

はじめに

Java は非常に普及率が高く、プログラミング言語としても言語処理系としてもよく整備されている。また Java は比較的新しい処理系であるため、新しいプログラミングパラダイムも多数盛り込まれており、プログラミングの学習に Java を採用するのは良い選択であるといえる。しかし、Java はその応用範囲の広さと実用性の高さから学ぶべき項目は非常に多く、実用的なプログラミングが可能になるまでには、多くの時間と労力を学習に費やさなければならない。本書は Java の導入学習からはじまり、初級レベルを脱するまでの教材であり、学習にかかる労力を最小限に抑えることを目標として執筆された。

Java をプログラミング学習の導入に用いることに関して異論を唱える人も少なからず存在する。Java 言語の学習においては、その言語自体の実用性と完成度の高さから、プログラミングの初心者にとって馴染みが薄く理解しにくいパラダイムが早い段階で出てくる。オブジェクト指向プログラミングに関連する抽象度の高いパラダイムはその 1 例である。そのような現状から、プログラミング学習の導入には演習 / 学習を念頭に置いた模擬的な言語処理系を用いて、その後で多種・多岐にわたる言語処理系の学習に繋げるというのがプログラミング学習における 1 つの正当な流れであるという主張には筆者も異論は無い。しかし、多くの時間をさいて異なる言語処理系にわたる形でプログラミング言語の学習をすることとは別に、Java という 1 つの言語処理系で短期間で最大限のことを学ぶことは、プログラミングの実利的な面を認識できるという意味で非常に大きな価値があると筆者は考えており、本書がそのために役立てば幸いである。

Java のような大きな体系の言語処理系を学ぶには、「どの範囲をどの深さで学ぶか」という点について、学習の初期の段階で絞り込む必要がある。なぜならば、利用可能性の認識ができないまま余りに多くの学習事項が提示されると、学習者の学習意欲が著しく低下するからである。実際に、書籍をはじめとする Java に関する情報リソースは非常に多く、学習初心者が独力で適切なものを選び集めるのは難しい。本書は Java の学習初心者に対して書かれたものであり、プログラミングの導入学習から GUI アプリケーション構築の基礎、グラフィックスの取り扱いの基礎までを対象範囲とする。具体的には

- (1) Java 言語の基礎事項
- (2) アルゴリズムの基本的な考え方
- (3) GUI アプリケーションの構築方法の基礎 (Swing と SWT の両方)
- (4) ビットマップグラフィックスの描画方法の基礎 (Swing と SWT の両方)
- (5) スレッドプログラミングの基礎

について解説する。本書の内容を学習することで、独力で学習を続けてゆくための基礎力が養えるものと筆者は期待している。

本書は Java 言語の全てについて言及するものではなく（そのような書物を完成させるのは現実的に不可能であるようにも思われる）、学習者が Java の実利的な側面を認識しながら、中級レベルのプログラミングに移行するまでの内容に限られている。しかし、大学の学部や高等専門学校の教育課程で要求されるプログラミング技能（Java に限る）の多くの部分を得ることができる内容になっている。

本書の使い方など

本書の内容は Java 言語について知識のない学習者を想定して書かれている。しかし、既に習得しているレベルに応じて、内容を選択して読み進めることも可能である。例えば、Java 以外のプログラミング言語を既に十分習得している人にとっては、第 3 章「代表的なアルゴリズム」は読む必要がないかもしれない。あるいは、Java 言語の基本的な部分の学習を終え、GUI の構築やグラフィックス作成の方法について知りたい人は第 4 章以降が必要となることも考えられる。ただ、他の言語の経験の有無を問わず、Java 言語について全く知らない人は第 1 章から読み進める必要がある。

「はじめに」でも述べたように、本書は Java 言語のプログラミングに関する全てに言及するものではなく、入門レベルから初級レベルを脱するまでの内容となっている。従って、本書の内容を習得し終えたら（あるいは学びながら）、他の情報リソースを引用しながら学習範囲を拡大することが望ましい。Java に関する技術資料のほとんどは ORACLE の Web サイトから入手可能であり、巻末の参考文献をはじめとする良い書籍が多数出版されているので、学習目的や関心に合ったものを入手することが容易である。Web の情報リソースに関していえば、非公式のもの（有志の方々の Web サイト）が多く公開されており、Java 言語の学習に必要な情報源は非常に豊富である。それら書籍や情報源にある内容を十分に理解して活用するための出発点としても本書が役立つと著者は考えている。

アルゴリズムの学習は特定のプログラミング言語に限定せずに深めてゆく必要がある。巻末の参考文献にある [2] はプログラミングに関する知識を深めるのに良い書籍であり、アルゴリズムの学習のために推奨する。本書で取り上げたアルゴリズムの選定に関しては参考文献 [5] に依るところが大きい。[5] ではプログラミングのための基礎知識となる必要最低限のアルゴリズムを厳選して紹介しているだけでなく、サンプルプログラムも無駄のない洗練された形で記述されている。ただし無駄なく綺麗に書かれたプログラムは、初心者の独学には敷居が高い場合もあり、本書では初心者の理解の流れに極力沿うような形を意識してサンプルプログラムを記述している。従って表現方法という意味では、本書のサンプルプログラムは美しくないものも多いので、本書でアルゴリズムの学習を終えた人やあるいは学習中の人がアルゴリズムの表現力を高めるために [5] は良い書籍であり、ここに推奨する。

前提とするシステム環境

本書の内容は Microsoft Windows を OS とするシステム環境を想定して執筆されている。また本書の内容を作成するにあたり、筆者は 32 ビット版の Windows7 を使用している。他のバージョンの Windows や 64 ビット版の Windows、あるいは Linux や Mac OS の環境においても本書の内容の本質的な部分はそのまま通用するが、システム環境の違いに由来する些細な差異に関しては、読者の環境に合わせて読み替えていただきたい。

目次

1	Java とは	1
1.1	Java 言語処理系	1
2	Java によるプログラミング入門	1
2.1	コマンド操作による最も素朴な実行例	1
2.1.1	開発環境「CPad for Java2」(JCPad) の利用 (Microsoft Windows 用)	2
2.2	プログラミング入門	5
2.2.1	クラスの宣言とメソッドの記述	5
	処理手順の基本的な考え方	5
2.2.2	変数	6
	四則演算	6
	変数の型について	6
2.2.3	整数のビット表現とビット演算	8
2.2.4	入出力 (1)	9
	標準出力	9
	標準入力	11
	コマンド引数	13
2.2.5	制御構造	14
	条件判定により処理を選択する方法 「if 文」	15
	フローチャート (流れ図)	15
	条件の記述と boolean 型変数	17
	プログラムの実行の繰り返し (その 1)	18
	再帰代入	19
	プログラムの実行の繰り返し (その 2)	19
2.2.6	入出力 (2): ファイル I/O	20
	ファイルからの入力	21
	エラーによる例外処理	23
	ファイルへの出力	27
2.3	基本的なデータ構造	29
2.3.1	配列	29
	キャストによる型変換	31
2.3.2	スタック, キュー	31
	スタック	31
	キュー	33
2.3.3	リスト	34
2.4	オブジェクト指向プログラミング	35
2.4.1	導入	35
	オブジェクト指向における部品の考え方	35
	オブジェクトプログラミングの概要	35
2.4.2	応用例	40
	標準で提供されているデータ構造について	43

2.4.3	抽象クラスとインターフェースについて	43
3	代表的なアルゴリズム	45
3.1	導入	45
3.2	探索	46
3.2.1	線形探索	46
3.2.2	2分探索	48
	アルゴリズムの計算量についての考え方	50
3.2.3	ハッシュ探索	51
	ハッシュ関数による高速探索の考え方	51
	良いハッシュ関数の実装のために	55
	標準で提供される探索メソッド	56
3.3	再帰的アルゴリズム	56
3.3.1	例 1. 階乗を計算するアルゴリズム	56
3.3.2	例 2. フィボナッチ数を求めるアルゴリズム	58
3.3.3	問題点	59
3.3.4	例 3. ハノイの塔	64
3.4	整列 (ソート)	68
3.4.1	バブルソート (単純交換ソート)	68
3.4.2	挿入ソート	71
3.4.3	シェルソート	73
3.4.4	クイックソート	75
3.4.5	マージソート	79
	標準で提供される整列メソッド	83
3.5	木構造	83
3.5.1	2分木	87
	2分探索木	87
3.5.2	完全2分木	90
3.5.3	ヒープソート	91
3.6	集合の扱い	96
	集合を構成するもの	96
	集合に関する表記	96
	集合に関する演算	97
	集合の表現	97
3.6.1	要素の追加と削除	97
3.6.2	集合の検査	99
3.7	文字列探索 (データ列の探索)	104
3.7.1	単純法 (力任せ法:Brute-force method)	104
3.7.2	KMP法 (Knuth Morris Pratt algorithm)	106
4	GUIアプリケーションの構築	111
4.1	基本的な概念	111
	アプリケーションプログラム構築の概略	112

4.2	使用するツールキット	112
4.3	Swing	113
4.3.1	GUI の構築	113
	GUI 構築の例	113
	ウィンドウの生成	114
	レイアウトマネージャについて	116
	GUI コンポーネントの生成と配置	116
4.3.2	イベント処理	117
	イベントのクラスについて	117
	イベント発生時に起動するメソッド	118
	キーボードイベントからのキーの取り出し	121
4.3.3	JFrame クラスとしてのアプリケーションの構築	121
	より安全な実装	123
4.3.4	Swing の利用に関する補足	124
4.4	SWT	124
4.4.1	概要	124
4.4.2	GUI アプリケーション構築の流れ	124
4.4.3	GUI コンポーネントの生成と配置	126
4.4.4	イベント処理の登録	128
4.4.5	実装例	128
	より安全な実装	133
	キーボードイベントからのキーの取り出し	133
4.4.6	SWT の利用に関する補足	133
4.5	Java アプレット	134
4.5.1	実装例	134
5	Swing によるグラフィックスの作成	137
5.1	概要	137
5.1.1	グラフィックス作成の流れ	137
5.1.2	プログラムの例	138
	paintComponent の引数について	139
5.2	描画のための基本的なメソッド	139
	座標系について	139
	描画色の指定	139
	直線の描画	140
	折れ線の描画	140
	線の属性の指定	140
	長方形の描画 (1) 枠	144
	長方形の描画 (2) 塗りつぶし	144
	楕円の描画 (1) 枠	144
	楕円の描画 (2) 塗りつぶし	144
	多角形の描画 (1) 枠	144
	多角形の描画 (2) 塗りつぶし	144

文字の描画	144
フォントの指定方法	144
システムで利用できるフォントを調べる方法	145
階調色の指定	146
5.3 BufferedImage の利用	147
5.3.1 BufferedImage に対する描画	148
5.3.2 BufferedImage オブジェクトを画像データとして出力する方法	149
5.3.3 画像ファイルの読み込みと表示	152
5.3.4 画素に対する操作	154
6 SWT によるグラフィックスの作成	157
6.1 概要	157
6.1.1 GC (Graphic Context)	157
6.1.2 グラフィックス作成の流れ	157
6.2 描画のための基本的なメソッド	158
座標系について	158
色の指定	158
直線の描画	159
折れ線の描画	159
線の属性の設定	159
長方形の描画 (1) 枠	159
長方形の描画 (2) 塗りつぶし	160
楕円の描画 (1) 枠	160
楕円の描画 (2) 塗りつぶし	160
多角形の描画 (1) 枠	160
多角形の描画 (2) 塗りつぶし	160
文字の描画	160
画素に対する操作	161
6.3 画像ファイルの入出力	164
6.3.1 画像ファイルの読み込み	164
6.3.2 画像ファイルへの出力	164
サンプルプログラム	165
7 スレッドプログラミング	170
7.1 基礎	170
Thread クラス	170
Runnable インターフェース	172
7.2 スレッドの同期	174
7.2.1 他のスレッドの終了を待ち受ける方法	174
join メソッド	174
7.2.2 synchronized 修飾子	175
スレッドの待機 / 実行の再開	179
wait メソッド	179

notifyAll メソッド	179
サンプルプログラム	179
スレッドの状態について	181
A 付録	182
A.1 数学関数	182
A.2 データ構造	183
A.2.1 Collection インターフェース	183
List インターフェース	183
ArrayList クラス	183
LinkedList クラス	184
データの整列 (ソート) について	185
Set インターフェース	187
HashSet クラス	187
TreeSet クラス	187
基本的なメソッド	187
Iterator	187
Map インターフェース	188
HashMap クラス	188
TreeMap クラス	188
基本的なメソッド	188
Iterator	189
A.3 Swing リファレンス	190
多くのコンポーネントで共通のメソッド	190
JTextField 文字列の入出力のためのコンポーネント	190
JPasswordField パスワード入力のためのコンポーネント	190
JLabel 文字列の表示のためのコンポーネント	190
JButton ボタン	190
JCheckBox チェックボックス	190
JRadioButton ラジオボタン	191
JComboBox コンボボックス (選択ボックス)	191
スクロール領域の設置	192
JTextArea テキスト編集領域	192
JSpinner 値を選択するためのコンポーネント (1)	192
JList 値を選択するためのコンポーネント (2)	193
JTable 表	194
JTree 階層構造図	194
JProgressBar 進捗バー	195
JSlider スライダー	196
主な GUI コンポーネントのクラス階層	196
代表的なイベント	197
A.4 SWT リファレンス	198
A.4.1 SWT の入手と利用の方法	198

A.4.2 GUI コンポーネント (コントロール)	199
多くのコントロールで共通のメソッド	199
Text 文字列の入出力のためのコントロール	199
Label 文字列の表示のためのコントロール	199
Button ボタン	199
Combo コンボボックス (選択ボックス)	201
Spinner 値を選択するためのコントロール (1)	202
List 値を選択するためのコントロール (2)	202
Table 表	202
Tree 階層構造図	204
ProgressBar 進捗バー	204
Slider / Scale スライダー / スケール	205
A.4.3 代表的なイベント	205
A.5 開発環境「Eclipse」	206
A.5.1 Eclipse の起動	206