

オブジェクト指向 Java プログラミング入門 第 2 版：補足資料

2018 年 3 月 10 日

1 はじめに

この文書は「オブジェクト指向 Java プログラミング入門 第 2 版」(以下, 教科書)を利用する上で必要になる, 統合開発環境の利用方法についてまとめています。本書の内容は, ソフトウェアのバージョンアップや状況の変化によって大きく変わってくるため, 書籍の中に含めるよりは補足資料として提供した方が良いと判断したものです。

本書の内容はプログラミングそのものを解説したものではありませんが, 教科書を利用する上では避けて通れないものです。よく読んで教科書の学習を進めていってください。

2 統合開発環境 Eclipse

教科書の 2 章でも触れていますが, プログラムを作成するためには,

1. ソースコードを書く
2. コンパイルを行う
3. 実行してテストする
4. (うまく実行しなかったら) デバッグする

が基本的な流れになります。第 2 章で説明したように, ソースコードの作成や編集には emacs などのエディタを用い, コンパイル以降の作業はターミナルでコマンドを実行していくのがシンプルな方法です。しかし, プログラムが大規模になっていくと, 作業が煩雑になってバグも見付けにくくなり, 開発効率が低下していきます。

そこで, 現在主流になっているのは統合開発環境を用いた開発です。統合開発環境とは, プログラミング専用のエディタ, 処理系のフロントエンド, そしてデバッガが一つになったもので, 上記の一連の作業を効率的に行うための便利な機能が搭載されています。

Java プログラミングを行うための統合開発環境として広く用いられているのは Eclipse です。Eclipse は Eclipse Foundation という組織によって開発されているオープンソースの統合開発環境です。プラグインとして様々な機能を組み込むことができ, Java だけでなく, TeX, C++, Python といった様々な言語への対応が可能になっています。Eclipse 自体も Java で開発されており, Windows 版, Linux 版, macOS 版が提供されています。すべて <http://www.eclipse.org/> から無料でダウンロードして利用することができます。

Eclipse には多くの使いやすい機能が備わっていますが, 初学者はすべての機能のマスターを目指すより,

基本的なものから始めて徐々に慣れていく方が良いでしょう。そこで、本節では、Eclipse を用いてプログラムを作成するのに必要と思われる最低限の機能について紹介します。その他多くの便利な機能を知りたい読者は、この文書を読み終えた後、Eclipse に装備されているチュートリアルやヘルプなどを実行して楽しんで下さい。それでは、Eclipse を使用するための準備から始めましょう。

2.1 準備

2.1.1 Java 開発環境 JDK の準備

Eclipse を実行させるためには、JDK (Java Development Kit) を準備する必要があります。JDK には Java Runtime Environment (JRE) と、アプレットやアプリケーションの開発便利なコマンド行開発ツールが含まれています。Linux 版、macOS 版、Windows 版、Solaris SPARC 版が無料で提供されています。

JDK は、Oracle Technology Network からダウンロードします。URL は、<http://www.oracle.com/technetwork/java/index.html> です。このページ右側の Software Downloads にある「Java SE」をクリックすると、ページの先頭に最新バージョンの Java SE (Java Platform, Standard Edition) の紹介が表示されます。JDK の「DOWNLOAD」をクリックして、お手持ちの実行環境に合ったプラットフォームの JDK を導入して下さい。

2.1.2 Eclipse の準備

次に、Eclipse の準備に取り掛かりましょう。Windows 版、Linux 版、および macOS 版が無料で提供されています。

Eclipse は Eclipse の公式 Web サイトからダウンロードします。URL は、<http://www.eclipse.org/downloads> です。本書では、Eclipse 4.7 Oxygen を準備しました。ダウンロードは、zip 形式、または tar.gz 形式の圧縮ファイルです。導入は非常に簡単です。圧縮ファイルを展開するだけです。このように公式サイトから導入する他にも、macOS なら Homebrew、Linux なら APT などのパッケージ管理システムを利用してインストールしても構いません。

ただし、公式サイトなどからインストールした場合は、メニューやメッセージが英語で表示されますので、使いにくい場合もあります。日本語化したい場合は、Mergedoc Project の日本語化プラグイン Pleiades (プレアデス) が利用できます。<http://mergedoc.osdn.jp/> にアクセスすると Pleiades All in One (Eclipse 本体と日本語化プラグインなどをまとめたパッケージ) と Pleiades プラグインがダウンロードできます。現在、前者は Windows 版と macOS 版が用意されていますので、手っ取り早く準備できます。Linux を利用している場合は、プラグインをダウンロードして自分で設定しなくてはなりません。

以上で、準備完了です。

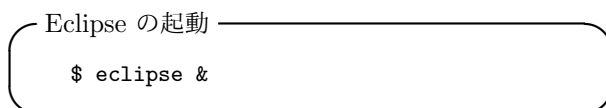
2.2 Eclipse を使ってみよう

教科書第 2 章の 2.1 節でも取りあげた Hello World プログラムを Eclipse で作成してみましょう。途中で聞きなれないキーワードが出てきますが、あまり気にしないでこの節の最後まで試していきましょう。Eclipse では、`javac` や `java` などのコマンドをキーボードから入力することなく、マウス操作のみで簡単にプログラムの作成や実行ができます*1。

*1 実際は内部で `javac` などのコマンドが適切に実行されています。どのように実行されているかを考えてみるのも良いでしょう。

2.2.1 Eclipse の起動

ターミナルに、以下のコマンドを打ち込んで、Eclipse を起動しましょう。



すると、図 1 の「Eclipse の起動」、図 2 の「ワークスペースの選択」の順に表示され、ソースファイルやクラスファイルなどを保管するワークスペースをどこにするのか聞いてきます。



図 1 Eclipse の起動

このまま [OK] を実行してもよいのですが、希望するディレクトリを入力することもできます。

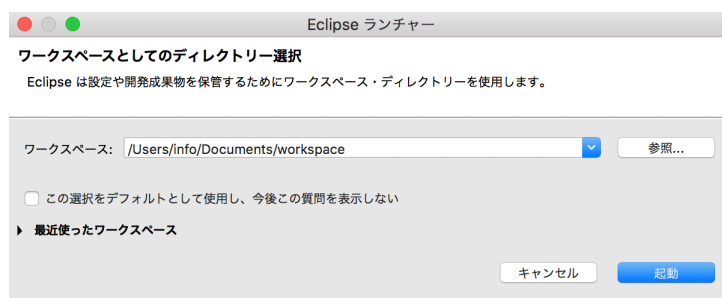


図 2 ワークスペースの選択

[OK] を実行すると、図 3 の「ようこそ」が表示されます。ここから、新機能の確認やチュートリアルを見ることができますが、ここでは、図 4 の「Java パースペクティブ」を表示させるために、右上の「ワークベンチ」をマウスでクリックして実行しましょう。

再び、図 3 の「ようこそ」を表示させたいときは、図 4 「Java パースペクティブ」のメニューバーから [ヘルプ] → [ようこそ] を実行します。

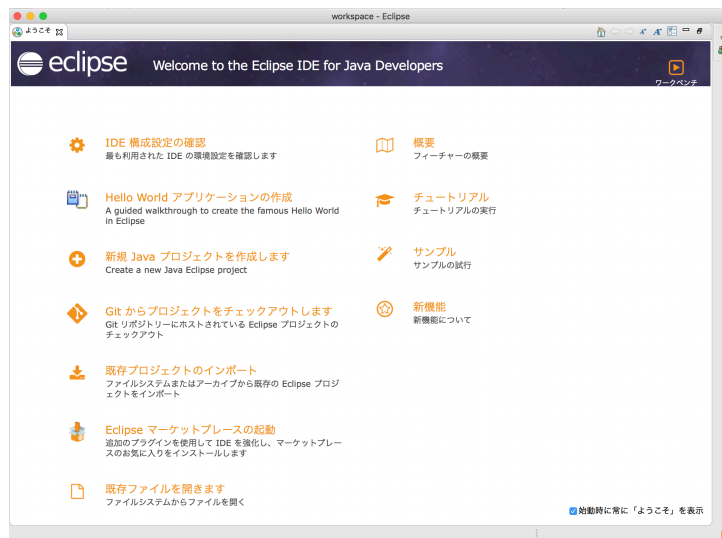


図 3 ようこそ

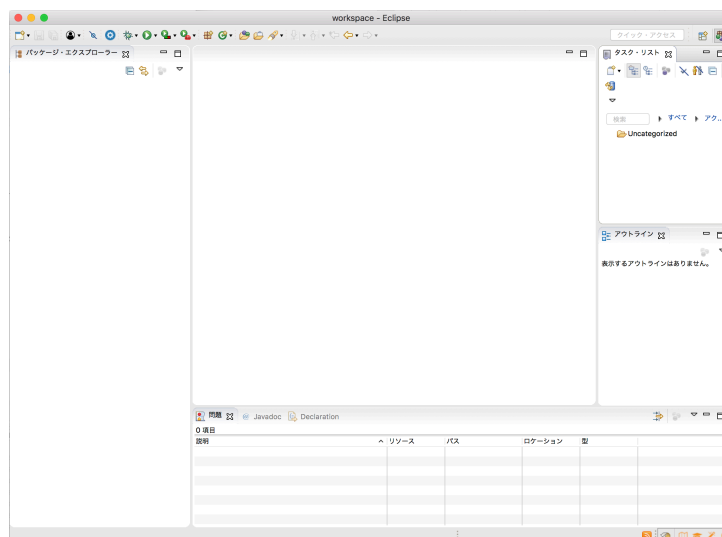


図 4 Java パースペクティブ

2.2.2 プロジェクトの作成

それでは、Eclipse を用いて Hello World プログラムの作成に取り掛かりましょう。最初に作成するのは、Java プロジェクトです。ここでは、プロジェクト名を「JavaBook」としましょう。

1. メニューバーから [ファイル] → [新規] → [プロジェクト] を選択。ドの選択」の「Java プロジェクト」を選択。
2. 図 5 の「JavaBook プロジェクトの作成」のように、プロジェクト名に JavaBook を入力。
3. [OK] をクリック。

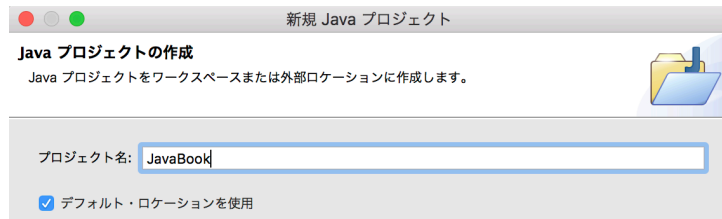


図 5 JavaBook プロジェクトの作成

これで、プロジェクト JavaBook をワークスペースに作成しました。「Java パースペクティブ」の左の「パッケージエクスプローラ」に「JavaBook」のアイコンが表示されます。

2.2.3 クラスの作成

次に、HelloWorld クラスを作成します。

1. 表示された アイコン「JavaBook」を右クリック。
2. 左ボタンに切替え, [新規] → [クラス] を選択。
3. 図 6 の「HelloWorld クラスの作成」のように、パッケージ名には何も入力せず、名前に HelloWorld と入力。
4. 「作成するメソッド・スタブ」の `public static void main(String[] args)` を選択。

OK をクリック。

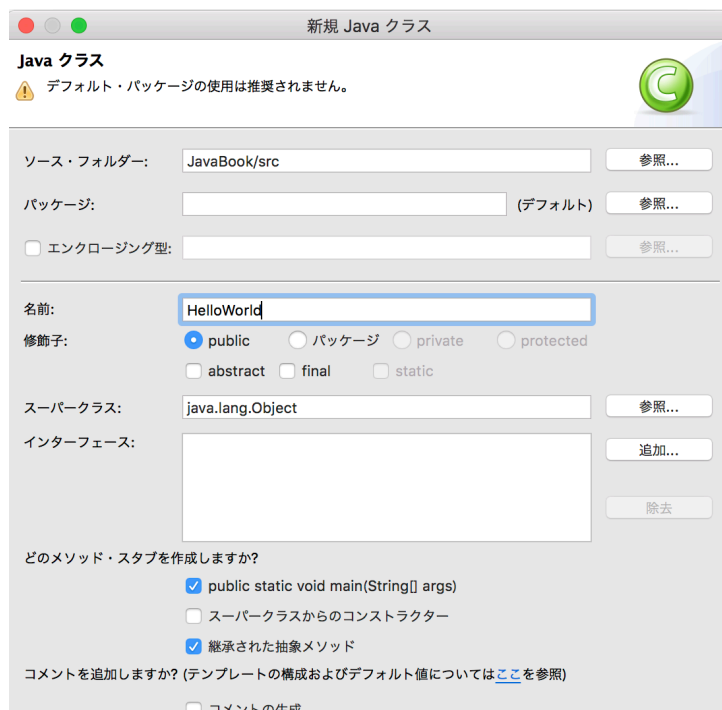


図 6 HelloWorld クラスの作成

これで、HelloWorld クラスが作成され、図 7 のように、HelloWorld.java の内容が中央のエディターに表示されます。

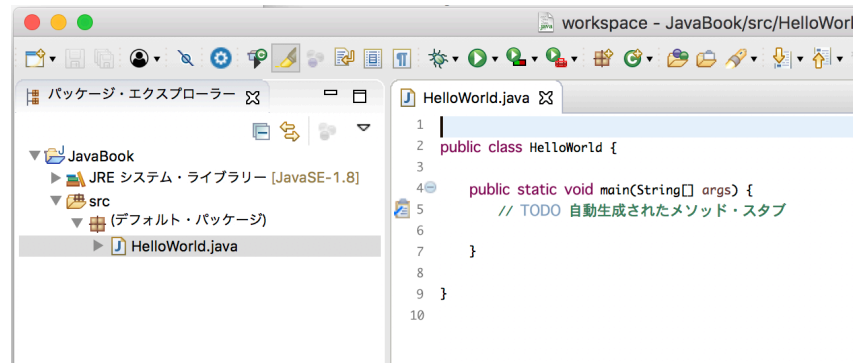


図 7 自動作成された HelloWorld.java

2.2.4 メソッドを書く

いよいよ、メソッドを書く段階になりました。メソッドとは、情報処理を行う方法です。画面に、Hello world! と表示するには、main メソッドに `System.out.println("Hello world!");` と記述します。main メソッドは、最初に実行されるメソッドです。詳しくは教科書で説明します。

また、図 7 のエディター内に、// で始まる行が記述されています。これは、// 以下、コメントであるというマークです。この main メソッドは自動生成されたメソッド・スタブであるというコメントが自動的に書かれています。TODO と書いてあるように、このメソッドは中の処理が書かれていませんので、それを書くことがユーザー（プログラマー）がやるべきことになります。

では、メソッドを書いてみましょう。

1. エディターの 9 行目をマウスでクリックし、System.out. までキーボード入力。
2. Eclipse は、その後に記述できる命令を探して、メニュー形式で表示します。

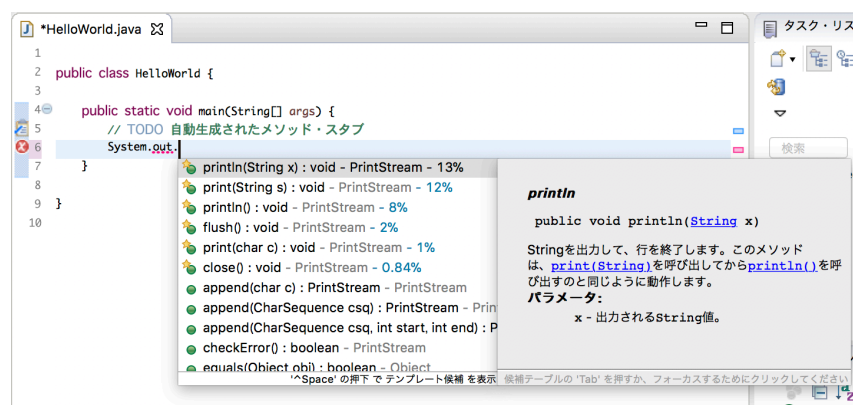


図 8 希望するメソッドの選択

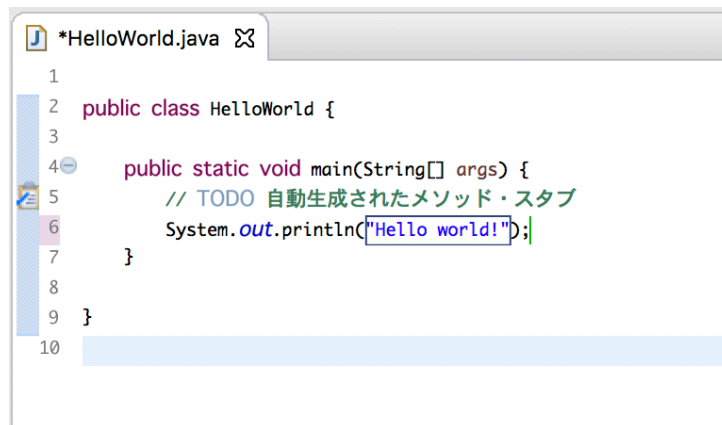


図 9 Hello World プログラム

3. ここでは、図 8 のように、println(String x) を選択.

4. そして、キーボードで ‘Hello world!’ と入力して最後に Enter を押す.

すると、図 9 のように作成できます. キーボードで入力したのはほんの少しでしたね. JDK のみを用いた場合は、多くの文字を手で入力する必要があり、一文字でもタイプミスすると正しくコンパイルできません. Eclipse を用いた場合は、このように親切にサポートしてくれますので、入力ミスによるエラーと修正の手間を低減することができます.

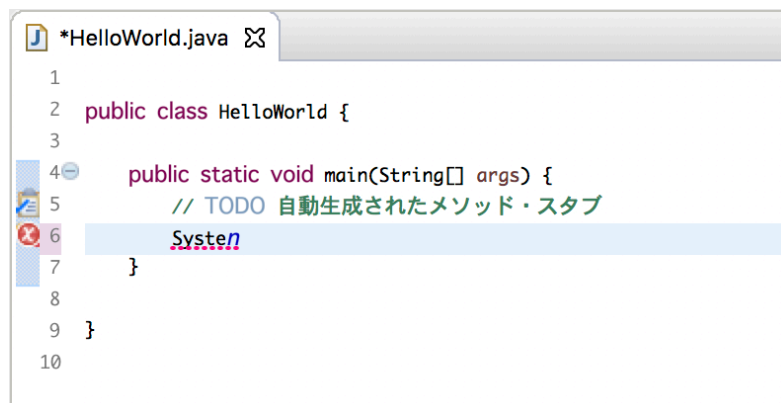


図 10 エラー箇所の表示

教科書第 2 章の 7 ページから 8 ページで、System とタイプミスするとコンパイルがエラーになる例を示しましたが、Eclipse では、キーボード入力すると同時に、図 10 のように、エラー箇所に点線を表示します. さらに、図 11 のように、エラー箇所をクリックすれば、エラー内容やヒントを示してくれます. 図 11 の場合はクイック・フィックスとして 4 つの対応を Eclipse が提案してくれています. 残念ながらここでは System を System と修正するという提案はないのですが、もし適切なものが提案されていれば、それをクリックするだけで自動で修正してくれます.



図 11 エラーの内容やヒントの表示

2.2.5 ソースファイルの保管

HelloWorld.java のエディターのどこでも構いません。右クリックして、[保管] を実行すると、編集した内容が保存されます。Ctrl + S や command + S などのショートカットキーも利用できます。こまめに保管するようにしておきましょう。

2.2.6 Java アプリケーションの実行

最後の瞬間です! HelloWorld を実行する準備が整いました。保管のときと同様に、HelloWorld.java のエディターを右クリックして、[実行] → [Java アプリケーション] を実行すると、図 12 のように、Hello world! が「コンソール」に表示されます。これで Hello world アプリケーションのビルドと実行が成功したことになります。どうです。非常に簡単にプログラムが作成できたでしょう。もう、Eclipse の魅力に取り付かれたはずですよ。詳細な使い方は、メニューバーの [ヘルプ] をいろいろ実行して、楽しんでみましょう。

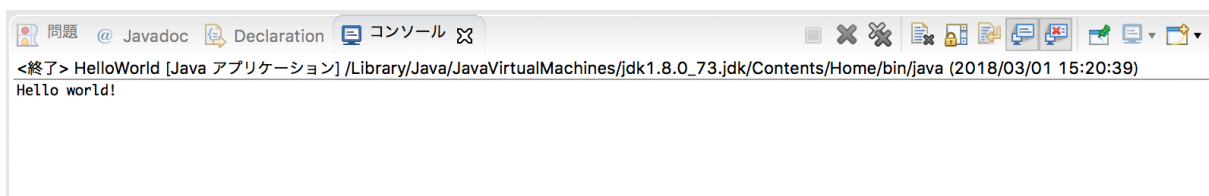


図 12 Hello World の実行