

IntelliJ IDEA 使用心得与常用快捷键

那种酸爽，根本说不出来----- by: Jimi没有Bond

Jimi是谁？ 就是洒家啊！

刚开始学习写Java的时候，用的eclipse，正式工作后，主要用的myeclipse，去年初在前辈的推荐下，在2折的时候买了正版的 IntelliJ IDEA 和 Pycharm，12.0版终生使用，一年更新。

使用前早就久闻其名，据说是最好的Java开发工具。

起先用的非常不习惯，但是花了钱的，硬着头皮用了下去。

短短一个礼拜，彻底爱上了它！就是辣么酸爽！-----谨以此文献给对于拥抱IDEA还存在疑虑的筒子们

JetBrains（这个公司的名称翻译过来就是“大脑喷射？”）公司的宣传语是这么说的：Develop with pleasure！（带着快乐开发！）

Why?

他们咋这么有信心呢？

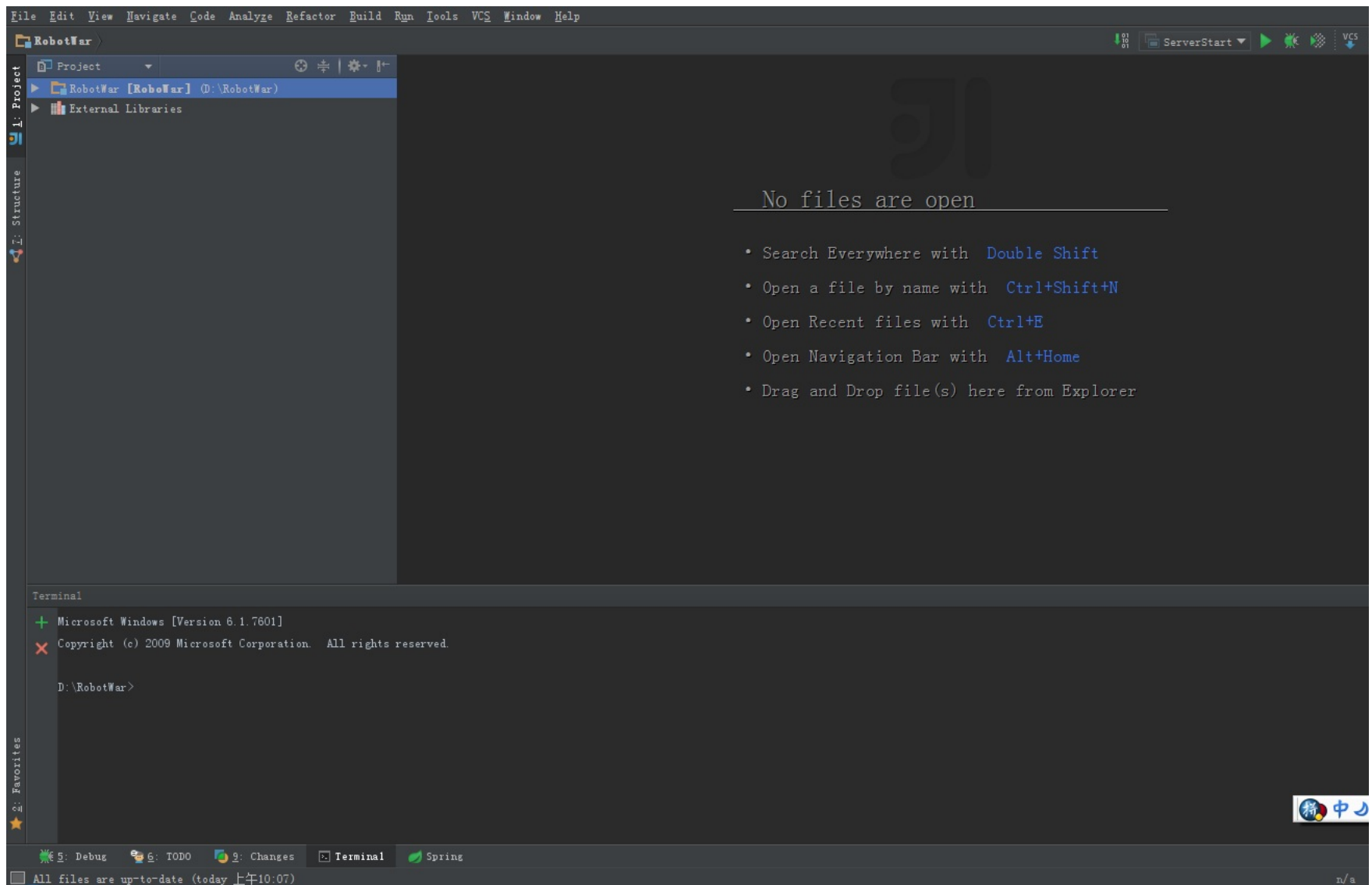
这到底是怎样的一个工具？

跟我一起来数数IDEA特别长的地方：

一.黑色主题 Darcula

眼睛舒服，最重要的是酷！

设置方式：FILE--Settings--Edit--Colors&Fonts--Scheme name



二.智能提示

创建或引入项目的时候，会自动引入缺少的包，真找不着的还可以search in the net

键入代码的时候，区分大小写的智能提示，自动引入包，如果有重名的会列出所有选择，但是比 myeclipse 的更聪明，至于怎么个聪明法，你自个儿试试就知道了

三.工具集成

基本上正式开发的常用工具基本都集成了，而且基本都在你非常容易触到的位置。

说说我比较常用的：

1.ant

你懂的

2.maven

你也懂的

3.SVN

相比之下，IDEA的SVN的提交提供了更多的选项和功能

提交的界面两侧会显示当前文件和修改的部分对比，可以非常方便地检查和修改文件。

提交前：

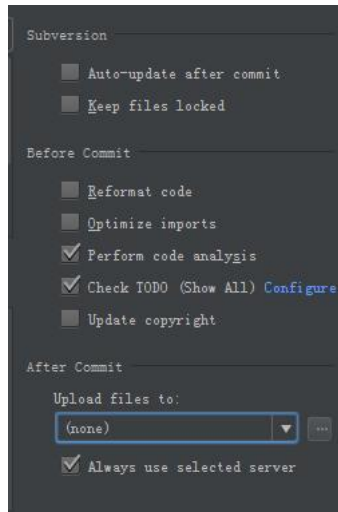
(1).可选择自动格式化

(2).可选择自动忽略没有使用的imports

(3).可选择分析代码

(4).检查是否有TODO

提交后甚至可以选择将改变上传到特定的目录



4.系统终端

有了这个东西，你就不必频繁地切换窗口了

打开方式：Tools--Open Terminal

快捷键：Alt+F12

5.SSH工具

打开方式：Tools--Start SSH session

6.数据库连接工具

打开方式：View--Tool Windows--Database

7.IDEA talk

神奇的东西，你可以联系局域网内其它的IDEA使用者，可以方便地把你的代码show给你的同事

打开方式：View--Tool Windows--IDEA talk

8.Changes

非常方便的changes视图，它会变色显示你所有改动过的文件，而且可以方便的与本地历史或线上历史做对比

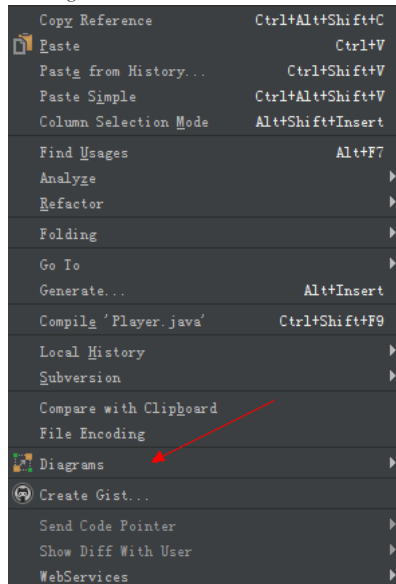
打开方式：View--Tool Windows--Changes

还有git、github、cvs、groovy console以及等等，真没有的你还可以：
FILE--Settings--Plugins

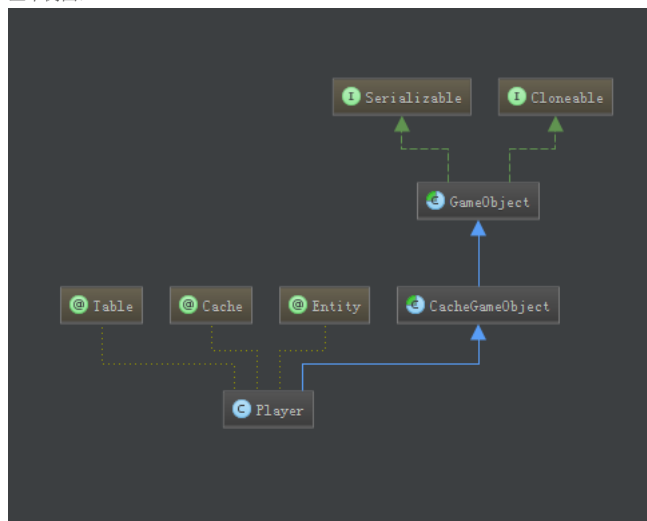
四.强大的绘图工具

读源码看不懂项目结构？写文章画类图好麻烦？

右键diagram帮助你



上个例图：



右键选择 Show Categories 有惊喜，什么值域、方法、构造器、内部类统统有

五.无缝接入eclipse和myeclipse

IDEA可以直接打开有.project文件的项目，也可以将自身项目导出成eclipse的项目。

而且因为IDEA的项目配置文件为1个.iml文件加一个独立文件夹.idea，

所以只要你将两者放入ignore目录，即使在你个人使用IDEA而团队使用其他IDE的情况下，也并不会对其他人造成麻烦。

导出方法：FILE--Export to eclipse

六.更小的体积，更快的速度

用过myeclipse的你明白的

综上，我想你应该能够明白，为啥用IDEA写代码，就是那么愉快，就是那么酸爽，就是那么令人停不下来！

接着分享开发中我比较常用的一些快捷键：

一.找文件找代码找引用相关

1.双击shift

在项目的所有目录查找，就是你想看到你不想看到的和你没想过你能看到的都给你找出来

2.ctrl+f

当前文件查找特定内容

3.ctrl+shift+f

当前项目查找包含特定内容的文件

4.ctrl+n

查找类

5.ctrl+shift+n

查找文件

6.ctrl+e

最近的文件

7.alt+F7

非常非常频繁使用的一个快捷键，可以帮你找到你的函数或者变量或者类的所有引用到的地方

二.编辑相关

1.shift+enter

另起一行

2.ctrl+r

当前文件替换特定内容

3.ctrl+shift+r

当前项目替换特定内容

4.shift+F6

非常非常省心省力的一个快捷键，可以重命名你的类、方法、变量等等，而且这个重命名甚至可以选择替换掉注释中的内容

5.ctrl+d

复制当前行到下一行

6.ctrl+x

剪切当前行

7.ctrl+c \ ctrl+v

大家都懂的

8.ctrl+z

撤销

9.ctrl+shift+z

取消撤销

10.ctrl+k

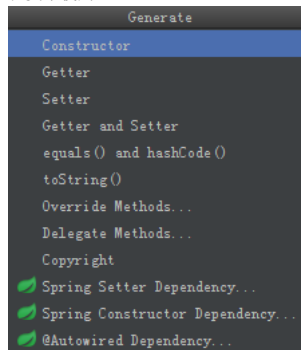
提交代码到SVN

11.ctrl+t

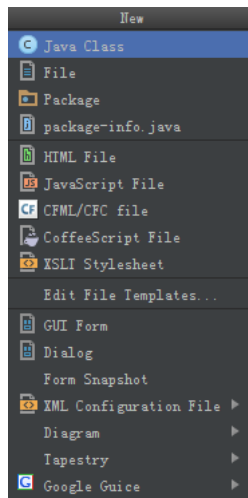
更新代码

12.alt+insert

非常非常以及相当方便的一个组合键，不信您往下看
在类中使用：



可以自动生成构造器、getter/setter等等常用方法
在项目目录上使用：



可用于新建各种文件。

13.alt+enter

又是一个大杀器，有多杀？

例1:

发现代码很“黄”怎么办？

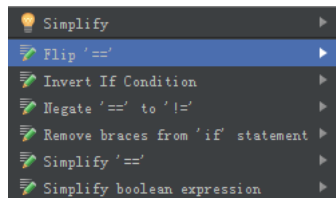
```
//如果猪脚炖好了
if (futureTask.get() == true) {
    System.out.println("-----继续炒菜-----");
}
```

鼠标移上去喵一下：

```
//如果猪脚炖好了
if (futureTask.get() == true) {
    System.out.println("-----继续炒菜-----");
}
```

'futureTask.get() == true' can be simplified to 'futureTask.get()' more... (Ctrl+F1)

试试alt+enter:



选择simplify看看

```
//如果猪脚炖好了
if (futureTask.get()) {
    System.out.println("-----继续炒菜-----");
}
```

我勒个去，代码简洁了有木有啊！

PS:黄色警告编译可以通过，不影响程序的正常运行，一般都是一些对于代码优化的建议，我遇到过的有：

(1).can be simplify

代码可以简化

(2).Variable is never use

声明的变量从未被使用

(3)unnecessary boxing

不必要的声明包装

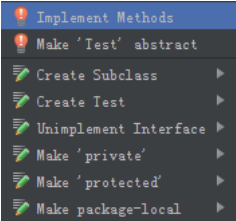
以及等等等，多留意下，对于编码的某些细节能够更了解。

例2:

写个类实现某个接口，加上 implements Runnable 后你就会发现这条红色的波浪线:

```
public static class Test implements Runnable {
```

咱试试在这行上来个 alt+enter:



感动地哭了有木有? 你想到的没想到的IDEA都帮你想到了。

看到代码里提示的各种红XX，就试试alt+enter吧！什么未抛异常啊没有try catch啊都能搞得定。

14.ctrl+alt+L

自动格式化代码，我已经养成了写完代码就来一发的习惯。

你可以个性化设置你自己的代码风格：File--Settings--CodeStyle

-----一切-----一切-----一切-----一切-----一切-----一切-----

凡事有利有弊，IDEA同样是把双刃剑，最后说说它不好的地方

1.最重要的就是“贵”!

专业版个人许可2折可以接受，5折小贵，不打折桑不起!!! 囧RZ~

美刀的消费水准令国内大多在屌丝线上挣扎的猿类们望而却步。

好在JetBrains是有社区版的，最近对于学生也有免费的Key。

对于使用破解版的小伙伴，建议自己用用就好了，不必太声张。

用户养成良好的软件付费习惯，才能让我们的工作变得越来越有价值。

2.集成太完善了

集成的工具完全能够应付大多数的工作需求，对于不求甚解又偷懒的筒子们，直接使用，可能很多工具的细节，你都不会了解，也不会再去了解了。

比如：maven,ant

3.不自带JDK

不是缺点的缺点，myeclipse是集成的

差不多了，该撸代码了。

以上这些，就是使用IDEA一年半来几乎所有的心得了。

能力有限，砖头轻拍。

如果你有杀伤力更强大的手段，欢迎交流。

后记：最近偶然发现，这篇文章被人转载了，而且没注明出处，令人蛋疼。

在此重申下，转载，请注明出处。

来自：<http://www.blogjava.net/rockblue1988/archive/2014/10/24/418994.html>

posted on 2014-10-24 12:49 [Jimi](#) 阅读(214003) [评论\(15\)](#) [编辑](#) [收藏](#) 所属分类: [Java](#)