情報科学2

2019年　神戸大学　担当　片山　修

# プログラミング

プログラミングとは、コンピュータに実行させたい処理を記述していくことです。実際に記述されたものをプログラムといい、記述内容は、コンピュータに分かる「0,1」の2進数である「機械語」である必要があります。しかし、機械語の記述は人が理解しにくいため、人が書きやすく理解しやすいためのプログラミング言語がたくさん存在しています。機械語などコンピュータが直接読み取れる言語を「低級言語」といい、逆に人間が理解しやすいようになっている言語を「高級言語」といいます。高級言語で書かれたプログラムを機械語に変換することをコンパイルといいますが、プログラミング言語には、コンパイルが必要な「コンパイル言語」と必要ではない「インタープリタ言語」があります。

■コンパイル言語  
人が書いたプログラムをコンパイルによって機械語にするため、高速で動作します。その代わりにプログラミングでの記述が面倒です。  
例：C、C++、Java、

■インタープリタ言語  
手軽にプログラムを書くことができ、実行結果もすぐに得ることができます。しかし、プログラムを機械語に変換しながら実行しますので、実行速度が遅くなります。  
例：JavaScript、PHP、Python、Ruby

# アルゴリズム

アルゴリズムとは、ある問題を解くための実行手順を表したものです。これを図式化したものがフローチャートと呼ばれています。フローチャートでは次の記号が使用されます。

|  |  |  |
| --- | --- | --- |
| 記号 | 名称 | 働き |
|  | 端子 | 流れ図の最初と最後を示します。  記号の中に”開始(start)”、”終了(end)”  と表示します。 |
|  | 流れ線 | 記号同士を結び付けます。  上から下、左から右へと流れを追っていきます。  逆のときは矢印を付けます。 |
|  | 処理 | 作業におけるさまざまな処理を表します。 |
|  | 入出力 | 記号の中に、入力する情報や出力する情報を記述します。 |
|  | 結合子 | 記号中に記されている数字に流れが続いていきます。 |
|  | 判断 | 条件によって流れを分岐します。  A：B  =  <  Aの値がBの値より小さい場合は「＜」  等しい場合は「＝」  大きい場合は「＞」と判断  >  A：B  ≧  <  Aの値がBの値より小さい場合は「＜」  大きいか同じ場合は「≧」と判断  A<5  NO  YES  Aの値が５以上ならNO  ５より小さいならYESと  判断  ≧  <＜  A＞10 のとき①  A< 5 のとき②  5≦A≦10のとき③  A：10  >  ≦  ①  ②①  ③②①  A：5 |

**例：物を買う行動についての流れ図**

店に行く

物を買う

(所持金を減らす)

帰る

買う物がある

値段≦所持金

YES

YES

NO

NO

### 繰り返しのある流れ図

流れ図の中で数字を扱う場合、数字を入れておく入れ物として**変数**を用意します。その入れ物の名前を**変数名**と呼びます。変数名はアルファベットで表しますが、なるべく何を入れるものか分かるような名前をつけます。

両図とも変数Nに設定された回数だけ処理を繰り返して終了します。変数Iは繰り返し回数を数えるループカウンタです。両方とも同じ流れになっていますが、Nが0のとき左図では、1度も処理を実行することなく終了していますが右図では、処理を1回は行って終了します。

START

処理

I＋１→I

I：N

N ←繰り返し回数

I ←　0

≧

<

END

START

END

処理

I＋１→I

I：N

N ←繰り返し回数

I ←　0

≧

<

**例：合計を求める流れ図**

START

END

D =　-1

SUM ←　0

I ←　0

ファイルから1行ずつデータを読み

Dに代入する

YES

NO

SUM+D → SUM

10個の正の数字とデータの終わりをあらわす「-1」という数字が1行に1つ入っているファイルがあります。下の流れ図では、このファイルを1行ずつ読み込んでDとし、そのSUM+Dの結果をSUMに代入することで10個の数字の合計を求めています。

**例題：平均を求める流れ図**上の例の合計を求める流れ図を基に、平均を求める流れ図です。(ア)と(イ)に入る式を求めなさい。ただし、データの個数が10個であるとは分かっておらず、-1がデータの終わりであることだけが分かっているとする。

START

END

D =　-1

ファイルから1行ずつデータを読み

Dに代入する

YES

NO

SUM+D → SUM

(ア)　→ COUNT

SUM ←　0

I ←　0

COUNT ← 0

(イ) → HEIKIN

**例題：最大値を検出する流れ図**

　　身長のデータを１つずつ読み取り、身長の最大値を求めます。

START

ファイルから1行ずつデータを読み

Dに代入する

YES

0 → MAX

データEND

NO

(ア)

D：MAX

最大値

MAXを出力

(イ)

(ウ)

END

**例題：値を検出して挿入する流れ図**

　　　　次のように、Aという配列に5つの数字が入っていて昇順に並んでいます。

10

15

18

20

35

A

(0)

(1)

(2)

(3)

(4)

配列番号:

(5)]

ここに数19を挿入するための流れ図を作成します。(18と20の間に入ります。19でなくても対応できるようにしますが、35より大きい数字は考慮していません。)

START

END

1 → i

5 → j

(ア)

j：i

19：A(i)

<

A(j-1) → A(j)

(イ)

≧

=

>

(ウ)