



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

第三版：Android 35 道

跨进程通信的几种方式

Intent, 比如拨打电话

ContentProvider 数据库存储数据

Broadcast 广播通信

AIDL 通信, 通过接口共享数据

wait 和 sleep 的区别

wait 是 Object 的方法, wait 是对象锁, 锁定方法不让继续执行, 当执行 notify 方法后就会继续执行, sleep 是 Thread 的方法, sleep 是使线程睡眠, 让出 cpu, 结束后自动继续执行

String, StringBuffer, StringBuilder 的区别

String 不可改变对象, 一旦创建就不能修改

```
String str="aaa";  
str="bbb";
```

以上代码虽然改变了 str, 但是执行过程是回收 str, 把值赋给一个新的 str

StringBuffer 创建之后, 可以去修改

StringBuilder 也可修改, 执行效率高于 StringBuffer, 不安全

当字符赋值少使用 String

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

字符赋值频繁使用 StringBuilder

当多个线程同步操作数据，使用 StringBuffer

View 和 SurfaceView 的区别

View 基于主线程刷新 UI，SurfaceView 子线程又可以刷新 UI

View 的绘制原理

View 为所有图形控件的基类，View 的绘制由 3 个函数完成

measure, 计算视图的大小

layout, 提供视图要显示的位置

draw, 绘制

简述 JNI

是 java 和 c 语言之间的桥梁，由于 java 是一种半解释语言，可以被反编译出来，一种重要涉及安全的代码就使用了 C 编程，再者很多底层功能调用 C 语言都实现了 Java 没必要重复造轮子，所以定义了 JNI 接口的实现

简述 TCP, UDP, Socket

TCP 是经过 3 次握手，4 次挥手完成一串数据的传送

UDP 是无连接的，知道 IP 地址和端口号，向其发送数据即可，不管数据是否发送成功

Socket 是一种不同计算机，实时连接，比如说传送文件，即时通讯

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

音视频相关类

总体来说，分为几个类

- 1、 视频录制方面，Camera 摄像头录制视频类，MediaProjection 屏幕录制视频类
- 2、 编码方面，MediaCodec, MediaRecorder
- 3、 预览方面，SurfaceView, GLSurfaceView, TextureView, VideoView

9.进程和线程的区别

概念：进程包括多个线程，一个程序一个进程，多线程的优点可以提高执行效率，提高资源利用率

创建：Thread 类和 Runnable 接口

常用方法有：

- 1、 start()用于启动线程
- 2、 run()调用线程对象中的 run 方法
- 3、 join()合并插队到当前线程
- 4、 sleep()睡眠释放 cpu 资源
- 5、 setPriority()设置线程优先级

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

内存泄露如何查看和解决

概念：有些对象只有有限的生命周期，当他们的任务完成之后，它们将被垃圾回收，如果在对象的生命周期本该结束的时候，这个对象还被一系列的引用，就会导致内存泄露。

解决方法：使用开源框架 LeakCanary 检测针对性解决

常见的内存泄露有：

- 1、单例造成的内存泄露，例如单例中的 Context 生命周期大于本身 Context 生命周期
- 2、线程使用 Handler 造成的内存卸扣，当 activity 已经结束，线程依然在运行更新 UI
- 3、非静态类使用静态变量导致无法回收释放造成泄露
- 4、WebView 网页过多造成内存泄露
- 5、资源未关闭造成泄露，例如数据库使用完之后关闭连接

View 的分发机制，滑动冲突

View 的事件传递顺序有 3 个重要的方法，dispatchTouchEvent()是否消耗了本次事件，onInterceptTouchEvent()是否拦截了本次事件，onTouchEvent()是否处理本次事件，滑动冲突分为同方向滑动冲突，例如 ScrollView 和 ListView，同方向滑动冲突，可以计算 ListView 高度而动态设置 ListView 的高度，ScrollView 高度可变。例如 ViewPager 和 ListView,不同方向滑动冲突，一个是横向滑动一个

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



是垂直滑动，不同方向滑动可以判断滑动的 x, y 轴是横向还是垂直滑动，如果判断得到是横向滑动，就拦截 ListView 的事件，竖则反之。

RecyclerView 和 ListView 的区别

缓存上:前者缓存的是 View+ViewHolder+flag，不用每次调用 findViewById,后者则只是缓存 View

刷新数据方面，前者提供了局部刷新，后者则全部刷新

recyclerView 嵌套卡顿解决如何解决

设置预加载的数量 `LinearLayoutManager.setInitialPrefetchItemCount(4)`，默认是预加载 2 个，

设置子项缓存，设置自带滑动冲突解决属性 `rv.setHasFixedSize(true)`;

`rv.setNestedScrollingEnabled(false)`;

可以完美解决，不过 Google 不推荐 RecyclerView 嵌套使用,需要嵌套尽量找类似于 `ExpandableListView` 第三方控件来解决

谈 MVC , MVP, MVVM

MVC:View 是可以直接访问 Model 的! 从而, View 里会包含 Model 信息, 不可避免的还要包括一些 业务逻辑。在 MVC 模型里, 更关注的 Model 的不变, 而同时有多个对 Model 的不同显示, 及 View。所以, 在 MVC 模型里, Model 不依赖于 View, 但是 View 是依赖于 Model 的。不仅如此, 因为有一些业务逻辑在 View 里实现了, 导致要更改 View 也是比较困难的, 至少那些业务逻辑是无法重用的。

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

MVP: MVP 是从经典的模式 MVC 演变而来, 它们的基本思想有相通的地方: Controller/Presenter 负责逻辑的处理, Model 提供数据, View 负责显示。作为一种新的模式, MVP 与 MVC 有着一个重大的区别: 在 MVP 中 View 并不直接使用 Model, 它们之间的通信是通过 Presenter (MVC 中的 Controller)来进行的, 所有的交互都发生在 Presenter 内部, 而在 MVC 中 View 会从直接 Model 中读取数据而不是通过 Controller。

MVVM: 数据双向绑定, 通过数据驱动 UI, M 提供数据, V 视图, VM 即数据驱动层

dagger2

Dagger2 是一个主要用于依赖注入的框架, 减少初始化对象操作, 降低耦合度

16Android 性能优化

布局优化: 减少布局层级, 使用 ViewStub 提高显示速度, 布局服用, 尽可能少使用 warp_content, 删除空间中无用的属性, 避免过度绘制, 移除 window 默认背景, 按需显示展位图, 自定义 View 优化, 使用 canvas.clipRect()识别可见区域

- 1、启动速度: 采用分布加载, 异步加载, 延期加载提高应用初始化速度, 采用线程初始化数据等, 合理的刷新机制
- 2、内存方面: 防止内存泄露, 使用一些第三方工具检测解决
- 3、代码优化: 遵循 Java 生命周期
- 4、安装包优化: 删除无用资源, 优化图片, 代码混淆, 避免重复库存在, 插件化

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

Handler 原理

Handler, loop 轮询检测发送消息到 MessengerQuery, MessageQuery 对 Message 入列, Handler 回调方法处理消息, 重写 handleMessage 方法刷新 ui

SharedPreferences 跨进程使用会怎么样? 如何保证跨进程使用安全?

在两个应用的 manifest 配置中好相同的 sharedUserId 属性, A 应用正常保存数据, B 应用

```
createPackageContext("com.netease.nim.demo",  
CONTEXT_IGNORE_SECURITY)
```

获取 context 然后获取应用数据, 为保证数据安全, 使用加密存储

推送到达率如何提高

判手机系统, 小米使用小米推送, 华为使用华为推送, 其他手机使用友盟推送

activity, fragment 传值问题

通过 Bundle 传值, 在 activity 定义变量传值, 扩展 fragment 创建传值

activity 与 fragment 区别

生命周期:

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

fragment 从创建到销毁整个生命周期依次为 onAttach() → onCreate() → onCreateView() → onActivityCreated() → onStart() → onResume() → onPause() → onStop() → onDestroyView() → onDestroy() → detach()

与 activity 不同的方法有

- 1、 onAttach(): 当 Fragment 和 Activity 建立关联的时候调用;
- 2、 onCreateView(): 当 Fragment 创建视图调用;
- 3、 onActivityCreated: 与 Fragment 相关联的 Activity 完成 onCreate() 之后调用;
- 4、 onDestroyView(): 在 Fragment 中的布局被移除时调用;
- 5、 detach(): 当 Fragment 和 Activity 解除关联时调用;
- 6、 activity 常用的生命周期只有以下几个;
- 7、 onCreate(): 表示 Activity 正在被创建, 常用来 初始化工作, 比如调用 setContentView 加载界面布局资源, 初始化 Activity 所需数据等;
- 8、 onRestart(): 表示 Activity 正在重新启动, 一般情况下, 当前 Activity 从不可见重新变为可见时, onRestart 就会被调用;
- 9、 onStart(): 表示 Activity 正在被启动, 此时 Activity 可见但不在前台, 还处于后台, 无法与用户交互;
- 10、 onResume(): 表示 Activity 获得焦点, 此时 Activity 可见且在前台并开始活动, 这是与 onStart 的区别所在;

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

11、 onPause(): 表示 Activity 正在停止, 此时可做一些 存储数据、停止动画等工作, 但是不能太耗时, 因为这会影响到新 Activity 的显示, onPause 必须先执行完, 新 Activity 的 onResume 才会执行;

12、 onStop(): 表示 Activity 即将停止, 可以做一些稍微重量级的回收工作, 比如注销广播接收器、关闭网络连接等, 同样不能太耗时;

13、 onDestroy(): 表示 Activity 即将被销毁, 这是 Activity 生命周期中的最后一个回调, 常做 回收工作、资源释放;

14、 区别:

15、 Fragment 比 Activity 多出四个回调周期, 控制操作上更灵活;

16、 Fragment 可以在 xml 文件中直接写入, 也可以在 Activity 中动态添加;

17、 Fragment 可以使用 show()/hide()或者 replace()对 Fragment 进行切换, 切换的时候不会出现明显的效果, Activity 切换的时候会有明显的翻页或其他效果;

Fragment 中 add 与 replace 的区别?

add 不会重新初始化 fragment, replace 每次都会;

添加相同的 fragment 时, replace 不会有任何变化, add 会报

IllegalStateException 异常;

replace 先 remove 掉相同 id 的所有 fragment, 然后在 add 当前的这个 fragment, 而 add 是覆盖前一个 fragment。所以如果使用 add 一般会伴随 hide()和 show(), 避免布局重叠;

使用 add, 如果应用放在后台, 或以其他方式被系统销毁, 再打开时, hide()中

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

引用的 fragment 会销毁,所以依然会出现布局重叠 bug,可以使用 replace 或使用 add 时, 添加一个 tag 参数;

FragmentPagerAdapter 与

FragmentStatePagerAdapter 的区别与使用场景?

FragmentPagerAdapter 的每个 Fragment 会持久的保存在 FragmentManager 中,只要用户可以返回到页面中,它都不会被销毁。因此适用于那些数据相对静态的页, Fragment 数量也比较少的情况;FragmentStatePagerAdapter 只保留当前页面,当页面不可见时,该 Fragment 就会被消除,释放其资源。因此适用于那些数据动态性较大、占用内存较多,多 Fragment 的情况;

说下 Activity 的四种启动模式、应用场景 ?

standard 标准模式: 每次启动一个 Activity 都会重新创建一个新的实例,不管这个实例是否已经存在,此模式的 Activity 默认会进入启动它的 Activity 所属的任务栈中; singleTop 栈顶复用模式: 如果新 Activity 已经位于任务栈的栈顶,那么此 Activity 不会被重新创建,同时会回调 onNewIntent 方法,如果新 Activity 实例已经存在但不在栈顶,那么 Activity 依然会被重新创建; singleTask 栈内复用模式: 只要 Activity 在一个任务栈中存在,那么多次启动此 Activity 都不会重新创建实例,并回调 onNewIntent 方法,此模式启动 Activity A,系统首先会寻找是否存在 A 想要的任务栈,如果不存在,就会重新创建一个任务栈,然后把创建好 A 的实例放到栈中; singleInstance 单实例模式: 这是一种加强的 singleTask 模式,具有此种模式的 Activity 只能单独地位于一个任务栈中,且此任务栈中只有唯一一个实例;

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

说下 Activity 跟 window , view 之间的关系?

Activity 创建时通过 attach()初始化了一个 Window 也就是 PhoneWindow, 一个 PhoneWindow 持有一个 DecorView 的实例, DecorView 本身是一个 FrameLayout, 继承于 View, Activity 通过 setContentView 将 xml 布局控件不断 addView()添加到 View 中, 最终显示到 Window 于我们交互;

横竖屏切换的 Activity 生命周期变化?

不设置 Activity 的 android:configChanges 时, 切屏会销毁当前 Activity, 然后重新加载调用各个生命周期, 切横屏时会执行一次, 切竖屏时会执行两次;

onPause()→onStop()→onDestory()→onCreate()→onStart()→onResume()

设置 Activity 的 android:configChanges="orientation", 经过机型测试

在 Android5.1 即 API 21 级别下, 切屏还是会重新调用各个生命周期, 切横、竖屏时只会执行一次

在 Android9 即 API 28 级别下, 切屏不会重新调用各个生命周期, 只会执行 onConfigurationChanged 方法

官方纠正后, 原话如下

如果您的应用面向 Android 2.3.2 即 API 级别 10 或更高级别 (按照 minSdkVersion 和 targetSdkVersion)

如何将打开 res raw 目录中的数据库文件?

1、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。

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

2、复制的基本方法是使用 `getResources().openRawResource` 方法获得 `res` 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。

3、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。

什么是 aar?aar 是 jar 有什么区别?

“aar”包是 Android 的类库项目的二进制发行包。文件扩展名是 .aar，maven 项目类型应该也是 aar，但文件本身是带有以下各项的 zip 文件：

- 1、 /AndroidManifest.xml (mandatory)
- 2、 /classes.jar (mandatory)
- 3、 /res/ (mandatory)
- 4、 /R.txt (mandatory)
- 5、 /assets/ (optional)
- 6、 /libs/*.jar (optional)
- 7、 /jni/*.so (optional)
- 8、 /proguard.txt (optional)
- 9、 /lint.jar (optional)

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

这些条目是直接位于 zip 文件根目录的。其中 R.txt 文件是 aapt 带参数 -output-text-symbols 的输出结果。

jar 打包不能包含资源文件，比如一些 drawable 文件、xml 资源文件之类的，aar 可以。

SQLite 支持事务吗？添加删除如何提高性能？

SQLite 作为轻量级的数据库，比 MySQL 还小，但支持 SQL 语句查询，提高性能可以考虑通过原始经过优化的 SQL 查询语句方式处理

如何将 SQLite 数据库(dictionary.db 文件)与 apk 文件一起发布？

可以将 dictionary.db 文件复制到 Eclipse Android 工程中的 res aw 目录中。所有在 res aw 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。可以将 dictionary.db 文件复制到 res aw 目录中

注册广播的几种方法？

1、静态注册:在清单文件中注册，常见的有监听设备启动，常驻注册不会随程序生命周期改变

2、动态注册:在代码中注册，随着程序的结束，也就停止接受广播了补充一点：有些广播只能通过动态方式注册，比如时间变化事件、屏幕亮灭事件、电量变更事件，因为这些事件触发频率通常很高，如果允许后台监听，会导致进程频繁创建和销毁，从而影响系统整体性能

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

为什么 Android 引入广播机制？

- 1、从 MVC 的角度考虑(应用程序内) 其实回答这个问题的时候还可以这样问，android 为什么要那 4 大组件，现在的移动开发模型基本上也是照搬的 web 那一套 MVC 架构，只不过是改了点嫁妆而已。
- 2、android 的四大组件本质上就是为了实现移动或者说嵌入式设备上的 MVC 架构
- 3、它们之间有时候是一种相互依存的关系，有时候又是一种补充关系，引入广播机制可以方便几大组件的信息和数据交互。
- 4、程序间互通消息(例如在自己的应用程序内监听系统来电)
- 5、效率上(参考 UDP 的广播协议在局域网的方便性)
- 6、设计模式上(反转控制的一种应用，类似监听者模式)

了解 IntentServices 吗？

- 1、IntentService 是 Service 的子类，是一个异步的，会自动停止的服务，很好解决了传统的 Service 中处理完耗时操作忘记停止并销毁 Service 的问题
- 2、生成一个默认的且与线程相互独立的工作线程执行所有发送到 onStartCommand() 方法的 Intent, 可以在 onHandleIntent() 中处理。
- 3、串行队列, 每次只运行一个任务, 不在线程安全问题, 所有任务执行完后自动停止服务, 不需要自己手动调用 stopSelf() 来停止。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

如何提升 Service 进程优先级

在 AndroidManifest.xml 文件中对于 intent-filter 可以通过 `android:priority = "1000"` 这个属性设置最高优先级，1000 是最高值，如果数字越小则优先级越低，同时适用于广播。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题