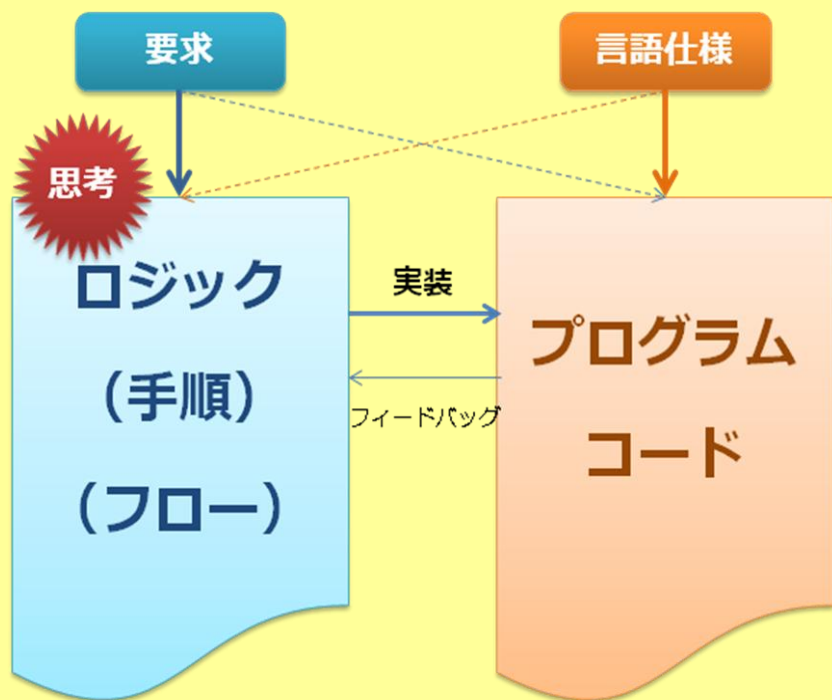


プログラムロジック 入門

～言語共通の論理的思考～
プログラミングの大原則

ロジックとプログラムコードは表裏一体の関係です。演習問題（要求）を分析して、目的と結果と条件を元に処理（目的達成のための手順）を思考して導き出すのがロジックです。ロジックからコードに実装（展開）したものがプログラムです。ロジックを考えコードに展開するすべての作業がプログラミングです。



内容

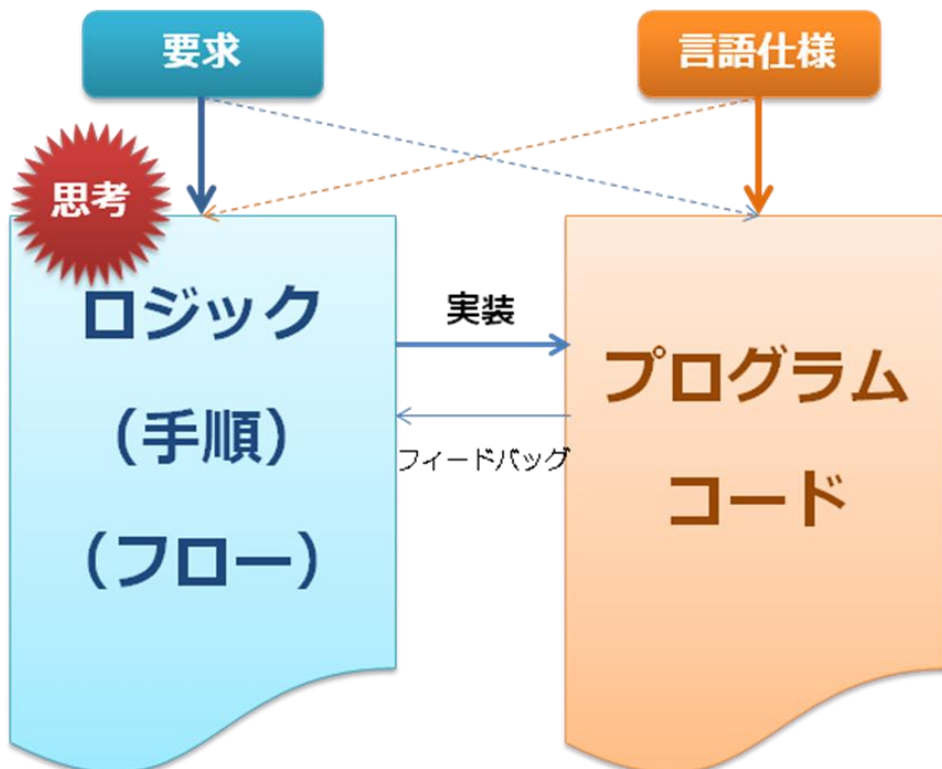
1	3
0 プログラミングの大前提.....	3
2	4
ロジックの構成要素.....	4
1 プログラムロジック 7つの構成要素.....	4
2 ロジックの要.....	5
3 合計を求める	6
4 平均を求める	7
5 最大値 / 最小値.....	8
6 2 つの変数の値を入れ換える	9
7 配列を逆順にする	10
8 再帰呼び出しとは	11
9 ファイル名を再帰的に表示する	11
10 二分探索	13
×「プログラミング言語（の文法下）でロジックを考える」	19
人の判断とコンピュータの判断の違い.....	20
それぞれの指は何本に見えますか?	21

1 はじめに

0 プログラミングの大前提

ロジックとは、ある目的を実現するための手順（過不足のないつながりの明確化と矛盾の排除）を思考し検証した結果を指します。ロジックを考えると、プログラムコードを書く前に、論理矛盾のない「目的実現のための手順」を机上で検証することが目的です。プログラミングとは、ロジックを考えそのロジックをプログラムコードに展開する作業すべてを指します。このように、ロジックは、どの環境、どの言語でも通用する共通の思考方法であり概念です。

はじめてプログラミング言語を学ぶ方は少し戸惑うかもしれません。言語文法を学べば自然にプログラムが書けるようになると思われている方もおられるでしょう。しかし、それは誤解なのです。本書では、ロジックの構成要素とプログラムロジックの構築方法（手順）について、一般的なアルゴリズムを用いてわかりやすく解説します。



© 2012 Maruyama Programming School

2 ロジックの構成要素

1 プログラムロジック 7つの構成要素

ロジックは、演算（計算、論理式）、変数（単独、配列、構造）、代入、判定、分岐、ジャンプ、関数 | メソッドの7つの要素で思考します。

演算（計算、論理式）

一般的な式です（四則演算、論理演算、高度演算など）

変数（単独、配列、構造）

演算結果やサブルーチンの結果を一時的に**保存する器**のことです

代入

変数に演算結果やサブルーチンの結果を一時的に**保存する**ことです

判定

論理式の結果です。結果は2種類で、0 or 1、真と偽、Yes or No、true or false という表現が一般的です。主に分岐に使用します。

分岐

判定した結果に応じて処理を振り分ける機能です

ジャンプ

実行先を変更（移動）します。Goto レス問題によりほとんど使用しません。

関数 | メソッド

プログラムの**実行単位**です。ある目的を実現するためにまとめられたロジックです。何度でも使用することができます。

プログラムロジックの思考はこれらの要素を組み合わせで行います。人間が行う、全体的、イメージという抽象的な思考ではなく、計算主体のデジタル思考、論理思考です。オブジェクト指向などのオブジェクトの表現には対応していませんが、メソッド内のロジックの表現は可能です。

2 ロジックの要

判定と論理式はロジックの要です。

プログラムロジックで使用する論理式には

- または（論理和、OR）
- かつ、そして（論理積、AND）
- 逆（否定、NOT）

があります。

ロジックを文章で表現する場合

「男性 かつ 18 才以上 かつ 25 才以下」

という書き方をします。

以下の文章は重複している箇所があるので最適化してみましょう。

「男性 かつ 18 才以上 かつ 25 才以下」または

「女性 かつ 18 才以上 かつ 25 才以下」

↓

「(男性 または 女性) かつ 18 才以上 かつ 25 才以下」

| ……戸籍上は男性か女性の何れかですので、判定は不要です。

↓

「18 才以上 かつ 25 才以下」

論理式の最適化は重要です。

複雑な論理式、多層構造の論理式など、一目で正しいかどうかを判断することは難しく、プログラムコードではさらに難しくなります。

論理式は表にまとめると判断しやすくなります。

以上、以下、未満、超過、等しい、…

論理式が正しいかどうかは、境目に注目します。

「18 才以上 かつ 25 才以下」は、

18 才を含む、25 才を含むです。17 才以下 または 26 才以上は対象外と理解できます。

このように、論理式のチェックは境目で行います。プログラムコードのデバッグも同様です。

論理式はロジックの中でとても重要な役割を担っています。
机上による学習よりも、ハードウェアの論理回路で学習することをお勧めします。

3 合計と平均

3 合計を求める

合計とは

二つ以上の数値を合わせまとめること。また、そのようにして出した数。

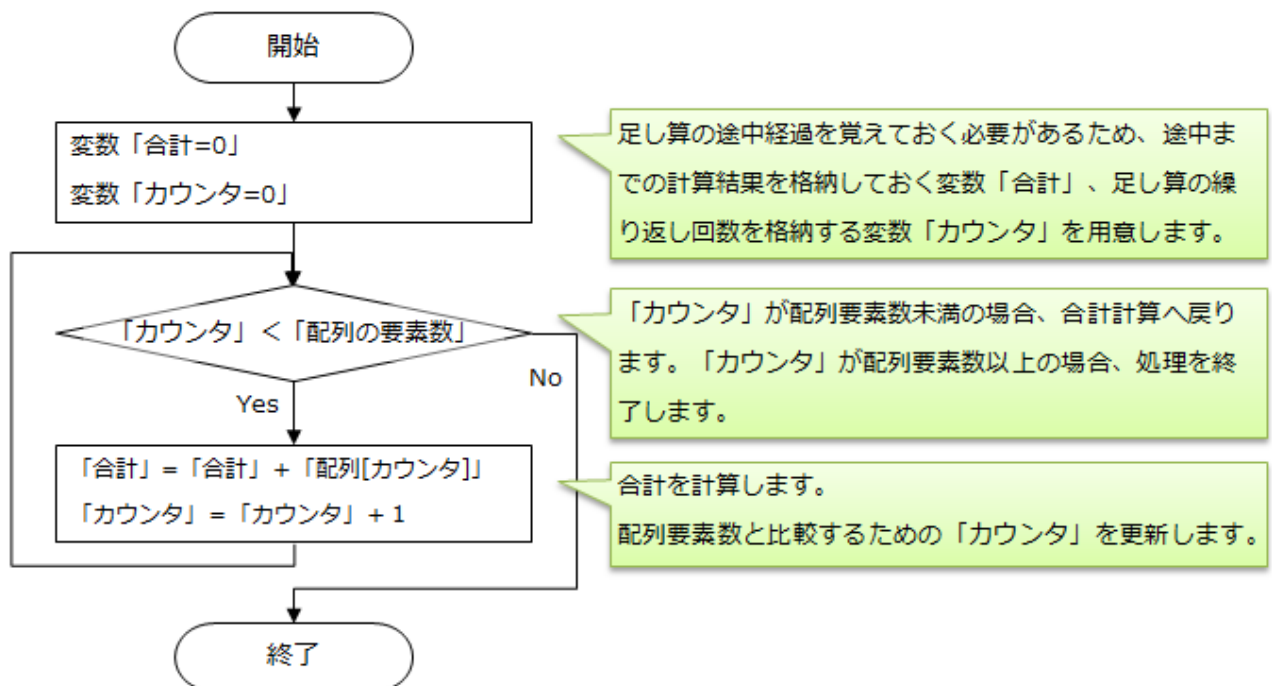
ここではあらかじめ配列に納められている数値の合計値を求めるロジックについて考えてみます。

合計 = 配列[0] + 配列[1] + 配列[2] + 配列[3] + 配列[4] + 配列[5]

この手順はとても簡単で、正しい答えが得られますが、汎用性がありません。配列の要素数が変わった時にはロジックを書き直さなければいけませんし、それ以前に配列の要素の数が多い時にはロジックを書くこと自体が大変な作業になります。別の手順を考えるために、まず、人間がこのような合計値を求める場合にどう計算するかを考えてみます。次のような手順で計算をするはずですが。

1. 最初に 2 つの数の足し算をする。
2. その足した値に 3 つ目の数を足して 3 つ目までの数の合計値を求める。
3. これを最後の数まで繰り返して計算結果を得る。

これをフローチャートで表現してみます。



フローチャートの書き方は手順を視覚化することが目的です。書き方は統一するようにします。