为何 IntelliJ IDEA 比 Eclipse 更好

[彭博](https://my.oschina.net/dtec) [彭博](https://my.oschina.net/dtec) 发布于2012年03月19日 收藏 109 [评论 167](http://www.oschina.net/news/26929/why-intellij-is-better-than-eclipse#news_comments_wrapper)

https://my.oschina.net/img/hot3.png

[](http://www.oschina.net/p/intellij+idea)

**圣战**

有一些没有唯一正确答案的“永恒”的问题，例如哪个更好：是Windows还是Linux，Java还是C#；谁更强壮：Chuck Norris还是Van Damme。

其中的一个圣战便是Java IDE的选择：



有许多争论的话题是讨论它们哪个有更多的插件、快捷键等等。它们之间有太多的不同而很难决定什么才是最重要的。

因此，人们都表示两个IDE在功能性上都差不多，选择某一个则是个人喜好问题了。

我认为这不仅仅是个人喜好的问题，有客观的原因能表明：[Intellij IDEA](http://www.oschina.net/p/intellij+idea) 肯定比[Eclipse](http://www.oschina.net/p/eclipse)更好.

我不会去大量比较两者细微不同，例如插件与快捷键之类的。因为有一系列的blog是关于这些的，例如：[[part 1](http://arhipov.blogspot.com/2011/06/whats-cool-in-intellijidea-part-i.html)] [[part 2](http://arhipov.blogspot.com/2011/07/whats-cool-in-intellijidea-part-ii-live.html)] [[part 3](http://arhipov.blogspot.com/2011/08/whats-cool-in-intellijidea-part-iii.html)]。我将会展示IDEA与Eclipse的一个最大的不同之处。通常IDEA和Eclipse的fans都不曾了解。前者已经习惯性的使用着它而不曾想象IDE缺少它会怎样，后者则没有使用它的习惯且难以想象有了它IDE能做得这么好。Eclipse的用户在把玩IDEA的时候通常都不会意识到这个不同点，因为他们习惯了以Eclipse模式工作。

**为何我如此确信？**

在开始说明之前，让我解释一下我是谁和为什么我的观点值得一听。

我使用Eclipse已经5年之久，我很了解它，我为它编写插件并真心地喜爱它。然后，我换了一个公司工作，并被强行转移到IDEA上开发，在这次“分手”后，我才最终意识到为什么IDEA更加敏锐。而如今我我已经使用IDEA一些年了，所以我能比较它们两者，因为我了解它们。

**IDEA与Eclipse主要的不同**

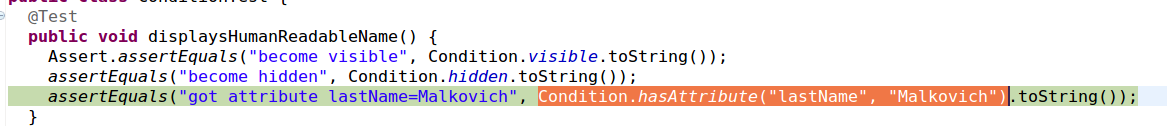
IDEA之所以更加敏锐的原因是：IDEA懂得上下文。这也是JetBrains员工命名IDEA为intelligent的原因。它真正的含义又是什么呢？IDEA检索你的整个项目，分析项目所有的内容，甚至构建语法树。幸亏有它，在任何时候无论你将鼠标放置在哪里，IDEA都知道你在哪里并知道在那能干什么。

仍旧不了然？别担心，在列举了以下几个例子后，一切就清楚了。

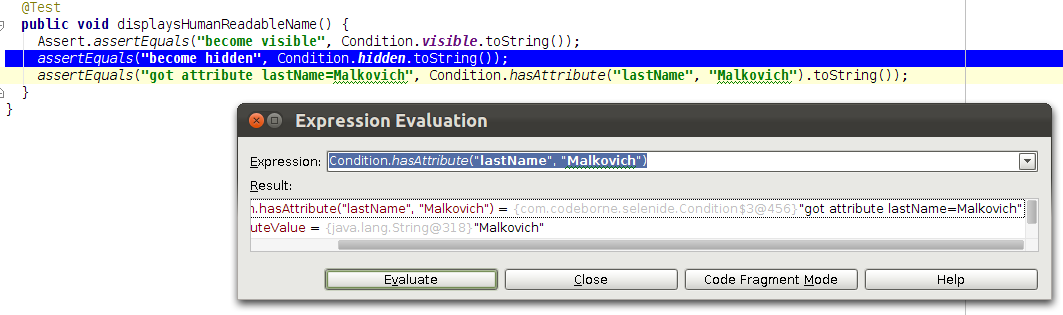
这种对上下文的理解表现在很多很多方式上，以下只是其中的一部分。

**1、调试**

通常在调试过程中，我们想要求一些表达式的值，在Eclipse中你需要选定这个表达式，选定整个表达式非常重要，否则你就没法求出它的值。然后，你可以使用Ctrl+Shift+I看到这个表达式的值。



使用IDEA你不需要选择任何东西，你只需要把光标移到你的表达式中然后按下Alt+F8。IDEA会很好的理解这个你可能需要的表达式，然后显示一个对话窗口，其中有一些它给出的建议参数变量。你也可以编辑并在对话框中立即将得到这个表达式的值。相当的方便！试过这个特性后，你就不会再想在Eclipse中调试了。



基本上在两个IDE上都能让你做同样的事，但是使用IDEA会更加简单和快速。对此我认真的表示，这个差异太大了：就像天堂跟人间一样。在一个小小的IDEA对话窗口中就提供了自动完成，语法高亮以及你需要的一切。

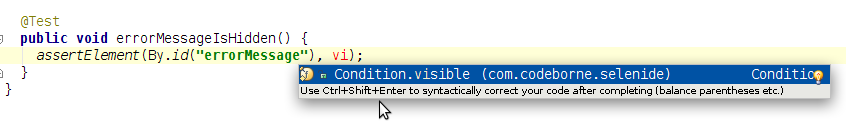
**2、自动完成**

自动完成使得IDE比notepad更加优秀，在这个领域能感知上下文的IDEA有了质的飞跃。例如，我们开始如下一行代码：

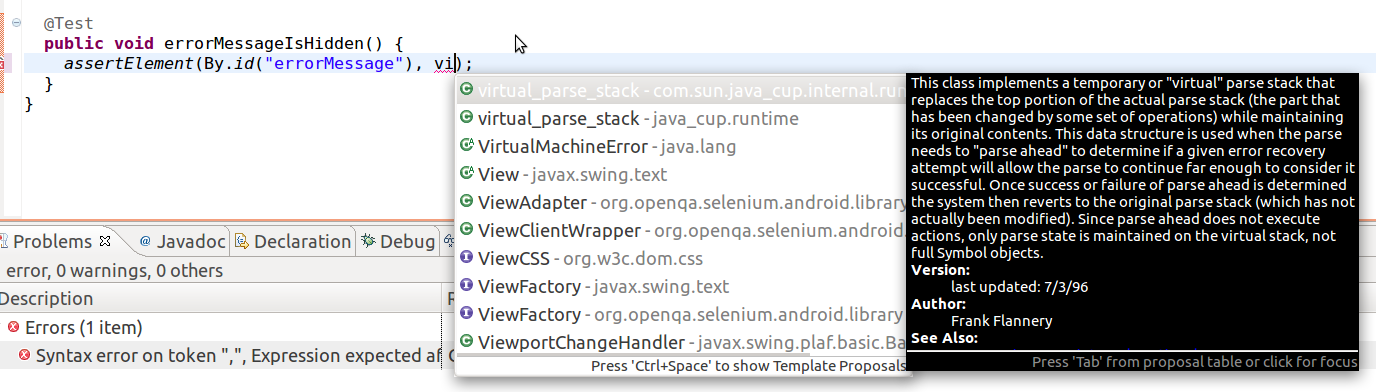
assertElement(By.id("errorMessage"), vi

现在我们想要查找以vi开头的选项有哪些。

IDEA怎么做的呢？无需等待任何的按键，它马上就能理解assertElement需要一个Conditaion类的示例作为第二个参数，而在Condition类中有一个静态变量名叫visible。IDEA会直接建议这个唯一的有效选项。



然而，Eclipse会怎么做呢？唉，它不能理解上下文，它不知道光标所在处就是assertElement方法的第二个参数。因此当你按下“神圣”的Ctrl+Space时，Eclipse只会简单的显示全局中以vi字母开头的一切。



在一个美丽的弹出框中我看到了很多漂亮的高亮的格式规范的无用的信息.:(

**3、重构**

专业的程序员能够高效的使用IDE提供的重构功能。所有的现代IDE都提供许多印象深刻的重构功能。但是还是那句，IDEA的重构功能也很聪明智能。它们能读懂你需要什么，然后针对不同的情况提供给你最适合的解决方案。

例如，假设我们有一个assertErrorMessageIsHidden方法：

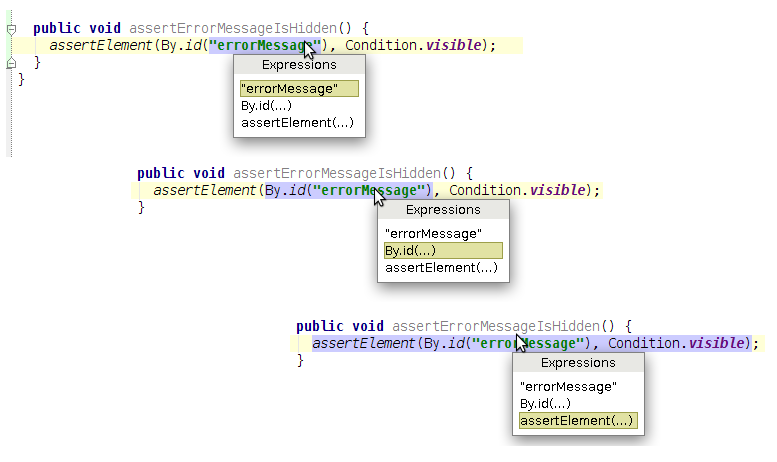
public void assertErrorMessageIsHidden() {

assertElement(By.id("errorMessage"), Condition.visible);

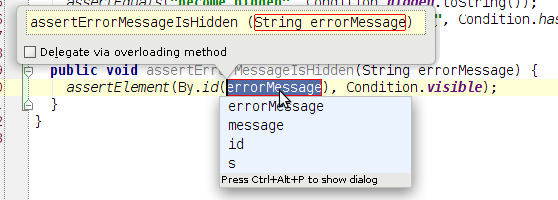
}

我们以字符串“errorMessage”作为该方法的一个参数。

让我们先从IDEA开始，将光标放在任何字符串“errorMessage”上，按Ctrl+Alt+P(代表“parameter”)，IDEA将会建议我们可能可以在哪些表达中用次字符串为参数。



当“errorMessage”表达式被选定以后，IDEA将会建议一些可能的能用做该参数的变量名称。

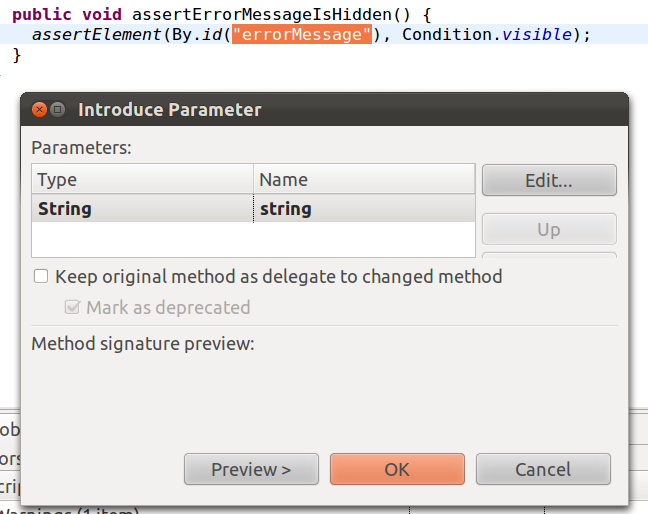


**每月惊奇**

你会经常地惊讶于IDEA的聪敏表现，比如它为你提示一些选项的名称。它考虑到方法名称，变量类型甚至是变量值，以及其他的此类变量的名称，与之前你给定的此变量的名称，除了忽略了你的生肖。相信我，你会想说“Wow,IDEA也能这样？”，我想说至少每个月一次吧。

现在然我们看看Eclipse是怎么样给出建议的。

别忘了，选择整个"errorMessage"表达式哦。选择一个“推荐参数”的重构功能（通过菜单，没有什么快捷键可供使用），获得差不多相同的结果，然而，Eclipse不会给出变量名提示选项，但幸亏不会。



**结论**

如果我们选择Java IDE，那么Intellij IDEA肯定要比Eclipse好的。这不仅仅是个人喜好问题。从客观上看，IDEA更好。它是的你能够快捷地编写和修改代码。它给出恰当的名称提示，查找适合的方法。它不需要你特别地选择表达式便会猜测到你想要做的以及你想如何对它命名。IDEA预示着并给你提示。

**P.S. Remarks**

我认为IDEA在作为Java IDE时确实比Eclipse更好。如果你考虑他们其他方面的能力例如用于其他例如C++、Python、Scala的IDE上，或者是作为构建桌面的应用的平台，Eclipse很有可能拔得头筹。

事实上这是依赖于对他们的定义的，Eclipse已将自己定位为一个抽象平台，用来使用插件构建任何工具，而IDEA则定位为一个“智能的Java IDE”，而且它正如此。

作为纪念我就尝试着列绝一些Eclipse可能比IDEA更好的地方：

* Eclipse看上去更棒，所有大题小做的SWT以及本地控件都用的很值。Eclipse看上去很稳定的深思熟虑过的程序，它有着漂亮的字体以及图标。然而当第一眼看到IDEA是，它像是作噱头的玩具一样，有着不完全的图标以及笨拙的Swing界面。
* Eclipse拥有更强的项目结构支持。在IDEA中，你的项目是有模块组成。在Eclipse中你拥有一个由工程组成的工作区，每个工程都能被单独打开或关闭，分组或隐藏。但是你真的需要么？
* 为Eclipse编写插件似乎很简单
* IDEA比Eclipse使用更多的系统资源。这其实也说得通--因为IDEA了解得更多
* 最后，Eclipse是免费的，然而IDEA Community版本已经够我用了

**最后的建议**

如果你需要漂亮的图标，用来创建桌面程序的平台，C++IDE或者你用一个低配的笔记本工作，那么Eclipse可能对你来说是更好的选择。但是如果你是一个正儿八经的Java开发者，你需要更快更方便的工具帮你集中注意于解决问题而不是让你分心，IDEA则正是你所需要的。

via:[dzone.com](http://java.dzone.com/articles/why-idea-better-eclipse) ,[OSChina](http://www.oschina.net/)原创编译