

1-3

プログラムの作成

学習の
ポイント

- Eclipseはプログラムを作るためのアプリケーションです。
- Eclipseを使って、Java言語でプログラムを作成する方法を学習します。

KEYWORD

- エディター
- 統合開発環境
- Eclipse

注①-7

Java言語によるプログラミングが行える統合開発環境には、ほかにも「NetBeans (ネットビーンズ)」などがあります。

■ Eclipse とは

プログラムを作成し、それをコンピュータに実行させるには、プログラムコードを記述するためのソフトウェア（**エディター**といいます）やコンパイラが必要です。また、コンパイルしたプログラムを実行し、その結果を表示する仕組みなどにも必要になります。このような一連の機能を提供するアプリケーションを**統合開発環境**（ごうがいはつかんきょう）といい、その1つに**Eclipse**（エクリプス）があります。

本節では、Eclipseを使ってJava言語によるプログラミングを行う方法を説明します（注①-7）。Eclipseはプログラミングを行うためのアプリケーションとして広く使われており、学校や職場のコンピュータで使えるようになっていることが多いでしょう。無償で使うことができるので、普段使用するパソコンにもインストールしておくと便利です。Eclipseのインストール方法は、本書の巻末付録Aで説明しています。

以降では、お手元のパソコンにEclipseがインストールされていることを前提に、プログラムの作成方法を説明していきます。

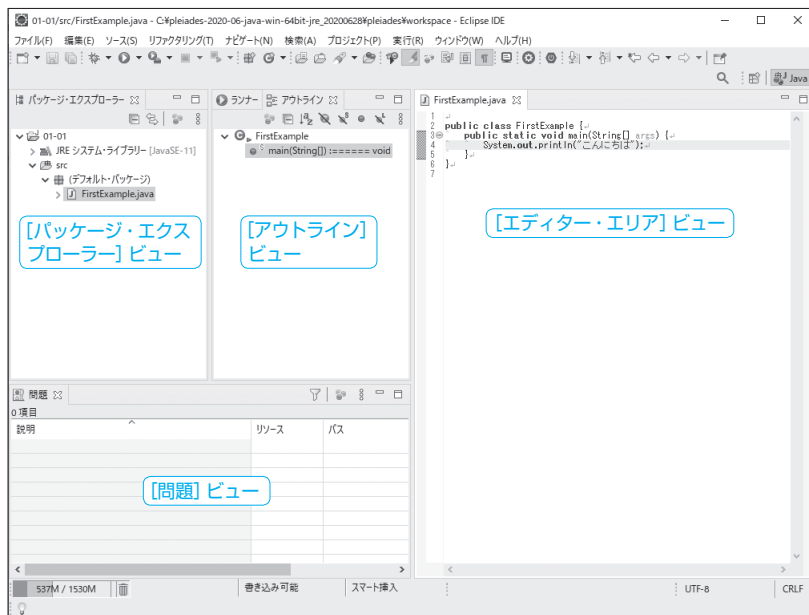
■ Eclipseの画面構成

それでは、Eclipseを使ってプログラミングをしてみましょう。Eclipseは画面①-1のような構成をしています（注①-8）。

注①-8

本書では、Windows版のEclipse 2020-12を使って説明を進めていきます。使用するEclipseのバージョンによっては画面構成が異なる場合があります。

画面①-1 Eclipseのウィンドウの構成



各ビューは、タブをドラッグすることで自由に位置を動かすことができます。また、[×] ボタンを押すと閉じることができます。画面に表示されていないビューを表示させるには、[ウィンドウ] メニュー→[ビューの表示] から表示させたいビューを選択します。

Eclipseのビューは、画面①-1に表示されているもの以外にもたくさんありますが、主に使用するのは次の4つです。

●[パッケージ・エクスプローラー] ビュー

プロジェクトの構成要素を表示します（プロジェクトについては後ほど説明します）。

●[エディター・エリア] ビュー

ここにJava言語のプログラムコードを記述します。

●[コンソール] ビュー

プログラムを実行した結果が表示されます。画面①-1では見えていませんが、プログラムを実行すると[問題]ビューと同じ場所に現れます。

●[問題] ビュー

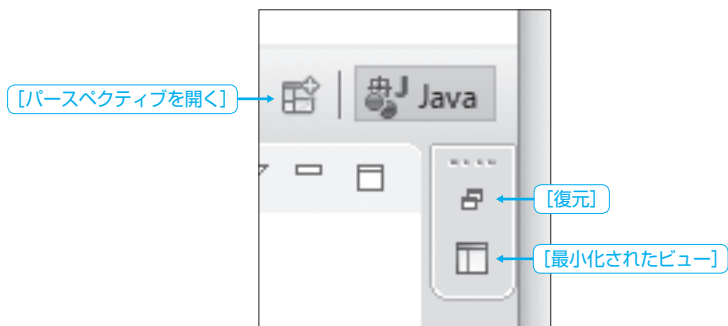
[エディター・エリア] ビューに入力したプログラムコードを保存したときに誤りがあると、問題点がここに報告されます。誤りや問題がなければ、ここには

何も表示されていません。

メモ

ビューを誤って最小化してしまった場合は、Eclipseの画面の端を見てください。画面①-2のように最小化されたビューが示されています。これをクリックすれば、ビューは元どおりに表示されます。

画面①-2 ビューとパースペクティブを復元するためのアイコン



Eclipseの表示が画面①-1のような状態になっていない、または操作しているうちに意図せず異なる状態になってしまった場合には、画面①-2の「パースペクティブを開く」アイコンを押して表示される選択肢の中から「Java (デフォルト)」を選択します。

なお「パースペクティブ」とは、画面のレイアウトに名前をつけたものです。

■ プログラムを作成して実行する

それでは、EclipseでJavaのプログラムを作成し、実行してみることにしましょう。例として、本章の最初に紹介したプログラム(List①-1)を使います。

プログラムを作成し実行するまでには、「プロジェクトの作成」「プログラムコードの作成」「実行」の3つのステップを踏む必要があります。これから説明する手順に沿って実際にEclipseを操作し、プログラムを実行してみましょう。

■ ステップ1：プロジェクトの作成

Eclipseでプログラムを作成するには、初めに**プロジェクト**を作成する必要があります。プロジェクトという言葉は少々大げさに聞こえるかもしれませんが、

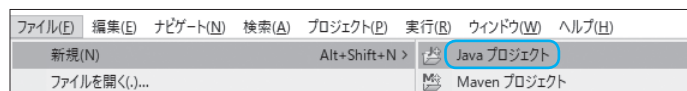
KEYWORD

● プロジェクト

「プログラムコードを管理する単位（フォルダのようなもの）」と考えてください。
プロジェクトは次の手順で作成します。

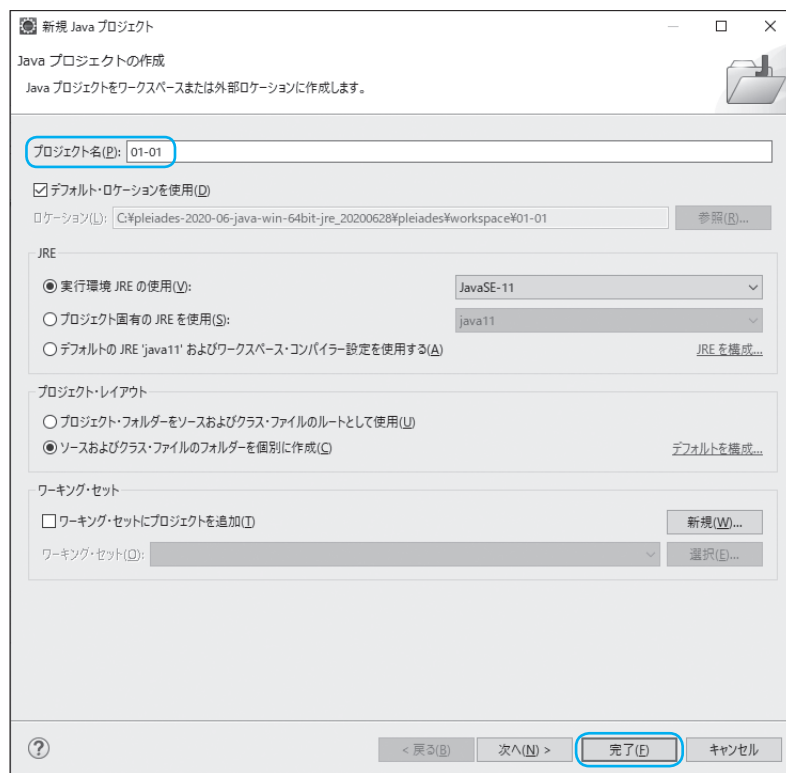
- ① [ファイル] メニュー → [新規] → [Java プロジェクト] を選択します (画面 ①-3)。

画面①-3 プロジェクトを新規作成するメニュー



- ② 表示された「新規Javaプロジェクト」ダイアログの「プロジェクト名」欄にプロジェクト名 (注①-9) を入力して、[完了] ボタンをクリックします (画面①-4)。

画面①-4 「新規Javaプロジェクト」ダイアログ

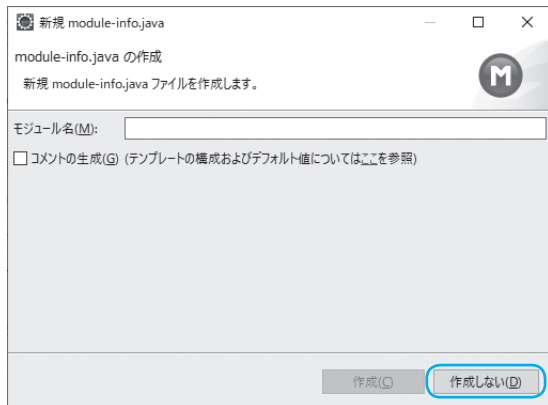


注①-9

本書ではプロジェクト名を「01-01」のように「〈章番号〉 - 〈通し番号〉」という形式で統一しています。

- ③ 表示された「新規 module-info.java」ダイアログの「作成しない」ボタンをクリックします (画面①-5)。

画面①-5 「新規 module-info.java」ダイアログ



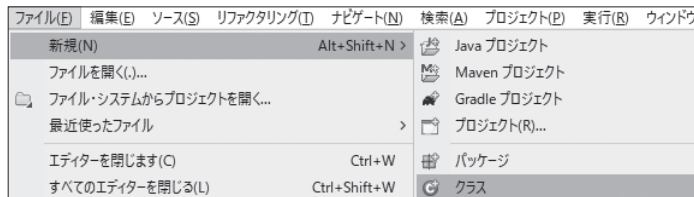
■ステップ2：プログラムコードの作成

続いて、プロジェクトの中にプログラムコードを作成します。Java 言語では、プログラムをクラス（第5章で説明します）という単位で作成していきます。Java 言語で「プログラムを作成する」ことは「クラスを作成する」ことだと理解しましょう。

次のようにして、プロジェクトに新しいクラスを追加します。

- ① [ファイル] メニュー → [新規] → [クラス] を選択します (画面①-6)。

画面①-6 クラスを新規作成するメニュー



- ② 表示された「新規Javaクラス」ダイアログの「名前」欄にクラス名を入力し、「完了」ボタンをクリックします (画面①-7)。クラス名はプログラムの名前とってください。好きな名前を設定できますが、ここでは**FirstExample**とします。

画面①-7 「新規Javaクラス」ダイアログ

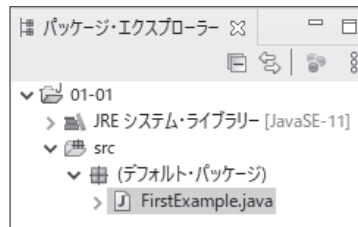


- ③ すると、[パッケージ・エクスプローラー] ビューが画面①-8のような状態になります。「01-01」というプロジェクトの中の「FirstExample.java」というファイルにプログラムコードが作成されたことを示しています(注①-10)。また、画面①-9のような[エディター・エリア]ビューが表示されます。プログラムコードの一部がすでに入力されていますが、これは入力されたクラス名を元に、Eclipseが自動的に入れたものです。

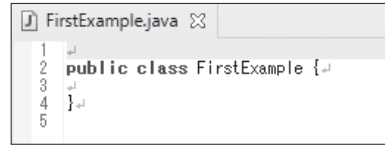
注①-10

srcは、プログラムコードのファイルが保存されているフォルダです(24ページで説明します)。(デフォルト・パッケージ)については実践編で説明します。

画面①-8 「FirstExample.java」というファイルが追加された



画面①-9 [エディター・エリア] ビューにプログラムコードの一部が入力されている



なお、[エディター・エリア] ビューのタブの右端にある [×] をクリックすると、そのタブが閉じられます。再度開くには、[パッケージ・エクスプローラー] ビューでプログラムコードのファイル（画面①-8では「FirstExample.java」）をダブルクリックします。

- ④ いよいよプログラムコードを書いていきます。[エディター・エリア] ビューの中に、次のプログラムコードをキーボードで入力してみましょう。

```
public class FirstExample {
    public static void main(String[] args) {
        System.out.println("こんにちは");
    }
}
```

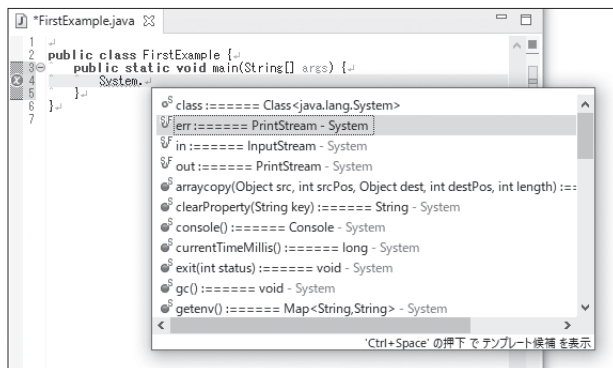
エディターの中では、プログラムコードの一部に色がついていたり、異なる書体が使われたりしますが、これらはプログラムコードを見やすくするためにEclipseが自動的に行ってくれています。入力する文字を間違えないように気をつけましょう。日本語以外の文字はすべて半角文字です。大文字・小文字の違いにも注意しましょう。

プログラムコードを入力している最中に、画面①-10のように突然コンテキストメニューが表示されて驚くかもしれません。これは、Eclipseがこれから入力すると予測されるプログラムコードの候補を表示しています（注①-11）。

注①-11

同じつづりの候補がいくつも表示されることがありますが、最初のうちは、入力したいつづりであればどれを選んでもかまいません。Java言語の学習や経験を積むうちに候補の意味がわかり、正確に選べるようになります。

画面●-10 入力中に表示されるコンテキストメニュー

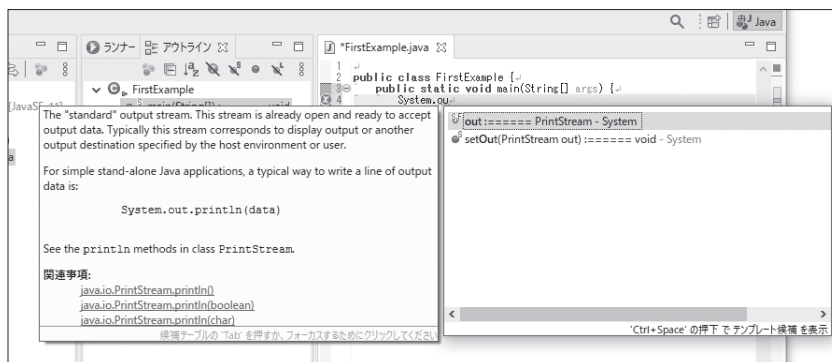


注●-12

コンテキストメニューが出たときに、入力したい候補を上下のカーソルキーで選択できます。
Enter キーで選んだ候補を入力できます。

これを無視してキーボードから入力続けてもよいですし、マウスカーソルで候補の中から入力したいものをダブルクリックしてもかまいません(注●-12)。なお、候補はキーボードから1文字打ち込むたびに絞られていきます。候補が多すぎるときには、選ぶ前に次の文字を入れてみましょう(画面●-11)。選択している候補の説明も同時に表示されます。

画面●-11 次の文字を入れると候補が絞られた



注●-13

インデント部分に入っている空白を **Tab** キーを押して増やしたり、**Delete** キーを押して減らしたりすることで、その行のインデントを調整することもできます。

プログラムコードの中の改行は **Enter** キーで行います。その際、Eclipseは自動的にインデントを行頭に入れてくれます。これもコードを見やすくするためのEclipseの機能です。また、「{」を入力した行を改行すると、次行のインデントが自動的に1段階深くなります。手でインデントを調整する場合、深くするには[ソース]メニュー→[右へシフト]を、浅くするには[ソース]メニュー→[左へシフト]を選んでください(注●-13)。

- ⑤ [ファイル] メニュー→ [保存] を選択してプログラムコードを保存します。
これでプログラムコードの作成は完了です。

■ステップ3：プログラムの実行

ステップ2で作成したプログラムを次のようにして実行します。

注①-14

2回目以降は[実行]メニュー→
[前回の起動を実行]を選択して
もかまいません。

- ① [実行] メニュー→ [実行] → [Java アプリケーション] (注①-14) を選択します。
② Eclipseのウィンドウ下部に[コンソール]ビューが現れ、「こんにちは」と表示されます(画面①-12)。このとき、裏ではEclipseがプログラムコードを自動的にコンパイルしています。これで、文字列を表示するプログラムが作成され、そして正しく実行されたことを確認できました。

画面①-12 [コンソール]ビューにプログラムの実行結果が表示された



■ プログラムコードの間違いを修正する

KEYWORD

- エラー

作成したプログラムコードに間違いが含まれていると、**エラー** (Error) が発生し、正しく実行できません。エラーには次の2種類があります。

KEYWORD

- コンパイルエラー
- ランタイムエラー
- 実行時エラー

- **コンパイルエラー** (Compile Error)
- **ランタイムエラー** (Runtime Error。「^{じっこうじ}実行時エラー」ともいいます)

■コンパイルエラー (Compile Error)

プログラムコードをコンパイルしているときに発生するエラーです。プログラムコードの記述に誤りがある場合に発生し、このエラーがあるとプログラムを実行できません。特定のキーワードのつづりミスや、文法上の誤りが原因となります。慣れないうちは、次のような基本的な入力ミスが考えられます。


- アルファベットや記号、空白文字を誤って全角で入力している
- 大文字と小文字を間違えている (たとえば、**System**という単語を **system**と

入力しては誤り)

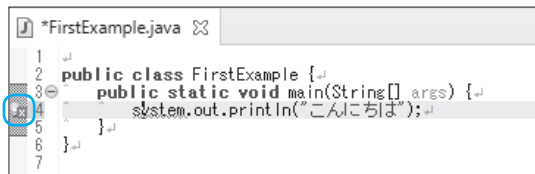
- 見た目の似た文字を誤って入力している



このような誤りをしないように、キーボードの入力モードを確認して、プログラムコードは半角アルファベットで入力するようにしましょう。空白部分は、半角の空白または **Tab** キーで入力します。全角の空白を使ってははいけません。そのほか、見た目が似ていて、入力を間違えやすいアルファベットと記号を以下に挙げておきます。

- 数字のゼロ「0」と、英小文字のオー「o」と、英大文字のオー「O」
- 数字のイチ「1」と、英大文字のアイ「I」と、英小文字のエル「l」
- 小括弧「()」と、中括弧「{}」と、大括弧「[]」
- セミコロン「;」と、コロン「:」
- ピリオド「.」と、カンマ「,」

Eclipseはプログラムコードを作成している最中、常に内容をチェックしていて、コンパイルエラーが発見された場合には、赤い下線と  マークで問題のある場所を示してくれます (画面●-13)。

画面●-13 Systemの最初のSを小文字にした誤りがある



 マークにマウスカースルを重ねるとエラーの内容が表示され、クリックすると解決方法の候補がいくつか提示されます。画面●-13はつづりの誤りなので、正しいつづりに直します。 マークの表示がなくなれば、プログラムコードは問題なくコンパイルされ、実行できます。

■ランタイムエラー (Runtime Error)

コンパイル時には発見されず、プログラムを実行している最中に見つかるプログラム上の問題のことです。文法上は間違いではなくても、命令を実行できない場合に発生します。たとえば「ファイルからデータを読み込もうとしたら、目的のファイルが存在しなかった」など、実行時になって初めてわかる要因で

発生します。Eclipseによって実行中のプログラムにランタイムエラーが発生した場合は、プログラムが強制終了され、[コンソール] ビューに赤い文字でエラーの情報が出力されます。

なお、本書の中で紹介しているプログラムコードで学習を進めている限り、ランタイムエラーで困ることはないはずです (注①-15)。

注①-15

エラーに対処するための詳しい説明は実践編の「例外処理」の章で説明します。

■ プロジェクトの保存場所

Eclipseで作成したプロジェクトは、本書の巻末付録Aの説明にあるように、最初の起動で指定したworkspaceフォルダの中に保存されます。先ほどは「01-01」という名前でプロジェクトを作成しましたので、「01-01」というフォルダがworkspaceフォルダの中に見つかるはずです。

この01-01フォルダの中には「src」フォルダと「bin」フォルダがあります。srcフォルダには、作成したクラスのプログラムコードが「クラス名.java」というファイル名で保存されています。先ほど作成されたのはFirstExample.javaです。一方、「bin」フォルダには、コンパイルしてできたバイトコードが「クラス名.class」という名前のファイルに保存されています。先ほど作成されたのはFirstExample.classです。

Eclipse上でプログラムを作成しているときには、こうしたファイルを直接扱うことはありませんが、作成したプログラムを他人に渡すときや、ほかの記録媒体 (メディア) にバックアップするときのために、この保存場所を知っておきましょう。

ワン・モア・ステップ！

.java ファイルと .class ファイル

Java 言語では、拡張子が「.java」のファイルにプログラムコードを保存します。
このとき、プログラムコードのファイル名はクラス名と同じにします。先ほどは、FirstExample クラスのプログラムコードをFirstExample.javaに保存していました。

一方、これをコンパイルしてできるバイトコードを記録しているのは、拡張子を.classにしたファイルです。先ほどの例ではFirstExample.classでした。このことから、Java 言語のプログラムは「クラス (class)」と密接な関係を持っていることがわかるのではないのでしょうか。

この .java ファイルと .class ファイルの意味と関係はJava言語を使うときの基礎知識でもあるので覚えておきましょう。クラスについての説明は第5章で行います。

登場した主なキーワード

- **Eclipse**：Java 言語によるプログラムの作成を支援するアプリケーションの1つ。
- **プロジェクト**：プログラムコードを管理する単位。
- **コンパイルエラー**：コンパイル時に検出されるプログラムコードに含まれる間違い。
- **ランタイムエラー**：プログラム実行時に検出されるプログラム上の問題。

まとめ

- Eclipseを使ってプログラムを作成する方法と、実行する方法を学びました。
- プログラムに誤りがあると、エラーが発生します。エラーにはコンパイル時に検出されるコンパイルエラーと、実行時に発生するランタイムエラーがあります。
- コンパイルエラーが検出されなければプログラムを実行できます。