- 什么是 Activity?

通俗一点说 Activity 就是一个界面，这个界面里面可以放置各种控件。Activity 的界面也是用 xml 文件表示的，放置在 res->layout 下面。每生成一个新的 Activity 后，我们需要在 AndroidManifest.xml 中注册一下这个 activity

- 请描述一下 Activity 生命周期。

onCreate(Bundle savedInstanceState)：创建 activity 时调用。设置在该方法中，还以 Bundle 的形式提供对以前储存的任何状态的访问！
onStart()：activity 变为在屏幕上对用户可见时调用。
onResume()：activity 开始与用户交互时调用（无论是启动还是重新启动一个活动，该方法总是被调用的）。
onPause()：activity 被暂停或收回 cpu 和其他资源时调用，该方法用于保存活动状态的，也是保护现场，压栈吧！
onStop()：activity 被停止并转为不可见阶段及后续的生命周期事件时调用。
onRestart()：重新启动 activity 时调用。该活动仍在栈中，而不是启动新的活动。
onDestroy()：activity 被完全从系统内存中移除时调用，该方法被调用

- 两个 Activity 之间跳转时必然会执行的是哪几个方法。

onCrante() //在 Activity 生命周期开始时调用
onRestoreInstanceState()//用来恢复 UI 状态
onReStart()//当 Activity 重新启动时调用
onStart()//Activity 对用户即将可见时调用
onResume()//当 Activity 与用户交互时，绘制界面
onSaveInstanceState()//activity 即将移出栈顶保留 UI 状态时调用
onPause()//暂停当前活动 activity,提交持久数据的改变，停止动画和其他占用 CPU 资源的东西，由于下一个 activity 在这个方法返回之前不会 resume,所以这个方法的代码执行要快。
onStop()//activity 不再可见时调用
onDestroy()//在 Activity 销毁钱被调用的最后一个方法。

- 横竖屏切换时候 Activity 的生命周期。

1、不设置 Activity 的 android:configChanges 时，切屏会重新调用各个生命周期，切横屏时会执行一次，切竖屏时会执行两次
2、设置 Activity 的 android:configChanges="orientation"时，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
3、设置 Activity 的 android:configChanges="orientation|keyboardHidden"时，切屏不会重新调用各个生命周期，只会执行 onConfigurationChanged 方法

- 如何将一个 Activity 设置成窗口的样式。

1、在你的 styles.xml 文件中可以新建一如下的类似 Dialog 的 style

```
<style name="Theme.FloatActivity" parent="android:style/Theme.Dialog">
 </style>
```

2、在 AndroidManifest.xml 中在你需要显示为窗口的 activity 中添加如下属性：
android:theme="@style/Theme.FloatActivity" 即可

也可以直接添加您对应需要展示为 Dialog style 的 Activity 的 android:theme 属性值为 android:theme="@android:style/Theme.Dialog"。

- 你后台的 Activity 被系统回收怎么办？

系统会帮我们记录下回收前 Activity 的状态，再次调用被回收的 Activity 就要重新调用 onCreate()方法，不同于直接启动的是这回 onCreate()里是带上参数 savedInstanceState。savedInstanceState 是一个 Bundle 对象，你基本上可以把他理解为系统帮你维护的一个 Map 对象，我们使用 savedInstanceState 可以恢复到回收前的状态。

- 如何退出 Activity？如何安全退出已调用多个 Activity 的 Application？

用 finish()方法退出 activity.
在 2.1 之前，可以使用 ActivityManager 的 restartPackage 方法。
它可以直接结束整个应用。在使用时需要权限 android.permission.RESTART_PACKAGES。
在 2.2，这个方法失效了,可使用以下几个人工的方法
1、记录打开的 Activity：
每打开一个 Activity，就记录下来。在需要退出时，关闭每一个 Activity 即可。
2、发送特定广播：
在需要结束应用时，发送一个特定的广播，每个 Activity 收到广播后，关闭即可。

- 如果后台的 Activity 由于某原因被系统回收了，如何在被系统回收之前保存当前状态？

被回收前调用 onSaveInstanceState()方法保存当前状态。

- 两个 Activity 之间怎么传递数据？

在 Intent 的对象中增加要传递的参数既可。
在 Intent 的对象的请求中加入键值对，对象名字.putExtra("键值对的名字","键值对的值")；
在另一个 Activity 中将 Intent 请求中的数据取出来：

```
Intent intent=getIntent（）；//
String value = intent.getStringExtra("testIntent");//将 testIntent 对应的值赋值给 value
```

- 怎么在启动一个 Activity 时就启动一个 service？

将启动 Service 的语句放在 onCreate()方法中。

- 同一个程序，但不同的 Activity 是否可以放在不同的 Task 任务栈中？

可以放在不同的 Task 中。需要为不同的 activity 设置不同的 taskaffinity 属性，启动 activity 的 Intent 需要包含 FLAG_ACTIVITY_NEW_TASK 标记。

- Activity 怎么和 service 绑定，怎么在 activity 中启动自己对应的 service？

1、Activity 能进行绑定得益于 Service 的接口。为了支持 Service 的绑定，实现 onBind 方法。
2、Service 和 Activity 的连接可以用 ServiceConnection 来实现。你需要实现一个新的 ServiceConnection，重写 onServiceConnected 和 onServiceDisconnected 方法，一旦连接建立，你就能得到 Service 实例的引用。
3、执行绑定，调用 bindService 方法，传入一个选择了要绑定的 Service 的 Intent（显式或隐式）和一个你实现了的 ServiceConnection 实例

- 什么是 Service 以及描述下它的生命周期。

Android Service 是运行在后台的代码，不能与用户交互，可以运行在自己的进程，也可以运行在其他应用程序进程的上下文里。需要通过某一个 Activity 或者其他 Context 对象来调用， Context.startService() 和 Context.bindService()。 如果在 Service 执行耗时的操作需要启动一个新线程来执行。

Android Service 只继承了 onCreate(),onStart(),onDestroy()三个方法，当我们第一次启动 Service 时，先后调用了 onCreate(),onStart()这两个方法，当停止 Service 时，则执行 onDestroy()方法，这里需要注意的是，如果 Service 已经启动了，当我们再次启 Service 时，不会在执行 onCreate()方法，而是直接执行 onStart()方法。

- Service 有哪些启动方法，有什么区别，怎样停用 Service？

两种启动 Service 的方式 Context.startService() 和 Context.bindService()。 区别为 Context.startService()：Service 会经历 onCreate -> onStart（如果 Service 还没有运行，则 android 先调用 onCreate()然后调用 onStart()；如果 Service 已经运行，则只调用 onStart()，所以一个 Service 的 onStart 方法可能会重复调用多次 ）；stopService 的时候直接 onDestroy，如果是调用者自己直接退出而没有调用 stopService 的话，Service 会一直在后台运行。该 Service 的调用者再启动起来后可以通过 stopService 关闭 Service

Context.bindService()：Service 会经历 onCreate() -> onBind()，onBind 将返回给客户端一个 IBind 接口实例，IBind 允许客户端回调服务的方法，比如得到 Service 运行的状态或其他操作。这个时候把调用者（Context，例如 Activity）会和 Service 绑定在一起，Context 退出了，Srevice 就会调用 onUnbind -> onDestroyed 相应退出，所谓绑定在一起就共存亡了 。

停用 service 使用 context.stopService()

- 不用 service，B 页面为音乐播放，从 A 跳转到 B，再返回，如何使音乐继续播放？

a 使用 startActivityForResult() 方法开启 b，b 类结束时调用 finish();

a 类的 intent 有一个子 activity 结束事件 onActivityResult()，在事件里继续播放音乐

- 什么是 IntentService？有何优点？

IntentService 也是一个 Service，是 Service 的子类，

IntentService 和 Service 有所不同，通过 Looper 和 Thread 来解决标准 Service 中处理逻辑的阻塞问题。

优点：Acitivity 的进程，当处理 Intent 的时候，会产生一个对应的 Service

Android 的进程处理器现在会尽可能的不 kill 掉你

非常容易使用

日历中 IntentService 的应用

- 什么时候使用 Service？

比如播放多媒体的时候用户启动了其他 Activity 这个时候程序要在后台继续播放，比如检测 SD 卡上文件的变化，再或者在后台记 录你地理信息位置的改变等等，总之服务嘛，总是藏在后头的。

- 请描述一下 Intent 和 Intent Filter。

Intent 在 Android 中被翻译为"意图"，熟语来讲就是目的，他们是三种应用程序基本组件—activity，service 和 broadcast receiver 之间互相激活的手段。 在调用 Intent 名称时使用 ComponentName 也就是类的全名时为显示调用。这种方式一般用于应用程序的内部调用，因为你不一定会知道别人写的类的全名。我们来看看隐式 Intent 怎么用？ 首先我们先配置我们的 Activity 的 Intent Filter

```
<intent-filter>
    <action android:name="com.example.project.SHOW_CURRENT" />
</intent-filter>
```

这样在调用的时候指定 Intent 的 action，系统就是自动的去对比是哪个 intent-filter 符合我们的 Activity，找到后就会启动 Activity。

一个 intent filter 是 IntentFilter 类的实例，但是它一般不出现在代码中,而是出现在 android Manifest 文件中，以<intent-filter>的形式. (有一个例外是 broadcast receiver 的 intent filter 是使用 Context.registerReceiver()来动态设定的，其 intent filter 也是在代码中创建的.)

一个 filter 有 action, data, category 等字段. 一个隐式 intent 为了能被某个 intent filter 接受，必须通过 3 个测试. 一个 intent 为了被某个组件接受，则必须通过它所有的 intent filter 中的一个.

- Intent 传递数据时，可以传递哪些类型数据？

Intent 间传送数据一般有两种常用的办法：

1.extra

2.data.

extra 可以用 Intent.putExtra 放入数据。新启动的 Activity 可用 Intent.getExtras 取出来 Bundle，然后用 Bundles.getLong, getInt, getBoolean, getString 等函数来取放进入的值。而 data 则是传输 url。url 可以是指我们熟悉的 http, ftp 等网络地址， 也可以指 content 来指向 ContentProvider 提供的资源。Intent.setData 可以放入数据，Intent.getData 可以取出数据。

- 说说 Activity，Intent，Service 是什么关系 。

一个 Activity 通常是一个单独的屏幕，每一个 Activity 都被实现为一个单独的类，这些类都是从 Activity 基类中继承来的，Activity 类会显示由视图控件组成的用户接口，并对视图控件的事件做出响应。

Intent 的调用是用来进行架构屏幕之间的切换的。Intent 是描述应用想要做什么。Intent 数据结构中两个最重要的部分是动作和动作对应的数据，一个动作对应一个动作数据。

Android Service 是运行在后台的代码，不能与用户交互，可以运行在自己的进程，也可以运行在其他应用程序进程的上下文里。需要通过某一个 Activity 或者其他 Context 对象来调用。

Activity 跳转到 Activity，Activity 启动 Service，Service 打开 Activity 都需要 Intent 表明跳转的意图，以及传递参数，Intent 是这些组件间信号传递的承载者。

- 请描述一下 Broadcast Receiver。

Broadcast Receiver 用于接收并处理广播通知（broadcast announcements）。多数的广播是系统发起的，如地域变换、电量不足、来电来信等。程序也可以播放一个广播。程序可以有任意数量的 broadcast receivers 来响应它觉得重要的通知。broadcast receiver 可以通过多种方式通知用户：启动 activity、使用 NotificationManager、开启背景灯、振动设备、播放

声音等，最典型的是在状态栏显示一个图标，这样用户就可以点它打开看通知内容。通常我们的某个应用或系统本身在某些事件（电池电量不足、来电来短信）来临时会广播一个 Intent 出去，我们可以利用注册一个 Broadcast Receiver 来监听到这些 Intent 并获取 Intent 中的数据。

- 在 manifest 和代码中如何注册和使 用 broadcast receiver 。

1）在 AndroidManifest.xml 中注册

```
<receiver android:name="Receiver1">
        <intent-filter>
            <!-- 和 Intent 中的 action 对应 -->
              <action android:name="com.forrest.action.mybroadcast"/>
        </intent-filter>
</receiver>
```

2）在代码中注册

```
1. IntentFilter filter = new IntentFilter("com.forrest.action.mybroadcast"); // 和广播中 Intent 的 action 对应
2. MyBroadcastReceiver br = new MyBroadcastReceiver();
3. registerReceiver(new MyBroadcastReceiver(), filter);
```

- 请介绍下 ContentProvider 是如何实现数据共享的。

ContentProvider 是通过提供 Uri 来实现数据共享

- 请介绍下 Android 的数据存储方式。

Android 提供了 5 种方式存储数据：
使用 SharedPreferences 存储数据；
文件存储数据；
SQLite 数据库存储数据；
使用 ContentProvider 存储数据；
网络存储数据；

- 为什么要用 ContentProvider？它和 sql 的实现上有什么差别？

使用 ContentProvider 可以将数据共享给其他应用，让除本应用之外的应用也可以访问本应用的数据。它的底层是用 SQLite 数据库实现的，所以其对数据做的各种操作都是以 Sql 实现，只是在上层提供的是 Uri。

- 请介绍下 Android 中常用的五种布局。

最常用的布局方式为 LinearLayout、RelativeLayout、FrameLayout、TableLayout AbsoluteLayout。其中 LinearLayout 和 RelativeLayout 是最常用的方式，他们可以通过在 xml 配置文件或者代码中进行布局。

FrameLayout 最简单的布局方式，放置的控件都只能罗列到左上角，控件会有重叠，不能进行复杂的布局。
LinearLayou 可以通过 orientation 属性设置线性排列的方向是垂直(vertical)还是纵向(horizontal).每行或每列只有一个元素，可以进行复杂的布局。
AbsoluteLayout 可以让子元素指定准确的 x/y 坐标值，并显示在屏幕上。AbsoluteLayout 没有页边框，允许元素之间互相重叠（尽管不推荐）。他是绝对坐标，所以在实际中不提倡使用。
RelativeLayout 允许子元素指定他们相对于其它元素或父元素的位置（通过 ID 指定）。因此，你可以以右对齐，或上下，或置于屏幕中央的形式来排列两个元素。元素按顺序排列，因此如果第一个元素在屏幕的中央，那么相对于这个元素的其它元素将以屏幕中央的相对位置来排列。这个是相对于 AbsoluteLayout 的，采用的相对坐标，所以在实际中比较常用。
TableLayout 将子元素的位置分配到行或列中。一个 TableLayout 由许多的 TableRow 组成，每个 TableRow 都会定义一个 row 。TableLayout 容器不会显示 row 、column 或 cell 的边框线。每个 row 拥有 0 个或多个的 cell ；和 html 中的 table 差不多。在实际中也经常使用。
有的时候我们也会用到 GridView，就像我们手机屏幕上摆放的各个图标应该就是用 GridView 排版的。 Padding 是文字相对于边框，而 Margin 是边框相对于父窗体。

- 谈谈 UI 中， Padding 和 Margin 有什么区别？

Padding 是文字相对于边框，而 Margin 是边框相对于父窗体。

- widget 相对位置的完成在 antivity 的哪个生命周期阶段实现。

- 请解释下在单线程模型中 Message、Handler、Message Queue、Looper 之间的关系。

- AIDL 的全称是什么？如何工作？能处理哪些类型的数据？

AIDL 是一种接口定义语言，用于约束两个进程间的通信规则，供编译器生成代码，实现 Android 设备上的进程间通信。
进程之间的通信信息首先会被转换成 AIDL 协议消息，然后发送给对方，对方受到 AIDL 协议消息后再转换成相应的对象。
AIDL 支持的类型包括 Java 基础类型和 String，List，Map，CharSequence，如果使用自定义类型，必须实现 Parcelable 接口。

- 请解释下 Android 程序运行时权限与文件系统权限的区别。

运行时 Dalvik(android 授权)
文件系统 linux 内核授权

- 系统上安装了多种浏览器，能否指定某浏览器访问指定页面？

在 action 赋值为"android.intent.action.VIEW" 时可接收如下 scheme 为"http" 等等类型的 data 。所以突发奇想，启动该程序后，指定 action 及 Uri ，即访问指定网页。

- 对多线程的运用和理解，及多线程之间 handle 的传值。

- 对 android 虚拟机的理解，包括内存管理机制垃圾回收机制。

- Framework 工作方式及原理，Activity 是如何生成一个 view 的，机制是什么。

- android 本身的一些限制，比如 apk 包大小限制，读取大文件时的时间限。

- 如何加载的音乐信息，如何改善其效率。

Android 系统提供了 MediaScanner，MediaProvider，MediaStore 等接口，并且提供了一套数据库表格，通过 Content Provider 的方式提供给用户。当手机开机或者有 SD 卡插拔等事件发生时，系统将会自动扫描 SD 卡和手机内存上的媒体文件，如 audio，video，图片等，将相应的信息放到定义好的数据库表格中。
改善效率可以从界面需要查询必备数据，不需要的不进行查询。

- ListView 如何提高其效率？

使用分页加载，不要一次性加载所有数据。

- 启动应用后，改变系统语言，应用的语言会改变么？

不会

- 启动一个程序，可以主界面点击图标进入，也可以从一个程序中跳转过去，二者有什么区别？

从主界面启动一个应用程序是通过快捷方式直接调用 mainActivity 启动的，从其他应用程序调用需要隐式的通过 Action 或者在 Intent 中需要使用 setClass(),且要写明包路径。

- Android 程序与 Java 程序的区别？

android 程序是 Java 编写的，但程序使用的 android 开发的 API，就是 andriod 的库。

- Android 中 Task 任务栈的分配。
- 在 Android 中，怎么节省内存的使用，怎么主动回收内存？
- 不同工程中的方法是否可以相互调用？
- 在 Android 中是如何实现判断区分通话记录中的电话状态，去电，来电、未接来电？
- dvm 的进程和 Linux 的进程, 应用程序的进程是否为同一个概念

DVM 指 dalivk 的虚拟机.每一个 Android 应用程序都在它自己的进程中运行,都拥有一个独立的 Dalvik 虚拟机实例.而每一个 DVM 都是在 Linux 中的一个进程,所以说可以认为是同一个概念.

- sim 卡的 EF 文件有何作用

SIM 卡的文件系统有自己规范,主要是为了和手机通讯,SIM 卡本身可以有自己的操作系统,EF 就是作存储并和手机通讯用的。

- 如何判断是否有 SD 卡？

在程序中访问 SDCard，你需要申请访问 SDCard 的权限。

在 AndroidManifest.xml 中加入访问 SDCard 的权限如下:

```
<!-- 在 SDCard 中创建与删除文件权限 -->
<uses-permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<!-- 往 SDCard 写入数据权限 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)

Environment.getExternalStorageState()方法用于获取 SDCard 的状态，如果手机装有 SDCard，并且可以进行读写，那么方法返回的状态等于 Environment.MEDIA_MOUNTED。

- 嵌入式操作系统内存管理有哪几种， 各有何特性。
- 什么是嵌入式实时操作系统, Android 操作系统属于实时操作系统吗?
- 一条最长的短信息约占多少 byte?
- Linux 中跨进程通信的几种方式 。
- 谈谈对 Android NDK 的理解。

Android NDK 是一套工具，允许 Android 应用开发者嵌入从 C、C++源代码文件编译来的本地机器代码到各自的应用软件包中。

- 谈谈 Android 的优点和不足之处。
- Android 系统中 GC 什么情况下会出现内存泄露呢？

出现情况:

1. 数据库的 cursor 没有关闭

2.构造 adapter 时,没有使用缓存 contentview

衍生 listview 的优化问题-----减少创建 view 的对象,充分使用 contentview,可以使用一静态类来优化处理 getview 的过程/

3.Bitmap 对象不使用时采用 recycle()释放内存

4.activity 中的对象的生命周期大于 activity

调试方法: DDMS==> HEAPSZIE==>dataobject==>[Total Size]

- Android UI 中的 View 如何刷新。

一般只是希望在 View 发生改变时对 UI 进行重绘。你只需在 Activity 中显式地调用 View 对象中的 invalidate()方法即可。系统会自动调用 View 的 onDraw()方法。

- 简单描述下 Android 数字签名。
- 什么是 ANR 如何避免它？
- android 中的动画有哪几类，它们的特点和区别是什么？

两种，一种是 Tween 动画、还有一种是 Frame 动画。Tween 动画，这种实现方式可以使视图组件移动、放大、缩小以及产生透明度的变化；另一种 Frame 动画，传统的动画方法，通过顺序的播放排列好的图片来实现，类似电影。

- handler 机制的原理。

Andriod 提供了 Handler 和 Looper 来满足线程间的通信.Handler 先进先出原则.Looper 类用来管理特定线程内对象之间的消息交换(Message Exchange).

1)Looper: 一个线程可以产生一个 Looper 对象,由它来管理此线程里的 Message Queue(消息队列).

2)Handler:你可以构造 Handler 对象来与 Looper 沟通,以便 push 新消息到 Message Queue 里;或者接收 Looper 从 Message Queue 取出)所送来的消息.

android 中线程与线程，进程与进程之间如何通信。

线程通信使用 Handler，

- 说说 mvc 模式的原理，它在 android 中的运用。

MVC(Model_view_contraller)"模型_视图_控制器"。 MVC 应用程序总是由这三个部分组成。Event(事件)导致 Controller 改变 Model 或 View，或者同时改变两者。只要 Controller 改变了 Models 的数据或者属性，所有依赖的 View 都会自动更新。类似的，只要 Contro

Android 中各种界面的监听操作就是 MVC 的应用。

- android 中有哪几种解析 xml 的类，官方推荐哪种？以及它们的原理和区别。

DOM 解析

优点：

XML 树在内存中完整存储，因此可以直接修改其数据和结构。 2.可以通过该解析器随时访问 XML 树中的任何一个节点。 3.DOM 解析器的 API 在使用上也相对比较简单。

缺点：

如果 XML 文档体积比较大时，将文档读入内存是非常消耗系统资源的。

使用场景：

DOM 是用与平台和语言无关的方式表示 XML 文档的官方 W3C 标准。DOM 是以层次结构组织的节点的集合。这个层次结构允许开发人员在树中寻找特定信息。分析该结构通常需要加载整个文档和构造层次结构，然后才能进行任何工作。DOM 是基于对象层次结构的。

SAX 解析

优点：

SAX 对内存的要求比较低，因为它让开发人员自己来决定所要处理的标签。特别是当开发人员只需要处理文档中所包含的部分数据时，SAX 这种扩展能力得到了更好的体现。

缺点：

用 SAX 方式进行 XML 解析时，需要顺序执行，所以很难访问到同一文档中的不同数据。此外，在基于该方式的解析编码过程也相对复杂。

使用场景：

对于含有数据量十分巨大，而又不用对文档的所有数据进行遍历或者分析的时候，使用该方法十分有效。该方法不用将整个文档读入内存，而只需读取到程序所需的文档标签处即可。

Xmlpull 解析

android SDK 提供了 xmlpull api，xmlpull 和 sax 类似，是基于流（stream）操作文件，然后根据节点事件回调开发者编写的处理程序。因为是基于流的处理，因此 xmlpull 和 sax 都比较节约内存资源，不会象 dom 那样要把所有节点以对橡树的形式展现在内存中。

xmlpull 比 sax 更简明，而且不需要扫描完整个流。

- DDMS 与 TraceView 的区别？

Traceview 是 android 平台配备一个很好的性能分析的工具。它可以通过图形化的方式让我们了解我们要跟踪的程序的性能，并且能具体到 method。

http://wbdban.iteye.com/blog/564309

DDMS 为我们提供例如:为测试设备截屏，针对特定的进程查看正在运行的线程以及堆信息、

- res 目录有默认几项 resource。

6 项，drawable-hdpi，drawable-ldpi，drawable-mdpi，layout，values。

- android 的哪个版本是一次重大的升级？

1、6 版本。

系统新功能

快速搜索框（全局搜索）

新的摄像机和照相机

电池用量指示

Android Market（菜场）升级

新平台的新技术

Android 1.6 升级 Linux 内核从 2.6.27 到 2.6.29.

NotifactionManager 使用原理

1. 通过 getSystemService 方法获得一个 NotificationManager 对象。
2. 创建一个 Notification 对象。每一个 Notification 对应一个 Notification 对象。在这一步需要设置显示在屏幕上方状态栏的通知消息、通知消息前方的图像资源 ID 和发出通知的时间。一般为当前时间。
3. 由于 Notification 可以与应用程序脱离。也就是说，即使应用程序被关闭，Notification 仍然会显示在状态栏 中。当应用程序再次启动后，又可以重新控制这些 Notification。如清除或替换它们。因此，需要创建一个 PendingIntent 对象。该对象由 Android 系统负责维护，因此，在应用程序关闭后，该对象仍然不会被释放。
4. 使用 Notification 类的 setLatestEventInfo 方法设置 Notification 的详细信息。
5. 使用 NotificationManager 类的 notify 方法显示 Notification 消息。在这一步需要指定标识 Notification 的唯一 ID。这个 ID 必须相对于同一个 NotificationManager 对象是唯一的，否则就会覆盖相同 ID 的 Notificaiton。