# 1 Java をはじめよう

## 1.1 本書の目指すところ

本書は，Java 初心者の皆さんを対象に，Java について学んでもらおうという目論見で書かれています.

皆さんはプログラムを勉強しようと思って本書を手にとられたと思います。もちろんその目的は様々でしょう。ゲームを作りたい，という熱い情熱があったり、いちいち人手でやるのが面倒なことを自動化したい，と必要に迫られていたり、将来プログラムができると就職に有利そうだから、という打算と計算によって始めようという方もいるかもしれません。場合によっては、学校の宿題で出たからという切実な理由からかもしれませんね。個人的にはこの理由が一番多いんじゃないかと予想しています.

さて，どうせプログラムを作るんだから、ナントカファンタジーとかアラホッサクエストみたいなゲームや、ワードとかエクセルみたいなアプリケーションを作りたいと思うのが世の常です。しかしながら、ほとんどの入門書を読んでも最初のプログラムは地味の国から地味を広げに来たみたいなプログラムです。

特に、プログラム系で世界的に流行っているのが、画面に「HelloWorld!」と表示するプログラムです。大抵の入門書はなぜか「HelloWorld」と表示するところからスタートしています．嘘だと思うなら、他の入門書を読んでみてください。大抵「HelloWorld!」からです。何か決まりでもあるんでしょうか.

もちろん、JAVA でだって HelloWorld!と書くことが出来ますよ。画面に字を表示することだって大切な機能の一つですしね。しかし、人生初のプログラムがそんな地味なものでいいんでしょうか？

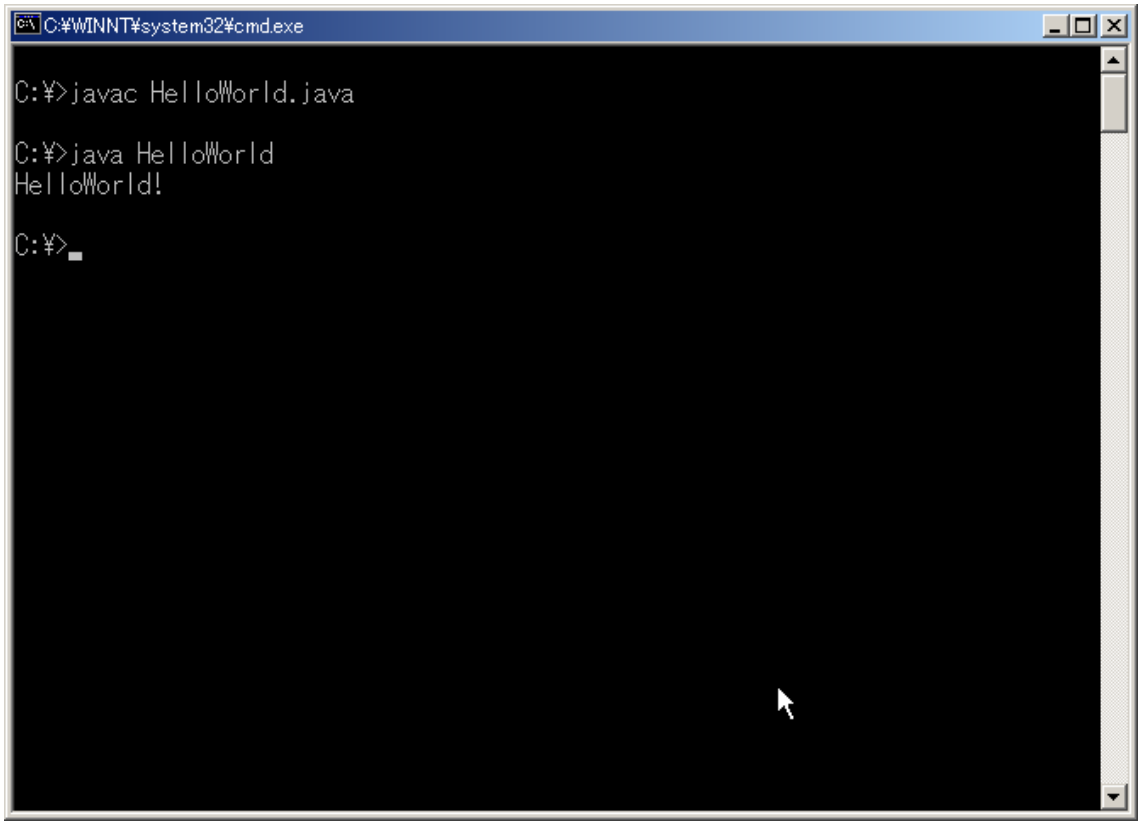


**図 1.1 地味の国から地味を広めに来たプログラムの実行結果**

・・・なんかつまんないですよね。せっかくプログラムを作るんですから、せめて他人に自慢できるくらいのプログラムが作りたいわけです。しかし、最初から豪快な実行結果を出すにはちょっとばかし初心者には荷が重いわけです。

そこで本書ではもう少しプログラムを書いた実感を持ってもらうために，こちらで用意したプログラムを改良してもらうところからスタートします.

みなさんは小学生の頃、自転車の練習をしたことを覚えていらっしゃいますか？最初のうちは乗れなくて苦労したことと思います。そんなとき、きっとお父さんやお兄さんが自転車の後ろを押さえて一緒に走ってくれたのではないでしょうか？もってもらって走るのは比較的簡単にできるわけです。で、何度も持ってもらって練習しているうちに、気が付けば自分ひとりの力でも自転車に乗れるようになっているわけです。

「持ってる？持ってる？」「持ってるよ～」なんて、実際には持っていないくせにお父さんのいう言葉に騙されて、持ってもらって走っていると思い込んだまま、実は自分の力で走っていた、そんな経験、誰にでもあるのではないかと思います。いや，筆者にはないので，もしかしたらマンガの中だけの話かもしれませんが。

何がいい対価といえば，いくら理屈で自転車の乗り方を学んだところで，実際に乗ったときどうなるのか分からなければ意味がないですよね．そもそも，理屈なんて知らなくたって，一度乗れてしまえば，後は感覚的に分かるようになってしまいます．そんなわけで、最初から理屈=構文を覚えるのではなくて，感覚的に「こう書くとこう動くんだ」というこ

とを学んでしまえば，あとは簡単です。そんな経験を踏まえて本書は構成されています。

本書では、構文についてはあまり細かく説明しません。「こういうプログラムを作ると，こういう結果になるよ」という例を中心に説明していきたいと思います．これによって、プログラムを実行したときの結果がどう変化するか、それを確認しながら、JAVA プログラムの本質に迫っていきたいと思います。

また，筆者が大学で学生達にプログラムを教えているときに，よく学生がやりがちな間違えポイントについても各章で述べています．皆さんもプログラムの勉強をしているときっとやる間違いだと思います．もちろん，全部覚える必要はありません．ただ，プログラムを作っているときにバグが発生したらちょっと思い出してください．もしかしたら，良くある間違えで，本書を読み返せばすぐに間違いが分かるかもしれません．もし，間違えやすいポイントだったとしたら，その解決方法についても述べていますので，参考にできると思います.

## 1.2 プログラム言語 Java

さて，まずは Java を勉強する前に，いったい JAVA とはいったい何なのかということについて説明したいと思います．特に，数あるプログラム言語の中で Java は他の言語とどう違うのかを説明したいと思います．そうすれば，他のプログラム言語じゃなくて Java を選ぶ理由というのもおのずと見えてきますよね.

そして，「Java の特徴」が分かれば，Java を勉強するモチベーションもあがるというものです．モチベーションがなければやる気も起きませんからね.

### 1.2.1 プログラム言語としての Java

#### 1.2.1.1 プログラム言語とは

JAVA とはプログラム言語の一種です。プログラム言語とはコンピュータに何か動作をさせるプログラムを作るための言語の事を言います。たとえば、ワードやエクセルなんていう市販ソフトもプログラムの一種ですし、Windows のような OS だって、一種のプログラムです。さらにいえば、プログラムはコンピュータの中だけで動くものではありません。たとえば、携帯電話の中でもプログラムは動いていますし、車でだって最近はエンジン制御のプログラムなんかが動いていたりします。オーディオ機器でも動いているし、場合に

よっては炊飯器の中でもプログラムが動いています。もう電気機器はすべからくプログラムで動いているといっても過言ではありません。

そんなプログラムを作るために必要なものがプログラム言語です。一般的なプログラミング言語としては、C/C++、Visual Basic、PERL など色々ありますが、その中の一つが JAVA というわけです。

なぜプログラム言語などというものが必要なのでしょうか？それは，コンピュータには人間の話す言葉が理解できないからです．単純に

「1+1 を計算しろ」

という命令だって，コンピュータにはできません．人間だったら 3 歳児だって理解出来そうな問題ですが，何しろコンピュータには日本語が通じないんですから，できるわけがありません．

コンピュータに理解できるのは，1 と 0 の組み合わせで作られる**機械語**だけなのです．1+1 を計算しろ，という命令をコンピュータに理解させるためにはこんな風に記述する必要があります．

```
100010101111111 100011001001100 100000001 …
```

こんな感じで 1 か 0 が並んでいる数列しかコンピュータには理解できません．

これでは，余りにも分かりづらい，ということでこの文字列を 16 進数という表現方法に変更します．そうすると，

```
457F  464C 0101…
```

こんな感じになります．少しわかりやすくなりましたか？う～ん，イマイチですね．

というわけで，人間とコンピュータを結ぶための言語変換が必要となります．そこで，コンピュータへの命令をわかりやすくするために開発されたのがプログラム言語です．プログラム言語のひとつである C 言語を使って 1+1 を実現するプログラムを作ると，

```
int main(){
        int x = 1+1;
}
```

となります．お，なんか 1+1 を計算しているっぽくなりました．

このようにして作成されたプログラムは，**コンパイル**という作業を通じて機械語に翻訳されます．最終的にはやっぱり 0 と 1 であらわされる数値になるんですね．ただし，コンパイル作業はコンピュータが自動で行ってくれるので人間は C や Java などのプログラム言語を使ってプログラムを書けばコンピュータに命令をさせることが出来るのです．そして，その命令の組み合わせによってアプリケーションが作られます．皆さんが使っているワードやエクセルなどのアプリケーションもプログラム言語によって書かれたものが，コンパイルによって機械語に翻訳されたものなのです．

というわけで，プログラム=コンピュータに命令を実行させるために書かなければいけないものと覚えてください．

#### 1.2.1.2 Java とヴァーチャルマシン

さて，前節で説明した機械語というのはコンピュータによって異なります．アメリカ人と日本人が言葉が通じないように，ウィンドウズとマックでは異なる形式の機械語を書かなければいけません． Linux などもまた違う書き方をします．そのため，プログラム言語は共通のものがあるのですが，機械語が異なるため，コンパイルはそれぞれのマシン用に行わなければいけません．

ウィンドウズのアプリケーションがマックや Linux で動かないのはこのためなのです．機械語がマシンによって異なるので，ウィンドウズにはウィンドウズの，マックにはマックの，Linux には Linux のアプリケーションを用意するときは，それぞれのマシン用にコンパイルしたものを作らなければいけませんでした．

電気店のパソコンソフトコーナーに行くと，Windows 用のソフトと Mac 用のソフトは別の棚においてあったりしますよね．これは，同じソフトを異なる OS で使うことが出来ないことが原因だったのです．

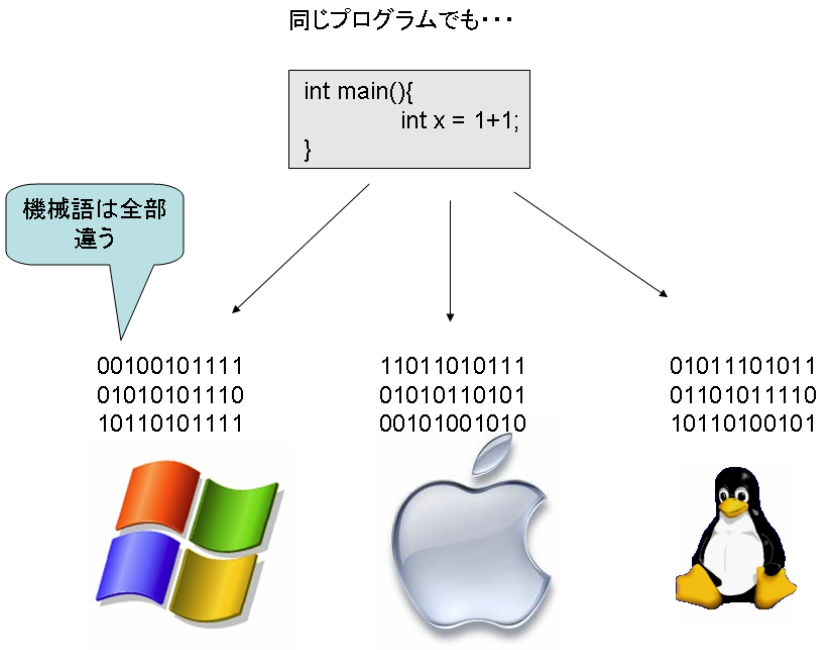


**図 1 機械語はマシンによって異なる**

それに対して，Java を使ってプログラムを作成した場合，**全てのコンピュータで同じように動く**ように作られています．

つまり，コンパイルを一回してアプリケーションを作ってしまえば，ウィンドウズでもマックでも Linux でも動かすことが出来るのです．これは，Java が JavaVM（Java ヴァーチャルマシン）という仕組みを採用しているためです．これは，簡単に言えば各コンピュータ上に Java 専用のコンピュータを仮想的に作成して，そのコンピュータを通して命令を実行する，という仕組みです．

Java バーチャルマシンは，JavaVM 用機械語が使えるという仕様に基づいて，ウィンドウズ用，マック用，Linux 用，はては携帯電話用まで色々なコンピュータに対応するものが作られています．

これによって，JavaVM をインストールしてあるマシンであれば，ひとつの Java アプリケーションをマシンの種類によらず動かすことができるのです．

つまり，Java を使えば，いちいち Windows 用か Mac 用かなど確認しなくても，誰でも使うことができる便利なソフトを開発することが出来るのです．

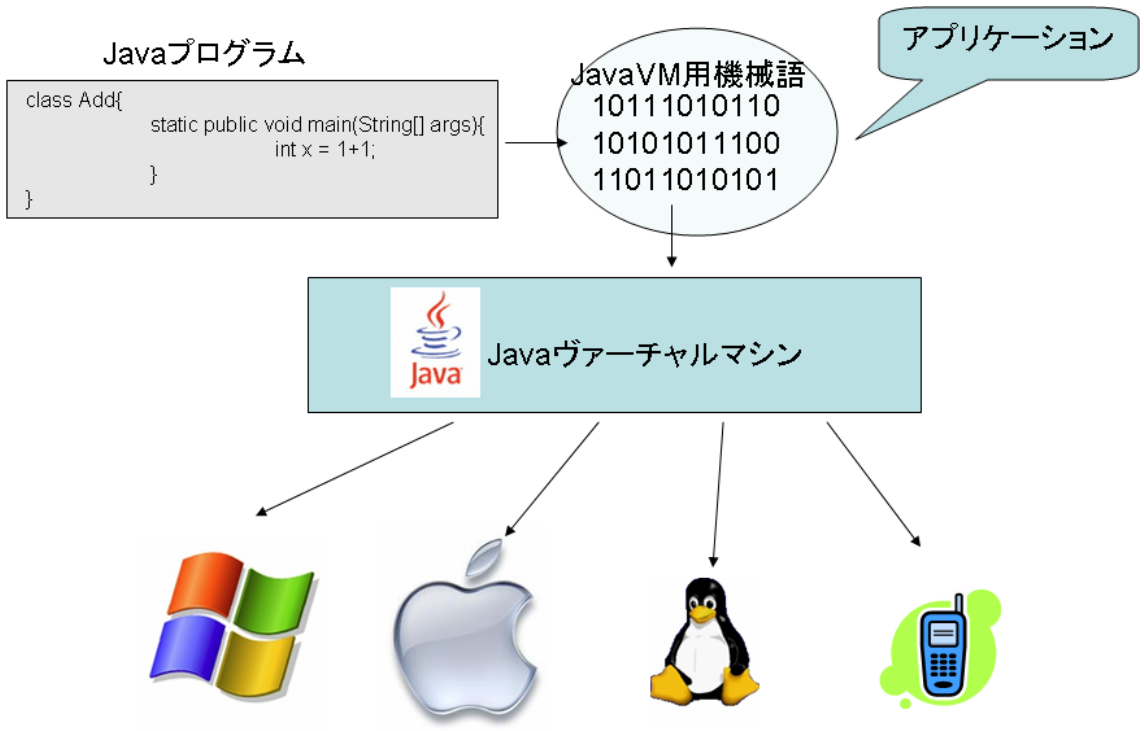


**図 2 JavaVM を使ってどこでも使える**

### 1.2.2 Java の特徴

Java が他のプログラム言語とちょっと違うことはわかったと思います．では，それを含めて，Java にはどんな特徴があるのかをここで列挙して行きたいと思います．

特に、JAVA が他のプログラム言語に比べて優れている点をあげてみましょう。

○ マルチプラットフォーム・Write Once, Run Anywhere
○ 用途に合わせた開発セットを提供
○ さまざまな API の提供
○ オブジェクト指向プログラミング言語

#### 1.2.2.1 マルチプラットフォーム・Write Once, Run Anywhere

すでに説明した通り，通常プログラム言語は、一つの OS で書いた場合、他の OS で動かすことは出来ません。Windows で書いたプログラムは MAC では動かないわけです。もちろん Linux で書いたプログラムは Windows では動きません。一方**JAVA はどこで書いても、どこでも動かすことができる**という利点があります。Windows で「やっほー」と表示するプログラムを書けば、Linux でも「やっほー」Mac でも「やっほー」．みんな仲良しです。

この便利な機能を使えば，Windows で開発したソフトを Mac で動かすことも，その逆も簡単に行うことが出来るわけです．家にある Windows マシンで開発したソフトを大学にもって行って，サーバで動かす事だって簡単に出来ちゃいます．あらすごい．

また，このどこでも動く性質を利用して、ネット上でも良く使われていたりします。**Java アプレット**という言葉を皆さんも聞いたことがあるのではないでしょうか？ブラウザ上でアプリケーションが動いてしまう仕組みです．普通のプログラムだと，使うマシンに適応したアプリケーションを作らないといけないため，ホームページのようにどんなマシンで見られているのかわからない場合は，アプリケーションを作ることが出来ません．このような場合でも，JAVA を使えば一つのアプリケーションでどんなマシンでも使うことが出来るため，ホームページ上で公開しても問題ないのです。あ、おなじネット上で見られるものでも JavaScript というものがありますが、これは JAVA とついているものの、JAVA とは別物ですので、ご注意を。

#### 1.2.2.2 用途に合わせた開発セットを提供

Java では，通常のアプリケーションを開発するためには基本的な Java プログラム開発用のセットが使われます．このセットのことを JavaSE(Java Standard Edition)とよびます．皆さんが主に使うのは，この基本セットになると思います．本書も，この基本セットである JavaSE のみを対象範囲としています．

基本的なセット以外に，企業などのサーバで利用することを目的に，JavaSE にサーバ機能を付加した JavaEE(Java Enterprise Edition)があります．このサーバ用 Java を使うことで，Web アプリケーションと呼ばれる，ホームページ上で動くアプリケーションんぼ開発が容易になります．ネットバンキングシステムやオンライン証券会社などでは，JavaEE を使っているところも数多くあります．

また，携帯電話をお持ちの方は、ゲームもいくつか持っていたりするんじゃないでしょうか？実は、携帯ゲームも JAVA で作られたものがほとんどです。このような携帯電話や PDA 用の Java として，JavaME(Java Micro Edition)があります．これを使えば、**携帯ア**

**プリを自作**する事が出来るわけです。筆者の友人などは携帯ゲームを開発して 300 万円も受けたとか言っています．うらやましい！皆さんも本書で Java を勉強して携帯アプリを開発して儲けちゃってください．私の取り分は 5%でかまいません．footnote まあ，本書の内容だけだと携帯専用アプリケーションを作るのにはちょっと情報が足りませんが・・・

このように，様々な用途で Java が使われています．Java プログラミングができれば，様々な場面で活躍の場が約束されていると行ってもいいでしょう．

#### 1.2.2.3 さまざまな API の提供

Java では，普通に作るには様々な知識が必要となる複雑な機能を API という形で最初から提供しています．

たとえば，インターネットに接続してホームページから JPEG ファイルなどの画像をダウンロードするアプリケーションを作りたいと思ったとしましょう．このようなアプリケーションを作るためには，まずインターネットの仕組みについて学ばなければいけません．TCP/IP とかパケットとか HTTP 通信とか. 普通の人には何のことだかよく分からないことを勉強してからでないと，作成することが出来ません．考えるだけでうんざりしてしまいます．

しかしながら，Java では最初から HTTP 通信することが出来る，つまりインターネットに簡単に接続して，ファイルを取得する機能が付随しています．そのため，お手軽にネットを利用したアプリケーションを作ったりすることが出来ます．

このような便利な機能はインターネット関連だけではなく，GUI（グラフィカルインターフェース）関連や，数学的なアルゴリズム処理などの機能もついていたりします．つまり，あまり難しいことを考えないでも作りたいプログラムを作ることが出来るわけです．

そんな API については，本書でも第 10 章で少し紹介しています．

#### 1.2.2.4 オブジェクト指向プログラミング言語

さらに、JAVA の特徴として、**オブジェクト指向プログラミング言語**であるということがあります。なんだそれ？実は，現在のプログラムの主流はこのオブジェクト指向プログラミングです。したがって、JAVA を学ぶということは、主流のプログラミング手法を学ぶということになるんです。ちなみに、他にもオブジェクト指向プログラミング言語としては、C++、C#、ruby、VisualBasic などがあります。それぞれに特徴がありますが、JAVA は最も洗練されたオブジェクト指向プログラミング言語の一つといわれています。

オブジェクト指向プログラミングを学びたいという要望にも Java は最適です．

#### 1.2.2.5 そのほかの特徴

Java は C，C++というプログラム言語の文法を継承しています．そのため，C や C++を

勉強した人にとっては覚えやすい言語です．もちろん，単に C や C++の言語を真似ているだけではなく，独自仕様をたくさん入れることによって，C,C++よりもはるかに洗練されたプログラム言語となっています．

また，C や C++といった言語には何十年という期間にわたって，プログラマを悩ませ続けたメモリ問題というものがあります．C や C++ではメモリの管理をプログラマがやっていたのですが，非常にバグが発生しやすいポイントでした．それに対して，Java ではメモリの管理をガベージコレクタと呼ばれる機能によって，Java 自体が行ってくれるため，C や C++に比べるとバグが発生する可能性が激減しました．C,C++を使ったことのない方にはさっぱりわからないかもしれませんが，わかる人にはわかる，すごい機能なのです．筆者などはガベージコレクタ機能でも Java を使う価値が十分あると思っています．

このように，Java を使う利点というのは実に様々です．まあ，今はピンと来ない話も多いかと思いますが，Java を勉強していくにつれ，色々な使い道が見えてくるようになると思います．

本書を読み終えたころにはどんな Java プログラミングでもばっちり・・・とは行きませんが，本書を足がかりに，Java についての知識を深めていってください．いずれ皆さんがすばらしいアプリケーションを開発してくれるようになることを期待しています．

とはいえ，それはまず本書で Java の基本を勉強した後の話になります．

まずは，Java の基本中の基本を学んで，広大な Java の世界への第一歩を踏み出してみましょう．

## 1.3 本書の構成

本書は、まず感覚的に Java をいじってみて、そのあとでどういう仕組みで動いているのかを理解していく、という構成になっています。

まずは、第 2 章で簡単なサンプルプログラムを用意しましたので、それを動かして見ましょう。また、この章では、Java を使ってどんなことができるのかの具体例を見ていきたいと思います。

第 3 章から本格的に Java 言語について学んでいくことになります。基本的な構文からスタートします．この辺はあまり面白くないので，挫折したくなるかもしれませんが，ぐっと我慢してください．

第 7 章まで進んだところで，ついに Java の本質であるクラスが登場します．第 7 章～第 9 章まで実に 3 章も使って Java のクラスについて説明しますので， 完璧にわからなくても良いですが，雰囲気をつかんでください．

第 10 章では，Java に元からついている便利な機能 JavaAPI を使う方法について説明します．JavaAPI は Java プログラミングをする上では必須といっても過言ではありません．うまく使いこなせる Java プログラマを目指しましょう．

第 11 章では，初心者から初級者へ進むとき最大の壁となる変数とメモリ管理についてお話したいと思います．この章で説明している参照型やオブジェクトという話は，最終的にはコンピュータの内部構造にまで話が広がる奥深い話ですが，そのさわりを簡単に説明したいと思います．これについての理解の深さが，今後の Java プログラマとしてもレベルアップに大きくかかわってきます．本章ではイメージをつかんでもらうことに重点を置いて説明しています．入門書でここまで説明しているものは少ないと思いますので，本章が本書のもっとも重要なポイントといえるかもしれません．・・・紹介文にも思わず力が入ってしまいました．

第 12 章では，Java のちょっと高度な機能と，今後 Java を勉強していく上で必要となる情報について説明しています．特に，後半の「調べ物をするときは」と「今後学ぶべきこと」は，本書を読み終えた後どうすればよいかのヒントになるはずです．次なるステップへ進むため，是非読んでください．

そして，各章の最後で，それぞれの章で初心者にありがちな間違いやすいポイントを説明しています．間違えるのは恥ずかしいことではありません．この間違えやすいポイント解説にあるような間違いをしたときに，どう直せばよいかのヒントになればと思います．

また，各章ごとに簡単な問題をつけておきました．問題を解くことで，多少理解が深まるのではないかと期待しています．面倒くさい人はやらなくてもかまいませんが．

さあ，以上で本書の構成の説明は終わりです．

次章からは早速 Java のプログラミングがスタートします．最初は，Java をインストールして，こちらで用意したプログラムを改造するところからスタートしてもらいます．

準備はいいですか？