1. **函数OrderProcess使用说明文档**

**函数作用：**将所有的数据以格式“问题ID|&|大类名|&|小类名|&|关键词”的形式输入到文本中。

**输入：**聚类文件夹、关键词文件、类命名文件、无效类文件、需要生成的文件

**输出：**数据的单独分类文件

**函数使用：**

String filefloderpath="resource/10";

String keywordspath="resource/关键词";

String classname="resource/Classname.txt";

String alldataResult="resource/全部数据的单独分类.txt";

String uselessfile="resource/useless.txt";

Process pro=**new** Process();

pro.OrderProcess(filefloderpath,classname,keywordspath,alldataResult,uselessfile);

**函数说明：**

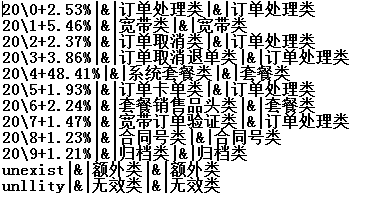
1. filefloderpath 为聚类文件位置，格式为“大文件夹/聚类文件夹”。
2. classname 为每个类的名称文件位置，格式为“大文件夹/类的名称文件”；

其中类的名称文件中的格式为：类的总个数\当前类的序号+百分比|&|小类名|&|大类名，

且最后两行为unexist|&|额外类|&|额外类

unllity|&|无效类|&|无效类

如图所示：



1. keywordspath 为关键词文件位置，格式为“大文件夹/关键词文件”
2. alldataResult 为输出文件位置，格式为“大文件夹/输出文件”
3. uselessfile 为无效数据文件位置，格式为“大文件夹/无效数据文件”
4. **函数deletesymbol使用说明文档**

**函数作用：**将问题ID中的一些符号（引号）去除。

**输入：**数据源文件

**输出：**去除了一些符号的数据源文件

**函数使用：**

String datasource="resource/2月test1W.txt";

DeleteSymbol ds=new DeleteSymbol();

ds.deletesymbol(datasource);

**函数说明：**

1. datasource为需要处理的文件位置，格式为“大文件夹/数据文件”
2. **函数findkbySilhouette和函数findkbyInertia使用说明文档**

**函数作用：**根据K的值，对每个K进行聚类并评估，返回最合适的K值。可以根据这个K值找到聚类结果文件。

findkbySilhouette以轮廓系数进行评估，数据量在1万左右时可用。

findkbyInertia以inertia进行评估，任意数据量可用。

**输入：**存放K值的文件、CSR矩阵文件、生成聚类结果的文件

**返回：**最合适的K值

**函数使用：**

String Ksource="resource/k.txt";

String CSRPath = "resource/2月test1Wcsr矩阵";

String clusterResult="resource/cluster";

FindK f = new FindK();

int K=f.findkbySilhouette(Ksource, CSRPath,clusterResult);

int K=f.findkbyInertia(Ksource, CSRPath,clusterResult);

**函数说明：**

1. Ksource为存放K值的文件位置，格式为“大文件夹/存放K值的文件”。文件中每个K 值占据一行。
2. CSRPath为存放CSR矩阵的文件位置，格式为“大文件夹/CSR矩阵文件”。
3. clusterResult为根据K值聚类的结果文件位置，格式为“大文件夹/聚类结果文件”。