



【プログラミング学習の方向性②

プログラミング的思考は授業にどのように取り入れると良いのか】

「プログラミング的思考」は新しい言葉ですが、その概念は日常生活でも意識できます。語弊を恐れずにいえば「段取り」と考えられるものは、プログラミング的に思考している可能性があります。例えば調理の手順は、より早く、おいしく・・・何を準備して、どのような手順で進めると良いのか、それを論理的に考えます。その際、「塩、コショウで下味をつける」や「火が通ったら」とレシピに書かれることもあるでしょう。例えば、図1のような大まかな処理を考えたら(抽象化)、今度は誰でもやり方が分かるように表現します。すると、下味をつける処理方法や火が通るという状態を定義して、小さな手順に「分解」し、分解した手順の流れを考えること(アルゴリズム)になります。また、それは他にどのようなケースに応用できるかについて、考えることも大切です(一般化)。



図1 抽象度の高い手順の例

しかし、これだけで終わってはいけません。なぜならここには、コンピュータ処理の特徴である、正確性、速度性の概念が足りないからです。時間を掛けて行う人間と、言われた通りに高速に正確に実行するコンピュータとは、当然その特徴を活かす考え方が異なります。

また、手でやるには効率が良くても、コンピュータにやらせるには難しいこともあります。例えば、人が正六角形を書く場合、コンパスで円を書き、半径の長さを取り、円周をその長さで分けていきます。しかし、コンピュータには、一辺を書き、角度を60度変えることを6回繰り返すことを指示すれば、書くことができます(図2)。さらに、コンピュータの場合は、やり直しが簡単なので、試しにやってみて結果を見ながら再調整するという思考方法も成立します。こうした思考様式は、コンピュータに処理をさせることで意識化できる独特な見方・考え方です。



図2 コンピュータで処理しやすい正六角形の書き方の例

先生たちは指導する單元の中にある、手順や段取りをプログラミング的に考えてみましょう。そうすることで、今までとは違ったものの見方で、教科目標を理解できるかもしれません。恐らくそれは、いわゆる深い学びの1つとなることでしょう。

宮城教育大学准教授 安藤明伸

