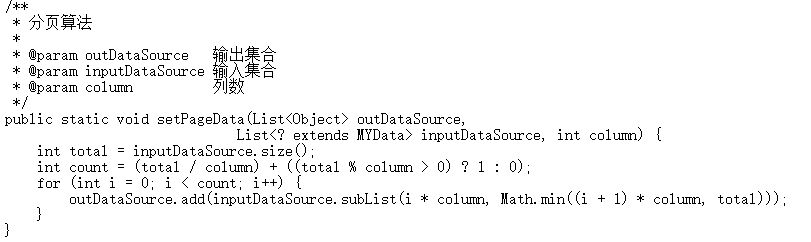
Android算法篇

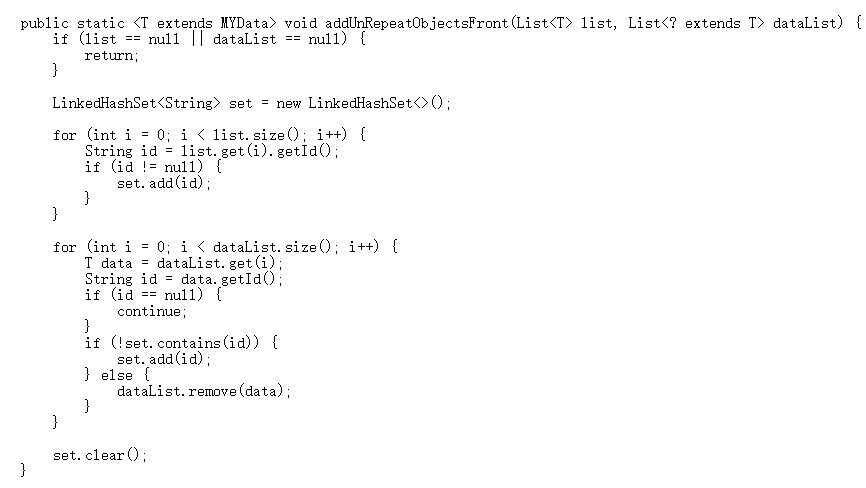
工程师 李洪江

一 集合分页算法

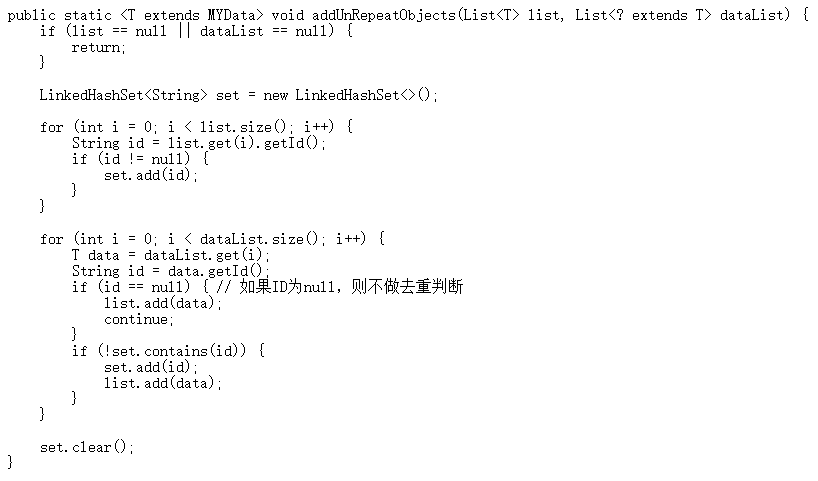


二 列表加载去重算法

【1】前加载去重复数据

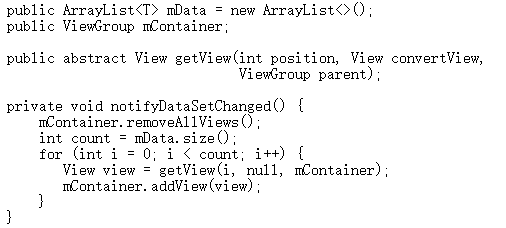


【2】后加载去重复数据

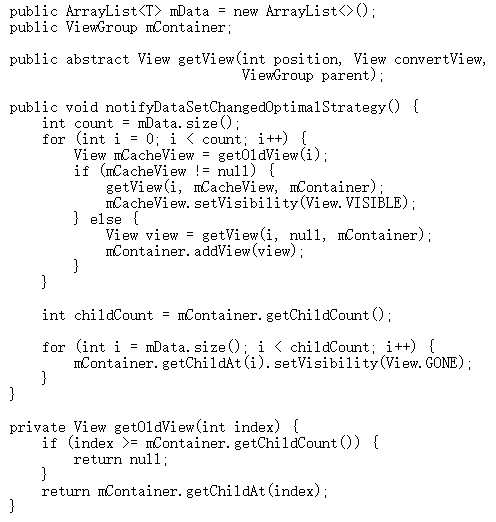


三 容器复用算法

【1】不复用写法-性能不好



相对较好的一种写法，简易版适配器写法



四 递归算法

【1】递归上传文件

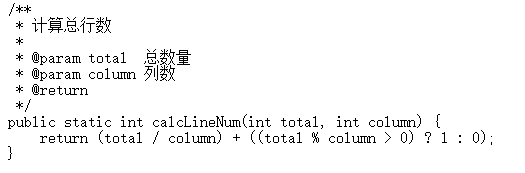
【2】递归遍历文件夹

【3】递归查找View

五 奇偶算法

【1】在列表控制线显示时候。

六 总行数



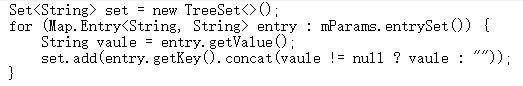
七 equals算法

【1】对象String时候，直接判断是否相等

【2】判断List集合是否包含对象，一种是遍历集合处理，第二种是重写equals方法，使用集合自带的contains方法，快速判断。

【3】判断HashSet做单选，多选处理，重写hash方法和equals方法达到去重效果

八 字典排序算法-tree数据结构使用

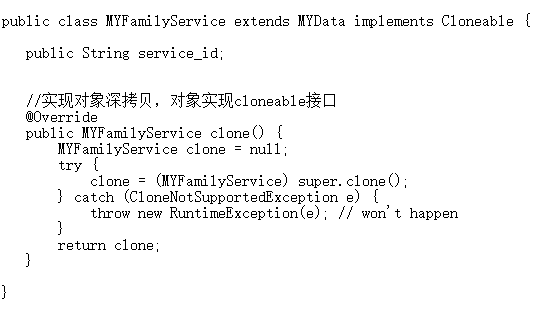


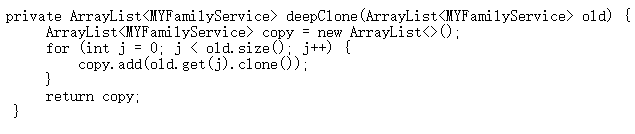
九 在写添加页面和编辑页面信息时候，复用界面写法。

【1】一种 使用状态字段来区分。

【2】一种 使用对象来区分。新数据和旧数据比较是否改变。

【3】涉及到深拷贝和浅拷贝知识。集合的add和addAll属于浅拷贝，修改对象属性值都会引起原来集合对象改变。





十 列表item拆分算法

【1】利用可见性拆分item项目。

【2】拆分N个小Item。

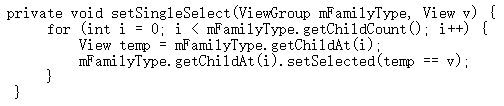
十一 单选和多选算法

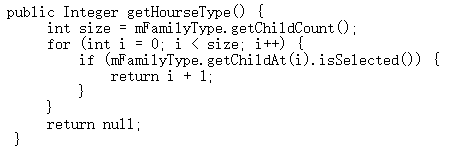
【1】不复用列表：

单选可采用保存View性能最好。

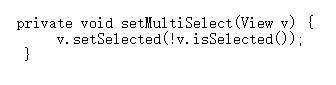
单选也可记录位置效率较好。

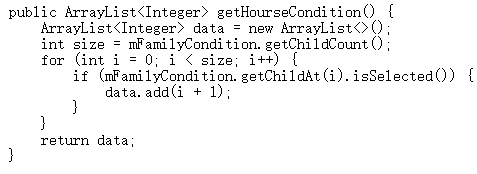
单选可采用遍历View性能一般。





多选可使用自身选中标记处理，最后遍历查看状态。





【2】复用View，不能使用本身记忆。

单选采用position记忆法，需要全部刷新。

单选采用HashSet方法记忆，不需要全部刷新。

多选采用数据本身记忆，不需要全部刷新。

【3】分页单选和复选

十二 图片压缩算法

【1】比例压缩

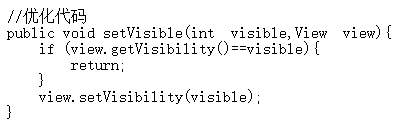
【2】质量压缩

十三View可见性算法

不建议这样写

IMG_256

建议这样写



十四 不同状态不同背景写法

【1】不建议代码设置不同背景写法

【2】建议使用选择器，利用enable，check，selected

【3】所有组件都有selected和enable状态。

【4】只有check类型控件才有check组件。

1. Arrays和Collections使用。

Arrays常用方法：sort排序，asList转化集合，binarySearch二分查找数据，fill填充方法，equals比较数组等。

Collections常用方法：sort排序。