# 资料整理

1. java控制多线程同时写一个文件  
   <https://blog.csdn.net/gxy3509394/article/details/7435993>

<https://www.cnblogs.com/xhy-shine/p/10725458.html>

try {

File file = new File("D:/test.txt");

//给该文件加锁

RandomAccessFile fis = new RandomAccessFile(file, "rw");

FileChannel fcin = fis.getChannel();

FileLock flin = null;

while (true) {

try {

flin = fcin.tryLock();

break;

} catch (Exception e) {

System.out.println("有其他线程正在操作该文件 ");

sleep(1000);

}

}

//读写操作

flin.release();

fcin.close();

fis.close();

fis = null;

}catch(Exception e)

{

e.printStackTrace();

}

}

## **Java基础数据类型和引用类型**

### 基本数据类型

* byte:java中最小的数据类型，内存中占8位(bit)，即一个字节取值范围-127~127,默认值0
* short：短整型，在内存中占16位，即2个字节，取值范围-32768~32717，默认值0
* int：整型，用于存储整数，在内存中占32位，即4个字节，取值范围-2147483648~21474783647，默认值0
* long：长整型，在内存中占64位，即8个字节-2^63 至 2^63-1,默认值0
* float:浮点型，在内存中占32位，即4个字节，用于存储带小数点的数字（与double的区别在于float类型有效小数点只有6~7位）默认值0
* double：双精度浮点型，用于存储带有小数点的数字，在内存中占64位，即8个字节，默认值0
* char：字符型，用于存储单个字符，占16位即2个字段 取值范围0~65535，默认值为空，char类型变量是用来存储Unicode编码的字符的，可以存储一个汉字，原因是Unicode编码字符集中包含了汉字。unicode编码占用两个字节，所以，char类型的变量也是占用两个字节。
* boolean:布尔类型，占1个字节，用于判断真或假（仅有两个值，true、false）,默认值false
* 引用类型数据

## 清单

Hook

反射

Volite

List map 安全删除

Hasmap原理

Java动态代理

Android存储路径 以及权限问题

## Inflater参数

inflater.inflate(R.layout.example\_fragment, container, false);

第一個參數：表示要被添加的view

第二个参数：不为null表示inflate布局的根节点的layout\_width和layout\_height属性将会失效

第三个参数：false：表示不会自动add，需要手动add

## AndroidQ适配查询uri

public static Uri getImageUri**(**Context context**,** String path**)** **{**

String name **=** getFileName**(**path**);**

ContentResolver resolver **=** context**.**getContentResolver**();**

Cursor cursor **=** resolver**.**query**(**Files**.**getContentUri**(**"external"**),** **(**String**[])null,** "\_display\_name=?"**,** **new** String**[]{**name**},** **(**String**)null);**

**if** **(**cursor **!=** **null** **&&** cursor**.**moveToFirst**())** **{**

**do** **{**

String id **=** cursor**.**getString**(**cursor**.**getColumnIndexOrThrow**(**"\_id"**));**

String data **=** cursor**.**getString**(**cursor**.**getColumnIndexOrThrow**(**"\_data"**));**

Uri uri **=** Uri**.**withAppendedPath**(**Files**.**getContentUri**(**"external"**),** id**);**

**if** **(**TextUtils**.**equals**(**data**,** path**))** **{**

**return** uri**;**

**}**

**}** **while(**cursor**.**moveToNext**());**

**}**

**return** **null;**

**}**