

# Java によるプログラミング入門 (2)

## 1 Lesson 0: Java の学習環境を作ろう

### 1.1 プログラミング学習の環境

Java の学習のために ( 自宅や研究室に ) 以下のような環境を整えましょう :

コンピュータ : 当然パーソナルコンピュータが必要です . ここでは Windows 環境を想定します<sup>1</sup> .

<sup>1</sup>Java は linux でも動作します .

Java の処理環境 : Java2 SE の開発キット (JDK) などの Java プログラムを処理し実行するためのソフトウェアをコンピュータに導入します . ( 別紙参照 )

エディタ : Java プログラムはテキストファイル ( 文字だけのファイル ) として作成します . 適当なエディタが必要です . Windows のメモ帳でもかまいませんが , 高機能エディタであればそれだけ作業効率が高くなります<sup>2</sup> .

<sup>2</sup> 最近 は Java の開発用にエディタを内蔵した統合開発環境として Eclipse (<http://www.eclipse.org/>) が良く使われます . 本講習では秀丸エディタを使います .

英語の辞書 : 英語の資料を読むためだけでなくプログラミングでの変数などの命名に意味の分かりやすい名前を正しい綴りの英語で付けるために辞書が必要です .

ノートと筆記具 : プログラミング中にさまざまなメモを取ったり , 紙の上で図示して考えたりするためにノートや筆記用具は必需品です .

## 1.2 プログラミングのための参考書

プログラミングをさらに学んで行くためにはいろいろな参考書が役に立ちます。学習する内容は大きく 4 つの領域に分かれます。

特定言語の解説書 : Java など、実際に使うプログラミング言語の解説書です。

プログラミング一般・アルゴリズムの技術書 : 大きなプログラムを分かりやすく書くための工夫や、大量のデータの並べ替えや検索などのアルゴリズムについての解説書が参考になります。

関連技術の専門書 : やりたいことに応じて、その専門書が必要になることがあります。高次の方程式を解くなどの数値計算、データベースやネットワークなどです。

ソフトウェア開発の指南書 : 継続して使っていくプログラムには、正しく動作することのほかに、他人が読みやすい・管理しやすい・改善しやすいといったことに気をつけねばなりません。

最初は、実際に使うプログラミング言語の入門書を一冊だけ買うことをおすすめします。(巻末のリスト参照)

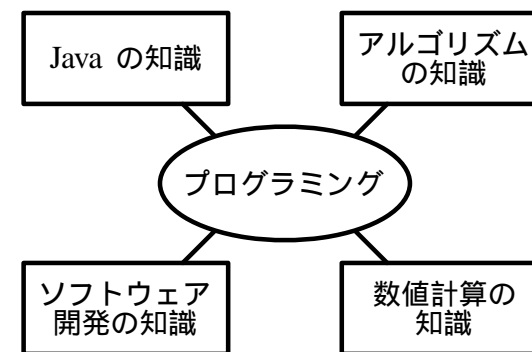


図 1 プログラミングの学習

## 1.3 Java プログラムの処理プロセス

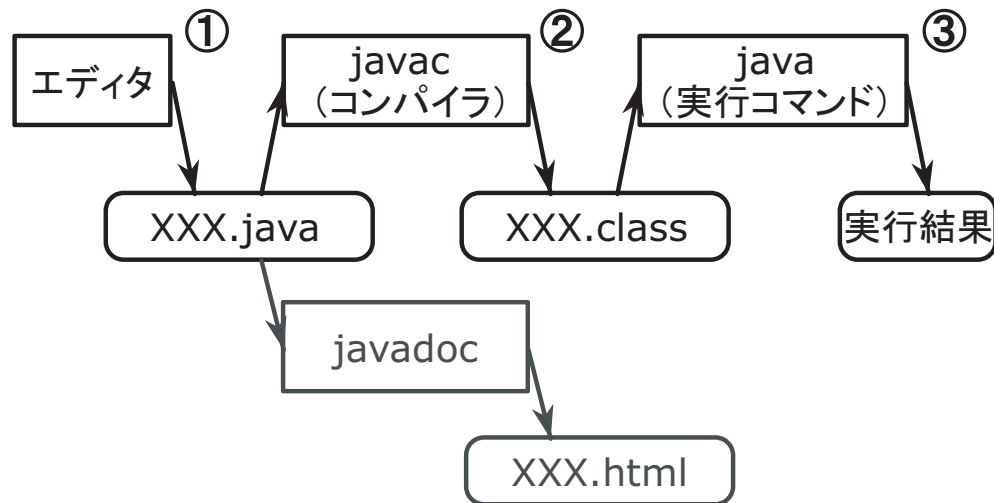


図2 Java プログラムの処理プロセス

Java プログラミングは以下の手順で行われます。<sup>3</sup>

1. 【プログラムを書く】エディタを使って，Java プログラム（拡張子は `.java` を用います）を書きます．
2. 【コンパイルする】書いたプログラムを `javac` というコマンド<sup>4</sup> で，機械が理解できる形式（拡張子は `.class` になります）に変換します．
3. 【実行する】次にこれを `java` というコマンド<sup>5</sup> で実行し，実行結果を得ます．

<sup>3</sup> 図に表れる `javadoc` は Java プログラムから Web ブラウザで読めるマニュアルを自動生成するためのものです．

<sup>4</sup> これも実はプログラムです

<sup>5</sup> これもです．

## 1.4 やってみよう – (1) プログラムを書く

教育用コンピュータシステムには、Java の開発環境 JDK がすでにインストールされています。早速、簡単なプログラムを入力、実行してみましょう。

1. まず、マイドキュメント (Mドライブ) を開き、そこに java フォルダを作ってください<sup>6</sup>。ここがあなたの作業フォルダです (閉じないで)。
2. スタート 全てのプログラム テキスト編集 秀丸を選んでください。(以降、スタート 秀丸 と選べるようになります)
3. 以下の 5 行からなるプログラムをそのまま打ち込んでください。
4. ファイル名をつけて保存で、先ほどの作業フォルダを選び、HelloCrier.java と名前をつけて保存してください。ちゃんと作業フォルダ内にファイルができましたか？

<sup>6</sup>Network drive がない人は、[マイコンピュータ] [ローカルディスク C:] [Documents and Settings] [a00xxxxxx] の中に java フォルダを作ってください

```
public class HelloCrier {
    public static void main(String args[]) {
        System.out.println("Hello World.");
    }
}
```

## 1.5 やってみよう – (2) javac でコンパイルする

【移動】続いて，[スタート] [プログラム] [アクセサリ] [コマンドプロンプト] と選び，コマンド入力画面を開きます．

```
C:¥ Documents and Settings ¥ a00xxxxx>
```

の表示が出ましたか？次に，HelloCrier.java を作った場所に移動します<sup>7</sup>．

```
M:
```

と打鍵し，Enter キーを入力してください．続いて，

```
cd java
```

と打鍵し，Enter キーを入力してください．

```
M: ¥ java>
```

と表示されたら，移動完了です．

【コンパイル】いよいよコンパイルのために，javac コマンドを使います．

```
javac HelloCrier.java
```

と打鍵し，Enter キーを入力してください．何も表示されなければ成功です．

【確認】作業フォルダに，HelloCrier.class ができていることを確認してください．

<sup>7</sup> すでに作った場所にいる場合移動は不要です．Network drive がなかった人は，cd java コマンドだけ入力して C:¥ Documents and Settings ¥ a00xxxxx ¥ java へ移動してください．

### 1.5.1 コンパイルに失敗したら

コンピュータちゃんは、融通が利きません。ちょっとでも何か気に入らないと、すぐエラーを出します。もし画面にいろいろメッセージがでるようなら以下のことを確認してください。

1. 投入したコマンド `javac HelloCrier.java` は間違っていますか？
2. `javac` が認識されていますか？（教育用システムでは大丈夫のはず）
3. 今いる場所は `M:¥java` ですか？<sup>8</sup>
4. プログラムをちゃんと“保存”しましたか？
5. `HelloCrier.java` は `M:¥java` にありますか？
6. 上記いずれでもなければ、`HelloCrier.java` の中身が間違えています。計算機のプログラムは少しの打ち間違いで実行できなくなります。よく注意してチェックしてください。プログラムは基本的に半角英数字で作成します。行末などの「; (セミコロン)」と「: (コロン)」や「,(コンマ)」と「.(ピリオド)」など細かな記号や「O (オー)」と「0 (ゼロ)」、「l (エル)」と「1 (イチ)」など見誤りやすい文字も含めてよくチェックしてください。

エラーメッセージも参考になります。手直しが終了したら、気を取り直して再度、コンパイルに挑戦してください<sup>9</sup>。

<sup>8</sup> 以下、network drive がなかった人は、適宜 `M:¥java` を `C:¥Documents and Settings¥a00xxxxx¥java` に読み替えてください。

<sup>9</sup> エディタでの編集作業とコマンドプロンプトでの実行を並行して行うときには、プログラムの「上書き保存」を忘れないようにしてください

## 1.6 やってみよう – (3) java で実行する

コンパイルが無事終了したら , java コマンドを使ってプログラムを実行します .  
コンパイルをした場所 ( HelloCrier.class がある場所 ) で ,

```
java HelloCrier
```

と打鍵してください<sup>10</sup> . プログラムが挨拶してくれたら成功です .

<sup>10</sup> プログラムの拡張子 .class は指定する必要がありません .

```
M: ¥ java> javac HelloCrier.java
```

```
M: ¥ java> java HelloCrier  
Hello World.
```

```
M: ¥ java>
```

続いて , もっと「プログラムらしい」プログラムを入力・実行してもらいます . まだ何も説明していないのでツライかもしれませんが , 何をしようとしているプログラムなのか , 想像しながら打ち込んでください . なお , 先頭の数字とコロン ( : ) は説明用につけたものです . 入力しないでください

## 1.7 使用するサンプルプログラム (TankCalculator.java)

```
1: public class TankCalculator {
2:     public static void main(String args[]) {
3:         final double FLOW_RATE = 1.0;
4:         final double TANK_AREA = 20.0;
5:         final double INITIAL_LEVEL = 10.0;
6:         double time;          //s
7:         double tankLevel;    //m
8:
9:         System.out.println("Flow Rate = " + FLOW_RATE + "m**3/s");
10:        System.out.println("Tank Area = " + TANK_AREA + "m**2");
11:        System.out.println("Initial Level = " + INITIAL_LEVEL + "m");
12:
13:        time = 30;
14:        tankLevel = INITIAL_LEVEL + FLOW_RATE*time/TANK_AREA;
15:        System.out.println("Tank Level at time "+ time + "s = " + tankLevel + "m");
16:        time = 60;
17:        tankLevel = INITIAL_LEVEL + FLOW_RATE*time/TANK_AREA;
18:        System.out.println("Tank Level at time "+ time + "s = " + tankLevel + "m");
19:    }
20: }
```

【補足】 // の後ろは、プログラムを後で読解しやすくするための注釈です。



### 1.7.1 サンプルプログラムの入力と実行

先ほどと同じように、秀丸エディタを開き、20 行のプログラムを書き込んで、TankCalculator.java と名付け、保存して、コンパイル、実行してください。

成功すれば、以下の実行結果が示されます。(失敗してもめげないで、2.5.1 節を参考に、原因を考え、再トライしてください)

```
Flow Rate = 1.0m**3/s
Tank Area = 20.0mm**2
Initial Level = 10.0m
Tank Level at time 30.0s = 11.5m
Tank Level at time 60.0s = 13.0m
```

ちなみに、実行結果をファイルに書き出すにはコマンドプロンプトの「リダイレクト」という機能を使います<sup>11</sup>。

```
java TankCalculator > result.txt
```

これにより result.txt というファイルが出来ているはずです。中身は数値や文字列だけのテキストファイルですのでエディタなどで内容を確認できます。

<sup>11</sup>java プログラムの中で明示的にファイルに出力することもできますですがここでは安直な方法を取ります

### 1.7.2 実行時のエラー

プログラムが無事にコンパイルされたからといって正しいプログラムであるとは限りません。

- 実行時のエラー：たとえば 0 で割り算するなどのエラーは数値が確定しないコンパイル時には発見できず，実行時のエラーとなります。
- 特にエラーは生じないが実行結果が正しくない。プログラムのどこかが誤っているために，文法的には正しいし，特に実行時のエラーも生じないが実行結果が正しくないということも生じます。

どこが誤っているのかを究明する作業をデバッグ（虫取り）と言います。プログラムの動作を系統的に確認して誤りが含まれる可能性のある箇所を次第に狭めてゆくなどの論理的な思考が求められます。

## 参考文献

- [1] アスキー書籍編集部：Java2 プログラミング講座，アスキー（2002）.
- [2] 高橋 麻奈：やさしい Java，ソフトバンクパブリッシング，2002
- [3] 結城 浩：改訂版 Java 言語プログラミングレッスン（上，下），ソフトバンクパブリッシング，2003
- [4] ケン アーノルド：プログラミング言語 Java，ピアソンエデュケーション，2001
- [5] 結城 浩：増補改訂版 Java 言語で学ぶデザインパターン入門，ソフトバンクパブリッシング，2004
- [6] マーチン ファウラー：リファクタリング プログラムの体質改善テクニック，ピアソンエデュケーション，2002
- [7] 奥村 晴彦ほか：Java によるアルゴリズム事典，技術評論社，2003
- [8] 河合：最新オブジェクト指向がわかる，技術評論社（2001）
- [9] 英和システムマネジメント：Java コーディング標準（2001）
- [10] Sun Microsystems: Java Code Conventions Java に関するコード化の慣習についての資料です．
- [11] B. W. Kernighan and R. Pike: (福崎訳) プログラミング作法，ASCII（2000）. プログラミングに関する作法についての教科書です．