

群 教 セ	G08 - 02
	令 2.275 集
	工業

「プログラミング技術」における プログラムの理解度を高める指導の工夫

—メモリ内部の様子を可視化したラミネートシートの活用と
協働的な活動を通して—

特別研修員 原 裕哉

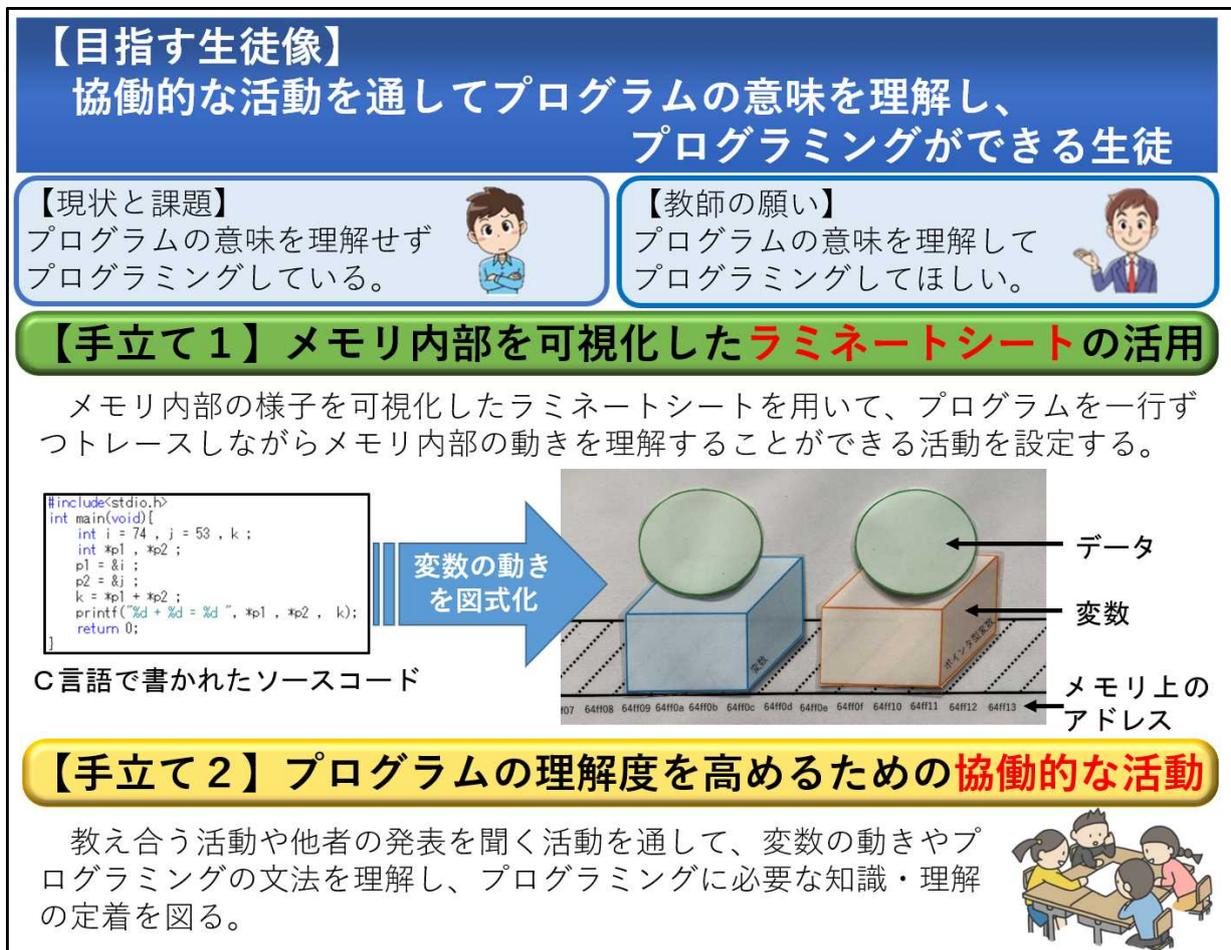
I 研究テーマ設定の理由

令和2年度県立学校教育指導の重点には、「工業の各分野に関する基礎的・基本的な知識と技術を習得させ、工業と社会の発展を図る創造的な能力と実践的な態度を育成する」とあるように、各教科において基礎・基本の定着を図り、主体的に学ぶ態度や創造的な能力の育成が必要であると考えます。

研究協力校の生徒の実態として、プログラミング技術における理解度が高い生徒と、そうでない生徒の差が大きい事が課題として挙げられる。理解度が高い生徒はプログラミングの楽しさを実感しており、難しい問題にも挑戦していこうとする意欲が伺える一方、プログラミングに苦手意識の強い生徒は考えることすら諦めてしまう生徒も多数見受けられる。苦手意識の強い生徒の特徴として、プログラムの文法を理解していなかったり、変数内のデータの動きが読み取れなかったりすることが挙げられる。そこで、基本的なプログラムを一行ずつトレースしながら、メモリ内部の動きを可視化して考えさせたい。さらには、協働的な活動を通して情報交換を行い、理解度を高めたいと考える。

II 研究内容

1 研究構想図



2 授業改善に向けた手立て

プログラミングの理解度を高めるために、次の二つの手立てを用いて実践授業を行った。

<手立て1> メモリ内部の様子を可視化したラミネートシートの活用

メモリ内部の様子を可視化したラミネートシートを用いて、プログラムを一行ずつトレースしながらメモリ内部の動きを理解することができる活動を設定する。

<手立て2> プログラムの理解度を高めるための協働的な活動

教え合う活動や他者の発表を聞く活動を通して、変数の動きやプログラミングの文法を理解し、プログラミングに必要な知識・理解の定着を図る。

手立て1では、通常見ることができないコンピュータのメモリ内部の様子を可視化した。プログラムを一行ずつトレースする過程で、変数の値がどのように変化していくのか理解させるためにラミネートシートを使用した。ラミネートシートを活用することで、マーカーで書いたり消したり、移動させたりすることが容易になり、自分の手を使って思考を表現しやすくなると考えた。

手立て2では、【個人ワーク】【グループ（ペア）ワーク】【発表】の順に、段階的に思考の過程を踏むことで、理解度を高められると考えた。特に、【グループ（ペア）ワーク】では、二、三人の少人数の班とすることで、生徒間での積極的なコミュニケーションが取れるよう促した。

Ⅲ 研究のまとめ

1 成果

- コンピュータのメモリの内部という実際には目に見えないものを可視化することで、変数の変化のイメージがし易くなり、プログラムの理解度が高まった。生徒自身の手を使ってラミネートシートを操作するというアナログ的な作業をすることで、より考えが深まり、理解度の向上につながったのではないかと考える。授業の最後に行った振り返り小テストでは80%以上の正答率となった。
- 今回の授業に対する生徒の意欲は非常に高く、ラミネートシートを使って自分の思考を表現し、積極的にグループで話し合う姿勢が強く表れていた。ラミネートシートが協働的な活動の助けとなり、自分の思考を表現するために有効的な手段になっていたと考える。
- 生徒自身が自分を評価するルーブリック評価により【個人ワーク】【グループ（ペア）ワーク】【発表】の過程を踏むことで、段階的に理解度が高まる傾向が見ることができた。また、感想文から「グループで説明し合うことで理解が深まった。」「ポイントの仕組みがよく分かった。」「講義形式の授業よりも、理解が深まった。」といった意見が多く見られたことから、つまづいてしまう生徒が多い本単元の第一ステップとして、研究で用いた手立ては理解度を高めるために効果的であったと考える。

2 課題

- 本研究で用いた課題では、与えられたプログラムからメモリの変化を表すものであって、班によって多少の違いはあるものの、解へのプロセスは一つになってしまう。グループワークを行う中で、解は一つでも様々なプロセスが考えられる問題や、生徒の自由な発想から、最適解や納得解を求めるような課題を設定する必要がある。
- グループワークを行った際、残念ながら対話的な活動が活発に行われなかった班も見受けられた。当事者意識をもたせるため、生徒全員が自分の考えを班員に説明する機会を設けたり、発表の際に一人の生徒だけが発言するのではなく、全員の生徒に発言機会を与えたりするなど、グループワーク時の態度目標を明確にする必要がある。

実践例

1 単元名 「配列とポインタ」 (第2学年・2学期)

2 本単元について

本単元はプログラミング言語Cにおける重要な単元となる反面、イメージすることが難しく、生徒の理解度のばらつきが多いことが特徴である。まず、メモリ上にはアドレスという番地に当たるものが割り振られていることを理解した後、アドレスを可視化したラミネートシートを配付する。次に、プログラムをトレースしていく中で変数の値がどのように変化していくのか模式的に考えてみる。その後、グループ編成し、自分の考えをメンバーに説明し、協働的な活動を通して理解を深める。最後に、班としての結論をまとめ、クラス全体で共有する。

目標	<p>配列とポインタに関する実習や演習を通して、次の事項を身に付けることができるよう指導する。</p> <p>ア プログラムを作成しようとする意欲があり、正しく動作するプログラムを完成させる粘り強い態度を身に付けるようにする。(関心・意欲・態度)</p> <p>イ 文法、定数や変数の種類や使用方法、入出力の方法、分岐や繰り返しなどプログラムの実行制御方法などの実際にプログラムを作成するための方法を考えるとともに、与えられた処理を行うための手順を分析し、プログラムとして表現する実践的な能力を身に付けるようにする。(思考・判断・表現)</p> <p>ウ 流れ図からソースプログラムを作成し、コンパイラを用いて実行可能プログラムを作成できるようにする。プログラムを検査することができ、目的どおりに処理を行っているかを確認し、バグがあればそれを取り除くことができるようにする。(技能)</p> <p>エ 定数や変数の種類や使用方法、入出力の方法、プログラムの実行制御の方法など文法を理解し、それを基に実際にプログラムを作成する知識を身に付けるようにする。(知識・理解)</p>	
評価規準	<p>(1) プログラムを作成しようとする意欲があり、正しく動作するプログラムを完成させる粘り強い態度を身に付けている。(関心・意欲・態度)</p> <p>(2) 多量のデータを効率よく扱うために、配列やポインタを用いたプログラムを書くことができ、そのプログラムの意味を説明することができる。(思考・判断・表現)</p> <p>(3) 問題を解決するためのアルゴリズムを考え、それを基にプログラミングすることができ、バグがあれば取り除くことができる。(技能)</p> <p>(4) 配列やポインタを使うための文法を理解し、それを基に実際にプログラムを作成する知識を身に付けている。(知識・理解)</p>	
過程	時間	主な学習活動
つかむ	第1時	・メモリ内部の模式図を用いて考える活動を通して、配列の概念を理解できるようにする。
追究する	第2時	・配列のサイズと要素番号について考える活動を通して、配列のサイズと要素番号の添え字の関係を説明できるようにする。
	第3時	・プログラミング演習を通して文字型配列と文字列の関係を理解する。
	第4時	・Null文字の意味とその働きについてプログラミング演習を通して理解する。
	第5時	・プログラミング演習を通して、文字の複写や文字列の長さを求めることができるようにする。
	第6～8時	・模式図にデータを入れて考える活動を通して、多次元配列を使ったプログラムを書くことができるようにする。
	第9時	・メモリ内部の模式図を用いて考える活動を通して、ポインタの概念を理解する。
	第10～11時	・メモリ内部の模式図を用いて考える活動を通して、&と*演算子を使ったプログラムのメモリ内部の動きを説明できるようにする。
	第12時	・メモリ内部の模式図を用いて考える活動を通して、配列を用いた場合とポインタを用いた場合とで、同じことができることを理解する。
まとめる	第13～14時	・プログラムを作成する活動を通して、ポインタを用いることでスッキリとした分かりやすいプログラムが書けることを理解する。
	第15～16時	・演習問題を解く活動を通して、学習内容の理解を深める。

3 本時及び具体化した手立てについて

本時は全 16 時間計画の第 10 時に当たる。ポインタを理解する上で大切な要素であるメモリ上のアドレスとポインタ型変数の特徴について学ぶ。理解の定着を図るため、以下のように手立てを具体化した。

<手立て1> メモリ内部の様子を可視化したラミネートシートの活用

メモリ内部の様子を可視化したラミネートシートを用いて、プログラムを一行ずつトレースしながらメモリ内部の動きを理解することができる活動を設定する。

<手立て2> プログラムの理解度を高めるための協働的な活動

教え合う活動や他者の発表を聞く活動を通して、変数の動きやプログラミングの文法を理解し、プログラミングに必要な知識・理解の定着を図る。

4 授業の実際

本時の学習課題の設定は、プログラムを一行ずつトレースしていく中で、変数の状態がどのように変化していくかを図式化して表現できるようになることとした。また、なぜそのようになるのかを自分の言葉で説明できるようにすることを意識させた。生徒は【個人ワーク】【グループ（ペア）ワーク】【発表】の順で理解度を高めていくこととした。

(1) 本時の課題と手立ての活用

【個人ワーク】

個人ワークでは学習課題のプログラムを個々で読み取り、メモリの状態を図式化した。プログラムを細分化して考え、その時々メモリの状態を図式化していくが、図1のように最後まで描けた生徒とそうでない生徒がいた。最後まで描けなかった生徒でもノートを見返すなど、ほぼ全ての生徒が制限時間いっぱいまで思考を働かせていた様子が見られた。

【グループ（ペア）ワーク】

グループワークでは個人ワークで考えた事を、ラミネートシートを用いて、思考の過程を共有し合い、理解度を高めた。図2のように、模式的に示したアドレス上に、変数の配置や変数に入るデータなどを書き込みながら説明し合っていた。また、発表に向けたプレゼンテーションソフトのスライド作成も同時に行わせた。変数の状態が変化していく様子を、Webカメラを用いてキャプチャするものであったが、分かりやすく説明ができるよう班員が協力し合って発表用スライドを作成している様子が見られた（図3）。

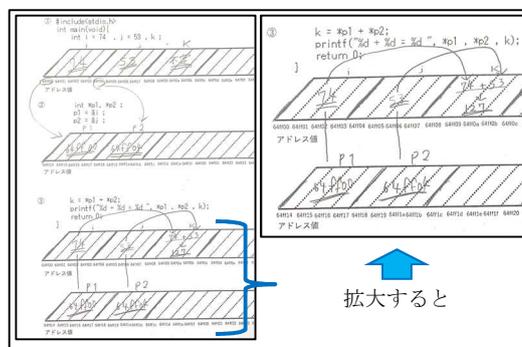


図1 生徒が描いたワークシートと拡大図

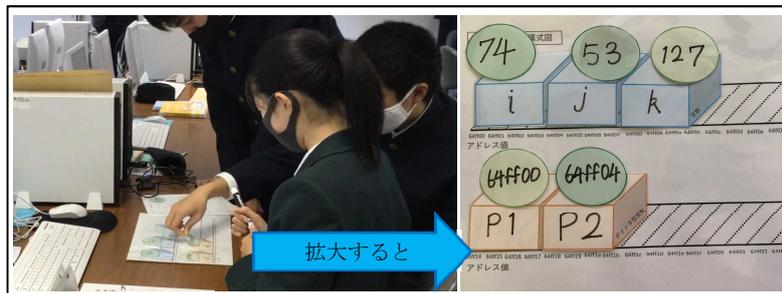


図2 ラミネートシートを用いた協働的な活動と拡大図



図3 Webカメラを用いたプレゼン資料作成

【発表】

全 16 グループのうち、3 グループが 2 分程度のプレゼンテーションを行った。グループワークで作成した発表用スライドをセンターモニターに写し、クラス全体で共有した。発表にあたり、説明している箇所をマウスで指し示すなど、他者に分かりやすく説明する工夫が見られた。発表した全てのグループがプログラムを理解し、変数に入る値や動きを的確に



図4 発表の実践

図式化することができ、正しく説明することができていた。視聴している生徒も発表を熱心に聴いている姿が見られた（図4）。

(2) Google Forms を用いた小テストと学習アンケート

学習内容の理解度を評価するために、Google Forms を用いた小テストを行った。5問の問いに対して満点を採った生徒が全体の59%を占め、4点の生徒が25%、3点の生徒が16%となり、2点以下の生徒はいなかった。また、アンケートにおいて理解度の自己評価をしたが、個人ワーク、グループワーク、発表の過程を踏むに連れて理解度が高まったと回答した生徒が多くなる傾向が見られた（図5）。

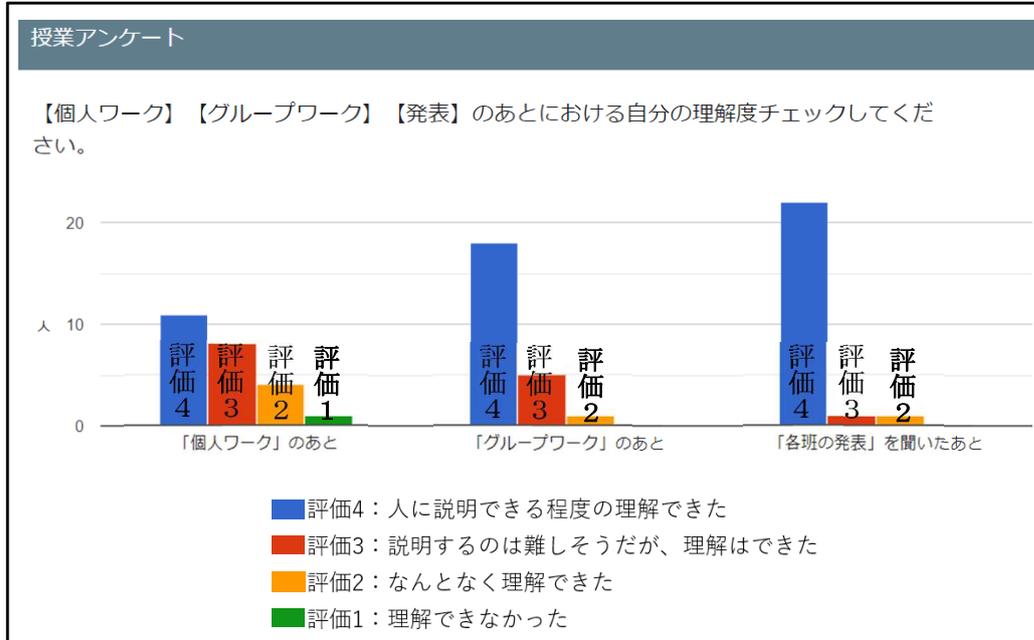


図5 授業後のルーブリック評価

5 考察

プログラミング言語を用いて目的プログラムが書けるようになるには、まずプログラムを読んで何をしているのか理解できる力を身に付けることが必要不可欠である。このことはプログラミングに関わらず、全ての言語を習得する上で共通していることであると考えられる。本研究の手立てであるメモリ内部を可視化したラミネートシートとグループワークにより、生徒たちはC言語で書かれたソースコードから、変数に代入される値や、値の変化を読み取ることができていた。特に、メモリの内部という無形なものを、有形なものとして捉えて考えられたことが大きかったと考える。もっとも、これまでのプログラミング技術の授業では、プレゼンテーションソフトのスライドを提示するのみの講義が多かった。生徒が理解し学習内容を定着させるためには、生徒の思考を働かせることはもちろんのこと、生徒自身の手を使ってラミネートシートを操作するというアナログ的な活動の大切さを発見できたことはとても大きかったと言える。

グループワークの際、生徒同士の対話的な活動が積極的に行われた。ラミネートシートが自分の思考を表現する助けとなり、協働的な活動が促進したと言える。また、クラス全体に向けて発表するという活動を課したことで、どのような伝え方をすれば聞き手に伝わるか自分たちなりの工夫を交えていた班もあり、生徒の主体的な思考を促すことができた。一方、対話的な活動が活発に行われなかった班もあったことは課題として残る。当事者意識をもたせるため、ルールや役割を決めるなど、グループワーク時の態度目標を明確にする必要があったと考える。また、本研究で用いた課題では与えられたプログラムからメモリの変化を表すものであって、班によって多少の違いはあるものの、解へのプロセスは一つとなってしまう。生徒の想像力を養うため、最適解や納得解を求めるような課題を設定する必要があった。今後も対話的な活動を繰り返すことで、生徒同士で協働して学び合い、深い学びにつながることを期待できる。

6 資料

ワークシート

P99 ポインタの基礎。
次のプログラムを読み、メモリ内部の状態を図で示さないか。

```
#include<stdio.h>
int main(void){
    int i = 74, j = 53, k;
    int *p1, *p2;
    p1 = &i;
    p2 = &j;
    k = *p1 + *p2;
    printf("%d + %d = %d ", *p1, *p2, k);
    return 0;
}
```

①. #include<stdio.h>
int main(void){
int i = 74, j = 53, k;

②. int *p1, *p2;
p1 = &i;
p2 = &j;

③. k = *p1 + *p2;
printf("%d + %d = %d ", *p1, *p2, k);
return 0;

メモリ内部を可視化したラミネートシート

メモリ内部の様式図

メモリとアドレス値

メモリ上に置かれた変数

変数に代入されるデータ

Google Forms を使った確認問題

ポインタの基礎 確認問題

*必須

番号、氏名を記入して下さい。(例. 41番、赤城 太郎) *

回答を入力

Q1.ポインタ型変数に格納される値は次のうちどれか? * 1ポイント

- 整数データ「1」や文字データ「A」といった値
- 変数名
- 変数のアドレス値

Q2.メモリ内部の状態が下図のようにになっているとき、「p = &i;」を実行したときの「p」の変数はどのような値が入るか。(半角で入力)

```
int i = 5;
int j;
int *p;
p = &i;
j = *p;
```

回答を入力

Q3.メモリ内部の状態が下図のようにになっているとき、「j = *p;」を実行したときの「j」の変数はどのような値が入るか。(半角で入力)

```
int i = 5;
int j;
int *p;
p = &i;
j = *p;
```

回答を入力

Q4.「int *p」と「j = *p」の「*」の名称を答えよ。* 2ポイント

```
int i = 5;
int j;
int *p;
p = &i;
j = *p;
```

	ポインタ宣言子	間接参照演算子	アドレス演算子
「int *p;」	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
「j = *p;」	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>