

Eclipse 常用快捷键

Eclipse 的编辑功能非常强大，掌握了 Eclipse 快捷键功能，能够大大提高开发效率。Eclipse 中有如下一些和编辑相关的快捷键。

1. 【ALT+/】

此快捷键为用户编辑的好帮手，能为用户提供内容的辅助，不要为记不全方法和属性名称发愁，当记不全类、方法和属性的名字时，多体验一下【ALT+/】快捷键带来的好处吧。

2. 【Ctrl+O】

显示类中方法和属性的大纲，能快速定位类的方法和属性，在查找 Bug 时非常有用。

3. 【Ctrl+/】

快速添加注释，能为光标所在行或所选定行快速添加注释或取消注释，在调试的时候可能总会需要注释一些东西或取消注释，现在好了，不需要每行进行重复的注释。

4. 【Ctrl+D】

删除当前行，这也是笔者的最爱之一，不用为删除一行而按那么多次的删除键。

5. 【Ctrl+M】

窗口最大化和还原，用户在窗口中进行操作时，总会觉得当前窗口小（尤其在编写代码时），现在好了，试试【Ctrl+M】快捷键。

查看和定位快捷键

在程序中，迅速定位代码的位置，快速找到 Bug 的所在，是非常不容易的事，Eclipse 提供了强大的查找功能，可以利用如下的快捷键帮助完成查找定位的工作。

1. 【Ctrl+K】、【Ctrl++Shift+K】

快速向下和向上查找选定的内容，从此不再需要用鼠标单击查找对话框了。

2. 【Ctrl+Shift+T】

查找工作空间（Workspace）构建路径中的可找到 Java 类文件，不要为找不到类而痛苦，而且可以使用“*”、“？”等通配符。

3. 【Ctrl+Shift+R】

和【Ctrl+Shift+T】对应，查找工作空间（Workspace）中的所有文件（包括 Java 文件），也可以使用通配符。

4. 【Ctrl+Shift+G】

查找类、方法和属性的引用。这是一个非常实用的快捷键，例如要修改引用某个方法的代码，可以通过【Ctrl+Shift+G】快捷键迅速定位所有引用此方法的位置。

5. 【Ctrl+Shift+O】

快速生成 import，当从网上拷贝一段程序后，不知道如何 import 进所调用的类，试试【Ctrl+Shift+O】快捷键，一定会有惊喜。

6. 【Ctrl+Shift+F】

格式化代码，书写格式规范的代码是每一个程序员的必修之课，当看见某段代码极不顺眼时，选定后按【Ctrl+Shift+F】快捷键可以格式化这段代码，如果不选定代码则默认格式化当前文件（Java 文件）。

7. 【ALT+Shift+W】

查找当前文件所在项目中的路径，可以快速定位浏览器视图的位置，如果想查找某个文件所在的包时，此快捷键非常有用（特别在比较大的项目中）。

8. 【Ctrl+L】

定位到当前编辑器的某一行，对非 Java 文件也有效。

9. 【Alt+←】、【Alt+→】

后退历史记录和前进历史记录，在跟踪代码时非常有用，用户可能查找了几个有关联的地方，但可能记不清楚了，可以通过这两个快捷键定位查找的顺序。

10. 【F3】

快速定位光标位置的某个类、方法和属性。

11. 【F4】

显示类的继承关系，并打开类继承视图。

调试快捷键

Eclipse 中有如下一些和运行调试相关的快捷键。

1. **【Ctrl+Shift+B】**：在当前行设置断点或取消设置的断点。
2. **【F11】**：调试最后一次执行的程序。
3. **【Ctrl+F11】**：运行最后一次执行的程序。
4. **【F5】**：跟踪到方法中，当程序执行到某方法时，可以按**【F5】**键跟踪到方法中。
5. **【F6】**：单步执行程序。
6. **【F7】**：执行完方法，返回到调用此方法的后一条语句。
7. **【F8】**：继续执行，到下一个断点或程序结束。

常用编辑器快捷键

通常文本编辑器都提供了一些和编辑相关的快捷键，在 Eclipse 中也可以通过这些快捷键进行文本编辑。

1. **【Ctrl+C】**：复制。
2. **【Ctrl+X】**：剪切。
3. **【Ctrl+V】**：粘贴。
4. **【Ctrl+S】**：保存文件。
5. **【Ctrl+Z】**：撤销。
6. **【Ctrl+Y】**：重复。
7. **【Ctrl+F】**：查找。

其他快捷键

Eclipse 中还有很多快捷键，无法一一列举，用户可以通过帮助文档找到它们的使用方式，另外还有几个常用的快捷键如下。

1. **【Ctrl+F6】**：切换到下一个编辑器。
2. **【Ctrl+Shift+F6】**：切换到上一个编辑器。
3. **【Ctrl+F7】**：切换到下一个视图。
4. **【Ctrl+Shift+F7】**：切换到上一个视图。
5. **【Ctrl+F8】**：切换到下一个透视图。
6. **【Ctrl+Shift+F8】**：切换到上一个透视图。

Eclipse 中快捷键比较多,可以通过帮助文档找到所有快捷键的使用,但要掌握所有快捷键的使用是不可能的,也没有必要,如果花点时间熟悉本节列举的快捷键,必将会事半功倍

1. edit->content Assist -> add Alt+/ 代码关联
2. Window -> Next Editor -> add Ctrl+Tab 切换窗口
3. Run/Debug Toggle Line Breakpoint -> add Ctrl+` 在调试的时候 增删断点
4. Source-> Surround with try/catch Block -> Ctrl+Shift+v 增加 trycatch 框框
5. Source -> Generate Getters and Setters -> Ctrl+Shift+. 增加 get set 方法

-----有用的快捷键-----

Alt+/ 代码助手完成一些代码的插入(但一般和输入法有冲突,可以修改输入法的热键,也可以暂用 Alt+/来代替)

Ctrl+1:光标停在某变量上,按 Ctrl+1 键,可以提供快速重构方案。选中若干行,按 Ctrl+1 键可将此段代码放入 for、while、if、do 或 try 等代码块中。

双击左括号(小括号、中括号、大括号),将选择括号内的所有内容。

Alt+Enter 显示当前选择资源(工程,or 文件 or 文件)的属性

-----Ctrl 系列-----

Ctrl+K:将光标停留在变量上,按 Ctrl+K 键可以查找到下一个同样的变量

Ctrl+Shift+K:和 Ctrl+K 查找的方向相反

Ctrl+E 快速显示当前 Editor 的下拉列表(如果当前页面没有显示的用黑体表示)

Ctrl+Shift+E 显示管理当前打开的所有 View 的管理器(可以选择关闭,激活等操作)

Ctrl+Q 定位到最后编辑的地方

Ctrl+L 定位在某行(对于程序超过 100 的人就有福音了)

Ctrl+M 最大化当前的 Edit 或 View(再按则反之)

Ctrl+/ 注释当前行,再按则取消注释

Ctrl+T 快速显示当前类的继承结构

Ctrl+Shift-T: 打开类型(Open type)。如果你不是有意磨洋工,还是忘记通过源码树(source tree)打开的方式吧。

Ctrl+O:在代码中打开类似大纲视图的小窗口

Ctrl+鼠标停留:可以显示类和方法的源码

Ctrl+H:打开搜索窗口

Ctrl+/(小键盘) 折叠当前类中的所有代码

Ctrl+×(小键盘) 展开当前类中的所有代码

-----Ctrl+Shift 系列-----

Ctrl+Shift+F 格式化当前代码

Ctrl+Shift+X 把当前选中的文本全部变味小写

Ctrl+Shift+Y 把当前选中的文本全部变为小写

Ctrl+Shift+O:快速地导入 import

Ctrl+Shift+R:打开资源 open Resource

-----F 快捷键 系列-----

F3:打开声明该引用的文件

F4:打开类型层次结构

F5:单步跳入

F6:单步跳过

F7:单步跳出

F8:继续, 如果后面没有断点, 程序将运行完

-----行编辑用-----

Ctrl+D: 删除当前行

Ctrl+Alt+ ↓ 复制当前行到下一行(复制增加)

Ctrl+Alt+ ↑ 复制当前行到上一行(复制增加)

Alt+ ↓ 当前行和下面一行交互位置(特别实用,可以省去先剪切,再粘贴了)

Alt+ ↑ 当前行和上面一行交互位置(同上)

Shift+Enter 在当前行的下一行插入空行(这时鼠标可以在当前行的任一位置,不一定是最后)

Ctrl+Shift+Enter 在当前行插入空行(原理同上条)

-----不常用的-----

Alt+← 前一个编辑的页面

Alt+→ 下一个编辑的页面(当然是针对上面那条来说了)

Ctrl+Shift+S:保存全部

Ctrl+W 关闭当前 Editor

Ctrl+Shift+F4 关闭所有打开的 Editor

Ctrl+Shift+G: 在 workspace 中搜索引用

Ctrl+Shift+P 定位到对于的匹配符(譬如{}) (从前面定位后面时,光标要在匹配符里面,后面到前面,则反之)

-----增量查找 -----

Ctrl+J 正向增量查找(按下 Ctrl+J 后,你所输入的每个字母编辑器都提供快速匹配定位到某个单词,如果没有,则在 status line 中显示没有找到了,查一个单词时,特别实用,这个功能 Idea 两年前就有了)

Ctrl+Shift+J 反向增量查找(和上条相同,只不过是后往前查)