# 2016-7-21

## 解决Eclipse中文乱码的四种方式

使用Eclipse编辑文件经常出现中文乱码或者文件中有中文不能保存的问题，Eclipse提供了灵活的设置文件编码格式的选项，我们可以通过设置编码 格式解决乱码问题。在Eclipse可以从几个层面设置编码格式：Workspace、Project、Content Type、File

本文以Eclipse 3.3（英文）为例加以说明：

1. 设置Workspace的编码格式：

Windows->Preferences... 打开"首选项"窗口，点击左侧导航树到General->Workspace，在右侧视图中找到“Text file encoding”选项设置，一种是默认（Default），另一种是从下拉列表中选择（Other）。Eclipse Workspace默认的编码方式是操作系统的编码格式，这跟操作系统的设置有关系；另外我们可以选择Other单选按钮，然后从按钮右侧的下拉列表中选 择需要的编码格式（GBK、ISO-8859-1、UTF-16、UFT-16 etc.）。

2. 设置Project的编码格式：

在 Workspace中新建的项目默认继承Workspace的编码设置，我们也可以单独更改某个项目的编码格式。右键点击工程，选择 Properties，打开项目属性设置窗口，左侧导航树选择Resource，在右侧视图中找到“Text file encoding”，两种设置项目的编码格式，默认选中的是“Inherited from container (XXX)”（注：XXX为Workspace设置的编码），我们也可以选择其他的编码格式，设置方式同Workspace。

3. 设置Content Type的编码格式：

有 时我们想使整个Workspace某种类型的文件保持同一种编码格式，这就需要用到Content Type设置来达到目的，具体方式如下：Windows->Preferences...打开"首选项"窗口，左侧导航树选择 General->Content Types，在右侧视图中选择Text->Java Source File，在最下侧有Default encoding输入框，手动输入编码格式，点击Update按钮使设置生效（切记啊！）。有两点需要注意一下：

a. 这个设置使Workspace所有项目下的相同类型的文件有相同的编码格式，改变项目的编码设置不影响项目中文件类型的编码设置；

b. Default encoding是手动输入的，输入的编码名称要准确，否则在打开此类型文件时会显示Unsupported Character Encoding。

4. 设置File的编码格式：

我 们还可以单独设置某个文件的编码格式，一种是通过在文件中设置编码格（如：'charset=UTF-8'）,还可以通过文件属性设置。右键点击某一文 件，选择Properties，打开文件属性设置对话框，右侧导航树选择Resource，在右侧视图中通过“Text file encoding”选择设置文件的编码格式。

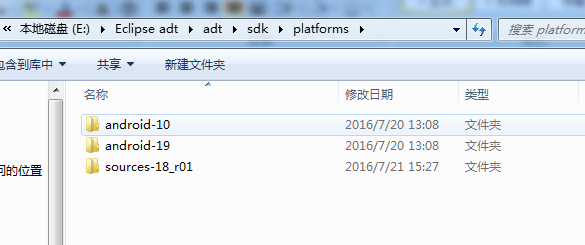
(http://blog.csdn.net/yangqicong11/article/details/10491389)

## 开发中关于编码的问题

在IDE中，整个IDE有个编码格式，IDE中的各个工程也可以有自己的编码格式，工程中不同的文件又可以有各自的编码格式，有时我们会发现，同一工程下的不同文件里，有的文件中没有乱码，而有的文件却乱码了，大多是因为他们各自的编码方式不同。

为了避免这种问题，最好同一个工程都使用一种编码，也可以统一一下整个IDE编码方式，具体的编码方式，最好采用的“utf-8”。

## 查看源码时遇到的问题

今天在看android源码时候，发现没有关联上，于是一番操作后关联上相关的文件（文件夹），但是居然不行，试了很多次，无论是用文件还是用文件夹，都没找到原因。后来发现是关联的文件夹里面没有对应的东西（.java文件），于是重新下了一个sources-18-r01，如下  


终于可以了。回想一下，如果开始直接在上面两个包中，搜索后缀名为.java的文件，发现没有相应的文件，就可以直接新下载一个，不用浪费时间在根本没有的东西上了（本来就没有，还以为是操作出现了瑕疵）。

2016-7-22

## 静态语句块

static{}(即static块)，会在类被加载的时候执行且仅会被执行一次，一般用来初始化静态变量和调用静态方法, 因为在虚拟机的生命周期中一个类只被加载一次；又因为static{}是伴随类加载执行的，所以，不管你new多少次对象实例，static{}都只执行一次。

## Android 开发习惯

（来自网络：http://www.zhihu.com/question/27227425）

作者：漫步  
链接：http://www.zhihu.com/question/27227425/answer/35973793  
来源：知乎  
著作权归作者所有，转载请联系作者获得授权。

受一个朋友邀请来回答这问题，以上各位都回答得很不错，不过对于一个Android开发新手来说，能有更详细的答案那可能会更有帮助。以下内容是来自eoe的一个网友的回答，但现在已经找不回那个帖子，文字已经珍藏起来。感谢原著作者的。

一、Android编码规范  
1.java代码中不出现中文，最多注释中可以出现中文；  
2.局部变量命名、静态成员变量命名：只能包含字母，单词首字母出第一个都为大写，其他字母都为小写；  
3.常量命名：只能包含字母和\_，字母全部大写，单词之间用\_隔开；  
4.layout中的id命名：命名模式为：view缩写\_模块名称\_view的逻辑名称  
view的缩写详情如下  
LinearLayout：ll  
RelativeLayout:rl  
TextView:tv  
ImageView:iv  
ImageButton:ib  
Button:btn  
5.activity中的view变量命名  
命名模式为：view缩写+逻辑名称  
建议：如果layout文件很复杂，建议将layout分成多个模块，每个模块定义一个moduleViewHolder，其成员变量包含所属view  
6.strings.xml中的id命名：

命名模式：activity名称\_功能模块名称\_逻辑名称 activity名称\_逻辑名称 common\_逻辑名称  
strings.xml中，使用activity名称注释，将文件内容区分开来  
7.drawable中的图片命名  
命名模式：activity名称\_逻辑名称/common\_逻辑名称  
7.styles.xml：将layout中不断重现的style提炼出通用的style通用组件，放到styles.xml中；  
8.使用layer-list和selector  
9.图片尽量分拆成多个可重用的图片  
10.服务端可以实现的，就不要放在客户端  
11.引用第三方库要慎重，避免应用大容量的第三方库，导致客户端包非常大  
12.处理应用全局异常和错误，将错误以邮件的形式发送给服务端  
13.图片的.9处理  
14.使用静态变量方式实现界面间共享要慎重  
15.Log(系统名称 模块名称 接口名称，详细描述)  
16.单元测试（逻辑测试、界面测试）  
17.不要重用父类的handler，对应一个类的handler也不应该让其子类用到，否则会导致message.what冲突  
18.activity中在一个View.OnClickListener中处理所有的逻辑  
19.strings.xml中使用%1$s实现字符串的通配  
20.如果多个Activity中包含共同的UI处理，那么可以提炼一个CommonActivity，把通用部分叫由它来处理，其他activity只要继承它即可  
21.使用button+activitgroup实现tab效果时，使用Button.setSelected(true)，确保按钮处于选择状态，并使activitygroup的当前activity与该button对应  
22.如果所开发的为通用组件，为避免冲突，将drawable/layout/menu/values目录下的文件名增加前缀  
23.数据一定要效验，例如  
字符型转数字型，如果转换失败一定要有缺省值；  
服务端响应数据是否有效判断；

二、Android性能优化  
1.http用gzip压缩，设置连接超时时间和响应超时时间  
http请求按照业务需求，分为是否可以缓存和不可缓存，那么在无网络的环境中，仍然通过缓存的httpresponse浏览部分数据，实现离线阅读。  
2.listview 性能优化  
1).复用convertView  
在getItemView中，判断convertView是否为空，如果不为空，可复用。如果couvertview中的view需要添加listerner，代码一定要在if(convertView==null){}之外。  
2).异步加载图片  
item中如果包含有webimage，那么最好异步加载  
3).快速滑动时不显示图片  
当快速滑动列表时（SCROLL\_STATE\_FLING），item中的图片或获取需要消耗资源的view，可以不显示出来；而处于其他两种状态（SCROLL\_STATE\_IDLE 和SCROLL\_STATE\_TOUCH\_SCROLL），则将那些view显示出来  
3.使用线程池，分为核心线程池和普通线程池，下载图片等耗时任务放置在普通线程池，避免耗时任务阻塞线程池后，导致所有异步任务都必须等待  
4.异步任务，分为核心任务和普通任务，只有核心任务中出现的系统级错误才会报错，异步任务的ui操作需要判断原activity是否处于激活状态  
5.尽量避免static成员变量引用资源耗费过多的实例,比如Context  
6.使用WeakReference代替强引用，弱引用可以让您保持对对象的引用，同时允许GC在必要时释放对象，回收内存。对于那些创建便宜但耗费大量内存的对象，即希望保持该对象，又要在应用程序需要时使用，同时希望GC必要时回收时，可以考虑使用弱引用。  
7.超级大胖子Bitmap  
及时的销毁(Activity的onDestroy时，将bitmap回收)  
设置一定的采样率  
巧妙的运用软引用  
drawable对应resid的资源，bitmap对应其他资源8.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  
9.线程也是造成内存泄露的一个重要的源头。线程产生内存泄露的主要原因在于线程生命周期的不可控  
10.如果ImageView的图片是来自网络，进行异步加载  
11.应用开发中自定义View的时候，交互部分，千万不要写成线程不断刷新界面显示，而是根据TouchListener事件主动触发界面的更新

三、Android UI优化  
1.layout组件化，尽量使用merge及include复用  
2.使用styles，复用样式定义  
3.软键盘的弹出控制，不要让其覆盖输入框  
4.数字、字母和汉字混排占位问题：将数字和字母全角化。由于现在大多数情况下我们的输入都是半角，所以 字母和数字的占位无法确定，但是一旦全角化之后，数字、字母的占位就和一个汉字的占位相同了，这样就可以避免由于占位导致的排版问题。  
5.英文文档排版：textview自动换行时要保持单词的完整性，解决方案是计算字符串长度，然后手动设定每一行显示多少个字母并加上‘n‘  
6.复杂布局使用RelativeLayout  
7.自适应屏幕，使用dp替代pix  
8.使用android:layout\_weight或者TableLayout制作等分布局  
9.使用animation-list制作动画效果

四、其他的一些Android开发建议

1.跟上时代的步伐，把Eclipse换成Android Studio，把SVN换成Git，这当然要适合项目开发属性的需要，Git学习中文网站：[https://git-scm.com/book/zh/v2](//link.zhihu.com/?target=https://git-scm.com/book/zh/v2" \t "_blank)

2.勤做总结，推荐使用印象笔记，把一些懂的经验总结起来，把还不懂的文章挂里面，有时间就慢慢消化；

3.定期code review，不断迭代，你总会发现一些不合理的代码，或者需要优化的地方。

4.关注一些知名的技术大V或网站，里面许多东西值得你去消化，推荐：[Android 开源项目集合](//link.zhihu.com/?target=http://www.codekk.com/" \t "_blank)，[http://tech.meituan.com/](//link.zhihu.com/?target=http://tech.meituan.com/" \t "_blank)，[stormzhang](https://www.zhihu.com/people/stormzhang)，但总归来说，去Android官网或者参考一份Java API文档虽说枯燥，但熟悉之后，你会有更大的进步。

5.如想更深入了解可阅读珍藏许久的文章：[Android应用程序开发以及背后的设计思想深度剖析](//link.zhihu.com/?target=http://www.uml.org.cn/mobiledev/201211063.asp%231" \t "_blank)

6.如果你公司没有强大的测试团队，发布应用前最好把应用放到测试平台去测测，比如云测之类的；

7.取应用包名的时候切忌取太容易重复的，如果同款手机已经有该包名，那么会因为签名不同而导致安装不上，这也怪中国安卓市场太多，无法像Google Play那样进行包名审核。

## 如何看懂android源码

1.首先看AndroidManifest.xml告诉你主activity service receiver contentprovider,

2.根据生命周期去看主activity的操作，对其它类的初始化 调用等 主要是oncreate onstart 不要太关注细节

3.再去看看其它类的实现，看看是否有继承原生的类 实现原生的接口，这个时候就是百度 官方文档 或者源码 查查这个类干什么的 有什么方法

4.看个大概理一下代码的结构，分析一下 比如说输入法 主要有键盘容器 按键区 候选区 键盘 按键

5.细节上加Log打印信息 看看自己的理解 修改bug

6.剩下的事情就是你可以考虑一下自己能怎么实现，哪个部分可以优化

如果单论一个简单工具型app来看的话，剥除layout xml等布局文件，整个项目代码量应该在5w-10w档次，更何况Android每个界面独自为Activity，首先看AndroidManifest.xml文件就可以对整个项目有大概了解有哪些Activity View Service，然后由点带面·~~

按照楼主描述就是新手刚入门，如果不是公司项目的话，建议仿照你要看的Android项目来自己实现一个，这样时间成本虽然更大，但是对项目掌握 学习更好。

如果是公司项目，那就加班去加断点看代码啊，还用问吗

——————————————————————————————————————

比如简单地实现一些material design，了解一些简单地开源项目，简单了解一些常用库的功能和用法。然后github上找别人比较简单地项目，比如简单的天气，新闻，笔记类项目，大神的项目就先别想，找些同样初学的人做的项目，我看的是第一个仿知乎日报的项目，叫做Kuaihu，想看的github上搜索去吧，这之前还看了下吴小龙的androidmvpsample，也在github有，画一个小时弄懂的才看kuaihu。kuaihu看了两天，结构还是比较清楚，注意各个类的继承和接口关系，然后看manifest找到主活动，然后从主活动看起，看它怎么加载界面，怎么加载数据。怎么把各种界面连接起来，其实无非就是一些点击事件，顺着这些点击事件，找下一个界面，难的应该是加载数据这一块，就是网络，图片，adapter，数据库的各种裙带关系。看着理解吧，不过话说回来，最终还是多花点时间多有点耐心，硬啃下来吧

# 2016-7-25

## Android System.out.println在控制台无输出

java中一般打印方法为System.out.println("");如果程序没问题，要打印的内容会在控制台显示。

但在Android平台，调用System.out.println("")函数，即使程序正确，也不会在控制台显示。不过可以在LogCat中查看。Android自身提供的打印方法是Log.v,引入android.util.Log 即可使用。查看打印的字符也不在console面板，而在Log面板上，可以选择ide右上角debug模式，查看Log面板。

2016-7-28

## 在debug时出现source not found的解决办法

在eclipse adt 中debug调试一个android项目时，总是提示source not found（进入了类ActivityThread.java这个类中，但是这个类找不到），网上找了很多原因，都没用，给出的解决方法大多如下：

方法a：

Debug 视图下

-》在调试的线程上 右键单击

-》选择Edit Source Lookup Path

-》选择Add

-》选择Project

-》选择正在调试的project

选择相应的Project 进行OK确定即可

方法b：

1.将工程目录下的.metadata\.plugins\下的 org.eclipse.debug.core和org.eclipse.debug.ui目 录删除

2.重启eclipse即可

方法c：

<http://stackoverflow.com/questions/1960158/eclipse-debugging-source-not-found>

方法d（最后解决问题的方法）：

<http://zhidao.baidu.com/question/1689479740656221868.html?fr=iks&word=activitythread.performlaunchactivity%28source+not+found%29&ie=gbk>

“提示source not found的时候，它会提示你是否查找的，你把放源代码的目录再告诉它一遍，它就能找到了，就是这么变态”

经过摸索，在方法a中，选择add之后，不要选择project

Debug 视图下

-》在调试的线程上 右键单击

-》选择Edit Source Lookup Path

-》选择Add

-》选择“File System Dicrionary”的选项（具体记不清了），然后选择相应的文件夹，把有源码的文件夹的地址放上就OK了，笔者的文件夹为E:\Eclipse adt\adt\sdk\platforms\android-19-，就是我们要看源码时配置的地址。这时候，我们就很困惑了，配源码时不是已经配置过一次了吗，为什么还要在配一次，原因正如一位网友所说，因为“它变态”！（Annoying for hours）

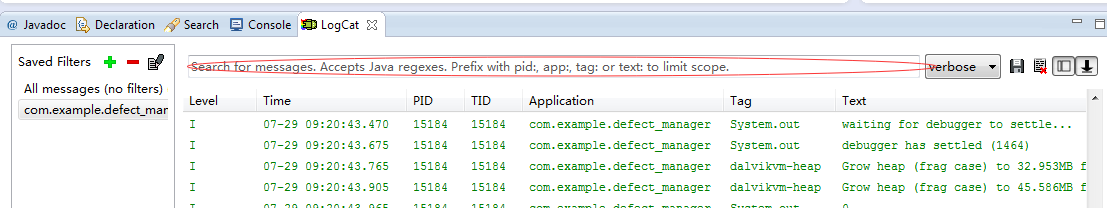
2016-7-29

## Eclipse连上真机logcat中出现很多无关信息

大部分都是手机中其他应用或者手机状态的信息，但很多时候无关信息太多会妨碍我们查看调试程序的信息，解决方法如下：

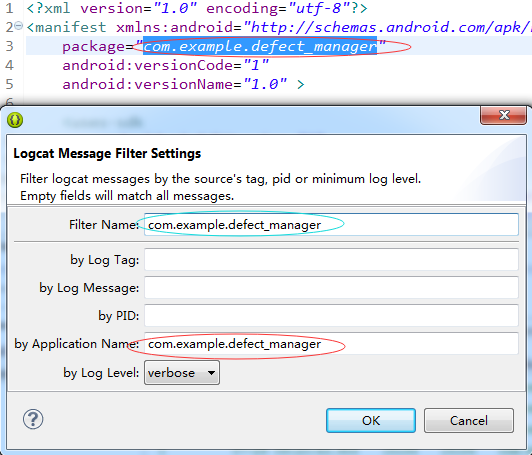
方法A.

logcat窗口上方有个搜索栏，可以用关键字搜索想要查看的相关信息，但是当信息太多时候，也是不好用（不推荐）；



方法B.

点击左侧绿色“+”按钮，出现：



其中，Filter Name（绿框）随便填，by Application Name（红框）中的信息填写manifest.xml文件中的package中的信息，这样，在logcat中就只会显示和你本项目有关的信息了。

## 连上真机后LogCat不显示日志的问题

有时候在Eclipse打开后，将真实的Android设备通过USB线连接到电脑，并不会在LogCat中看到有日志输入。

这可能是因为电脑中还安装了其它手机工具程序，占用了运行ADB（Android 调试桥）服务所需的端口，导致不能通过ADB连接到Android设备，进而导致没有日志输出。

可以用下面的步骤解决该问题：

1.关闭Eclipse；

2.打开操作系统命令行工具，使用cd命令切换到Android SDK的bin目录，输入adb nodaemon server，查看adb服务所使用的接口是否被占用，如果占用，给出的提示会显示占用的端口，一般是 5037；

3.继续在命令行工具中输入命令 netstat -ano | findstr 5037， 查看占用了该端口的进程信息，假设得到占用该端口的进程号为 xxx；

4.继续在命令行工具中输入命令 taskkill /F /PID xxx，杀掉这个进程；

5.重新启动Eclipse。

***Tips：*** Android调试桥（ADB）是一个客户端-服务器应用程序，可以通过它来连接到任何Android设备（真实设备或者虚拟机）。

## 对GIt中的私钥和公钥的解释

GIT服务器上存储的是公钥，你本地存储的是私钥，当你push本地代码库到远程代码库，服务器会要求你出示私钥，并且用你出示的私钥和它的公钥配对来完成认证。

由于使用的是**不对称加密**，所以公钥可以公开，只要保管好私钥就可以。  
路人甲只要没有私钥，所以无法push。这样才能保证开源项目的完整性，否则阿猫阿狗都去push代码，那项目不就乱套了？

如果路人甲想对你的项目做出贡献，那么以GITHUB为例，他会**发送一个pull request**给你，然后**由你**来审核他作出的改变，如果审核通过，那么你就可以**将他的pull request合并**到你工程的某一分支中。这里只以GITHUB为例，私有的GIT服务器可能没有类似功能，比如GITLAB就没有pull request

# 2016-8-1

## ListView的item中的TextView控件点击不灵敏

在item的布局文件里面的*TextView控件中*加上属性

android:textIsSelectable=*"true"*

同时，在item的*根部局文件中*加上属性

android:descendantFocusability=*"blocksDescendants"*

调试过程中，发现虽然控件点击灵敏了，但是发现

### 第一次点击TextView控件无效，之后的点击才有效

网上给的原因是：当前（第一次点击之前）TextView控件没有获取焦点，给其加上属性android:focusable=*"true"*

即可

## 点击按钮返回上个Activity

开发中遇到需要点击按钮返回上上个Activity的需求，这里记录下解决的方法：

A，B，C都是activity，顺序是A--->B--->C

先说一下怎么返回上个Activity：

从C返回B，C--->B：在C中（按钮的监听事件中）使用this.finish(),结束当前activity，就会直接到B中；

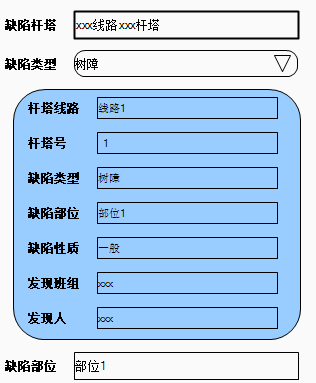
从C--->A，直接返回，不经过B:

当从B--->C时，在跳转代码写完的时候，在B中调用this.finish()，接着，在C中使用this.finish(),就不会经过B了，因为B已经finish（）了，从而达到C直接返回A的效果；

**Tips：**最好不要使用Intent跳转来达到返回上一个Activity的效果，因为它只是界面上过去了，但并不是返回，而是界面跳转【可以通过activity中的界面转换的效果（界面是左移还是右移）来观察他们的区别】。

## Spinner中不同的下拉选项有不同的布局

如图所示：



问题描述：

缺陷类型是可选择的（选项在xml文件中配置），当选择的是“树障”时，出现蓝色部分，如果是其他选项，则不出现；

解决办法：

将蓝色部分在整个布局文件中，用LinearLayout布局属性包裹起来，在activity中定义此LinearLayout，

private LinearLayout defect\_SZSJ; // 树障数据

defect\_SZSJ = (LinearLayout) findViewById(R.id.defect\_SZSJ);// 树障数据

接着定义一个Spinner，

private Spinner spinner;

spinner = (Spinner) findViewById(R.id.sp\_Defectclass);

然后调用Spinner控件的setOnItemSelectedListener（）方法**进行监听**，在此方法中

String value=spinner.getSelectedItem().toString();

将得到的值（value）进行判断，

if (value.equals("树障")) {

defect\_SZSJ.setVisibility(View.VISIBLE);

} else {

defect\_SZSJ.setVisibility(View.GONE);

}

总结一下：

1、LinearLayout类的setVisibility（\*\*）方法的使用；

2、不止控件类，布局类也是可以定义的，也有其相应的方法和属性。

## 对LayoutInflater的理解并与findViewById()和setContentView()比较

LayoutInflater()：

——是用来找res/layout/下的xml布局文件，并且实例化（实例化为View类对象）；

——对于一个没有被载入或者想要动态载入的界面，都需要使用LayoutInflater.inflate()来载入；

findViewById()：

——找xml布局文件下的具体widget控件(如Button、TextView等)；

——对于一个已经载入的界面，就可以使用Activiyt.findViewById()方法来获得其中的界面元素；

setContentView()：

—— 一旦调用, layout就会立刻显示UI；

setContentView()一旦调用, layout就会立刻显示UI；而inflate只会把Layout形成一个以view类实现成的对象，有需要时再用setContentView(view)显示出来。一般在activity中通过setContentView()将界面显示出来，但是如果在非activity中如何对控件布局设置操作了，这就需要LayoutInflater动态加载。

*LayoutInflater与setContentView()的字面理解（个人理解）：*

前者是“在平板中布局”，后者是“设置上下文视图”，因此：

就可以理解activity中一般使用setContentView（）了，因为activity是content的子类；

而非activity中一般使用LayoutInflater，因为LayoutInflater是在“平板上”布局，不是“activity”

（http://www.cnblogs.com/top5/archive/2012/05/04/2482328.html

<http://blog.csdn.net/lmj623565791/article/details/38171465）>

# 2016-8-8

## Android中EditText控件自动换行

在EditText控件中添加android:inputType=*"textMultiLine"*属性（没想到inputType中还可以设置自动换行）

有时还会遇到EditText中文英文长度控制，英文字母最多N个，而中文最多N/2个。这样的话直接用LengthFilter就无法实现了，但是我们可以通过自定义InputFilter实现：

（http://blog.csdn.net/songzhiyong1121/article/details/8758902）

## in的嵌套查询

in关键字用于where子句中用来判断查询的表达式是否在多个值的列表中。返回满足in列表中的满足条件的记录。

（http://www.cnblogs.com/kissdodog/archive/2013/06/03/3116284.html）

## 一个低级错误

今天在使用StringBuffer的时候，直接

StringBuffer chosedPeople;

没有new StringBuffer(),然后

chosedPeople.append(\*\*);

很显然的报了

NullPointerException，

以后要注意

## Android中什么时候需要为控件注册监听

Android中不需要注册监听就能获取控件中的（输入）内容，如

~~edt\_mission\_time.setOnClickListener(this);~~

也能通过edt\_mission\_time.getText().toString();

获得控件edt\_mission\_time中的内容。（因为监听是对行为的响应，并产生另外的行为），所以，当只需要获取信息而不需要作出相应的动作时，是不需要注册监听的

## 使用SQLite遇到的坑

想在SQLite中将一个字段设为主键并自增，遇到一点问题：

字段为ID，数据类型为INT(11),当设置ID为“主键”时没有问题出现，但是当添加“自增”时，就提示需要将INT改为INTEGER。之后还是提示不行，于是抱着尝试的态度将INTERGER(11)改成INTEGER就可以了。（此处是使用SQLiteStudio，不知道使用命令行是不是一样的）

## Android中数据库的关闭问题

通常在一个Android Activity中有很多方法，可能在不同的方法中都会对数据库进行操作，那么db的关闭问题就不得不考虑，下面提供两个笔者的解决办法：

*问题由来：*  
在方法A中db.close(),在方法B中又调用了db的方法，会提示:

错误( java.lang.IllegalStateException: attempt to re-open an already-closed object)，

因为在用db之前需要将其打开;

（这里A方法在B方法之前调用，所以执行到B的时候db已经关闭了）

*方法1：*

在每次用到db的时候，都将其打开，用完后关闭；

（不推荐，消耗资源，性能差）

*方法2：*

了解活动的生命周期，以及活动中各方法间的调用（先后执行）关系，将db.close(),放到最后执行的方法中，就不需要每次没打开和关闭了；

（使用此方法，要掌握好activity的生命周期，以及活动中方法执行的先后顺序）

*两者比较：*

当activity中多个方法经常和数据库有交互的时候，方法2是更好的选择；

当整个activity跟数据库只有很少的交互的手，方法1似乎比方法2更好。

## SQLite中模糊查询的条件是变量

今天在用SQLite写模糊查询的时候，条件是个变量，费了些时间，如下：

*原来的语句：*

"select FZR,RWFL,GZFX,RWNR,ZT,RWSJ from task where CYRY like '%''+username+''%' and RWSJ = ?"

这里的username是个变量，但是尝试了几次，也换了不同的写法，在(java)代码中运行的时候都解析不出它的值（值应该是test1），还是把它当成了字符串，最后终于测试到了正确的写法：

*正确的写法：*

"select FZR,RWFL,GZFX,RWNR,ZT,RWSJ from task where CYRY like '%"+username+"%' and RWSJ = ?"

浪费了些时间，在此记下来，希望以后再遇到能很快解决。

*Tips（小技巧）：*

代码是在Eclipse中写的，其中的代码会有颜色的区分，可以注意到，原来错误的写法中，username的颜色是蓝色，和整个SQL语句中的颜色一致，可见是被当成了 字符串；在正确的写法中的username是黑色的，可见是被当成了 变量。（以前没注意到此技巧，如果之前注意到的话，这次能节约很多的调试时间）

## 在一个Activity中调用另一个xml布局文件

有时候，我们需要在一个Activity中调用另一个xml布局文件，即非本Activity所绑定的xml布局文件中的控件，这时候就不能直接findViewById，不然会报错指向空对象，这时就需要像下面这样做。

LayoutInflater factory = LayoutInflater.from(当前类.this);

View layout = factory.inflate(R.layout.你要获取的另一个XML, null);

TextView textview = (TextView) layout.findViewById(R.id.控件ID);

一行写完就是：

TextView textview = (TextView) LayoutInflater.from(当前类.this).inflate(R.layout.你要获取的另一个XML, null).findViewById(R.id.控件ID);

## Invalid layout of java.lang.String at value问题的解决

问题描述：

在Eclipse中建立android工程后在里面写一个java类，在类中写上main方法，然后运行这个类，提示上述错误

解决办法：

选中此类——>右击run as——>run Configuration——>在classpath选显卡中将android4.4.2 Remove 掉即可；

不足：

如果有其他类中仍然用了main方法运行，还需在那个类中重复此操作。

## Eclipse中的内存分析

<http://my.oschina.net/jielucky/blog/169793>

<http://www.cnblogs.com/java-class/archive/2013/04/18/3027884.html>

<http://chiyx.iteye.com/blog/1528782>

<http://my.oschina.net/biezhi/blog/286223>

<http://blog.csdn.net/rogerjava/article/details/23255277>

## 日期正则表达式demo

**public** **class** RegexUtil {

**static** **boolean** *flag* = **false**;

**public** **static** **final** String *date\_regex* = "^[0-9]{4}-(0?[0-9]|1[0-2])-(0?[1-9]|[12]?[0-9]|3[01])$";

**public** **static** **boolean** isRegexValidateTrue(String str) {

**try** {

Pattern pattern = Pattern.*compile*(*date\_regex*);

Matcher matcher = pattern.matcher(str);

*flag* = matcher.matches();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** *flag*;

}

}