



# Java による アルゴリズムと データ構造の基礎

永田 武【著】

コロナ社



## まえがき

本書は、大学、短大と高等専門学校の学生、あるいは情報系企業に入社した新人の方を対象として記述したものです。アルゴリズムとデータ構造は、今日の情報化された現代社会の膨大な情報を蓄積・管理して、利用するための情報処理システム開発の基礎となる内容です。

アルゴリズムとは、問題を解くための手順を表現したもので、コンピュータにおける情報処理の基盤となるものです。そして、データ構造とは、データの集合をコンピュータで効率的に扱うために一定のルールに従って格納するときの形式のことです。多くの場合、データ構造が決まれば、利用するアルゴリズムは比較的容易に決まります。しかし場合によっては、与えられた仕事をこなすための最適なアルゴリズムを利用するために、そのアルゴリズムを使うことが前提となっているデータ構造が選択されることもあります。このようにアルゴリズムとデータ構造は切っても切れない関係にあります。

本書は、学校においては、週一回の半期で履修できる程度の内容になっています。各章の最後には、関連プログラムの項を設けました。ここでは、実用的なプログラムを掲載していますので、卒業研究などの場面でも役に立つと思います。OSはWindowsでもLinuxでもかまいません。自分の手で作成し動作を確認すると理解が深まると思います。また、基本情報処理技術者試験の過去問題にも触れ、読者がより深く理解できるように工夫しました。さらに、付録にはWindowsやLinuxでの開発において役立つ内容を記載していますので参照してください。本書がアルゴリズムとデータ構造の学習への扉となれば、著者にとって望外の喜びです。

最後に、本書の出版の機会を与えていただいた株式会社コロナ社に厚くお礼申し上げます。

2019年3月

永田 武

## 目 次

### 第 1 章 Java 入門

|     |                 |    |
|-----|-----------------|----|
| 1.1 | Java の特徴        | 1  |
| 1.2 | Java プログラム開発の流れ | 2  |
| 1.3 | Java プログラミングの作法 | 3  |
| 1.4 | Java プログラミングの基礎 | 5  |
| 1.5 | オブジェクト指向        | 8  |
| 1.6 | 新しいクラスの作成       | 8  |
| 1.7 | 関連プログラム         | 11 |
|     | 演習問題            | 24 |

### 第 2 章 基本的なアルゴリズム

|     |                      |    |
|-----|----------------------|----|
| 2.1 | フローチャート              | 26 |
| 2.2 | 判 断                  | 27 |
| 2.3 | 反復（ループ）              | 29 |
| 2.4 | 基本情報技術者試験での疑似言語の記述形式 | 32 |
| 2.5 | 関連プログラム              | 33 |
|     | 演習問題                 | 36 |

### 第 3 章 配 列

|     |                  |    |
|-----|------------------|----|
| 3.1 | 配列とは             | 39 |
| 3.2 | 多次元配列            | 41 |
| 3.3 | Java クラスライブラリの利用 | 43 |
| 3.4 | クラスの配列           | 44 |
| 3.5 | 関連プログラム          | 45 |
|     | 演習問題             | 53 |

## 第4章 再帰

|     |            |    |
|-----|------------|----|
| 4.1 | 再帰とは       | 56 |
| 4.2 | 階乗         | 57 |
| 4.3 | ユークリッドの互除法 | 58 |
| 4.4 | ハノイの塔      | 59 |
| 4.5 | 関連プログラム    | 61 |
|     | 演習問題       | 65 |

## 第5章 連結リスト

|     |          |    |
|-----|----------|----|
| 5.1 | 連結リストとは  | 67 |
| 5.2 | 単方向リスト   | 68 |
| 5.3 | 双方向リスト   | 73 |
| 5.4 | 循環リスト    | 74 |
| 5.5 | 双方向循環リスト | 74 |
| 5.6 | 関連プログラム  | 77 |
|     | 演習問題     | 83 |

## 第6章 スタックとキュー

|     |                  |    |
|-----|------------------|----|
| 6.1 | スタック             | 87 |
| 6.2 | キュー              | 91 |
| 6.3 | Java クラスライブラリの利用 | 94 |
| 6.4 | 関連プログラム          | 96 |
|     | 演習問題             | 97 |

## 第7章 木構造

|     |         |     |
|-----|---------|-----|
| 7.1 | 木構造とは   | 100 |
| 7.2 | 2分探索木   | 103 |
| 7.3 | ヒープソート  | 106 |
| 7.4 | 関連プログラム | 113 |
|     | 演習問題    | 114 |

## 第8章 探索

|     |            |     |
|-----|------------|-----|
| 8.1 | 線形探索       | 117 |
| 8.2 | 番兵を用いた線形探索 | 119 |
| 8.3 | 2分探索       | 121 |
| 8.4 | ハッシュ法      | 124 |
| 8.5 | 関連プログラム    | 132 |
|     | 演習問題       | 133 |

## 第9章 ソート (その1)

|     |         |     |
|-----|---------|-----|
| 9.1 | ソートとは   | 137 |
| 9.2 | バブルソート  | 138 |
| 9.3 | 選択ソート   | 140 |
| 9.4 | 挿入ソート   | 142 |
| 9.5 | 関連プログラム | 145 |
|     | 演習問題    | 146 |

## 第10章 ソート (その2)

|      |         |     |
|------|---------|-----|
| 10.1 | シェルソート  | 148 |
| 10.2 | クイックソート | 152 |
| 10.3 | マージソート  | 155 |

|      |                        |     |
|------|------------------------|-----|
| 10.4 | Java クラスライブラリの利用 ..... | 159 |
| 10.5 | 関連プログラム .....          | 162 |
|      | 演習問題 .....             | 166 |

## 第 11 章 グラフ

|      |               |     |
|------|---------------|-----|
| 11.1 | グラフとは .....   | 169 |
| 11.2 | 最短経路問題 .....  | 171 |
| 11.3 | 関連プログラム ..... | 177 |

## 付 録

|    |                            |     |
|----|----------------------------|-----|
| A. | vi によるソースファイルの作成 .....     | 182 |
| B. | Windows と Linux コマンド ..... | 183 |
| C. | CLASSPATH の設定方法 .....      | 184 |

|            |     |
|------------|-----|
| 参考文献 ..... | 185 |
|------------|-----|

|          |     |
|----------|-----|
| 索引 ..... | 186 |
|----------|-----|

### 演習問題解答について

下記の書籍詳細ページ内の「▶関連資料」をクリックすると、演習問題の解答を確認することができます。

<http://www.coronasha.co.jp/np/isbn/9784339028966/>

※コロナ社の Web ページから本書の書名検索でも書籍詳細ページにアクセスできます。

# 第 1 章 Java 入門

現代社会の膨大な情報を利用するための情報処理システムの開発には、アルゴリズムとデータ構造についての理解が重要である。そのプログラムの設計において、データ構造の選択は重要な課題である。これまでに、配列、連結リスト、スタック、キュー、木構造やグラフなどのデータ構造が開発され、それらのデータ構造を用いた効率の良い探索やソートなどのアルゴリズムが開発されている。本書では、そのようなアルゴリズムとデータ構造について解説を行うが、アルゴリズムの確認のために Java 言語を用いる。本章では、Java の特徴、プログラム開発の流れ、プログラミングの作法、プログラミングの基礎、オブジェクト指向、新しいクラスの作成、および関連プログラムについて説明する。

## 1.1 Java の特徴

Java は、1994 年に発表されたオブジェクト指向言語である。Java のコンパイラは、実行するプラットフォームに対応した形式（ネイティブコード（native code））に変換するのではなく、その手前の中間言語まで変換する。中間言語は、バイトコード（bytecode）と呼ばれる形式で保存され、いろいろな OS で解釈される。

Java は、簡潔さと機能の豊富さのバランスのとれた言語であり、以下のような特徴を有している。

- ・オブジェクト指向
- ・プラットフォームに依存しない性質
- ・豊富なネットワーク関連の機能
- ・充実した標準クラスライブラリ
- ・プロセスを並行動作させるマルチスレッド
- ・使われなくなったメモリ領域を自動的に整理するガーベージコレクション

・ 例外処理

## 1.2 Java プログラム開発の流れ

図 1.1 に Java プログラム開発の流れを示す。

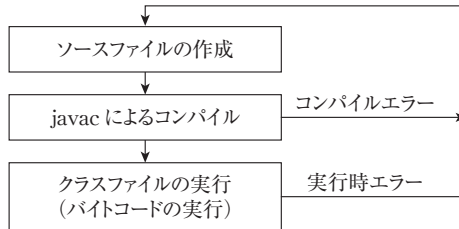


図 1.1 Java プログラム開発の流れ

### (1) ソースファイルの作成

システムの要求仕様に従って書いた設計書に基づいて、ソースコード (source code) を作成する。ソースコードを作成することをコーディング (coding) という。Windows では TeraPad, Linux では vi などのエディタ (editor) を用いてコーディングし、プログラム名を付けて、コンピュータに保存する。保存したファイルをソースファイル (source file) と呼ぶ。Java では、保存するソースファイル名とクラス名を同じにしなければならない。また、ソースファイルの拡張子を “.java” にする必要がある。

### (2) javac によるコンパイル

ソースコードをバイトコードに変換するためにコンパイル (compile) を実施する。コンパイルは、ソースコードをコンピュータが理解できるコードに翻訳することである。コンパイルには、コマンドプロンプトで javac コマンドを用いる。

コンパイルが成功すると、なにも表示されずに入力プロンプトが表示される。このとき、拡張子が “.class” というバイトコードが生成される。これをク



ラスファイル (class file) と呼ぶ。

一方、コンパイルに失敗すると、コンパイルエラーが表示される。エラーメッセージには行番号やその理由が表示されているので、それらの情報を参考にソースファイルを修正して再度コンパイルする。

### (3) クラスファイルの実行 (バイトコードの実行)

クラスファイルとは、コンピュータ固有の環境とは独立した Java 仮想マシン (Java VM) 上の実行ファイルである。クラスファイルの実行には java コマンドを用いる。コンパイルでエラーとならなかったクラスファイルでも実行時にエラーが発生する場合がある。一般に、プログラムには誤りが含まれる。この誤りのことをバグ (bug) と呼ぶ。このバグを修正することをデバッグ (debug) という。

## 1.3 Java プログラミングの作法

### (1) コメント

可読性のあるよいプログラムを作成するためには、コメントを付けることが重要である。コメントは表 1.1 に示すような種類がある。

表 1.1 コメントの種類

| No. | 種類          | 記述方法  |
|-----|-------------|---|
| 1   | ブロックコメント    | 複数行にわたるコメントを記述するときには、 <code>/* */</code> でコメントを囲む。  |
| 2   | 1行コメント      | 1行のコメントを記述するときには、 <code>/* */</code> または、 <code>//</code> を用いる。                               |
| 3   | 後置きコメント     | 1行の対象コードの行末に <code>//</code> を用いて、短いコメントを記述する。  |
| 4   | ロジックのコメント   | デバッグ時にロジックをコメントアウトする際に使用する。該当のコードの先頭に <code>//</code> を記述する。                                  |
| 5   | ドキュメント用コメント | ドキュメント用コメントは、 <code>javadoc</code> コマンドで API ドキュメントを生成する際に処理されるもので、 <code>/** */</code> を用いる。 |

## (2) 識別子の命名規則

変数名やクラス名には、下記の規則以外の任意の名前を付けることができる。

- ・予約語（public など）は使用できない。
- ・先頭に数字は使用できない。
- ・記号は \$（ドルマーク）と \_（アンダースコア）のみ使える。

しかし、慣習的に表 1.2 のような命名法が用いられているので、それに従ったほうがよい。

表 1.2 慣習的な命名法

| No. | 命名法          | 説明   |
|-----|--------------|--|
| 1   | クラス名の先頭は大文字  | クラス名の先頭を大文字にすることにより、クラス名であることを明確にする。(例) Person                 |
| 2   | メソッド名の先頭は小文字 | メソッド名の先頭は小文字にすることにより、コンストラクタやクラス名と区別をつけやすくする。(例) bubbleSort    |
| 3   | 変数名の先頭は小文字   | 変数名の先頭は小文字にすることにより、コンストラクタやクラス名と区別をつけやすくする。(例) cost            |
| 4   | 定数名はすべて大文字   | 予約語 final が付いた代入不可の定数は、すべて大文字とする。(例) ARRAY_MAX                 |
| 5   | キャメルケース      | 複数の単語を接続して識別子とするときは、後の単語の先頭を大文字にして区切りを表現する。<br>(例) vehicleType |
| 6   | スネークケース      | 複数の単語を接続して識別子とするときは、_（アンダースコア）で区切る。(例) vehicle_type            |

## (3) インデント

インデント（indentation style）とは、メソッドの中身や if 文の処理ブロックなどを明確にするために、対象となる部分の文字を、何文字か字下げすることである。読みやすいプログラムとするために、インデントは 4 文字が適切である。インデントは、Tab キーを用いるか、スペースで行うことができるが、いずれかに統一したほうがよい。

## (4) クラスの形式

Java のクラスは一般的に図 1.2 の形式で記述する。図に示すように、クラスは大きく 3 つの部分から構成される。まず、このクラス（class）が保持して

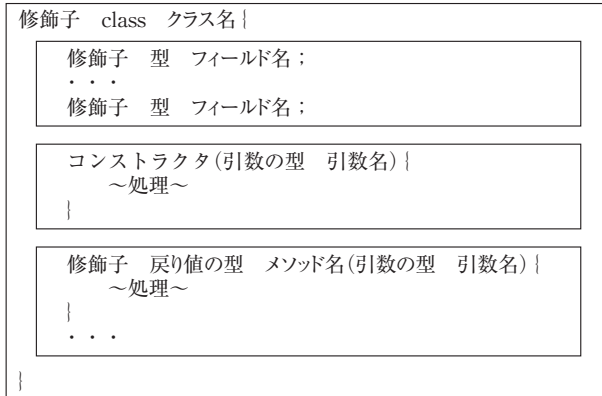


図 1.2 Java のクラスの形式

いる変数であるフィールド (field) の集合、つぎに、クラスのインスタンス生成時に実行されるメソッドであるコンストラクタ (constructor)、そして、このクラスが保持している処理に対応するメソッド (method) の集合である。フィールドはメンバ変数 (member variable) とも呼ばれる。また、修飾子には、アクセス修飾子 (public, protected, private) とそれ以外の修飾子 (static, final, abstract など) があるが、それらの詳細についてはほかの Java の書物を参照してほしい。本書では、アルゴリズムの確認のために提示するプログラムにおいて必要に応じて説明する。

## 1.4 Java プログラミングの基礎

それではプログラミングの定番である「Hello world Java!」と画面に表示させるプログラムを例題に、プログラム開発の流れを説明する。以下の作業は、Windows の場合はコマンドプロンプトを、Linux の場合は端末を表示させて行う。

### (1) ソースファイルの作成

まず、プログラム 1.1 を参照してソースファイルの作成をする。なお、Linux における「vi によるソースファイルの作成」については、本書の最後

# 索引

|             |          |
|-------------|----------|
| <b>【あ行】</b> |          |
| 後入れ先出し      | 87       |
| 後判定         | 32       |
| 安定          | 137      |
| 安定ソート       | 137      |
| 行きがけ順       | 102      |
| インスタンス      | 8        |
| インデックス      | 39       |
| インデント       | 4        |
| エッジ         | 100, 169 |
| エディタ        | 2        |
| エンキュー       | 91       |
| オブジェクト      | 8        |
| オープンアドレス法   | 128      |
| 重み付き無向グラフ   | 172      |
| 親           | 100      |

|             |      |
|-------------|------|
| <b>【か行】</b> |      |
| 階乗          | 57   |
| 外部ソート       | 138  |
| ガウスの消去法     | 50   |
| 帰りがけ順       | 102  |
| 拡張 for 文    | 30   |
| カプセル化       | 8    |
| 可変長引数       | 164  |
| 関係演算子       | 28   |
| 完全二分木       | 103  |
| キー          | 145  |
| 木構造         | 100  |
| 疑似言語        | 32   |
| キュー         | 91   |
| クイックソート     | 152  |
| 空間計算量       | 117  |
| クラス         | 4, 8 |
| クラス階層       | 8    |
| クラスファイル     | 2    |

|            |     |
|------------|-----|
| グラフ        | 169 |
| 継承         | 8   |
| 結合荷重       | 77  |
| ゲッター       | 9   |
| 子          | 100 |
| 降順         | 137 |
| 構造化プログラミング | 26  |
| 後続ノード      | 68  |
| 誤差逆伝播法     | 77  |
| 後退代入       | 51  |
| コーディング     | 2   |
| 根          | 100 |
| コンストラクタ    | 5   |
| コンパイル      | 2   |
| コンパレータ     | 160 |

|             |     |
|-------------|-----|
| <b>【さ行】</b> |     |
| 再帰          | 56  |
| 再帰的定義       | 56  |
| 再帰的呼出し      | 61  |
| 最小ヒープ       | 107 |
| 最大公約数       | 58  |
| 最大ヒープ       | 107 |
| 最短経路問題      | 171 |
| 先入れ先出し      | 91  |
| サブクラス       | 8   |
| 参照          | 39  |
| 参照型         | 39  |
| ジェネリックス     | 43  |
| シェルソート      | 148 |
| 時間計算量       | 117 |
| シグモイド関数     | 78  |
| 自己参照型       | 69  |
| ジャンプ探索      | 132 |
| 循環リスト       | 74  |
| 昇順          | 137 |
| 衝突          | 125 |

|          |     |
|----------|-----|
| 枢軸       | 153 |
| スタック     | 87  |
| スーパークラス  | 8   |
| セッター     | 9   |
| 線形探索     | 117 |
| 線形リスト    | 67  |
| 先行ノード    | 68  |
| 前進消去     | 50  |
| 選択ソート    | 140 |
| 先頭ノード    | 68  |
| 挿入ソート    | 142 |
| 双方向循環リスト | 74  |
| 双方向リスト   | 73  |
| 添え字      | 39  |
| 属性       | 8   |
| ソースコード   | 2   |
| ソースファイル  | 2   |
| ソーティング   | 117 |
| ソート      | 137 |

|             |     |
|-------------|-----|
| <b>【た行】</b> |     |
| ダイクストラ法     | 171 |
| 多次元配列       | 41  |
| 探索          | 117 |
| 単方向リスト      | 68  |
| チェイン法       | 125 |
| デキュー        | 91  |
| デッキ         | 94  |
| デバッグ        | 3   |
| 通りがけ順       | 102 |
| ドット演算子      | 9   |

|             |     |
|-------------|-----|
| <b>【な行】</b> |     |
| 内部ソート       | 138 |
| ニューラルネットワーク | 77  |
| ニューロン       | 77  |
| 根           | 100 |

ネイティブコード 1  
ノード 100, 169

## 【は行】

排他的論理和 78  
バイトコード 1  
配列 39  
バグ 3  
ハッシュ関数 124  
ハッシュテーブル 124  
ハッシュ値 124  
ハッシュ法 124  
ハッシング 124  
ハノイの塔 59  
幅優先探索 101, 177  
バブルソート 138  
番兵 119  
ヒープ 106  
ヒープソート 107  
ピボット 152  
フィボナッチ数列 61  
フィールド 5  
深さ優先探索 101, 179  
複素数 34

プッシュ 87  
ブランチ 100, 169  
プリミティブ型 39  
フローチャート 26  
プロパティ 8  
分割統治法 56  
ポップ 87

## 【ま行】

前判定 31  
マージソート 155  
末尾ノード 68  
マップ 145  
無限ループ 29, 32  
無向グラフ 169  
メソッド 5, 8  
メッセージパッシング 8  
メモ化 64  
メンバ変数 5

## 【や行】

有向グラフ 169  
ユークリッドの互除法 58

## 【5行】

乱数 13  
乱数の種 15  
リンク 100, 169  
隣接行列 169  
隣接リスト 169  
レベル 100  
連結リスト 67  
論理演算子 28

## 【英字】

Java 仮想マシン 3  
Java クラスライブラリ 43, 93, 159  
new 演算子 39  
O 記法 117

## 【数字】

2分木 103  
2分探索 121  
2分探索木 103  
2分ヒープ 107  
3項演算子 29

— 著者略歴 —

1980年 広島大学大学院博士課程前期修了（回路システム工学専攻）  
1980年 株式会社 東芝勤務  
1989年 松江工業高等専門学校講師  
1991年 松江工業高等専門学校助教授  
1995年 博士（工学）（広島大学）  
1997年 広島工業大学助教授  
2001年 広島工業大学教授  
現在に至る

特種情報処理技術者  
電気学会フェロー

主な著書  
電力システム工学の基礎（コロナ社）  
データベースの基礎（コロナ社）

## Java によるアルゴリズムとデータ構造の基礎

Algorithms and Data Structures with Java

© Takeshi Nagata 2019

2019年8月30日 初版第1刷発行



検印省略

|     |            |
|-----|------------|
| 著者  | なが た 武     |
| 発行者 | 株式会社 コロナ社  |
| 代表者 | 牛来真也       |
| 印刷所 | 萩原印刷株式会社   |
| 製本所 | 有限会社 愛千製本所 |

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844・電話(03)3941-3131(代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-02896-6 C3055 Printed in Japan

(新井) N



**JCOPY** <出版者著作権管理機構 委託出版物>

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。