1. -----------自动代码--------

常用的有fori/sout/psvm+Tab即可生成循环、System.out、main方法等boilerplate样板代码

例如要输入for(User user : users)只需输入user.for+Tab

再比如，要输入Date birthday = user.getBirthday();只需输入user.getBirthday().var+Tab即可。代码标签输入完成后，按Tab，生成代码。

Ctrl+Alt+O 优化导入的类和包

Alt+Insert 生成代码(如get,set方法,构造函数等)   或者右键（Generate）

fori/sout/psvm + Tab

Ctrl+Alt+T  生成try catch  或者 Alt+enter

CTRL+ALT+T  把选中的代码放在 TRY{} IF{} ELSE{} 里

Ctrl + O 重写方法

Ctrl + I 实现方法

Ctr+shift+U 大小写转化

ALT+回车    导入包,自动修正

ALT+/       代码提示

CTRL+J      自动代码

Ctrl+Shift+J，整合两行为一行

CTRL+空格   代码提示

CTRL+SHIFT+SPACE 自动补全代码

CTRL+ALT+L  格式化代码

CTRL+ALT+I  自动缩进

CTRL+ALT+O  优化导入的类和包

ALT+INSERT  生成代码(如GET,SET方法,构造函数等)

CTRL+E      最近更改的代码

CTRL+ALT+SPACE  类名或接口名提示

CTRL+P   方法参数提示

CTRL+Q，可以看到当前方法的声明

Shift+F6  重构-重命名 (包、类、方法、变量、甚至注释等)

Ctrl+Alt+V 提取变量

2. -----------查询快捷键--------

Ctrl＋Shift＋Backspace可以跳转到上次编辑的地

CTRL+ALT+ left/right 前后导航编辑过的地方

ALT+7  靠左窗口显示当前文件的结构

Ctrl+F12 浮动显示当前文件的结构

ALT+F7 找到你的函数或者变量或者类的所有引用到的地方

CTRL+ALT+F7  找到你的函数或者变量或者类的所有引用到的地方

Ctrl+Shift+Alt+N 查找类中的方法或变量

双击SHIFT 在项目的所有目录查找文件

Ctrl+N   查找类

Ctrl+Shift+N 查找文件

CTRL+G   定位行

CTRL+F   在当前窗口查找文本

CTRL+SHIFT+F  在指定窗口查找文本

CTRL+R   在 当前窗口替换文本

CTRL+SHIFT+R  在指定窗口替换文本

ALT+SHIFT+C  查找修改的文件

CTRL+E   最近打开的文件

F3   向下查找关键字出现位置

SHIFT+F3  向上一个关键字出现位置

选中文本，按Alt+F3 ，高亮相同文本，F3逐个往下查找相同文本

F4   查找变量来源

CTRL+SHIFT+O  弹出显示查找内容

Ctrl+W 选中代码，连续按会有其他效果

F2 或Shift+F2 高亮错误或警告快速定位

Ctrl+Up/Down 光标跳转到第一行或最后一行下

Ctrl+B 快速打开光标处的类或方法

CTRL+ALT+B  找所有的子类

CTRL+SHIFT+B  找变量的类

Ctrl+Shift+上下键  上下移动代码

Ctrl+Alt+ left/right 返回至上次浏览的位置

Ctrl+X 删除行

Ctrl+D 复制行

Ctrl+/ 或 Ctrl+Shift+/  注释（// 或者/\*...\*/ ）

Ctrl+H 显示类结构图

Ctrl+Q 显示注释文档

Alt+F1 查找代码所在位置

Alt+1 快速打开或隐藏工程面板

Alt+ left/right 切换代码视图

ALT+ ↑/↓  在方法间快速移动定位

CTRL+ALT+ left/right 前后导航编辑过的地方

Ctrl＋Shift＋Backspace可以跳转到上次编辑的地

Alt+6    查找TODO

3.---------------------其他快捷键-------------------

SHIFT+ENTER 另起一行

CTRL+Z   倒退(撤销)

CTRL+SHIFT+Z  向前(取消撤销)

CTRL+ALT+F12  资源管理器打开文件夹

ALT+F1   查找文件所在目录位置

SHIFT+ALT+INSERT 竖编辑模式

CTRL+F4  关闭当前窗口

Ctrl+Alt+V，可以引入变量。例如：new String(); 自动导入变量定义

Ctrl+~，快速切换方案（界面外观、代码风格、快捷键映射等菜单）

4.--------------svn快捷键---------------

ctrl+k 提交代码到SVN

ctrl+t 更新代码

5.--------------调试快捷键---------------

其实常用的 就是F8 F7 F9 最值得一提的 就是Drop Frame  可以让运行过的代码从头再来

alt+F8          debug时选中查看值

Alt+Shift+F9，选择 Debug

Alt+Shift+F10，选择 Run

Ctrl+Shift+F9，编译

Ctrl+Shift+F8，查看断点

F7，步入

Shift+F7，智能步入

Alt+Shift+F7，强制步入

F8，步过

Shift+F8，步出

Alt+Shift+F8，强制步过

Alt+F9，运行至光标处

Ctrl+Alt+F9，强制运行至光标处

F9，恢复程序

Alt+F10，定位到断点

6.--------------重构---------------

Ctrl+Alt+Shift+T，弹出重构菜单

Shift+F6，重命名

F6，移动

F5，复制

Alt+Delete，安全删除

Ctrl+Alt+N，内联

============================================================

 十大Intellij IDEA快捷键

Intellij IDEA中有很多快捷键让人爱不释手，stackoverflow上也有一些有趣的讨论。每个人都有自己的最爱，想排出个理想的榜单还真是困难。

以前也整理过Intellij的快捷键，这次就按照我日常开发时的使用频率，简单分类列一下我最喜欢的十大快捷-神-键吧。

1 智能提示:

Intellij首当其冲的当然就是Intelligence智能！基本的代码提示用Ctrl+Space，还有更智能地按类型信息提示Ctrl+Shift+Space，但因为Intellij总是随着我们敲击而自动提示，所以很多时候都不会手动敲这两个快捷键(除非提示框消失了)。用F2/ Shift+F2移动到有错误的代码，Alt+Enter快速修复(即Eclipse中的Quick Fix功能)。当智能提示为我们自动补全方法名时，我们通常要自己补上行尾的反括号和分号，当括号嵌套很多层时会很麻烦，这时我们只需敲Ctrl+Shift+Enter就能自动补全末尾的字符。而且不只是括号，例如敲完if/for时也可以自动补上{}花括号。

最后要说一点，Intellij能够智能感知Spring、Hibernate等主流框架的配置文件和类，以静制动，在看似“静态”的外表下，智能地扫描理解你的项目是如何构造和配置的。

2 重构:

Intellij重构是另一完爆Eclipse的功能，其智能程度令人瞠目结舌，比如提取变量时自动检查到所有匹配同时提取成一个变量等。尤其看过《重构-改善既有代码设计》之后，有了Intellij的配合简直是令人大呼过瘾！也正是强大的智能和重构功能，使Intellij下的TDD开发非常顺畅。

切入正题，先说一个无敌的重构功能大汇总快捷键Ctrl+Shift+Alt+T，叫做Refactor This。按法有点复杂，但也符合Intellij的风格，很多快捷键都要双手完成，而不像Eclipse不少最有用的快捷键可以潇洒地单手完成(不知道算不算Eclipse的一大优点)，但各位用过Emacs的话就会觉得也没什么了(非Emacs黑)。此外，还有些最常用的重构技巧，因为太常用了，若每次都在Refactor This菜单里选的话效率有些低。比如Shift+F6直接就是改名，Ctrl+Alt+V则是提取变量。

3 代码生成：

这一点类似Eclipse，虽不是独到之处，但因为日常使用频率极高，所以还是罗列在榜单前面。常用的有fori/sout/psvm+Tab即可生成循环、System.out、main方法等boilerplate样板代码，用Ctrl+J可以查看所有模板。后面“辅助”一节中将会讲到Alt+Insert，在编辑窗口中点击可以生成构造函数、toString、getter/setter、重写父类方法等。这两个技巧实在太常用了，几乎每天都要生成一堆main、System.out和getter/setter。

另外，Intellij IDEA 13中加入了后缀自动补全功能(Postfix Completion)，比模板生成更加灵活和强大。例如要输入for(User user : users)只需输入user.for+Tab。再比如，要输入Date birthday = user.getBirthday();只需输入user.getBirthday().var+Tab即可。

4 编辑：

编辑中不得不说的一大神键就是能够自动按语法选中代码的Ctrl+W以及反向的Ctrl+Shift+W了。此外，Ctrl+Left/Right移动光标到前/后单词，Ctrl+[/]移动到前/后代码块，这些类Vim风格的光标移动也是一大亮点。以上Ctrl+Left/Right/[]加上Shift的话就能选中跳跃范围内的代码。Alt+Forward/Backward移动到前/后方法。还有些非常普通的像Ctrl+Y删除行、Ctrl+D复制行、Ctrl+</>折叠代码就不多说了。

关于光标移动再多扩展一点，除了Intellij本身已提供的功能外，我们还可以安装ideaVim或者emacsIDEAs享受到Vim的快速移动和Emacs的AceJump功能(超爽！)。另外，Intellij的书签功能也是不错的，用Ctrl+Shift+Num定义1-10书签(再次按这组快捷键则是删除书签)，然后通过Ctrl+Num跳转。这避免了多次使用前/下一编辑位置Ctrl+Left/Right来回跳转的麻烦，而且此快捷键默认与Windows热键冲突(默认多了Alt，与Windows改变显示器显示方向冲突，一不小心显示器就变成倒着显式的了，冏啊)。

5 查找打开：

类似Eclipse，Intellij的Ctrl+N/Ctrl+Shift+N可以打开类或资源，但Intellij更加智能一些，我们输入的任何字符都将看作模糊匹配，省却了Eclipse中还有输入\*的麻烦。最新版本的IDEA还加入了Search Everywhere功能，只需按Shift+Shift即可在一个弹出框中搜索任何东西，包括类、资源、配置项、方法等等。

类的继承关系则可用Ctrl+H打开类层次窗口，在继承层次上跳转则用Ctrl+B/Ctrl+Alt+B分别对应父类或父方法定义和子类或子方法实现，查看当前类的所有方法用Ctrl+F12。

要找类或方法的使用也很简单，Alt+F7。要查找文本的出现位置就用Ctrl+F/Ctrl+Shift+F在当前窗口或全工程中查找，再配合F3/Shift+F3前后移动到下一匹配处。

Intellij更加智能的又一佐证是在任意菜单或显示窗口，都可以直接输入你要找的单词，Intellij就会自动为你过滤。

6 其他辅助：

以上这些神键配上一些辅助快捷键，即可让你的双手90%以上的时间摆脱鼠标，专注于键盘仿佛在进行钢琴表演。这些不起眼却是至关重要的最后一块拼图有：

Ø  命令：Ctrl+Shift+A可以查找所有Intellij的命令，并且每个命令后面还有其快捷键。所以它不仅是一大神键，也是查找学习快捷键的工具。

Ø  新建：Alt+Insert可以新建类、方法等任何东西。

Ø  格式化代码：格式化import列表Ctrl+Alt+O，格式化代码Ctrl+Alt+L。

Ø  切换窗口：Alt+Num，常用的有1-项目结构，3-搜索结果，4/5-运行调试。Ctrl+Tab切换标签页，Ctrl+E/Ctrl+Shift+E打开最近打开过的或编辑过的文件。

Ø  单元测试：Ctrl+Alt+T创建单元测试用例。

Ø  运行：Alt+Shift+F10运行程序，Shift+F9启动调试，Ctrl+F2停止。

Ø  调试：F7/F8/F9分别对应Step into，Step over，Continue。

此外还有些我自定义的，例如水平分屏Ctrl+|等，和一些神奇的小功能Ctrl+Shift+V粘贴很早以前拷贝过的，Alt+Shift+Insert进入到列模式进行按列选中。

Ø  Top #10切来切去：Ctrl+Tab

Ø  Top #9选你所想：Ctrl+W

Ø  Top #8代码生成：Template/Postfix +Tab

Ø  Top #7发号施令：Ctrl+Shift+A

Ø  Top #6无处藏身：Shift+Shift

Ø  Top #5自动完成：Ctrl+Shift+Enter

Ø  Top #4创造万物：Alt+Insert

太难割舍，前三名并列吧！

Ø  Top #1智能补全：Ctrl+Shift+Space

Ø  Top #1自我修复：Alt+Enter

Ø  Top #1重构一切：Ctrl+Shift+Alt+T

CTRL+ALT+ left/right 前后导航编辑过的地方

Ctrl＋Shift＋Backspace可以跳转到上次编辑的地方

---------------------

作者：有趣的灵魂算我一个

来源：CSDN

原文：https://blog.csdn.net/weixin\_42189233/article/details/80566249

版权声明：本文为博主原创文章，转载请附上博文链接！