

# Javaプログラミング実習

## 18. デバッグ

株式会社ジーードライブ

# 今回学ぶこと

---

- Eclipseのデバッガを使ったプログラムのデバッグ方法

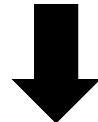
# デバッグとは

- ・ プログラムが期待通りに動作しない場合、その原因となっている問題（バグ）を修正する作業のこと
  - 多くの場合「バグ」は、文法上のミスやスペルミスではない

例：

158円と198円の商品購入代金(356円)に消費税8%を掛けて計算すると384円(小数点以下切捨)になるはずだが、383円と表示されてしまった。

原因：



各商品ごとに消費税を掛けて計算を行ったため、誤差が生じた。

# デバッガとは

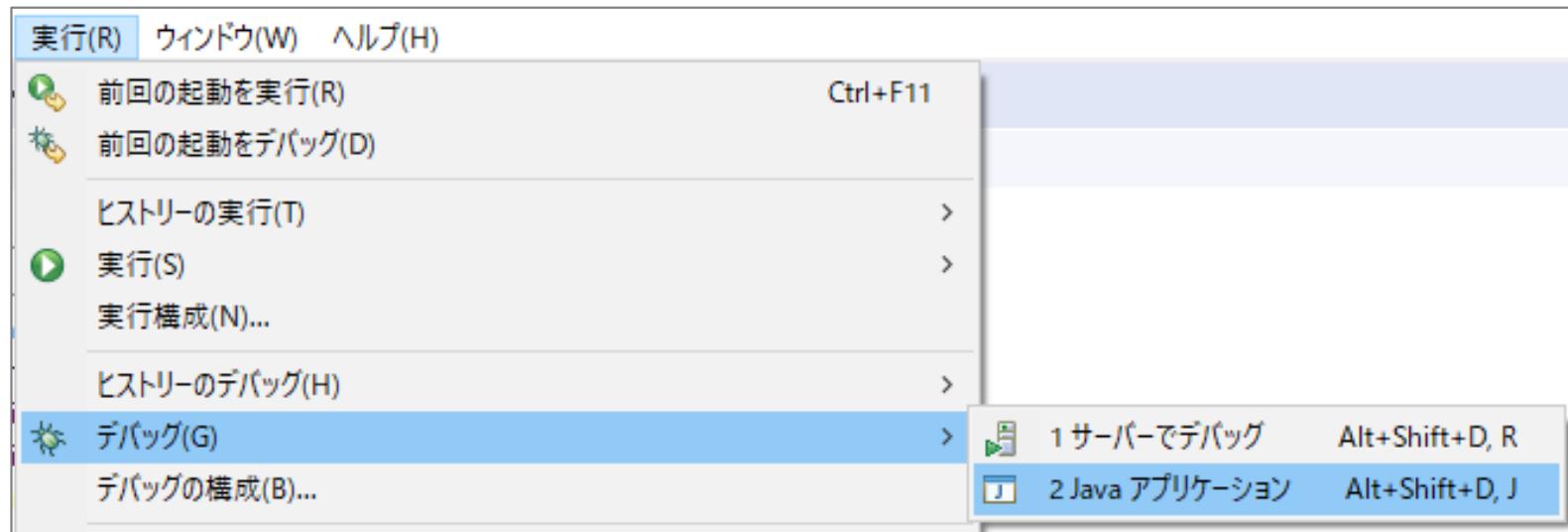
---

- ・ デバッガとは、プログラムを1行ずつ実行（ステップ実行）して、内部で使用する変数の状態や制御の流れを確認することができるツール
- ・ 通常は超高速で実行されるプログラムの動きをじっくり観察できるので、プログラムの問題発見がしやすくなる

自動で間違いを発見するツールではなく  
間違いの発見を助けてくれるツール

# Eclipseのデバッガの利用方法

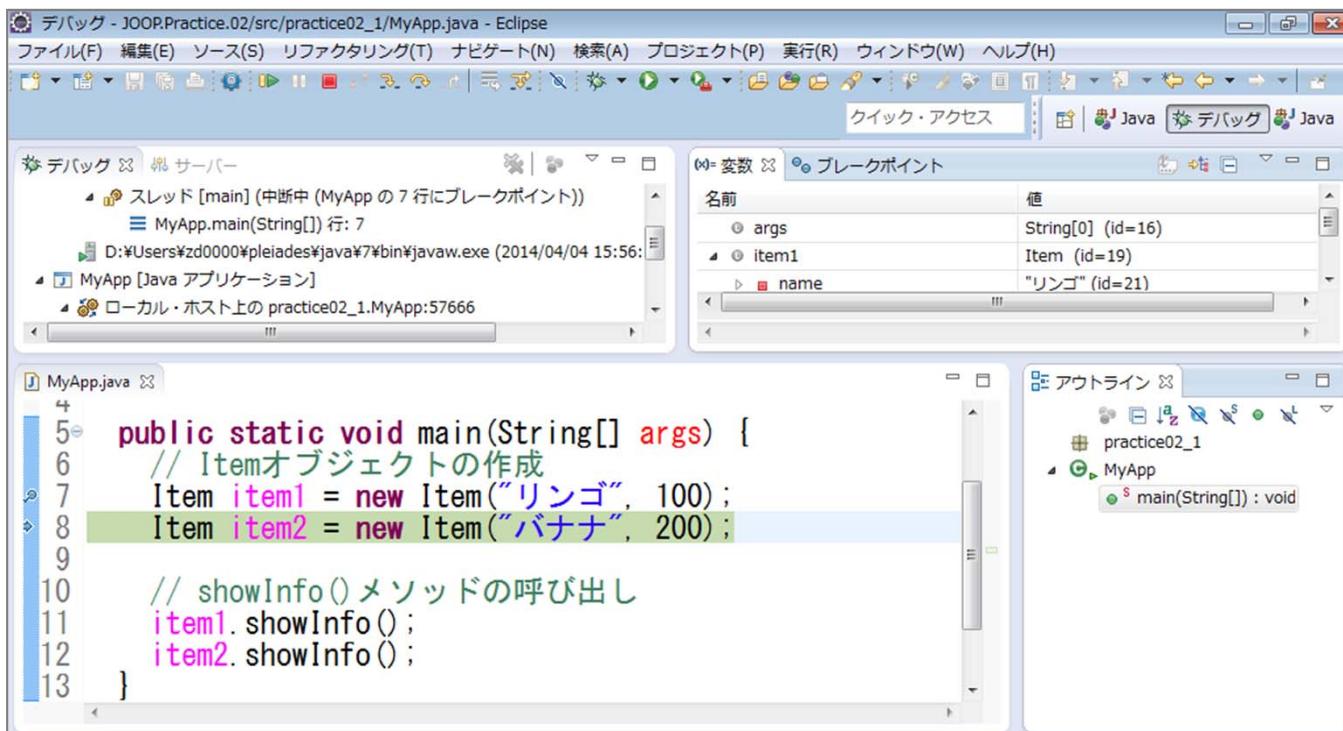
- デバッグの開始
  - デバッグ対象のファイルを選択し、  
実行 ⇒ デバッグ ⇒ Javaアプリケーション



あらかじめブレークポイントを設定しておく必要がある

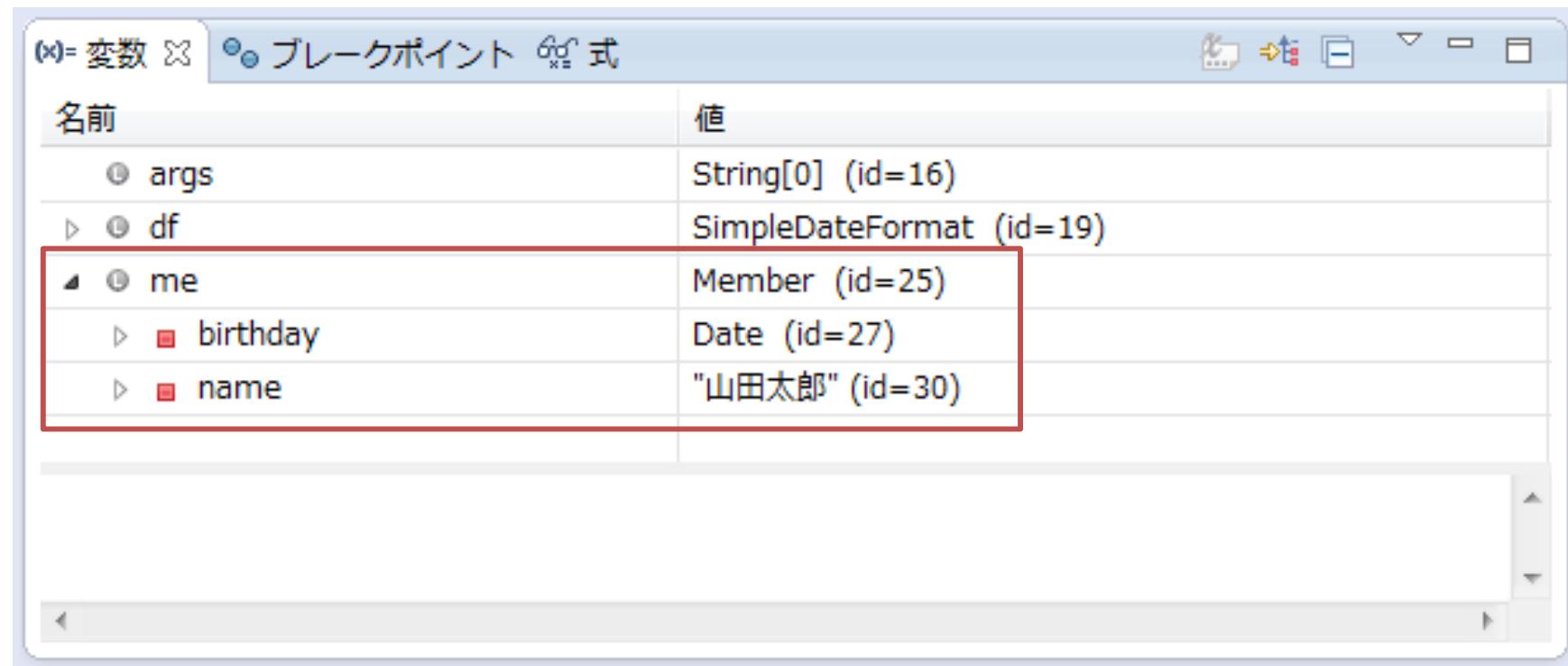
# デバッグ画面

- ・ デバッグを開始すると「デバッグ・パースペクティブ」に切り替わる
  - パースペクティブ切り替えの確認画面が表示される  
⇒ 切り替わらない場合は、手動で切り替える



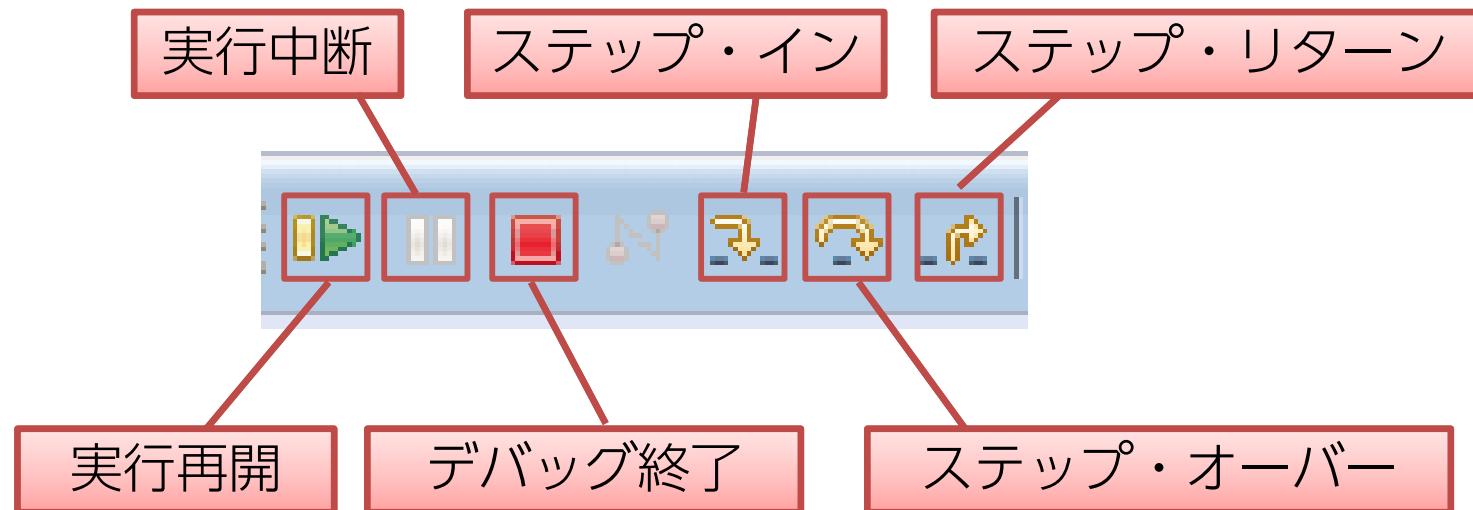
# 変数ビュー

- 変数ビューでは、実行中のプログラムの変数の状態を参照することができる
- 必要に応じて、変数の値を編集することも可能



# デバッガの制御ボタン

- プログラムの実行を進めたり、一時停止するなどの制御を行うことができる



# 実行再開と一時停止

- ▶ (実行再開)ボタンを押すと、次のブレークポイントまでのプログラムを実行する
  - ブレークポイントで一時停止の状態になる
- ⏸ (実行中断)ボタンを押すと、その瞬間に実行していた行でプログラムを一時停止する

The screenshot shows a Java code editor with the following code:

```
7
8 public static void main(String[] args) throws ParseException {
9     SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
10    Member me = new Member("山田太郎", df.parse("1990-01-01"));
11    me.showInfo();
12
13    System.out.println(me);
14
15    Item f1 = new Item("リンゴ", 100);
16    f1.showInfo();
17
18    Item f2 = new Item("バナナ", 200);
19
20 }
```

Annotations in red boxes:

- 次に実行する行 (The next line to be executed)
- ブレークポイント  
ダブルクリックで生成／削除できる (Breakpoint)  
Double-click to generate/delete

A red arrow points from the "ブレークポイント" annotation to the line "System.out.println(me);". Another red arrow points from the "次に実行する行" annotation to the line "13".

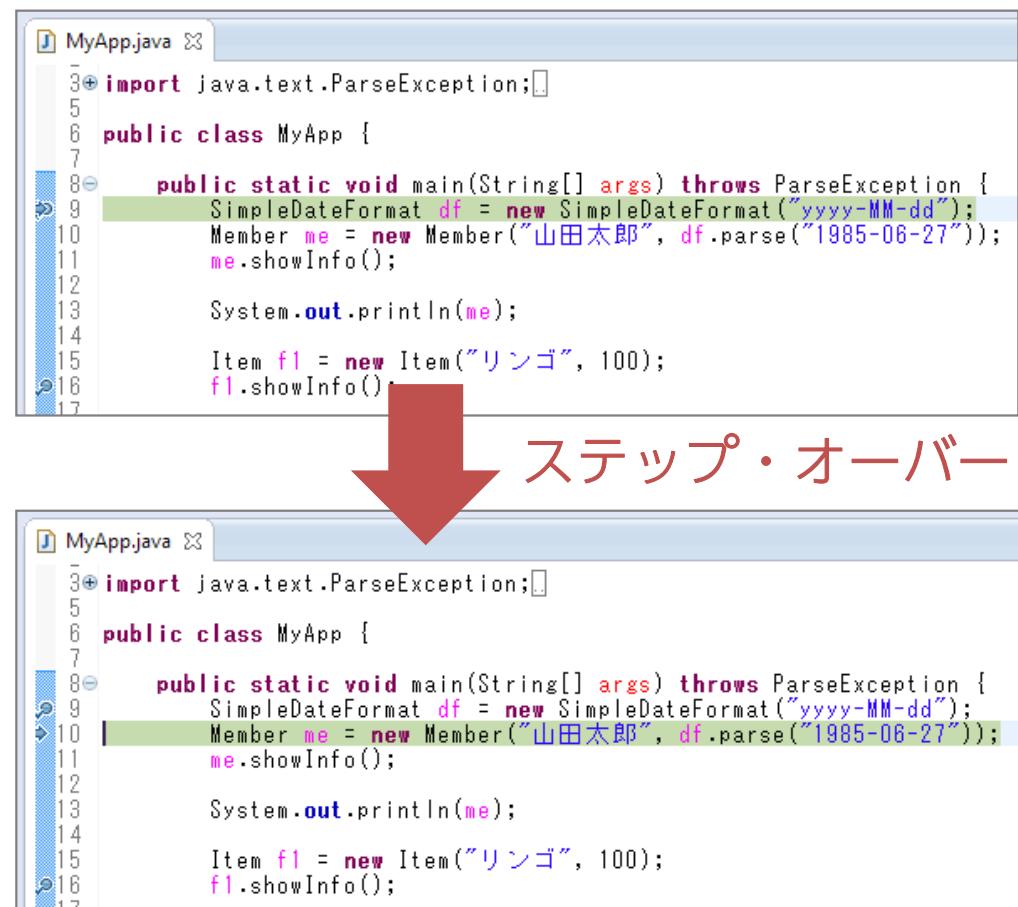
# ステップ実行機能

---

- プログラムは通常高速に実行されるが、デバッガを使用するとプログラムを一時停止して、1行ずつプログラムの実行を進めることができる
- ステップ・オーバー
  - メソッドの呼び出しがあるステートメントではメソッドを実行するがメソッド内には入らない
- ステップ・イン
  - メソッドの呼び出しがあるステートメントではメソッドの中に入っていく
- ステップ・リターン
  - 現在実行中のメソッドを最後まで処理しメソッドを抜ける

# ステップ実行機能

- デバッグを開始した直後は、最初のブレークポイントで実行が一時停止する
- この状態から、（ステップ・オーバー）ボタンをクリックすると、先頭行を実行し、次の行の実行直前で一時停止する



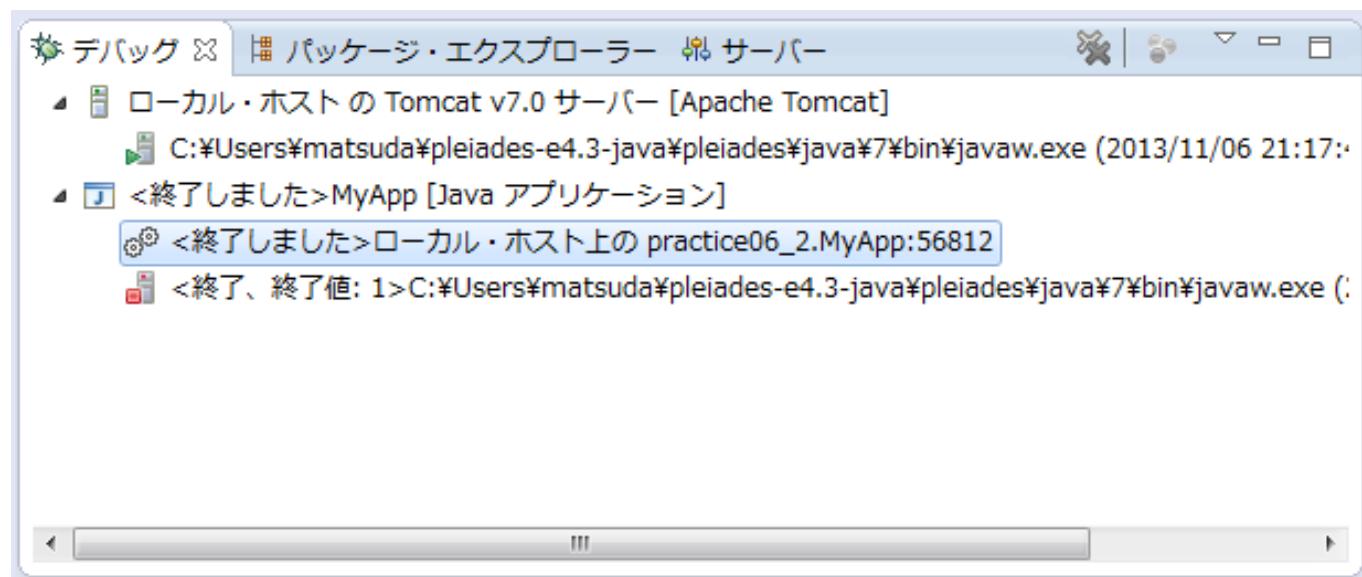
The screenshot shows two instances of a Java code editor for `MyApp.java`. In the top instance, line 9 is highlighted with a green background, and the cursor is at the start of line 10. A red arrow points down to the bottom instance, where line 10 is now highlighted. Both instances show the same code:

```
3+ import java.text.ParseException;
4
5 public class MyApp {
6
7
8+ public static void main(String[] args) throws ParseException {
9     SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
10    Member me = new Member("山田太郎", df.parse("1985-06-27"));
11    me.showInfo();
12
13
14
15    Item f1 = new Item("リンゴ", 100);
16    f1.showInfo();
17}
```

A red arrow points from the top code window to the bottom code window, indicating the progression of the step-over operation.

# デバッグの終了

- デバッグ中に  (デバッグ終了)ボタンを押すと、デバッガが停止する
  - デバッグ作業終了時は必ず停止ボタンを押す



デバッガ停止後のデバッグビューの様子

# パースペクティブの切替え

- デバッグを終了した時点では「デバッグ」パースペクティブのままなので、ワークベンチ右上の「Java」パースペクティブをクリックしてパースペクティブを元に戻す

