2017-6-23

The SuperDiary team

赖成锴 廖溢机 刘中喆 杨照功 杨梓阳 朱达辉

https://github.com/SDPCoder/SuperDiary/

SuperDiary

软件设计文档

目录

[技术选型理由 4](#_Toc486024944)

[模块划分 4](#_Toc486024945)

[设计技术 5](#_Toc486024946)

[设计模式 7](#_Toc486024947)

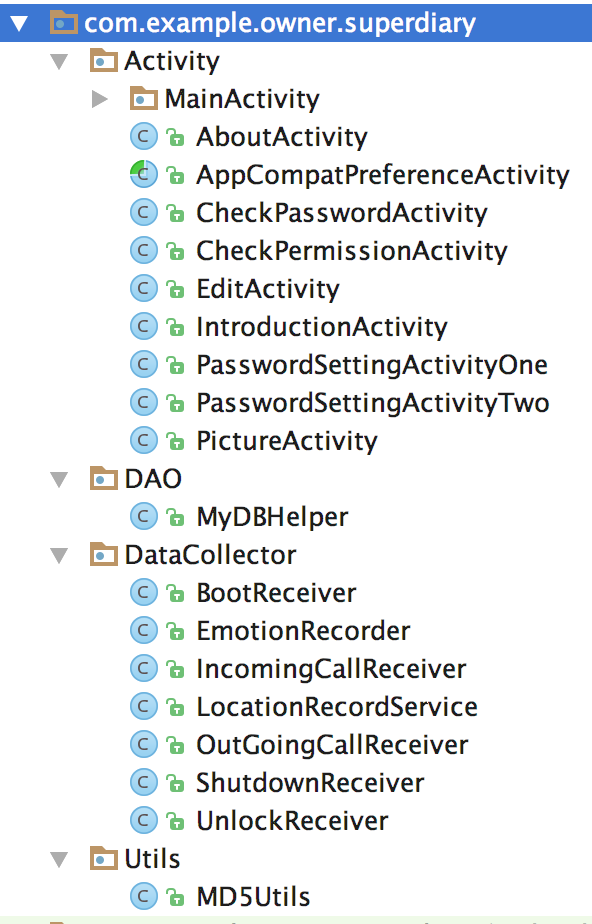
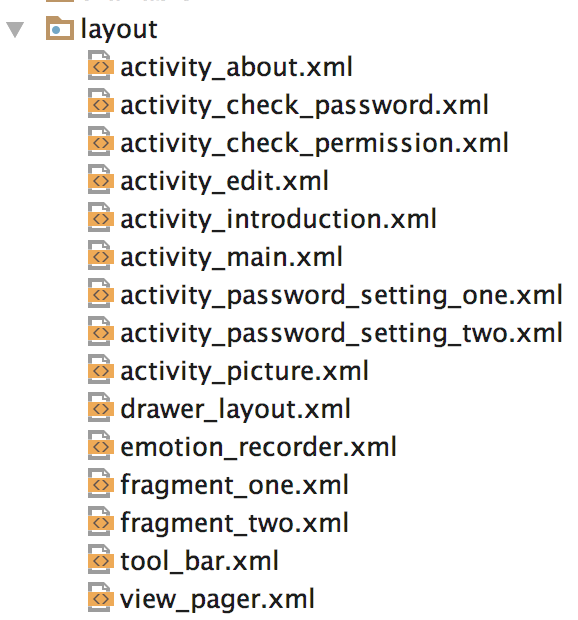
# 技术选型理由

本项目采用**Android应用开发**。Android系统是世界上覆盖率最高的移动操作系统。比起iOS系统，Android系统更为开放，提供更多的API。如开机广播、锁屏广播、持续运行的后台服务等，这些API是iOS所不能提供的，也是我们的APP所必须的。

此外由于Android系统是开放的，在Android系统上开发并不需要提交高额的费用。

# 模块划分

我们的模块划分为数据库访问模块、页面模块（XML）、页面与后台交互模块（Activity）、数据收集模块、小工具模块。小工具模块的主要内容是MD5加密模块。

在开发初期，这个模块的划分并不清晰。主要的分工模式是按Activity划分，每个人实现相应的功能。经过一段时间的开发之后，我们发现，Activity代码中有大量重复实现的功能。我们经过研究得出结论，可以将数据库的访问抽象出来，然后专门由一个人维护。此外，数据的收集也应该由专门的模块进行，其他人只需要调用接口即可。于是我们对软件进行重构，改进了设计方案。如此可以有效降低开发成本，提高代码的复用性与可维护性。

# 设计技术

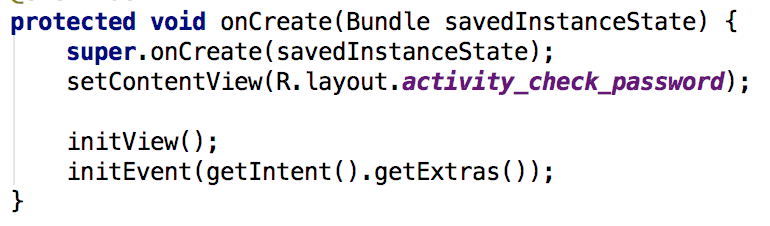
1. **结构化编程**

首先，如上所述，我们的软件进行了模块划分，这是结构化编程的体现之一，不再赘述。

此外，我们的代码中也体现了结构化编程的思想。结构化编程强调，代码应该自顶向下，我们的代码体现了该思想。

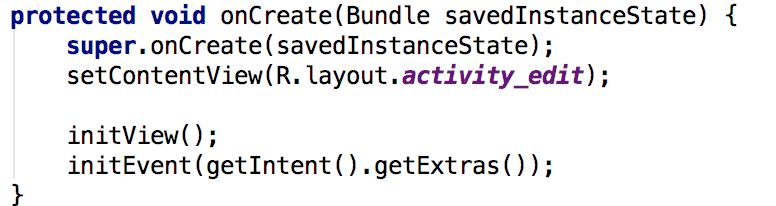
具体如下：

1.1 所有Activity中获得View对象的代码，都写在initView()函数中，由Activity的OnCreate函数调用。



上图是checkPasswordActivity.java的20~26行。

1.2 所有Activity中设置监听器的代码，都写在initEvent(…)函数中， 由Activity的OnCreate函数调用。



上图是EditActivity.java的24~30行。



上图是EditActivity.java的32~56行。

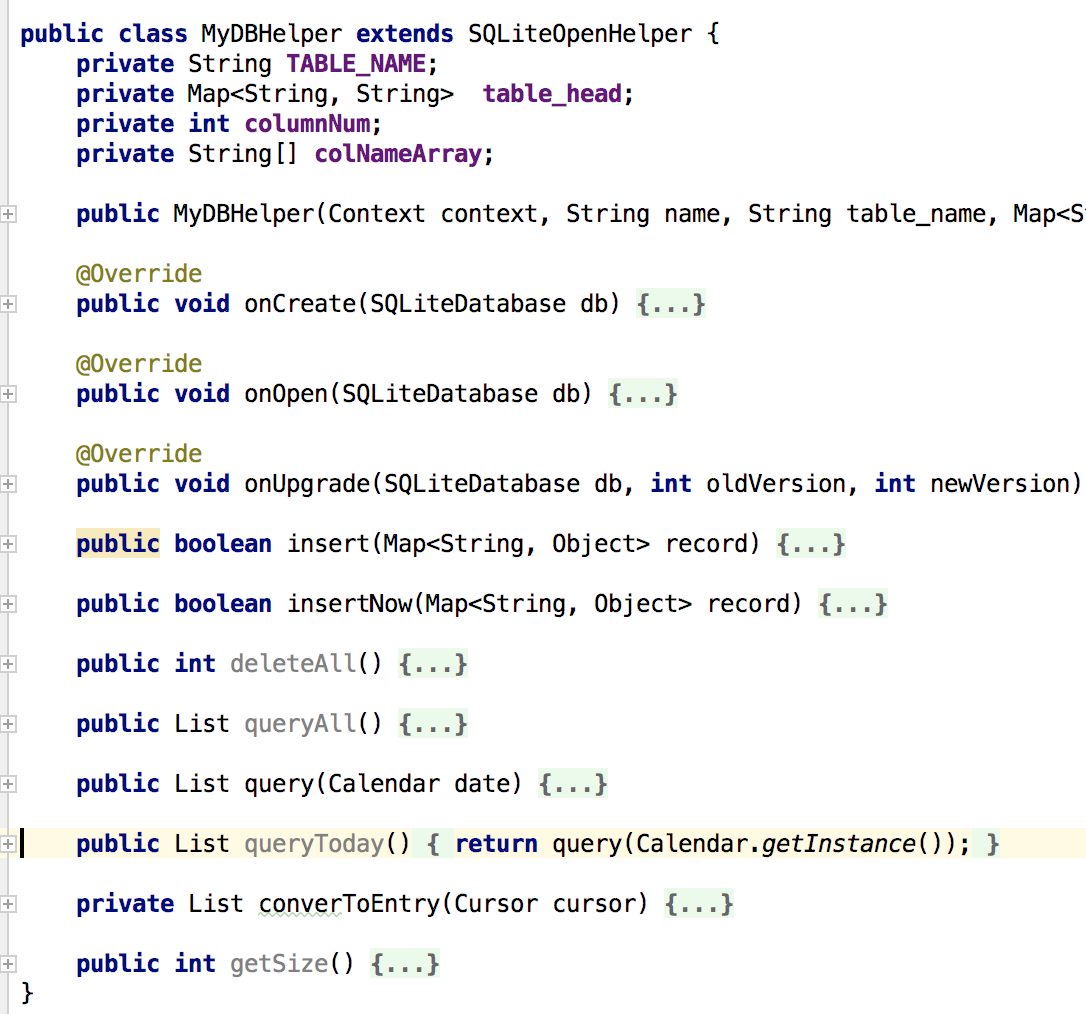
我们还规定了良好的代码规范，这满足了结构化编程中清晰第一的原则。

所有函数以首字母小写的驼峰命名法命名。

（EditActivity.java, Line 64）

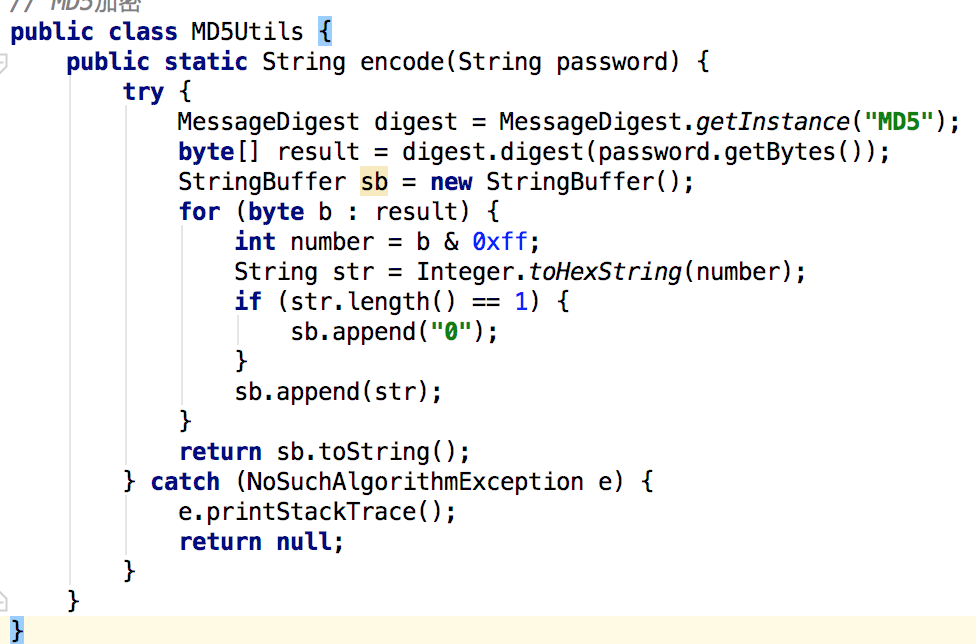
## 面向对象编程

面向对象编程的最基本要求是将由数据和操作数据的方法组成的对象作为编程的基本元素。在本次项目中，我们将数据库抽象为一个数据库对象，然后通过代理类MyDBHelper来提供对其的增删查。



（MyDBHelper.java）

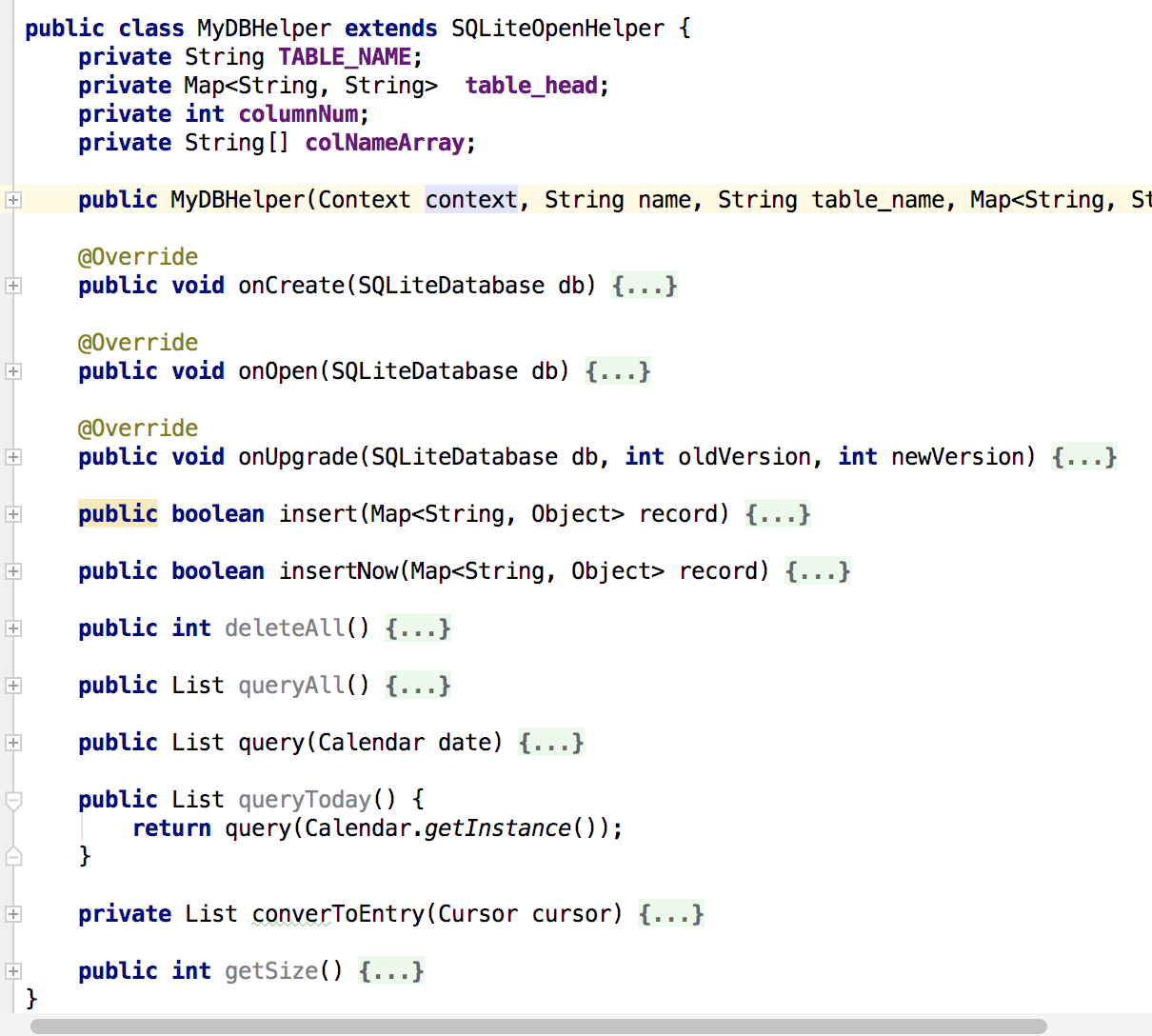
此外，我们也将MD5加密工具抽象成一个对象，对外提供加密函数。输入一个字符串，则可以输出加密后的字符串。



# 设计模式

1. 数据访问对象模式

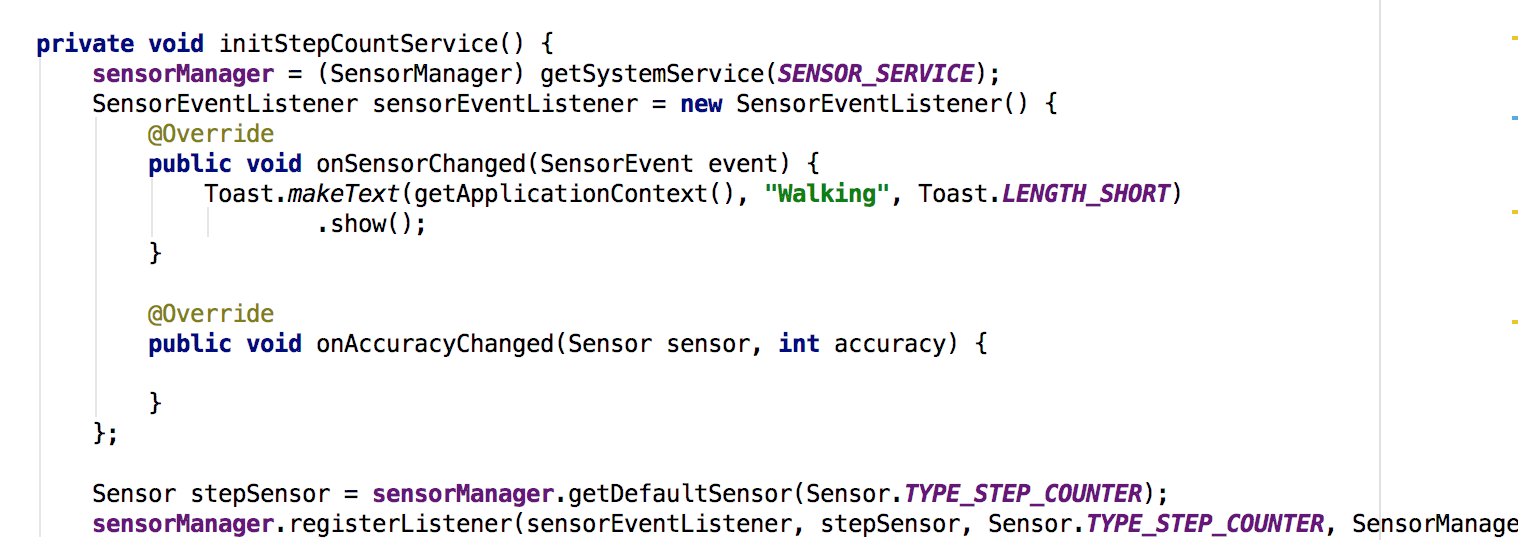
数据访问对象模式（Data Access Object Pattern）或 DAO 模式用于把低级的数据访问 API 或操作从高级的业务服务中分离出来。我们的代码中的MyDBHelper实现了低级数据访问API的抽象。



1. 单例模式

单例模式可避免对个对象造成的语义不明确和代码冲突。Android 框架中有许多单例的类，我们使用了其中一些。

如在LocationRecordServices中，我们使用了GPS传感器的API。该API返回一个单例的对象，表示手机传感器。



通过对这些单例API的调用，我们对单例模式有了更深入的认识。