<http://help.eclipse.org/juno/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2Ftasks%2Ftask-stepping.htm>

当一个线程被挂起时，程序可以一行一行的执行，如果执行一步操作时遇到一个断点，执行会在断点处暂停。

**Step over**

1. 在Debug透视图的编辑器中突出显示该堆栈帧（stack frame）中的当前执行流所在的行。
2. 在view Toolbar中单机Step OverStep Over按钮，或者按下F6，那么当前选择的行将被执行并且当前线程挂起到下一个可执行的行

**Step Into**

1. 在Debug透视图的编辑器中突出显示该堆栈帧（stack frame）中的当前执行流所在的行。
2. 在view Toolbar中单机Step IntoStep Into按钮，或者按下F5，在当前选定的行要执行的下一个表达式被调用，线程讲挂起到下一个可执行行中被调用的方法中。

**Step Into Selection**

1. 在Debug透视图的编辑器中突出显示该堆栈帧（stack frame）中的当前执行流所在的行。
2. 在java Editor中把鼠标放在当前执行行所在的方法上，
3. 单机Step Into selection按钮，或者按下Ctrl-F5，继续执行，执行流讲执行到所选择的方法中

注：在view Toolbar中没有按钮(myeclipse8)，需要在java editor中鼠标右键菜单中选择

**Step with filters**

1. 在Debug view toolbar切换到Use Step Filters Use Step Filters按钮，或者按下 shift+F5,那么每一步操作（over,into,return）将应用在 java>Debug>Step Filtering中定义的过滤规则。如果碰到符合过滤规则的代码，线程讲继续执行，直到未过滤的代码执行完或者碰到下一个未过滤的断点。
2. 这种情况适合当调用了过多的第三方库的时候，跳过哪些类，如：jdk,spring,hibernate等。

**Step Return**

1. 在Debug透视图的编辑器中突出显示该堆栈帧（stack frame）中的当前执行流所在的行。
2. 在Debug view toolbar切换到Step Return Step Return按钮，或者按下 shift+F7,直到下一个return语句在当前方法中被执行，执行下一个可执行的行挂起（想当于执行了step into进入某个方法后，退出此方法，返回到该方法的调用处）。

**Run to line**

1. 当一个线程被暂停，它是可以继续执行，直到指定的行被执行。这是一个方便的方法来暂停没有设置断点的行。
2. 放置你的光标到你想到的行
3. 从鼠标右键的弹出菜单中选择Run to line命令Run To Line或者按下Ctrl+R,挂起到光标所在的位置

注意：

1. 线程的执行行将永远不会挂起
2. 在到达指定的行之前，断点和异常可以导致线程被挂起
3. 有用的两个点
   1. Step with filters过滤器非常有用，当我们不想看第三方库的执行流程
   2. Run to line 也很不错，很多时候，我们都喜欢，使劲按F6