



情報科におけるプログラミング教育

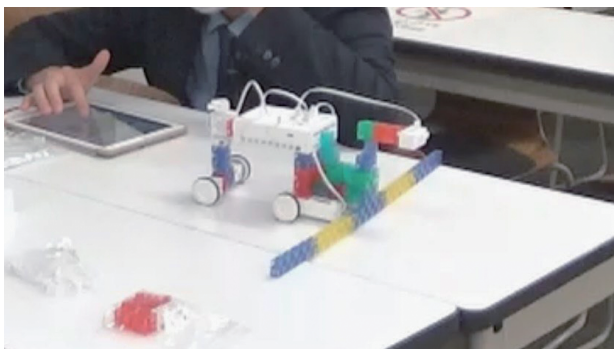
—問題解決活動を組み込んだ授業設計—

東京理科大学 教育支援機構 教職教育センター 准教授 おおaura ひろき 大浦 弘樹

高等学校学習指導要領の改訂で、情報科では必修科目「情報Ⅰ」と選択科目「情報Ⅱ」の履修となりました。その1つである「情報Ⅰ(3) コンピュータとプログラミング」では「アルゴリズムを表現する手段、プログラミングによってコンピュータや情報通信ネットワークを活用する方法について理解し技能を身に付けること」「目的に応じたアルゴリズムを考え適切な方法で表現し、プログラミングによりコンピュータや情報通信ネットワークを活用するとともに、その過程を評価し改善すること」と書かれています。つまり、プログラミングの基本的な知識・技能を身に付けるとともに、問題解決活動の中で活用できる思考力・判断力・表現力の育成が求められています。2023年8月号の情報のヒントでは、Python 指導における Google Classroom の活用というテーマで、Google Colaboratory を用いた授業事例の紹介がありました。そこで、今回はプログラミング教育における問題解決活動を組み込んだ授業設計のヒントについてご紹介します。

プログラミング教材を利用した問題解決活動

わたしが担当する情報科教育法では、Sony Global Education 社が販売しているプログラミング教材「KOOV (クーヴ)」を用いた雪かきロボット・ワークショップを行っています¹⁾。KOOV は、自由に形を組み立てられる「ブロック」と、プログラミングでモーターなどの動きを制御できる「電子パーツ」を組み合



学生が制作した雪かきロボット例

わせてロボットを制作できます。また、Scratch (スクラッチ)²⁾のようなビジュアルプログラミングだけでなく、上級者向けにテキストプログラミングで直接コードを書くこともできます。

本ワークショップでは、机の上に散らばった雪に見立てたブロックを机の外に「できるだけ多く落とす」ロボットの制作を目標に、2、3名のグループに分かれて問題解決活動に取り組みます。その様子を観察すると、学生は大きく「形」と「動き」の2つのアプローチで雪かきの方略を考えます。例えば、できるだけ広い範囲のブロックを動かすために、左右に広がるアームを付けます。また、机の上の広い範囲を動き回れるように、直進と方向転換(回転)を制御するプログラムを書きます。挑戦的なグループでは、障害物センサーを付けて机の端に近づくと方向転換をするプログラムを書きます。

ロボット制作の制限時間を経過したら、最後は各グループが制限時間1分で制作したロボットを起動し、どれだけ多くのブロックを落とせたか、重量で測って勝者を決めます。本ワークショップは主に小学生向けに設計されたものですが、大学生であっても熱心に取り組み、時には歓声上がるなど、大変盛り上がります。学生には、本ワークショップを通してプログラミング教育における問題解決活動のあり方に加えて、楽しさも忘れてはならない要素の1つであると指導しています。

問題解決活動を組み込んだ授業設計の理論：未来の学習のための準備 (PFL)

わたしが専門とする学習科学では「未来の学習のための準備 (Preparation for Future Learning: PFL)」と呼ばれる理論があります³⁾。PFLでは、多くの授業で通例である、ある単元について基本的な事項を説明してから演習を行う「説明→演習」の順序ではなく、探索的な問題解決活動を行ってから説明を行う「問題解決活動→説明(→演習)」の順序で授業設計をします。意外に思われる方もいるかと思いますが、従来の説明

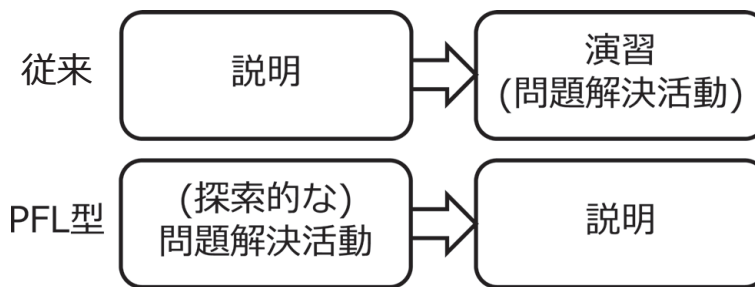
→演習の順序よりも問題解決活動→説明の順序で授業を行う方が、特に応用力が高まる実験結果がいくつも出ています。

説明→演習の順序による授業では、説明を通して学習した内容が他の状況で活用される「学習からの転移（応用）」をねらいますが、PFLでは学習したい内容を必要とする問題解決活動を説明に先行させることで、目的とする「学習への転移」を促そうとします。プログラミングの世界では、1つのプログラミング言語を習得している人は、何も習得していない人と比べると、新たなプログラミング言語を習得しやすい、という話がよく聞かれます。その視点を授業設計に取り入れたのがPFLなのです。

PFL型の授業設計と「生産的な失敗」の重要性

では、どのようにPFL型の授業を設計すればよいのでしょうか。その設計原則として、まず1) 学び手が問題をイメージしやすく（すでに知っている）既有知識を活用しやすい状況を設定する必要があります。前述のワークショップでは「雪かき＝ブロックを机から落とす」というイメージしやすい目標が設定されています。次に、2) これから学習する知識を活用しないと解決できないが既有知識を活用して試行錯誤できる問題の設定が必要です。雪かきワークショップでは、ブロックで形を作ることで工夫はできるものの、順次・分岐・反復処理を組み合わせたプログラミングをしなければより多くのブロックを落とすことはできません。そして、3) グループで協調（協働）し試行錯誤しながら取り組める活動にします。各個人で問題解決に取り組むと、どうしても自身のアイデアに限られてしまいます。2、3名のグループで取り組むことで、自身のアイデアに対するフィードバックをメンバーから得たり、メンバーから自身では気づかなかったアイデアに気づく機会が得られます。

一方で、これから学習する内容に関する説明（知識）が与えられていなければ、問題の解決はうまくいきません。通常、授業は生徒が失敗しないように設計するものですが、説明を与える前の問題解決活動で「生産的な失敗」をさせることで、説明による学習効果を高めることをねらいます³⁾。具体的には、うまくいかないかにかかわらず、できるだけ多くのアイデア（案）を出してもらいます。まず自分でやってみることで主体性を高め、これから学習する内容に関



PFL型授業の流れのイメージ

連する既有知識を活性化するのがねらいです。これは、人は新しい内容について学習をする際、既有知識に関連付けて理解する（学ぶ）という「構成主義」の考え方に基づくものです。また、あれこれ試してみてもうまくできないことで既有知識では足りないギャップ、あるいは新たな内容を学習する必要性に気づくことで、説明を聞くことへの「動機づけ」にもつながるのがポイントです。

探索的な問題解決活動を通して既有知識が活性化され、新たな内容の学習への動機づけがなされた状態で、いよいよ説明を行います。その際に重要となるのは、自身が先行して考えたアイデアと、説明に基づく正しいアイデアを比較しながら「振り返る」ことです。この振り返りによって、自身が考えたアイデアのどこが足りなかったのか、正しいアイデアで行うとどのようなメリットがあるのかを比較しながら考えさせることで、構成主義的な学びを促すとともに、学習の意義や価値に気づくきっかけを作ることができます。

今回の記事では、プログラミング教育における問題解決活動を組み込んだ授業設計というテーマで、問題解決活動の設定や説明を与えるタイミングについてご紹介しました。本記事の内容がプログラミング教育に加え、他の単元の指導でもお役に立てば幸いです。

参考文献

- 1) SONY (2023.08.18) CurioStep 子どもたちの好奇心を育む教育プログラム「キュリオステップ」ワークショップ <https://www.sony.com/ja/SonyInfo/csr/ForTheNextGeneration/curiostep/activity/workshop/022.html>
- 2) SCRATCH <https://scratch.mit.edu/>
- 3) 大島純, 千代西尾祐司(編) 益川弘如, 河崎美保, 山口悦司, 大浦弘樹, 望月俊男, 北澤 武, 大島律子, 河野麻沙美, 大崎理乃 著 (2019) 主体的・対話的で深い学びに導く学習科学ガイドブック 北大路書房