

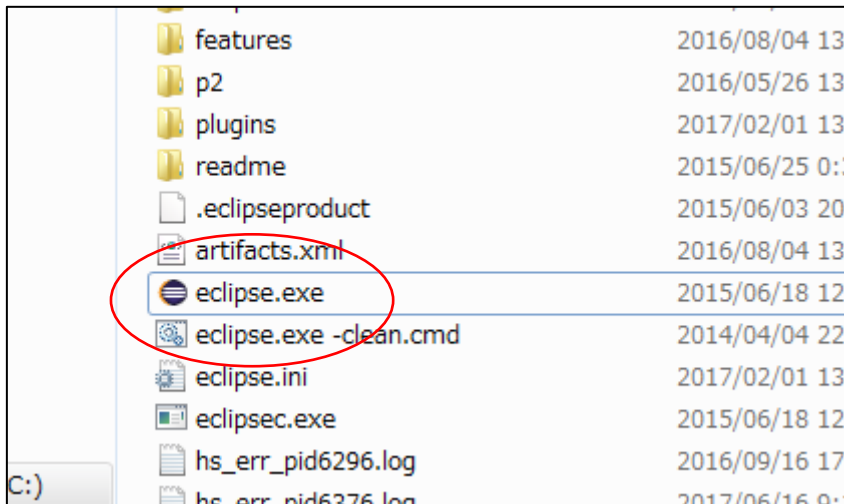
エクリプスについての 入門書

第1版

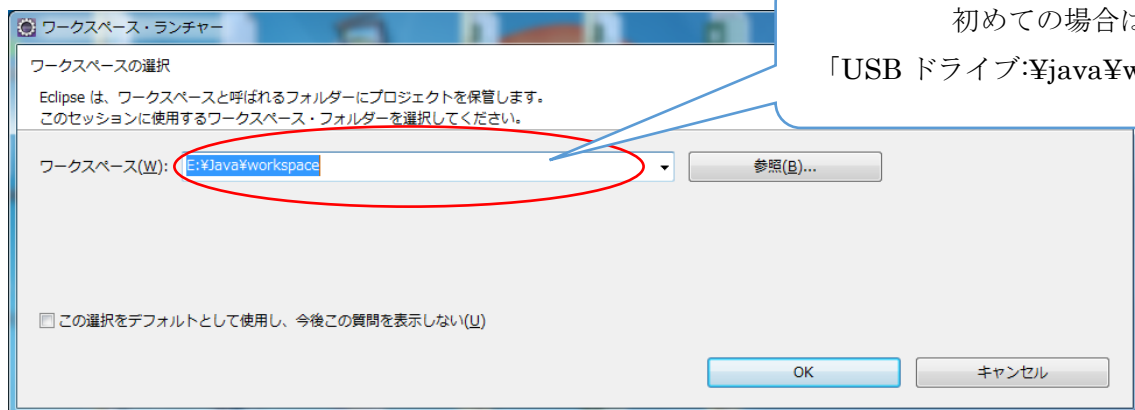
1. エクリプスの起動.....	3
2. プロジェクトを作ろう	5
3. プログラムを作って実行してみよう	7
クラスの作成	7
プログラムの記入	9
プログラムの実行	11
4. 便利な機能	12
コンパイルエラー	12
名前の変更	13
用語集	15

1. エクリプスの起動

エクリプスは PC の「C:\¥pleiades-4.6.3-java-win-32bit-jre_20170422」にあります。
ダブルクリックで起動します。



しばらくすると、ワークスペースを選択する画面になります。



初めて起動する時は「ワークスペース」の所に「USB のドライブ:¥java¥workspace」と入力しましょう。入力した OK ボタンをクリックします。

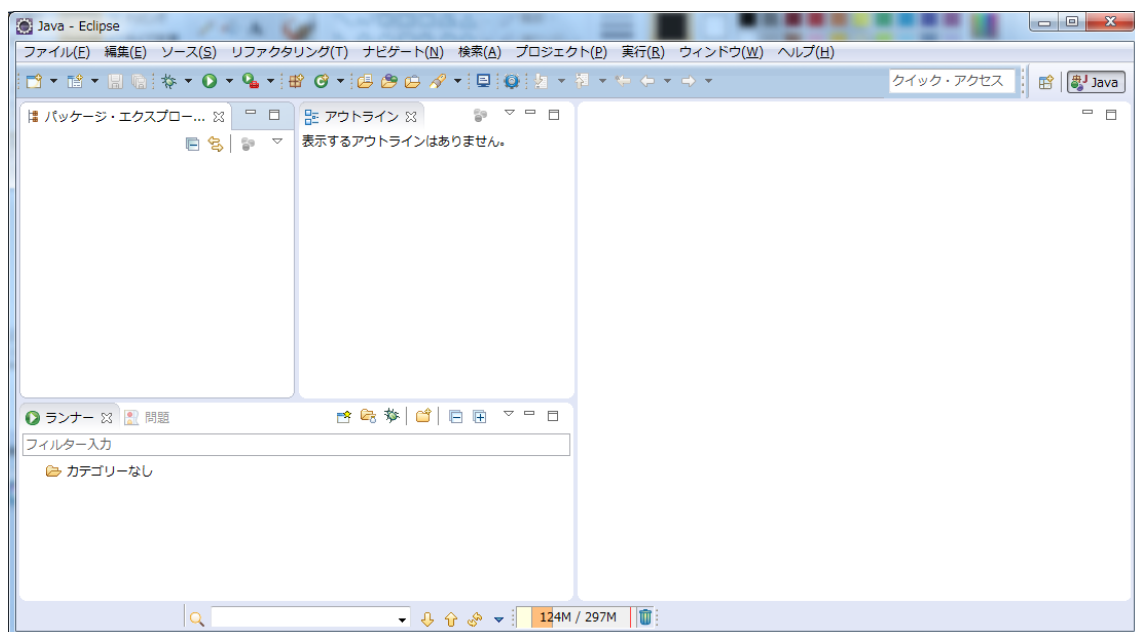
例) USB が E ドライブだったら

「E:¥java¥workspace」となる。

次のような画面が出れば OK です。

すでにワークスペース内にプロジェクトがある場合は、[プロジェクト](#)が表示されます。

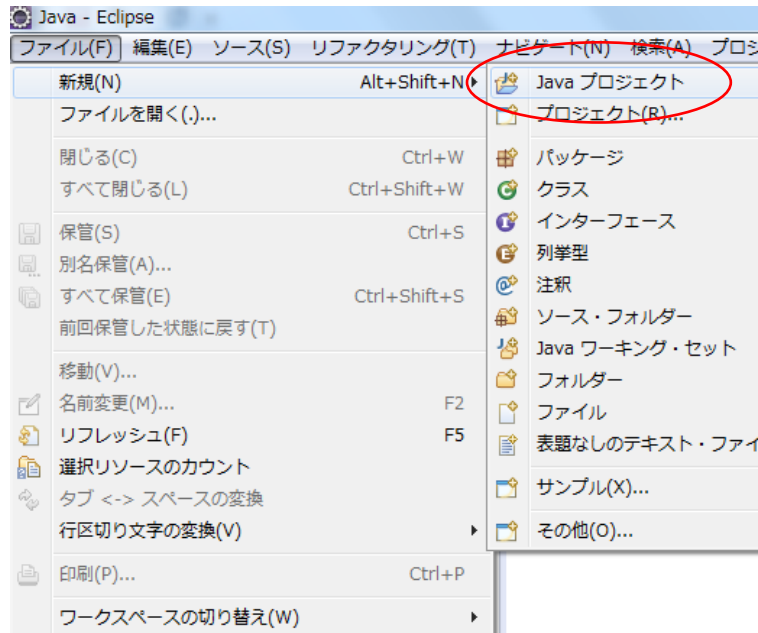
※初回は、まだワークスペース内にプロジェクトが無いので、↓のようになります。



2. プロジェクトを作ろう

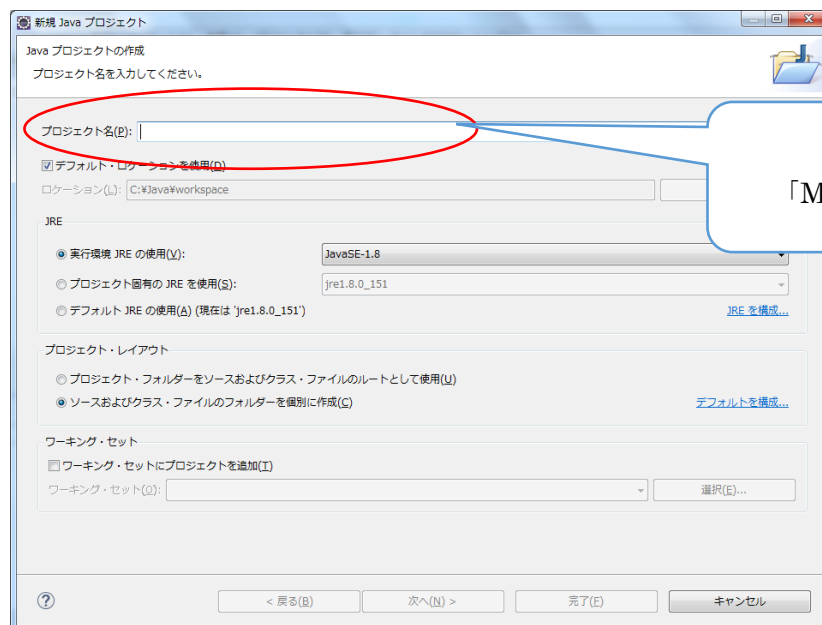
Java のプロジェクトを作ります。

左上のメニューから「ファイル」－「新規」－「Java プロジェクト」とポイントします。



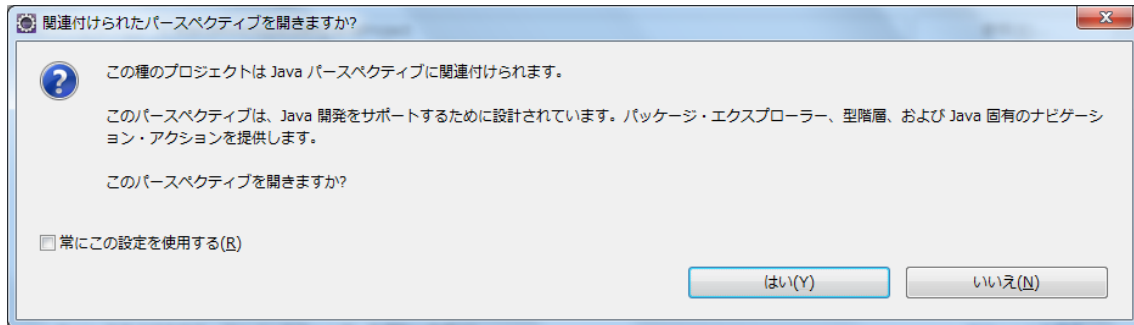
❗ もし、「Java プロジェクト」が無い場合は、一度「プロジェクト(R)」を選択して表示されたダイアログから「Java プロジェクト」を選択します。

プロジェクト名を入力して「完了」をクリック

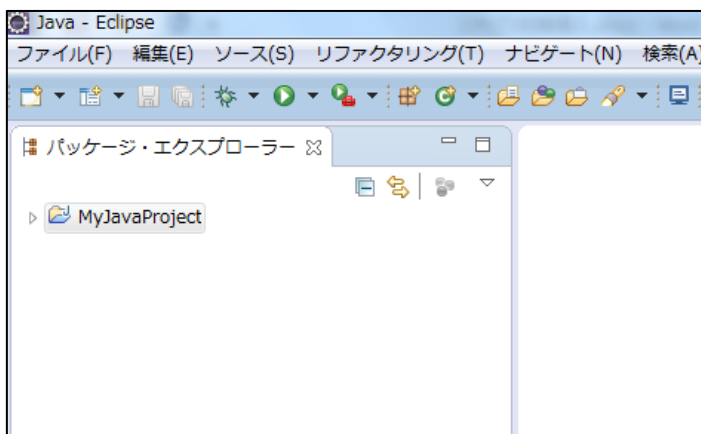


最初の方は
「MyJavaProject」と入力しよう

パースペクティブの関連づけダイアログが表示された「OK」で閉じる



このように、プロジェクトが作られれば OK



ちなみに、ワークスペース内にプロジェクトは複数作成可能

3. プログラムを作って実行してみよう

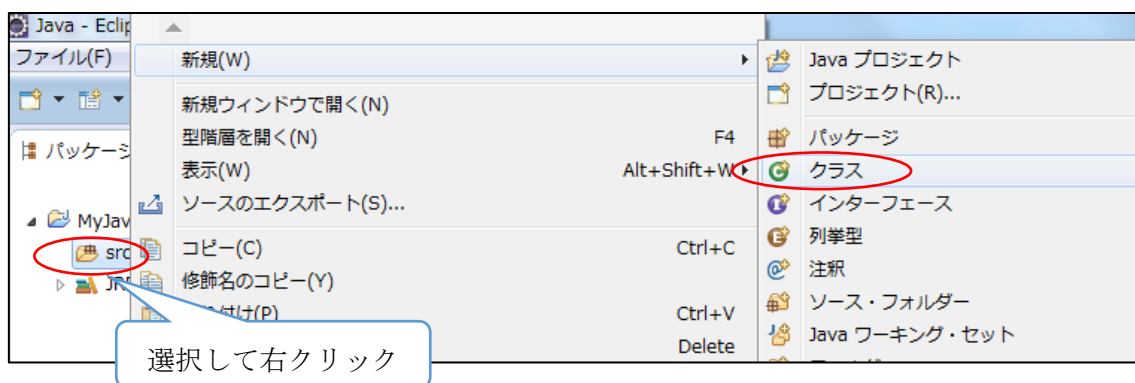
プロジェクトのツリーを開く



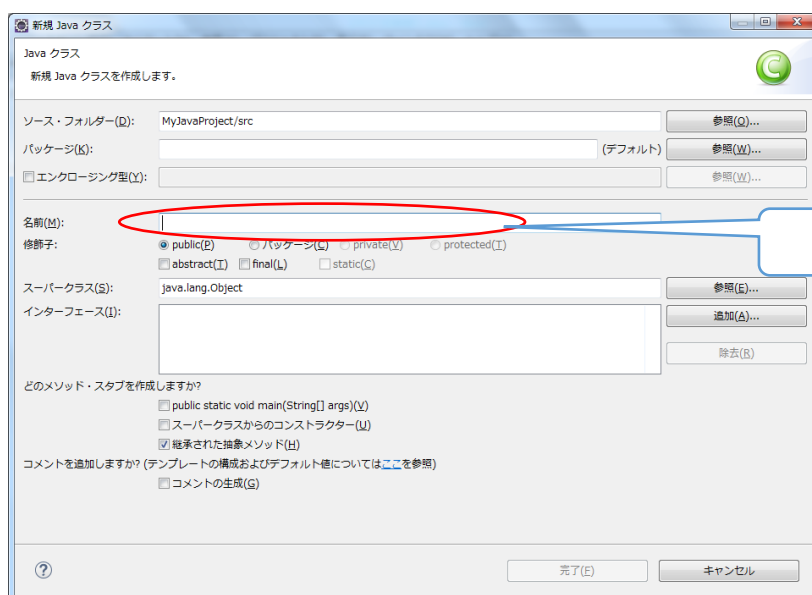
クラスの作成

プロジェクト内に、クラスを作ります。src を選択して右クリック

「新規」－「クラス」を選択する

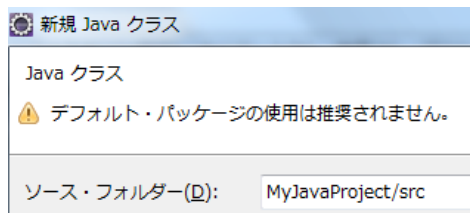


表示された画面で、クラス名を入力して完了をクリック



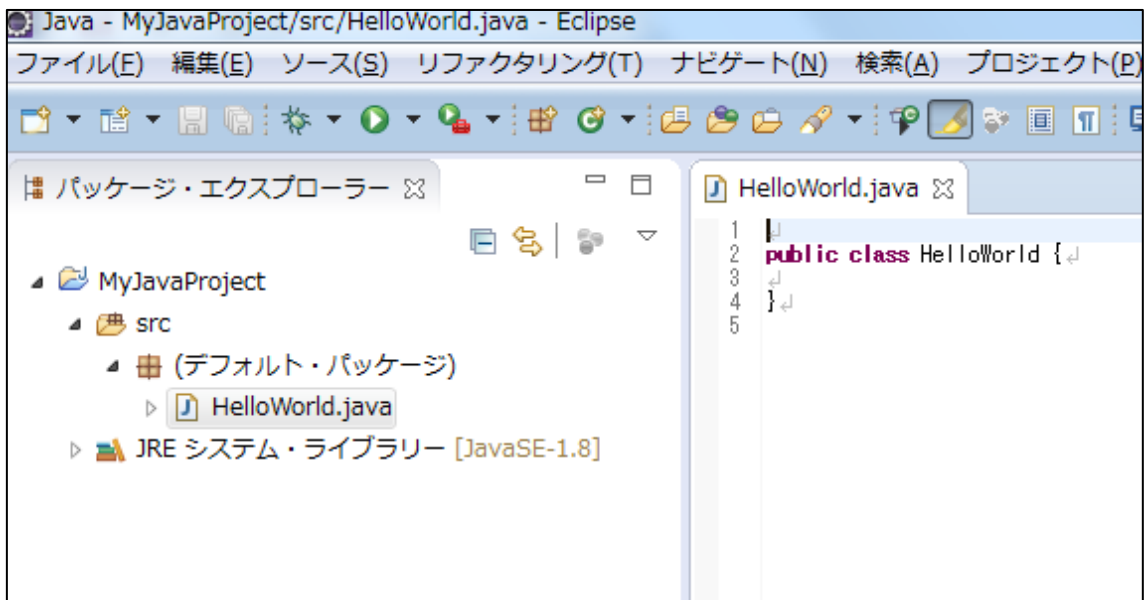


初めて場合、ダイアログの上部に



と表示されますが、無視してください。

↓の用になれば OK です。



プログラムの記入

エディタで実際にプログラムを書いていきます。

初めての時は、ひとまず↓を書いてみましょう。

A screenshot of an IDE window titled 'HelloWorld.java'. The code is as follows:

```
1  public class HelloWorld {
2      public static void main(String[] main){
3          System.out.println("hello world");
4      }
5  }
6
7
```

エクリプスには入力補完機能があります。

入力中に気付くと思いますが

例えば、↓の様に main メソッドの開始のカッコを入力してエンター機を押すと・・・

A screenshot of the IDE showing the first part of the code:

```
public class HelloWorld {
    > public static void main(String[] main){
    }
}
```

↓ エンターキー

A screenshot of the IDE showing the code after pressing the Enter key:

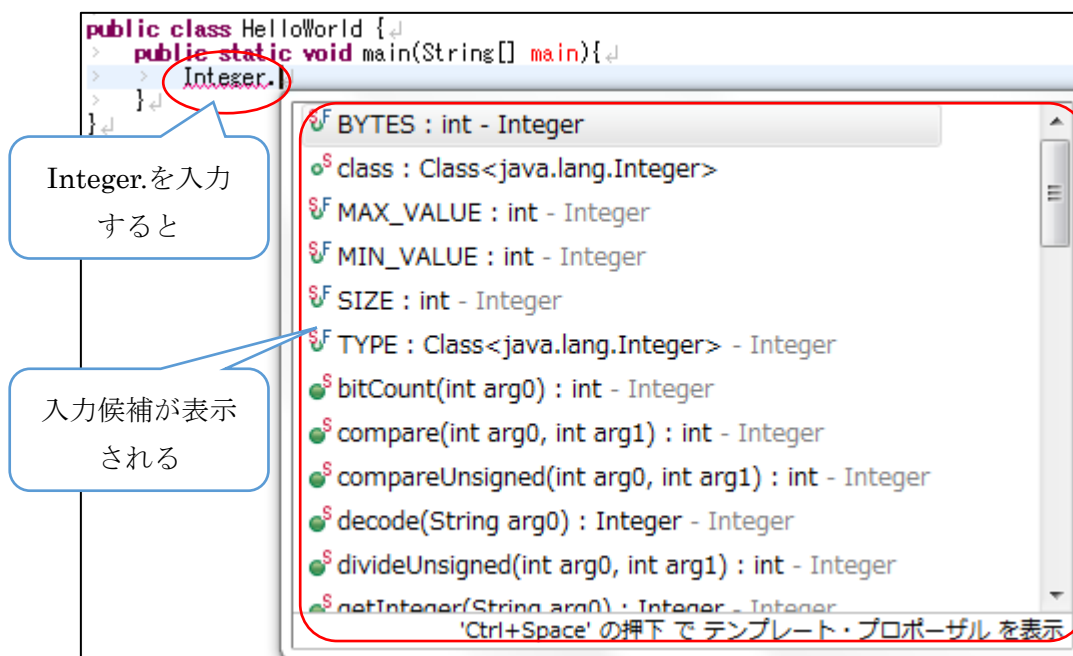
```
public class HelloWorld {
    > public static void main(String[] main){
    > }
    }
}
```

The closing brace '}' on line 3 is circled in red. A blue callout box points to it with the text '閉じカッコを自動で追加!'.

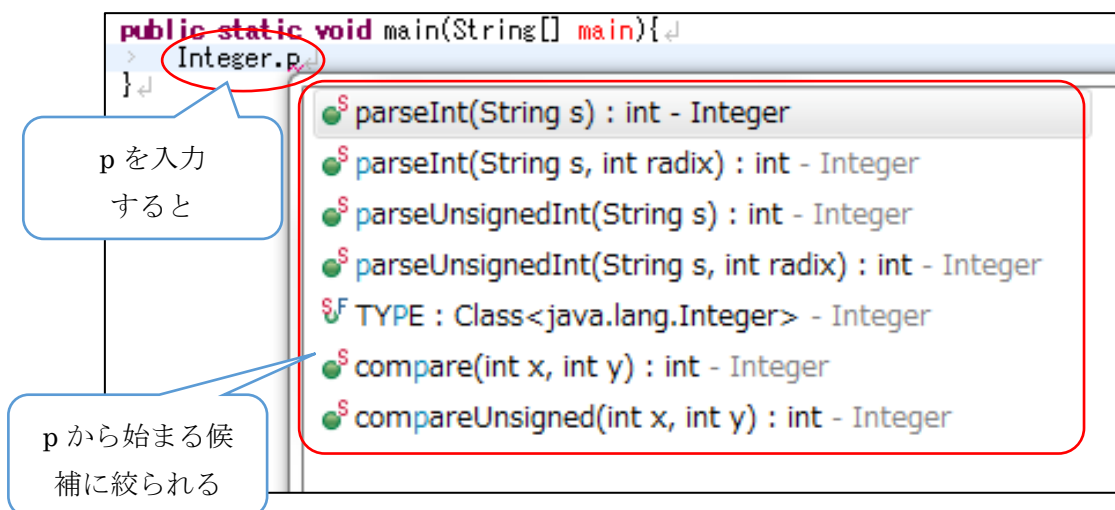
他にも for 文や if 文でも同様に自動で追加してくれます。

他にも、インスタンスやクラス名の後にドット（.）を入力すると、次に来る入力候補を一覧表示してくれます。

例えば、よく使っている文字列を数値に変換する `Integer.parseInt` を入力する時は「Integer.」と入力した時点で、候補を出してくれます。



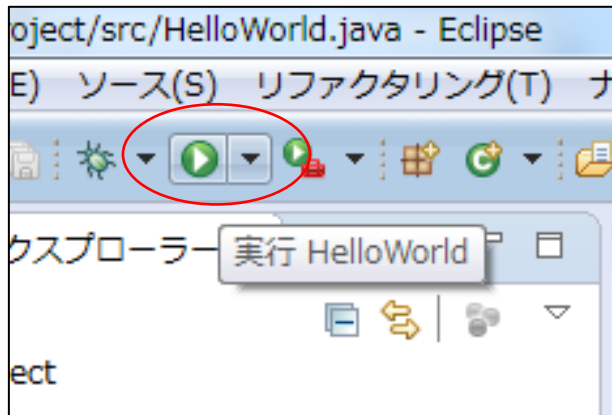
さらに、続けて「p」を入力すると「p」から始まる候補に絞り込めます。



入力したい候補が見つかったらダブルクリックするとエディタに反映されます。

入力補完を使って、効率よく入力しましょう。

プログラムの実行



実行するには、実行ボタンをクリックします

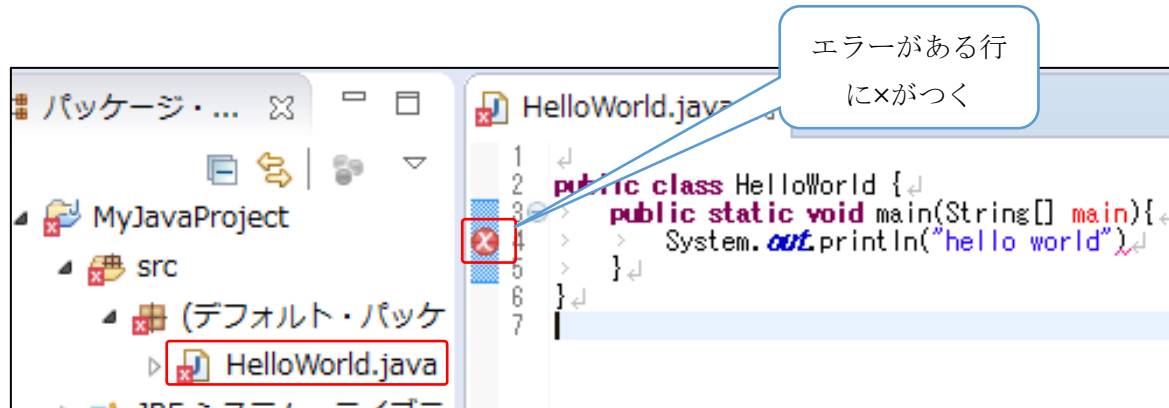
実行結果は下部の「コンソール」タブに表示されます。



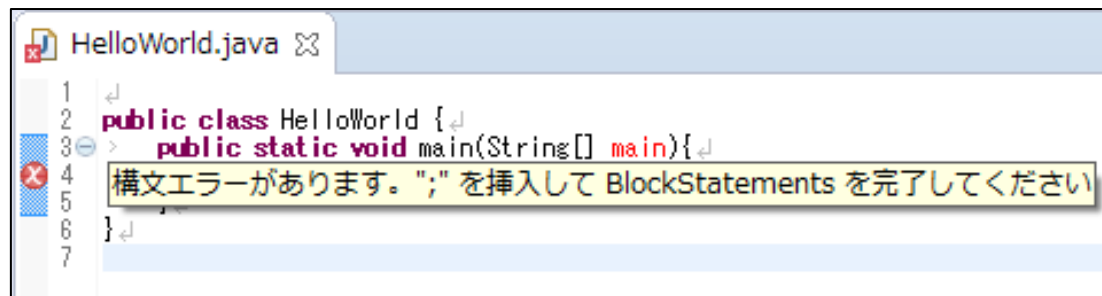
4. 便利な機能

コンパイルエラー

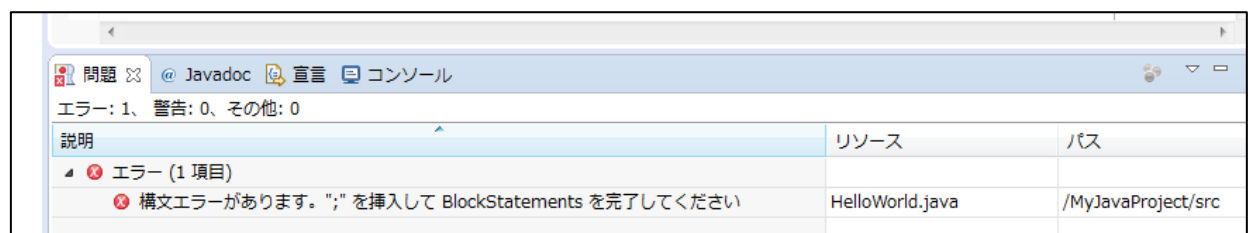
コンパイルエラーがあるときは、入力途中でもわかります。



X印にカーソルを合わせると、エラーメッセージが表示されます。

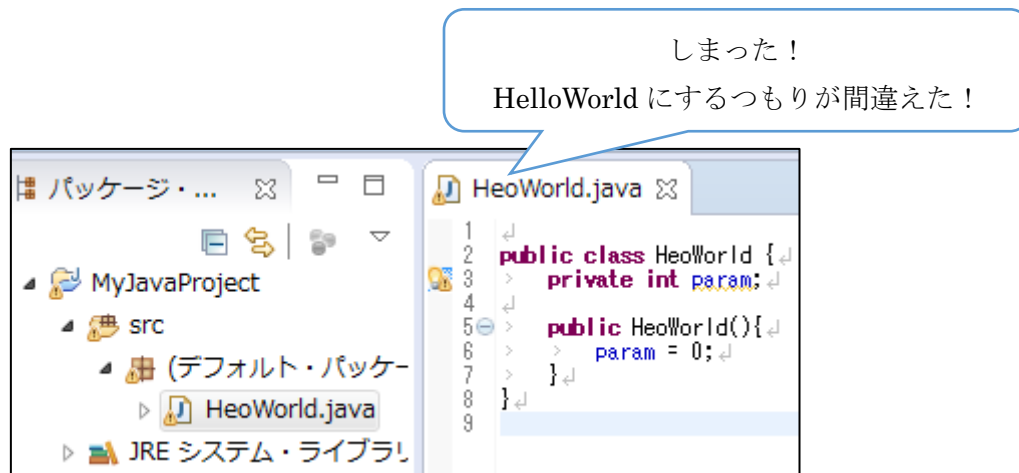


エラーは同様に、画面下部の「問題」タブにも表示されます。



名前の変更

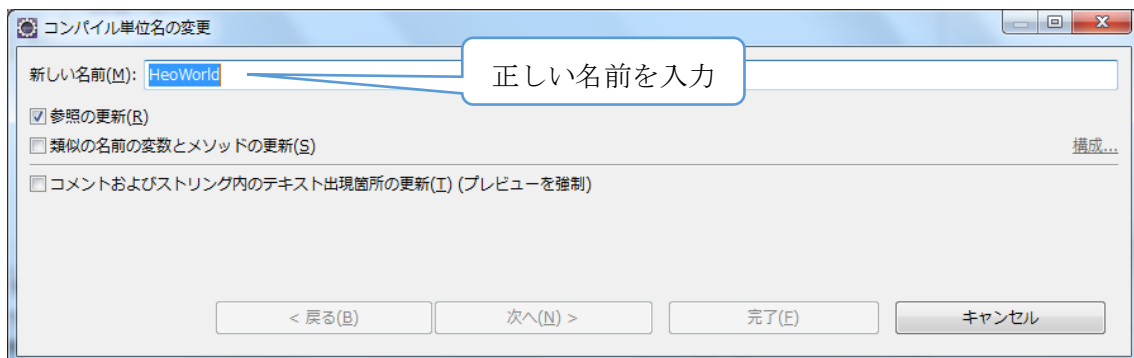
クラス名を間違えると、クラスメイトファイル名を書き直したり、色々面倒ですが
そういう時は、「リファクタリング」を使うと楽ちんです。



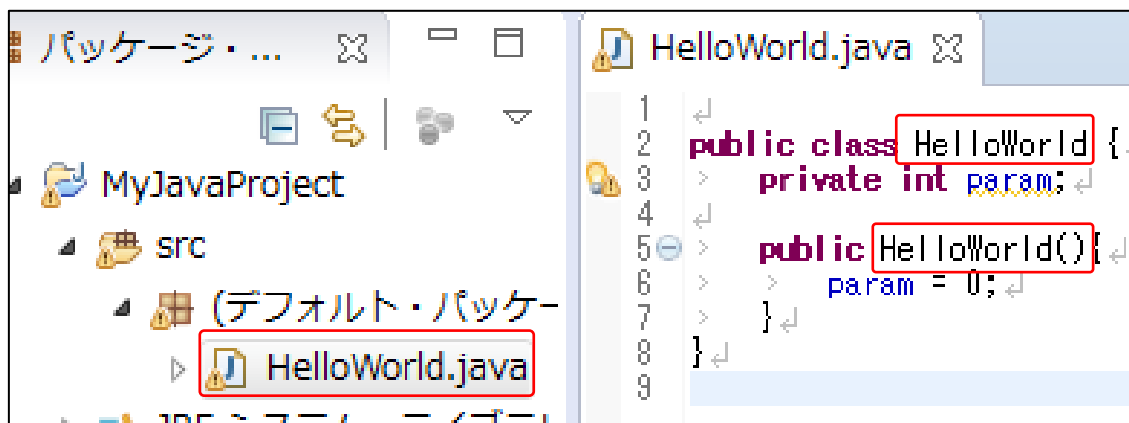
こういうときは「リファクタリング」間違えて名前をつけたファイルを選択して
右クリック「リファクタリング」－「名前の変更」



表示されたダイアログで正しい名前を入力して完了をクリックすると・・・



自動で、ファイル名、クラス名、コンストラクタ名まで修正されました！！



他にも、いろいろと便利な機能があります！

用語集

用語	説明
ワークスペース	プロジェクトの集合。すべてプロジェクトと設定を含む
プロジェクト	ソース ファイルの集合。通常は1つのプロジェクトで1つのアプリケーションとなる。
パースペクティブ	エディタやビューを組み合わせで配置したもの
プラグイン	後から機能を付け足すことが出来るライブラリのようなもの プラグインを追加でインストールすると、エクリプスの機能を拡大することが出来る