北京理工大学计算机学院

《Android技术开发基础》课程设计

班级\_07112106\_\_\_学号\_\_\_1120213595\_\_\_姓名\_\_\_林翔\_\_\_\_\_

#### **1 App的运行与开发环境**

1. **运行环境**： 8.0以上版本Android的Android手机/平板
2. **部署方法：** 直接安装贝利体育.apk即可
3. **开发环境：** Android Studio 3.1.2
4. **手写代码行数：**手机端约1800行（Kotlin）

**2 App功能说明**

**2.1 app介绍**

本app为一款**跑步记录**软件。主要实现功能为**地图显示**、**实时记录跑步运动数据**、**历史跑步数据显示**、**删除历史数据**等。

**2.2 app界面**

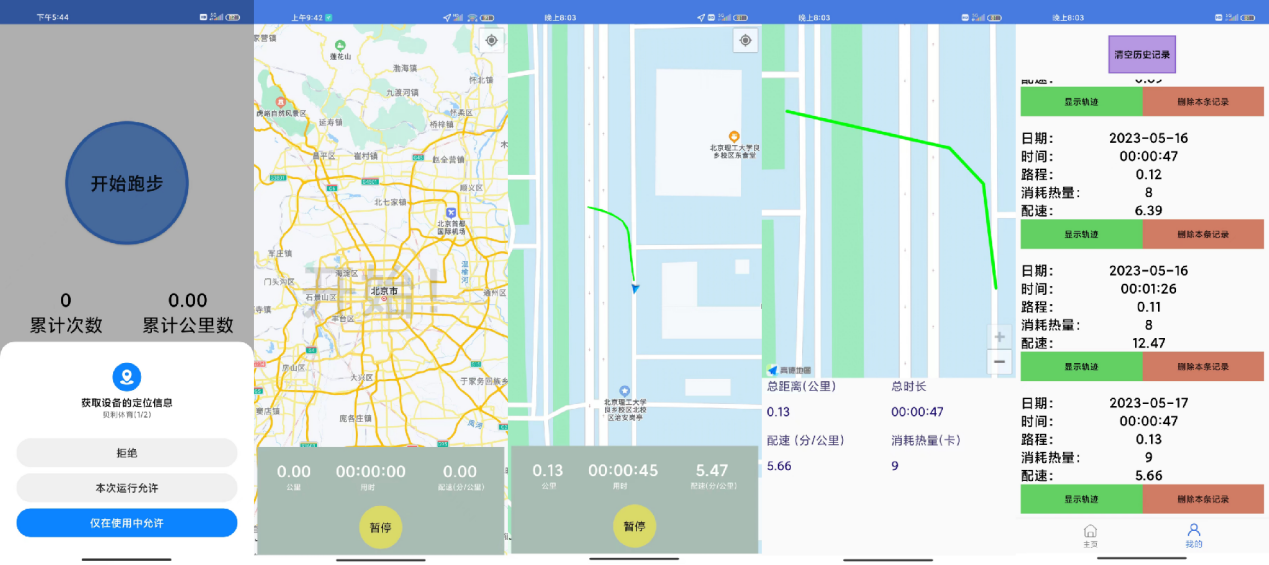
（1）首次进入app，会自动发起权限申请，请求获取定位权限以及读取文件的权限。

1. 在导航栏“主页”下，为开始运动按钮，并实时显示所有历史记录的总和次数以及累计公里数。
2. 单击主页下的开始运动按钮，会跳转到运动界面。在倒计时三秒后开始自动记录运动路径、用时、里程、配速等信息。

（4）单击运动界面下的暂停按钮，运动时间记录暂停，并显示完成、继续按钮。

（5）单击运动界面下的完成按钮，会结束本次运动记录并自动保存，之后页面跳转至本次运动的详细信息页面，显示全程运动路径、运动时间、运动里程等信息。

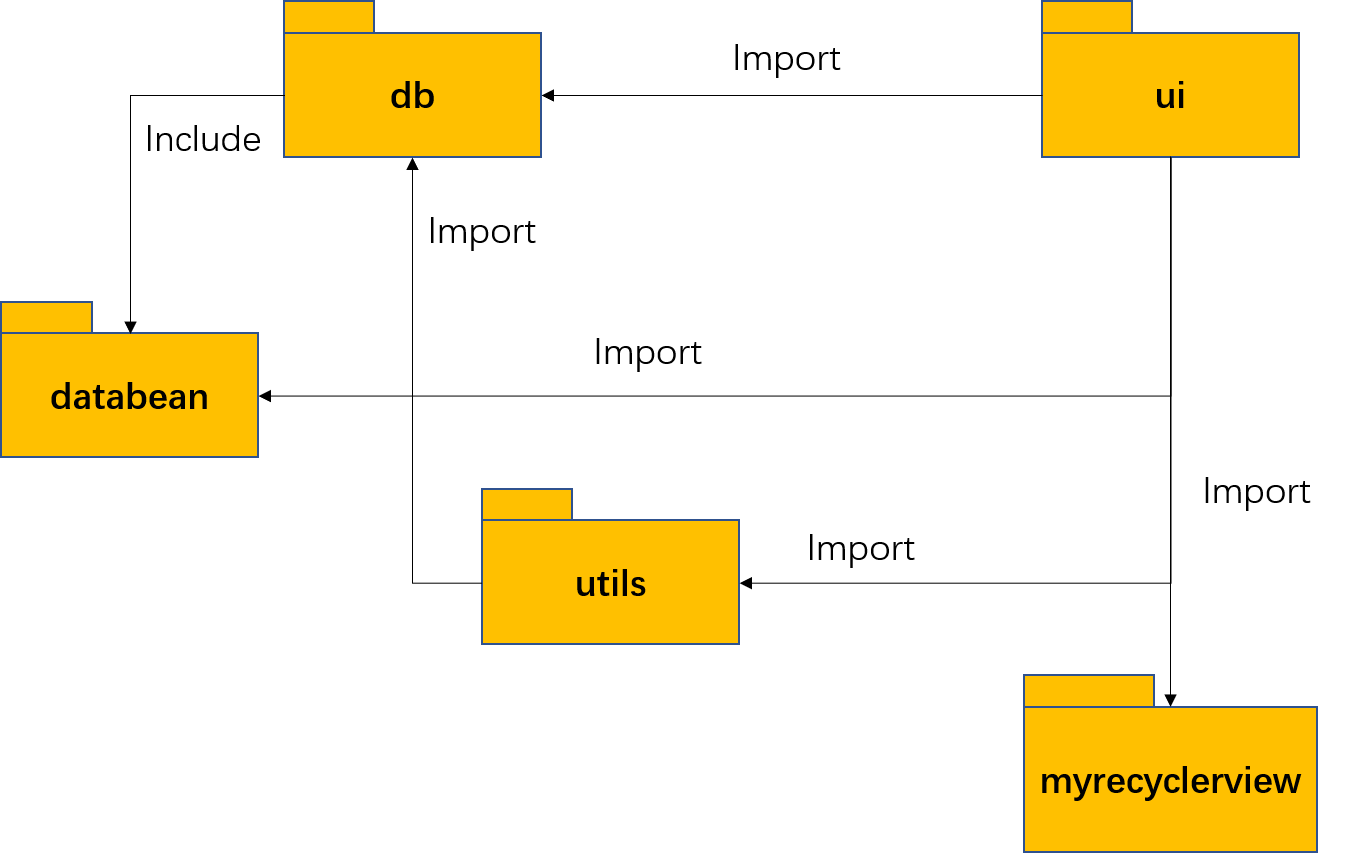
（6）在导航栏“我的”下，显示所有运动历史记录，可以随时查看某次运动的详细信息，并实现删除单条或删除全部记录功能。



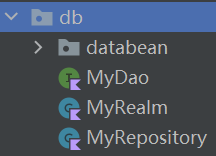
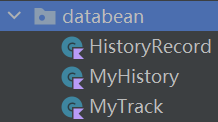
#### **3 App架构设计及技术实现方案**

**3.1 包介绍**

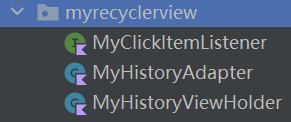
包图如下：



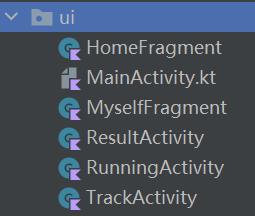
db包：包含数据对象包（databean）以及数据库相关类

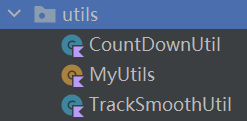
myrecyclerview包：与组件recyclerview相关的类



ui包：包含activity和fragment类

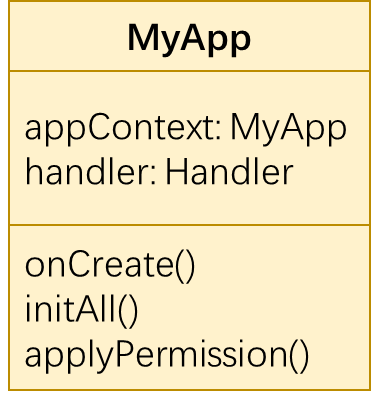


utils包：包含相关工具类



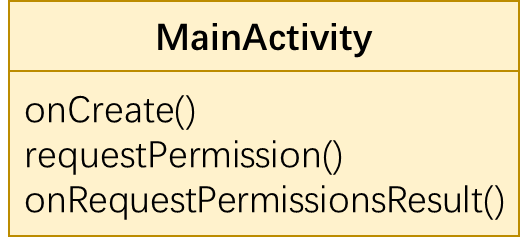
**3.2 类介绍**

**3.2.1 MyApp类**

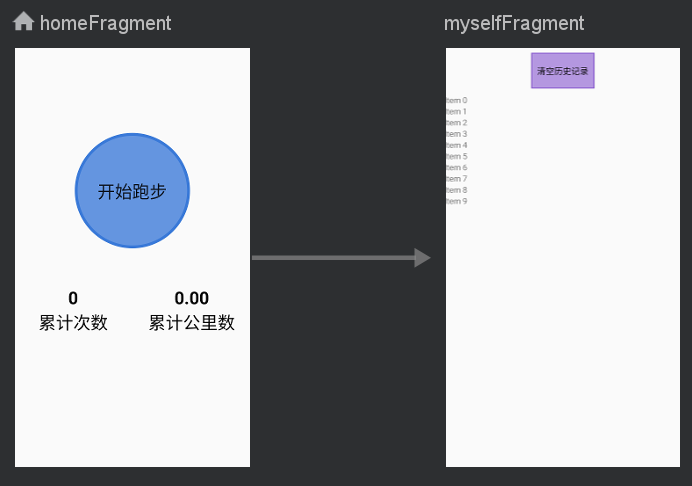


程序入口。继承Application类，在onCreate()方法中调用initAll()和applyPermission()方法。其中initAll()为初始化数据库、appContext以及handler。applyPermission()方法用于实现地图使用的相关权限申请。

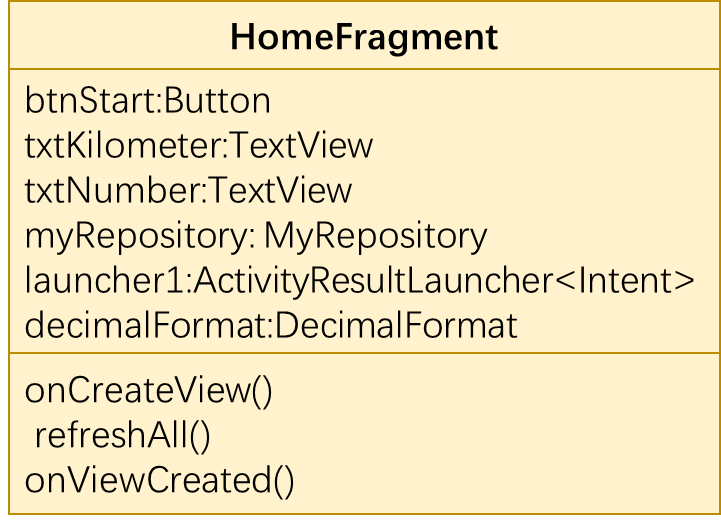
**3.2.2 MainActivity类**



本类引用第三方组件EasyPermission（权限申请）和Navigation（导航）；其中requestPermission方法用于申请权限；重写方法onRequestPermissionsResult()设置权限请求结果；在onCreate()中关联NavController与BottonNavigationView，并申请权限。其中导航为两个fragment的切换。

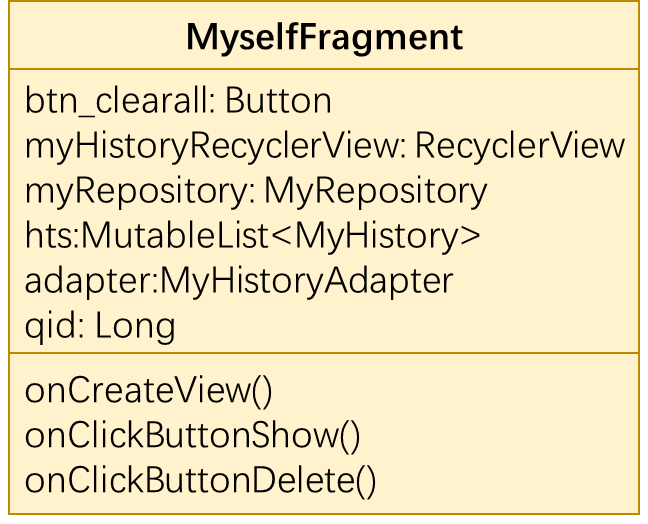


**3.2.3 HomeFragment类**



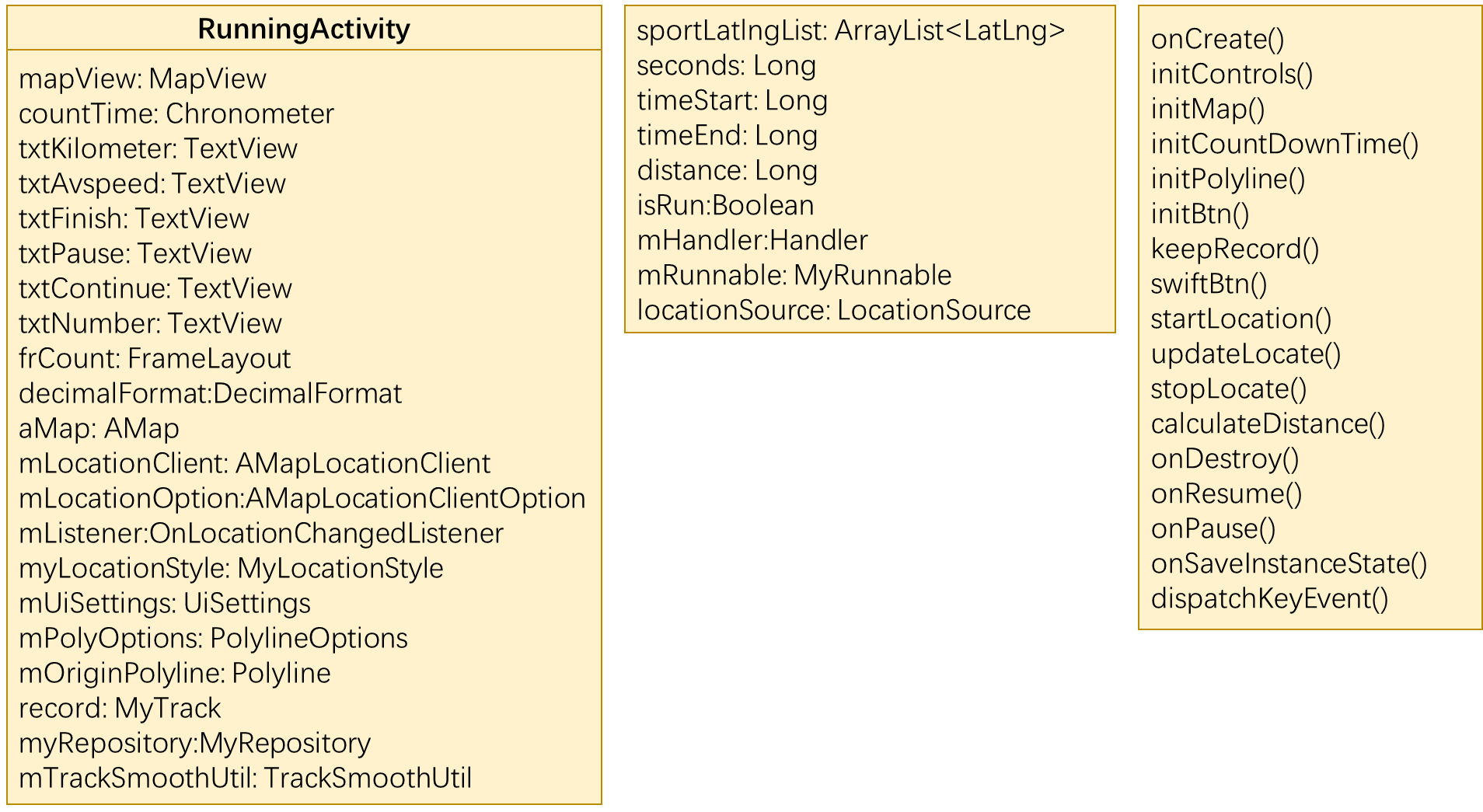
其中launcher1用于从其他activity（RunningActivity）中获取结果，如果有新的运动记录，则调用refreshAll()方法以刷新组件。

**3.2.4 MyselfFragment类**



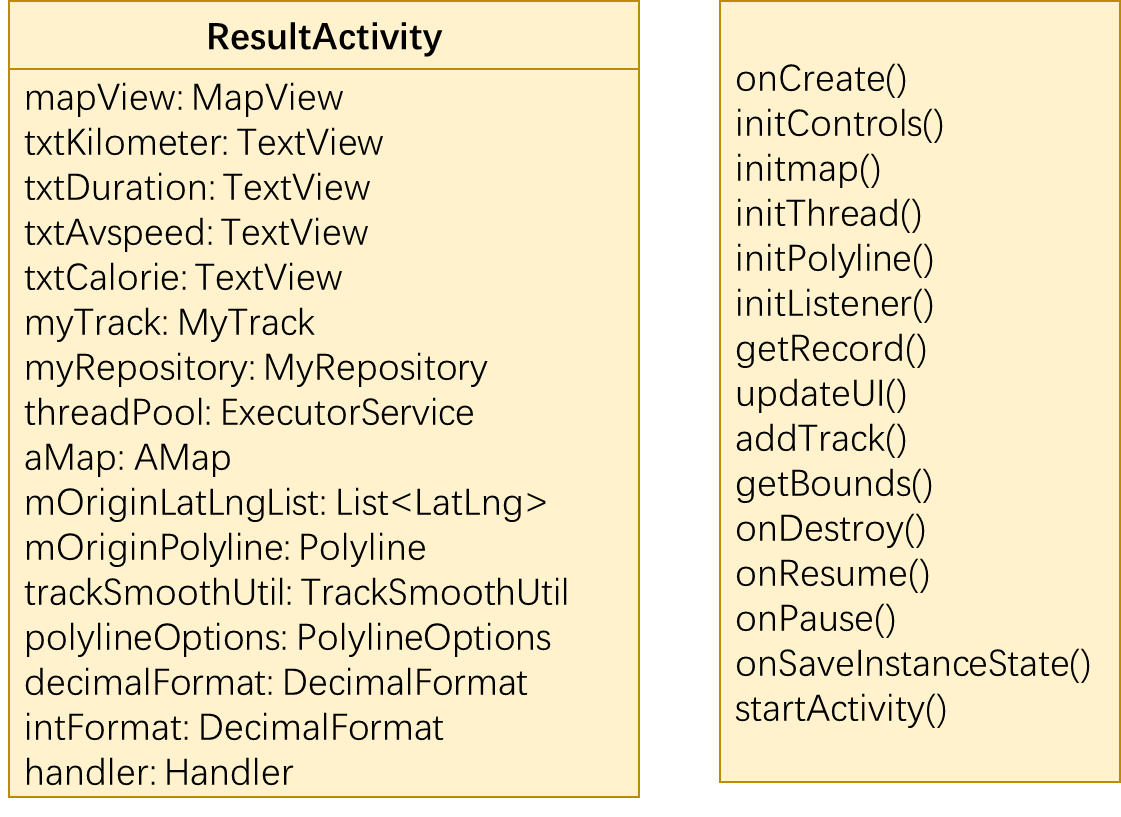
本类视图采用recyclerview，与myrecyclerview包相关。本类调用自定义的接口MyClickItemListener，重写了方法onClickButtonShow()与onClickButtonDelete()，实现显示历史轨迹以及删除单条记录的功能。

**3.2.5 RunningActivity类**



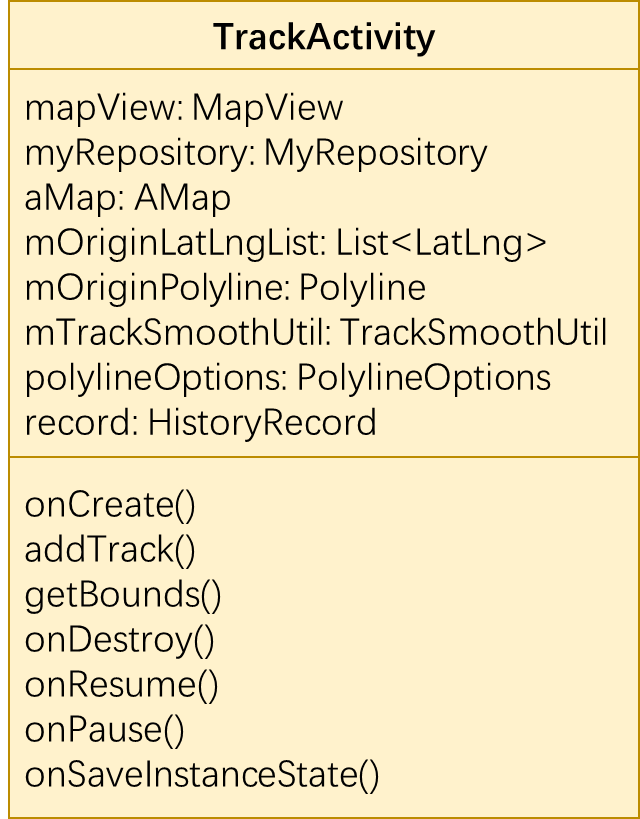
本类为本APP主要功能实现类。引用第三方组件高德地图相关部分。initControls()、initMap()、initCountDownTime()、initPolyline()、initBtn()方法初始化了控件获取、地图、倒计时、线段、按钮等。keepRecord()实现了记录的保存。swiftBtn()实现了按钮的切换。startLocation()开始定位。updateLocate()更新定位并绘制路径、计算配速等。stopLocate()停止定位。CalculateDistance()计算距离。重写onDestroy()、onResume()、onPause()、onSaveInstanceState()四个方法实现地图的生命周期调用。重写dispatchKeyEvent()方法以禁用返回键。

**3.2.6 ResultActivity类**



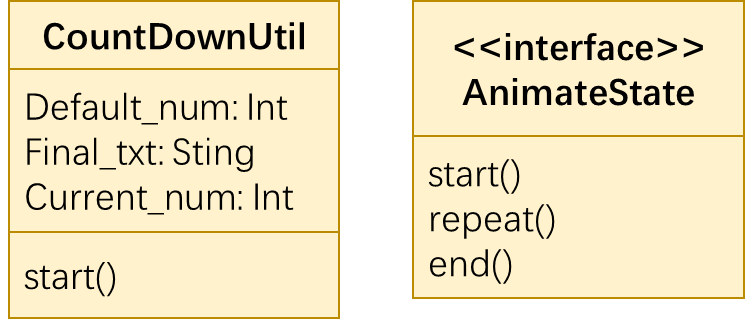
在RunningActivity结束并成功保存运动数据后，调用本类的startActivity()方法传入运动数据并启动本类。initControls()、initmap()、initThread()、initPolyline()、initListener()初始化了控件、地图、线程、线段、监听器等。getRecord()获取传入的运动数据，获取成功则调用updateUI()以更新组件。addTrack()用于添加路径到地图上。getBounds()获取地图限制范围。重写onDestroy()、onResume()、onPause()、onSaveInstanceState()四个方法实现地图的生命周期调用。

**3.2.7 TrackActivity类**



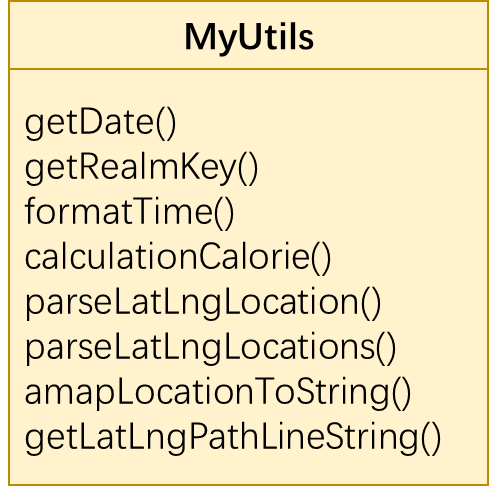
本类为历史轨迹显示的界面。在MyselfFragment中单击显示历史轨迹按钮则跳转至本页面。addTrack()用于添加路径到地图上。getBounds()获取地图限制范围。重写onDestroy()、onResume()、onPause()、onSaveInstanceState()四个方法实现地图的生命周期调用。

**3.2.8 CountDownUtil类**



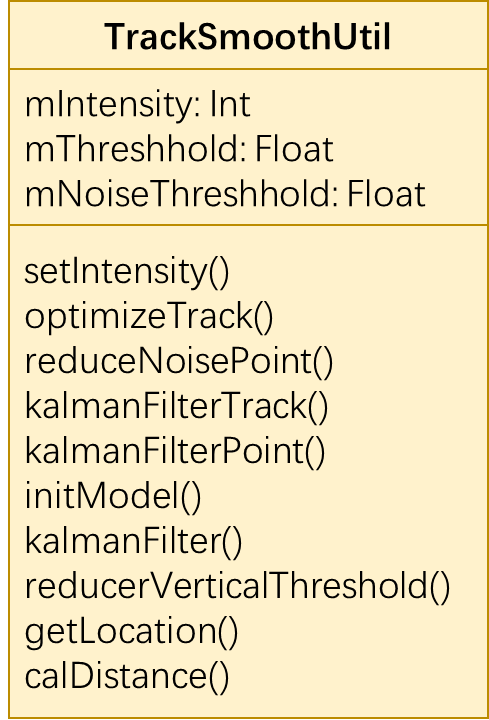
倒计时动画类。start()方法供外部调用，传入需要添加动画的组件以及组件的当前状态，动画中应用了透明度改变与缩放。定义内部接口AnimateState，给组件动画添加监听，在组件状态改变时调用不同方法。

**3.2.9 MyUtils类**



工具组件类。getDate()获取当前时间的格式。getRealmKey()获取RealmKey的密钥（Realm数据库的密钥必须为64位）。formatTime()将秒转化为XX:XX:XX的格式。calculationCalorie()计算热量消耗。parseLatLngLocation()、parseLatLngLocations()、amapLocationToString()、getLatLngPathLineString()方法与地图定位的类型改变有关，从高德地图官网获取。

**3.2.10 TrackSmoothUtil类**



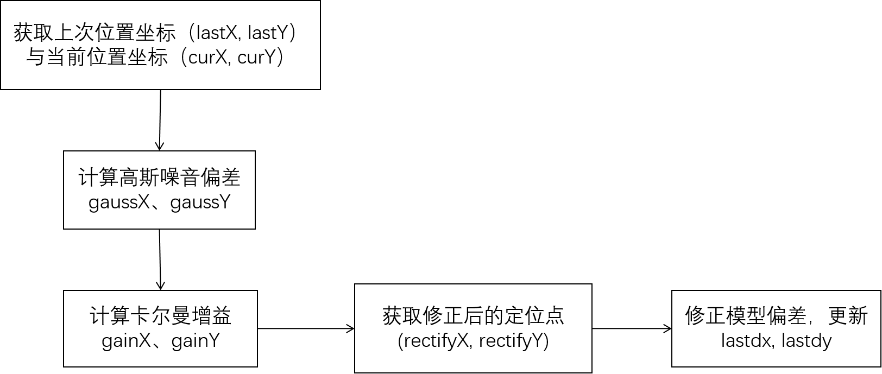
路径优化类。外部调用optimizeTrack()方法，对输入的点集进行去噪、滤波、抽稀等操作，输出新的点集，使路径更平滑。本类主要采用卡尔曼滤波算法（参考高德地图官方文档），算法主要介绍如下：

卡尔曼滤波分为预测过程和更新过程两个过程，在预测过程有：

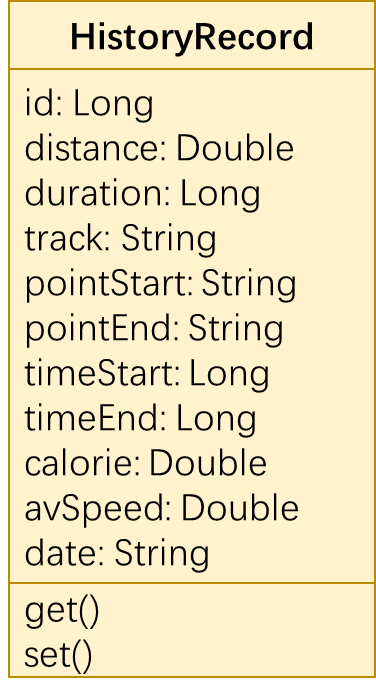
其中，表示在第周期的预测值，为系数，表示处理噪声，为本周期的预测误差，为协方差。

更新过程有：

其中，为卡尔曼增益，为缩放系数，为测量噪声的平均值，为物理量的测量值。在代码实现中，流程为：

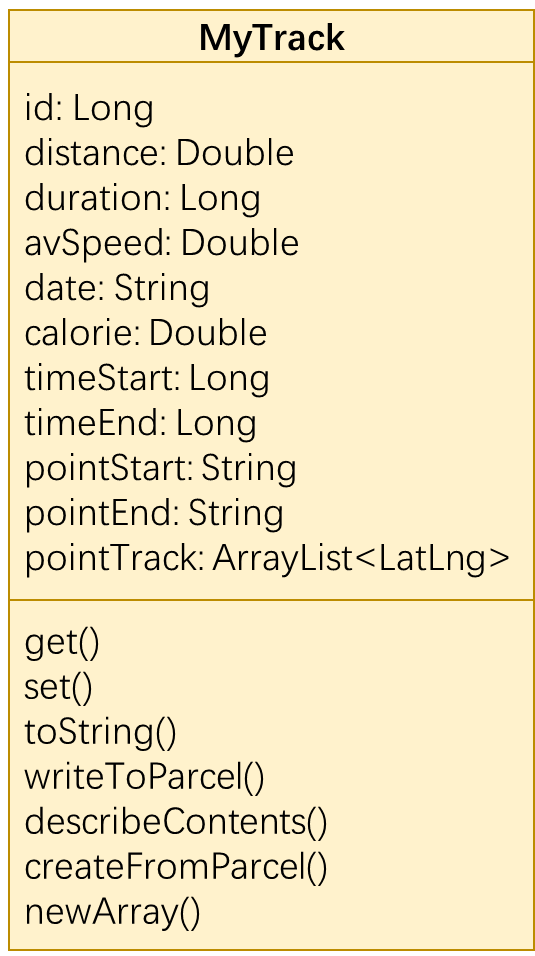


**3.2.11 HistoryRecord类**



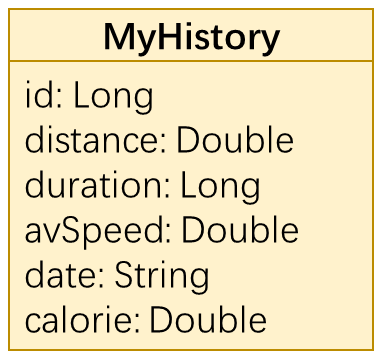
历史记录类。本类继承Realm()类，实现Serializable接口。对每一个成员均实现get()和set()方法。为数据库存储的信息。

**3.2.12 MyTrack类**



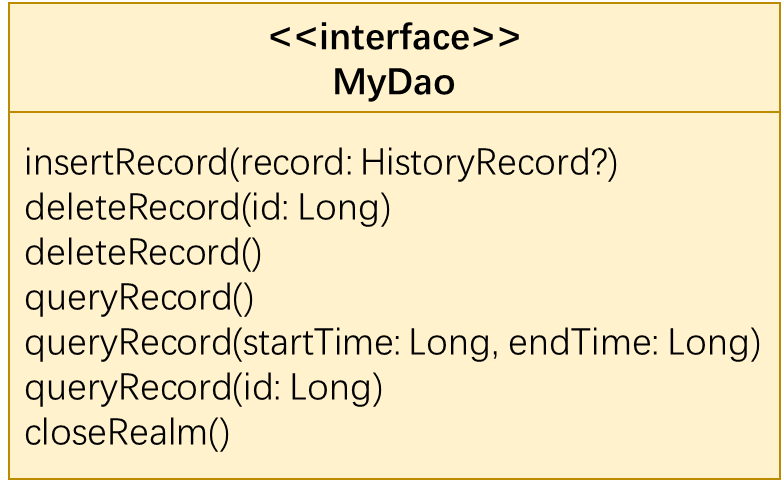
历史轨迹记录类。实现Parcelable接口。对每一个成员均实现get()和set()方法。重写了toString()、writeToParcel()、describeContents()、createFromParcel()、newArray()等方法。

**3.2.13 MyHistory类**



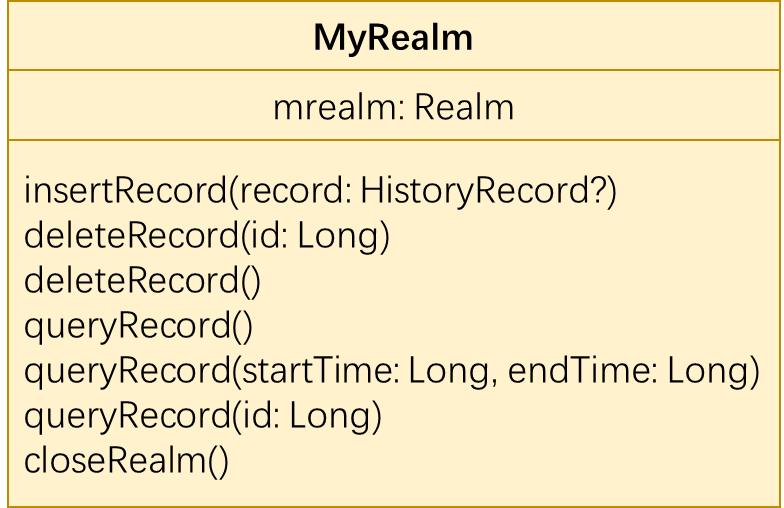
历史记录的数据类。主要用于recyclerview所显示的信息。

**3.2.14 MyDao接口**



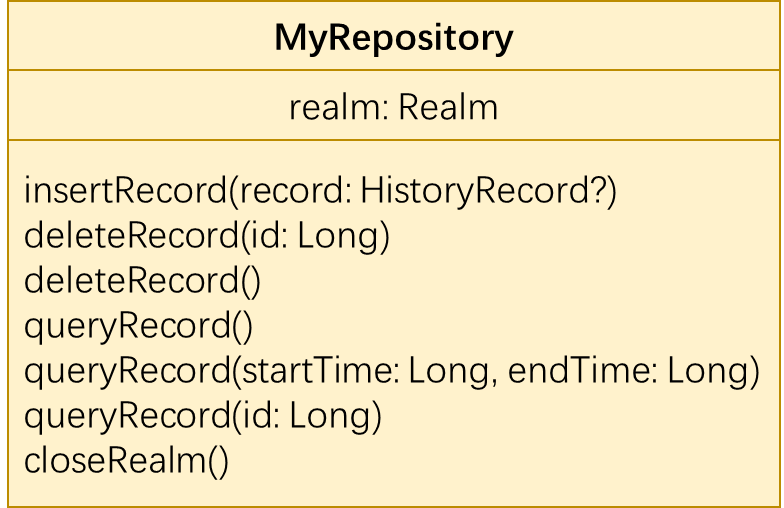
数据库的Dao接口。使用重载，定义了对数据库进行增加、删除、查询操作的方法。同时定义关闭数据库的方法。

**3.2.15 MyRealm类**



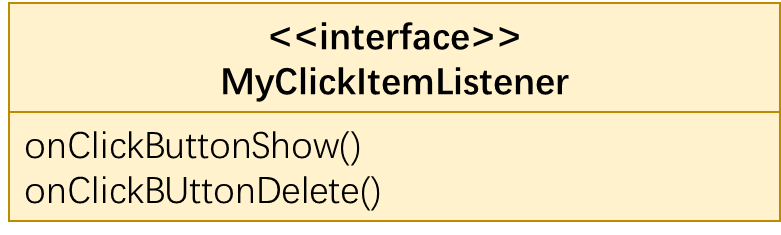
数据库对象类。实现了自定义接口MyDao。在初始化时创建数据库mrealm。

**3.2.16 MyRepository类**



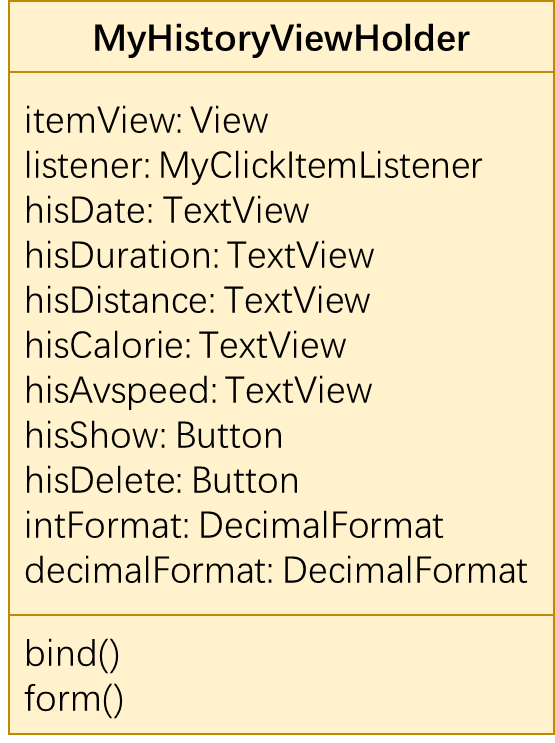
供外部调用的数据库repository类。实现了自定义接口MyDao。从外界传入MyRealm类对象，实现对数据库的操作。

**3.2.17 MyClickItemListener接口**



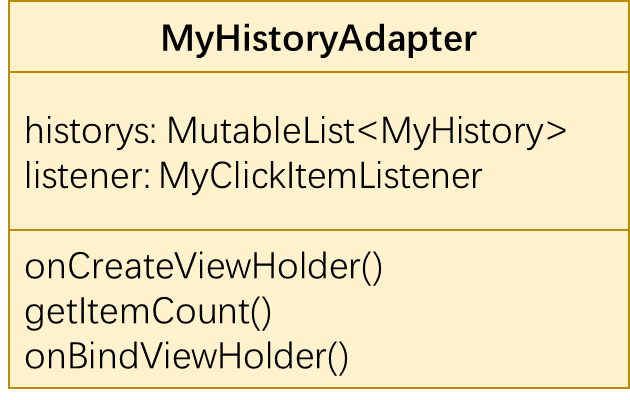
RecyclerView的事件响应接口。定义了单击按钮“显示历史轨迹”“删除本条记录”的方法。

**3.2.18 MyHistoryViewHolder类**



ViewHolder类。继承RecyclerView.ViewHolder()类，从外部传入View以及Listener对象，在bind()中实现事件绑定。form()供外部调用以实例化ViewHolder对象。

**3.2.19 MyHistoryAdapter类**

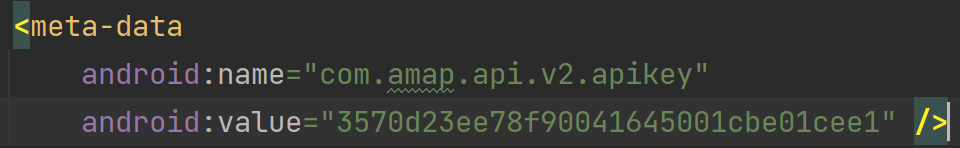


Adapter类。继承RecyclerView.Adapter<MyHistoryViewHolder>()类，从外部传入要显示的数据集合和listener对象，实现数据的提取与显示。并重写onCreateViewHolder()、getItemCount()、onBindViewHolder()等方法。

**3.3 组件介绍**

**3.3.1 高德地图组件**

本项目主要以该组件为基础，进行后续的开发。该组件主要实现功能有地图显示与交互、在地图上绘制线段、坐标获取、距离计算等。使用时需要在高德地图官网上注册并获取key值，并在清单文件中添加下述语句：



本项目主要用到下述几个类：

|  |  |
| --- | --- |
| AMap类 | 地图对象类，封装了地图的属性设置、图层变更、事件交互等接口的类。 |
| AMapOptions类 | 用于MapView对象的初始化。 |
| MapView类 | 一个显示地图的视图（View），它负责从服务端获取地图数据，它将会捕捉屏幕触控手势事件。 |
| UiSettings类 | 地图内置UI及手势控制器。 |
| Polyline类 | 定义地图线段覆盖物。 |
| PolylineOptions类 | 线段的选项类。 |
| LatLng类 | 存储经纬度坐标值的类，单位角度。 |

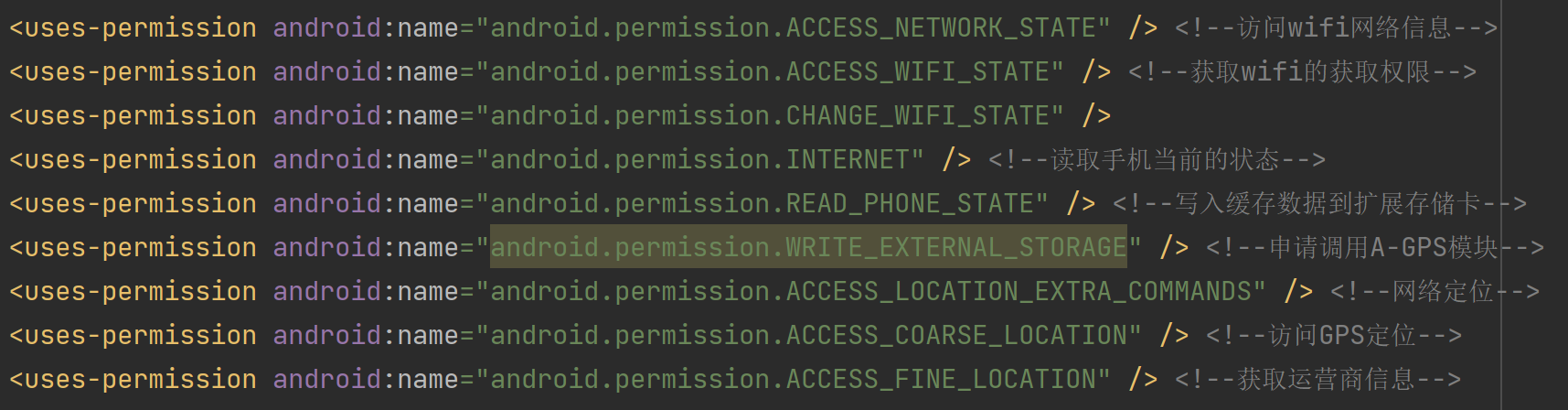
**3.3.2 Realm数据库组件**

Realm数据库的初始化需要指定数据库名称、版本号、密钥等。构建数据库模型需要继承自RealmObject类，并将class设为open。在activity中使用realm数据库需要手动打开和关闭。本项目中使用到的主要方法如下：

|  |  |
| --- | --- |
| 增加数据 | 使用executeTransaction方法进行同步操作。 |
| 查询数据 | findAll()：查询全部；findFirst()：查询第一条；  equalTo()：根据条件查询；  where()：区间查询 |
| 删除数据 | deleteFromRealm()同步删除；  deleteAll()删除全部 |

**3.3.3 EasyPermissions库**

权限申请库。首先需要在清单文件中声明需要申请的权限，如下：



之后在activity中重写方法onRequestPermissionsResult()，以及编写申请权限的代码。

**3.4 主要功能逻辑**

**3.4.1 倒计时动画**

CountDownUtil类中：定义了两个start()函数。其中一个供外部调用，传入控件视图对象和控件状态，另一个供内部调用，增加了倒计时的时间。动画上，选取透明度改变动画和缩放动画，并为缩放动画添加监听，在重复播放时改变显示的文本，在其结束时将控件设为不可见。最后将两个动画添加到同一个animationSet中以达到同时播放的效果。定义的AnimateState接口供外部调用，以添加在动画开始、重复、结束时的代码。

**3.4.2 定位显示**

先获取地图控件的map对象aMap，再给aMap添加locationSource监听，在监听方法activate()中设置定位改变监听为mListener，启动定位startLocation()。并在updateLocation()中为mListener添加定位改变方法，达到实时获取当前位置的效果。

**3.4.3 定位更新**

RunningActivity类中：在倒计时结束时，调用startLocation()方法开始定位，并为mLocationClient对象添加定位监听对象aMapLocationListener，在定位成功时调用updateLocate()方法以更新视图控件（绘制轨迹、计算路程、计算配速等）。

#### **4 技术亮点、技术难点及其解决方案**

**4.1 技术亮点**

1. 采用多线程技术，在地图更新显示、数据库存取等部分都应用了线程进行处理，使APP运行更流畅稳定。
2. 使用Realm数据库保存数据。具有快速、性能优越等特点。
3. 使用了路径优化算法，使路径的显示更丝滑逼真。
4. 相比于北理工的体育app（北理体育，已停用），界面更为简洁美观，且支持查看历史轨迹、删除历史记录等功能。
5. 合理设置接口，使代码更简洁直观。
6. 分模块进行开发，写了许多个demo，开发合理有序。

**4.2 技术难点及解决方案**

1）权限申请？

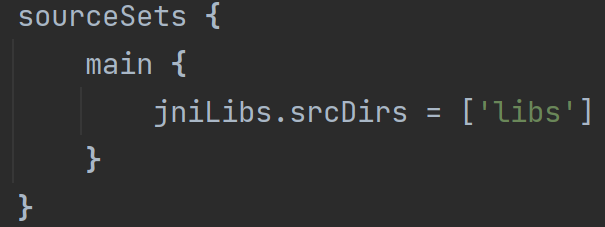
解决方案：查阅相关资料，使用EasyPermissions库完成，并在清单文件添加相应权限。

1）高德地图api的应用？

解决方案：在官网申请key，在获取本机安全码时SHA1时遇到重重困难，在终端尝试各种指令都无果，最终在csdn的角落找到解决方法，得到正确前置操作步骤及对应的指令。

2）地图显示为黑屏？

解决方案：查询相关资料，在模块的build.gradle中添加配置。



3）实时的路径绘制？

解决方案：在每次位置发生更改后及时进行视图更新。

4）路径不丝滑？

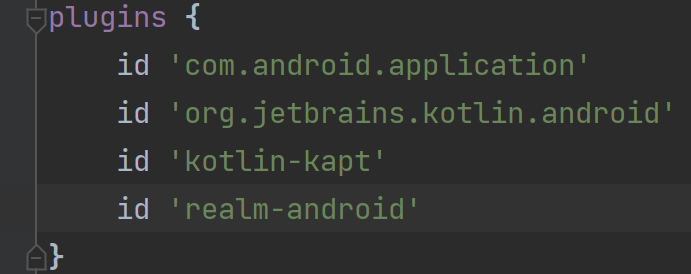
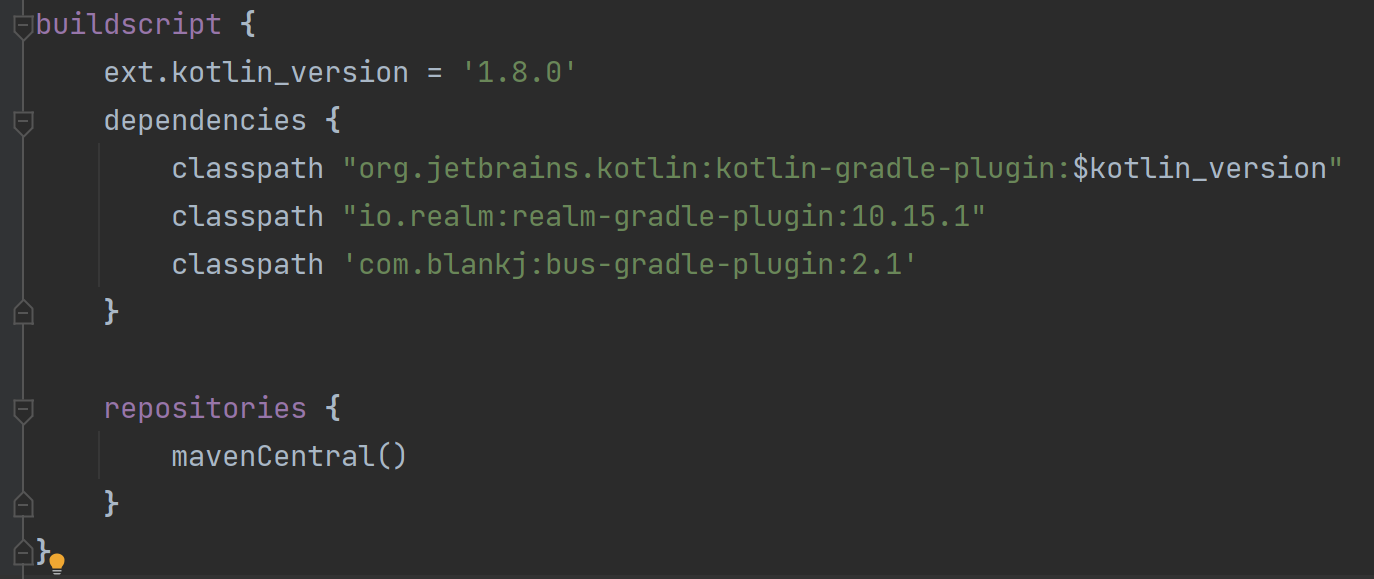
解决方案：最开始采用高德地图内置路径优化方法，但问题在于内置的路径优化仅针对车辆轨迹使用（即轨迹只会在主干道上显示且非常直，不真实）；之后查阅各种资料，终于在高德地图官网找到解决策略，即采用卡尔曼滤波方法，使用路径优化类来解决。

5）倒计时动画实现？

解决方案：开始没有思路，直接在activity中应用动画方法，但总是无法做出想要的效果。在参考相关资料后，将其单独封装为一个类，在类中修改代码，调试更为方便，并使用了类内接口以完成代码拓展。

1. Realm数据库的使用？

解决方案：直接在项目的build.gradle中添加classpath并在模块的build.gradle中添加两个plugin会报错！！无法应用？？最终解决措施为在项目的build.gradle重新添加整个buildscript配置，并同步等待下载（存储空间又占好多）。



1. Realm数据库编译报错？【Execution failed for task :app:kaptDebugKotlin】

解决方案：经过一早上的检查，发现两个问题：首先对于继承自RealmObject的类，必须让其为open！！其次，对于@Required的成员，只能使用（var x: Int? = null）！的形式声明，而不能用（var x: Int = 0）！！

1. 新建数据库对象报错？

解决方案：经排查后发现为encryptionKey的位数错误。Realm数据库要求必须为64位，经过一系列操作终于凑出64位。

1. 在不同activity中重复使用地图视图时会崩溃？

【Pointer tag for 0x7518d725e0 was truncated】

解决方案：经过官网查找，在清单文件中的application里加上android:allowNativeHeapPointerTagging="false"即可解决问题。

#### 5 简要开发过程

4月21日 确定选题——做一个类似北理体育的记录跑步数据的运动软件，并查找相关资料，使用高德地图API，注册，获取安全码SHA1（弄了接近1h，哭）。完成权限申请（采用easypermission）。

4月22日 学习地图使用，实现地图显示。

4月24-25日 学习路线绘制相关API，参考高德官网实例初步实现路线绘制。

4月27-30日 完成路径优化类。

5月4-6日 进行页面布局设计，实现导航栏，完成页面跳转规划。

5月7-9日 学习使用动画，完成倒计时动画类。

5月10日 完成运动界面的三个按钮（暂停、完成、继续）的逻辑实现。

5月10-11日 成功完成数据库接入，并规划数据库类。

5月12日 完善数据库的应用。

5月13日 完成recyclerview的实现，添加删除单条数据/全部数据的功能

5月14日 开始检查功能代码，添加细节（UI组件更新，禁用返回键、屏幕旋转等）。

5月15日 跳转代码细节，并完成界面的美化（添加圆形按钮、合理色彩搭配等）。至此app开发基本完工。

5月16=18日 开始撰写开发文档。

#### 6 学习感悟及对本课程的建议（可选）

本门课程算是我本学期最有收获的一门课了（上学期的java也是哈哈哈），能自主完成一个app的开发，从无到有，实在是非常有成就感！！也算完成我的一个小心愿了（之前本校的体育app深受同学们的诟病，我就想着何时能自己开发一个类似app），终于在学过本门课程后有了动手实践的能力，真的很感谢能有这个机会！！本学期课程都压在上半学期学的还是比较匆忙的，但我也是坚持每节课都到教室（有时看着越来越空的教室也会不禁感叹），很幸运能跟着老师一步步完成理论学习，感谢老师的指导，也感谢能坚持下来的自己。