

安卓开发期中复习

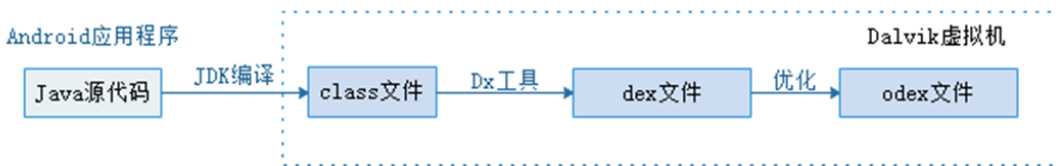
作者: cqg

版本: 2024-05-16

简答题

1. 简述Android源代码的编译过程★

Java源文件经过JDK编译器编译为class文件后，Dalvik虚拟机中的Dx工具会将部分class文件转换成dex文件，dex文件还会在Dalvik虚拟机中进一步优化成odex文件



2. 简述Android体系结构包含的层次及各层的特点★

Android系统架构从高到低分为四层，分别为依次是应用程序层、应用程序框架层、核心类库和Linux内核，各层的特点具体如下：

1. 应用程序层：一个核心应用程序的集合，安装在手机中的应用程序都属于这一层。
2. 应用程序架构层：主要提供了构建应用程序时用到的各种API。例如活动管理器（Activity Manager）。
3. 核心类库：主要包含了系统库和Android运行环境。
4. Linux内核：它为Android设备的各种硬件提供了底层的驱动，如显示驱动、音频驱动。

3. 简述ListView控件与RecyclerView控件的区别

1. 展示效果：RecyclerView控件可以通过LayoutManager类实现横向或竖向的列表效果、瀑布流效果和GridView效果，而ListView控件只能实现竖直的列表效果。
2. 适配器：RecyclerView控件使用的是RecyclerView.Adapter适配器，该适配器将BaseAdapter中的getView()方法拆分为onCreateViewHolder()方法和onBindViewHolder()方法，强制使用ViewHolder类，使代码编写规范化，避免了初学者写的代码性能不佳。
3. 复用效果：RecyclerView控件复用Item对象的工作由该控件自己实现，而ListView控件复用Item对象的工作需要开发者通过convertView的setTag()方法和getTag()方法进行操作。
4. 动画效果：RecyclerView控件可以通过setItemAnimator()方法为Item添加动画效果，而ListView控件不可以通过该方法为Item添加动画效果。

4. 简述实现Button控件的点击事件的方式★

1. 在布局文件中指定onClick属性的方式设置点击事件。
2. 使用匿名内部类的方式设置点击事件。
3. 通过为Activity实现OnClickListener接口的方式设置点击事件。

5. 简述Activity的生命周期的方法及什么时候被调用★

1. onCreate(): Activity创建时调用, 通常做一些初始化设置。
2. onStart(): Activity即将可见时调用。
3. onResume(): Activity获取焦点时调用。
4. onPause(): 当前Activity被其他Activity覆盖或屏幕锁屏时调用。
5. onStop(): Activity对用户不可见时调用。
6. onDestroy(): Activity销毁时调用。
7. onRestart(): Activity从停止状态到再次启动时调用。

6. 简述Activity的四种启动模式及其特点

1. standard: 每启动一个Activity就会在栈顶创建一个新的实例。
2. singleTop: 当被启动的Activity位于栈顶时, 复用该Activity实例。
3. singleTask: 当被启动的Activity在栈中存在实例时, 会直接复用此Activity实例, 并把当前Activity上面的所有实例弹出栈。
4. singleInstance: 会启动一个新的任务栈来管理该Activity。

7. 简述Activity、Intent、IntentFilter的作用

Activity表示活动, Android的四大组件之一, Intent表示意图, IntentFilter表示过滤器, 他们作用具体如下:

1. Activity是一个负责与用户交互的组件, 每个Android应用中都会用Activity来显示界面以及处理界面上一些控件的事件。
2. Intent被称为意图, 是程序中各组件间进行交互的一种重要方式, 它不仅指定当前组件要执行的动作, 还可以在不同组件之间进行数据传递。
3. IntentFilter为过滤器, 当发送一个隐式Intent后, Android系统会将它与程序中的每一个组件的过滤器进行匹配, 匹配属性有 action、data、category, 需要这三个属性都匹配成功才能唤起相应的组件。

8. 简述数据库事务的4个基本要素

1. 原子性: 表示事务是一个不可再分割的工作单位, 事务中的操作要么全部成功, 要么全部失败回滚。
2. 一致性: 表示事务开始之前和结束之后, 数据库的完整性没有被破坏。也就是说数据库事务不能破坏关系数据的完整性以及业务逻辑上的一致性。
3. 隔离性: 表示并发的任务是相互隔离的, 也就是一个事务内部的操作都必须封锁起来, 不会被其他事务影响到。
4. 持久性: 表示事务一旦提交后, 该事务对数据做的更改便持久保存在数据库中, 并不会被回滚, 即使出现了断电等事故, 也不会影响数据库中的数据。

9. 简述Android数据存储的方式★

1. 文件存储: Android提供了openFileInput()和openFileOutput()方法来读取设备上的文件, 其读取方式与Java中I/O程序是完全一样的。
2. SharedPreferences: 这是Android提供的用来存储一些简单的配置信息的一种机制, 它采用了XML格式将数据存储到设备中。通常情况下, 我们使用SharedPreferences存储一些应用程序的各种配置信息, 如用户名、密码等。
3. SQLite数据库: SQLite是Android自带的一个轻量级的数据库, 他运算速度快, 占用资源少, 还支持基本SQL语法, 一般使用他作为复杂数据的存储引擎, 可以存储用户信息等。
4. ContentProvider: Android四大组件之一, 主要用于应用程序之间的数据交换, 它可以将自己的数据共享给其他应用程序使用。
5. 网络存储: 需要与Android网络数据包打交道, 将数据存储到服务器上, 通过网络提供的存储空间来存储/获取数据信息。

10. 简述内部存储与外部存储的概念与区别★

- 内部存储：将应用程序中的数据以文件的形式存储到应用程序中（该文件默认位于 `data/data/<packagename>/` 目录下），此时存储的文件被其所在的应用程序私有化，如果其他应用程序想要操作本应用程序中的文件，则需要申请权限。当创建的应用程序被卸载时，其内部存储文件也随之被删除。
- 外部存储：将数据以文件的形式存储到一些外部设备（例如SD卡或者设备内嵌的存储卡）上，属于永久性的存储方式（外部存储的文件通常位于 `storage/emulated/0` 目录下，不同厂商生产的手机存储路径可能会不同）。外部存储的文件可以被其他应用程序共享，当将外部存储设备连接到计算机时，这些文件可以被浏览、修改和删除，因此这种方式不安全。

11. 简述内容提供者的工作原理

假设B程序需要操作A程序数据库中的数据，一般需要A程序使用ContentProvider暴露数据，才能被其他程序操作。B程序通过ContentResolver操作A程序暴露出来的数据，而A程序会将操作结果返回给ContentResolver，然后ContentResolver再将操作结果返回给B程序。

12. 简述内容观察者的工作原理

使用ContentObserver观察A程序的数据时，首先要在A程序的ContentProvider中调用ContentResolver的notifyChange()方法。调用此方法后，当B程序操作A程序中的数据时，A程序会向“消息中心”发送数据变化的消息，此时C程序会观察到“消息中心”的数据有变化，会触发ContentObserver的onChange()方法。

13. 广播机制的实现过程

Android中的广播使用了观察者模式，即基于消息的发布/订阅事件的模式。广播发送者和接收者分别处于观察者模式中的消息发布和订阅两端。广播机制的实现过程具体如下：

1. 广播接收者通过Binder机制在AMS(Activity Manager Service)中进行注册。
2. 广播发送者通过Binder机制向AMS发送广播。
3. AMS查找符合相应条件（IntentFilter/Permission）的广播接收者，将广播发送到相应的消息循环队列中。
4. 执行消息循环时获取到发送的广播，然后回调广播接收者中的onReceive()方法并在该方法中进行相关处理。

14. 简述有序广播和无序广播的区别★

1. 发送广播时使用的方法不同：有序广播使用sendOrderedBroadcast()发送广播，而无序广播使用sendBroadcast()方法发送广播。
2. 广播接收者执行的顺序
 - 有序广播的接收者是顺序执行的。有序广播按照广播接收者声明的优先级别被依次接收。当在高级别的广播接收者逻辑执行完毕之后，广播才会继续传递。当优先级相同时，先注册的广播接受者优先执行。
 - 无序广播是完全异步执行的。当发送无序广播时，所有监听这个广播的广播接收者都会接收到此广播消息，但接收和执行的顺序不确定。
3. 拦截广播：有序广播的接收者可拦截广播。如果优先级较高的广播接收者将广播终止，那么广播将不再向后传递。而无序广播则不能被拦截。
4. 效率：有序广播的效率比无序广播低。

15. 简述Service的两种启动方式

Service的启动方式分别可以调用startService()、bindService()方法，这两个启动方式的区别如下所示：

1. 生命周期

- startService(): 使用该方法开启Service时，执行的生命周期方法依次为onCreate()、onStartCommand()、onDestroy()。
- bindService(): 使用该方法开启Service时，执行的生命周期方法依次为：onCreate()、onBind()、onUnbind()、onDestroy()。

2. 停止服务的方法

- startService(): 调用stopSelf()、stopService()方法停止服务。
- bindService():调用unbindService()方法停止服务。

3. 组件的关联

- startService(): 当一个组件通过此方法开启服务时，服务与开启该组件没有关联，即使开启服务的组件被销毁，服务依旧运行。
- bindService(): 当一个组件通过此方法开启服务时，服务会与该组件绑定，组件一旦被销毁，该服务也会被销毁。

16. 简述Service的生命周期★

使用不同的方式启动服务，其生命周期会不同。开启服务的方法分别为startService()、bindService()。

当通过startService()方法启动服务时，执行的生命周期方法依次为onCreate()、onStartCommand()、onDestroy()

当通过bindService()方法启动服务时，执行的生命周期方法依次为onCreate()、onBind()、onUnbind()、onDestroy()。

其生命周期方法的具体含义如下：

1. onCreate(): 第一次创建服务时执行的方法。
2. onStartCommand(): 调用startService()方法启动服务时执行的方法。
3. onBind(): 调用bindService()方法启动服务时执行的方法。
4. onUnbind(): 调用unBindService()方法断开服务绑定时执行的方法。
5. onDestory(): 服务被销毁时执行的方法。

论述题

Android应用程序的基本组成

Android应用程序大致包含其内部四大组件：Activity、Service、Broadcast Receiver和Content Provider，以及资源文件、Activity Manager、Intent和数据等几个基本组件（但Android不一定要全部包含这几个部分）。Activity，Service和Broadcast Receiver都是由Intent异步消息激活的。在AndroidManifest.xml中，它们以不同的XML标签分别定义。

1. Android的四大组件：

1. Activity：活动是应用程序中用户界面的单个屏幕。每个活动都是用户与应用程序交互的基本单元。一般含有一组用于构建用户界面UI的 widget 控件，如按 Button、文本框 TextBox、列表List等
2. Service：服务是在后台执行长时间运行操作的组件，没有用户界面。例如，音乐播放器应用程序可能会使用服务来播放音乐。

3. Broadcast Receiver: 广播接收器是用于接收系统或应用程序发出的广播消息的组件。例如, 当设备电量低时, 系统会发送一个广播消息, 应用程序可以注册一个广播接收器来捕获此消息并采取相应的措施。
4. Content Provider: 内容提供者管理应用程序的数据集, 允许不同的应用程序共享数据。例如, 联系人应用程序可能会使用内容提供者来提供联系人信息给其他应用程序。
2. 资源文件: 含布局文件 main.xml等, 资源文件strings.xml等。
3. Activity Manager: 管理应用程序的生命周期。
4. Intent: 一种运行机制, 用来描述一个程序想要做些什么事、从一个Activity跳到另一个Activity等。
5. 数据 (包括文件数据和数据库数据) 可以用数据库、文件、网络等存储, 外界可以通过一套标准而统一的接口和数据打交道。当需要在不同程序间交换数据时, 可能就要用到Content Provider。