

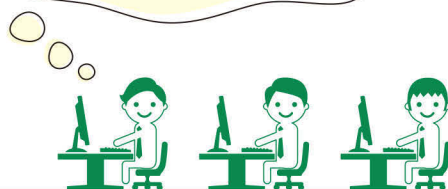
ちゃんと使える力を身につける

Java プログラミング入門

大澤文孝 著

解消

Javaの本、
何冊か読んだけど
やっぱりよくわかりません...



Java言語を しくみから理解できる!

- ✓ 型、クラス、オブジェクトなど、Javaの難しい概念を理解する!
- ✓ Javaプログラミングで間違いがちなポイントをしっかりフォロー!
- ✓ クラスライブラリを使った実践的なプログラミングの仕方を身につける!

ちゃんと使える力を身につける

Java プログラミング入門

大澤文孝 著



■本書のサンプルファイルについて

本書のなかで使用されているサンプルファイルは以下のURLからダウンロードできます。

<https://book.mynavi.jp/supportsite/detail/9784839956745.html>

■サンプルファイルのダウンロードにはインターネット環境が必要です。

■サンプルファイルはすべてお客様自身の責任においてご利用ください。サンプルファイルを使用した結果で発生したいかなる損害や損失、その他いかなる事態についても、弊社および著作権者は一切その責任を負いません。

■サンプルファイルに含まれるデータやプログラム、ファイルはすべて著作物であり、著作権はそれぞれの著作者にあります。本書籍購入者が学習用として個人で閲覧する以外の使用は認められませんので、ご注意ください。営利目的・個人使用にかかわらず、データの複製や再配布を禁じます。

注 意

■本書での説明は、Windows 10、NetBeans 8を使用して行っています。使用している環境やソフトのバージョンが異なると、画面が異なる場合があります。あらかじめご了承ください。

■本書に登場するハードウェアやソフトウェア、ウェブサイトの情報は本書初版第1刷時点でのものです。執筆以降に変更されている可能性があります。

■本書の制作にあたっては正確な記述につとめましたが、著者や出版社のいずれも、本書の内容に関して何らかの保証をするものではなく、内容に関するいかなる運用結果についても一切の責任を負いません。あらかじめご了承ください。

本書中の会社名や商品名は、該当する各社の商標または登録商標です。本書中では™および®は省略させていただいております。

はじめに

Javaは、やはり、難しいプログラミング言語だな。

本書を書いているときに思った、正直な感想です。それは、オブジェクト指向のプログラミング言語であるからです。

オブジェクトというのは、「モノ (Object)」のことですが、分かりやすく言うと「パーツ」です。Javaの文法は、パーツ (Javaでは「クラス (Class)」と呼んでいます) をたくさん作り、それを組み合わせて全体を作ることを前提としています。形の違うパーツを組み合わせることはできないので、パーツの作り方には、規約があります。この規約が、事態を複雑にしています。

プログラミングが初めての人にとって、この規約すべてを理解することは不可能ですし、無意味です。なぜなら、はじめの頃は、既存のパーツを組み合わせて使うことだけで十分で、新しくパーツを作るところまでは至らないからです。

やらないことを習得しても混乱するだけです。そこで本書では、広く浅く全体を説明するのではなく、本当に基礎となる部分、そして、これからの応用につながる部分だけを、できるだけ細部まで説明するよう努めました。本書では、主に次の内容を扱います。実践的なAndroidアプリやWebシステムなどを作る前段階として、十分な基礎知識を身につけられるはずです。

- (1) Javaでの規約、決まりごと
- (2) 既存のパーツの使い方
- (3) 既存のパーツと一体となって自分のプログラムを動かす方法

本書には、素敵で実用的なサンプルが掲載されているわけではありませんが、代わりに、Javaプログラミングに必要なエッセンスを、ふんだんに詰め込んだつもりです。皆様のJavaプログラミングの第一歩の手助けとなれば、幸いです。

CHAPTER	1	Java って何？	13
		この章の内容	14
SECTION	01	いろいろなところで動く Java	15
		習得しやすく安全なプログラミング言語	15
		C 言語や C++ を、もっと簡単・堅牢に	15
		Java で書けば、どのような環境でも動く	17
		OS が違うと、プログラムは動かない	17
		JRE が違いを吸収する	19
		「Write once, run anywhere.」をスローガンに	21
		Java を習得すれば、いろいろなところで動くプログラムが書ける	22
COLUMN		Java とコーヒーとの関係	22
SECTION	02	何を作るのかで異なる作法	23
		パソコン用のプログラムが Android で動くわけではない	23
		API クラスライブラリが違う	23
		作法が違う	25
		Java で作れる代表的なアプリケーション	27
		「言語としての Java」と「作法や API クラスライブラリとしての Java」	31
		端折らずに決まりを理解しよう	32
CHAPTER	2	Java の開発環境を整えよう	33
		この章の内容	34
SECTION	01	Java のプログラムを作るために必要なもの	35
		Java のプログラムを作るときの流れ	35
		JDK の種類	36
		Java のバージョン	38
COLUMN		Java の互換性	38
SECTION	02	統合開発環境を使った開発	39
		JDK での開発と統合開発環境を使った開発の違い	39
		編集してコンパイル、実行までできる統合開発環境	40
		Eclipse と NetBeans	41
COLUMN		Android Studio	41
SECTION	03	統合開発環境「NetBeans」をインストールする	42

	NetBeansをダウンロードする	42
COLUMN	JDKのバージョンの命名規則	43
COLUMN	32ビット版か64ビット版かを調べる	43
	NetBeansをインストールする	44

CHAPTER 3 main メソッドで始めるコンソールアプリ 47

	この章の内容	48
SECTION 01	プロジェクトを新規作成する	49
	プログラムを書くための新規プロジェクトを作ろう	49
	NetBeansのプロジェクトの画面を確認する	52
SECTION 02	コンソールアプリケーションの約束ごと	55
	作られたソースコードの構造を理解する	55
COLUMN	コメントと JavaDoc	56
	コンソールアプリケーションでのお約束	59
COLUMN	決まりごとが多いのは、チーム開発のため	61
SECTION 03	はじめての Java プログラミング	62
	画面に文字を表示するプログラムを書く	62
	実行する	64
COLUMN	コマンドプロンプトから実行する	67
	CHAPTER 4からの解説にあたって	68

CHAPTER 4 Javaプログラムの基礎 (1) ～「APIクラスライブラリ」と「文字列」 71

	この章の内容	72
SECTION 01	プログラムの構成単位となる「クラス」	73
	プログラムをまとめるための実体に相当する「クラス」	73
	プログラムはクラスの内部にメソッドとして記述する	74
	メソッドの定義	75
	文はメソッドの内部に記述する	76
	クラスを作り、メソッドを書くことから始まる	77
COLUMN	クラスやメソッドの命名規則	78
	クラスをとりまとめるパッケージ	78
	パッケージはフォルダ名と合致する	80

	パッケージ名にドメイン名を逆順にしたものを指定する	80
SECTION 02	基本的な機能を提供する「API クラスライブラリ」	82
	基本的な機能を提供するAPI クラスライブラリ	82
	クラスライブラリはパッケージ化されている	83
COLUMN	API クラスライブラリを確認する	87
	System.out.println の意味を調べる	87
	標準入出力	88
COLUMN	リダイレクト	90
	文字を書き出す PrintStream	91
COLUMN	ストリーム	94
	メソッドに値を引き渡す「引数」	95
SECTION 03	文字列の表記方法	96
	ダブルクォーテーションで括って文字列を表現する	96
COLUMN	「文字」と「文字列」	96
	改行など特殊文字	97
	文字列中にダブルクォーテーションを含む場合	98
	円記号自体	99
	文字列の連結	100
CHAPTER 5	Java プログラムの基礎 (2) ～「変数」と「オブジェクト」	101
	この章の内容	102
SECTION 01	値を一時的に保存する変数	104
	値の種類を示す「型」	104
	変数とは	105
	変数の宣言	105
	変数の宣言では型を決める	105
	変数を宣言する	105
	変数を使ってみる	108
	変数の宣言	109
COLUMN	メソッドの外で宣言する	110
	変数への値の代入	111
	変数の参照	113
COLUMN	変数の宣言と同時に代入する	114

SECTION 02	Stringオブジェクトを理解する	115
	オブジェクトはクラスをメモリに展開したもの	115
	String型の「型」とはクラスのこと	115
	クラスからオブジェクトを作る	117
	オブジェクトは「データ」と「メソッド」が関連づけられたもの	118
	メソッドの呼び出しと戻り値	119
	String オブジェクトのメソッドを呼び出す例	120
COLUMN	文字列のメソッドを直接呼び出す	121
	String オブジェクトのメソッドを呼び出してもオブジェクトは変わらない	122
	引数を伴ったメソッドの呼び出し	123
	クラスとオブジェクトとの違い	126
	Java はオブジェクト指向のプログラミング言語	127
SECTION 03	数値の計算とプリミティブ型	128
	四則演算する	128
COLUMN	桁数が大きな数字を見やすく書く	129
	数値の表現に使う「型」	130
COLUMN	float 型と long 型の数値表現	131
	整数を使った計算の例	132
	プリミティブ型はオブジェクトではない	133
	小数以下の値を扱うときの注意	134
	double 型にキャストする	135
	数値を直接表記する場合も注意	137
COLUMN	float 型や double 型で金銭の計算をしてはいけない	138
SECTION 04	文字列と数値とを相互に変換する	139
	数値を文字列に変換する	139
	キャストでは変換できない	140
	String.valueOf メソッドで変換する	140
	インスタンスメソッドと静的メソッドの見分け方	142
	インスタンスメソッドと静的メソッドとの違い	143
	文字列を数値に変換する	145
	変換方法のまとめ	146
SECTION 05	APIを呼び出してみよう	147
	メッセージボックスを表示してみる	147
	showMessageDialog メソッドの書式を調べる	147

	パッケージ名を省略する import	150
	import 文の書き方	151
COLUMN	NetBeans で import 宣言を追加する	152
	API クラスライブラリの書式に合わせてプログラムを書く	152
CHAPTER 6	Java プログラムの基礎 (3) ～条件分岐の基本	153
	この章の内容	154
SECTION 01	条件判定の基本	155
	ユーザーが文字列入力できるようにする	155
COLUMN	入力文字の無害化 (サニタイズ)	156
	showInputDialog メソッドで1行分の文字入力を受け付ける	156
	showMessageDialog メソッドで文字列を表示する	158
	未入力のときにエラーメッセージを表示したい	158
	条件分岐の基本	159
	「真」と「偽」と boolean 型	160
	条件分岐し、異なるメッセージを表示する	161
	比較のための演算子	161
	null と等しいかどうかを判定する	162
SECTION 02	文字列を比較する	164
	未入力かどうかをチェックする	164
	文字列は equals メソッドで比較する	165
	「== 演算子」は、オブジェクトの合致を示す	166
COLUMN	NetBeans で変数の中身を調べる	167
	内容が等しいかどうかを調べたいときは equals メソッドを使う	168
COLUMN	[取消] ボタンをクリックするとエラーになる	170
SECTION 03	条件を組み合わせる	171
	条件を入れ子にする	171
	if 構文のブロックのなかに if 構文を入れる	171
	条件を組み合わせる	173
	「または」で表現する	174
COLUMN	条件の順序を変えた場合	176
	条件を羅列する	176
COLUMN	NetBeans による、より良いプログラムへの変更の提案	178

CHAPTER	7	Javaプログラムの基礎 (4) ～さまざまな比較とループ処理	179
		この章の内容	180
SECTION	01	特定の書式にあてはまるか比較する	182
		特定の書式に合致するかどうかを調べられる正規表現	182
		郵便番号を調査する例	183
		正規表現のパターン	183
		正規表現と合致するかどうかを調べる	186
SECTION	02	数値を比較する	187
		数値が特定の範囲かどうかを調べる	187
		数値以外を入力したとき	189
		1. 数値に変換できるかどうかを前もって調べる	191
		2. 例外を捕らえる	192
SECTION	03	switch構文を使って複数に分岐する	195
		入力された数字の値に応じて分岐する	195
		変数を活用して短く書く	196
		たくさんの値を一度に比較する「switch構文」	198
SECTION	04	繰り返し処理する① 条件を満たしている間、繰り返す	201
		文字入力されるまで繰り返し入力を求める	201
		条件を満たしている間、繰り返す「while構文」	202
		文字入力がないときに繰り返す	203
COLUMN		変数のスコープに注意	204
		breakで繰り返し処理をやめる	205
		[取消] ボタンで中断するようにする	206
COLUMN		returnでメソッドから直接戻る	208
SECTION	05	繰り返し処理する② 指定回数だけ繰り返す	209
		指定回数の繰り返しは、「実行された数を数える」のが基本	209
		変数を使ってカウントする	210
COLUMN		数を数える変数には「i」「j」や「n」「m」などが使われる	212
		指定された回数を繰り返すのに便利なfor構文	212
		「1を増やす」の略記	213
COLUMN		任意の値を四則演算する	214
COLUMN		「i++」と「++i」の違い	215
		for構文を使うと、指定回数の繰り返しが簡単	215

	繰り返し処理はプログラムの基本	216
CHAPTER 8	配列とコレクション	217
	この章の内容	218
SECTION 01	連続するデータをまとめて配列に保存する	220
	1つの変数に1つの値しか保存できない場合の問題点	220
	5日間の気温を保存するには、5つの変数が必要	220
	繰り返し処理もできない	221
	1つの変数で複数の値を保存できる配列	222
	配列の確保	222
	配列の要素への代入や参照	223
	配列をループ処理する	224
	for ループですべての要素を処理する	225
	初期化子を使って、代入を簡単に記述する	227
	初期化子で配列を確保する	227
	要素の数を length で知る	228
	値の平均を求めている	229
COLUMN	オーバーフローに注意	230
SECTION 02	後から要素の数を変更できる ArrayList オブジェクト	231
	要素を増減できる ArrayList オブジェクト	231
	ArrayList オブジェクトを作る	232
COLUMN	ジェネリクスで型を強要する	233
	値を追加する	234
	値を取得する	235
	値を変更する	236
	途中に挿入したり削除したりする	237
	要素を検索する	238
	ArrayList は配列ではない	239
	保存できるのはオブジェクトのみ	240
	ユーザーに数を入力してもらい、最後に、その平均を求める例	241
SECTION 03	さまざまなコレクションと列挙操作	245
	複数の値をまとめて保持するコレクション	245
	「リスト」と「セット」	246

	コレクションの共通機能を実現する Collection インターフェイス	246
	HastSet を使った例	250
	for-each ループで列挙操作する	252
COLUMN	配列で for-each 構文を使う	255
	Iterator を用いて列挙する	255
SECTION 04	キーと値を保持するマップ	258
	キーに対して値を設定するマップ	258
	マップを使った例	259
	HashMap の生成と値の追加	261
COLUMN	変数名を書かずに略記する	262
	キーが登録されているかどうかの確認と値の取得	262
	キーや値の列挙	263
	値の一覧を取得する	264
	キーの一覧を取得する	265
COLUMN	キーと値を両方列挙する	266
	マップを使うと長い if 文から逃れられる	267
SECTION 05	配列やコレクションのコピー	268
	代入では元の配列が変わってしまう	268
	参照ではなくコピーしたいなら clone メソッドを使う	270
	シャローコピーとディープコピー	271
	オブジェクトの代入には要注意	272
CHAPTER 9	クラス、インターフェイス、そして、実用的な Java プログラミングに向けて	273
	この章の内容	274
SECTION 01	クラスとのつきあい方	276
	クラスを作れなくても大丈夫	276
	Java プログラミングで必要なクラスの知識	277
SECTION 02	クラスを改良するための継承とインターフェイス	278
	クラスを改良するための「継承」	278
	基のクラスと違うところだけを書く	279
COLUMN	@Override 属性	280
	自身を示す「this」と基底クラスを示す「super」	281
	機能が実装されていることを保障する「インターフェイス」	282

	すべてのメソッドをオーバーライドする	282
	たくさんのインターフェイスを実装できる	283
SECTION 03	クラスの継承やインターフェイス実装の実際	285
	定期的に実行する例	285
	Timer クラスで定期的に実行する仕組み	285
	TimerTask クラスから継承したクラスを作る	287
	10 秒ごとに時刻を表示する例	288
COLUMN	日付の書式を変更する	289
	ArrayList オブジェクトを並べ替える例	290
	基本的な並べ替え	290
	逆順に並べ替える	291
	逆順に並べ替えるサンプル	292
	Comparator インターフェイスの compare メソッドで比較する	294
	相手の都合に合わせるように書く	296
COLUMN	その場でクラスを作る無名クラス	298
SECTION 04	Android アプリにおける Java	299
	Android 固有の機能を提供するライブラリ	299
	自動生成される雛形を改良する	300
	Activity オブジェクトとイベント	302
	イベントはインターフェイスで通知される	303
COLUMN	ボタンがタップされたときの処理をする実際のプログラム	304
	Android システムから呼び出される	304
	システムが何を要求しているのかを知る	305
COLUMN	並列実行するためのスレッド処理	306
	索引	307



Java って何？

CHAPTER

1

Javaは、デスクトップアプリケーションやAndroidアプリの開発、Webシステムの構築、そして、小さな組み込みハードウェアに至るまで、幅広く利用できるプログラミング言語です。

この章では、Javaの特徴やメリット、そして、Javaを学習するときに注目したいポイントを説明します。



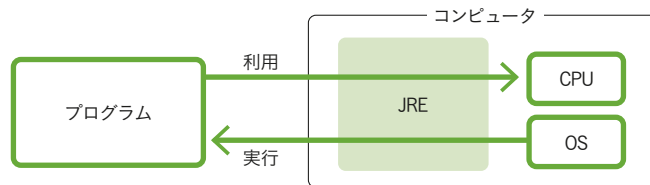
この章の内容

SECTION

01 いろいろなところで動く Java

Javaで作ったプログラムは、WindowsやOS X、Androidなど、さまざまな環境で動きます。これを実現しているのが、「JRE (Java Runtime Engine)」と呼ばれる仮想的な実行環境です。

JREはCPUやOSの違いを吸収するもので、Windows用やOS X用、Android用などさまざまな環境に対応したものが用意されています。



SECTION

02 何を作るのかで異なる作法

JREは、CPUやOSの差を吸収しますが、Windows用のプログラムがAndroidでそのまま動くわけではありません。なぜなら、Windows用のJREとAndroid用のJREとでは、実装されている機能の差があるため、そのまま動くのは、共通の機能しか使わない、ごく一部の簡単なプログラムに限られます。

また、Webのシステムとして作るのか、Androidアプリとして作るのかなど、どのように実行するかによっても、作り方が異なります。

Javaで作れるアプリケーションの種類は、主に以下の6つです。

- コンソールアプリケーション (本書で扱うもの)
- GUIアプリケーション
- Web システム
- 組み込みプログラミング
- Java アプレット
- Android アプリ



いろいろなところで動く Java

SECTION

01



Javaは、汎用的なプログラミング言語です。OSの違いを気にすることなく、パソコンやスマートフォン、Webシステムに至るまで、いろいろなところで動くプログラムを書けます。



習得しやすく安全なプログラミング言語

Javaは、1990年代に、サン・マイクロシステムズ社（現在はオラクル社に吸収合併）のJames Arthur Gosling（ジェームズ・アーサー・ゴスリン）氏、Bill Joy（ビル・ジョイ）氏らが中心となって考案されたプログラミング言語です。

現在では、オープンソース化されており、言語仕様の拡張や改訂は、「JCP（Java Community Process）」という標準化プロセスで策定されています。



C言語やC++を、もっと簡単・堅牢に

Javaが考案された当時、主流のプログラミング言語は、「C言語」と、それにオブジェクト指向という概念を取り入れた「C++（シーplusplusまたはシーplusplus）」でした。

オブジェクト指向とは、データとプログラムを、ひとまとめに
して管理するプログラミングの技法です。詳細は、
CHAPTER 5で説明します。



C言語やC++は自由度が高く、細かい制御まで処理できる反面、少しでも間違ったプログラムを記述すると、大きく誤動作する心配がありました。ひとたび誤動作すると、エラーが表示されることなく強制終了したり、まったく応答しなくなったり、壊れたかのように異常な挙動を繰り返してしまうことも、珍しくありません。

また、C言語はともかくとして、C++は、言語仕様がとても複雑怪奇で習得が難しいのも難点でした。

Javaでは、次の方法で、これらの問題を解決しました。

プログラムが応答しなくなるなど、まったく制御不能になってしまうことを「暴走」と言います。

①言語仕様の簡素化

Javaでは、C言語やC++と似た文法をとりつつも、**言語仕様の簡素化**を目指しました。その結果、習得しやすくなりました。

②メモリの状態の監視

プログラムが異常な動きをする大半の問題は、コンピュータのメモリに対する不正な読み書きです。

コンピュータのメモリには、処理途中のさまざまなデータが置かれています。何らかの理由で、データがまだ入っていないメモリを読み込もうとしたり、メモリ上のデータを特定する位置がズレたりしたときに、強制終了や無応答の問題が起きます。

そこでJavaには、メモリの状態を管理する機能が内蔵されました。メモリ管理機能によって、正しくないメモリの場所を参照したときには、そこで処理が停止し、エラーが表示されるようになりました。

この機構によって、誤動作する理由をつかみやすくなり、より**堅牢なプログラム**を作れるようになりました。



C言語やC++では、使い終わったメモリをOSに戻すことを忘れたためにメモリが足りなくなるとい現象が、しばしば起きます。
しかしJavaでは、ガベージコレクション（Garbage collection。略してGC。ごみ集めのこと）という機能によって、使い終わったメモリが自動的にOSに戻されるので、戻し忘れの心配もありません。

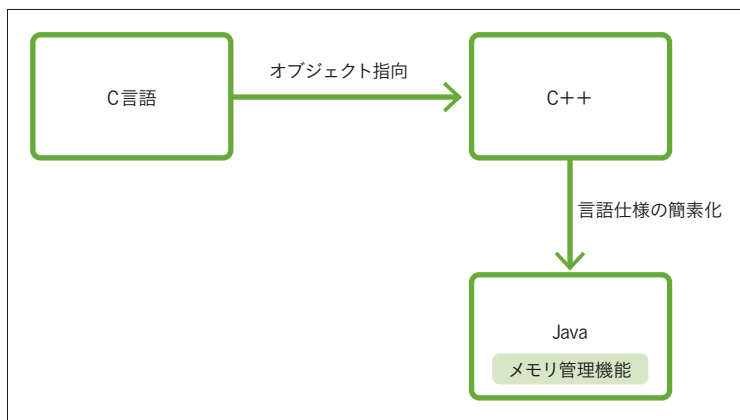


図 1-1-1 JavaはC言語やC++のいいとこ取りのプログラミング言語



Javaで書けば、どのような環境でも動く

実は、「言語仕様の簡素化」と「メモリの状態の監視」は、Javaにとって、些細なことです。

Javaが生まれた背景には、もっと壮大な目標がありました。それは、「作ったプログラムを、どのようなコンピュータでも動かせるようにすること」です。



OSが違くと、プログラムは動かない

一般に、特定のOS用として作ったプログラム（ソフトウェア、アプリケーション）は、別のOSでは動きません。

たとえば、「Windows用のプログラム」はMac（OS X）では動きませんし、「OS X用のプログラム」はWindowsパソコンでは動きません。

ましてや、iOSやAndroidなどのスマートフォンのプログラム（スマートフォンアプリ）をパソコンで動かすことはできませんし、逆も然りです。動かない理由は、大きく2つあります。

①CPUが違う

プログラムを実行するのは、パソコンやスマートフォンに搭載された「CPU（central processing unit）」です。CPUによって、命令の種類や形

式が異なります。

たとえば、パソコンではインテル社製（もしくはその互換品）のCPUが多く使われていますが、スマートフォンではARM社製やApple社製のCPUなどが使われています。

そのため、一度、パソコン用のプログラムとして作ってしまうと、それをスマートフォンなどの他のCPUが搭載されたコンピュータでは動かせないのです。

②OSによって制御の方法が異なる

OSには、「ウィンドウを表示する」「マウス操作する」「キーボード操作する」「ファイルを読み書きする」「音を鳴らす」など、ありとあらゆるものを操作する小さなプログラムが収録されています。これを「**API (Application Programming Interface)**」と言います。

プログラムは、このAPIの機能を用いることで、さまざまな処理を実行します。

搭載されているAPIの種類や利用方法は、OSによって異なります。そのため、Windows用プログラムはOS Xでは動かない（逆も同じ）のです。

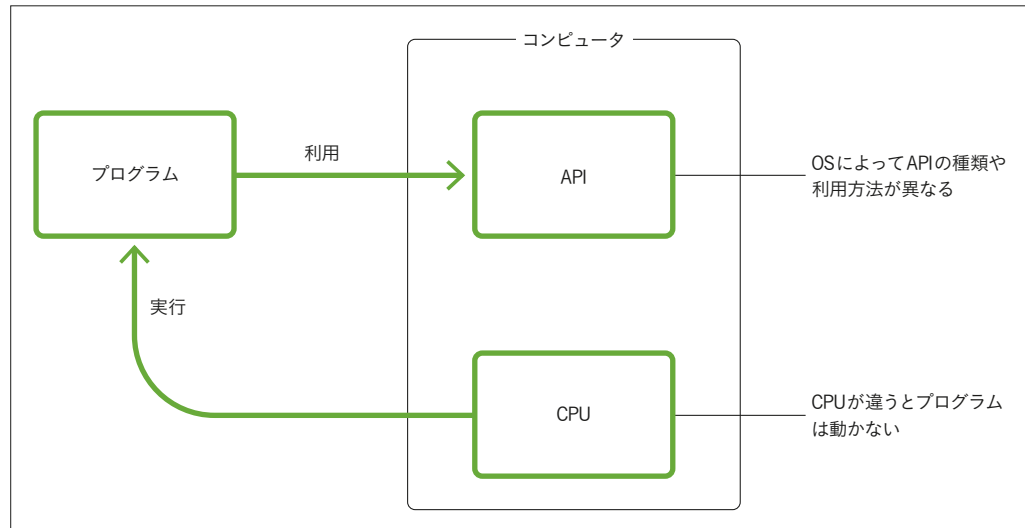


図 1-1-2 CPUやOSが違くと動かない



JREが違いを吸収する

搭載されているCPUやOSはパソコンやスマートフォンによって異なるため、同じプログラムが動かないのは、当たり前のことです。

しかし、この当たり前のことを、Javaは巧妙な手口で解決しました。それは、CPUやOSに依存しない、仮想的な実行環境を導入するというものです。この仮想的な実行環境のことを「**JRE** (Java Runtime Engine)」といいます。

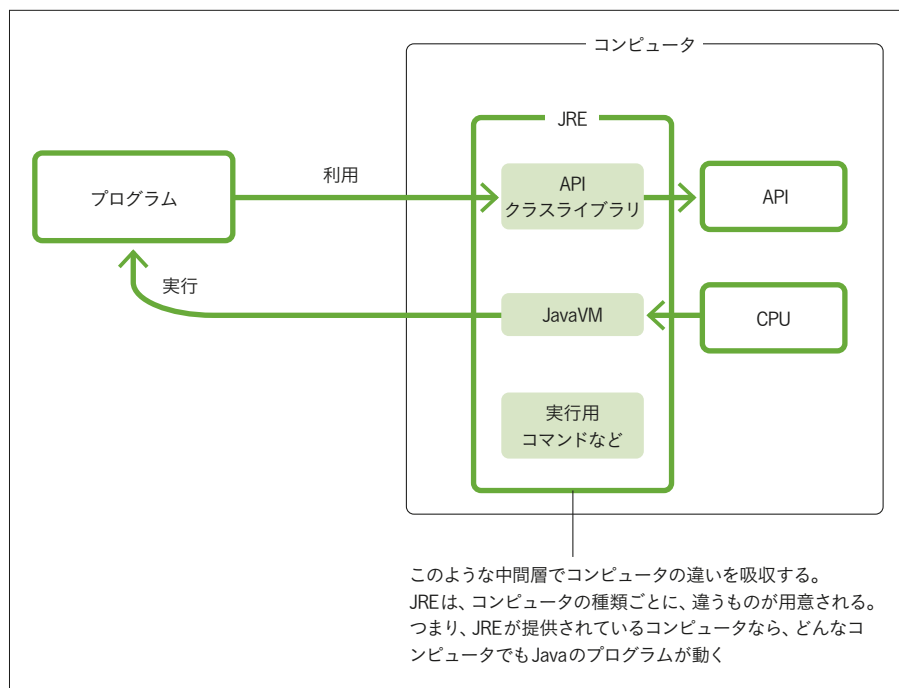


図1-1-3 JREの役割

JREは、「**JavaVM** (Java Virtual Machine。Java仮想マシン)」と「**API クラスライブラリ** (API Class Library)」、そして、Javaのプログラムを実行するためのいくつかのコマンドが含まれたパッケージです。

「Windows用」「OS X用」「Linux用」「スマートフォン用」など、コンピュータの種類に応じて、異なるJREが提供されます (CHAPTER 2で説明するように、JREは無償で入手できます)。

①CPUの違いを吸収する JavaVM

JavaVMは、仮想的なコンピュータです。Javaのプログラムは、「**Java バイトコード**」（「中間コード」と呼ばれることもある）と呼ばれる、CPUに依存しない汎用的なプログラムとして作られます。

この汎用的なプログラムを解釈して、実際のCPUで実行できるようにするのが、JavaVMの役割です。

たとえば、WindowsやOS Xなどインテル社CPU向けのJavaVMなら、Javaバイトコードをインテル社CPUの命令に変換します。そしてAndroidなど、ARM社のCPU向けのJREなら、ARM社のCPUの命令に変換するのです。

このような動作によって、CPUの違いを吸収します。

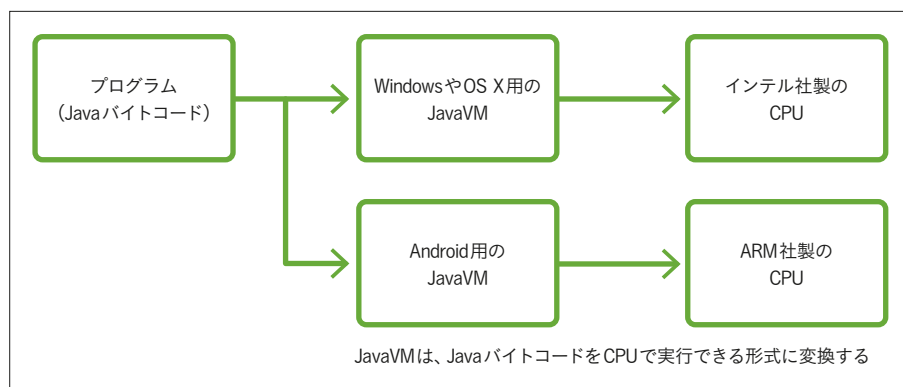


図 1-1-4 JavaVMの役割

②OSの違いを吸収する API クラスライブラリ

API クラスライブラリは、OSで提供される「ウィンドウを表示する」「マウス操作する」「キーボード操作する」「ファイルを読み書きする」「音を鳴らす」などの機能をJavaで使える形で用意したものです。

Javaのプログラムでは、たとえば、ウィンドウを表示するのに、OSのAPIを使うのではなく、API クラスライブラリとして提供されているウィンドウ表示機能を利用します。

すると、JREによって実際のOSのAPIの機能を実行するように変換されます。つまり、Windows上で実行されるときにはWindowsのAPIが、Mac上で実行されるときにはOS XのAPIが呼び出されるように調整されます。

このようにAPI クラスライブラリは、**OSの違いを吸収します**。

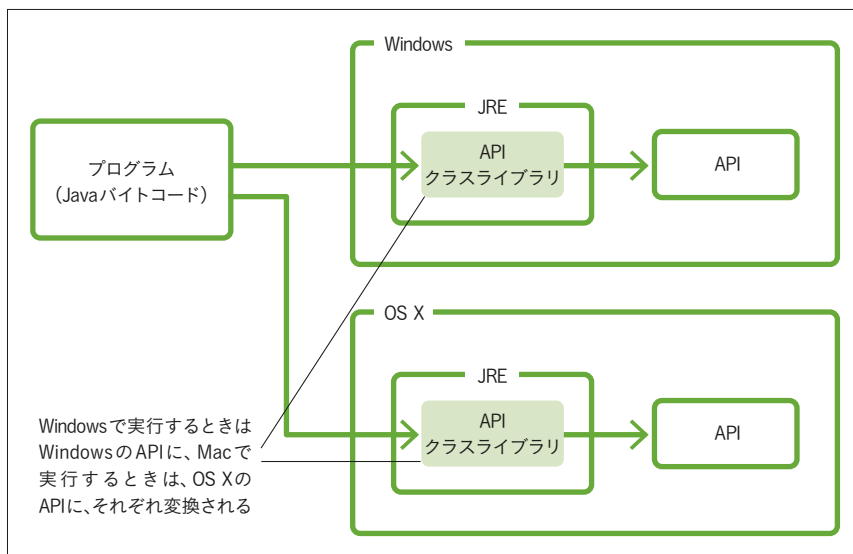


図 1-1-5 APIクラスライブラリの役割

APIクラスライブラリは、省略して「API」と呼ばれることもあります。しかし、その表記だと、OSのAPIと区別が付かないので、本書では、「APIクラスライブラリ」と表記します。



「Write once, run anywhere.」をスローガンに

ここまで説明してきたように、Javaのプログラムは、JREさえあれば、どのようなコンピュータ上でも動くのが大きな特徴です。

サン・マイクロシステムズ社は、これを「Write once, run anywhere.」（一度プログラムを書けば、どこでも動く）というスローガンで表現し、Javaを一躍有名にしました。

実は、「Write once, run anywhere.」は少し誇張した表現で、実現できるのは一部のプログラムに限られます。詳しくは次節で説明します。





Javaを習得すれば、 いろいろなところで動くプログラムが書ける

コンピュータのプログラムを作るためのプログラミング言語は、たくさんあります。

それらのなかで、Javaが人気を保っているのには、主に、2つの理由があります。

①いろいろなところで動くプログラムが書ける

JREがCPUやOSの違いを吸収するため、いろいろなところで動くプログラムを書けます。

ひとたび習得すれば、パソコン用のデスクトップアプリケーションから、Webシステム、そして、Androidアプリに至るまで、さまざまなプログラムを作れます。

②Javaから発展して他のプログラミング言語を習得するのが容易

Javaは、文法がC言語やC++に似ています。そして、文法に癖がないのも特徴です。Javaを習得すれば、それを足がかりにして、他のプログラミング言語を習得するのが容易です。



COLUMN

Javaとコーヒーとの関係

Javaという言語名の由来は諸説ありますが、「ジャワコーヒー（Java Coffee）」に関係しているという話が有力で、コーヒーと深い関係があります。実際、Javaのアイコンは、コーヒーをあしらった模様です。

そしてJavaバイトコードの先頭は、いつも「CAFEBABE」（Cafe Babe）という文字から始まっています（Javaバイトコードは、バイナリエディタと呼ばれる特殊なソフトを使うと、中身を覗けます）。



「CA FE BA BE」
（Cafe Babe）で始まる

```

BZ (Admin) - Example01.class
ファイル(E) 編集(E) 表示(V) 移動(D) ツール(T) ヘルプ(H)
C:\Users\Yosawa\Documents\NetBeansProjects\Example01\build\classes\example01\Example01.class 518
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F 012345678
0 CA FE BA BE 00 00 00 34-00 1C 0A 00 05 00 13 08 .. ....4.
10 00 14 0A 00 15 00 16 07-00 17 07 00 18 01 00 06 .....
20 3C 69 6E 69 74 3E 01 00-03 28 29 56 01 00 04 43 <init>...
30 6F 64 65 01 00 0F 4C 69-6E 65 4E 75 6D 62 65 72 ode...Lin
40 54 01 00 00 0F 01 00 10 40 0F 00 01 00 00 01 70 T.L.L

```




何を作るのかで異なる作法

Javaは、JREの存在によって、OSやCPUの種類を問わず、さまざまなプログラムを作れます。しかし、どのプログラムでも同じように作るわけではありません。文法は同じものの、何を作るかによって、その書き方が大きく異なります。



パソコン用のプログラムがAndroidで動くわけではない

P.021で紹介した「Write once, run anywhere.」というスローガンは、印象的で魅力的です。しかし少し、言葉足らずで誤解を招く言葉です。

この言葉は、あたかも、「Javaを使ってパソコン向けにプログラムを書けば、それはAndroidでも動く」というように聞こえます。しかし、それは間違いです。



APIクラスライブラリが違う

間違いの理由のひとつは、APIクラスライブラリにあります。

「Windows用のJRE」と「OS X用のJRE」は、どちらも同じ機能をもつAPIクラスライブラリを備えています。ウィンドウ操作やマウス操作などは、同じ方法で処理できます。そのため、Windows向けに作ったJavaのプログラムは、OS Xでも動きます。

しかし、Android用のJREは、WindowsやOS XのJREと、機能が大きく異なります。

Android用のJREには、「Android向けに特別に拡張されたAPIクラスライブラリ」が搭載されています。逆に、パソコン用のウィンドウ操作やマウス操作などをする一部のAPIクラスライブラリが、搭載されていません。

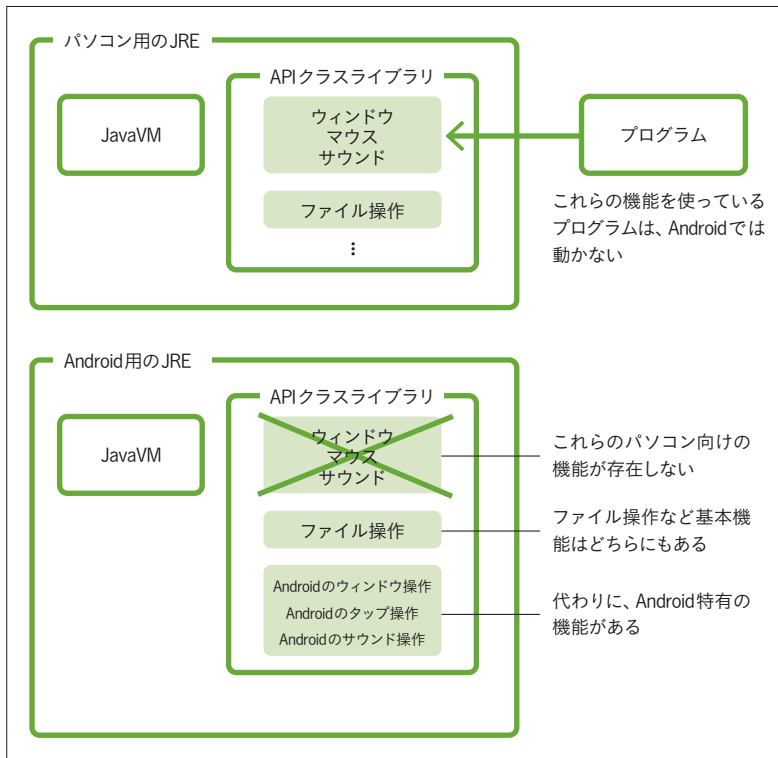


図1-2-1
パソコン用のJREとAndroid用のJREとで搭載されているAPIクラスライブラリが異なる

そのためパソコン向けに作ったJavaプログラムは、Androidでは動きません。逆も然りです。このようにJavaでは、JREに含まれているAPIクラスライブラリの存在も重要です。

「Write once, run anywhere.」が実現するのは、対象となる環境に「同じAPIクラスライブラリが提供されている」のが前提です。

図1-2-1のように、搭載されているAPIクラスライブラリが異なるときは、そのままでは動きません。

MEMO

原理的に言えば、AndroidにパソコンのJREと同じAPIクラスライブラリが搭載されれば、パソコン向けに作られたJavaプログラムが、そのまま動く可能性があります。



作法が違う

では、同じクラスライブラリが提供されるなら実行できるのかというと、そういうわけでもありません。

すぐあとに説明するように、Javaでは、デスクトップアプリケーションやWebシステム、Androidアプリなど、さまざまな種類のプログラムを作れますが、どのような種類のものを作るのかによって、書き方の作法や実行方法が異なります。

Javaで作ったプログラムを実行するには、次に挙げるいずれかの方法をとります。

① java コマンドで実行する方法

ひとつめの方法は、JREに含まれる「java」コマンド（Windowsの場合はjava.exe）というコマンドで実行する方法です。

このコマンドは、Javaで書かれたプログラム（Javaバイトコード）を読み込み、それをJREに渡すことでプログラムを実行します。

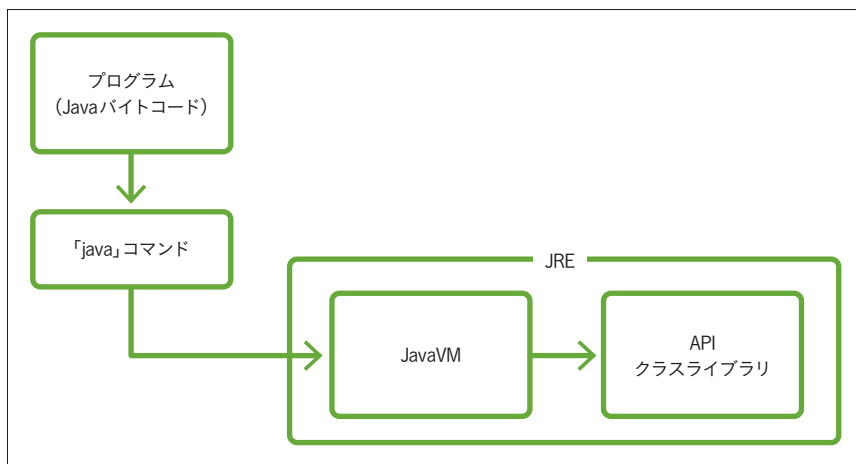


図 1-2-2 「java」コマンドによるプログラムの実行

②他のプログラムから読み込まれて実行される方法

もうひとつの方法は、作ったプログラムを決められた場所に置いておく、必要に応じて読み込まれて実行される方法です。

たとえば、JavaでWebシステムを作るときには、ユーザーがWeb ブラ

ウザなどで操作すると、接続先のWebサーバ上で、あらかじめ配置しておいたプログラムが必要に応じて読み込まれて実行される仕組みになっています。

Androidの場合も同様で、インストールしたJavaのプログラム(Androidアプリ)は、ユーザーがメニュー操作などで起動すると、読み込まれて実行されます。

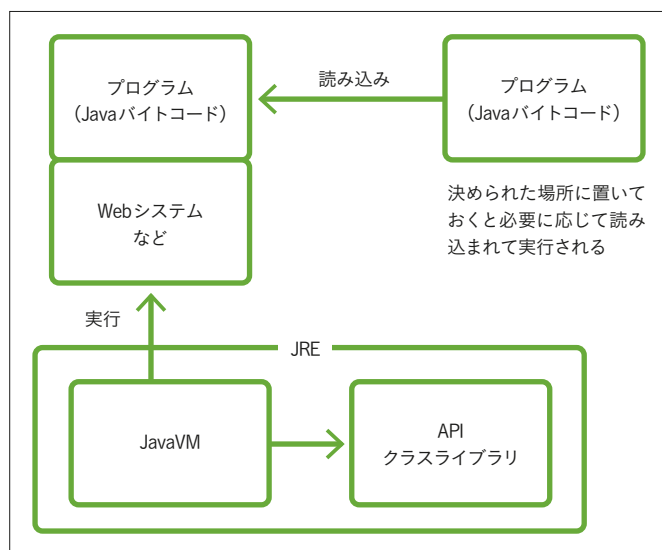


図 1-2-3 読み込まれて実行される方法

どちらの場合も、書き方の作法があります。

たとえば、①の「java」コマンドで実行するプログラムは、最初に実行する箇所は、「main」と命名しなければならないという決まりがあります。

②の場合も同様に、決まりがあります。決まりは、WebシステムやAndroidアプリなど、実行環境によって異なりますが、いずれにしても、mainと書かれた場所から処理が始まるという決まりはありません。そのため、①の方式を前提として作られたプログラムは、実行できません。

Javaのプログラムというのは、図1-2-4に示すように、「何を作るのか」によって、土台の形が決まっており、**その土台に収まる形式で作らなければなりません。**

どれもJavaで作られたプログラムですが、土台の形が違うところでは、動かせないのです。

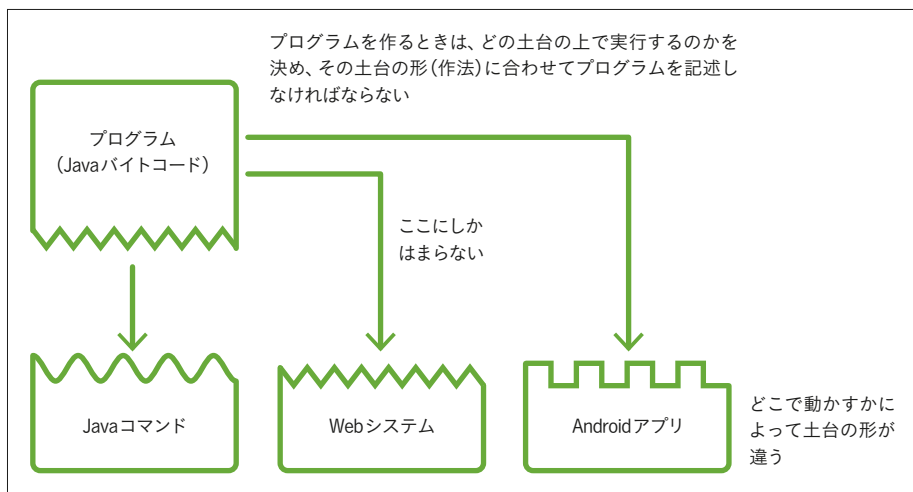


図 1-2-4 Javaのプログラムは土台の形に合わせる



Javaで作れる代表的なアプリケーション

では、「土台」に相当する、アプリケーションの種類には、どのようなものがあるのでしょうか？

細かいものまで入れるとキリがありませんが、主なものは、次の6種類です。

① コンソールアプリケーション

Windowsの「コマンドプロンプト (command prompt)」やMacの「ターミナル (terminal)」、Linuxの「シェル (shell)」などのように、テキストとキーボード操作だけでやりとりするアプリケーションのことを「**コンソールアプリケーション** (Console Application)」といいます。

コンソールアプリケーションは、もっともシンプルな形態のプログラムで、Javaの言語文法などを習得するのに適しています。

本書では、このコンソールアプリケーションを使って、Javaの基本を説明していきます。



MEMO

本書ではコンソールアプリケーションを扱いますが、Net Beansというソフトを使うため、コマンドプロンプトやターミナルそのものはほとんど使いません。



```
C:\Users\osawa\Documents\NetBeansProjects\Example12\build\classes>java example12
Example12
Hello Java
C:\Users\osawa\Documents\NetBeansProjects\Example12\build\classes>
```

図 1-2-5 コンソールアプリケーションの例

② GUI アプリケーション (AWT、Swing)

ウィンドウやグラフィックなどを表示するデスクトップアプリケーションです。Windows パソコンや OS X、Linux などで行えます。

GUI アプリケーションには、「AWT (Abstract Windowing Tools)」や「Swing」、「SWT (Standard Widget Toolkit)」という API クラスライブラリが使われます。



図 1-2-6 GUI アプリケーションの例

③ Java アプレット

Java アプレット (Java Applet) は、Web ブラウザの内部で実行されるプログラムです。

HTML のなかに領域を作り、そのなかに、GUI アプリケーションと同様に、ボタンやテキストボックスなどの各種コントロールを配置したり、グラフィックなどを描画したりできます。

機能は、②の GUI アプリケーションと似ていますが、ファイルの読み書きやパソコンの設定変更などができないように制限されています。

なぜなら、Java アプレットは、Web ブラウザの内部でユーザーの許可なしに実行できるため、それらの機能を許可してしまうと、悪意ある Java アプレットが勝手にデータを書き換えたり、盗んだりできてしまう恐れがあるからです。

なお、最近のブラウザは Java アプレットに対応しないものもあります。たとえば、Windows 10 に付属の「Microsoft Edge」というブラウザや 2015 年 9 月以降にリリースされた「Chrome ブラウザ（バージョン 45 以降）」では、Java アプレットは動きません。

Web ブラウザで動くプログラミング言語として「JavaScript」がありますが、「JavaScript」と「Java」とは、名前が似ているだけで、まったく別のプログラミング言語です。類似性は、一切ありません。

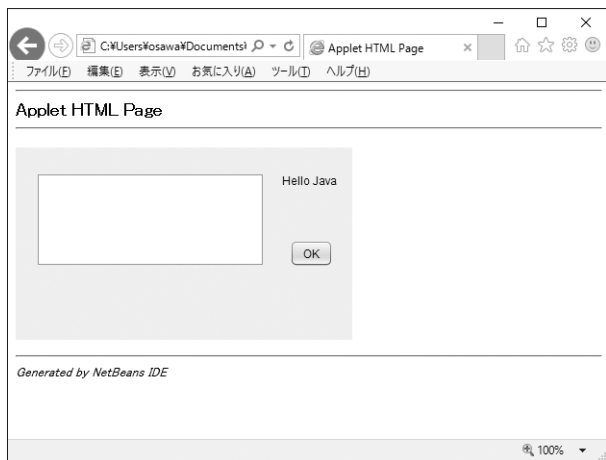


図 1-2-7 Web アプレットの例

④ Web システム

Web サーバ側で実行する Java です。これは「サーブレット (Servlet)」と呼ばれます。

プログラムは、Web サーバ側に配置しておき、ユーザーが Web ブラウザなどでアクセスしてきたときに、そのプログラムが実行されます。CGI (Common Gateway Interface) と似た仕組みで動きます。

Web システムは、扱う範囲がとても広く、実行環境の準備も複雑なので、それだけで 1 冊の本ができるほどのトピックです。本書では扱いません。



試し読みはお楽しみ
いただけましたか？

ここからはManatee
おすすめの商品を
ご紹介します。

Manatee Tech Book Zone 

3.6
2017

1

おすすめ
電子書籍

2

Manatee

体験型の本書でプログラミングの第1歩を踏みだそう!

『やさしくはじめるiPhoneアプリ作りの教科書 [Swift 3& Xcode 8.2対応]』は、iPhoneアプリを作ってみたく初心者のための入門書です。プログラミングが初めての人、苦手意識がある人でも楽しく学んでいけるよう、なるべくやさしく、イラストや図をたくさん使って解説しています。本書では実際にサンプルアプリを作りながら学んでいきますが、イラストによる解説で、一歩ずつ丁寧に、iPhoneアプリ作りの基本と楽しさを学べます。

イラストによる解説で、プログラミングはじめての人でも学べる

Chapter 1-3 アプリ開発の基本的な流れは?

アプリは5つの手順で作る

さて、iPhoneアプリはどのように作っていくのでしょうか? 最初はSwiftというプログラミング言語を学ばなければ、アプリを開発することはできません。Swiftを学んでからプログラミングを始めると、最初はSwiftというプログラミング言語を学んでから、実際にアプリを開発することになります。Swiftを学んでからプログラミングを始めると、最初はSwiftというプログラミング言語を学んでから、実際にアプリを開発することになります。

1. プロジェクトを作る
2. アプリの画面を作る
3. 部品とプログラムをつなぐ
4. プログラムを書く
5. シミュレーターでテストする

「プログラミング」

豊富なイラストで「なぜ?」を解消! Javaの第一歩を踏み出そう

『スッキリわかるJava入門 第2版』は、Javaの基礎から初学者には難しいとされるオブジェクト指向まで、膨らむ疑問にしっかり対応しました。Javaプログラミングの「なぜ?」がわかる解説と約300点の豊富なイラストで、楽しく・詳しく・スッキリとマスターできる構成となっています。「なんとなくJavaを使っているけれど、オブジェクト指向の理解には自信がない」「学習の途中で挫折してしまった」という方にもおすすめです。

0.1 ようこそ Java の世界へ

0.1.1 Java を使っていること

Java とはプログラムを作るために利用するプログラミング言語の一つです。Java を使えば、さまざまなコンピュータで動作する多様なプログラムを開発することができます。



図 0.1 Java を使っていること

このような特徴から、Java はさまざまな分野で利用されています。

- 学習が容易な基本文法
- 大規模開発を支援するオブジェクト指向の仕組み
- 意図に準拠された静的型検査
- 多様なコンピュータ環境で動作する汎用性

このような特徴を持つ Java を自由自在に用いて、プログラムを開発できるようになりたいという人々のために、この本は書かれました。初めてプログラミング言語に触れるという方にも Java の基本を学ぶための「Java プログラミング入門」に書かれています。ステップアップを図るために、読み進められるようになっています。ぜひこの「手始め」のステップを踏んで、Java をマスターしていきましょう。

「書くのは、読むよりも簡単ではないから...」
「それは間違いない! だが!」の様に「記憶がないから...?」

書くのが苦手な人でも、読むのが苦手な人でも、この本が役に立ちます。

実行結果

こんにちは! (1)

0.1.2 たくさんの文章を表示しよう

例題は 2 行以上の文章を表示させることもできます。次のように 2 行を書き足してコンパイル・実行しましょう。

リスト 0-4

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("こんにちは!");
        System.out.println("どうしてそうなるの?");
        System.out.println("お勉強が楽しいです!");
    }
}
```

実行結果

こんにちは!

どうしてそうなるの?

お勉強が楽しいです!

会話のやりとりの中でも、開発現場でのヒントが詰め込まれている

作成するサンプルアプリはシンプルで、意味を理解しながら作っている

テンプレートを選ぶ

アプリを作るには、あらかじめ用意されているテンプレートを選びます。テンプレートはあらかじめ用意されているテンプレートで、その中から「空白のアプリ」として新しく作成することができます。そのテンプレートは、あらかじめ用意されているテンプレートで、その中から「空白のアプリ」として新しく作成することができます。



テンプレートを選ぶ

Single View Application: Single View Application は、単一の View を持つアプリケーションです。このテンプレートは、単一の View を持つアプリケーションを作成するためのテンプレートです。このテンプレートは、単一の View を持つアプリケーションを作成するためのテンプレートです。

616 Chapter 2 iPhone アプリ作り

やさしくはじめる iPhone アプリ作りの教科書 [Swift 3& Xcode 8.2 対応]

マイナビ出版
森崎尚 (著者)、
まつむらまきお (イラスト)
312 ページ 価格: 3,002 円 (PDF)



Swift
(iOS 開発)

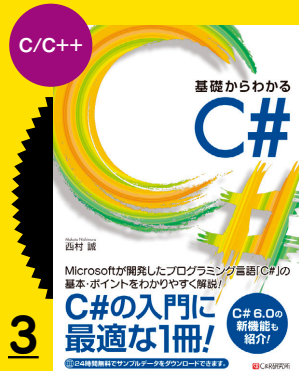
スッキリわかる Java 入門 第2版

インプレス
中山清義・国本大悟 (著者)
658 ページ
価格: 2,376 円 (PDF)



Java

**幅広いジャンルで活躍！
C# がキホンから学べる本**



3

基礎からわかる C#

本書はプログラミングの経験がある人を対象とした、プログラミング言語「C#」の入門書です。C# の概要から基本的な文法、特徴的な機能まで、わかりやすく解説しています。C# 6.0 の新機能についても解説しています。

シーアンドオール研究所
西村誠(著者)
168 ページ 価格: 1,944 円(PDF)

**プログラミング未経験でも
Android アプリを開発！**



4

**イラストでよくわかる
Android アプリの作り方
Android Studio 対応版**

親しみやすいイラストや、ステップバイステップでの丁寧な解説が基本コンセプト。開発環境「Android Studio」に対応し、Android のプログラムを作りながら、自然に Java というプログラム言語の知識が身につきます。

インプレス
羽山博・めじろまち(著者)
価格: 2,138 円(PDF)

**JavaScript を網羅的に
取り上げた骨太の 1 冊**



5

JavaScript 逆引きハンドブック

JavaScript の逆引きリファレンスの決定版。JavaScript の機能を網羅的に取り上げていて、骨太の 1 冊になっています。JavaScript の基本的な処理や便利な Tips はもちろん、HTML5 の API についても数多く掲載しています。

シーアンドオール研究所
古旗一浩(著者)
993 ページ 価格: 3,694 円(PDF)

**初めてのウェブ開発も安心
Ruby の文法を基礎から解説**



6

**改訂 3 版
基礎 Ruby on Rails**

Ruby の文法やオブジェクト指向の考え方を初歩から解説。アプリケーションのモックアップ作り、データベースを導入し、ログイン・ログアウト機能を加え、最終的にはメンバーや記事の管理ページまでできあがります。

インプレス
黒田努・佐藤和人(著者)
536 ページ 価格: 3,240 円(PDF)

**初歩から順に理解できる
PHP とデータベース**



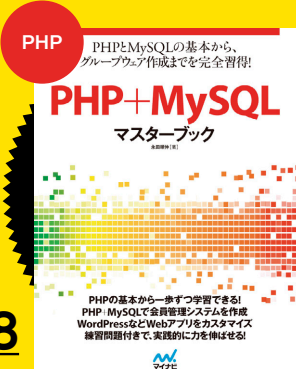
7

いちばんやさしい PHP の教本

PHP とデータベースの基本を順番に学んで、実践的なプログラムを完成させていく PHP の入門書です。大事なポイントや勘違いしやすいポイントは講師がフォロー。セミナーを受けている感覚で読み進められます。

インプレス
柏岡秀男・池田友子(著者)
240 ページ 価格: 1,836 円(PDF)

**PHP5.4 の基本から
MySQL との連携まで！**



8

PHP+MySQL マスターブック

この一冊で PHP と MySQL の基本と Web アプリケーションの構築法について学習できる実践的なプログラミング入門です。現場必須のプログラム構築法、API の活用法から、セキュリティ技術まで詳しく解説します。

マイナビ出版
永田順伸(著者)
384 ページ 価格: 2,916 円(PDF)

はじめてプログラミングに触れる前に読んでおこう



9

プログラミングの基本から手取り足取りじっくり解説



10

必ずアルゴリズムの意味がわかるようになる入門書！



11

プログラミングの世界へようこそ

全くの初心者がプログラミングを勉強したいとき、さまざまな疑問が湧いてきます。「どの言語を覚えればいい？」「文系でも大丈夫？」本書はプログラミングに触れる前に知っておきたい基本をイラスト付きで解説します。

マイナビ出版
尾川一行・中川聡(著者)
192 ページ 価格：1,933 円(PDF)

目指せプログラマー！
プログラミング超入門

本書は Windows 開発の標準ツールとも言える「Visual Studio」を使い、C# というプログラミング言語でプログラミングの基本を学びます。最終的には、ちょっとしたアクションゲームが作れるくらいになるのが目標です。

マイナビ出版
掌田津耶乃(著者)
312 ページ 価格：2,074 円(PDF)

楽しく学ぶ
アルゴリズムとプログラミングの図鑑

図解とイラストを豊富に使ったアルゴリズムの入門書。アルゴリズムとは「問題を解決するための考え方」です。それが分かってくれば、8 種類のプログラミング言語を使ったサンプルプログラムを実際に試しましょう。

マイナビ出版 森巧尚(著者)、まつむらまきお(イラスト)
300 ページ 価格：2,689 円(PDF)

プログラミングの初心者が Python 3 を学ぶのに最適



12

Go 言語の基礎から応用までポイントがよくわかる



13

R 言語の機能を目的から見つけ出せる！



14

基礎 Python 基礎シリーズ

プログラミングの初心者を対象にした Python 3 の入門書です。変数の取り扱いから、リスト、タプルといった Python 固有のデータの操作、制御構造や関数などについて、初心者でも基礎から学習できるように説明しました。

インプレス
大津真(著者)
312 ページ 価格：2,894 円(EPUB)

改訂 2 版
基礎からわかる Go 言語

Google が開発したプログラミング言語「Go」の基礎から応用までをわかりやすく解説した 1 冊です。最新の Go 1.4 のバージョンに対応して改訂しました。Linux、Mac OS X、Windows の各環境に対応しています。

シーアンドアール研究所 西川昇(著者)
240 ページ 価格：2,138 円(EPUB)

改訂 3 版
R 言語逆引きハンドブック

本書では、最新バージョンの R 3.3.0 に対応し、R 言語の機能を目的から探すことができます。統計が注目を集めるなか、R を利用するユーザーも増えています。初心者でも使えるように、導入から丁寧に解説しています。

シーアンドアール研究所 石田基広(著者)
800 ページ 価格：4,860 円(PDF)