解法を得意とするし、秩序型が優勢の生徒は、検索型解法、分析型が優勢の生徒は、CT:5が得意、社交型の生徒は、人の意見や行動から全体をまとめようと動く。生徒一人ひとりが、自身の特性を活かすことができれば、教室内(グループ)がチームとして、CTを展開できるようになると考えている。苦手を克服することも大切であるが、苦手なところは、仲間の力を借りるという姿勢を育むことの方が、重要であると考えている。

謝辞

本研究の一部は、早稲田大学2018年度特定課題研究費基礎助成(2018K-415)によるものである。

参考文献

- (1) 文部科学省(2017)小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ),文部科学省,http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm, 2018.04.01 Access.
- (2) 文部科学省(2017)中学校学習指導要領解説 技術・家庭編,文部科学省,http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afiledfile/2017/12/27/1387018_9.pdf, 2018.04.01 Access.
- (3) Minecraftで未来のプログラマーを育成:「Education Day」レポート, C|net Japan, <https://japan.cnet.com/article/35074304/>, 2018.05.05 Access.
- (4) 新井紀子:“AI VS. 教科書が読めない子どもたち”, pp.229-232, 東洋経済新報社(2018).
- (5) 岡部直治:“体系数学1(代数編)”, pp.91-93, 数研出版株式会社(2014).
- (6) 吉田賢史, 篠田有史, 大脇巧己, 松本茂樹:“能動的学習を刺激する認知思考特性と思考表現特性を利用した学び”, PC Conference, pp.241-244(2016)