最近，经常用Java读写操作文件，所以总结几点常用的读写文件方法。

头文件：

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.OutputStreamWriter;

其实不用记，需要的时候，IDE会帮你补全。

一般读写都会提示需要try{}catch(){}异常处理,编译器也会帮助提示补全，一般就是IOException或FileNotFoundException。

**读：**

String filePath="T:\\2016busafctripChain\\xxx.xx"；

BufferedReader br=new BufferedReader(new FileReader(new File(filePath)));//这里还可以更复杂，选择以哪种形式读入，这种写法通用

String line=null;

try{

**while**((line=br.readLine())!=**null**){

执行操作items=line.split(",");

}

}

catch()

{

}

**写：**一般会用ArrayList<Type>把结果装入列表中，统一写入文件，这样操作比较方便，我也是这么做的。Type中记得要重写下public String toString()函数，用于写入文件或打印。

BufferedWriter bw;

**try** {

br.close();//关闭读文件指针

bw = **new** BufferedWriter(**new** OutputStreamWriter(**new** FileOutputStream(writeFilePath), "UTF-8"));//用UTF-8格式写入

或者bw = **new** BufferedWriter(**new** FileWriter(**new** File(

writeFilePath)));一般写入

Iterator<Result> it = result.iterator();//利用迭代操作

**while** (it.hasNext()) {

bw.write(it.next() + "\r\n");

}

bw.flush();

bw.close();

} **catch** (IOException e) {

e.printStackTrace();}