TestEditor

part I

1.IFile ---> IEditorInput

2.调用editor中的方法

3.得到筛选出的keyword，打印出来

4.比较筛选出来的word是不是我们认知上的word

part II

1.part I.1

2.part I.2

3.得到筛选出的关键字的IToken

4.比较IToken和我定义的关键子的Token类型是不是相同的

problem

1.怎么把指定路径的File类型，转换成IFile类型，然后变成IEditorInput。

＊IFile类型很容易转换成File类型的 file ifile.getRawLocation().makeAbsolute().toFile();

＊但是File类型的转换到IFile类型就有一个问题，这个file一定要是在workspace里面的，原因是应为IFile是IResource的子接口

2.从哪里设置接口来得到，codescanner扫描到的word是什么

首先肯定是要调用到codescanner中的私有方法，所以肯定是要先实例我们的scanner

2.1不知道获取word方法的信息是怎么得到的，保存在哪个对象当中，这个是关键，如果知道这个，就能找到我们定义的Document的信息

2.2从哪里调用，写的测试的方法应该放在哪里

想法1：放在一个类的main函数里，然后运行一下

不能在main函数中间使用**IWorkspace** workspace=*ResourcesPlugin*.*getWorkspace*();

**IFile** ifile=workspace.getRoot().getProject("test").getFile("testeditor");

这个语句，You can't just start using Eclipse resources in the main class of an ordinary Java program. You have to start Eclipse using either the eclipse command or by specifying the org.eclipse.equinox.launcher.Main main class. This is because there is a lot of initialization that needs to be done before the workspace can be made available.

所以一开始的想法，会报workspace is closed的错误

想法2：放在一个button里面

erro1：实例化editor的时候设置的site参数是null，然后就会报the site for an editor must be an IeditorSite的错误

最终方案：直接用IDE.openEditor()方法，指定要打开的文件和用哪一个editor打开

3.最关键的问题就是怎么拿到，我的Document的Token还有scan到的words

WordRule 这个类中的fbuffer 就是我们输入的字的流,每键入一下他就会触发一下，比如我打aaa fbuffer.toString() 就是>>>>a<<<<< >>>>>aa<<<<< >>>>>aaa<<<<

26.9.2014 下午开会

接下来的工作就是先找到javaeditor的testcase 然后观察editor到底测试了一些什么

光是editor部分，首先是对editor的一些操作需要测试：editor的开，关，editor action的正常使用