**IIC修正算法小文档**

# 一、IIC修正算法设计

## 1、修正条件

（1）提供cit文件的文件路径

（2）提供IIC文件的文件路径

（3）提供idf（即里程索引）文件的文件路径

## 2、设计类

### 2.1实体类

**长短链索引数据类** 【IndexSta】

该类是与数据库中IndexSta表对应的长短链索引数据类

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 数据类型 | 说明 |
| 1 | iID | int | 长短链索引id |
| 2 | iIndexID | int | 记录ID |
| 3 | lStartPoint | Long | 长短链对应的起始文件指针 |
| 4 | lStartMeter | String | 长短链对应的起始公里标 |
| 5 | lEndPoint | Long | 长短链对应的终止文件指针 |
| 6 | LEndMeter | String | 长短链对应的终止公里标 |
| 7 | lContainsPoint | Long | 长短链所包含的采样点数 |
| 8 | lContainsMeter | String | 长短链所包含的公里数（单位为公里） |
| 9 | sType | String | 长短链类别 |

**偏差类型类**【ExceptionType】

该类记录偏差的类型。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 数据类型 | 说明 |
| 1 | EXCEPTIONEN | String | 偏差类型英文名称 |
| 2 | EXCEPTIONCN | String | 偏差类型中文名称 |

**偏差类** 【Defects】

该类对应的是数据库中的fix\_defects表，另外该类中中还包含一个获取公里标GetMeter()的方法，单位为厘米

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 数据类型 | 说明 |
| 1 | iRecordNumber | Int | 记录号 |
| 2 | iMaxpost | Int | 单位为公里 |
| 3 | dMaxminor | double | 单位为米 |
| 4 | bFix | bool | 默认为false |

### 2.2操作类

该类主要是对iic文件进行修正，需要提前获取到cit文件的文件路径、iic文件的文件路径，具体操作步骤如下

（1）获取idf里程索引文件中的索引数据信息，定义成List<IndexSta> listIC，即IndexSta表中的数据。

（2）获取iic文件中的偏差数据，定义成List<Defects> listDC，即fix\_defects表中的数据，获取的数据需要符合一定的条件，条件如下：

select RecordNumber,maxpost,maxminor from fix\_defects where maxval2 is null or maxval2<>-200

（3）调用cit操作类，参考《cit操作API文档》获取公里标信息，定义List<Milestone> listMilestone

（4）对偏差数据集合listDC和里程标数据集合listMilestone进行嵌套循环匹配相同里程标（这里注意里程标单位，偏差数据中是厘米）；然后根据该公里标所在的文件流位置，在索引数据listIC中找到索引的里程标；对偏差数据集合listDC中的数据进行修改。这样形成一个修改后的偏差数据集合listDC。

（5）对修改后的偏差数据存储到iic文件中，即对iic文件进行修改，是一个更新过的iic文件。