**Cursor(光标)类**

在你理解和使用 Android Cursor 的时候你必须先知道关于 Cursor 的几件事情：

Cursor 是每行的集合。使用 moveToFirst() 定位第一行。你必须知道每一列的名称。  
你必须知道每一列的数据类型。Cursor 是一个随机的数据源。所有的数据都是通过下标取得。

关于 Cursor 的重要方法：close()关闭游标，释放资源。

copyStringToBuffer(int columnIndex, CharArrayBuffer buffer)在缓冲区中检索请求的列的文本，将将其存储。  
getColumnCount()返回所有列的总数  
getColumnIndex(String columnName)返回指定列的名称，如果不存在返回-1  
getColumnIndexOrThrow(String columnName)从零开始返回指定列名称，如果不存在将抛出IllegalArgumentException 异常。  
getColumnName(int columnIndex)从给定的索引返回列名  
getColumnNames()返回一个字符串数组的列名  
getCount()返回Cursor 中的行数  
moveToFirst()移动光标到第一行  
moveToLast()移动光标到最后一行  
moveToNext()移动光标到下一行  
moveToPosition(int position)移动光标到一个绝对的位置  
moveToPrevious()移动光标到上一行  
下面来看看一小段代码：

if (cur.moveToFirst() == false)  
{  
//为空的Cursor  
return;  
}

访问 Cursor 的下标获得其中的数据

int nameColumnIndex = cur.getColumnIndex(People.NAME);  
String name = cur.getString(nameColumnIndex);  
现在让我们看看如何循环 Cursor 取出我们需要的数据

while(cur.moveToNext())  
{  
//光标移动成功  
//把数据取出  
}

当cur.moveToNext() 为假时将跳出循环，即 Cursor 数据循环完毕。

如果你喜欢用 for 循环而不想用While 循环可以使用Google 提供的几下方法：

isBeforeFirst()  
返回游标是否指向之前第一行的位置  
isAfterLast()  
返回游标是否指向第最后一行的位置  
isClosed()  
如果返回 true 即表示该游戏标己关闭  
有了以上的方法，可以如此取出数据

for(cur.moveToFirst();!cur.isAfterLast();cur.moveToNext())  
{  
    int nameColumn = cur.getColumnIndex(People.NAME);  
    int phoneColumn = cur.getColumnIndex(People.NUMBER);  
    String name = cur.getString(nameColumn);  
    String phoneNumber = cur.getString(phoneColumn);  
}

Tip:在Android 查询数据是通过Cursor 类来实现的。当我们使用 SQLiteDatabase.query()方法时，就会得到Cursor对象， Cursor所指向的就是每一条数据。结合ADO.net 的知识可能好理解一点。

Cursor 位于 android.database.Cursor类，可见出它的设计是基于数据库服务产生的。

SurfaceView

View位于UI主线程，SurfaceView不是，可以用于被动更新

最近项目上想使用ActionBarActivity，但是只要我继承ActionBarActivity，程序启动时立马闪退，logcat也没报任何信息。

经过仔细的排查终于找到了原因所在：

由于项目中使用的Theme为[Android](http://lib.csdn.net/base/15" \o "Android知识库" \t "_blank):Theme.Light.NoTitleBar导致的错误，然后我又试了试其他的主题好像也不行，没办法 我有重新建了个工程，看看官方使用的是哪个Theme。最终，发现使用Theme.AppCompat.Light.NoActionBar时可以正常使用。

**Timer与TimerTask**

Timer是一种定时器工具，用来在一个后台线程计划执行指定任务。它可以计划执行一个任务一次或反复多次。TimerTask一个抽象类，它的子类代表一个可以被Timer计划的任务。

* 实现自定义的TimerTask的子类，run方法包含要执行的任务代码。
* 实例化Timer类，创建计时器后台线程。
* 实例化任务对象 (new TimerTask() ).
* 制定执行计划。这里用schedule方法，第一个参数是TimerTask对象，第二个参数表示开始执行前的延时时间（单位是milliseconds，这里定义了5000）。