

# eclipseのデバッグ機能の使い方

作成日: 2007.04.16

作成したプログラムが自分の意図した通りに動かない場合、そのプログラムにはバグ(不具合)がある といひ、このバグを発見し取り除くことをデバッグといひます。

eclipseではデバッグを支援する機能が備わっており、この機能を使えば、プログラムを1行ずつ実行させたり、実行中に変数の値を調べたりすることができ、バグの発見をスムーズに行うことができます。

この文書ではeclipseのデバッグ機能の基本的な使い方を解説します。

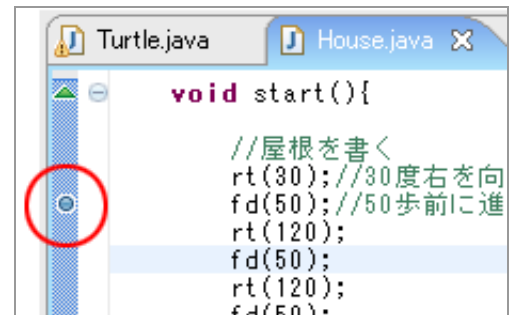
## デバッグの手順

### 1. ブレークポイントをつける

プログラムを停止させたい行に「ブレークポイント」をつけて指定します。  
エディタの指定したい行の左端でダブルクリックでブレークポイントを設定します。  
もう一度ダブルクリックでブレークポイントを解除します。  
デバッグ実行時には、ブレークポイントがついている行(を実行する直前)でプログラムが停止します。

### 2. デバッグ実行をする

デバッグしたいプログラムソースを表示している状態で、メニューバーから「実行」→「デバッグ」→「Java アプリケーション」を選択します。  
デバッグ実行時にブレークポイントの行にさしかかるとプログラムが停止し、自動的にデバッグ画面(デバッグパースペクティブ)に切り替わります。



ブレークポイントがついた状態

### 3. 停止した部分で変数の値を確認したり、1行ずつ実行して動作を確認する。

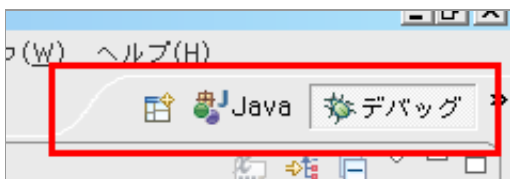
ステップイン、ステップオーバー、ステップリターンの機能や、「変数」ビューで値を確認し、バグを発見します。

### 4. 問題のある部分(バグ)を発見したら、コードを修正する

バグを発見したら、デバッグ実行を終了し、コードを書き直して、再度動作検証をします。無事、正常に動作するようになったらデバッグ完了です。正常に動作しない場合は手順1に戻ってデバッグを続けます。

## デバッグパースペクティブ

### パースペクティブ

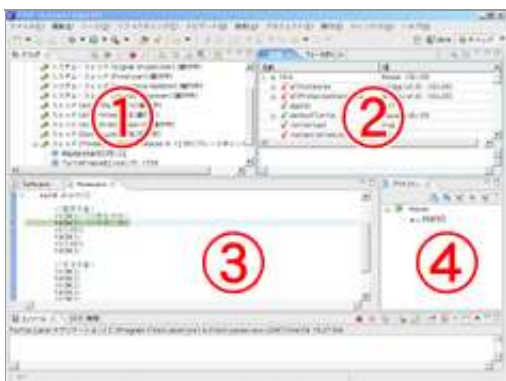


eclipseを使うときの目的に応じた画面構成(ビューの組み合わせと配置)のことをパースペクティブといひます。eclipseでは目的に応じて、様々なパースペクティブが用意されており、簡単に切り替えて使うことができます。

普段Javaプログラムを作成するのに使っているのが「Javaパースペクティブ」です。デバッグのときに自動的に切り替わったのが「デバッグパースペクティブ」です。

パースペクティブを切り替えるには画面右上のパースペクティブタブを操作します。デバッグが済んだら、Javaパースペクティブに戻しておくといひでしょう。

### デバッグパースペクティブ(デバッグ画面)の使い方



#### ①「デバッグ」ビュー

デバッグ機能を使うことができます。プログラムを1行ずつ進めたり、メソッドの中に入るなどことができます。(詳細後述)

#### ②-1「変数」ビュー

プログラムの現在の(停止した時点の)変数の値を表示します。(詳細後述)

#### ②-2「ブレークポイント」ビュー

ブレークポイントがどこにつけられているか一覧表示し、設定/除去を編集することができます。

#### ③「エディタ」ビュー

現在どの行を実行しているかを緑色のハイライトで表示します。

普段のエディタと同様に使えます。

④「アウトライン」ビュー

現在停止している位置をメソッドのネスト構造で表示します。

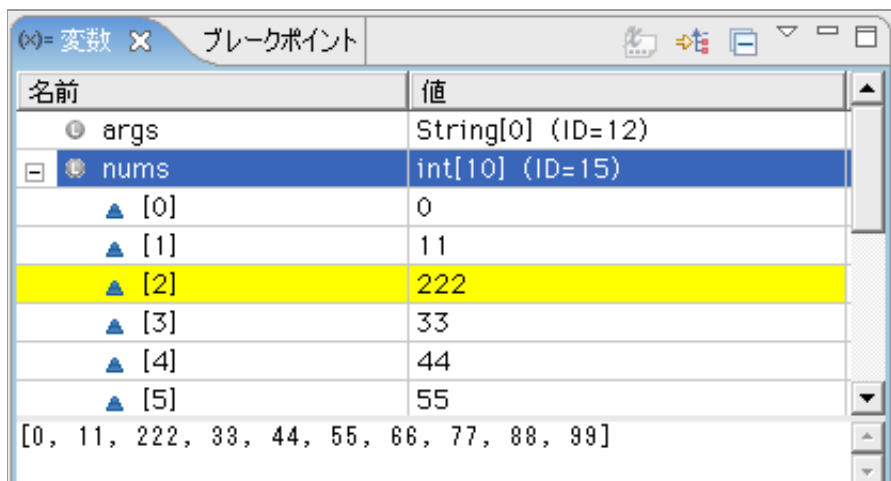
## デバッグ機能の使い方

### 「デバッグ」ビュー



再開	ブレークポイントが見つかるまで、プログラムを実行します。
中断	プログラムの動作中、現在実行しているところで停止します。
終了	プログラムを終了します。
ステップイン	現在のコード中のメソッドの中に入り、メソッド内の最初のコードの実行前で停止します。 コード中にメソッド呼び出しがない場合、ステップオーバーと同じ働きをします。
ステップオーバー	現在のコードを実行して、次のコードの実行前で停止します。
ステップ・リターン	現在の(スタックに積まれている)メソッドが終了するまでプログラムを実行し、次のコードの実行前で停止します。

### 「変数」ビュー



プログラムが停止した時点での、変数の名前と値を表示します。  
ここで表示される変数は、プログラムが停止している時点で参照できる変数です。  
つまり、

- ・実行中メソッド内のローカル変数
- ・実行中メソッドを持つオブジェクトのフィールド

です。

クラス型の変数値は+ボタンがついており、展開すると、そのオブジェクトのもつフィールドも確認することができます。また、配列型の変数値も+ボタンがついており、展開すると、その配列の要素も確認することができます。

デバッグ実行を1ステップずつ進めて変数に変更があった場合、変更された変数は黄色でハイライトされます。

変数の行を選択すると、下のスペースに変数の詳しい値(変数.toString()の値)が表示されます。この機能で配列やリストの中身を簡単に調べることができます。

下のスペースから直接任意のコードを実行することもできます。(コードを書き、文字列を選択して右クリック→「実行」)

### ショートカットキー

前回の起動をデバッグ	F11
前回の起動を実行	Ctrl+F11
ステップオーバー	F5
ステップイン	F6
ステップ・リターン	F7