

《Android 开发从入门到实战》

数据存储

让人人都能享受到高品质的教育服务

数据存储

目 录

SharedPreferences

A

数据库SQLite

B

存储卡的文件操作

C

Application

D

共享参数的用法

- SharedPreferences 是Android的一个**轻量级存储工具**，采用的存储结构是Key-Value的键值对方式。
- 共享参数的存储介质是符合XML规范的配置文件。保存路径是：/data/data/应用包名/shared_prefs/文件名.xml

共享参数的使用场景

➤ 共享参数主要适用于如下场合：

- 简单且孤立的数据。若是复杂且相互间有关的数据，则要保存在数据库中。
- 文本形式的数据。若是二进制数据，则要保存在文件中。
- 需要持久化存储的数据。在App退出后再次启动时，之前保存的数据仍然有效。

➤ 实际开发中，共享参数经常存储的数据有App的个性化配置信息、用户使用App的行为信息、临时需要保存的片段信息等。

实现记住密码功能

➤ 上一章的实战项目在登录页面下方有一个“记住密码”复选框，现在利用共享参数对该项目进行改造，使之实现记住密码的功能。改造的内容主要有3处：

- 声明一个共享参数对象，并在onCreate函数中调用getSharedPreferences方法获取共享参数的实例。
- 登录成功时，如果用户勾选了“记住密码”，就使用共享参数保存手机号码与密码。
- 再次打开登录页面时，App从共享参数中读取手机号码与密码，并展示在界面上。

实现“记住密码”的演示效果

chapter06

☒ 密码登录

☐ 验证码登录

手机号码：

15960238698

登录密码：

.....

忘记密码

☒ 记住密码

登 录

登录成功时保存密码

chapter06

☒ 密码登录

☐ 验证码登录

手机号码：

15960238698

登录密码：

.....

忘记密码

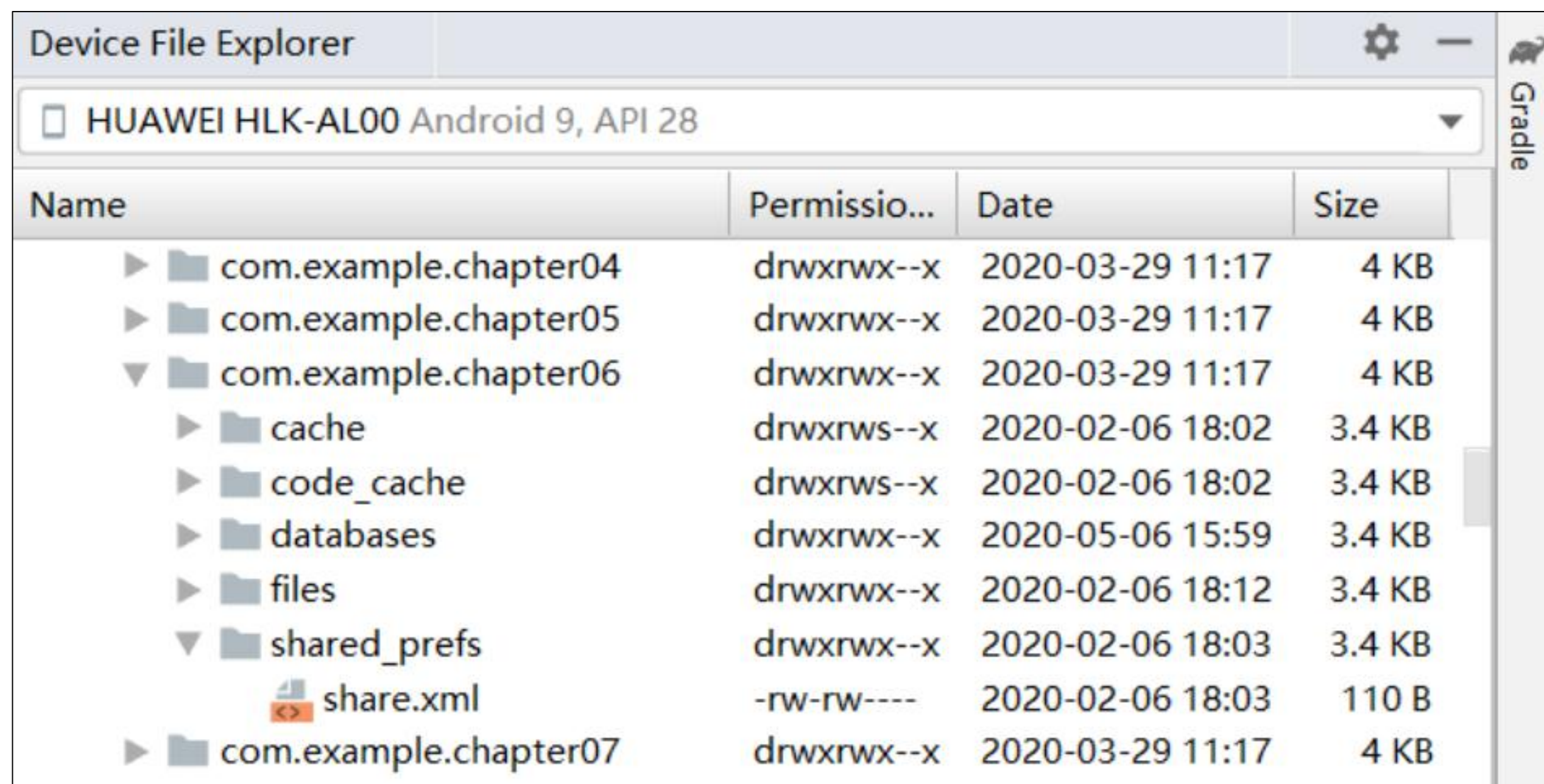
☐ 记住密码

登 录

重新进入时读取密码

利用设备浏览器寻找共享参数文件

- 共享参数的文件路径为 “/data/data/应用包名/shared_prefs/***.xml” ， 单击Android Studio右下角的竖排标签 “Device File Explorer” ， 即可打开设备文件浏览器。



Name	Permissio...	Date	Size
▶ com.example.chapter04	drwxrwx--x	2020-03-29 11:17	4 KB
▶ com.example.chapter05	drwxrwx--x	2020-03-29 11:17	4 KB
▼ com.example.chapter06	drwxrwx--x	2020-03-29 11:17	4 KB
▶ cache	drwxrws--x	2020-02-06 18:02	3.4 KB
▶ code_cache	drwxrws--x	2020-02-06 18:02	3.4 KB
▶ databases	drwxrwx--x	2020-05-06 15:59	3.4 KB
▶ files	drwxrwx--x	2020-02-06 18:12	3.4 KB
▼ shared_prefs	drwxrwx--x	2020-02-06 18:03	3.4 KB
share.xml	-rw-rw----	2020-02-06 18:03	110 B
▶ com.example.chapter07	drwxrwx--x	2020-03-29 11:17	4 KB

SQL的基本语法

- 标准的SQL语句分为三类：数据定义、数据操纵和数据控制，但不同的数据库往往有自己的实现。
- SQLite是一种小巧的嵌入式数据库，由于它属于轻型数据库，不涉及复杂的数据控制操作，因此App开发只用到数据定义和数据操纵两类SQL。
- SQLite的SQL语法与通用的SQL语法略有不同。

SQLite的数据定义语言

- 数据定义语言（DDL）描述了怎样变更数据实体的框架结构。
- DDL语言主要包括三种操作：创建表格、删除表格、修改表结构。
 - 创建表格，格式为 “CREATE TABLE IF NOT EXISTS 表格名称 (以逗号分隔的各字段定义);”
 - 删除表格，格式为 “DROP TABLE IF EXISTS 表格名称;”
 - 修改表结构，格式为 “ALTER TABLE 表格名称 修改操作;”
 - SQLite只支持增加字段，不支持修改字段，也不支持删除字段。

SQLite的数据操纵语言

- 数据操纵语言（DML）它描述了怎样处理数据实体的内部记录。
- 表格记录的操作类型包括添加、删除、修改、查询四类：
 - 添加记录，格式为 “INSERT INTO 表格名称 (以逗号分隔的字段名列表) VALUES (以逗号分隔的字段值列表);”
 - 删除记录，格式为 “DELETE FROM 表格名称 WHERE 查询条件;”
 - 修改记录，格式为 “UPDATE 表格名称 SET 字段名=字段值 WHERE 查询条件;”
 - 查询记录，格式为 “SELECT 以逗号分隔的字段名列表 FROM 表格名称 WHERE 查询条件;”

数据库管理器SQLiteDatabase

- SQLiteDatabase是SQLite的数据库管理类，它提供了若干操作数据表的API，常用的方法有3类：
 - openDatabase：打开指定路径的数据库。
 - isOpen：判断数据库是否已打开。
 - close：关闭数据库。
 - getVersion：获取数据库的版本号。
 - setVersion：设置数据库的版本号。

数据库管理器SQLiteDatabase

- 2. 事务类，用于事务层面的操作。
 - beginTransaction：开始事务。
 - setTransactionSuccessful：设置事务的成功标志。
 - endTransaction：结束事务。

数据库管理器SQLiteDatabase

➤ 3. 数据处理类，用于数据表层面的操作。

- execSQL：执行拼接好的SQL控制语句。
- delete：删除符合条件的记录。
- update：更新符合条件的记录。
- insert：插入一条记录。
- query：执行查询操作，返回结果集的游标。
- rawQuery：执行拼接好的SQL查询语句，返回结果集的游标。

数据库帮助器SQLiteOpenHelper

- SQLiteOpenHelper是Android提供的数据库辅助工具，用于指导开发者进行SQLite的合理使用。
- SQLiteOpenHelper的具体使用步骤如下：
 - 新建一个继承自SQLiteOpenHelper的数据库操作类，提示重写onCreate和onUpgrade两个方法。
 - 封装保证数据库安全的必要方法。
 - 提供对表记录进行增加、删除、修改、查询的操作方法。

游标Cursor

- 调用SQLiteDatabase的query和rawQuery方法时，返回的都是Cursor对象，因此获取查询结果要根据游标的指示一条一条遍历结果集合，Cursor的常用方法可分为3类。
- 游标控制类方法，用于指定游标的状态。
 - close：关闭游标。
 - isClosed：判断游标是否关闭。
 - isFirst：判断游标是否在开头。
 - isLast：判断游标是否在末尾。

游标Cursor

- 游标移动类方法，把游标移动到指定位置。
 - moveToFirst: 移动游标到开头。
 - moveToLast: 移动游标到末尾。
 - moveToNext: 移动游标到下一条记录。
 - moveToPrevious: 移动游标到上一条记录。
 - move: 往后移动游标若干条记录。
 - moveToPosition: 移动游标到指定位置的记录。

➤ 获取记录类方法，可获取记录的数量、类型以及取值。

- getCount: 获取结果记录的数量。
- getInt: 获取指定字段的整型值。
- getLong: 获取指定字段的长整型值。
- getFloat: 获取指定字段的浮点数值。
- getString: 获取指定字段的字符串值。
- getType: 获取指定字段的字段类型。

优化记住密码功能

- 利用共享参数实现记住密码，只能记住一个用户的登录信息，并且手机号码跟密码不存在从属关系，如果换个手机号码登录，前一个用户的登录信息就被覆盖了。
- 真正的记住密码功能是先输入手机号码，然后根据手机号匹配保存的密码，一个密码对应一个手机号码，从而实现具体手机号码的密码记忆功能。
- 运用SQLite技术分条存储不同用户的登录信息，并提供根据手机号码查找登录信息的方法，这样可以同时记住多个手机号码的密码。

利用SQLite记住密码的步骤

➤ 利用SQLite记住密码有以下三处改造：

- 声明一个用户数据库的帮助器对象，然后在活动页面的onResume方法中打开数据库连接，在onPasue方法中关闭数据库连接。
- 登录成功时，如果用户勾选了“记住密码”，就使用数据库保存手机号码与密码在内的登录信息。
- 再次打开登录页面，用户输入手机号完毕后点击密码输入框时，App到数据库中根据手机号查找登录记录，并将记录结果中的密码填入密码框。

优化 “记住密码” 的演示效果

chapter06

☒ 密码登录 ☐ 验证码登录

手机号码: 15960238696

登录密码: 请输入密码 忘记密码

☐ 记住密码

登 录

输入11位手机号码

chapter06

☒ 密码登录 ☐ 验证码登录

手机号码: 15960238696

登录密码: 忘记密码

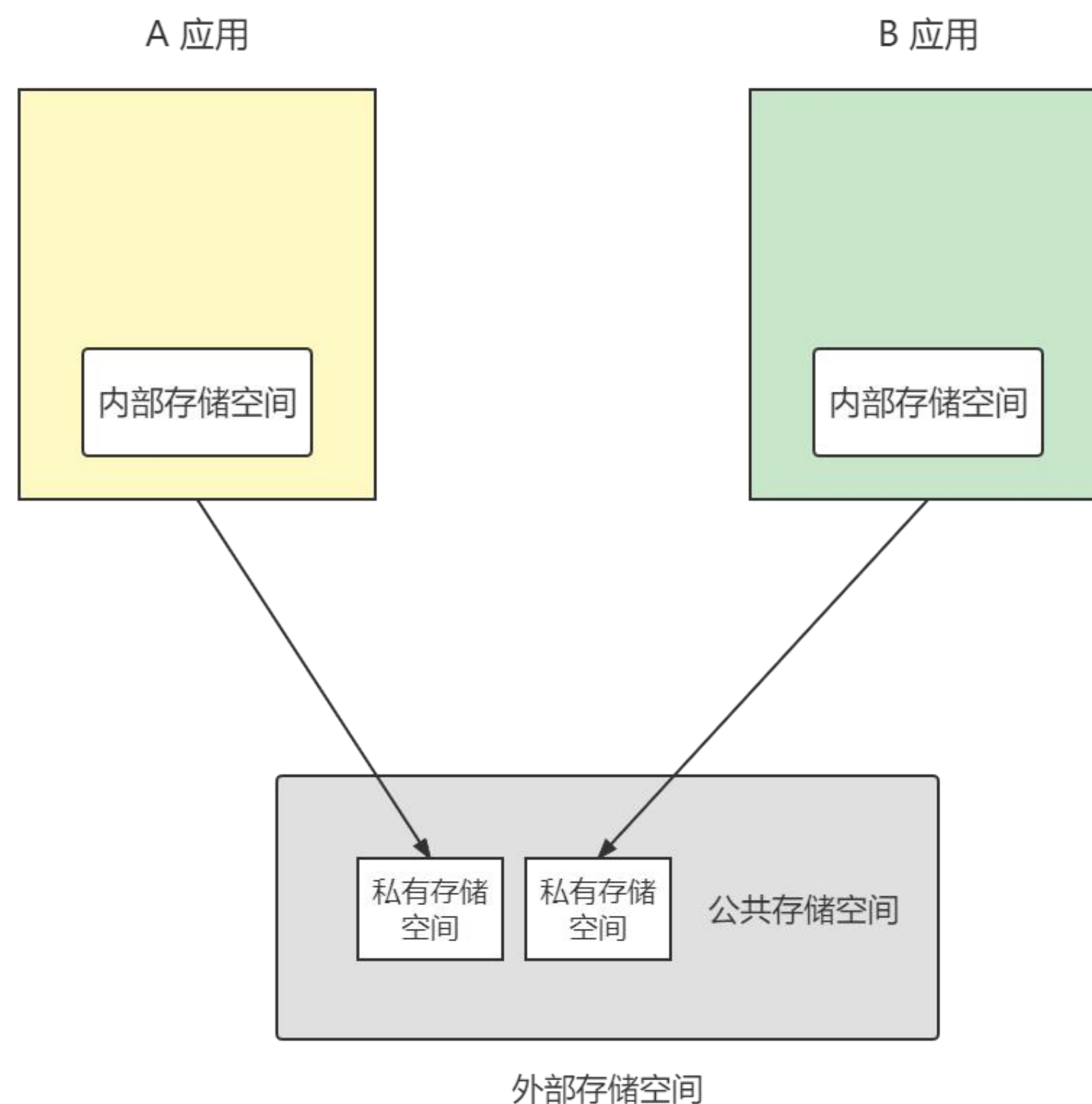
☐ 记住密码

登 录

光标跳到密码输入框

私有存储空间与公共存储空间

- Android把外部存储分成了两块区域，一块是所有应用均可访问的公共空间，另一块是只有应用自己才可访问的私有空间。



获取公共空间和私有空间的存储路径

- 获取公共空间的存储路径，调用的是Environment类的getExternalStoragePublicDirectory方法；获取应用私有空间的存储路径，调用的是getExternalFilesDir方法。

在存储卡上读写文本文件

- 文本文件的读写一般借助于 `FileOutputStream` 和 `FileInputStream`。
 - `FileOutputStream`用于写文件。
 - `FileInputStream`用于读文件。

读写文本文件的演示效果

chapter06

姓名：李四

年龄：20

身高：160

体重：50

☐ 已婚

保存文本到存储卡

用户注册信息文件的保存路径为：
/storage/emulated/0/Android/data/
com.example.chapter06/files/Download/
20200516164109.txt

将注册信息保存到文本文件

chapter06

删除所有文本文件

找到最新的文本文件，路径为/
storage/emulated/0/Android/data/
com.example.chapter06/files/Download/
20200516164109.txt，内容如下：
姓名：李四
年龄：20
身高：160cm
体重：50kg
婚否：未婚
注册时间：2020-05-16 16:41:09

从文本文件读取注册信息

在存储卡上读写图片文件

- Android 的位图工具是Bitmap，App读写Bitmap可以使用性能更好的BufferedOutputStream和BufferedInputStream。
- Android还提供了BitmapFactory工具用于读取各种来源的图片，相关方法如下：
 - decodeResource：该方法可从资源文件中读取图片信息。
 - decodeFile：该方法可将指定路径的图片读取到Bitmap对象。
 - decodeStream：该方法从输入流中读取位图数据。

读写图片文件的演示效果

chapter06



把资源图片保存到存储卡

图片文件的保存路径为：
/storage/emulated/0/Android/data/
com.example.chapter06/files/Download/
20200528165934.jpeg

把资源图片保存到存储卡

chapter06

删除所有图片文件

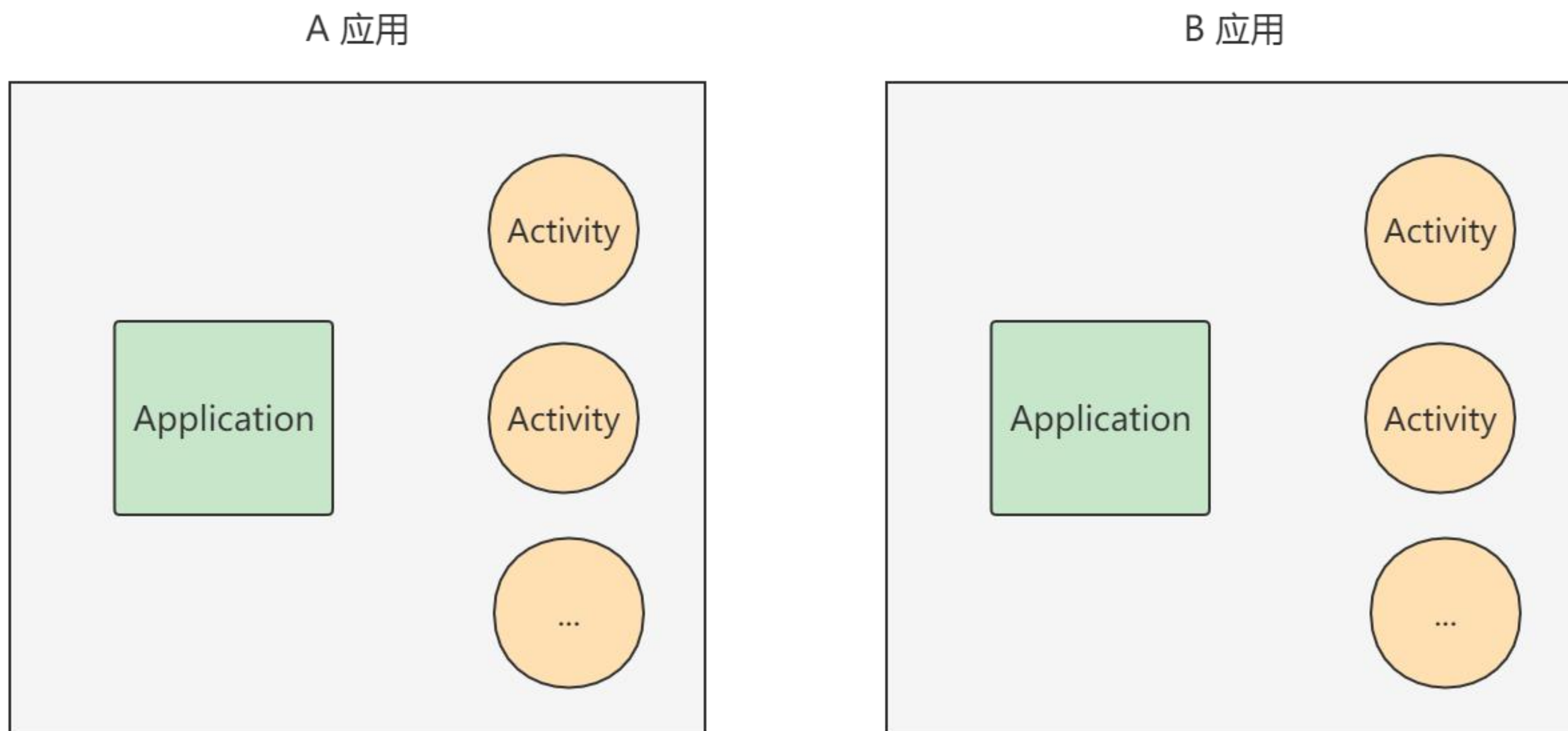
找到最新的图片文件，路径为/
storage/emulated/0/Android/data/
com.example.chapter06/files/Download/
20200528165934.jpeg



从存储卡读取图片文件

Application的生命周期

- Application是Android的一大组件，在App运行过程中有且仅有一个Application对象贯穿整个生命周期。



利用Application操作全局变量

- 全局的意思是其他代码都可以引用该变量，因此全局变量是共享数据和消息传递的好帮手。
- 适合在Application中保存的全局变量主要有下面3类数据：
 - 会频繁读取的信息，如用户名、手机号等。
 - 不方便由意图传递的数据，例如位图对象、非字符串类型的集合对象等。
 - 容易因频繁分配内存而导致内存泄漏的对象，如Handler对象等。

全局变量的实现

➤ Java代码可利用自定义Application的静态成员变量实现全局变量的功能。具体需要完成以下3项工作：

- 写一个继承自Application的类MainApplication。该类要采用单例模式，内部声明自身类的一个静态成员对象，然后提供该静态对象的获取方法getInstance。
- 在Activity中调用MainApplication的getInstance方法，获得MainApplication的一个静态对象，通过该对象访问MainApplication的公共变量和公共方法。
- 不要忘了在AndroidManifest.xml中注册新定义的Application类名，即在application节点中增加android:name属性，值为“.MainApplication”。

利用Room简化数据库操作

- 使用数据库帮助器编码的时候，开发者每次都得手工实现以下代码逻辑：
 - 重写数据库帮助器的onCreate方法，添加该表的建表语句；
 - 在插入记录之时，必须将数据实例的属性值逐一赋给该表的各字段；
 - 在查询记录之时，必须遍历结果集游标，把各字段值逐一赋给数据实例；
 - 每次读写操作之前，都要先开启数据库连接；读写操作之后，又要关闭数据库连接；

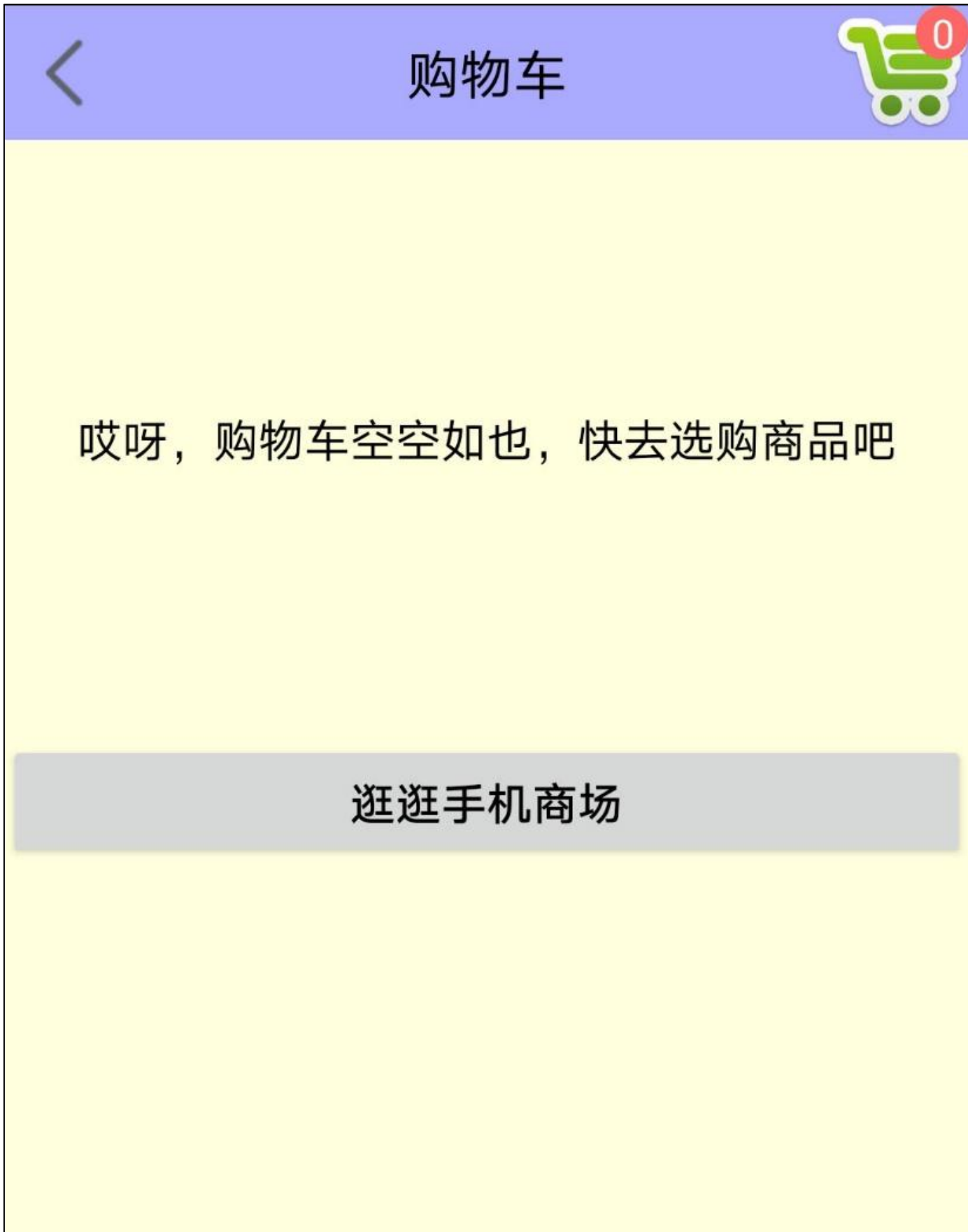
Room框架的导入

- Room是谷歌公司推出的数据库处理框架，该框架同样基于SQLite，但它通过注解技术极大简化了数据库操作，减少了原来相当一部分编码工作量。
- 在使用Room之前，要先修改模块的build.gradle文件，往dependencies节点添加下面两行配置，表示导入指定版本的Room库：
 - implementation 'androidx.room:room-runtime:2.2.5'
 - annotationProcessor 'androidx.room:room-compiler:2.2.5'

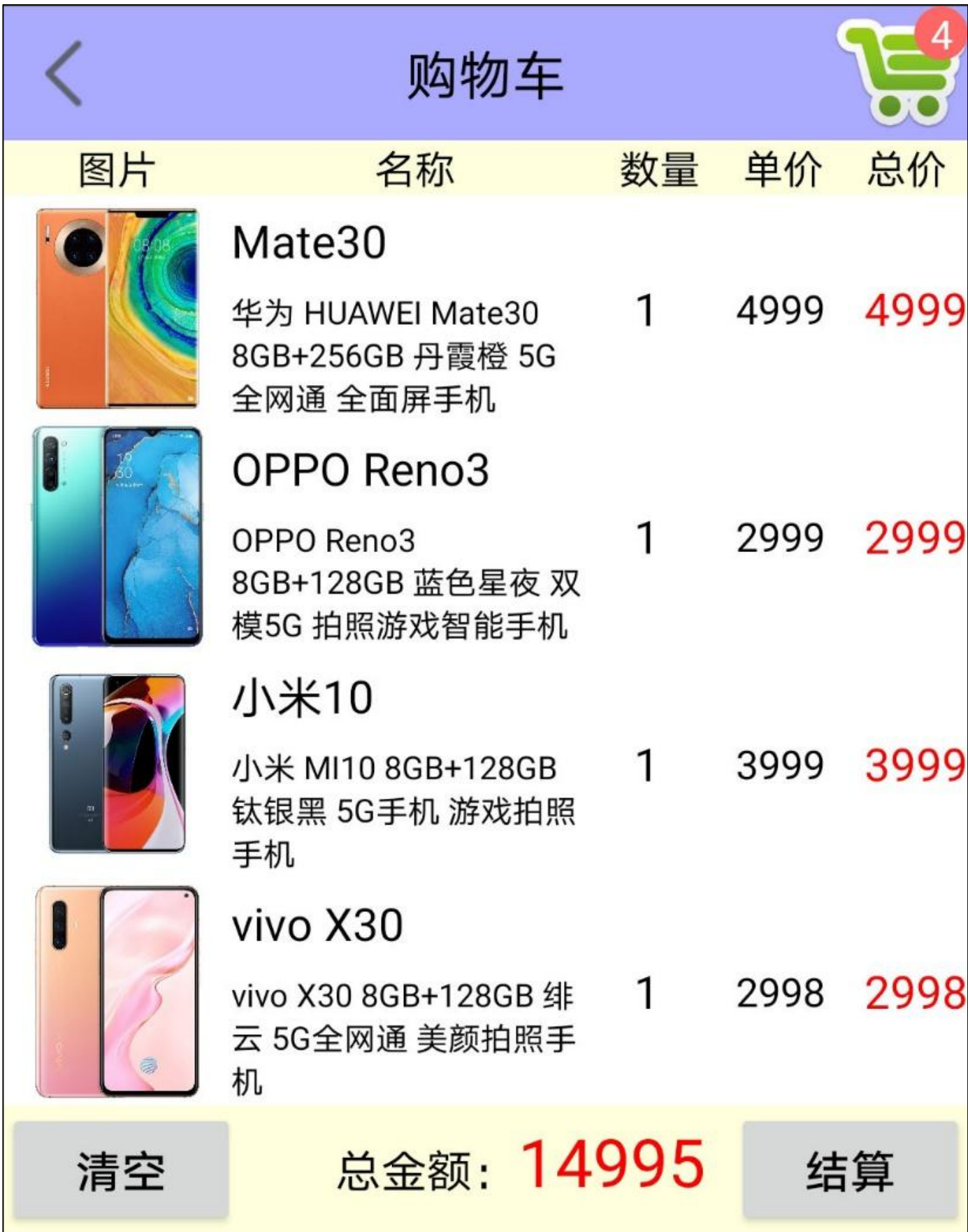
Room框架的编码步骤

- 以录入书籍信息为例，使用Room框架的编码过程分为下列五步：
 - 编写书籍信息表对应的实体类，该类添加 “@Entity” 注解。
 - 编写书籍信息表对应的持久化类，该类添加 “@Dao” 注解。
 - 编写书籍信息表对应的数据库类，该类从RoomDatabase派生而来，并添加 “@Database” 注解。
 - 在自定义的Application类中声明书籍数据库的唯一实例。
 - 在操作书籍信息表的地方获取数据表的持久化对象。

实战项目：购物车



购物车的初始界面



添加商品后的购物车

购物车的功能要求

- 购物车存放着用户准备购买的商品，一开始是空的，随着商品被加入购物车，购物车中就会显示已添加的商品列表。
- 除了购物车页面，其它页面（如商场频道页面、商品详情页面），都可能在右上角或者右下角找到购物车图标。购物车图标上会显示已添加的商品数量，且商品数量是实时更新的。
- 购物车页面、商场频道页面、商品详情页面多处都会显示商品的小图或者大图，如何迅速且高效地加载图片是个需要研究的课题。

界面设计

- 线性布局LinearLayout: 购物车界面从上往下排列, 用到了垂直方向的线性布局。
- 网格布局GridLayout: 商场页面的陈列橱柜, 允许分行分列展示商品。
- 相对布局RelativeLayout: 页面右上角的购物车图标, 图标右上角又有数字标记, 按照指定方位排列控件正是相对布局的拿手好戏。
- 其他常见控件尚有文本视图TextView、图像视图ImageView, 按钮控件Button等。

购物车用到的存储技术

- 数据库SQLite：最直观的是数据库，购物车里的商品列表一定放在SQLite中，增删改查都少不了它。
- 全局内存：购物车图标右上角的数字表示购物车中的商品数量，该数值建议保存在全局内存中，这样不必每次都到数据库中执行count操作。
- 存储卡文件：App把下载的商品图片保存在存储卡中，这样下次就能直接从存储卡获取商品图片，加快浏览速度。
- 共享参数SharedPreferences：是否首次访问网络图片，这个标志位推荐放在共享参数中，因为它需要持久化存储，并且只有一个参数信息。