**1、请介绍下 ContentProvider 是如何实现数据共享的**在 Android 中如果想将自己应用的数据（一般多为数据库中的数据）提供给第三发应用，那么我们只能通过 ContentProvider 来实现了。  
ContentProvider 是应用程序之间共享数据的接口。使用的时候首先自定义  
一个类继承 ContentProvider，然后覆写 query、insert、update、delete 等  
方法。因为其是四大组件之一因此必须在 AndroidManifest 文件中进行注册。  
把自己的数据通过 uri 的形式共享出去  
android 系统下 不同程序 数据默认是不能共享访问  
需要去实现一个类去继承 ContentProvider  
public class PersonContentProvider extends ContentProvider{  
public boolean onCreate(){  
}  
query(Uri, String[], String, String[], String)  
insert(Uri, ContentValues)  
update(Uri, ContentValues, String, String[])  
delete(Uri, String, String[])  
}  
<provider  
android:exported=*"true"*  
android:name=*"com.itheima.contenProvider.provider.PersonContentPro*  
*vider"*android:authorities=*"com.itheima.person"* />  
第三方可以通过 ContentResolver 来访问该 Provider。

**2、请介绍下 Android 的数据存储方式**a. File 存储  
b. SharedPreference 存储  
c. ContentProvider 存储  
d. SQLiteDataBase 存储  
传智播客武汉校区就业部出品 务实、创新、质量、分享、专注、责任  
e. 网络存储

**3、为什么要用 ContentProvider？它和 sql 的实现上有什  
么差别？**  
ContentProvider 屏蔽了数据存储的细节,内部实现对用户完全透明,用户只  
需要关心操作数据的 uri 就可以了，ContentProvider 可以实现不同 app 之间  
共享。  
Sql 也有增删改查的方法，但是 sql 只能查询本应用下的数据库。而  
ContentProvider 还可以去增删改查本地文件. xml 文件的读取等。

**4、说 说 ContentProvider 、 ContentResolver 、**

**ContentObserver 之间的关系**a. ContentProvider 内容提供者，用于对外提供数据  
b. ContentResolver.notifyChange(uri)发出消息  
c. ContentResolver 内容解析者，用于获取内容提供者提供的数据  
d. ContentObserver 内容监听器，可以监听数据的改变状态  
e. ContentResolver.registerContentObserver()监听消息。

**5、ListView 如何提高其效率？**

当 convertView 为空时，用 setTag()方法为每个 View 绑定一个存放控件的ViewHolder 对象。 当 convertView 不为空， 重复利用已经创建的 view 的时候， 使用 getTag()方法获取绑定的 ViewHolder 对象，这样就避免了 findViewById 对控件的层层查询，而是快速定位到控件。  
① 复用 ConvertView，使用历史的 view，提升效率 200%  
② 自定义静态类 ViewHolder，减少 findViewById 的次数。提  
升效率 50%  
③ 异步加载数据，分页加载数据。  
④ 使用 WeakRefrence 引用 ImageView 对象

**6、当 ListView 数据集改变后，如何更新 ListView**

使用该 ListView 的 adapter 的 notifyDataSetChanged()方法。该方法会 使 ListView 重新绘制。

**7、ListView 如何实现分页加载**

① 设 置 ListView 的 滚 动 监 听 器 ： setOnScrollListener(new OnScrollListener{….})

在 监 听 器 中 有 两 个 方 法 ：

滚 动 状 态 发 生 变 化 的 方 法 (onScrollStateChanged)和 listView 被滚动时调用的方法(onScroll)

② 在滚动状态发生改变的方法中，有三种状态：

手指按下移动的状态： SCROLL\_STATE\_TOUCH\_SCROLL: // 触 摸滑动

静止状态： SCROLL\_STATE\_IDLE: // 静止

对不同的状态进行处理： 分批加载数据，只关心静止状态：关心最后一个可见的条目，如果最后一个 可见条目就是数据适配器（集合）里的最后一个，此时可加载更多的数据。在每 次加载的时候，计算出滚动的数量，当滚动的数量大于等于总数量的时候，可以 提示用户无更多数据了。

**8、ListView 可以显示多种类型的条目吗**

可以，ListView 显示的每个条目都是通过 baseAdapter 的 getView(int position, View convertView, ViewGroup parent)来展示的，理 论上我们完全可以让每个条目都是不同类型的 view。 比如：从服务器拿回一个标识为 id=1,那么当 id=1 的时候，我们就加载类 型一的条目，当 id=2 的时候，加载类型二的条目。常见布局在资讯类客户端中 可以经常看到。 除 此 之 外 adapter 还 提 供 了 getViewTypeCount （ ） 和 getItemViewType(int position)两个方法。在 getView 方法中我们可以根据不 同的 viewtype 加载不同的布局文件。

**9、ListView 如何定位到指定位置**

可以通过 ListView 提供的 lv.setSelection(listView.getPosition());方法。

**10、如何在 ScrollView 中如何嵌入 ListView**

通常情况下我们不会在 ScrollView 中嵌套 ListView，但是如果面试官非让 我嵌套的话也是可以的。 在 ScrollView 添加一个 ListView 会导致 listview 控件显示不全，通常只会 显示一条，这是因为两个控件的滚动事件冲突导致。所以需要通过 listview 中的 item 数量去计算 listview 的显示高度，从而使其完整展示，如下提供一个方法 供大家参考。  
lv = (ListView) findViewById(R.id.*lv*);  
adapter = **new** MyAdapter();  
lv.setAdapter(adapter);  
setListViewHeightBasedOnChildren(lv);  
----------------------------------------------------  
**public void** setListViewHeightBasedOnChildren(ListView listView) {  
ListAdapter listAdapter = listView.getAdapter();  
**if** (listAdapter == **null**) {  
**return**;  
}  
**int** totalHeight = 0;  
**for** (**int** i = 0; i < listAdapter.getCount(); i++) {  
View listItem = listAdapter.getView(i, **null**, listView);  
listItem.measure(0, 0);  
totalHeight += listItem.getMeasuredHeight();  
}  
ViewGroup.LayoutParams params = listView.getLayoutParams();  
params.height = totalHeight + (listView.getDividerHeight() \*  
(listAdapter.getCount() - 1));  
params.height += 5;// if without this statement,the listview will be a  
// little short  
listView.setLayoutParams(params);  
}

现阶段最好的处理的方式是： 自定义 ListView，重载 onMeasure()方法，设置 全部显示。

package com.meiya.ui;  
import android.widget.ListView;  
/\*\*  
\*\*  
@Description: scrollview 中内嵌 listview 的简单实现  
\*\*  
@File: ScrollViewWithListView.java  
\*\*\*  
@Version  
\*/  
public class ScrollViewWithListView extends ListView {  
public ScrollViewWithListView(android.content.Context context,  
android.util.AttributeSet attrs) {  
super(context, attrs);  
}  
/\*\*  
\* Integer.MAX\_VALUE >> 2,如果不设置，系统默认设置是显示两条  
\*/  
public void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {  
int expandSpec = MeasureSpec.makeMeasureSpec(Integer.MAX\_VALUE >> 2,  
MeasureSpec.AT\_MOST);  
super.onMeasure(widthMeasureSpec, expandSpec);  
} }

**11、ListView 中如何优化图片**

图片的优化策略比较多。

1、处理图片的方式：

如果 ListView 中自定义的 Item 中有涉及到大量图片的，一定要对图片进行 细心的处理，因为图片占的内存是 ListView 项中最头疼的，处理图片的方法大 致有以下几种：

①、不要直接拿路径就去循环 BitmapFactory.decodeFile;使用 Options 保 存图片大小、不要加载图片到内存去。

②、对图片一定要经过边界压缩尤其是比较大的图片，如果你的图片是后台 服务器处理好的那就不需要了

③、在 ListView 中取图片时也不要直接拿个路径去取图片，而是以 WeakReference （ 使 用 WeakReference 代 替 强 引 用 。 比 如 可 以 使 用 WeakReference mContextRef）、SoftReference、WeakHashMap 等的来存 储图片信息。

④、在 getView 中做图片转换时，产生的中间变量一定及时释放

2、异步加载图片基本思想：

1）、 先从内存缓存中获取图片显示（内存缓冲）

2）、获取不到的话从 SD 卡里获取（SD 卡缓冲）

3）、都获取不到的话从网络下载图片并保存到 SD 卡同时加入内存并显示 （视情况看是否要显示）

原理： 优化一：先从内存中加载，没有则开启线程从 SD 卡或网络中获取，这里注意从 SD 卡获取图片是放在子线程里执行的，否则快速滑屏的话会不够流畅。

优化二：于此同时，在 adapter 里有个 busy 变量，表示 listview 是否处于 滑动状态，如果是滑动状态则仅从内存中获取图片，没有的话无需再开启线程去外存或网络获取图片。

优化三：ImageLoader 里的线程使用了线程池，从而避免了过多线程频繁 创建和销毁，如果每次总是 new 一个线程去执行这是非常不可取的，好一点的 用的 AsyncTask 类，其实内部也是用到了线程池。在从网络获取图片时，先是 将其保存到 sd 卡，然后再加载到内存，这么做的好处是在加载到内存时可以做 个压缩处理，以减少图片所占内存。

**12、ListView 中图片错位的问题是如何产生的**

图片错位问题的本质源于我们的 listview 使用了缓存 convertView，假设一 种场景，一个 listview 一屏显示九个 item，那么在拉出第十个 item 的时候，事 实上该 item 是重复使用了第一个 item，也就是说在第一个 item 从网络中下载 图片并最终要显示的时候，其实该 item 已经不在当前显示区域内了，此时显示 的后果将可能在第十个 item 上输出图像，这就导致了图片错位的问题。所以解

**13、如何刷新 ListView 中单个 item 的数据，不刷新整个  
ListView 的数据？**

修改单个 Item 的数据,然后调用适配器的 notifyDataSetChanged()方法

**14、adapter 有什么作用？常见的 Adapter 有哪些？**Adapter 是连接后端数据和前端显示的适配器接口。常见的 Adapter 有

SimpleAdapter，ArrayAdapter，CursorAdapter，Simple CursorAdapter，BaseAdapter，HeaderViewListAdapter，ListAdapter，ResourceCursorAdapter，SpinnerAdapter，WrapperListAdapter 等

**15、ListView 的优化方案**

1，如果自定义适配器，那么在 getView 方法中要考虑方法传进来的参数 contentView 是否  
为 null，如果为 null 就创建 contentView 并返回，如果不为 null 则直接使用。在这个方  
法中，尽可能少创建 view。  
2，给 contentView 设置 tag(setTag()),传入一个 viewHolder 对象，用于缓存要显示的数  
据，可以达到图像数据异步加载的效果  
3，如果 listview 需要显示很多 Item，采用分页加载+异步方式，  
（4） 快速滑动时不加载图片  
（5） 如果自定义的 item 中有图片，需要处理图片（减少图片所占内存）

**16、如何将 SQLite 数据库(dictionary.db 文件)与 apk 文件一起发布?**  
解答：可以将 dictionary.db 文件复制到 Eclipse Android 工程中的 res raw 目录中。  
所有在 res raw 目录中的文件不会被压缩，这样可以直接提取该目录中的文件。

**17、如何将打开 res raw 目录中的数据库文件?**在 Android 中不能直接打开 res raw 目录中的数据库文件，而需要在程序第一次启动时将该文件复制到手机内存或 SD 卡的某个目录中，然后再打开该数据库文件。复制的基 本 方 法 是 使 用 getResources().openRawResource 方 法 获 得 res aw 目 录 中 资 源 的InputStream 对象，然后将该 InputStream 对象中的数据写入其他的目录中相应文件中。在Android SDK 中可以使用 SQLiteDatabase.openOrCreateDatabase 方法来打开任意目录中的SQLite 数据库文件。

**18、请继承 SQLiteOpenHelper 实现：**

1）.创建一个版本为 1 的“diaryOpenHelper.db”的数据库，  
2）.同时创建一个 “diary” 表（包含一个\_id 主键并自增长，topic 字符型 100  
长度， content 字符型 1000 长度）  
3）.在数据库版本变化时请删除 diary 表，并重新创建出 diary 表。  
public class DBHelper extends SQLiteOpenHelper {  
public final static String DATABASENAME = "diaryOpenHelper.db";  
public final static int DATABASEVERSION = 1;  
//创建数据库  
public DBHelper(Context context,String name,CursorFactory factory,int  
version)  
{  
super(context, name, factory, version);  
}  
//创建表等机构性文件  
public void onCreate(SQLiteDatabase db)  
{

String sql ="create table diary"+  
"("+  
"\_id integer primary key autoincrement,"+  
"topic varchar(100),"+  
"content varchar(1000)"+  
")";  
db.execSQL(sql);  
}  
//若数据库版本有更新，则调用此方法  
public void onUpgrade(SQLiteDatabase db,int oldVersion,int newVersion)  
{  
String sql = "drop table if exists diary";  
db.execSQL(sql);  
this.onCreate(db);  
}  
}

**19、Android 如何访问 SharedPreferences 中的数据**  
访问 SharedPreferences 中的数据代码如下：  
SharedPreferences sharedPreferences = getSharedPreferences( “ itcast ” ,  
Context.MODE\_PRIVATE);

//getString()第二个参数为缺省值，如果 preference 中不存在该 key，将返回缺省值  
String name = sharedPreferences.getString(“name”, “”);  
int age = sharedPreferences.getInt(“age”, 1);  
如 果 访 问 其 他 应 用 中 的 Preference ， 前 提 条 件 是 ： 该 preference 创 建 时 指 定 了  
Context.MODE\_WORLD\_READABLE 或 者 Context.MODE\_WORLD\_WRITEABLE 权 限 。 如 ： 有 个<package name>为 cn.itcast.action 的应用使用下面语句创建了 preference。  
getSharedPreferences(“itcast”, Context.MODE\_WORLD\_READABLE);  
其他应用要访问上面应用的 preference，首先需要创建上面应用的 Context，然后通过  
Context 访问 preference ，访问 preference 时会在应用所在包下的 shared\_prefs 目录找  
到 preference ：  
Context otherAppsContext = createPackageContext( “ cn.itcast.action ” ,  
Context.CONTEXT\_IGNORE\_SECURITY);  
SharedPreferences sharedPreferences =  
otherAppsContext.getSharedPreferences(“tsing”, Context.MODE\_WORLD\_READABLE);  
String name = sharedPreferences.getString(“name”, “”);  
int age = sharedPreferences.getInt(“age”, 0);  
如果不通过创建 Context 访问其他应用的 preference，可以以读取 xml 文件方式直接访问  
其他应用 preference 对应的 xml 文件，如：  
File xmlFile =  
new File(“/data/data/<package name>/shared\_prefs/tsing.xml”);//<package name>  
应替换成应用的包名