**一些说明**

这就是一个简单的课程表软件能手动添加课程，删除课程，其中还包括了自己写的一个计算简单数学表达式的计算器，可以计算带括号、加减乘除、幂运算(^)、及百分号的规范的数学表达式。

（此说明是对提交代码的说明。）

### MainActivity

课程表主要活动，主要是loadData()函数加载数据库中的课程信息saveDate()保存数据到数据库，还有界面的逻辑实现和菜单的实现

### DatabaseHelper

创建课程的数据库id为主键，course表中有string (course\_name ,teacher ,class\_room )

Int(day ,class\_start,class\_end)

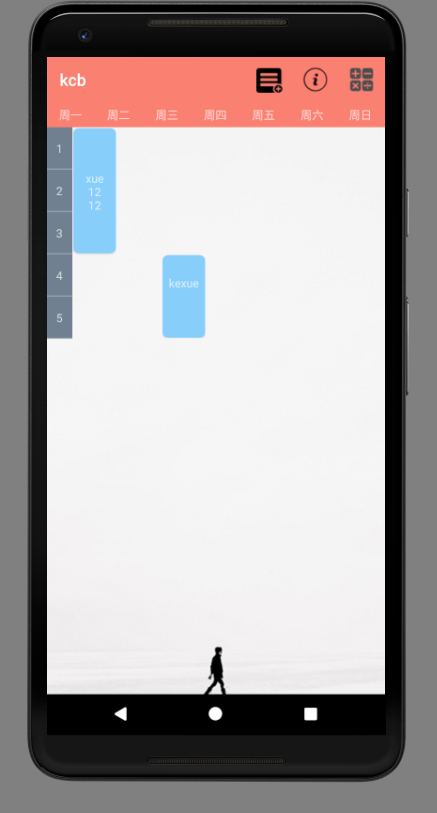
### AddCourseActivity

加课程活动

### Activity\_mian,activity\_add-course,activity\_about,left\_view,course\_card

课程表的所有见面文件，用Scrollview和帧布局嵌套线性布局做了主界面。

课程Card比较特殊，采用了cardview需要依赖。具体界面如下

，

用scrollview,线性布局做了添加界面，添加课程：



关于界面：



### Menu----toolbar

菜单，分别为添加课程，关于，计算器

### calMainActivity

这个是计算器主要的界面逻辑实现，在这个文件的末尾可以看到一个类的私有函数 \* calculate() \* ; 在这个函数中调用了同文件夹下的另外一个类 \* calculator （是一个静态类）中的一个功能接口\* .do\_calculate(mathLine) \*\*， 这个接口接受一个String的数学表达式如："((((1-2%)+3)4)/5)^3"，我用正则表达式来要求表达式尽量规整（括号对称，符号不乱写），而规整的表达式输入 是在用户输入是限定其输入标准的表达式，目前只支持简单的运算{"+","-","","/","^","%"}，运算可以拓展，只是目前没有接着写。而这个 功能接口返回一个运算结果String;运算结果中包含了java自带的运算异常

### Calculator

这便是上文中提到的计算实现类：其中包含

calcultor //主静态类

do\_calculate(String mathLine) //**计算的接口函数**

**输入：** (String)算术式，如：可以复杂到 "((((1-2%)+3)\*4)/5)^3"

**输出** (String) 异常处理后的运算结果，精度默认为Double的精度

makeArray(String mathLine) //处理用户输入并将其存入一个缓存堆栈 tempStack,可以不用管

makeStak(String pushTem) //从tempStack中逐项入栈的函数功能。可以不用管，但是如果拓展运算功能的话需要多匹配一些运算符号的正则表达式！

calculate() //正真运算功能实现的地方，如果需要拓展运算功能需要在这里添加对应运算符号的匹配和功能实现！

### CalculateLogs

数据库文件，建一个表用来存储历史记录

写insert（）插入函数 delone()删除一条clearAll()删所有数据getAllRecords读取所有数据

“=”是插入数据

双击del是删除一条所选数据

长按Ac是删除所有数据



图为查看历史数据



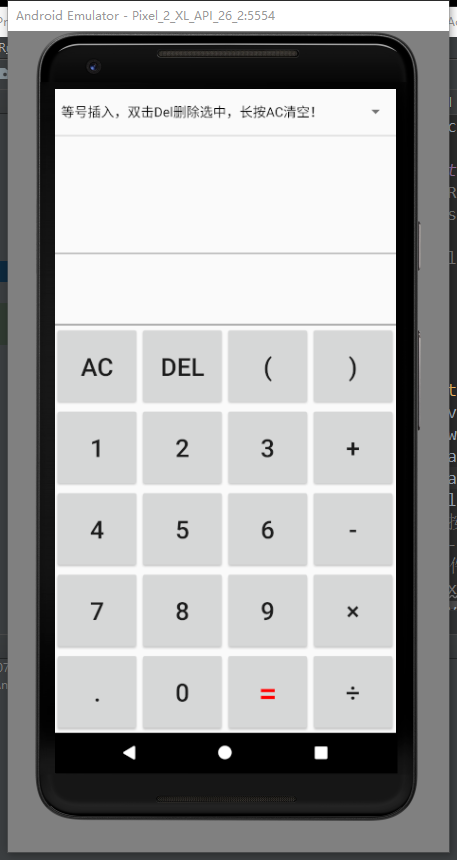
图为清除所选数据



图为清空所有数据

### calactivity\_main

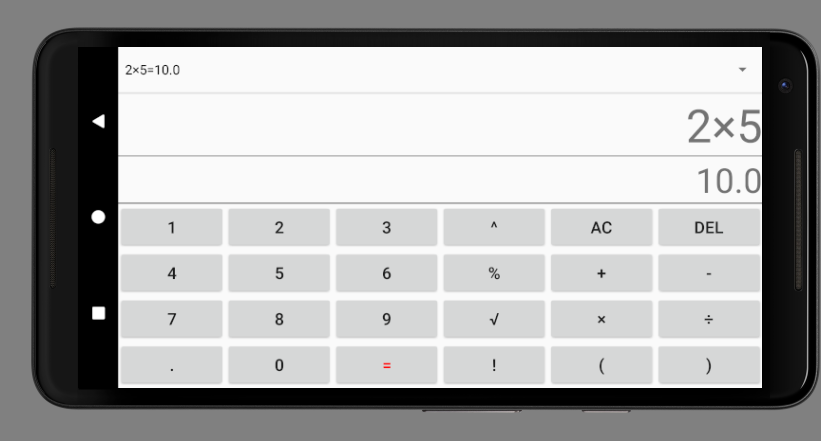
竖屏界面。线性布局加textview ,button ,spinner 等控件，去掉标题栏和状态栏显得更简洁。界面如图



### calactivity\_main(land)

横屏界面。线性布局加textview ,button ,spinner 等控件比竖屏多4个按钮。

横向界面如图：

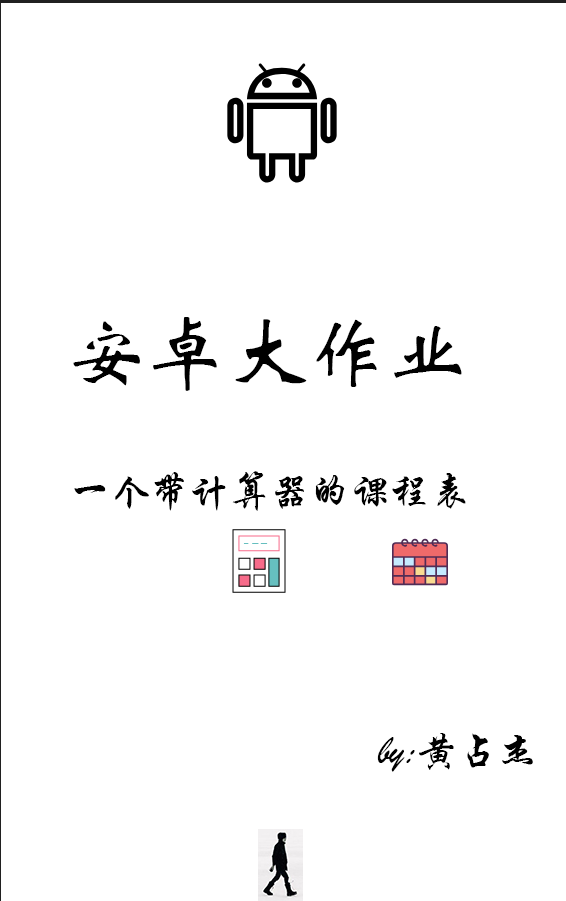


### PreActivity

开启程序界面

### Activity\_pre

如图：



功能：添加课程，删除课程，显示课程，计算器，看计算器的历史记录，删除历史纪录，删除一条历史记录，实时计算，横屏计算

亮点：实时计算，计算器功能，简单明了的布局

技术：SQL数据库技术