### 源码分析相关面试题

* [Volley源码分析](http://www.jianshu.com/p/ec3dc92df581)
* [注解框架实现原理](http://www.jianshu.com/p/20da6d6389e1)
* [okhttp3.0源码分析](http://www.jianshu.com/p/9ed2c2f2a52c)
* [onSaveInstanceState源码分析](http://www.jianshu.com/p/cbf9c3557d64)
* [静默安装和源码编译](http://www.jianshu.com/p/2211a5b3c37f)

### Activity相关面试题

* [保存Activity的状态](http://www.jianshu.com/p/cbf9c3557d64)

### 与XMPP相关面试题

* [XMPP协议优缺点](http://www.jianshu.com/p/2c04ac3c526a)
* [极光消息推送原理](http://www.jianshu.com/p/d88dc66908cf)

### 与性能优化相关面试题

* [内存泄漏和内存溢出区别](http://www.jianshu.com/p/5dd645b05c76)
* [UI优化和线程池实现原理](http://www.jianshu.com/p/c22398f8587f)
* [代码优化](http://www.jianshu.com/p/ebd41eab90df)
* [内存性能分析](http://www.jianshu.com/p/2665c31b9c2f)
* [内存泄漏检测](http://www.jianshu.com/p/1514c7804a06)
* [App启动优化](http://www.jianshu.com/p/f0f73fefdd43)
* [与IPC机制相关面试题](http://www.jianshu.com/p/de4793a4c2d0)

### 与登录相关面试题

* [oauth认证协议原理](http://www.jianshu.com/p/2a6ecbf8d49d)
* [token产生的意义](http://www.jianshu.com/p/9b7ce2d6c195)
* [微信扫一扫实现原理](http://www.jianshu.com/p/a9d1f21bd5e0)

### 与开发相关面试题

* [迭代开发的时候如何向前兼容新旧接口](http://www.jianshu.com/p/cbecadec98de)
* [手把手教你如何解决as jar包冲突](http://www.jianshu.com/p/30fdc391289c)
* [context的原理分析](http://www.jianshu.com/p/2706c13a1769)
* [解决ViewPager.setCurrentItem中间很多页面切换方案](http://www.jianshu.com/p/38ab6d856b56)

### 与人事相关面试题

* [人事面试宝典](http://www.jianshu.com/p/d61b553ff8c9)

### 本文配套视频

* [String字符串优化配套视频](https://v.qq.com/x/page/k0393ataw3l.html)
* [其他优化配套视频](https://v.qq.com/x/page/j0393gm2p7j.html)

## String字符串优化

最常见的例子就是当你要频繁操作一个字符串时，使用StringBuffer代替String。 还比如：使用int数组而不是Integer数组。 避免创建短命的临时对象，减少对象的创建就能减少垃圾收集，进而减少对用户体验的影响。

## ListView优化

1. Item布局，层级越少越好，使用hierarchyview工具查看优化。
2. 复用convertView
3. 使用ViewHolder
4. item中有图片时，异步加载
5. 快速滑动时，不加载图片
6. item中有图片时，应对图片进行适当压缩
7. 实现数据的分页加载

## 减少不必要的全局变量

尽量避免static成员变量引用资源耗费过多的实例，比如Context。 因为Context的引用超过它本身的生命周期，会导致Context泄漏。所以尽量使用Application这种Context类型。 你可以通过调用Context.getApplicationContext()或 Activity.getApplication()轻松得到Application对象。

## Cursor（游标）回收

Cursor是Android查询数据后得到的一个管理数据集合的类，在使用结束以后。应该保证Cursor占用的内存被及时的释放掉，而不是等待GC来处理。并且Android明显是倾向于编程者手动的将Cursor close掉，因为在源代码中我们发现，如果等到垃圾回收器来回收时，会给用户以错误提示。

## Receiver（接收器）回收

调用registerReceiver()后未调用unregisterReceiver(). 当我们Activity中使用了registerReceiver()方法注册了BroadcastReceiver，一定要在Activity的生命周期内调用unregisterReceiver()方法取消注册 也就是说registerReceiver()和unregisterReceiver()方法一定要成对出现，通常我们可以重写Activity的onDestory()方法，在onDestory里进行unregisterReceiver操作

## Stream/File（流/文件）回收

主要针对各种流，文件资源等等如： InputStream/OutputStream，SQLiteOpenHelper，SQLiteDatabase，Cursor，文件，I/O，Bitmap图片等操作等都应该记得显示关闭。

## 避免内部Getters/Setters

在Android中，虚方法调用的代价比直接字段访问高昂许多。通常根据面向对象语言的实践，在公共接口中使用Getters和Setters是有道理的，但在一个字段经常被访问的类中宜采用直接访问。 for循环 访问成员变量比访问本地变量慢得多，如下面一段代码：

for(int i =0; i < this.mCount; i++) {}

永远不要在for的第二个条件中调用任何方法，如下面一段代码：

for(int i =0; i < this.getCount(); i++) {}

对上面两个例子最好改为：

int count = this.mCount; / int count = this.getCount();   
for(int i =0; i < count; i++) {}

* 欢迎关注微信公众号，长期推荐技术文章和技术视频

微信公众号名称：Android干货程序员

img

img