

[インタビュー]

小学生のプログラミング教育を考える

サイエンス倶楽部
専務取締役 広永 雅史

なぜ小学生のためのプログラミング教育が必要と、お考えになりましたか？

加速度的な社会の変化にそれまでITは、理系のごく一部、IT企業に携わる特殊技能を持った専門職の人たちの感覚があり、おもに文系出身者がつく生業にはむしろ関係ない世界。専門職が担うごく一部の特殊技術という印象でしたが、みるみる日常生活の中に浸透し、次々と便利な生活を実現してきています。

しかし、ITを組み込まれた機器を扱う人間の方がついていけない。むしろ機器に人間が操られていると、感じる場合があります。

小学生に目を向けても、デジタル世代だけあって、アナログなものへの関心度が低く、プロセスに対してめんどろくさがる傾向が感じられる。広く捉えると、そもそもモノ全般に無関心なのです。

「なぜ？」がなく、「なぜ、そうなるんだ？」に何らかの答えがえられません。プロセスに興味を持っていないからです。

我々としては、「それではいけない。そんな小学生が社会人になったら・・・」と、社内で話し合い危機感を共有してきました。

そうした課題認識から、我々が25年かけて培ってきた自然科学や理科の教育とITの接点を持つために、プログラミング教育を実施することを決断しました。

その際に、「パソコンは道具であり、プログラムは言語なのだから、どう使うかが大切だ」という我々の教育の基本理念が形成されていきました。

いわばプログラミング「を」学ぶのではなく、プログラミング「で」学ぶことが大切だ、ということになったのです。

文科省も、2020年には、小学生にプログラミング教育をスタートさせようとしています。加えてSEやプログラマーの不足が予想されることから、プログラマーを育てる

ことが、日本の社会において、必須の課題となりつつあります。

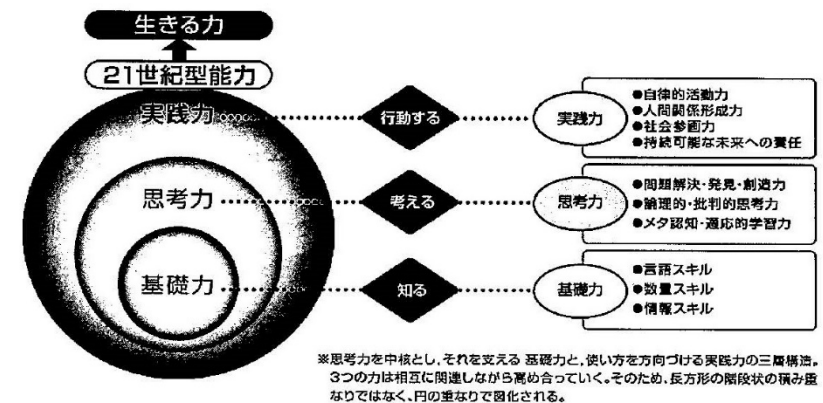
とはいえ我々のプログラミング教育は、プログラミングができる人材の裾野を広げることを目的としているわけではありません。

我々の真の目的は、プログラミングを通じて、創造力や論理的思考力、問題解決能力など、実社会で求められる能力・資質を育むことなのです。

我々サイエンス倶楽部のスタートの発端は、「子供達の原体験が大切」であり、そこから教育することでした。そのためサイエンス倶楽部としての実学教育は、ロボットやゲームやアプリもありますが、制御系をきっちり学ばせる方が良いということになったのです。

その結果、「ものづくり×プログラミング」教室とし、実在する機器を制御するためのプログラミングを行ないます。そのため、機器の仕組みだけでなく、センサーやギアなど、機器を動かすために必要な知識についても体験を通して学んでいきます。

基礎力・思考力・実践力の3層構造から21世紀型能力を導き出し、それを生きる力にする、という下図にあるような素晴らしいコンセプトを、どのように創りだし、実践されていますか？



〔基礎力〕

理科・実験教育は、知識や技術を習得するための「基礎力」が大切です。そのことを、否定すべきではありません。

プログラミング教育においても、「基礎力」ということで、C言語をきっちり習得させるようにしています。加えてマイコン、回路、周辺機器、センサーの基礎知識や扱うス

キルは、応用実践していくために絶対に必要となるため、教育しています。

〔思考力〕

「思考力」については、〔仮説〕→〔方法〕→〔実験〕→〔結果〕→〔考察〕の思考プロセスが大切です。

あらかじめ教えずに、「まず君の考え方は？」と質問し、「予想したものを書いてみよう」と促し、実際に試させて結果を考察させます。

こうした〔仮説〕から〔考察〕に至るプロセスが、大切です。

そのプロセスは、プログラミングのフローチャートやアルゴリズム作成と同じです。考える道筋がしっかりしていること、そのことが大切なのです。

その意味では、サイエンス倶楽部でやってきた思考プロセスの醸成による自分から考える学習を、プログラミング教育でも実践しているのです。ITの仕事をしている多くの保護者の方々からは、

「この教育実践により獲得される課題解決能力こそが、これからの人材に必要とされており、是非そうした教育をしてほしい」と、言われています。

〔実践力〕

「実践力」については、生徒1人ひとりが自ら考える自律的活動力を育成し、どう生徒と生徒をつないで人間関係形成力を創っていくかが、必要となります。

教師は、大卒のサイエンス倶楽部卒業生が担当しており、1クラスは10人の少人数で教えています。

グループワークを重視しており、各生徒の学習のスピードに差異はありますが、最終ゴールまで全生徒が到達できるように指導しています。

どう人と人をつなぎ、モノを創りだしていくことができるのか、そうした力の向上を、PDCAを回すことで確認していく必要があると思います。

そのためにグループワークで、プログラミング教育をしています。公式があって、教えられるほどつまらないことはありません。

したがって、最初は教えません。

現象が先あって「ああでもない。こうでもない。」と試行錯誤しながら、チャレンジすることで楽しくなります。

さらにはクラスのほかの生徒の意見を聞きながら育っていくのです。

このようにグループワークを推進することで、生徒の理解速度の違いがあっても、結果的に到達地点には、1人の生徒も落伍することなく、全員ゴールインできることとなるのです。

以上のように、「基礎力」「思考力」「実践力」の3層構造によるプログラミング教育は、サイエンス倶楽部の理科の実験教育をアレンジして実施しています。

そして3層構造のプログラミング教育を積み重ねることで、次代を担う小学生たちの21世紀型能力が育成されていくのだと展望しています。

小学生低学年と高学年では、プログラミング能力に、どの程度の差異があるのでしょうか？

問題意識や課題認識が大切であり、プログラムの書き方にのみ、終始すべきではないと判断しています。

モノを深く幅広く捕らえる原体験の違いが、問題意識や課題認識の差があるかないかにつながり、技術やスキルに反映されると考えます。

そのため小学校低学年でも、プログラミングに強い生徒はおりますので学年にこだわり、その枠に縛られるべきではありません。

できる生徒を、どれだけ伸ばせるかが課題であり、その先鞭をつけるのが、我々民間の教室ではないかと判断しています。

小さいころからより多くの質の良い体験を積み重ねることが、何よりも大切です。敢えて言えば、今の学生の多くに、その点が欠けているように思えます。

専門知識や専門スキルの習得だけに特化した教育からは、斬新な発想力は生みだせません。

その結果、言われたことをやるだけのエンジニアしか育ちません。



教育受講後に、生徒たちは、どのようにレベルアップしているのですか？

ものに対する視点が、根本的に変わっていきます。

今まではプログラムが組み込まれ実装された状態にのみ、興味や関心をしめしていた子どもたちが、以後そのものが表現するプロセスに興味をもち、別の表現方法を求める欲求が出てきます。

そうしたプロセスを通じて、単にパソコンにプログラムを打ち込むのではなく、自分はどういうことをしたいのか、課題を明確にして、モノにプログラムをはしらせて、いわば Tinkering (ティンカリング) の限界を突破していきます。

【ものづくり】と【プログラミング】は、片方では駄目で、2つを両輪にして「プロテック・サイクル」として回してしていくことが、^{*}知^{*}の醸成をもたらすことになります。

今後は、以上の成果を踏まえて、IoTの領域を取り入れたカリキュラムを作っていく予定であります。

小学生にとって、授業時間 2 時間は、長くはありませんか？ どのような工夫をされていますか？

2 時間も続けて授業ができるのか、信じない人もいます。

確かに小学生を前に講義をしだしたら 10 分と持たないかもしれません。

しかし生徒たちが、2 時間でも長くないと感じているのは、自ら主体的に授業に取り組めるような参加型の授業を行っているからです。

授業の手順はありますが、なるべく「教えない教育」をしています。

生徒たちは、まわりの生徒たちとコミュニケーションを取りながら、学びを体感していきます。

小学生は、学生と違って、理屈に縛られない、抜群な柔軟性を持ち合わせています。いったん興味関心のエンジンがかかるとそのマシンのポテンシャルは計り知れません。

少人数制にして、1 人ひとりの個性を意識し、多様性を活かしたチームで授業して行くことがこれからの教育に必要なのでしょうか。

サイエンス倶楽部の教育が、社会人になった時、どう活かされるのか、教えてください。

さらに、より大局的観点から、「ものづくり × プログラミング教室」が、日本企業のイノベーション能力、競争力強化にどのように貢献しうるのか、近未来の可能性を含め、お話しください。

中学生になると大学のゼミにお邪魔して、学生さんと一緒に実験を学ばせてもらうプログラムがあります。その場でよく、指導教授から、「この生徒たちは科学に対する姿勢ができています」「自ら乗り越え、自分から切り拓いていく力がある」と評価してくれています。

幼児、小学生のときから、モノゴトに興味関心を持つことが大事です。

無気力、無関心からは、何も生まれません。

生徒たちは、授業の中で、普通に自分の考え、予想をアウトプットすることをコミュニケーションのようにしています。つまり、仮説→目的・方法→結果→考察という思考プロセスを自然に体得しています。

こうした基礎力が備わることで、モノを作り出す、考え出す力を持った社会人に成長していきます。実際に OB、OG からも見取れます。

テクニカルなプログラミングだけの習得よりも、ロジカルシンキングや問題発見型の人材が育成され、イノベーション能力の強化につながります。

さらにビジョン形成力やリーダーシップを持った人材が輩出されるようになり、日本企業の競争力強化に貢献するようになるでしょう。

そして、そうした多様な人材の中から、ビル・ゲイツやスティーブ・ジョブスのような若き起業家も台頭してくるのではないのでしょうか。



以上