WeDo 2.0のプロジェクト学習でプログラミング的思考を身につける

この章では、科学的な設定の中でWeDo 2.0を使いながらプログラミング的思考スキルを身につける方法を説明します。

教育版レゴ® WeDo 2.0のプロジェクト学習でプログラミング的思考を発達させる

教育版レゴ® が自信を持って提供するこのプロジェクトは、小学校の授業で使用するために特別にデザインされており、子どもたちがプログラミング的思考力を身につけるのに役立ちます。

プログラミング的思考とは、日常的問題を解決するのに通常の学習で使える一連のスキルです。WeDo 2.0では、プロジェクトの各ステージを通してこうしたスキルが身につきます。各プロジェクトでは、教師のためにスキル向上ポイントを示していますが、自分や子どもたちにとって最も関連するプロジェクトに絞り込んでもよいでしょう。

WeDo 2.0のどのプロジェクトも、レゴのブロックを使ったアクティビティをアイコン式プログラミング言語と結びつけて、プログラミングの基本ルールを学びながら、問題の解決方法を見つけられるように工夫されています。

WeDo 2.0でコーディング・アクティビティを行いながらプログラミング的思考を発達させることで、子どもたちは自分の作品が動き出すのを見て笑顔になり、さらなる探求に意欲を燃やします。

© 2017 The LEGO Group.

コンピューターサイエンス、プログラミング的思考、コーディング

人類の創成期からある科学や技術という学習分野と比べると、コンピューターサイエンスは歴史こそはるかに浅いものの、科学や技術へのアプローチ方だけでなく、暮らし方にも影響を及ぼします。

コンピューターサイエンスは、科学（Science）、技術（Technology）、工学（Engineering）、数学（Mathematics）と特質を同じくする「STEM」学問分野です。

STEMのどの学問分野も物の見方と生涯役に立つ手法を発達させる機会となります。質問したり、解決法を練ったり、結果を伝えたりする能力はこうした手法の例です。

プログラミング的思考もこの手法に属し、問題解決に使える一般的な思考方法のひとつです。

プログラミング的思考はスキルの集まりとして説明することもでき、そのひとつが論理的思考です。論理的思考を考え出す作業を記述する際、「コード」や「コーディング」を使うことができます。

つまり、STEMの枠内でコーディングとは、プログラミング的思考を発達させるための一手段ととらえることができます。

STEM学問分野

科学（理科）、技術、工学、数学（算数）、
コンピューターサイエンス

物の見方と生涯役に立つ手法を身につける

1. 質問し問題を解決する。
2. モデルを作る。
3. 試作品をデザインする。
4. 調べる。
5. データを分析し、理解することができる。
6. 計算論的思考を用いる。

a.問題点を分ける
b.発表・プレゼンテーション
c.論理的思考
d.試行錯誤
e.規則性を見出す

7. 根拠を使って、議論を行うことができる。
8. 情報を取得・評価・伝達する。

© 2017 The LEGO Group.

プログラミング的思考とは

「プログラミング的思考」という表現を最初に提唱したのは数学者のシーモア・パパートですが、この概念を広めたジャネット・ウィング教授は、この概念を次のように定義しました。

「問題とその解決策を公式化して、解決方法を 何らかの形式で表せるようにする思考プロセス。情報処理ツールを使って効果的に実行可能である。」（ウィング教授、2011 年）

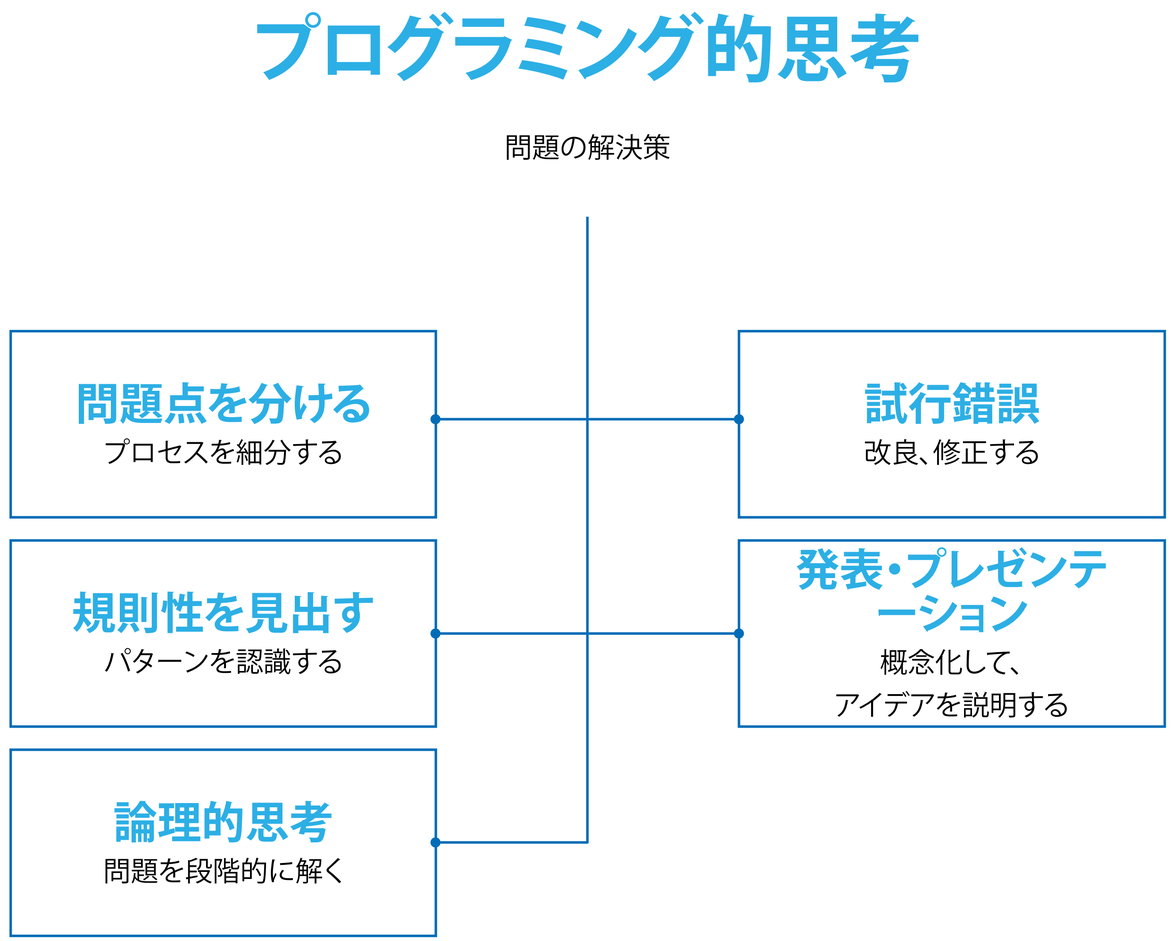
プログラミング的思考は、様々な分野や状況で、日常的に利用されます。この思考に基づくスキルは、科学（理科）、工学、数学（算数）分野で発揮されます。こうしたスキルは以下のように定義できます。

問題点を分ける
分解とは、問題を細分して、解決策を見つけるプロセスを簡略化する能力です。分解することで、問題を他人に説明したり、タスクを分割するのが簡単になります。分解は一般化につながるのが通例です。

例：休暇にでかける場合、その準備（プロジェクト）は、航空チケットの予約、ホテルの予約、荷づくりなど、いくつかのサブタスクに分けることができます。

規則性を見出す（パターンの認識）
一般化とは既知または既視感のあるタスクの部分を認識する能力です。一般化の結果、アルゴリズムの考案が簡単になることがよくあります。

例：信号は同じ動作をずっと反復することで機能します。



© 2017 The LEGO Group.

プログラミング的思考とは

論理的思考
論理的思考とは、問題を解決するために、体系的な一連の手順を作り上げる能力です。

例1：レシピを参考にして料理する際は、食事を作るために、一連の手順に従います。
例2：コンピューターゲームをする際は、一連の操作をコード化して、コンピューターに動作を指示します。

試行錯誤または修正
これは、試作品がプランどおり動くかどうかを検証し、動かない場合、改良すべき点を特定する能力です。プログラム内の誤りを見つけて修正するためにコンピュータープログラマーが必ず行うプロセスでもあります。

例1：料理する際は、時々味見をして調味料の量が適切かどうかを確認します。
例2：文章のスペルミスや句読点の打ち忘れを探す際は、間違いを見つけそれを修正して、正しく読めるようにします。

発表・プレゼンテーション力
発表・プレゼンテーション力とは、重要でない部分を省略して問題や解決策を説明する能力です。アイデアを概念にする能力とも言えます。

例：自転車を描写する際、一部の特徴に絞って説明します。タイプや色に触れるかもしれません。自転車に興味がある人にはもっと細かく説明するでしょう。

© 2017 The LEGO Group.

プログラミング的思考力を身につけるプロセス

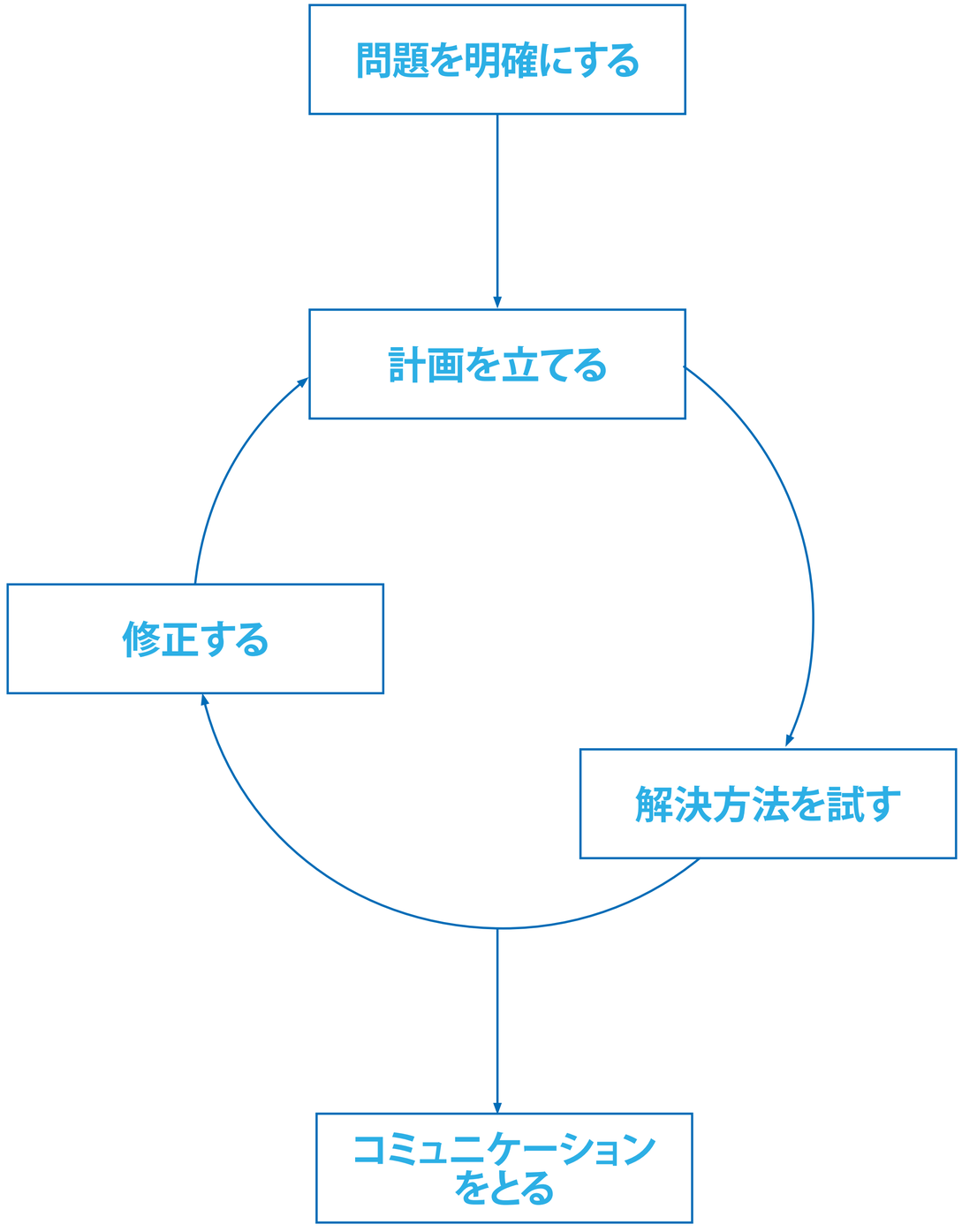
工学的デザインプロセスの利用
問題の解決策を探す際、エンジニアが踏むのがデザインプロセスです。エンジニアは一連のステージを経て、解決策にたどり着きます。各ステージでは一部のスキルを用いるか習得します。こうしたスキルこそ我々が「プログラミング的思考スキル」と呼ぶものです。

WeDo 2.0で、子どもたちは類似のプロセスを踏みます。

問題を明確にする
子どもたちにトピックが提示されます。このトピックは、改善したい問題や状況に発展していきます。細かい点がたくさんある問題もあります。そうした問題は細分すると解決しやすくなります。

問題を単純に定義し、何らかの達成基準を特定することで、子どもたちは「問題点を分ける」スキルを習得します。

質問形式にすると:
- 問題を自分で説明することができますか?
- 問題をうまく解決できたかどうかの判断方法を説明することができますか?
- 問題を細分して解決可能な単位に分解することができますか?



© 2017 The LEGO Group.

プログラミング的思考力を身につけるプロセス

計画を立てる
子どもたちは問題に対して色々な解決策をしばらく考えてから、そのうちの１つの詳しい実行プランを立てます。解決方法をつきとめるまでに通過する必要のある手順を明確にします。過去に見たことのありそうなタスクの部分を特定することで「規則性を見出す」スキルがみがかれます。

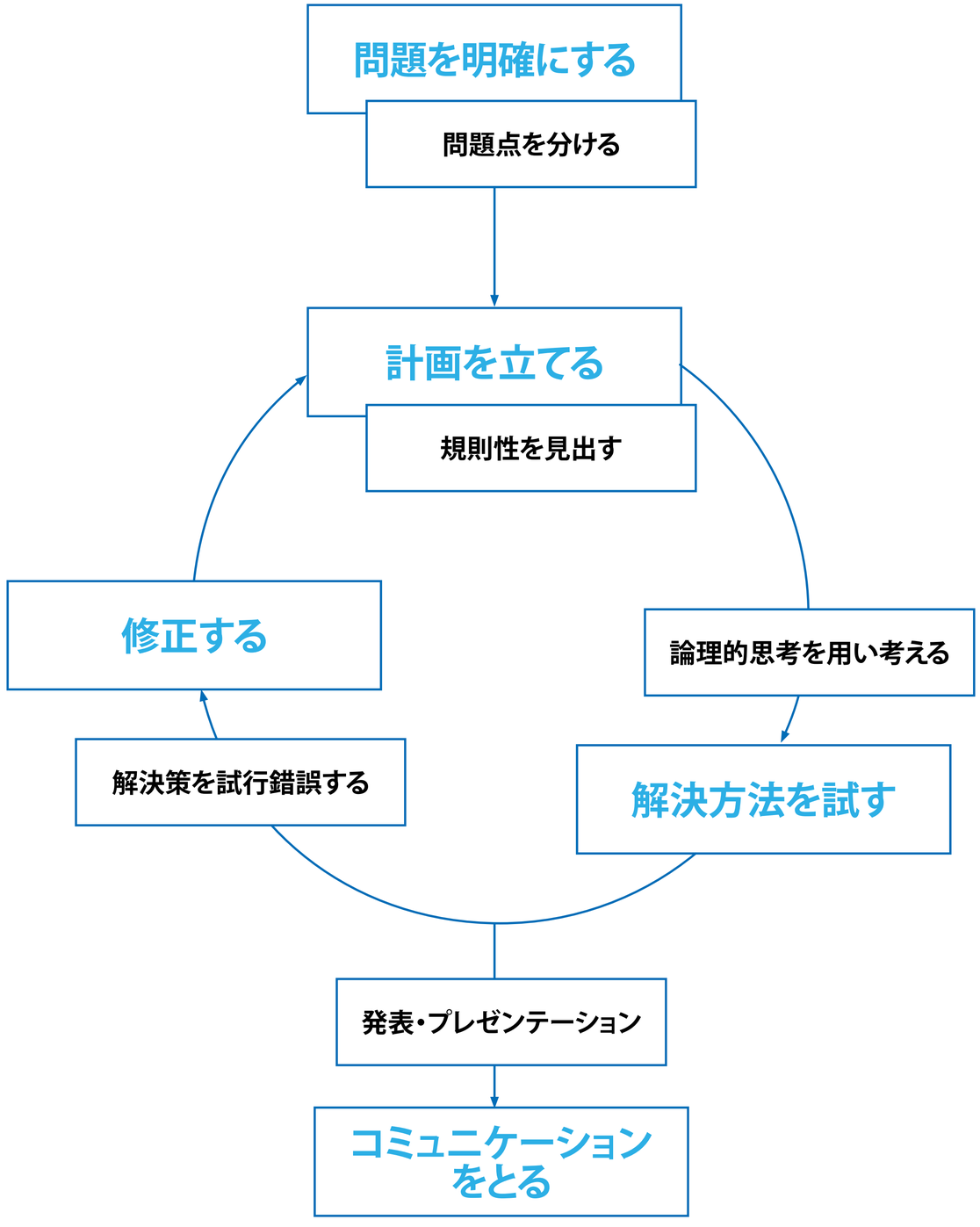
質問形式にすると：
- プログラムの動作リストを作成することができますか？
- 使えそうなプログラムの一部を特定することができますか？
- プログラムの一部を再利用できますか？

試行する
次に子どもたちはそれぞれ、解決方法の仕上げに取り組みます。プロセスのこのステージではアイコン式プログラミング言語を使って、レゴ®のモデルを起動させます。アイデアをコード化する際、論理的思考力が身につきます。

質問形式にすると：
- プログラムの解決方法をプログラミングすることができますか？
- 連続したプログラム、ループ文、条件文を使うことができますか？

修正する
子どもたちは、プログラムとモデルが成功の判断基準を満たすかどうかに基づき自分たちの解決方法を評価します。評価スキルを使って、プログラムの一部を変更、調節、修正、改良する必要があるかどうかを判断します。

質問形式にすると：
- プログラムを反復（ループ）していますか？
- プログラムの問題を調節していますか？
- 解決方法が問題と関連付けられているかどうか判断することができますか？



© 2017 The LEGO Group.

プログラミング的思考力を身につけるプロセス

コミュニケーションをとる
子どもたちは、成功基準をどのように満たしているかを説明しながら、クラス全体に自分の解決方法の完成形を披露します。解決方法を適度に詳しく説明することで、抽象的概念とコミュニケーションスキルがみがかれます。

質問形式にすると：
- 解決方法の最も重要な部分を説明していますか？
- 理解が深まるように十分な詳細を提示していますか？
- 成功基準に照らした評価をきちんと説明していますか？

© 2017 The LEGO Group.

コーディングを通じてプログラミング的思考を身につける

論理的思考を身につけるために、プログラミングの基本ルールを子どもたちに紹介します。解決方法を練る際、子どもたちは連続した動作と構造を体系化して、モデルを動かします。

以下は、WeDo 2.0のプログラミングの基本ルールのうち最もよく使われるものです。

1.出力
出力とは、作成されたプログラムによって制御されるもので、WeDo 2.0では例えば、サウンド、ライト、ディスプレイやモーターのオンとオフの切り替えが挙げられます。

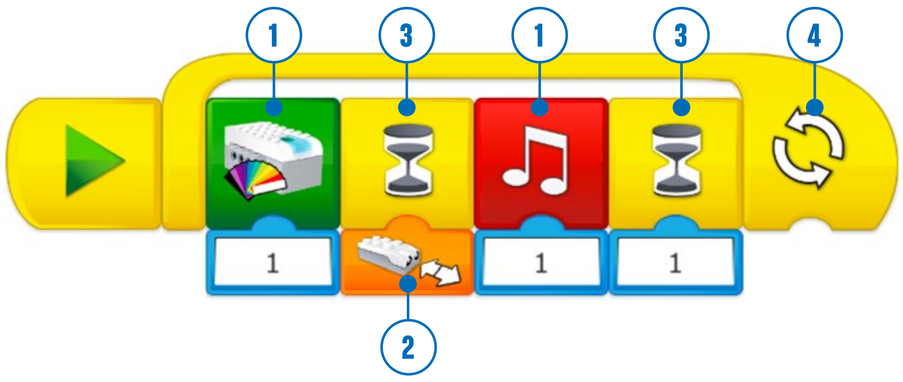
2.入力
入力とは、コンピューターやデバイスが受信する情報のことで、数やテキストの値の形でセンサーを使用して入力可能です。例えば、距離などを検知または測定するセンサーは、こうした値をデジタル入力信号に変え、プログラムで使用できるようにします。

3.イベント（待機）
連続動作を実行する前に、何か起こるまで次の動作を待つようにプログラムに指示することができます。プログラムは一定時間、またはセンサーが何かを検知するのを待つことができます。

4.ループ
ずっとまたは一定時間反復されるように動作をプログラミングすることができます。

5.機能
機能とは動作の集まりであり、特定の状況でまとめて使用されます。
例えば、ライトを点滅させるのに使用できるブロックの集まりは「点滅機能」と総称されます。

6.条件
条件とは、特定の状況でのみ実行される動作をプログラミングする場合に使用されます。プログラム内で条件を作成すると、その条件が満たされなければ、プログラムの一部は実行されなくなります。例えば、チルトセンサーが左方向にかたむくとモーターは始動し、右方向にかたむくとモーターは停止するという条件を付す場合、チルトセンサーが左方向にかたむかない限りモーターは始動せず、右方向にかたむかない限りモーターは停止しなくなります。



© 2017 The LEGO Group.