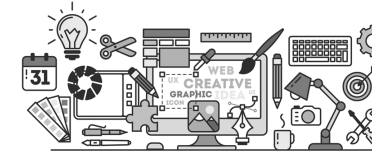


学びが深まる プログラミング教育入門①



プログラミング教育の 意図するもの

赤堀 侃司 ICT CONNECT 21会長/
東京工業大学名誉教授

【連載監修】ICT CONNECT 21事務局

教育の情報化の進展に寄与し、社会の発展に貢献することが目的の団体。さまざまな属性の人が集まるオープンな場の提供や、生涯を通じて学べる学習環境づくりに取り組んでいる。

なぜ、プログラミング教育なのか

2020年度からの小学校新学習指導要領の全面実施に伴って、プログラミング教育についての不安の声が、教育関係者のなかから浮上してきた。そもそも、なぜ、プログラミング教育なのか理解されないのに、どうして実施できるのか、という声も大きい。その背景について述べるには、紙幅が決定的に不足しているので、詳細は、恐縮だが参考文献の拙著をご覧ください。ここでは、その入り口だけを述べよう。

子どもからお年寄りまで持っているスマホは、当然ながらコンピュータであり、電気炊飯器、電気洗濯機などの家電製品のなかには、コンピュータが組み込まれている。自動車は、かつては機械工学の技術であったが、今日では、電子情報技術と言われる。駅の自動改札、スーパーマーケットのレジなど、あらゆる分野に、コンピュータが入ってきた。まるで、現代の巨人とでも呼びたい道具・技術、というより、文明とでも呼びたいような、巨大な存在になった。この世界のすべてを制御していると言っても、過言ではない。

その元、つまり、動かしているエンジン

は、プログラムである。プログラムは人間がつくったものなので、人間がコンピュータを制御しているが、近年になって、コンピュータ自身が学習し始めた。それが、人工知能(AI)である。ここに至って、人とコンピュータのかかわり方について、振り返って考えてみるようになった。

これまでは、コンピュータは道具であった。今でも道具であるが、微妙にニュアンスが違って来た。たとえばよくないが、かつて白人が黒人の奴隷を使ったことがあった。それは、奴隷を牛や馬や荷車と同じように、道具として認識していたからである。もし奴隷が知識を持ち、その知識が、主人を超えたらどうであろうか。さらに、奴隷が自由に学習し始めて、将来は、主人よりもはるかに知能の高い人間になったら、もはや主人という椅子が危なくなるかもしれない、という危機感を持つだろう。

奴隷をAIやロボットに置き換えると、現代社会の実相を反映している。道具をうまく使って、人間の仕事に役立てる、つまり教育で言えば、ICTの利活用の考え方である。

しかし、道具自身が、人間と同じように学習し始めたら、どうであろうか。道具を使うという主人の立場から、一度振り返って、こ

の道具はどういう仕組みなのか、と相手を知るといふ立場になるであろう。

イギリスの教育研究機関が、人は、コンピュータの利用者(ユーザー)から著者(オーナー)になるべきだ、と提言して、道具としてのICTの教科を、Computingの教科に変えた。それは、読み書き算に続く、四番目のリテラシーとして提唱されたComputational Thinkingの考え方が、影響を与えたからであった。イギリスのカリキュラムが、世界にプログラミング教育のきっかけをつくった、と言ってもよいだろう。

なぜ、プログラミング的思考が必要なのか

先の英語を、日本の文部科学省は、プログラミング的思考と訳した。この用語が、教育関係者にも、理解されていないようだ。それは、プログラミングをすること、つまりプログラム言語で、フローチャートに沿って書くことと混同しているからである。

音楽は、音符という記号で曲を表し、ダンスは、体の動きで自分を表現し、短歌や俳句は、日本語という記号で表現し、料理は、食材を使って目的の品をつくり、建築家は、設計図を使って、目的の建築物を表している。

このように、何か目的とするもの——これが自分が意図するもの、と呼んでいるが——その意図を記号で表現して、意図に合うように、組み替えながら改善する論理的な思考力だと、プログラミング的思考を定義している。このように、意図を表す方法や記号は多様であって、プログラム言語は、その記号の一部なのである。

どのような意味があるのか

多くの意味があるが、ここでは、一つだけ紹介したい。私は、総務省のプログラミング教育の実証事業にかかわってきたが、どのプロジェクトでも、どの子どもも、プログラミングの体験を通して、「むずかしいが、おもしろい、楽しい」と述べている。むずかしい、という言葉に注目してほしい。簡単だったから、先生にほめられたから、ほうびをもらえたから、おもしろい、楽しい、と言っているのではない。

プログラミングの経験をすれば、すぐわかるように、一度にすぐプログラムが成功することは、ほとんどない。プログラミング(コーディング)すると、ほとんどがエラーとなっていて、返ってくる。返ってくるという表現は、人とコンピュータが対話しているイメージを

表している。このとき、相手が人の場合では、怒る場面であっても、相手がコンピュータの場合は、必ず、こちらがミスしていると思うのである。実際にそのとおりで、そこで気がついて、修正をする(デバッグ)ことになる。

国語や算数などのテストで正解なら喜び、誤答ならがっかりすることが通例であるが、プログラミングの場合は、誤答のほうが、普通である。だから、むずかしい。しかし、そこを乗り越えるときに、このうえなく嬉しい気持ちになる。そうか、そうだったのか、という自分の理解が深まったという気持ちがおもしろい、楽しい、という表現になった。

つまり、失敗におもしろさを感じたのである。それは、発見と言ってもよい。考えてみれば、失敗もせず、無難に、成功ばかりの人生はなく、失敗しながら、しかし、なんとか修正しながら、デバッグしながら、人は生きてきた。そこに、おもしろさを感じることは、プログラミング教育の優れた特性ではないか。それは、社会に出ても、必要な経験になるだろう。社会では、試行錯誤で、失敗を繰り返しながら、仕事をやるからである。

〈参考文献〉赤堀侃司『プログラミング教育の考え方とすぐに使える教材集』ジャムハウス、2018年。