

プログラミング基礎の入口

1 導入

この講義で最重要なのは、プログラムは「状態を持ち、その状態を条件にしたがって更新していく手順」として読むことです。

コードを一行ずつ眺めるだけでは、何を 入力 として受け取り、どの時点で値が変わり、どこで処理が終わるのかが見えにくいです。この講義では、変数、条件分岐、反復、関数の役割を整理します。

2 用語と定義

変数とは、値を保存し、途中で更新できる名前付きの箱です。

条件分岐とは、条件に応じて実行する処理を分けることです。

反復とは、同じ種類の処理を繰り返すことです。

関数とは、入力を受け取って出力を返すひとまとまりの処理です。

3 方針

まず「何が変わるか」を変数として見ます。つぎに「どこで道が分かれるか」を条件分岐で見ます。そのあと、「どこまで繰り返すか」を反復で見て、最後に関数で処理を分割します。

4 直感的な説明

たとえば合計点を求めるプログラムでは、点数を足していくための変数があり、科目数ぶんだけ反復し、入力が不正なら分岐で止めることがあります。これらがどうつながっているかを自然言語で説明できることが、コードを書く前の基礎です。

5 厳密な説明

5.1 1. 状態の更新

変数 x に新しい値を代入することは、状態を更新することです。

5.2 2. 条件分岐

「もし $x > 0$ なら A、そうでなければ B」という形は、場合分けを明示したものです。

5.3 3. 反復

n 個の要素を順番に処理するなら、反復はふつう n 回実行されます。これが計算量にもつながります。

6 見分け方

- 値の変化を追いたいなら、まず変数の役割を書き出します。
- 分岐が多いときは、条件を日本語で言いなおします。
- 同じ形の処理が続くなら、反復や関数でまとめられないか考えます。

7 最終形

状態 → 分岐 → 反復 → 関数化

8 一言でいうと

- プログラミングの基礎は、コードを記号の列ではなく、状態を更新する手順として読むところから始まります。

9 関連リンク

→ [講義](#) [計算量の基本](#) [lecture](#) [information](#) [algorithm](#)
<https://study.bem130.com/lecture/information/algorithm/foundation/計算量の基本-講義/>