MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy
讲师：高焕堂（台湾）

http://www.microoh.com

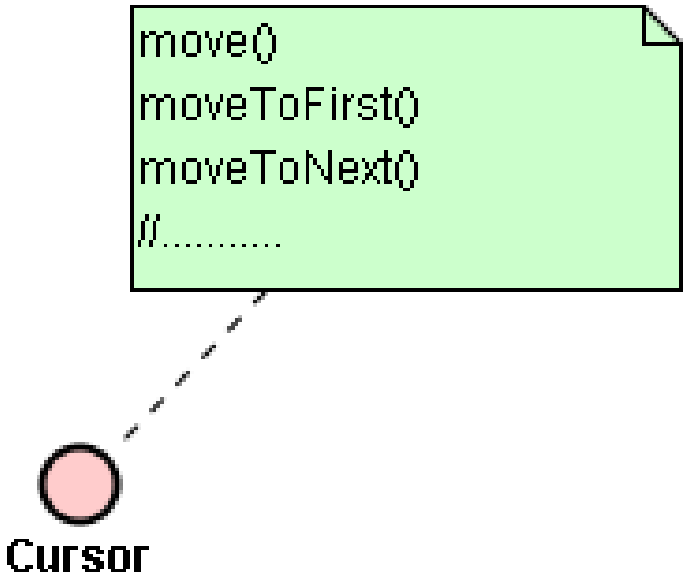
F06_b

观摩：ContentProvider架构與DB引擎移植方法(b)

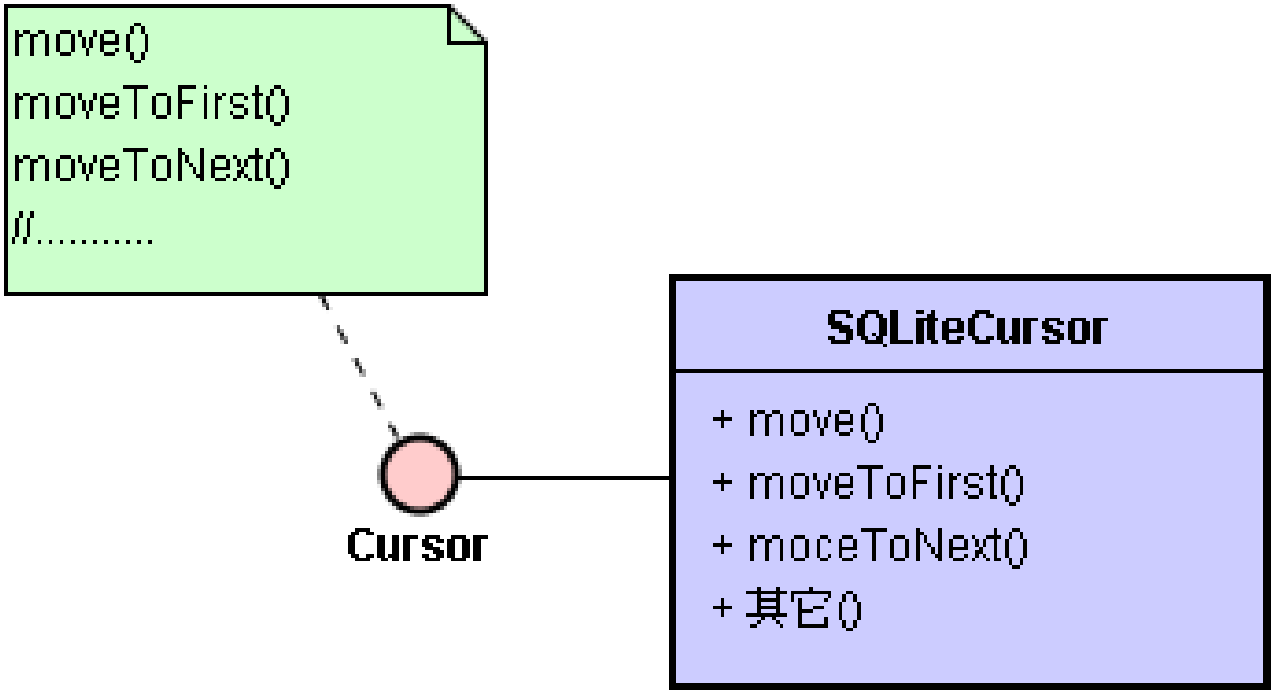
By 高煥堂

2、从Cursor接口谈起

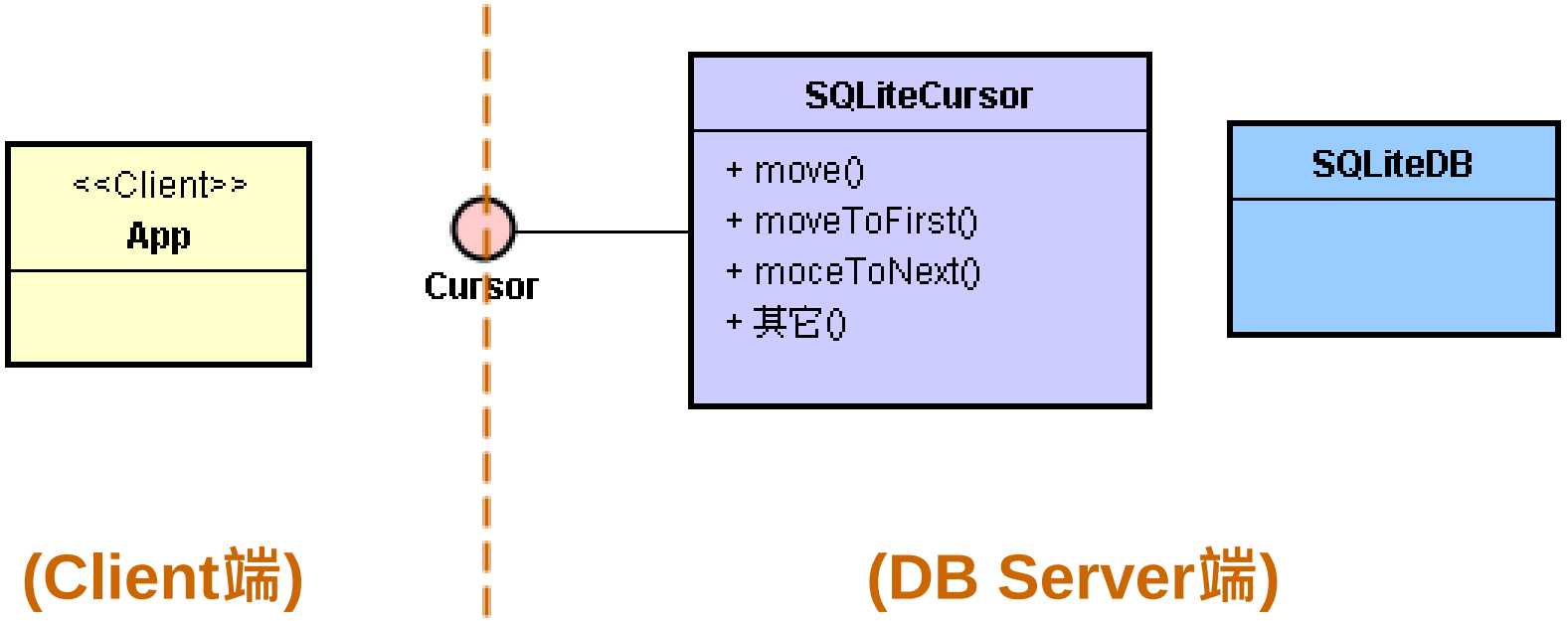
- 在Android里定义了一个Cursor接口，让App(如Activity或Service等)能透过此Cursor接口来浏览DB里的各笔数据或内容。



- 如果搭配SQLite(数据库)引擎的话，就可设计一个类来实现这Cursor接口。其实，Android里已经撰写了这个实现类，名叫：SQLiteCursor。


move()
moveToFirst()
moveToNext()
//.....
Cursor
SQLiteCursor
+ move()
+ moveToFirst()
+ moceToNext()
+ 其它()

- 于是，Cursor成为Client端浏览SQLite数据库内容的标准接口了。

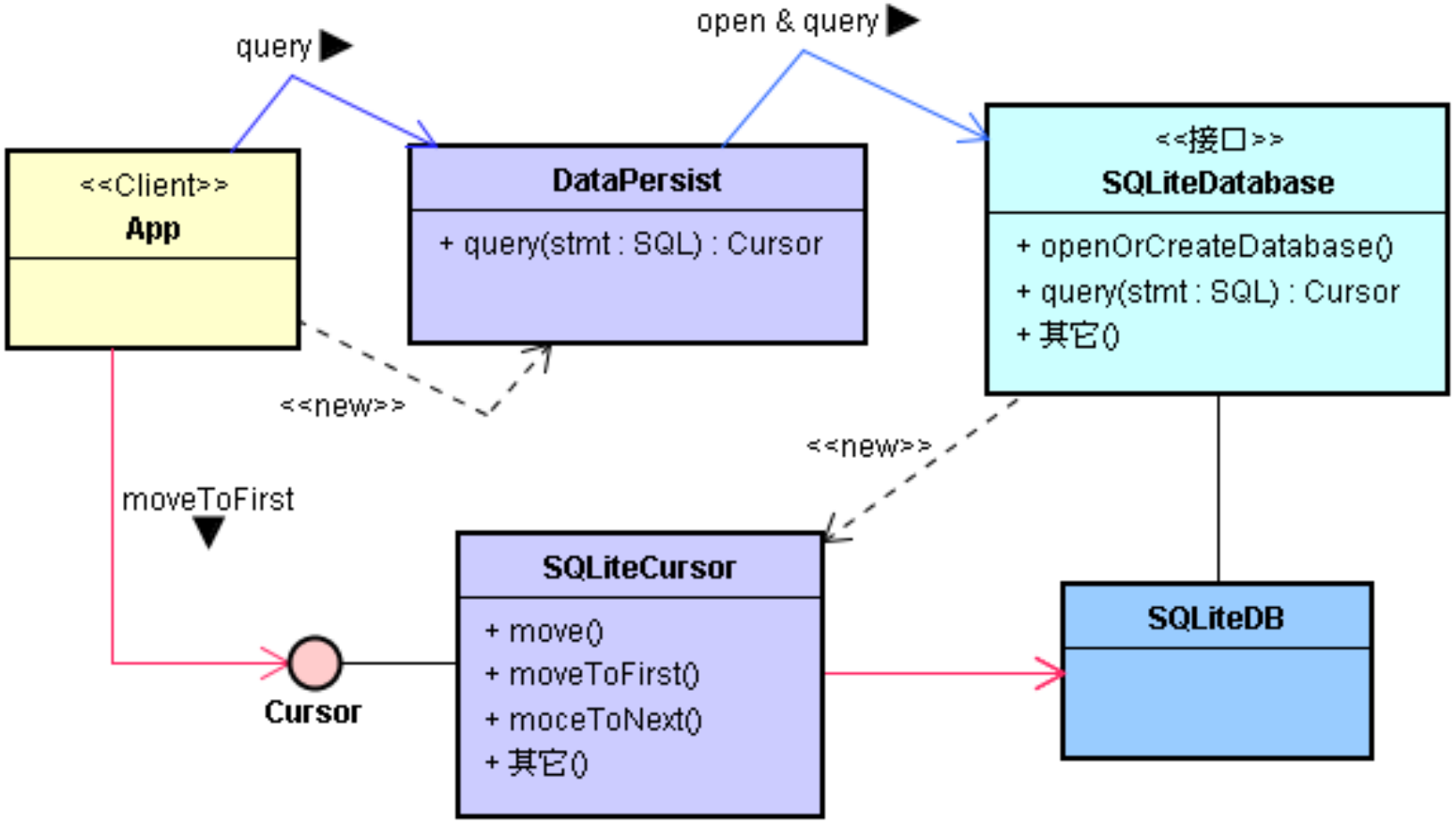


3、通用性接口Cursor的使用范例

范例架构

取得Cursor接口

- 撰写一个DataPersist类，来开启DB，并提供query()函数；在执行query()时就创建一个SQLiteCursor对象，将其Cursor接口回传给Client端。


query
open & query
<<Client>>
App
DataPersist
+ query(stmt : SQL) : Cursor
<<接口>>
SQLiteDatabase
+ openOrCreateDatabase()
+ query(stmt : SQL) : Cursor
+ 其它()
<<new>>
<<new>>
moveToFirst
SQLiteCursor
+ move()
+ moveToFirst()
+ moceToNext()
+ 其它()
SQLiteDB
Cursor

使用Cursor接口

- 接下来，只要调用Cursor接口的函数，就能浏览DB里的内容了。

范例代码

- 由于Android平台里已经有了SQLite引擎了，而且已经写好了SQLiteCursor实现类。因此在这范例里，我们只需要DataPersist类和Activity的子类，就行了。

DataPersist.java代码

```
/* ----- DataPersist.java ------*/
// ……
public class DataPersist {
    private static final String DATABASE_NAME = "StudDB";
    private static final String TABLE_NAME = "Student";
    private final int DB_MODE = Context.MODE_PRIVATE;
    private SQLiteDatabase db=null;
```

```
public DataPersist(Context ctx) {
  try {  db = ctx.openOrCreateDatabase(DATABASE_NAME,
                    DB_MODE, null);   }
  catch (Exception e) {  Log.e("ERROR", e.toString());   return;   }
  try {
         db.execSQL("drop table "+ TABLE_NAME); }
  catch (Exception e) {  Log.e("ERROR", e.toString());   }
  db.execSQL("CREATE TABLE " + TABLE_NAME + " ("  +
           "stud_no" + " TEXT," + "stud_name" + " TEXT" + ");");
  String sql_1 = "insert into "+ TABLE_NAME +
                 " (stud_no, stud_name) values('S101', 'Lily');";
 String sql_2 = "insert into " + TABLE_NAME +
                 " (stud_no, stud_name) values('S102', 'Linda');";
```

```
    String sql_3 = "insert into " + TABLE_NAME +
                    " (stud_no, stud_name) values('S103', 'Bruce');";
    try {  db.execSQL(sql_1);   db.execSQL(sql_2);
              db.execSQL(sql_3); }
    catch (SQLException e)
              {  Log.e("ERROR", e.toString());  return;  }
  }
   public Cursor query(String[] projection, String selection, String[]
                                        selectionArgs, String sortOrder) {
      Cursor cur = db.query(TABLE_NAME, projection, null, null,
                                        null, null, null);
      return cur;
   }
   public void close(){   db.close();   }
}
```

ac01.java代码

```
/* ----- ac01.java  ------*/
// …….
public class ac01 extends ListActivity {
    private static final String[] PROJECTION =
            new String[] { "stud_no", "stud_name" };

    @Override protected void onCreate(Bundle
                    savedInstanceState) {
        super.onCreate(savedInstanceState);

        DataPersist dp = new DataPersist(this);
        Cursor cur = dp.query(PROJECTION, null, null, null);
        ArrayList<Map<String, Object>> coll =
                new ArrayList<Map<String, Object>>();
        Map<String, Object> item;
        cur.moveToFirst();
```

```
        while(!cur.isAfterLast()) {
            item = new HashMap<String, Object>();
            item.put("c1", cur.getString(0) + ",  " + cur.getString(1));
            coll.add(item);
            cur.moveToNext();
           }
        dp.close();
        this.setListAdapter(new SimpleAdapter(this, coll,
              android.R.layout.simple_list_item_1, new String[] { "c1" },
                          new int[] {android.R.id.text1}));
    }
    @Override
    protected void onListItemClick(ListView l, View v,
                  int position, long id) {
            finish();
}}
```

- 指令：

  DataPersist dp = **new** DataPersist(**this**);

- 此时，诞生一个DataPersist对象；并开启了DB。然后执行指令：

  Cursor cur = dp.query(*PROJECTION*, **null**, **null**, **null**);

- 这查询出某些数据，创建Cursor对象，并回传之。

~ Continued ~