

Z会プログラミング講座

メンター用補足資料



●「プログラミング」とは ー児童用テキストの補足ー

■プログラミングって何だ？

一般的に「プログラミング」という言葉から想起されるのは、「BASIC」「C言語」「Java」といったいわゆる「テキスト型のプログラミング言語」を用いて、実際にプログラムを書いていくことでしょう。もちろんその認識は正しく、多くの場合そのような意味合いで使われますが、ここではもう少し広い意味で捉えています。

今、世界中には何百万という「アプリ」が存在します（正確な統計データはありませんが、App Storeでも Google Play でも登録数が200万を超えているという情報があります）。その中には一人の手で作られたものもあれば、企業でチームが作ったものもあります。しかしそのいずれにも共通しているのは、「作りたいイメージのものが先にあり、その完成形を目指して作られた」ということではないでしょうか。チームでアプリを作る際には（そしてアプリに限らずチームで何かを作成するときにはこのような手順をたどるのでしょうが）、必要な機能などを確認し、それを元におおまかな設計を考え、さらにそれらを細分化して詳細を考え、やっと実際にプログラムを書くことができます。個人でアプリを作る際にも、意識するかしないかは別として、似たような工程をたどっていることでしょう。

実際にプログラムを書くことは当然大事なことでありますが、それが全てではありません。時代が変われば使われるプログラミング言語も変わり、10年前の先端技術は時代遅れの遺物にもなりかねません（事実、iPhoneアプリを開発するのに使う言語は、2010年代になって登場したものです）。しかしそこに至るまでの過程や、プログラム上の誤り（バグ）を取り除き完成形に近づけていく過程は大きく変わりません。もちろんここでも、時代によって使うツールなどは変わでしょう。しかし基本的な考え方は変わらないはずです。ここでは、このような一連の流れをまとめて「プログラミング」と呼んでいます（プログラムを書くことはコーディングと呼びます）。

■プログラミング的思考

「学校教育にプログラミング」という文脈では、**プログラミング的思考**という言葉が多く登場します。この言葉が一人歩きしていることも否定できませんが、次期学習指導要領についての中教審の答申では、「プログラミング的思考」を次のように定義しています。

自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力のこと。

幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について（答申）（中教審第197号）

「記号の組み合わせ」に限定しているように感じられるのが気になりますが、先に述べた「**作りたいイメージのものが先にあり、その完成形を目指して作る**」ための工程を抽象化したのがこの「プログラミング的思考」です。何もプログラミング的思考は「コンピュータ」だけのものではなく、われわれの身の回りに「プログラミング的思考」が必要となる場面は多数あることになります。

●身近な「プログラミング的思考」の例

日曜日は意外と忙しい。1週間分の食料品も買わなければならないし、最近子供が生まれた友人への出産祝いも買って届けたい。それだけじゃなくて子供をどこかに連れて行ってあげたい。さあ、どうする？



やりたいことが多いから計画を立てよう。まず、土曜日の晩までに1週間分のおおまかな献立を立てて、何を買うか考えておこう。そうだ、せっかくだから話題のレゴランドに行ってみようか。子供も楽しめるだろうし、出産祝いもそこで買えばいいじゃない。珍しいものだから喜ばれるよね。車でいくと混むというけれども、時間を選べば意外となんとかなるかも。帰りに友達のところに寄ったあと、近所のスーパーで買い物をして帰ってあげればいいよね。みんな疲れているから、晩ご飯はできあいのお惣菜でもいいかな。よし、じゃあ、渋滞状況を調べて出発時間と帰る時間を決めよう。……

多少こじつけの感があるものの、「意図する一連の活動を実現」するために、どのような動きを組み合わせれば最大限の効果が得られるかを考える、ということは日常生活で行っていることです。考えてその通りに行動して、必要があれば修正をし、最後に振り返りを行うということも、場合によってはセットで行っているでしょう。**プログラミング的思考は特別なものではなく、身近にあるものです。そして、意識して行うことで、精度を上げていけるはずのものです。**

もちろん、こうした思考を養うために「プログラミング」が絶対に必要なものではありません。しかし、プログラミングを通すことで効率よく身につけられると考えられているものです。ですから「プログラミング教育の必修化」というときに、「プログラムを書く」ことだけを取り入れるのには意味がありません。「プログラミング的思考」を身につけさせるためのツールのひとつとして「プログラムを書く」ことがあり、そこで興味を持つ人がいれば、さらに深く勉強してプログラマーなど専門的な職業を目指すという道があります。一方で、社会に出てもプログラミングの知識が必要とされる職場ばかりではありません。誰もがプロ級になる必要はないのです。しかし英語と同じく、様々な仕事に深く関わってくることであり、誰にも最低限のスキルが必要となってくることでしょう。その意味でも、学校教育にプログラミングが入ってくることは避けられない流れであると考えています。

■ビジュアル型プログラミング言語

ご存じの方も多いかと思いますが、最後に「ビジュアル型プログラミング言語」についても触れておきます。ビジュアル型の場合、キーボードの操作をしなくともよい、命令を覚えなくともよい、純粋にプログラミングに必要な思考が学べるといったメリットのある一方、テキスト型への移行（つまり、実用的な言語への移行）が難しいというデメリットもあります。しかし、児童の発達段階を考えれば、最初からテキスト型の言語を教えるメリットはかなり薄くなるのも事実です（嫌いにさせてしまうことが一番怖い）。そのため、まずはビジュアル型で慣れ、もっと学びたいという児童にはテキスト型を勧めるという流れが自然でしょう。また、一部のビジュアル言語では、テキスト型言語の入門としてよく使われる JavaScript のコードを出力する機能を持ったものがあります。興味のある方は「ビジュアル言語 JavaScript」といったキーワードでいくつかの言語がヒットしますので、調べてみてください。

●本講座のスタンス

本講座はコーディングの講座ではありません。プログラミングの講座です。そして、放課後や長期休業中、土曜日など学校の授業外で行うものです。そこで指導に当たっていただくメンターの皆さんは、必ずしも「コーディング」に詳しい方ばかりではないという想定でいます。

従って本講座では、「教える」という場面は極力減らしたいと考えています。協働学習、アクティブ・ラーニングといった「子供たちが主体的に学ぶ」「知恵を出し合って学ぶ」学び方を取り入れ、メンターには「子供たちの議論を活発にすること」を目的に講座に参加していただきたいと考えています。この点については、後ほど触れていきます。

メンターの皆様に期待するのは、人生経験を通して身につけてきた「プログラミング的思考」です。それぞれの切り口から「プログラミング的思考」を子供たちに伝えてください。それぞれのバックグラウンドで培ってきたスキルを生かしていただき、子供たちに最高の経験を提供してください。

●マインドストームを組み立てようー児童用テキストの補足ー

レゴ®マインドストーム®は、レゴ社が販売する教育用ロボットです。多くの国で使われ、世界大会も行われています。日本国内でもさまざまな事業者が工夫を凝らした教材を作成しており、マインドストームを用いたロボット教室も数多く開かれています。本講座にも、そうした教材や教室ですでにマインドストームの学習をしている児童が参加するかもしれません。

先にも触れた通り、本講座は「プログラミング的思考」を養うことが主の目的であり、マインドストームの制御を極めることは考えていません。ここでメンターの皆様に身につけていただきたいのは、マインドストームの基本的な制御の仕方です。子供たちのほうが詳しい場面も考えられますが、そのときには**子供たちが先生**です。「講師」ではなく「メンター」という言い方をしている背景には、こうした思いもあります。

■組み立てについて

組み立てには時間がかかります。慣れた大人でも、基本的な形を組み立てるのに 20 分はかかります。子供たちに 30 分で組み立てさせるのは、酷なようにも思います。

しかし、工夫の余地はあります。組み立てガイドを見ると、左右対称になっている部分があります。そのような場面は一緒に作ると時間の短縮になります。あるいは、全体像を把握しておく必要があるものの、子供たちに分担させることもできるでしょう。

《考えてみましょう》子供たちを班の中でさらに2組に分けるものとして、組み立てガイドをどのように分担させれば速く組み立てが終わるでしょうか？

●簡単なプログラムを作ってみようー児童用テキストの補足ー

先にも触れた通り、「プログラム」といっても、命令文をコンピュータに打ち込んでいくというイメージのものではありません。タブレット上で部品をつなげ、ロボットの動きを制御するための一連の命令を組み立てます。今回使用するのはタブレット版アプリで、機能に制限がある代わりに、わかりやすいのが特徴的です。

この講座では全てのプログラミングブロックは使いませんが、以下のブロックについて、理解しておいてください。児童用テキストには次の内容を全て書かせる必要はありません。児童が体験の中から次のようなことを読み取り、書くことができれば十分です。

なお、大まかに次のルールがあることをご承知ください。

緑色のブロック（緑色タブの中にある）→モーターを動かすなど、EV3 そのものの動きを制御

黄色のブロック（黄色タブの中にある）→ループなど、プログラムの流れを制御

紫色のタブ内のブロックはコメント用のものです。

また、キャンバスに配置されたプログラミングブロックのアイコン部分をタップすると、他の機能を選ぶことができます。例えば、L モーターブロックのアイコン部分をタップすると、「オフ」「オン」「秒数」「角度」「回転数」という選択肢が表示されます。これは、モーターをどのようなオプションで制御するのかの選択肢です。それぞれのプログラミングブロックで、ほかにどのような機能があるのかを確認しておいてください。

	名前：Mモーターブロック Mモーターを動かす（今回はMモーターを使わない想定）。Mモーターはキット内に1つしか含まれないため、同時に1つのMモーターブロックしか使わない（追加でMモーターを購入すれば増やすことができる）
	名前：L モーターブロック L モーターを動かす。複数のL モーターを同時に制御することはできないため、「まっすぐに動かす」といった制御をする場合には、ステアリングブロックなどを用いるか、「ワイヤー接続」をして複数のL モーターブロックを同時に動かす必要がある。
	名前：ステアリングブロック 2つのL モーターを同時に動かす。ステアリングの名前の通り、進行方向を指示することで左右のモーター出力を制御してくれる。
	名前： タンクブロック 2つのL モーターを同時に動かす。ただしステアリングブロックと違い、2つのL モーターブロックの出力をそれぞれ手動で制御する。今回は使わない想定。

※なお、出力や進行方向はマイナスの指定もできる。マイナスの場合は「反対方向」へ動くことになる。

	名前：表示ブロック EV3 の液晶画面に文字や絵を表示させる。目の形やハートの形などが最初から用意されており、表示させるものだけではなく、表示させる位置も指示することができる。
	名前：音ブロック 音声を再生させる。組み込まれているもの、自分で録音したものを再生できる。組み込まれているものは、カテゴリは日本語になっているものの、ファイル名は英語のため、フォローが必要。
	名前：インテリジェントブロック EV3 のランプを光らせるブロック。点灯、点滅もオプションで選ぶことができる。
	名前：待機^{たいき}ブロック 指示された状態になるまで、次の命令を実行せずに待つ「〇秒待つ」という形のほか、センサーの入力を待って次に進むという使い方もできる。
	名前：ループブロック このブロック内にある命令をくり返す。くり返す回数は、無限、回数指定、センサー入力から選ぶことができる。どの場合も「指示された状態になるまでループさせる」。
	名前：スイッチブロック 指定された条件を満たすときと満たさないときで、それぞれ別の動きをさせるためのブロック。今回は使わない想定。

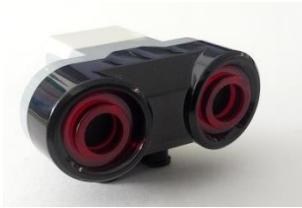
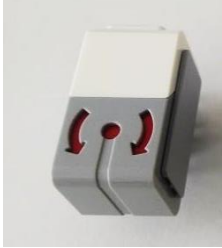
なお、児童用テキストにある練習問題は、次のようなプログラムで実現できます。



ループを使わなくてもよいですし、吠えるのと前後への移動が同時になるよう工夫してもよいでしょう。大事なのは、必要なプログラミングブロックが何であるのかを考え、組み合わせ、自分たちなりの答えを作ることができるかどうか、という点です。

●センサーを使ってみようー児童用テキストの補足ー

マインドストームの標準キットには、4種類のセンサーが含まれています（うち、タッチセンサーは2つ同梱）。主な機能と使い方の例を紹介します。こちらも、児童用テキストに以下の全てを書かせる必要はありません。

	名前：超音波センサー 前方にある障害物との距離を測る。向かい合ったときの左側から超音波が出て、右側で受信する。可測領域は直線状ではなく、同心円状に広がっている。そのため、EV3 の進行方向から少しずれた場所にある物体を感知し、そこまでの距離を出してしまうこともある。「衝突寸前で止まる」「物体との距離が変化したらアクションを起こす」などに使う。
	名前：カラーセンサー 約 2cm 以内の色や反射光の強さなどを感知する。検知できる距離は短いため、読ませたい色のものをほぼ接触させることになる。センサー部分を下に向けて「ラインレース」に使ったり、前に向けて「読ませる色によって違う動作をさせる」という場面で使ったりされる。また、反射光を検知させることで「机から落ちそうになったら止まる」というプログラムに使われたりもする。
	名前：タッチセンサー スイッチ部分が「押されている」「押されていない」「押されて離れた（バンプ）」という3つの状態を検知するセンサー。衝突を検知することが多いが、「押されたら動き始める」「衝突したら止まる」というスタート・ストップスイッチのような役割をすることもある。2つ同梱されているため、前後に取り付けることも多い。
	名前：ジャイロセンサー 回転を検知するセンサー。扱い方が難しく、使う場面は限られる。数値が安定しないときには、一度ケーブルを取り外して制止した状態でつけ直すといよい。また、激しく動かすと急に値がおかしくなることもある。

●プログラムを作ってみようー児童用テキストの補足ー

児童用テキストP5にあるプログラムや、ジグソー活動で配布したプログラムを参考にしてください。基本的な組み合わせだけで作成できるプログラムです。

●指導にあたって

通常学校で教えられる指導内容であれば、児童の方が指導者よりも詳しいというケースはあまり多くありません。しかし情報機器の操作、「プログラミング」等に関しては、児童のほうが圧倒的に詳しいというケースは珍しくないのが現実です。それに対応するためには、考え方は2つです。

- 圧倒的な専門性を身につけた者が指導者となる
- 指導者の役割を「教える」ではなく、「引き出す」「広げる」ことにおく

従来の「指導者」のイメージは前者でした。しかし現実問題として、特に本講座では「地域人材に指導者となってもらう」ことが求められているとなると、本講座の指導者は後者であるべきです。

そして昨今、学校の教育現場でも「アクティブ・ラーニング」という言葉が聞かれるようになりました。文科省はこの言葉を次のように説明しています。

教員による一方向的な講義形式の教育とは異なり、学修者の能動的な学修への参加を取り入れた教授・学習法の総称。学修者が能動的に学修することによって、認知的、倫理的、社会的能力、教養、知識、経験を含めた汎用的能力の育成を図る。発見学習、問題解決学習、体験学習、調査学習等が含まれるが、教室内でのグループ・ディスカッション、ディベート、グループ・ワーク等も有効なアクティブ・ラーニングの方法である。

平成 24 年 8 月 28 日 中央教育審議会「新たな未来を築くための大学教育の質的転換に向けて～生涯学び続け、主体的に考える力を育成する大学へ～（答申）」

「教える」ではなく「引き出す」「広げる」ことが目的の本講座に求められるものは、アクティブ・ラーニングであると言えます。

■ジグソー法

アクティブ・ラーニングに確立した「方法」はありません。今なお多くの先生方が試行錯誤をくり返し、どのように指導することが児童・生徒のためになるのかが日々研究されています。

本講座では、大学発教育支援コンソーシアム推進機構（CoREF）が提唱する「知識構成型ジグソー法」を取り入れたアクティブ・ラーニングを実践します。「知識構成型ジグソー法」とは次のようなものです。なお、問いや資料は教材として作成済みです。

STEP.0 問いを設定する

まず先生は、単元での「問い（課題）」を設定します。この時、既に知っていることや、3つか4つの知識を部品として組み合わせることで解けるものになるように設定し、その問いを解くのに必要な資料を、知識のパートごとに準備します。

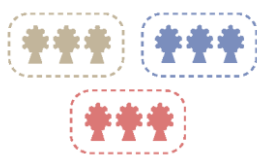
STEP.1 自分のわかっていることを意識化する

「問い」を受け取ったら、はじめに一人で今思いつく答えを書いておきます。



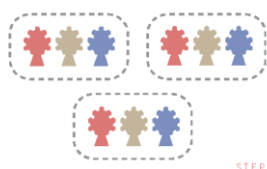
STEP.1

STEP.2 エキスパート活動で専門家になる



同じ資料を読み合うグループを作り、その資料に書かれた内容や意味を話し合い、グループで理解を深めます。この活動をエキスパート活動と呼びます。担当する資料にちょっと詳しくなります。

STEP.3 ジグソー活動で交換・統合する



次に、違う資料を読んだ人が一人ずついる新しいグループに組み替え、さきほどのエキスパート活動でわかってきた内容を説明し合います。このグループでは、元の資料を知っているのは自分一人なので、自分の言葉で自分の考えが伝わるように説明することになります。この活動が、自分の理解状況を内省したり、新たな疑問を持つ活動につながります。同時に他のメンバーから他の資料についての説明を聞き、自分が担当した資料との関連を考える中で、理解を深めていきます。理解が深まったところで、それぞれのパートの知識を組み合わせ、問いへの答えを作ります。

STEP.4 クロストークで発表し、表現をみつける



答えが出たら、その根拠も合わせてクラスで発表します。他者の意見に耳を傾けて、自分たちも全体への発表という形で表現をし直します。各グループから出てくる答えは同じでも根拠の説明は少しずつ違うでしょう。互いの答えと根拠を検討し、その違いを通して、一人ひとりが自分なりのまとめ方を吟味するチャンスが得られ、一人ひとりが納得する過程が生まれます。

STEP.5 一人に戻る



はじめに立てられた問いに再び向き合い、最後は一人で問いに対する答えを記述してみます。

東京大学 CoREF「知識構成型ジグソー法」(<http://coref.u-tokyo.ac.jp/archives/5515>)

時間的制約もあるため、すべてのSTEPが実施できないかもしれません(特にSTEP.4のクロストーク)。いずれのパートでも、メンターは「教える」よりは「見守る」ことを徹底してください。エキスパート活動の際でも、基本的な事実だけは伝えますが、児童どうしの対話を促してください。

■傾聴

本講座では、「ティーチング」よりも「コーチング」的な発想が必要とされます。ティーチングが「教え込む」のに対し、コーチングは「伴走する」イメージでよいでしょう。また、コーチングでは「双方向コミュニケーション」が求められますが、ここではメンターの考えを伝えることは極力控えてください。児童が何を考えているのか、何を伝えようとしているのか、徹底して傾聴してください。その上でさらに深く問うてみたり、他の児童にも意見を求めたりといった活動をお願いいたします。

児童への指導にあたり、次のような観点でお声がけをお願いします。

OK	NG
•「なんでできたんだろう？」 •「こんなふうになっちゃったね？」 (うまくいかなかった事実の確認) •「教えて！」 •いったん受け入れる •見守る •ヒントを出す	•「なんでできなかったんだろう？」 •「わかる？」 •否定する •教えようとする •答えを言う

完全にゼロにするのは難しいとしても、「教えない」と考えておいたほうがよいかもしれません。

【メモ】

●「防災」についてー児童用テキストの補足ー

東日本大震災以降、「防災」は我が国において重大なテーマのひとつとなっています。小学校でも3・4年生の社会科で「災害」について学ぶほか、5年生の理科では「台風」について、6年生の理科では「地震」について学びます。知識を獲得することは大事ながら、学んだことが現実の世界とつながっていることを実感することも欠かせません。そのために学校では「総合的な学習の時間」が設定されたり、先生方は指導の際に「現実の社会との関わり」を意識させるような工夫をされたりしています。本講座でも、「防災」という題材を「プログラミング的思考」という観点で取り扱っていきます。

■防災という言葉

「防災」を字義通り捉えれば「災害を防ぐこと」となります。しかし現実の災害では必ず「想定外」が起きます。どれだけ準備をしても、被害はゼロにできません。本講座ではその後の「二次被害を防ぐ」という点や「復興」という点も含めて「防災」という言葉を使います。

児童用テキストに引用した『静岡県第4次地震被害想定』にも、次のような言葉があります。

(この調査は)最悪の事態を想定した結果であり、本想定でも示したように、事前に予防的な対策を講じることにより被害を大幅に減らすことができることにも注目していただきたい。

日頃からの備えが大事であり、そのためには感心を持つこと、正確な知識を持つこと、予防策を実際に講じることが必要となってきます。本講座では「起きてしまったらどうするか」という観点で考えることとなりますが、児童にはここから感心を持ち、正確な知識を持ち、予防策を考え実行できるようになっていてもらいたいと考えています。