**Android：**

**6.数据持久化的三种储存方式：**

1. **文件存储**

（1）方法介绍：

android中的context类提供了一个openFileOutPut（）的方法，可以将数据存储到指定文件中。该方法接收两个参数，第一个是文件名，第二个是文件操作模式。

对于文件名，注意不可以包含路径，因为所有文件都是默认存储到指定目录下的（data/data/<package name>/files/）

对于第二个参数，主要选择两种：

MODE\_PRIVATE:该方式为默认操作方式，表示指定同样文件名时，所写入的内容会覆盖原文件中的内容

MODE\_APPEND:如果该文件已存在，则往文件里追加内容。不存在则创建新文件

此外，MODE\_WORLD\_READAVLE和MODE\_WORLD\_WRITEABLE两种模式允许其他程序对我们程序中文件进行读写操作，由于这两种模式过于危险易引起安全性问题，已在Android4.2中被废弃

1. 操作流程：

通过openFileOutPut（）的方法得到一个FileOutputStream对象，再借助它构造一个OutputStreamWriter对象，再使用OutputStreamWriter构建一个BufferedWriter对象，就完成了将文本写入文件中的操作了

1. 适用范围：

适用于存储一些见到的文本数据或二进制数据

·(4)文件中读取数据：

1. 方法介绍：

与上面类似，context类提供了一个openFileInput（）方法，用于从文件中读取数据。该方法只接受一个参数，即所读取文件的文件名,系统会自动到指定目录下加载数据，返回一个FileInputStream对象，之后通过java流将数据读取出来（data/data/<package name>/files/）

1. 操作流程：

通过openFileInput（）方法获得一个FileInputStream对象，然后借助它构建出一个InputStreamReader对象，接着再使用InputStreamReader构建出一个BufferedReader对象，通过BufferedReader将内容读取并存放在一个StringBuilder，最后将内容返回就可以额

·（5）总结：

文件存储的核心技术：

context类中的两个方法：

openFileInput（）方法：读取

openFileOutput（）方法：存储

之后再利用java的各种流进行读写操作

1. **Sharedpreference存储**

（1）方法介绍：

通过键值对的方式存储数据且支持多种不同数据类型存储。

对于获取SharedPreferences对象有三种方法：

[1].context类中getSharedPreference（）方法

该方法接受两个参数，第一个参数用于指定SharedPreferences文件名称，如果指定文件不存在则创建一个，第二个参数用于指定操作模式，目前只有MODE\_PRIVATE一种模式可选，也是默认的操作模式，表示只有当前应用程序才能对这个sharedPreference文件进行读写，MODE\_WORLD\_READAVLE和MODE\_WORLD\_WRITEABLE和MODE\_MULTI\_PROCESS模式都已废弃

[2].Activity中的getSharedPreference（）方法

该方法与上述[1]类似，但它只接受一个操作模式参数，因为使用这个方法会将当前活动的类名作为SharedPreferences的文件名

[3].PreferenceManager类中getDefaultSharedPreference（）方法

这是一个静态方法，它接收一个context参数并自动使用当前应用程序的包作为前缀来命名sharedPreference文件

对于添加数据，添加什么类型数据就调用editor中哪一类方法，如putString（）中为字符串型数据，putInt（）为int型数据

（2）操作流程：

[1].获取一个SharedPreferences对象

[2].调用SharedPreferences中的edit（）方法获取一个SharedPreferences.Editor对象

[3].向SharedPreferences.Editor对象中添加数据

[4].调用apply（）方法将添加的数据提交，完成数据存储操作

1. 适用范围：

适用于保存键值对型数据

·（4）sharedPreference读取数据

sharedPreference读取数据非常的简单，实例化一个sharedPreference对象后，直接调用其中的getInt，getString等方法来读取相应类型的数据（类似上述添加数据的put方法），get类方法接收两个参数，第一个是键，通过存储数据时使用的键来得到对应的值，第二个参数是默认值，所传入的键找不到对应类型的值时的默认值

1. **数据库存储**
2. 方法介绍：

android内置了轻量级的关系型数据库SQlite数据库，开发者可直接借助SQLiteOpenHelper类对数据库进行创建和升级。

SQLiteOpenHelper中有两个构造函数可供重写，一般使用参数较少那个，这个构造方法接收四个参数，第一个是context，固定的，第二个是数据库名，第三个参数允许我们查询数据时返回一个自定义的Cursor，一般传入null，第四个参数表示当前数据库版本号，可用于数据库升级操作

在创建的帮助类中先定义我们的所要建的表，然后在onCreate（）方法中调用SQLiteDatabase的execSQL（）方法执行就完成了建库同时建表的操作

对于SQLiteOpenHelper实例中getWritableDatabase（）方法和getReadableDatabase（）方法的区别在于，当数据库不可写入时，getReadableDatabase（）方法返回的对象将以只读形式去打开数据库，而getWritableDatabase（）方法将出现异常

1. 操作流程：

创建一个帮助类继承抽象类SQLiteOpenHelper并重写oncreate（）和onUpgrade（）以及其中的构造方法，再之后构建出SQLiteOpenHelper实例后调用getReadableDatabase（）或getWritableDatabase（）方法就能创建数据库了，此时重写的onCreate（）也会得到执行，再之后进行常规的数据库增删查改即可

1. 适用范围：

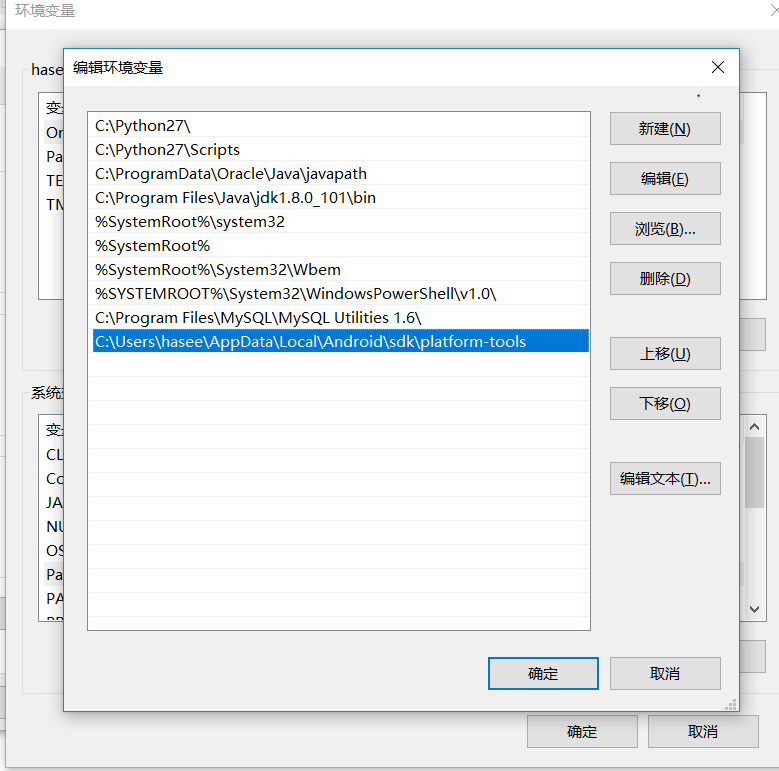
适用于存储大量复杂的关系型数据

·（4）adb工具：

由于所创建的数据库文件（db后缀）无法用File Explorer打开，为了确认我们建表成功，我们可以通过adb shell来对数据库和表的创建情况进行检查。

Adb工具是Android SDK自带的一个测试工具，使用该工具可直接对连接在电脑上的手机或模拟器进行调试，它存放在SDK的platform-tools目录下，想在命令行使用这个工具则要配置环境变量。

由于linux和mac较少，这里仅讨论Windows系统，右击计算机-属性-高级系统设置-环境变量，然后找到Path，将其值改为platform-tools目录的完整路径



然后打开命令行界面，输入adb shell进入控制台，再cd /data/data/com.example.工程名/databases进入到目录下，再ls查看目录文件，输入sqlite3 数据库名 进入到数据库，可以查看到该数据库下所建的表就确认了（此处非重点，不赘述）

·（5）升级数据库

当我们想对所建立的数据库内的表进行增加或是删除，上述所提到的onUpgrade（）方法就起到了作用。以添加表为例，首先，在onCreate（）方法中照葫芦画瓢，先在帮助类中定义所想添加的表的内容，并在onCreate（）方法中一样调用execSQL（）方法建表，由于所在数据库已经存在，onCreate（）方法不会再次执行，那么我们就可以在onUpgrade（）方法中用db调用execSQL（）进行drop的删除原有表操作，然后再添加onCreate（db）重新调用上面的onCreate（）方法实现数据库的更新。记得将前面的构造函数的第四个参数改为大于1的数让onUpgrade（）方法得以执行哦

·（6）对数据库中的表的数据进行增删查改

四种操作大同小异，都是通过调用SQLiteDatabase中的方法（以实例db为例）实现，下面一一简单列举讲解（查询操作仅列最简单的一个）

增（添加数据）：

db.insert(),接收三个参数，第一个参数表名，即目标表名，第二个参数用于在未指定添加数据的情况下给某些可为空的列自动赋值null，一般用不到这个功能传null即可，第三个参数是一个ContentValues对象，它提供一系列put（）方法重载用于向该对象添加数据，只需要将每个列名及相应待添加数据传入即可。

删（删除数据）：

db.delete(),接收三个参数，第一个表名，第二第三个约束条件

查（查询数据）：

db.query（），接收七个参数，第一个表名，第二个查询目标列名（不指定默认所有列），第三第四约束条件（不指定默认所有行），第五个参数是对查询结果以哪一列进行排列（group by对象），第六个参数是对group by后含有什么或值为什么进行进一步过滤（having），第七个参数是对最终结果进行排列的方式，以上三个不指定即默认不进行group by操作，having操作及默认的排序方式

改（修改数据）：

db.update(),接收四个参数，第一个参数表名，第二个contentValues对象名，把更新的数据组装在这里，第三个第四个用于约束更新某一行或某几行中的数据，不指定则默认更新所有行。第三行第四行例如name=？，new String[]{“name的名字”}，即可实现对name这一行的数据更改。