# 快捷键

## 系统默认快捷键

alt+insert--- getter和setter方法

Ctrl+Alt+shift+C ----得到全限定类名 或者 选中类名 copy reference

Ctrl+Alt+v ----返回值自动补全

Ctrl+F9:使模板引擎编译

查找到出错的地方：F2

查找功能：CTRL + SHIFT + A

快捷键 :backSpace

Ctrl+1:大小写转换

Ctrl+2:驼峰转换

### alt+enter:

1.提供代码提示

2.自动创建函数：只要写好函数名称，Alt+Enter一下，它就能帮我们自动创建好函数；

3list replace：写完一个list用for遍历；Alt+Enter一下，他就告诉你还有更好的重构方法

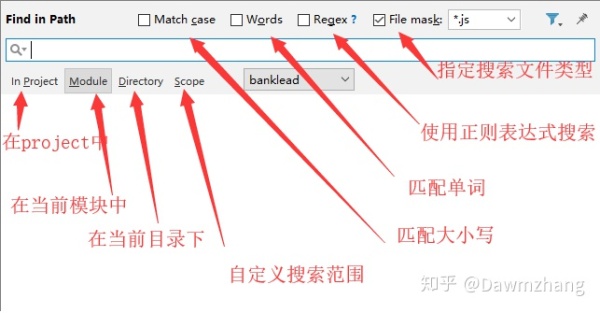
4.实现接口:当我们需要实现接口的时候，只需要在接口的名称上Alt+Enter就能做到

5.单词拼写

6.导包

### Ctrl+Shift+F 搜索

作用：快速搜索字符串、



## 视图查看

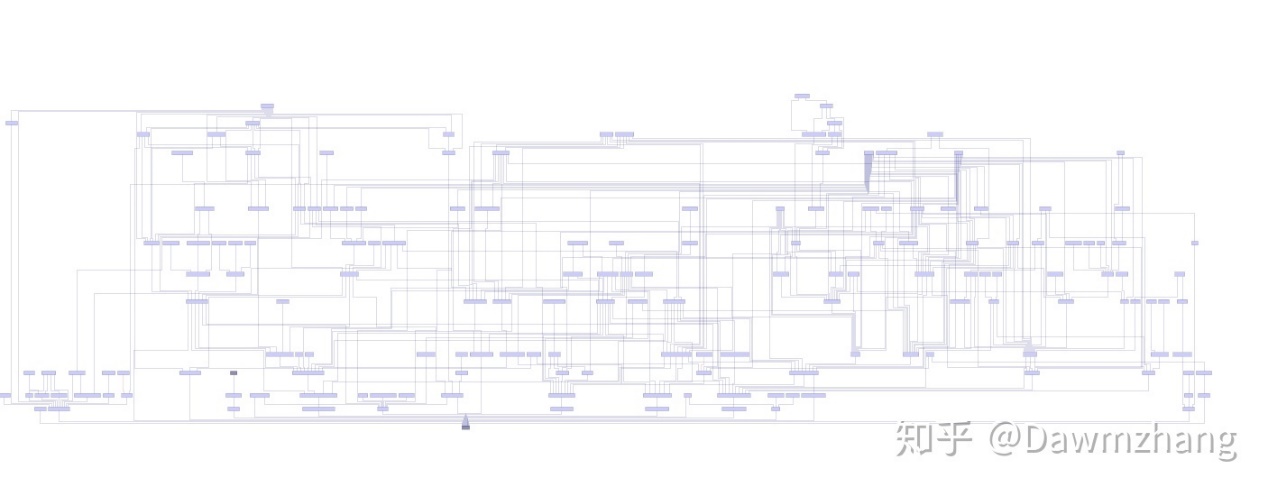
Ctrl+F12 查看file，method结构图、类继承机构图

（不知道方法结构，Ctrl+F12一下，方法，参数，返回值，一清二楚的展现出来）

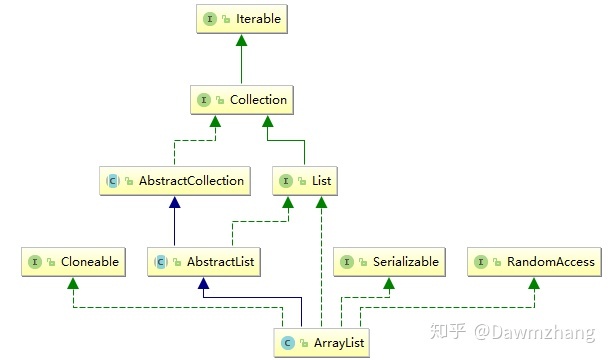
Ctrl+shift+Alt+U 查看maven依赖，类图

使用maven做项目时，依赖往往非常多，在配置依赖的文件中只需要Ctrl+shift+Alt+U，所有依赖都以拓扑图的方式展现出来，所有的关系和位置都能找到；

如下图（抱歉，项目依赖太多，具体的依赖需要放大）



类图：我们阅读源码的时候，想知道各个类之间的关系怎么办，Ctrl+shift+Alt+U，所有依赖都以拓扑图的方式展现出来，所有的关系和位置都能找到，以ArrayList为例：



Ctrl＋Alt+H 查看方法调用层次；

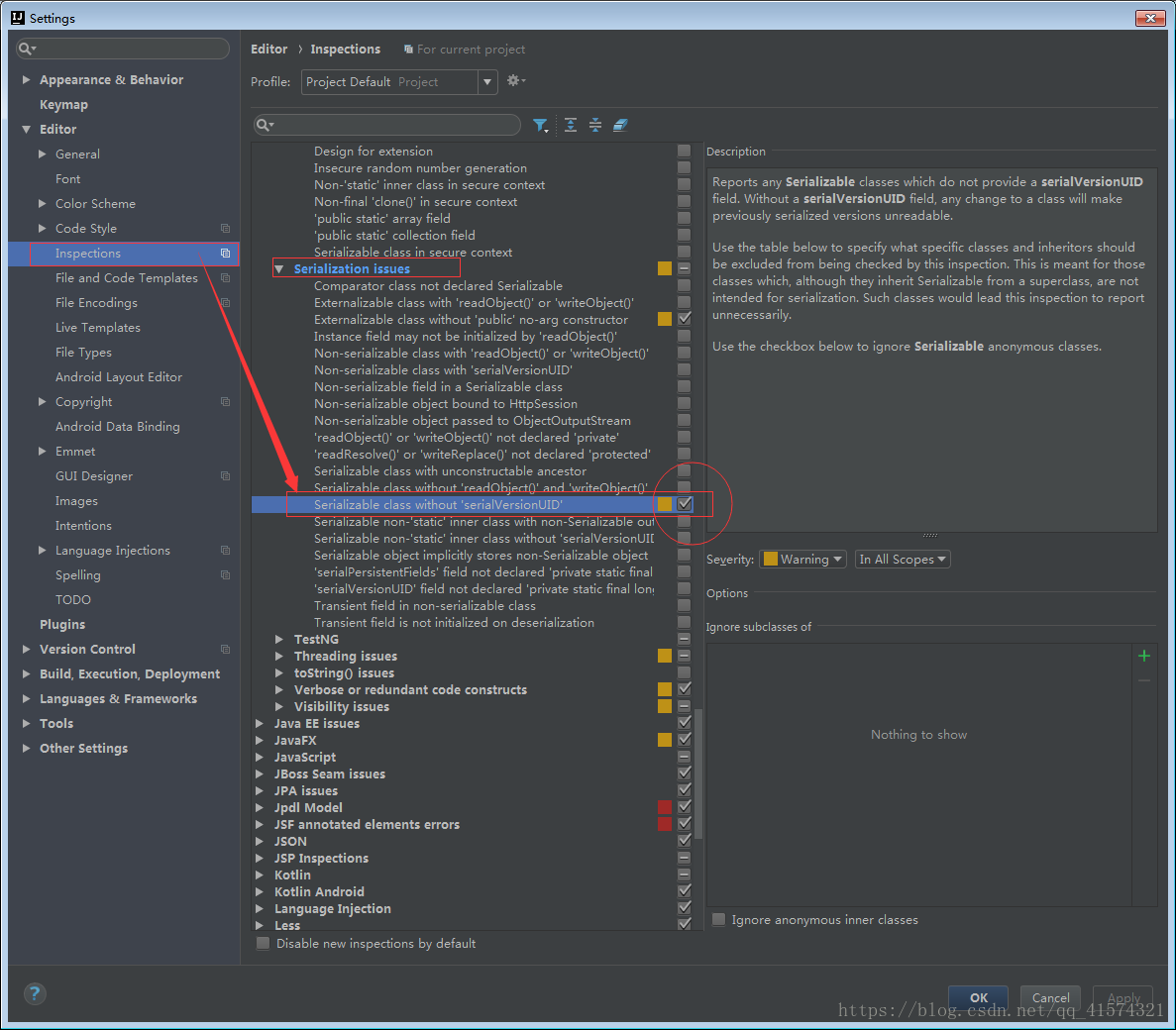
IDEA 录

# IDEA设置

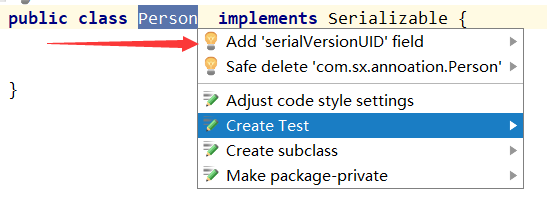
## IDEA自动生成序列化Id

### 开启生成自动序列化

ctrl+alt+s –> inspections –>Serialization issues

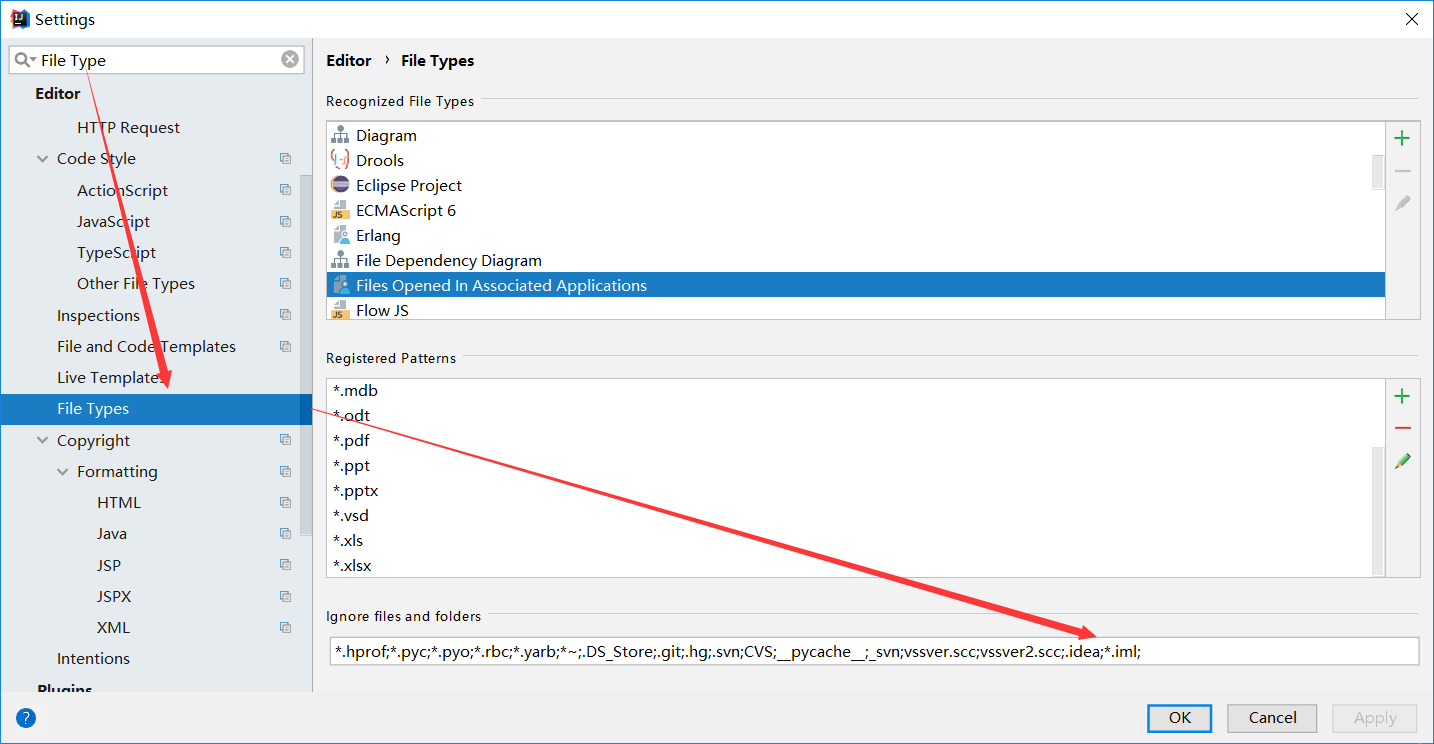


2、鼠标放在实现Serializable接口的类名，alt+enter，选择生成ID



## 隐藏IDEA的项目文件

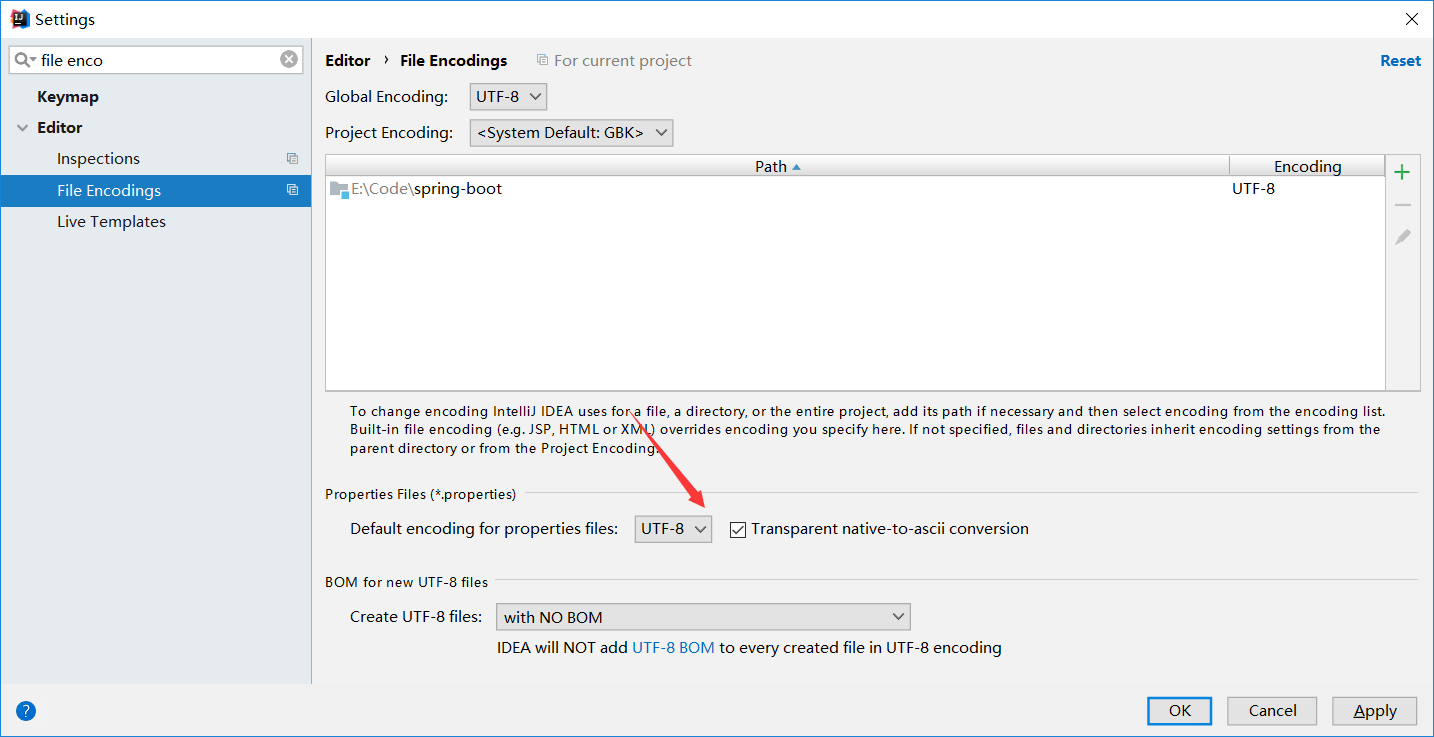
ctrl+alt+s



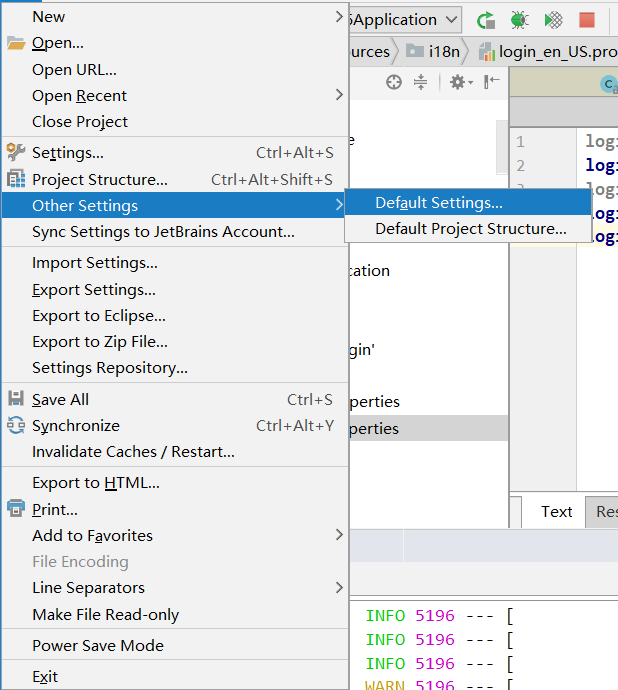
## 设置文件的编码

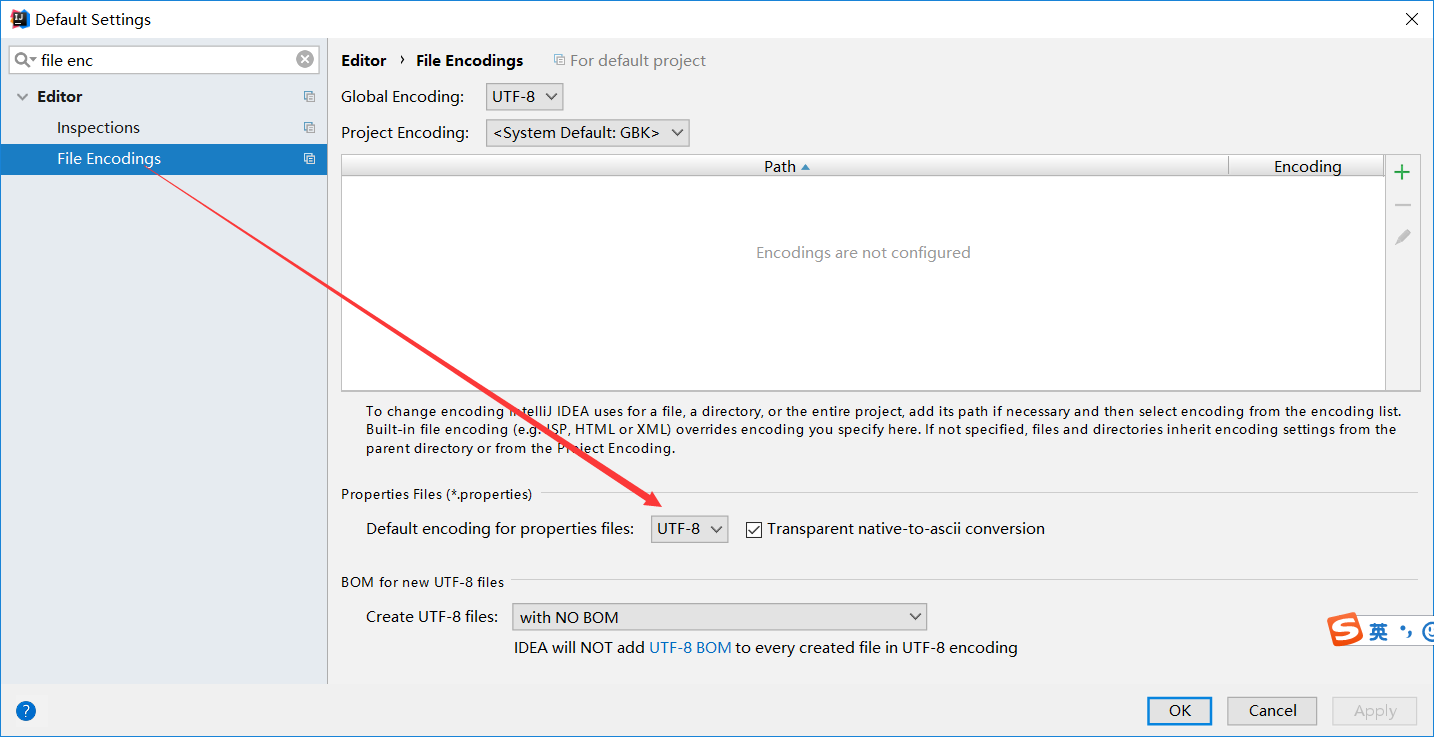
### 本次项目设置

让文件运行时转为ascii码

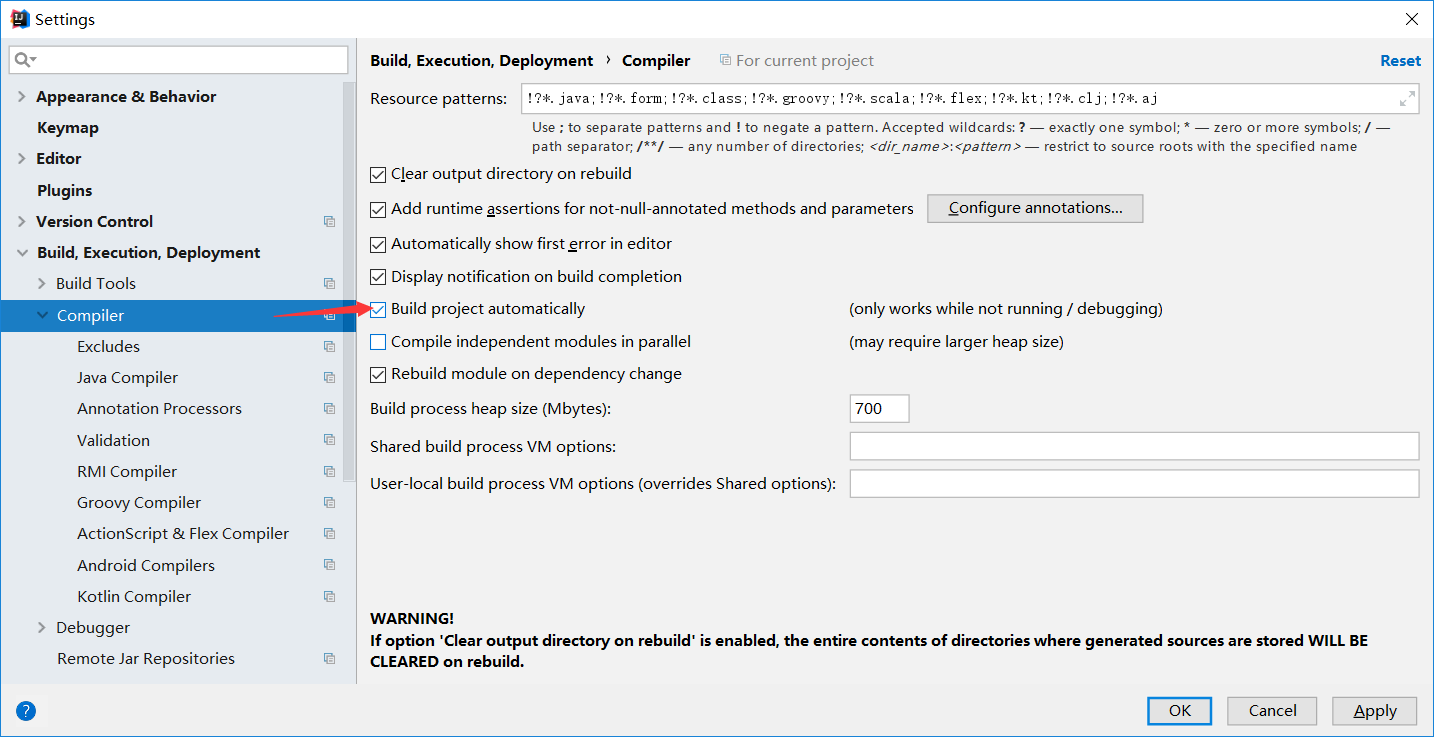


### 全局设置





## 设置修改后自动编译



2、打开设置 --> 查找Registry --> 找到并勾选

piler.automake.allow.when.app.running

如果你的 IDEA 比较新，可能找不到这个



然后关闭设置，按上面设置的快键键，如 Command + Shift + M

找到并勾选compiler.automake.allow.when.app.running



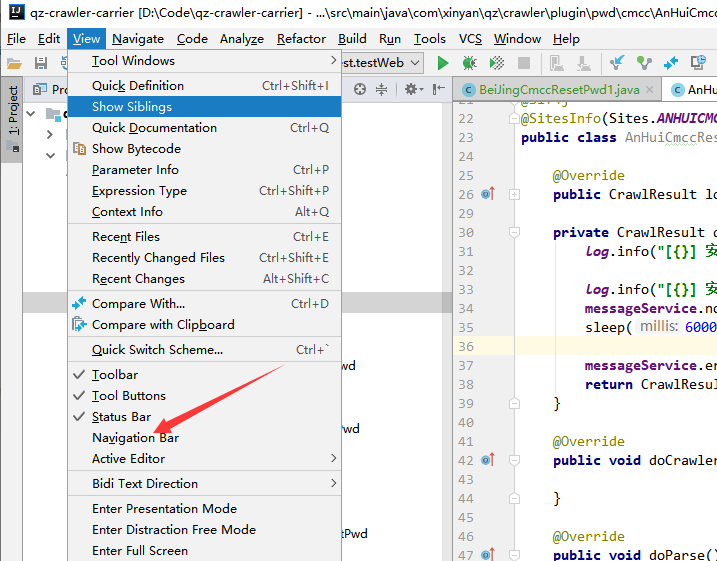
最后重启idea

## junit的Test

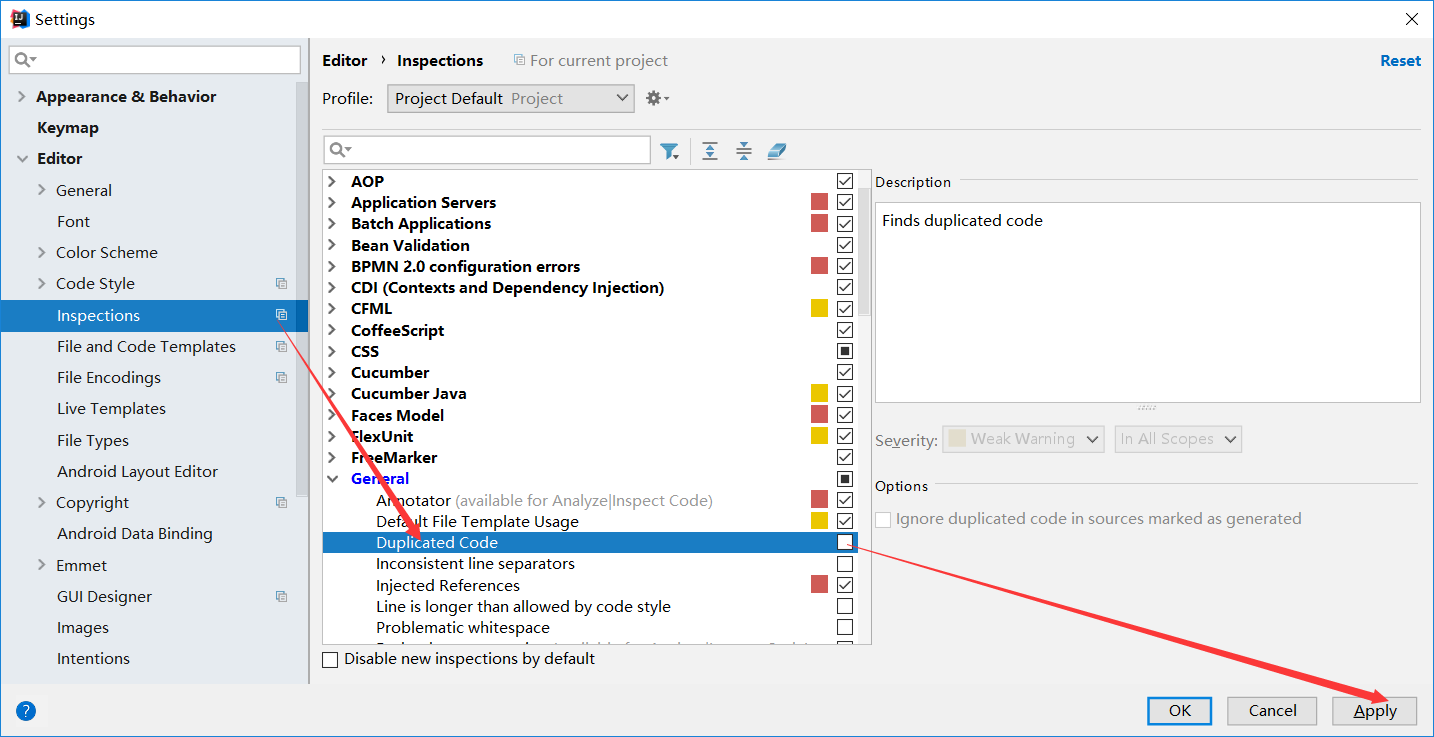
<https://blog.csdn.net/QQljh123456/article/details/81583786>

## 配置maven仓库

## 去掉导航栏

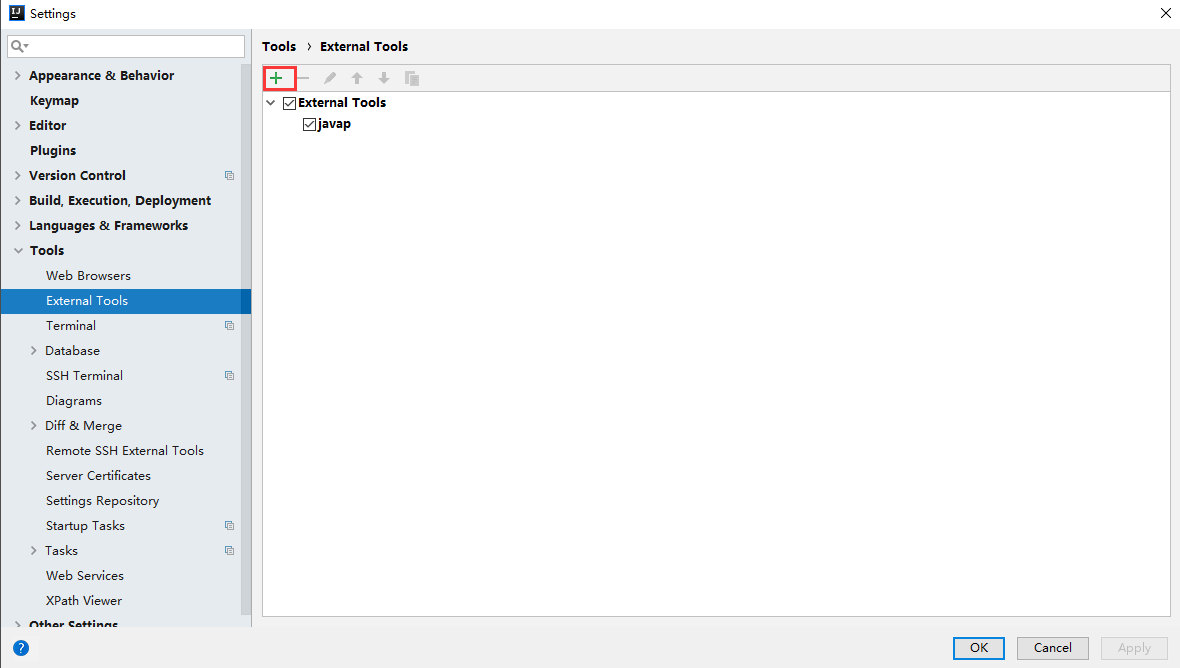


## 去除重复代码下波浪线

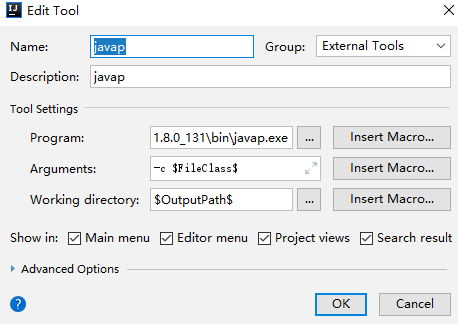


## 配置反编译工具

### External Tools配置框



### 配置javap



说一下三个框框分别表示什么意思，从上到下：

1）Program：jdk的javap工具在系统的绝对位置，从上面可以看出有$JDKPath$这个属性

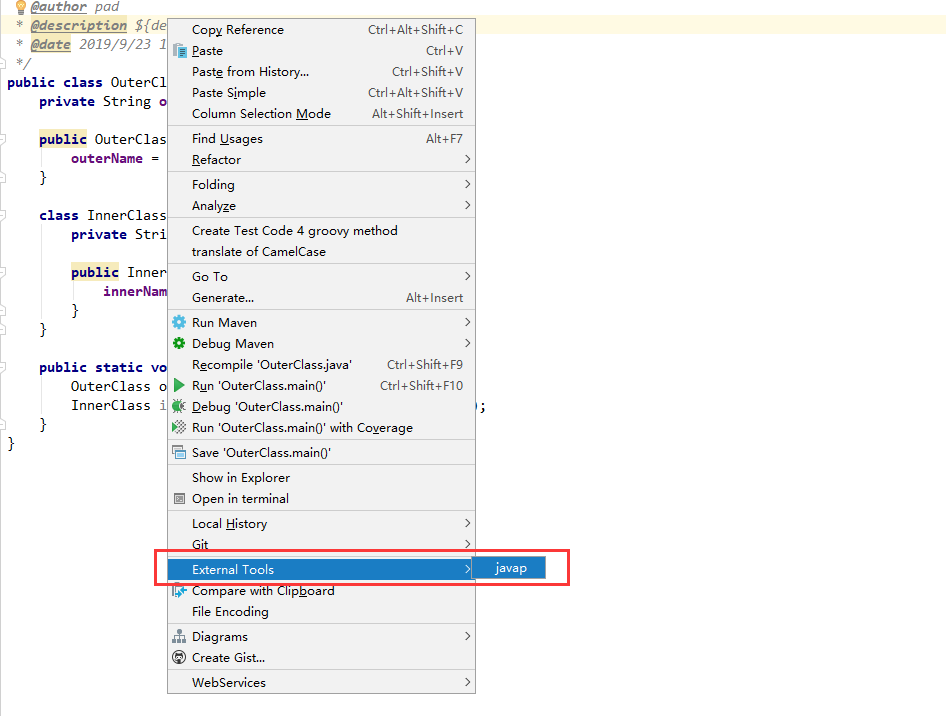
2）Parameters：其实就是通用的反编译的命令，"-c  <%要反编译的类名%>"

 在这里，要注意的一点是，命令可以自己选择，博主记得还有一个"-v  <%要反编译的类名%>"的命令也挺常用的。

     如果想要两种命令都有，可以再新增一个工具，就是再从第二步开始。只要保证名字不一样就行啦。

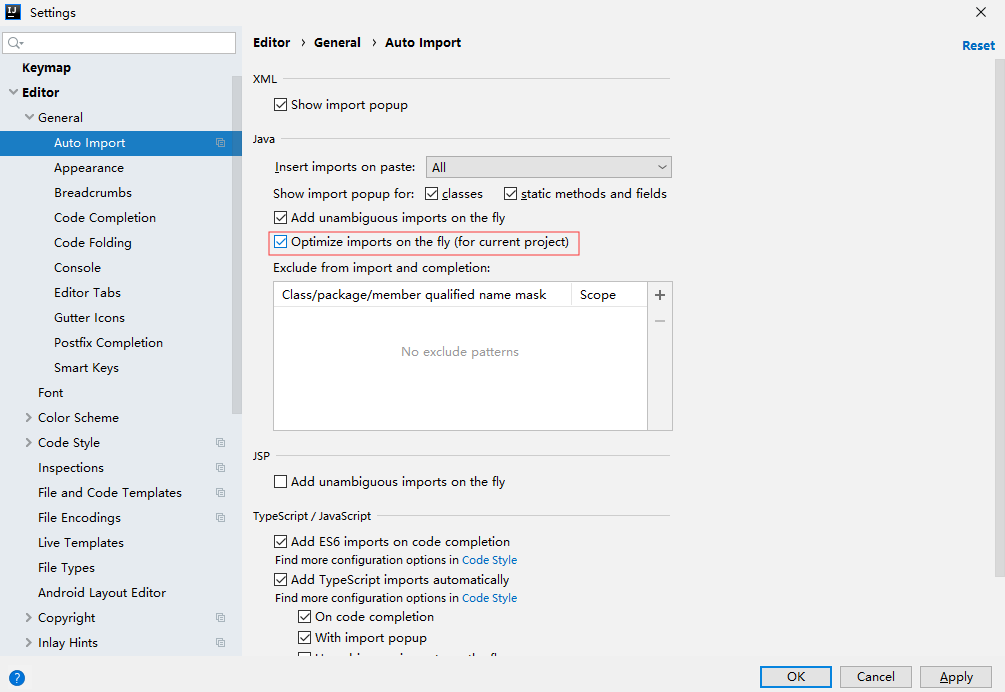
3）Working directory：这个其实就是要反编译的类所在的项目在系统的绝对位置配置好后，点击ok，再点击Settings框的ok，就可以啦

### 使用



使用起来很简单，我们只需要在想要反编译的java文件的编辑框右键，这时候会出现External Tools的这个选项，点击，有咱们刚才配的javap的工具，点击即可（若没有这个External Tools这个选项，说明你忘记点击ok啦，工具没有配置上去）。

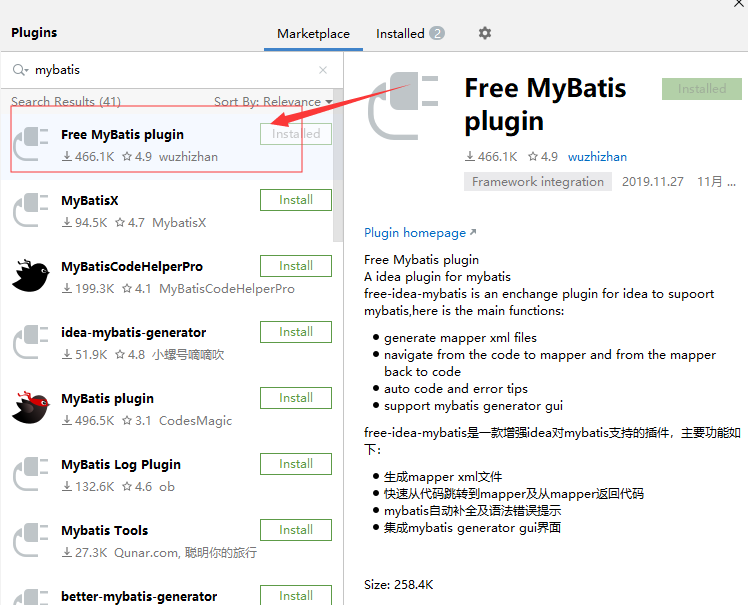
## 设置IDEA自动导包优化



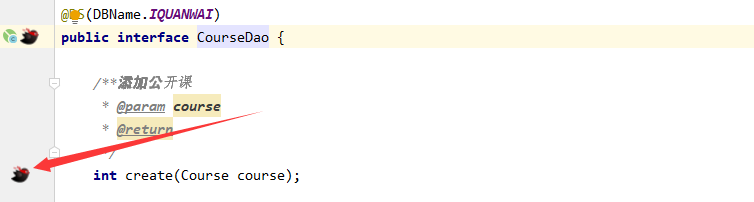
## 配置mybatisX插件

### Free Mybatis plugin跳转插件

1、下载

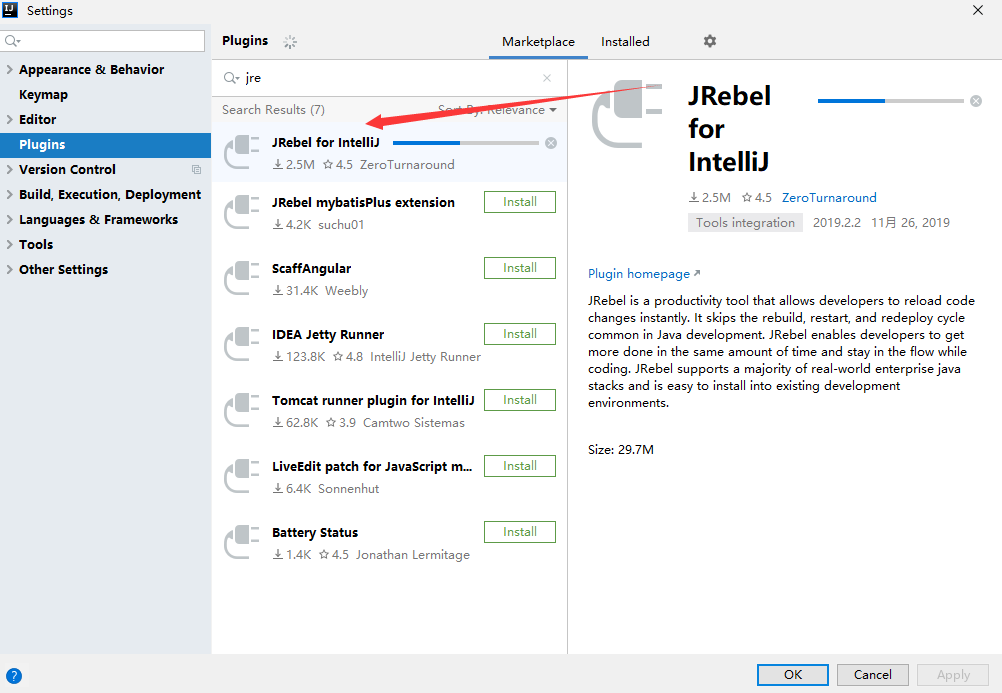


2、使用



## 安裝熱部署插件

### 下载插

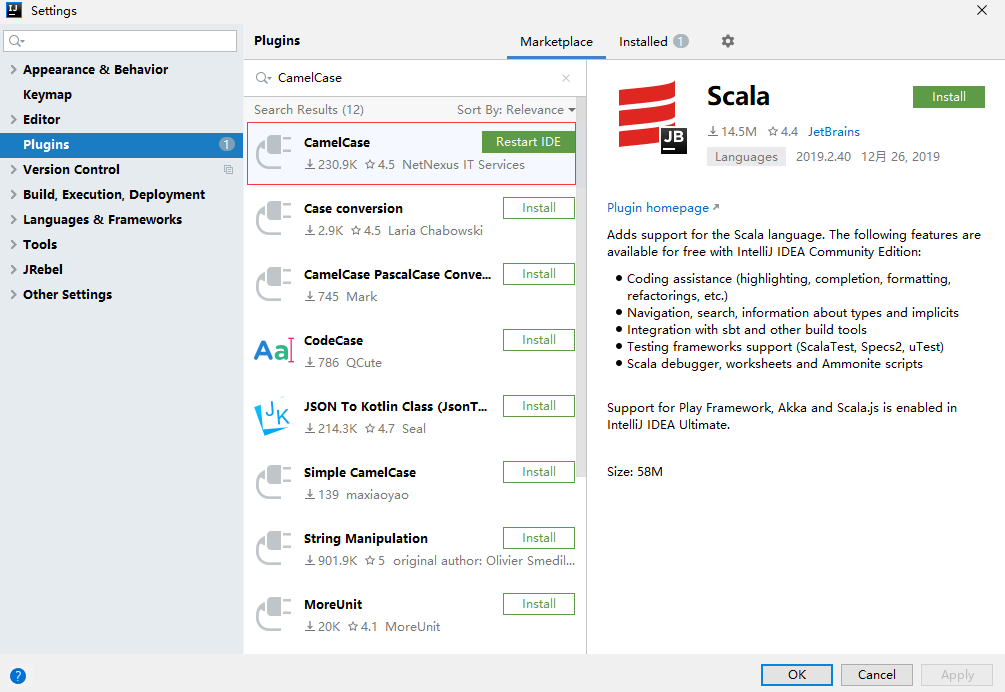


### 破解

<https://blog.csdn.net/is_chenzh/article/details/82148157>

## 驼峰插件

### 下载



### 使用

# 模板

## IDEA自带模板

### main方法生成（psvm）

|  |
| --- |
| public static void main(String[] args) {  } |

### 输出语句

普通输出语句（sout）

System.out.println();

带方法参数的输出语句(soutp)

System.out.println("args = [" + args + "]");

带方法名字的输出语句

System.out.println("helloword.main");

带变量的输出语句（soutv）

int i = 0;

System.out.println("i = " + i);

注：只能选择就近i。

### for循环

普通for循环(fori)

|  |
| --- |
| for (int i = 0; i < ; i++) { //代码} |

增强for循环(iter)

|  |
| --- |
| for (String arg : args) {//代码} |

普通for循环增强版:循环数组(itar)

|  |
| --- |
| for (int i = 0; i < args.length; i++) {  String arg = args[i];  } |

### list普通for循环(list.fori)

|  |
| --- |
| List list= new ArrayList();  for (int i = 0; i < list.size(); i++) {  } |

### list循环(list.for)

增强for循环遍历，如果没有使用泛型，会自动设置为object类型

|  |
| --- |
| for (Object o : list) {  } |

### list逆序遍历(list.forr)

|  |
| --- |
| for (int i = list.size() - 1; i >= 0; i--) {  } |

### 条件判断是否为null(ifn)

判断list是否为null

|  |
| --- |
| List list= new ArrayList();  if (list == null) {  } |

### 条件判断是否非null(inn)

|  |
| --- |
| List list1= new ArrayList();  if (list1 != null) {  } |

### 判断为null进阶版

xxx.nn:判断是否不为null

xxx.null判断是否为null

### 常量定义

prsf：

## 设置注释模板

## 自定义模板

# IDEA功能

## 查找替换

### 查找



1、查找的内容

2、查询全部

3、Select All Ocurrence 选中所有匹配的Ctrl+Alt+Shift+J(原有的快捷键)

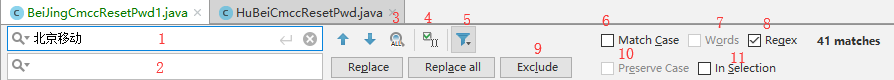
4、过滤

5、Match 匹配大小写

6、Words 匹配的内容必须是单词

7、支持正则表达式。

### 查找替换(Ctrl+R)



1:搜索的内容。

2:要替换的内容。

3:查询全部。

4:选择全部选中的。Ctrl+Alt+Shift+J(原有的快捷键)

5:过滤操作。

6:Match Case 区分搜索内容的大小写

7:word:匹配的内容是一个单词 例如匹配word ，word123 将不会匹配。

8:Regex:使用正则表达式匹配。

9:Exclude排除内容

10：preserve Case 保留大小写

11:in select 在选中的内容中去查找和替换。

# git操作

## 引入控制

