Java程序员应该知道的10个调试技巧

调试可以帮助识别和解决应用程序缺陷，在本文中，将使用大家常用的的开发工具Eclipse来调试Java应用程序。

但这里介绍的调试方法基本都是通用的，也适用于NetBeans IDE，我们会把重点放在运行时上面。

在本文中使用的是Eclipse Juno版（Eclipse 4.2），在开始前给大家提3点建议：

1.不要使用System.out.println作为调试工具。

2.把所有涉及到的组件日志级别激活并使用。

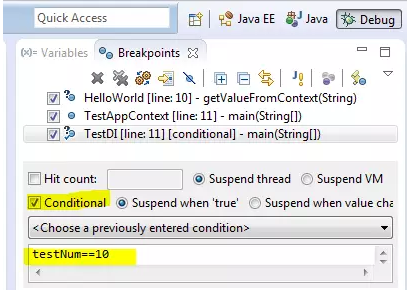
3.使用日志分析器来读取日志。

一、条件断点

如果你不知道如何添加断点，只需点击左边面板（行号前面）断点即被创建。

在调试界面中，“断点”视图会把所有被创建的断点列出来。

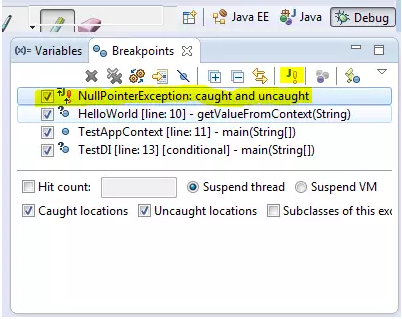
我们可以给它加一个布尔条件，也就是说，该断点会被激活并且如果布尔条件为真，就会执行该断点，否则将会跳过往下执行。



二、异常断点

在断点视图中，有一个J!标记按钮！我们可以使用该按钮来添加一个Java异常断点。

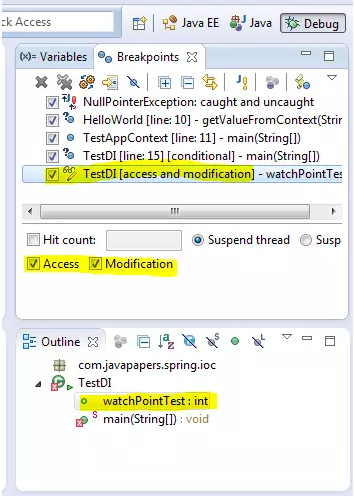
例如，我们想让程序在遇到空指针异常（NullPointerException）时，仍然能继续调试，那么我们可以使用该按钮来添加一个异常断点！



三、监视点

这是一个非常好的功能，当选定的属性访问或修改程序时，程序会停止执行并允许进行调试。

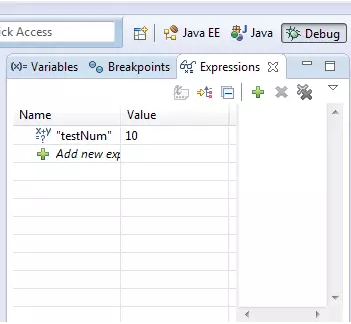
在Outline视图中选择一个类变量并从上下文菜单中选择切换监视点，属性监视点将会被创建，在断点（Breakpoints）视图中会把所有监视点用列表的形式显示出来。



四、评估/检查

按Ctrl+Shift+D或者Ctrl+Shift+I来显示选定变量或者表达式的值。

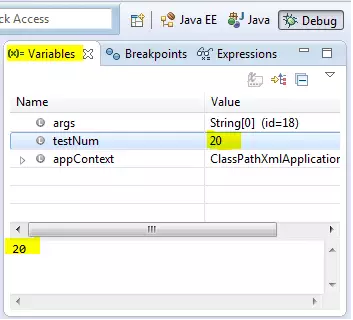
我们也可以给一个变量或表达式添加永久观察点，当程序在调试时，这些观察点就会在表达式视图（Expression view）中显示出来。



五、修改变量值

在调试过程中，我们可以修改变量值。

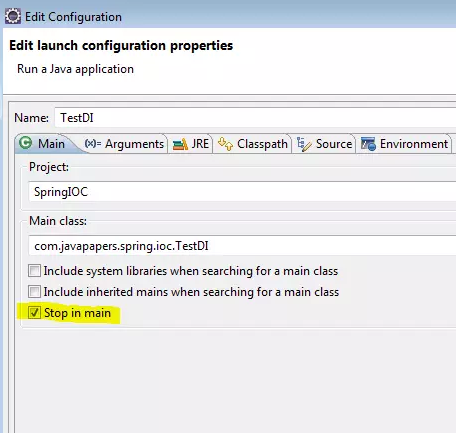
先选好一个变量然后进入变量视图（Variables view），根据变量类型在其对应的Value列里输入值即可。



六、在Main函数里面停止执行

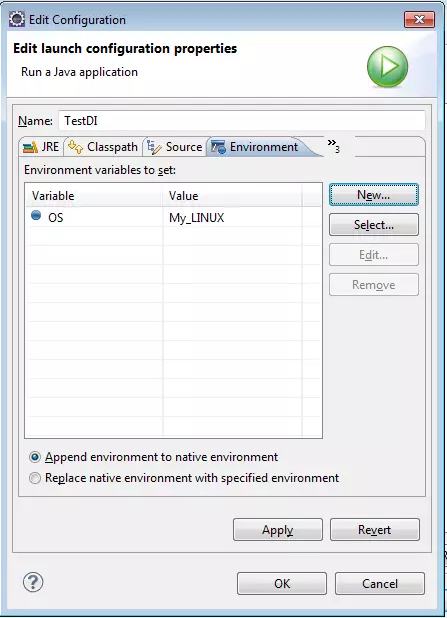
在运行/调试设置中，编辑配置对话框中有“Main”这个选项卡，我们可以勾选“Stop in main”这个复选框。

如果选中，那么在调试一个基于main方法的Java程序时，程序会在main方法第一行位置便停止执行。



七、环境变量

并不是在系统属性中添加环境变量，我们可以在编辑配置对话框中很方便地进行添加。



八、Drop to Frame

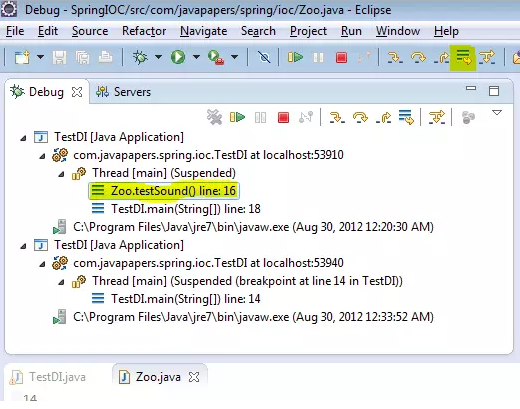
这也是我最喜欢的一个功能。

调试期间，可以重新跳到调用堆栈框架的开始处执行，并且变量值也会回到最初。

根据回档调整堆栈的深度，这个功能的主要用途是所有变量状态可以快速回到方法开始执行时候的样子，

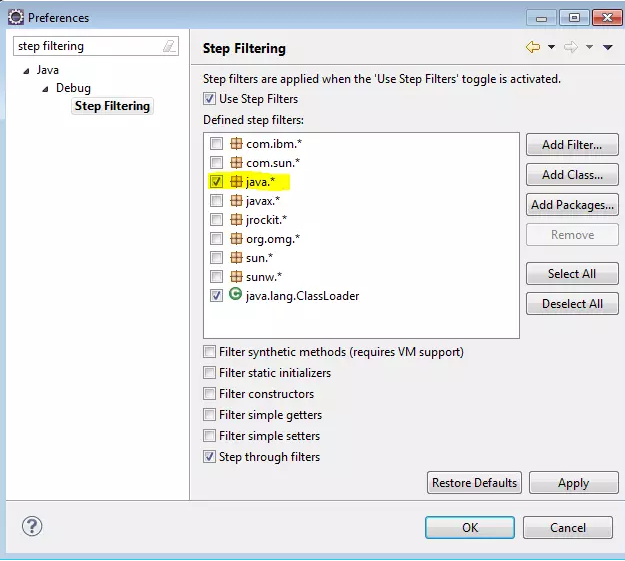
然后你可以重新进行一遍一遍执行，这样就可以在你关注的地方进行多次调试，

但是在执行过程中也会产生一些副作用，比如插入到数据库里面的数据是无法删除的！



九、分布过滤

当我们进入（F5）方法的时候，我们还可以访问其外部库（比如java.\*），我们可能不需要这个库，就可以在Perference选项卡页面添加一个过滤器来排除这个包。



十、进入、跳出和返回

我把这个放在最后一点，在调试过程中，这些是必须要了解（最好掌握）的东西：

F5——进入：移动到下一个步骤，如果当前行有一个方法调用，该控件将会跳转到被调用方法的第一行执行。

F6——跳出：移动到下一行。如果在当前行有方法调用，那么会直接移动到下一行执行。不会进入被调用方法体里面。

F7——返回：从当前方法中跳出，继续往下执行。

F8——移动到下一个断点处执行。

