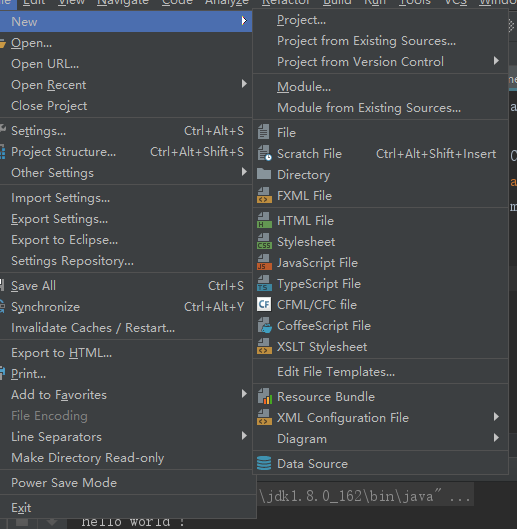
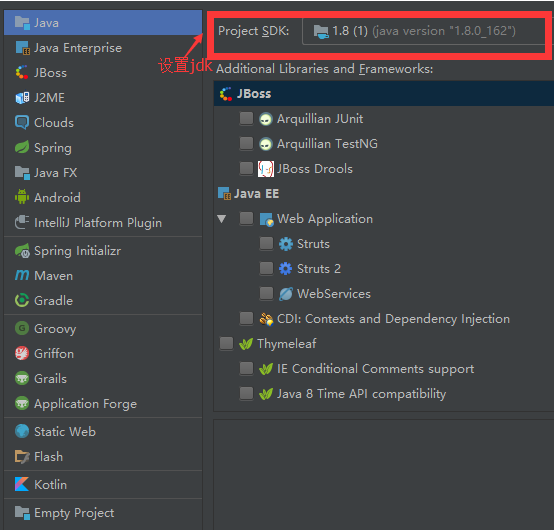
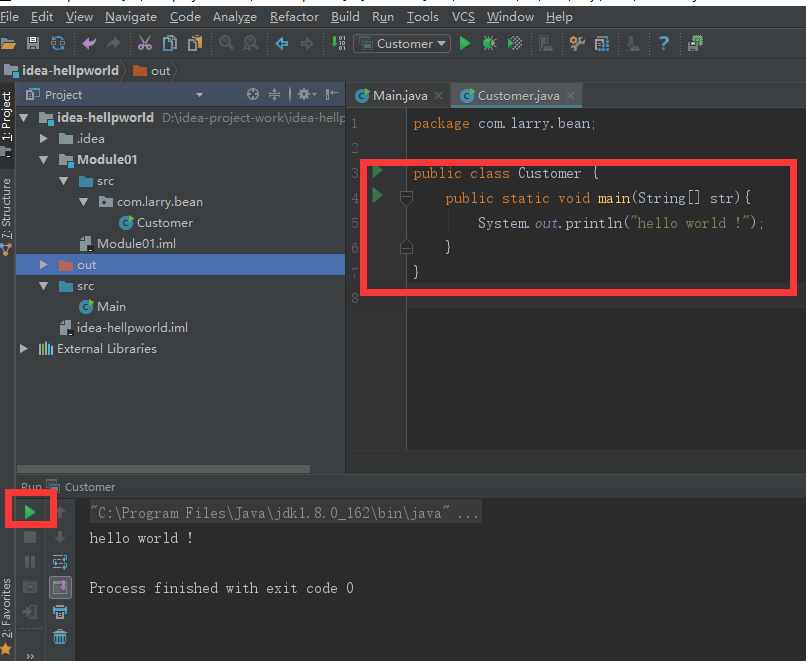
# Idea 学习操作

## 创建项目

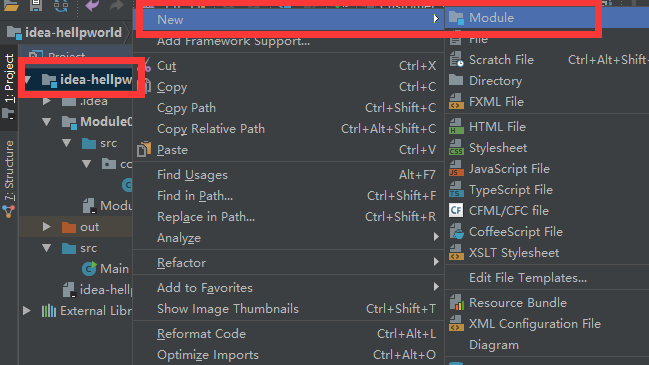


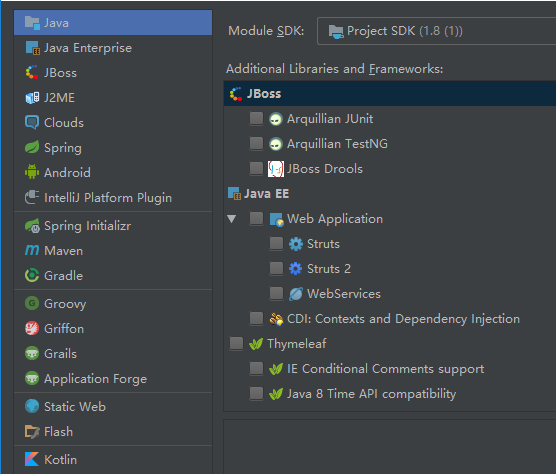


## 设置jdk相应的项目，执行操作

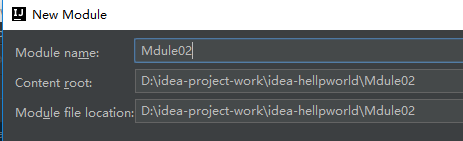


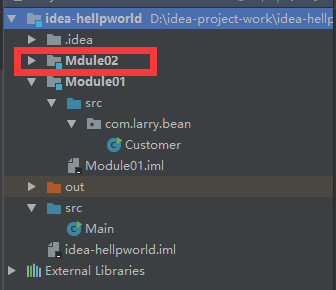
创建子项目（也叫创建模块）：点击项目右键new->Module,出现第二张图，指定对应的项目(java,web,maven,gradle项目)



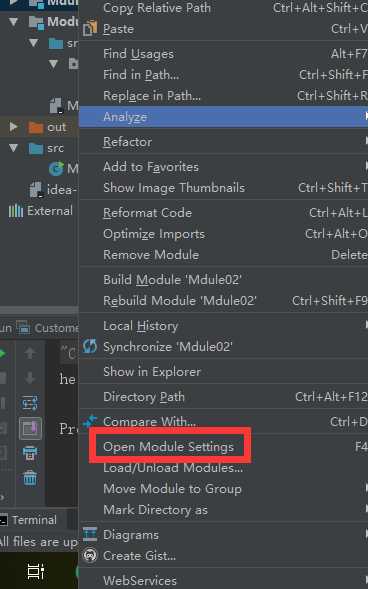


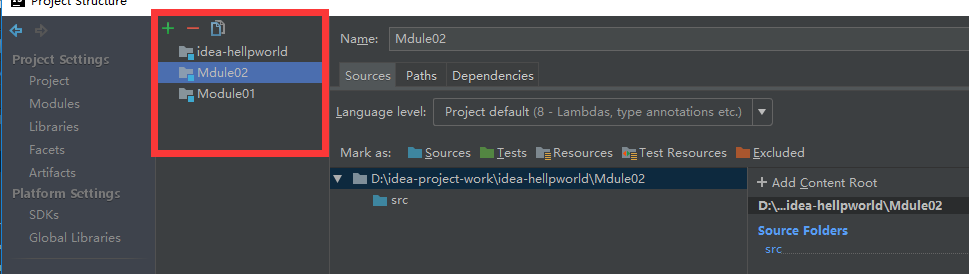
创建模块:



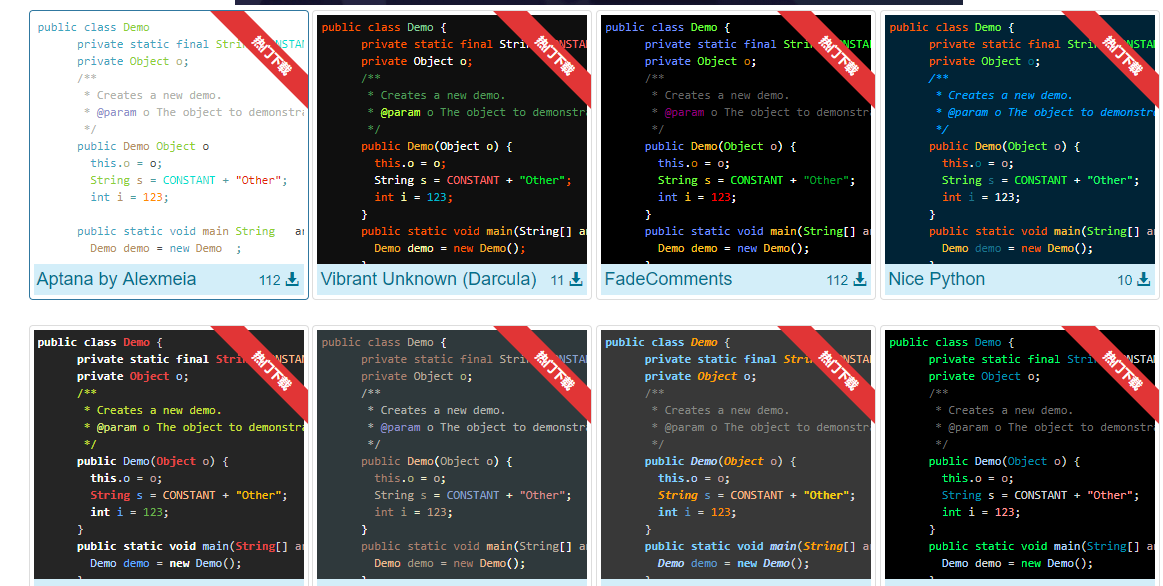


删除模块：对应模块右键出现菜单，点击Open Module Setting出现第二张图，就可以找项目（项目空间）下对应模块选择，右键delete（点击上面的红色“-”）选中apply->ok,但是模块mdule项目中还看到，在进行鼠标右键就可以看到delete(不做以上的delete删除操作，是没法看到以下delete)操作即可。

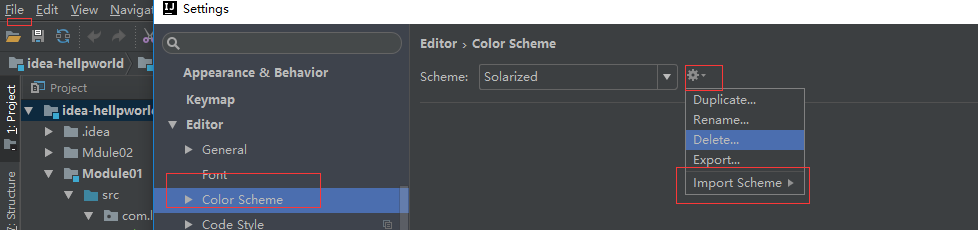




<https://www.riaway.com/> 下载主题

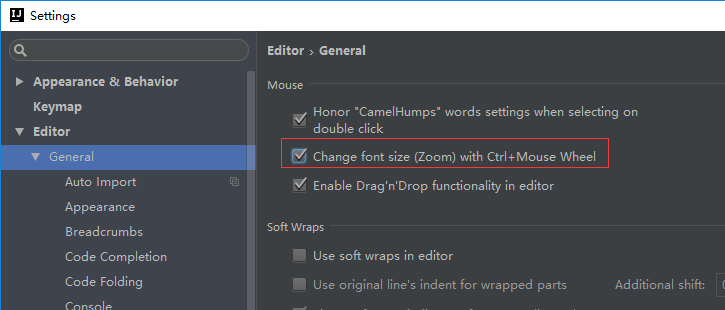


## 设置主题

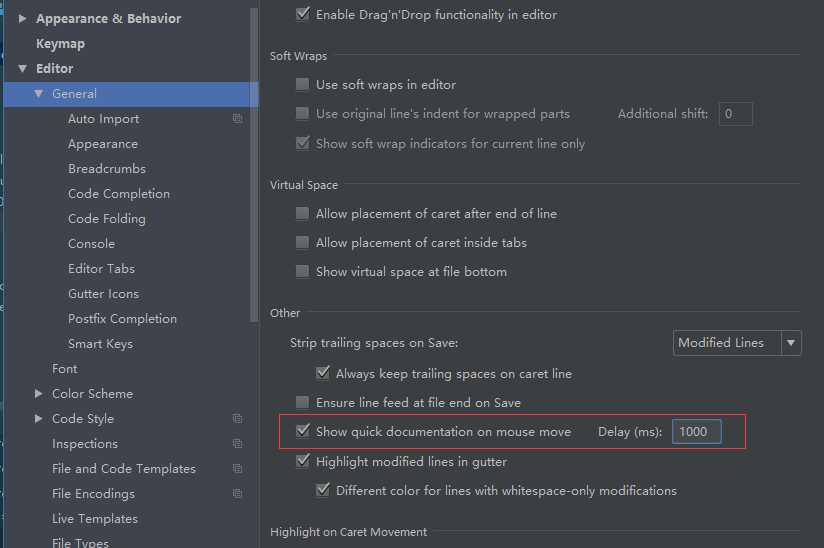


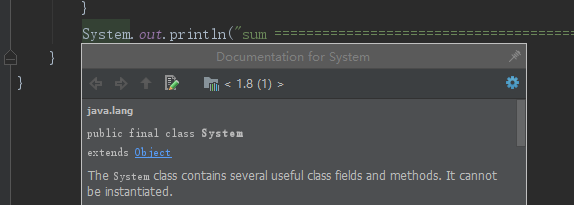
## Editor----》General

### 设置滚轮字体大小

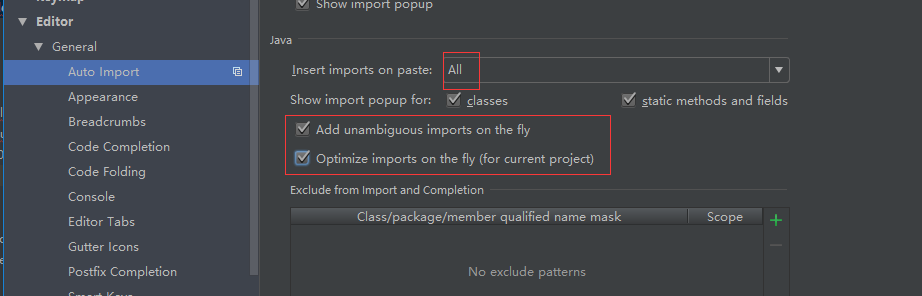


### 设置鼠标悬浮提示

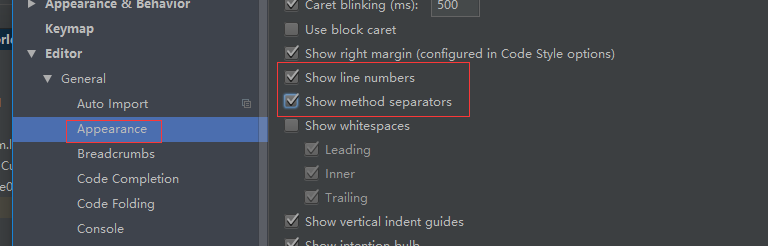




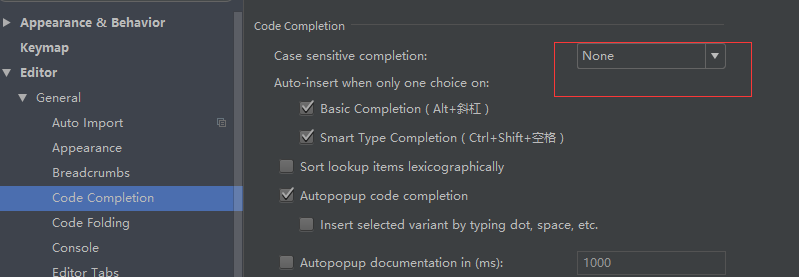
### 设置自动导包功能



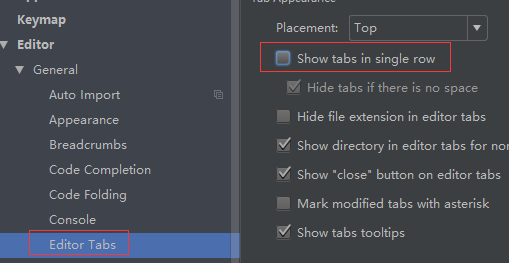
### 设置显示行号和方法间的分隔符



### 忽略大小写提示



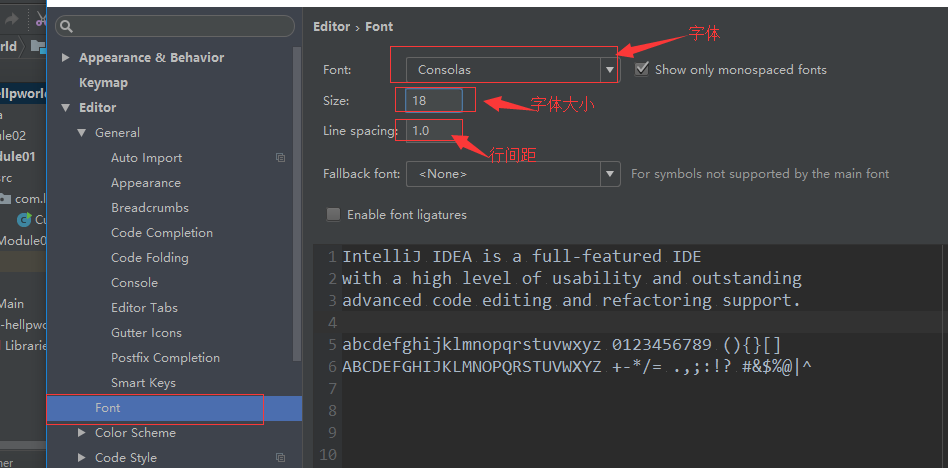
### 设置取消单行显示tabs的操作





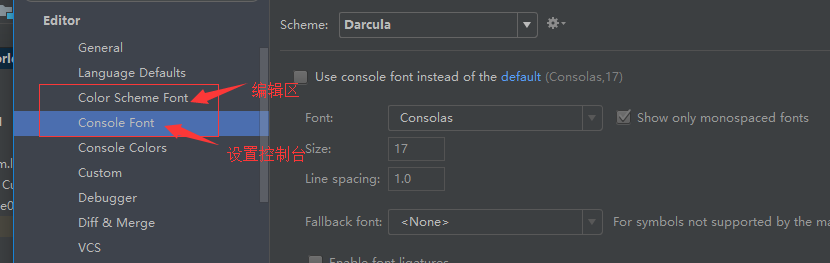
## Editor----》Font

### 设置默认的字体、字体大小、字体行间距



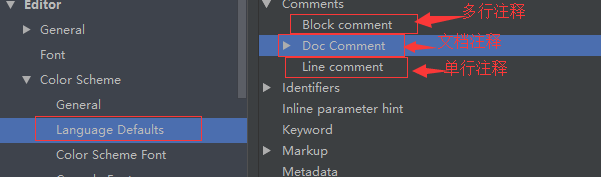
## Editor----》Color Scheme

### 修改当前主题的字体、字体大小、字体行间距（可忽略）



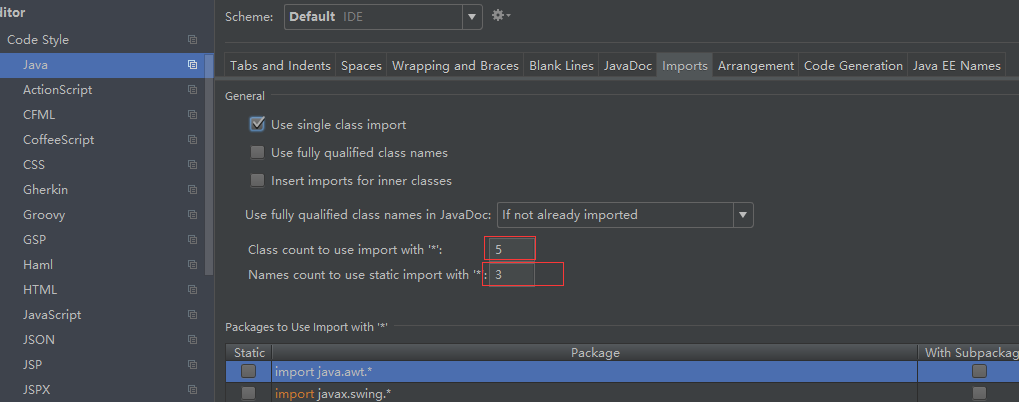
单独设置编辑区和控制台的

### 修改代码中注释的字体颜色



## Editor----》Code Style

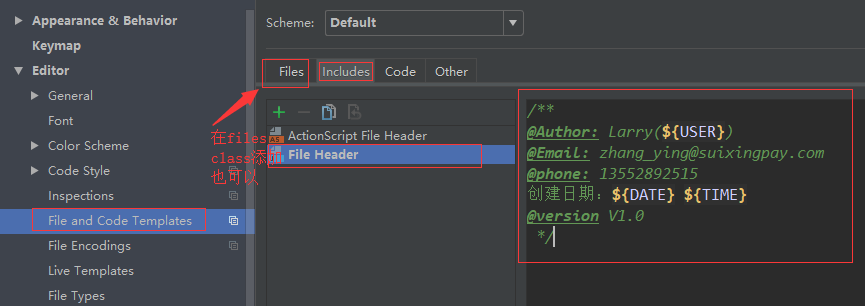
### 设置超过指定import个数，改为\*（可忽略）

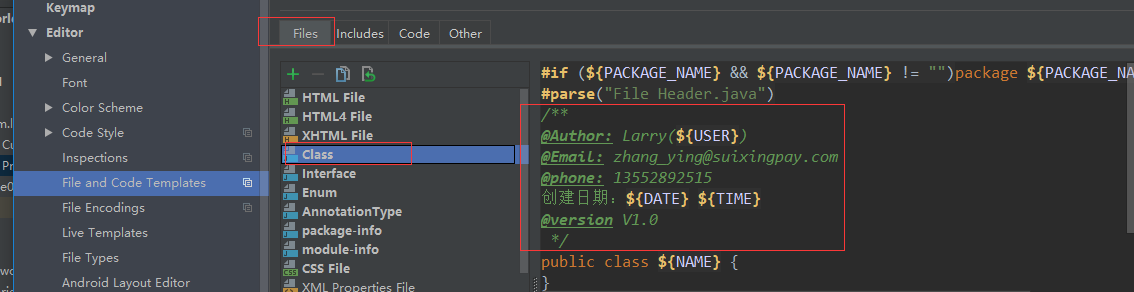


可以不做修改

## Editor----》File and Code Templates

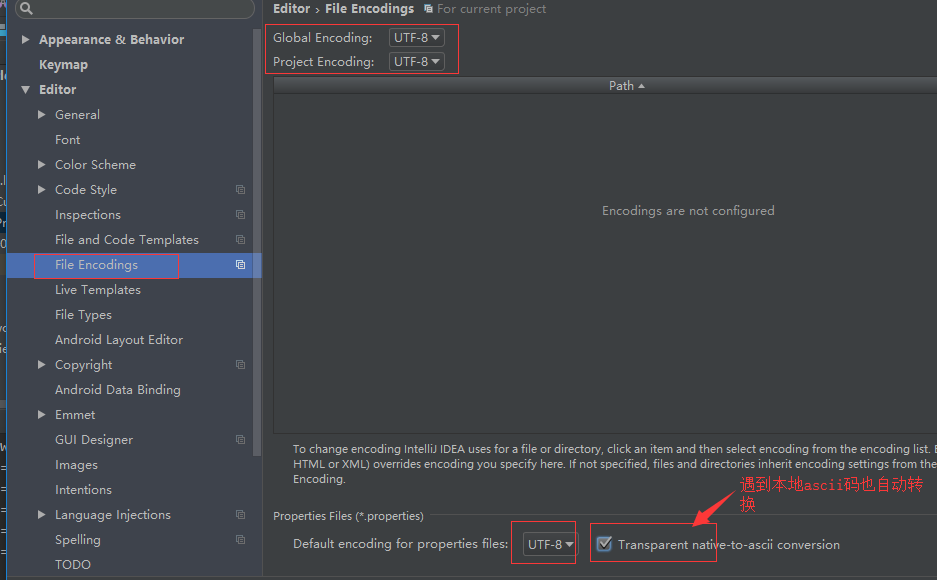
### 修改类头的文档注释信息





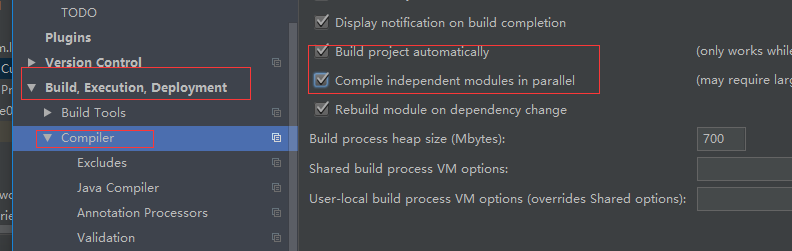
## Editor----》File Encodings

### 设置项目文件编码

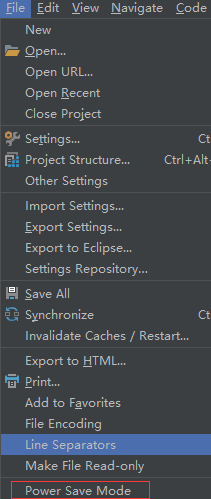


## Build，Execution，Deployment

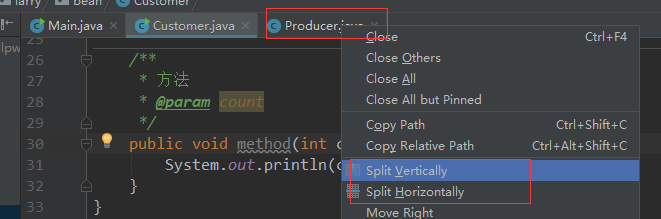
### 设置自动编译



### 设置省电模式（可忽略）

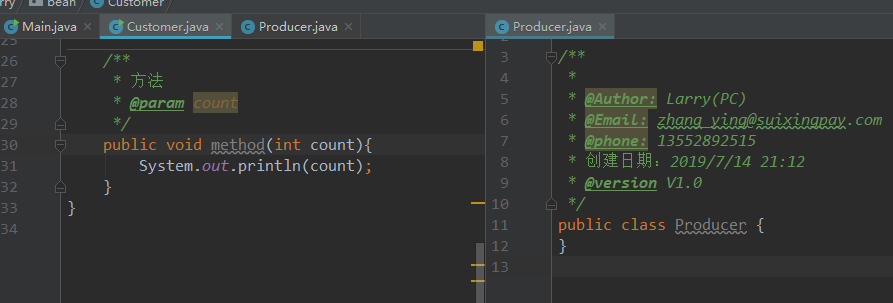


### Split vertically , Split Horizontally

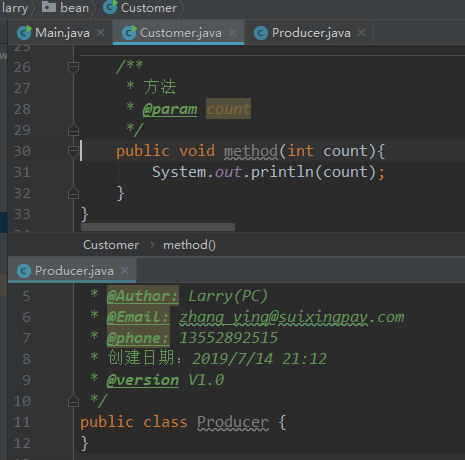


对应的参考类鼠标右键：

Split vertically 垂直地（左右）



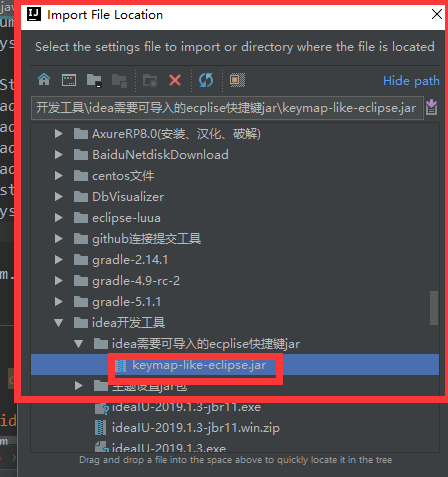
Split Horizontally 水平（上下）

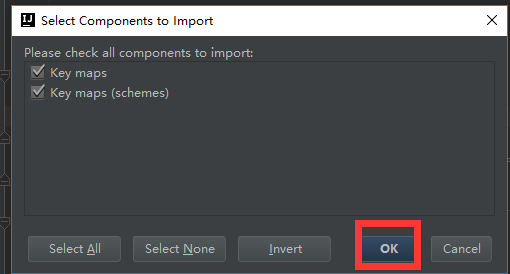


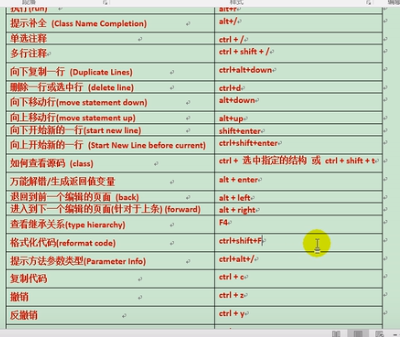
## IDEA-快捷键设置

### 导入ecplise 快捷键jar

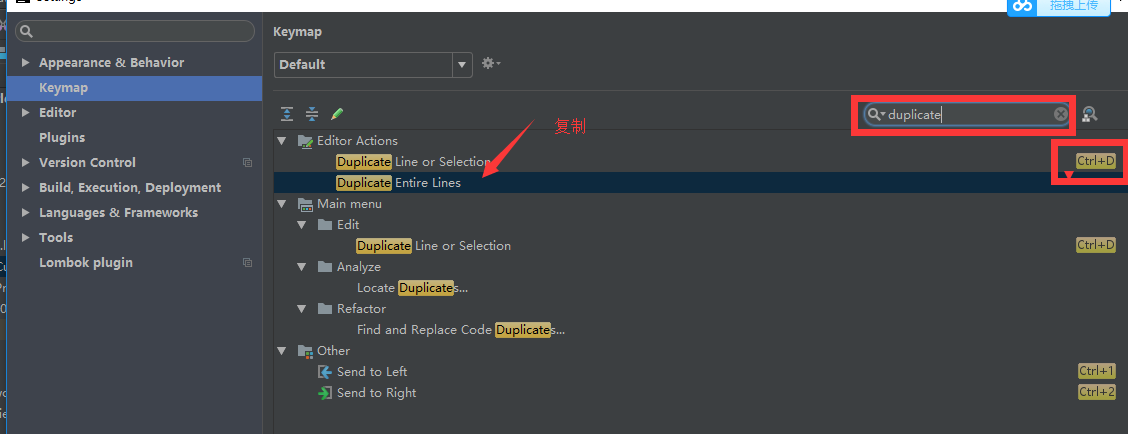
Import Setting



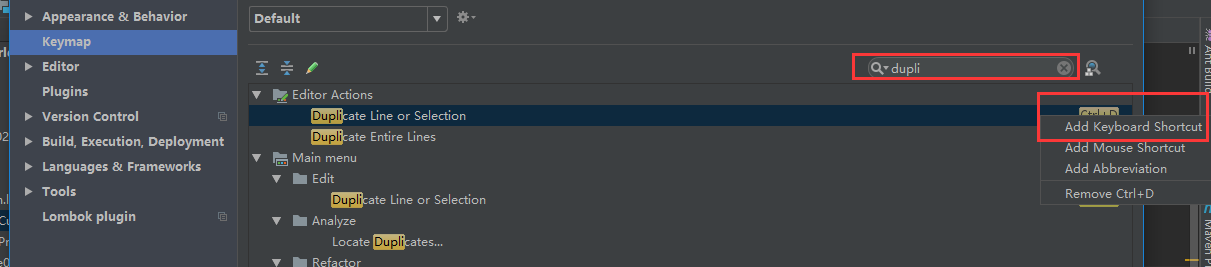


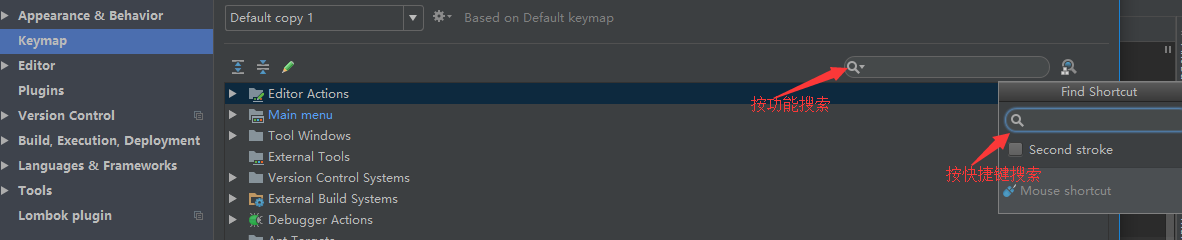


搜索快捷键



修改指定的快捷键





## 关于模板（Templates）

Editor-Live Templates 和 Editor-General -Postfix Completion

### Live Templates（实时代码模板）功能介绍

代码提示模板，这是用户定义的，可修改

### Postfix Completion

代码提示模板，这个是系统自带，不可修改的

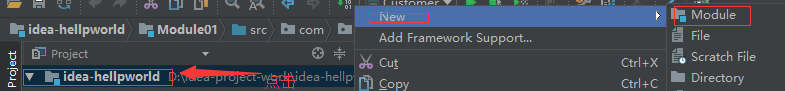
## 代码提示常用模板

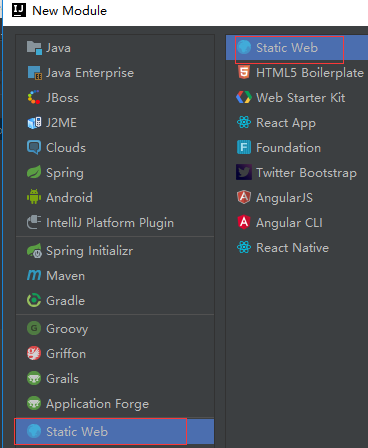
//模板六：prsf  
private static final int *count* = 10;  
//变形：psf  
public static final Customer *customer* = new Customer();  
//变形：psfs  
public static final String *mation* = "chain";  
  
//模板一：psvm（main）  
public static void main(String[] args) {  
 //模板二：sout  
 System.*out*.println("hello !");  
 //变形：soutp / soutm / soutv / xxx.sout  
 //soutp  
 System.*out*.println("args = [" + args + "]");  
 //soutm  
 System.*out*.println("Student.main");  
 int num = 10;  
 //soutv  
 System.*out*.println("num = " + num);  
 int num1 = 10;  
 //xxx.sout  
 System.*out*.println(num1);  
  
 //模板三：fori  
 String[] arr = new String[]{"zhangsan","lisi","wangwu"};  
 for (int i = 0; i < arr.length; i++) {  
 System.*out*.println(arr[i]);  
 }  
 //变形：iter  
 for (String s : arr) {  
 System.*out*.println(s);  
 }  
 //变形：itar  
 for (int i = 0; i < arr.length; i++) {  
 String s = arr[i];  
 System.*out*.println(s);  
 }  
 //模板四：list.for  
 ArrayList list = new ArrayList();  
 list.add("a");  
 list.add("b");  
 list.add("c");  
 for (Object o : list) {  
 System.*out*.println(o);  
 }  
 //变形：list.fori / list.forr  
 //list.fori  
 for (int i = 0; i < list.size(); i++) {  
 }  
 //list.forr 倒序遍历  
 for (int i = list.size() - 1; i >= 0; i--) {  
 }  
  
 //模板五：ifn  
 if (list == null) {  
 }  
 //变形：inn  
 if (list != null) {  
 }  
 //变形：xxx.nn / xxx.null  
 //xxx.nn  
 if (list != null) {  
 }  
 //xxx.null  
 if (list == null) {  
 }

## 创建java web Project或Module

### 创建的静态java web

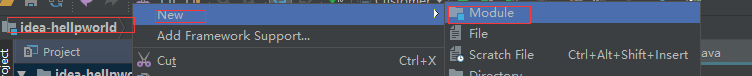
在module里创建的

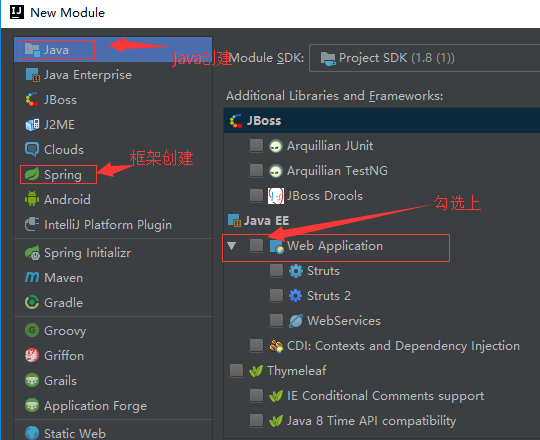


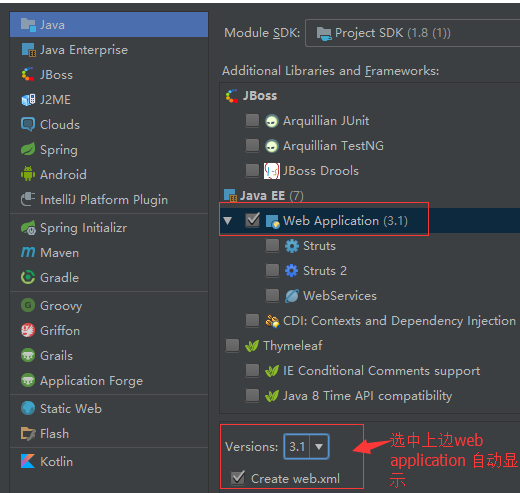


### 创建动态java web

在Module中创建的

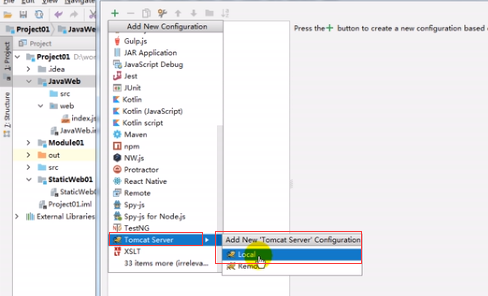


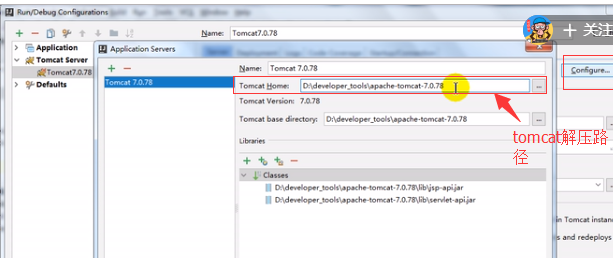


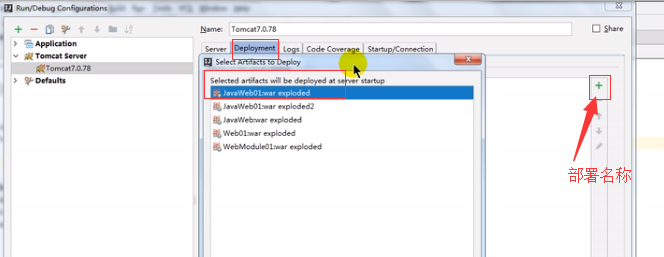


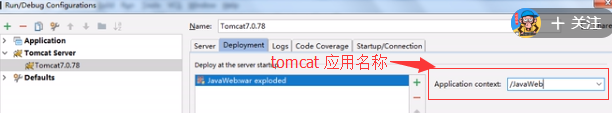
### 添加tomcat

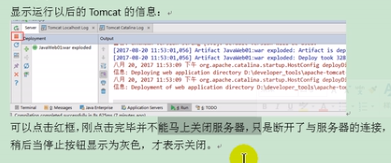
点击Run—》Edit configurations—》点击“+”



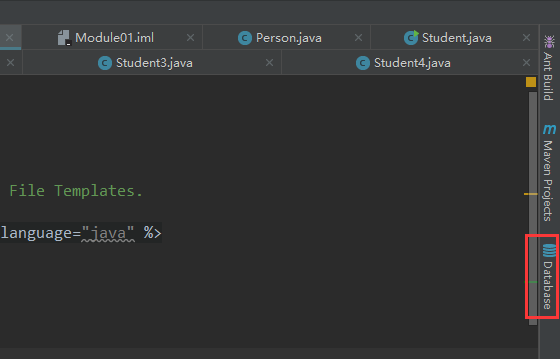


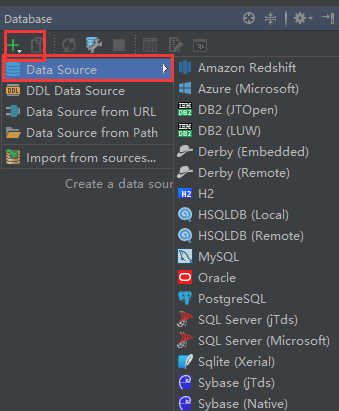


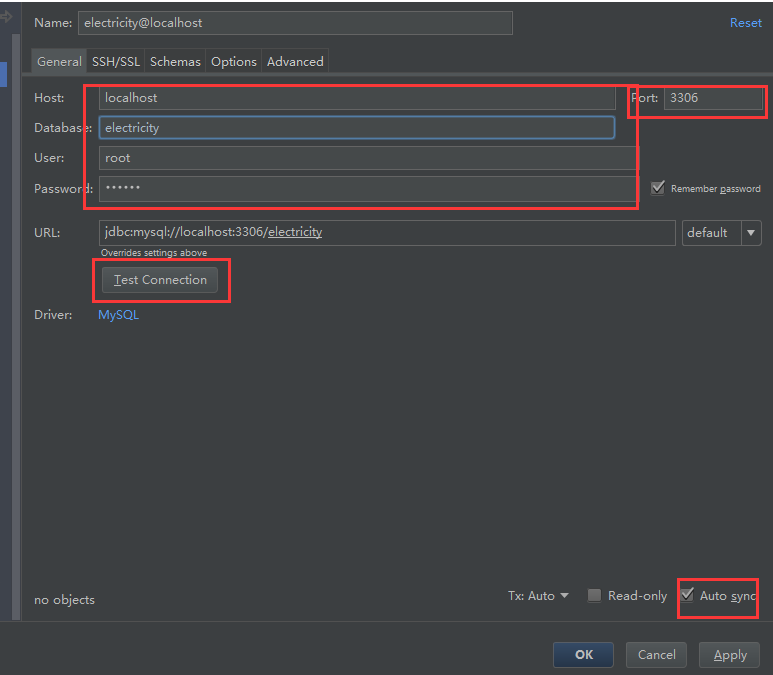


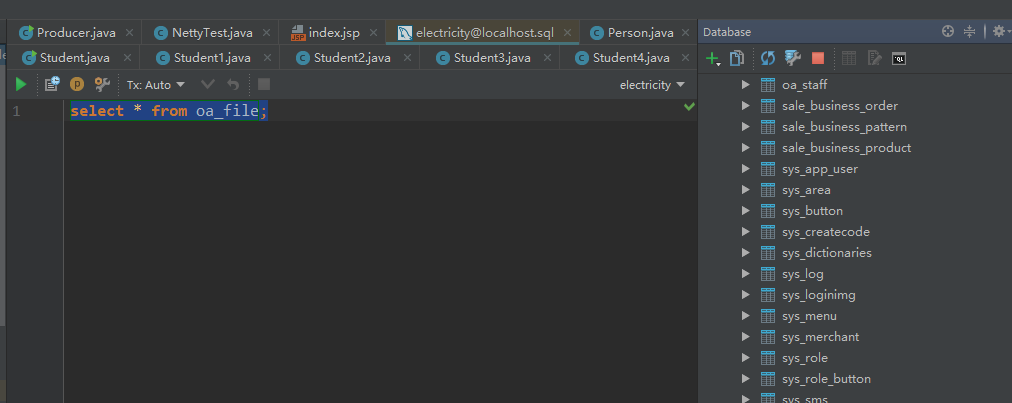


### 连接数据源

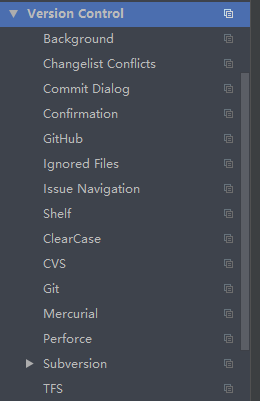




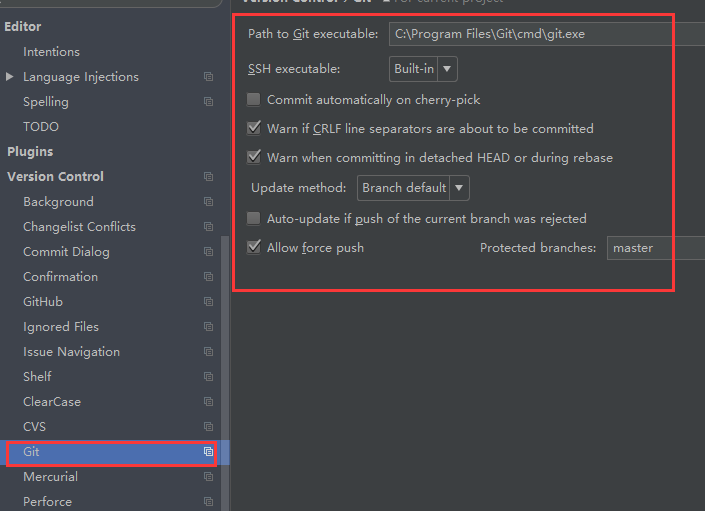




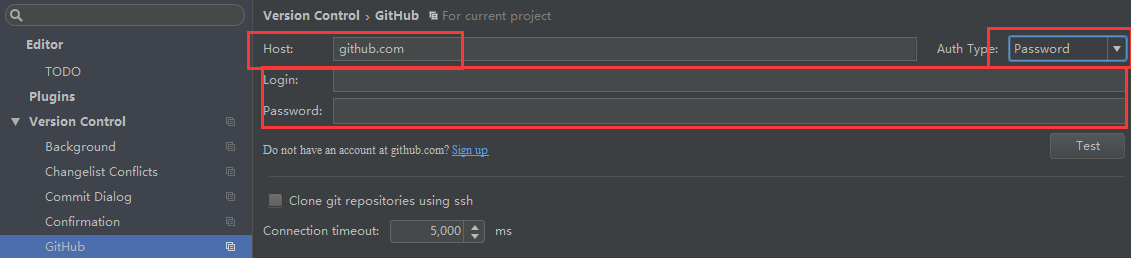
## 版本控制（version control）



### 安装好git的客户端

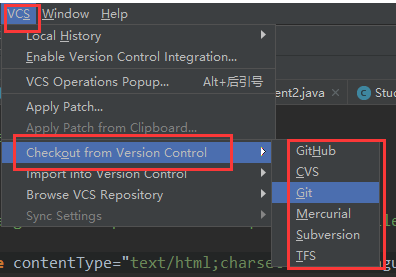


#### Idea连接github配置

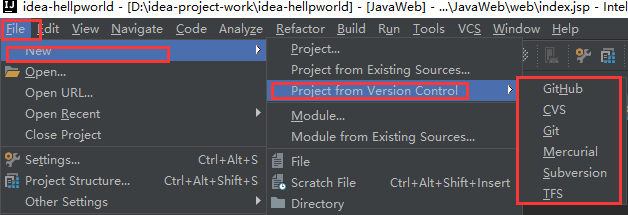


#### 获取github项目（命令git clone https://github.com/larry091001/xxxx.git）

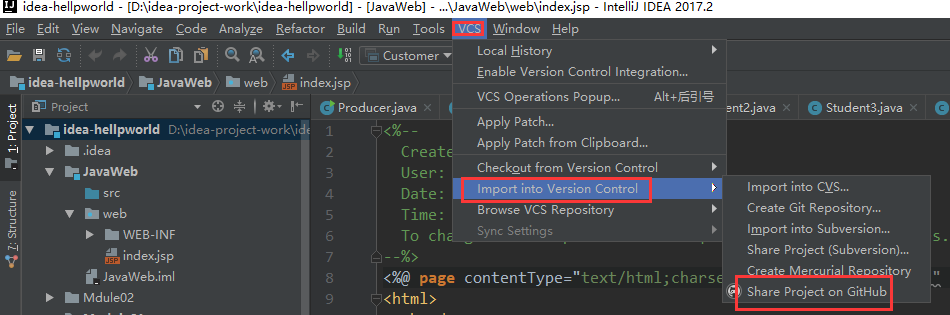
第一种方式：



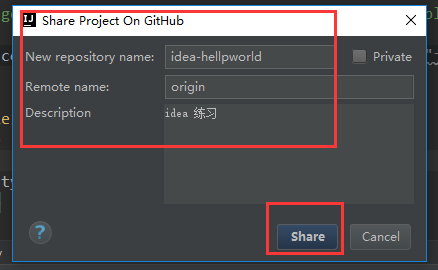
第二种方式：



### Shere本地项目到github

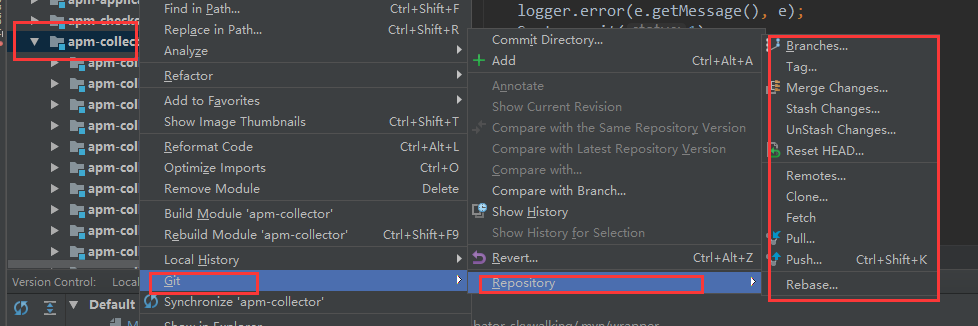


输入账号密码，下一步



点击下一步确定就可以了。

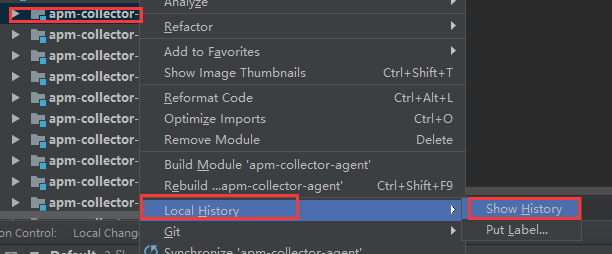
### Git的几个常用的操作：

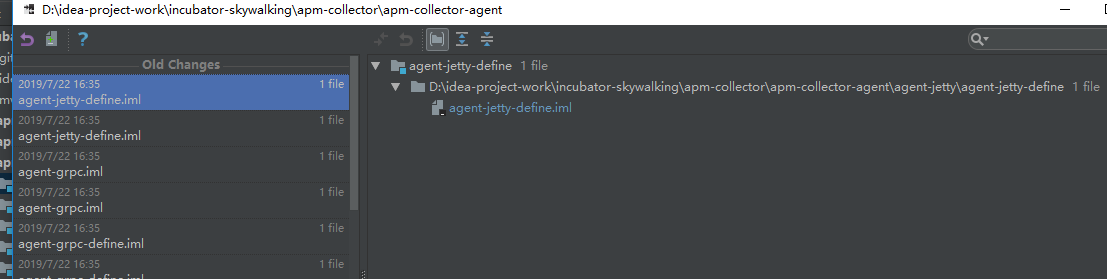




提交代码是先pull—》commit—》push

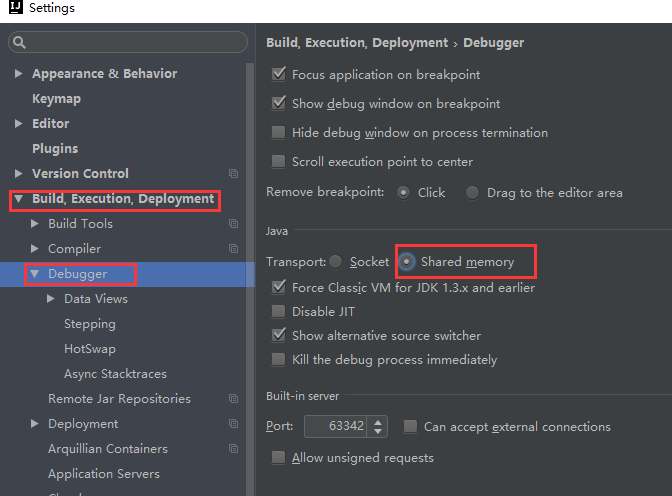
### 查看本地的历史版本



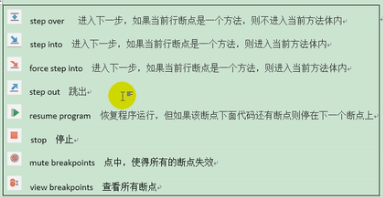


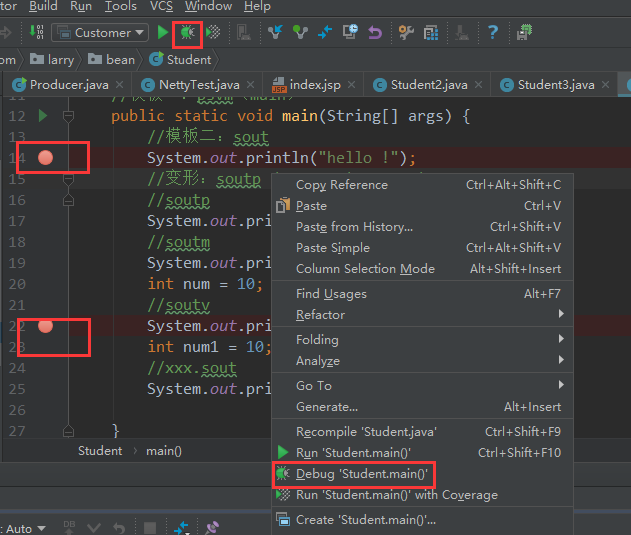
# 断点调试

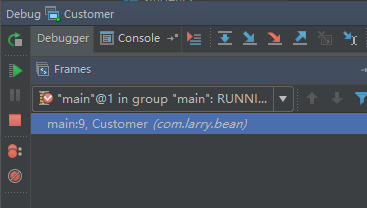
## Debug的设置



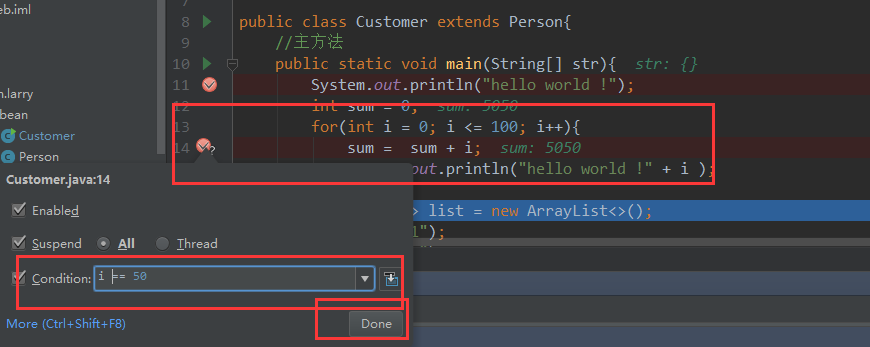
## 常用断点调试快捷键



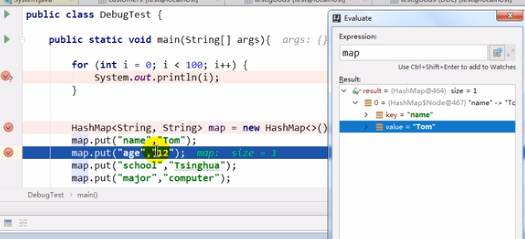




## 条件断点（对断点右键）



## 查看表达式的值（ctrl+u）



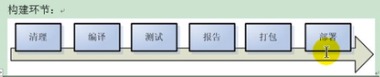
# 配置maven

## Maven介绍

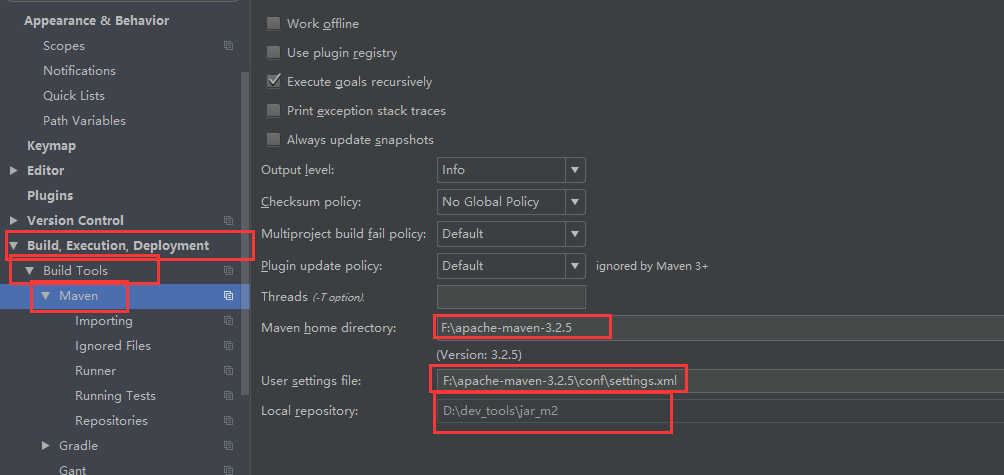


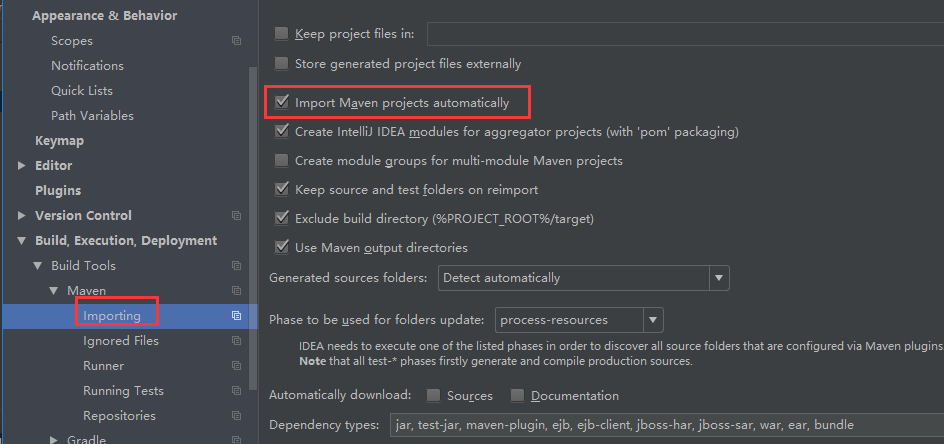
Make—》Ant—》Maven—》Gradle

构建环节：



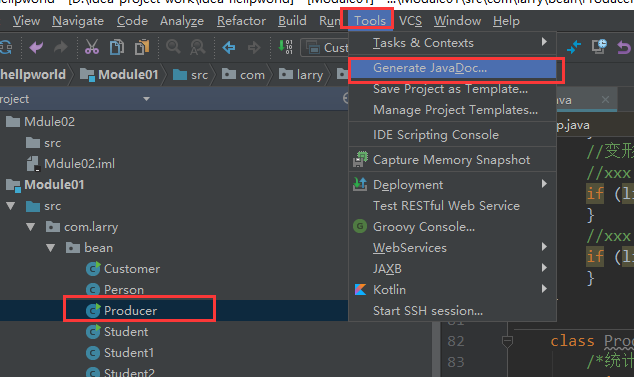
## 配置maven

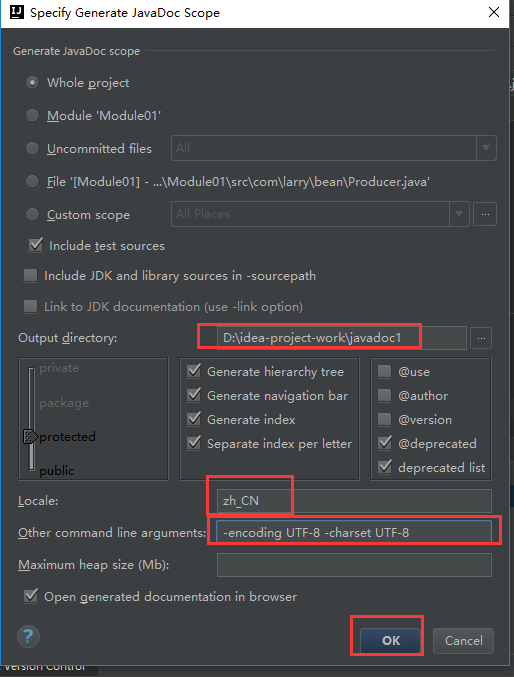


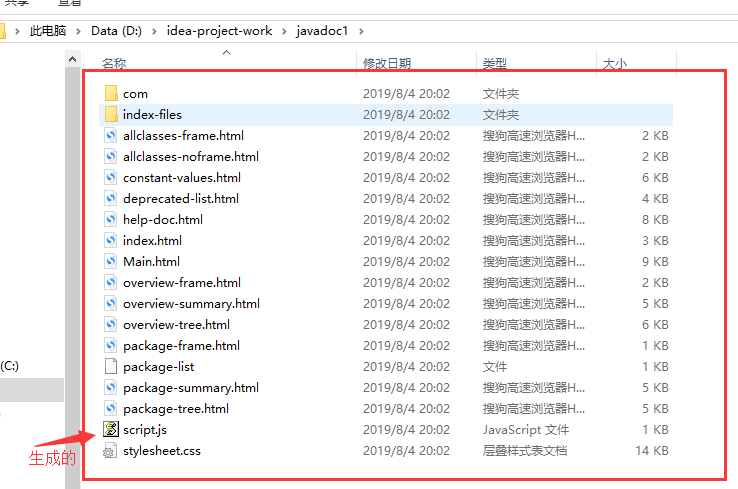


# 生成javadoc理缓存和索引\_插件下载

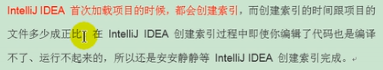
## Javadoc





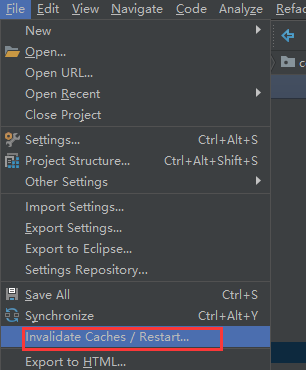


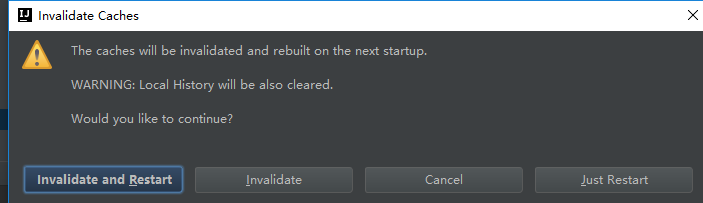
## 缓存和索引的清理





清除缓存





[**IntelliJ Idea 常用快捷键列表**](https://www.cnblogs.com/zhangpengshou/p/5366413.html)

Ctrl+Shift + Enter，语句完成  
“！”，否定完成，输入表达式时按 “！”键  
Ctrl+E，最近的文件  
Ctrl+Shift+E，最近更改的文件  
Shift+Click，可以关闭文件  
Ctrl+[ OR ]，可以跑到大括号的开头与结尾  
Ctrl+F12，可以显示当前文件的结构  
Ctrl+F7，可以查询当前元素在当前文件中的引用，然后按 F3 可以选择  
Ctrl+N，可以快速打开类  
Ctrl+Shift+N，可以快速打开文件  
Alt+Q，可以看到当前方法的声明  
Ctrl+P，可以显示参数信息  
Ctrl+Shift+Insert，可以选择剪贴板内容并插入  
Alt+Insert，可以生成构造器/Getter/Setter等  
Ctrl+Alt+V，可以引入变量。例如：new String(); 自动导入变量定义  
Ctrl+Alt+T，可以把代码包在一个块内，例如：try/catch  
Ctrl+Enter，插入下一行  
Ctrl+Alt+L，格式化代码  
Ctrl+Alt+I，将选中的代码进行自动缩进编排，这个功能在编辑 JSP 文件时也可以工作  
Alt+ Enter，优化导入的类和包  
Ctrl+R，替换文本  
Ctrl+F，查找文本  
Ctrl+Shift+Space，自动补全代码  
Ctrl+空格，代码提示（与系统输入法快捷键冲突）  
Ctrl+Shift+Alt+N，查找类中的方法或变量  
Alt+Shift+C，最近的更改  
Alt+Shift+Up/Down，上/下移一行  
Shift+F6，重构 – 重命名  
Ctrl+X，删除行  
Ctrl+D，复制行  
Ctrl+/或Ctrl+Shift+/，注释（//或者/\*\*/）  
Ctrl+J，自动代码（例如：serr）  
Ctrl+Alt+J，用动态模板环绕  
Ctrl+H，显示类结构图（类的继承层次）  
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  
Ctrl+Alt+left/right，返回至上次浏览的位置  
Alt+left/right，切换代码视图  
Alt+Up/Down，在方法间快速移动定位  
Ctrl+Shift+Up/Down，向上/下移动语句  
F2 或 Shift+F2，高亮错误或警告快速定位  
Tab，代码标签输入完成后，按 Tab，生成代码  
Ctrl+Shift+F7，高亮显示所有该文本，按 Esc 高亮消失  
Alt+F3，逐个往下查找相同文本，并高亮显示  
Ctrl+Up/Down，光标中转到第一行或最后一行下  
Ctrl+B/Ctrl+Click，快速打开光标处的类或方法（跳转到定义处）  
Ctrl+Alt+B，跳转到方法实现处  
Ctrl+Shift+Backspace，跳转到上次编辑的地方  
Ctrl+O，重写方法  
Ctrl+Alt+Space，类名自动完成  
Ctrl+Alt+Up/Down，快速跳转搜索结果  
Ctrl+Shift+J，整合两行  
Alt+F8，计算变量值  
Ctrl+Shift+V，可以将最近使用的剪贴板内容选择插入到文本  
Ctrl+Alt+Shift+V，简单粘贴  
Shift+Esc，不仅可以把焦点移到编辑器上，而且还可以隐藏当前（或最后活动的）工具窗口  
F12，把焦点从编辑器移到最近使用的工具窗口  
Shift+F1，要打开编辑器光标字符处使用的类或者方法 Java 文档的浏览器  
Ctrl+W，可以选择单词及语句  
Ctrl+Shift+W，取消选择光标所在词  
Alt+F7，查找整个工程中使用地某一个类、方法或者变量的位置  
Ctrl+I，实现方法  
Ctrl+Shift+U，大小写转化  
Ctrl+Y，删除当前行

Shift+Enter，向下插入新行  
psvm/sout，main/System.out.println(); Ctrl+J，查看更多  
Ctrl+Shift+F，全局查找  
Ctrl+F，查找/Shift+F3，向上查找/F3，向下查找  
Ctrl+Shift+S，高级搜索  
Ctrl+U，转到父类  
Ctrl+Alt+S，打开设置对话框  
Alt+Shift+Inert，开启/关闭列选择模式  
Ctrl+Alt+Shift+S，打开当前项目/模块属性  
Ctrl+G，定位行  
Alt+Home，跳转到导航栏  
Ctrl+Enter，上插一行  
Ctrl+Backspace，按单词删除  
Ctrl+”+/-”，当前方法展开、折叠  
Ctrl+Shift+”+/-”，全部展开、折叠  
【调试部分、编译】  
Ctrl+F2，停止  
Alt+Shift+F9，选择 Debug  
Alt+Shift+F10，选择 Run  
Ctrl+Shift+F9，编译  
Ctrl+Shift+F10，运行  
Ctrl+Shift+F8，查看断点  
F8，步过  
F7，步入  
Shift+F7，智能步入  
Shift+F8，步出  
Alt+Shift+F8，强制步过  
Alt+Shift+F7，强制步入  
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  
Ctrl+F8，切换行断点  
Ctrl+F9，生成项目  
Alt+1，项目  
Alt+2，收藏  
Alt+6，TODO  
Alt+7，结构  
Ctrl+Shift+C，复制路径  
Ctrl+Alt+Shift+C，复制引用，必须选择类名  
Ctrl+Alt+Y，同步  
Ctrl+~，快速切换方案（界面外观、代码风格、快捷键映射等菜单）  
Shift+F12，还原默认布局  
Ctrl+Shift+F12，隐藏/恢复所有窗口  
Ctrl+F4，关闭  
Ctrl+Shift+F4，关闭活动选项卡  
Ctrl+Tab，转到下一个拆分器  
Ctrl+Shift+Tab，转到上一个拆分器  
【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  
【查找】  
Ctrl+F，查找  
Ctrl+R，替换  
F3，查找下一个  
Shift+F3，查找上一个  
Ctrl+Shift+F，在路径中查找  
Ctrl+Shift+R，在路径中替换  
Ctrl+Shift+S，搜索结构  
Ctrl+Shift+M，替换结构  
Alt+F7，查找用法  
Ctrl+Alt+F7，显示用法  
Ctrl+F7，在文件中查找用法  
Ctrl+Shift+F7，在文件中高亮显示用法

**IntelliJ IDEA 2019 快捷键终极大全**

* 自动代码
* 查询快捷键
* 其他快捷键
* 调试快捷键
* 重构
* 十大Intellij IDEA快捷键
  + 1 智能提示
  + 2 重构
  + 3 代码生成
  + 4 编辑
  + 5 查找打开
  + 6 其他辅助
  + 太难割舍，前三名并列吧！

# 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。

1. Ctrl+Alt+O 优化导入的类和包
2. Alt+Insert 生成代码(如get,set方法,构造函数等) 或者右键（Generate）
3. fori/sout/psvm + Tab
4. Ctrl+Alt+T 生成try catch 或者 Alt+enter
5. CTRL+ALT+T 把选中的代码放在 TRY{} IF{} ELSE{} 里
6. Ctrl + O 重写方法
7. Ctrl + I 实现方法
8. Ctr+shift+U 大小写转化
9. ALT+回车 导入包,自动修正
10. ALT+/ 代码提示
11. CTRL+J 自动代码
12. Ctrl+Shift+J，整合两行为一行
13. CTRL+空格 代码提示
14. CTRL+SHIFT+SPACE 自动补全代码
15. CTRL+ALT+L 格式化代码
16. CTRL+ALT+I 自动缩进
17. CTRL+ALT+O 优化导入的类和包
18. ALT+INSERT 生成代码(如GET,SET方法,构造函数等)
19. CTRL+E 最近更改的代码
20. CTRL+ALT+SPACE 类名或接口名提示
21. CTRL+P 方法参数提示
22. CTRL+Q，可以看到当前方法的声明
23. Shift+F6 重构-重命名 (包、类、方法、变量、甚至注释等)
24. Ctrl+Alt+V 提取变量

# 查询快捷键

1. Ctrl＋Shift＋Backspace可以跳转到上次编辑的地
2. CTRL+ALT+ left/right 前后导航编辑过的地方
3. ALT+7 靠左窗口显示当前文件的结构
4. Ctrl+F12 浮动显示当前文件的结构
5. ALT+F7 找到你的函数或者变量或者类的所有引用到的地方
6. 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
7. Ctrl+Shift+Alt+N 查找类中的方法或变量
8. 双击SHIFT 在项目的所有目录查找文件
9. Ctrl+N 查找类
10. Ctrl+Shift+N 查找文件
11. CTRL+G 定位行
12. CTRL+F 在当前窗口查找文本
13. CTRL+SHIFT+F 在指定窗口查找文本
14. CTRL+R 在 当前窗口替换文本
15. CTRL+SHIFT+R 在指定窗口替换文本
16. ALT+SHIFT+C 查找修改的文件
17. CTRL+E 最近打开的文件
18. F3 向下查找关键字出现位置
19. SHIFT+F3 向上一个关键字出现位置
20. 选中文本，按Alt+F3 ，高亮相同文本，F3逐个往下查找相同文本
21. F4 查找变量来源
22. CTRL+SHIFT+O 弹出显示查找内容
23. Ctrl+W 选中代码，连续按会有其他效果
24. F2 或Shift+F2 高亮错误或警告快速定位
25. Ctrl+Up/Down 光标跳转到第一行或最后一行下
26. Ctrl+B 快速打开光标处的类或方法
27. CTRL+ALT+B 找所有的子类
28. CTRL+SHIFT+B 找变量的类
29. Ctrl+Shift+上下键 上下移动代码
30. Ctrl+Alt+ left/right 返回至上次浏览的位置
31. Ctrl+X 删除行
32. Ctrl+D 复制行
33. Ctrl+/ 或 Ctrl+Shift+/ 注释（// 或者/…/ ）
34. Ctrl+H 显示类结构图
35. Ctrl+Q 显示注释文档
36. Alt+F1 查找代码所在位置
37. Alt+1 快速打开或隐藏工程面板
38. Alt+ left/right 切换代码视图
39. ALT+ ↑/↓ 在方法间快速移动定位
40. CTRL+ALT+ left/right 前后导航编辑过的地方
41. Ctrl＋Shift＋Backspace可以跳转到上次编辑的地
42. Alt+6 查找TODO

# 其他快捷键

1. SHIFT+ENTER 另起一行
2. CTRL+Z 倒退(撤销)
3. CTRL+SHIFT+Z 向前(取消撤销)
4. CTRL+ALT+F12 资源管理器打开文件夹
5. ALT+F1 查找文件所在目录位置
6. SHIFT+ALT+INSERT 竖编辑模式
7. CTRL+F4 关闭当前窗口
8. Ctrl+Alt+V，可以引入变量。例如：new String(); 自动导入变量定义
9. Ctrl+~，快速切换方案（界面外观、代码风格、快捷键映射等菜单）

# 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。

1. alt+F8 debug时选中查看值
2. Alt+Shift+F9，选择 Debug
3. Alt+Shift+F10，选择 Run
4. Ctrl+Shift+F9，编译
5. Ctrl+Shift+F8，查看断点
6. F7，步入
7. Shift+F7，智能步入
8. Alt+Shift+F7，强制步入
9. F8，步过
10. Shift+F8，步出
11. Alt+Shift+F8，强制步过
12. Alt+F9，运行至光标处
13. Ctrl+Alt+F9，强制运行至光标处
14. F9，恢复程序
15. Alt+F10，定位到断点

# 重构

1. Ctrl+Alt+Shift+T，弹出重构菜单
2. Shift+F6，重命名
3. F6，移动
4. F5，复制
5. Alt+Delete，安全删除
6. 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。关注Java技术栈微信公众号，在后台回复关键字：IDEA，可以获取一份栈长整理的 IDEA 最新技术干货。

## 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。关注Java技术栈微信公众号，在后台回复关键字：IDEA，可以获取一份栈长整理的 IDEA 最新技术干货。

## 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