

快捷键

<https://github.com/SeaSoonKeun/IntelliJ-IDEA-Tutorial/blob/master/keymap-win-mac.md>

- `fn + delete` 删除光标后一个字符
- `option+delete` 键，删除光标之前的一个单词（英文有效）；
- `fn+option+delete`，删除光标之后的一个单词；
- `command+delete`，删除掉该文件。
- `command + alt + L` 格式化代码
- `command + alt + I` 自动缩进，不用多次使用tab或着backspace键，也是比较方便的
- `command + 7` 查看结构
- `shift + enter` 光标所在行下空出一行，光标定位到新行
- `command + option + enter` 光标所在行上空出一行，光标定位到新行
- `command + Shift + Z` 取消撤销
- `command + Shift + Backspace` 退回到上次修改的地方
- `command + 0` 根据输入的 类名 查找类文件
- `command + ←` 移动光标到行首
- `command + →` 移动光标到行尾
- `command + Option + 左方向键` 退回到上一个操作的地方
- `command + Option + 右方向键` 前进到上一个操作的地方
- `command + 前方向键` 当前光标跳转到当前文件的前一个方法名位置
- `command + 后方向键` 当前光标跳转到当前文件的后一个方法名位置
- `Option + Enter` 根据光标所在问题，提供快速修复选择，光标放在的位置不同提示的结果也不同

常用模版

【1】模板1： main方法： main 或者 psvm

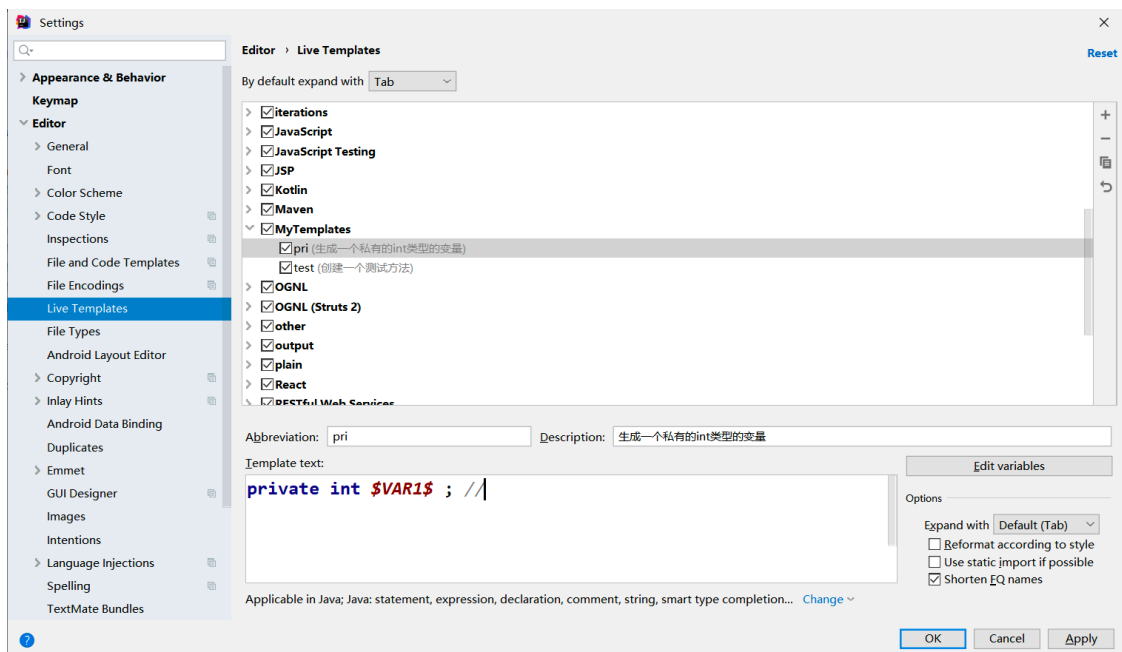
【2】模板2： 输出语句： sout 或者 .sout 一些变型： soutp:打印方法的形参
soutm:打印方法的名字 soutv:打印变量

【3】模板3： 循环 普通for循环： fori（正向） 或者 .fori（正向） . forr(逆向) 增强for循环： iter 或者 .for （可以用于数组的遍历，集合的遍历）

【4】模板4： 条件判断 ifn 或者 .null ： 判断是否为null （if null） inn 或者 .nn ： 判断不等于null (if not null)

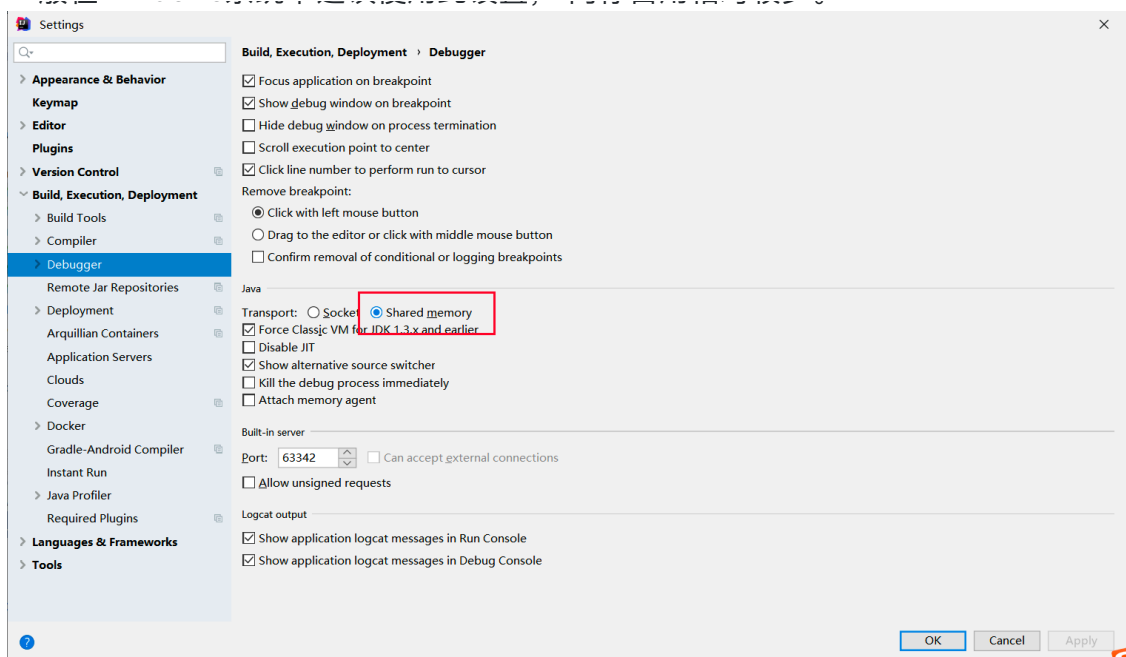
【5】模板5： 属性修饰符： prsf : private static final psf :public static final

自定义模版










断点调试

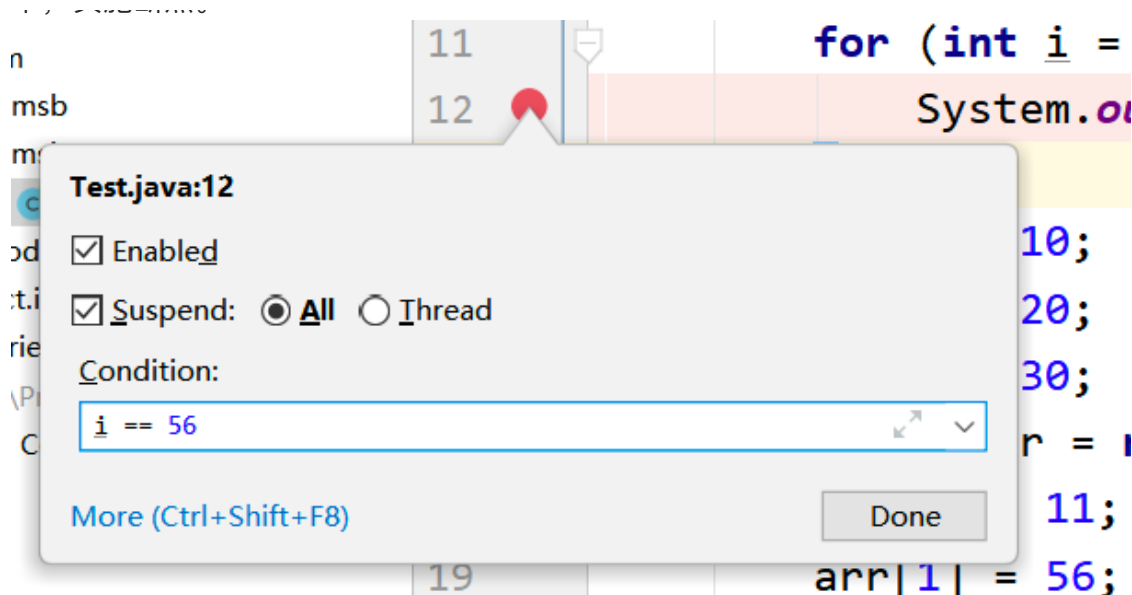
调试在开发中大量应用： 【1】 Debug的优化设置： 更加节省内存空间： 设置Debug连接方式，默认是Socket。 Shared memory是Windows 特有的一个属性，一般在Windows系统下建议使用此设置。 内存占用相对较少。



【2】常用断点调试快捷键：

-  一步一步的向下运行代码，不会走入任何方法中。
-  一步一步的向下运行代码，不会走入系统类库的方法中，但是会走入自定义的方法中。
-  一步一步的向下运行代码，会走入系统类库的方法中，也会走入自定义的方法中。
-  跳出方法
-  结束程序
-  进入到下一个断点，如果没有下一个断点了，就直接运行到程序结束。
-  在当前次取消未执行的断点。

【3】条件判断：说明：调试的时候，在循环里增加条件判断，可以极大的提高效率，心情也能愉悦。具体操作：在断点处右击调出条件断点。可以在满足某个条件下、实施断点。



【4】查看表达式的值：选择行，alt+f8。

