

目录

第一章 android 开发基础.....	2
一、选择题.....	2
二、填空题.....	4
三、简答题.....	4
第二章 设计应用界面	6
一、选择题.....	6
二、填空题.....	10
三、简答题.....	10
第三章 Fragment 和图形控件.....	12
一、选择题.....	12
二、填空题.....	13
第四章 菜单和动作条	15
一、选择题.....	15
二、填空题.....	16
三、简答题.....	16
第五章 发送和接收消息.....	17
一、选择题.....	17
二、填空题.....	19
三、简答题.....	19
第六章 多任务和服务	23
一、选择题.....	23
二、填空题.....	25
三、简答题.....	25
第七章 数据存储.....	29
一、选择题.....	29
二、填空题.....	30
三、简答题.....	30
第八章 多媒体和触摸事件.....	32
一、选择题.....	32
二、简答题.....	32
第九章 内容提供者.....	33
一、选择题.....	33
二、解答题.....	33

第一章 android 开发基础

一、选择题

1. Android 项目工程下面的 **assets** 目录的作用是什么(B)

A、放置应用到的图片资源。

B、主要放置一些文件资源,这些文件会被原封不动打包到 apk 里面

C、放置字符串, 颜色, 数组等常量数据

D、放置一些与 UI 相应的布局文件, 都是 xml 文件

2. 关于 **res/raw** 目录说法正确的是(A)

A、这里的文件是原封不动的存储到设备上不会转换为二进制的格式

B、这里的文件是原封不动的存储到设备上会转换为二进制的格式

C、这里的文件最终以二进制的格式存储到指定的包中

D、这里的文件最终不会以二进制的格式存储到指定的包中

3. 下面哪一个不属于 **Android** 体系结构中的应用程序层(C)

A、电话簿

B、日历

C、SQLite

D、SMS 程序

4. 下面哪种说法不正确(B)

A、Android 应用的 **gen** 目录下的 **R.java** 被删除后还能自动生成;

B、**res** 目录是一个特殊目录, 包含了应用程序的全部资源, 命名规则可以支持数字(0-9)下横线(_),大小写字母(a-z, A-Z);

C、**AndroidManifest.xml** 文件是每个 **Android** 项目必须有的, 是项目应用的全局描述。其中指定程序的包名(**package="..."**)+指定 **android** 应用的某个组件的名字

(**android:name="..."**)组成了该组件类的完整路径

D、**assets** 和 **res** 目录都能存放资源文件, 但是与 **res** 不同的是 **assets** 支持任意深度的子目录, 在它里面的文件不会在 **R.java** 里生成任何资源 ID

5. 下面关于 **Android dvm** 的进程和 **Linux** 的进程,应用程序的进程说法正确的是(D)

A、DVM 指 **dalvik** 的虚拟机.每一个 **Android** 应用程序都在它自己的进程中运行,不一定拥有一个独立的 **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.

B、DVM 指 **dalvik** 的虚拟机.每一个 **Android** 应用程序都在它自己的进程中运行,不一定拥有一个独立的 **Dalvik** 虚拟机实例.而每一个 DVM 不一定都是在 **Linux** 中的一个进程,所以说不是一个概念.

C、DVM 指 dalvik 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 不一定都是在 Linux 中的一个进程,所以说不是一个概念.

D、DVM 指 dalvik 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.

6. 下面哪个属于 Android 体系结构中的应用程序? (C)

(A) SQLite (B)OpenGL ES (C)浏览器 (D)WebKit

7. Android 应用程序需要打包成(C)文件格式在手机上安装运行

(A).class (B).xml (C).apk (D).dex

8. 在 Android 应用程序中, 图片应放在那个目录下? (D)

A、raw B、values C、layout D、drawable

9. 下列选项哪个不是 Activity 启动的方法 (B)

A startActivity B goToActivity
C startActivityForResult D startActivityFromChild

10. 对于一个 在 AndroidManifest.xml 中对他这样进行定义

```
<intent-filter>
```

```
<action android:name="android.intent.action.MAIN" />
```

```
<category android:name="android.intent.category.LAUNCHER" />
```

```
</intent-filter>
```

这样的描述代表什么含义 (B)

- A、无明确含义 每个 Activity 都需要这样定义
- B、代表该 Activity 将作为程序的主活动, 并且在 LAUNCHER 菜单中启动
- C、代表该 Activity 将在桌面上建立图标并启动
- D、代表该 Activity 的优先级高于其他的 Activity

11. 在 AndroidManifest.xml 中描述一个 Activity 时 该 Activity 的 label 属性是指定什么的 (B)

- A、指定该 Activity 的图标
- B、指定该 Activity 的显示标签
- C、指定该 Activity 和类相关联的类名
- D、指定该 Activity 的唯一标示

12. 关于 AndroidManifest.xml 文件 以下描述错误的选项有哪些 (A)

- A、在所有的元素中只有<manifest>和<application>是必需的且只能出现一次
- B、处于同一层次元素, 不能随意打乱顺序
- C、元素属性一般都是可选的 但是有些属性是必须设置的
- D、对可选的属性 即使不写 也有默认的数值项说明

13. 假设 assets 目录下有文件结构 html/hello.html,用 loadUrl()方法将该网页加载至

webView 时,需传入的参数是 (B)

- A, file:///asset/html/hello.html
- B, file:///android_asset/html/hello.html
- C, file:///androidasset/hello.html
- D, file:///assets/html/hello.html

二、填空题

- 1、 Android 的四大组件是 activity, service, broadcast 和 contentprovide。
- 2、 Android SDK 主要以 java 语言为基础。
- 3、 android 虚拟设备的缩写是 (AVD)
- 4、 Android SDK 目前支持的操作系统 (Linux)
- 5、 AndroidSDK 提供一些开发工具可以把应用软件打包成 Android 格式文件 (APK)
- 6、 当启动一个 Activity 并且新的 Activity 执行完后需要返回到启动它的 Activity 来执行的回调函数是 startActivityResult()。
- 7、 Android 提供了一个 startActivityResult () 方法来实现 Activity 组件之间回传数据。

三、简答题

- 1、 简述 **Android** 系统架构从高到低的几个层次。(简述 **Android** 应用程序结构有哪些)四个层次, 分别是:
应用程序层 (Applications)、
应用程序框架层 (Application Framework)、
各种库 (Libraries) 和运行环境 (Runtime)、
Linux 内核 (Linux kernel)
- 2、 **Android** 工程下面有个 **gen** 目录, 该目录下有个 **R.java** 文件, 该文件的作用是什么, 能不能修改, 为什么。
答: 作用: 该文件相当于项目的字典项目中所涉及到的用户界面、字符串、图片、声音等资源都会在该文件中创建一个 ID 编号, 这些编号为整形, 以 16 进制自动生成。项目要使用这些资源时, 会通过这个类得到资源的引用。这个文件是只读模式, 不能够被修改
- 3、 把文件名为“图片 1.jpg”的一张图片放到 **android** 工程下的 **res- drawable-mdpi** 下, 会不会报错, 为什么?
答: 不会。因为使用 **Android** 系统的手机可能采用不同的分辨率, 所以在开发的时候会要求有不同分辨率的图片, **drawable-hdpi** 存高分辨率的, **drawable-mdpi** 存中等分辨率的, **drawable-ldpi** 存低分辨率的, 所以, 你要把相同的图片用图片处理软件处理成不同的分辨率的图片, 然后分别保存, 当然了, 如果你嫌麻烦, 也可以在三个目录下保存相同的图片, 这样也没有问题, 但是不符合开发的初衷。
- 4、 **Manifest.xml** 文件中主要包括哪些信息?

AndroidManifest.xml文件代码说明

代码元素	说 明
manifest	xml文件的根结点，包含了 package 中所有的内容
xmlns:android	命名空间的声明。使得 Android 中各种标准属性能在文件中使用。
package	声明应用程序包
uses-sdk	声明应用程序所使用的 Android SDK 版本
application	application 级别组件的根结点。声明一些全局或默认的属性，如标签、图标、必要的权限等。
android:icon	应用程序图标
android:label	应用程序名称
activity	Activity 是一个应用程序与用户交互的图形界面。每一个 Activity 必须有一个< activity >标记对应。
android:name	应用程序默认启动的活动程序 Activity 界面
intent-filter	声明一组组件支持的Intent值。在 Android 中，组件之间可以相互调用，协调工作，Intent提供组件之间通讯所需要的相关信息
action	声明目标组件执行的Intent动作。
category	指定目标组件支持的Intent 类别

第二章 设计应用界面

一、选择题

1. 下面的对自定义 **style** 的方式正确的是(A)

- A、`<resources>`
`<style name="myStyle">`
`<itemname="android:layout_width">fill_parent</item>`
`</style>`
`</resources>`
- B、`<style name="myStyle">`
`<itemname="android:layout_width">fill_parent</item>`
`</style>`
- C、`<resources>`
`<itemname="android:layout_width">fill_parent</item>`
`</resources>`
- D、`<resources>`
`<stylename="android:layout_width">fill_parent</style>`
`</resources>`

2. 下列样式表定义正确的是(C)

- A、`<style name="text">`
`<item name="android:textColor">#FF00FF</item>`
`</style>`
- B、`<resources>`
`<style name=" android:textColor ">#FF00FF </style>`
`</resources>`
- C、`<resources>`
`<style name="text">`
`<item name=" android:textColor">#FF00FF</item>`
`</style>`
`</resources>`
- D、`<resources>`

```
<style name="text">

<item name="textColor">#FF00FF</item>

</style></resources>
```

3. Android 是如何组织 Activity 的(A)
- A、以栈的方式组织 Activity B、以队列的方式组织 Activity
- C、以树形方式组织 Activity D、以链式方式组织 Activity。
4. onPause 什么时候调用(C)
- A . 当界面启动时 B . 当 onCreate 方法被执行之后
- C . 当界面被隐藏时 D . 当界面重新显示时
14. 在表格布局中, android:collapseColumns="1,2"的含义是(C)
- A、在屏幕中, 当表格的列能显示完时, 显示1, 2列
- B、在屏幕中, 当表格的列显示不完时, 折叠
- C、在屏幕中, 不管是否能都显示完, 折叠1、2列
- D、在屏幕中,动态决定是否显示表格。
15. 绝对布局中, android:layout_x 的含义有(B)
- A、以手机左下为原点, 组件显示到屏幕中的横向坐标值。
- B、以手机左上为原点, 组件显示到屏幕中的横向坐标值。
- C、以手机右下为原点, 组件显示到屏幕中的横向坐标值。
- D、以手机右下为原点, 组件显示到屏幕中的横向坐标值。
16. ScrollView 中, 可以直接包含多少个组件(C)
- A . 三个 B . 两个 C . 一个 D . 无数个
17. 关于 Activity 说的法不正确的是(C)
- A . Activity 是为用户操作而展示的可视化用户界面
- B . 一个应用程序可以有若干个 Activity
- C . Activity 可以通过一个别名去访问
- D . Activity 可以表现为一个漂浮的窗口
18. 在 Activity 的生命周期中, 当它从可见状态转向半透明状态时, 它的哪个方法必须被调用(B)
- A . onStop() B . onPause() C . onRestart()

D . onStart()

19. **activity** 对一些资源以及状态的操作保存，最好是保存在生命周期的哪个函数中进行

(D)

A、 onPause() B、 onCreate() C、 onResume() **D、 onStart()**

20. 退出 **activity** 对一些资源以及状态的操作保存，可以在生命周期 的哪个函数中进行

(A)

A、 onPause() B、 onCreate() C、 onResume() D、 onStart()

21. 下面退出 **Activity** 错误的方法是(D)

A、 finish() B、 抛异常强制退出

C、 System.exit() **D、 onStop()**

22. 在一个相对布局中怎样使一个控件居中(C)

A、 android:gravity="center" B、 android:layout_gravity="center"

C、 android:layout_centerInParent="true" D、 android:scaleType="center"

23. **Activity** 生命周期中 第一个需要执行的方法是什么(B)

A、 onStart **B、 onCreate**

C、 onReStart D、 onResume

24. 在 **Activity** 的生命周期中，当 **Activity** 被某个 **AlertDialog** 覆盖掉一部分后，会处于

哪种状态? (A)

(A) 暂停(B) 活动(C) 停止(D) 销毁

25. 关于 **RelativeLayout** 描述正确的是(C)

(A)该布局为绝对布局，可以自定义控件的 x,y 的位置

(B)该布局为切换帧布局，可实现标签切换的功能

(C)该布局为相对布局，其中控件的位置都是相对位置

(D)该布局为表格布局，需要配合 **TableRow** 一起使用

26. 下列不属于 **android** 布局的是(C)

A、 **FrameLayout**

B、 **LinearLayout**

C、 BorderLayout

D、 **TableLayout**

E、RelativeLayout

27. 如果使用 **LinearLayout** 实现组件垂直排列，那么在 **xml** 中应添加什么属性进行设置? (**A**)

A、android:orientation="vertical" B、android:vertical="true"

C、android:orientation="horizontal" D、android:horizontal="true"

28. 下列说法哪个不正确(**C**)

A、拥有 android:configChanges="orientation|keyboardHidden" 标签的 Activity 在横竖屏转换时不会再执行 onCreate 方法

B、默认情况下对一个 Activity 的对象进行横竖屏切换，该对象的 onCreate 方法在每次切换时都会执行

C、一个 Activity 的对象 a1 上弹出了一个模拟对话框形式的 Activity 的对象 a2，按返回键后 a1 执行了 onStart 和 onResume 方法, a2 执行了 onPause, onStop 和 onDestroy 方法

D、一个界面上的 EditText 中输入文字后，再按下 Home，该界面消失，等再回到该界面文字内容仍在，onCreate 方法也不会执行

29. AppWidget 窗口小部件时不可以使用下列哪个布局(**C**)

A,FrameLayout B,LinearLayout **C,TableLayout** D,RelativeLayout

30. 下列哪个属性是专用于相对布局的(**D**)

A.android.orientation

B.android:stretchColumns

C.android:layout_alignParentRight

D.android:layout_toRightOf

31. 为了使 android 适应不同分辨率的机型，布局时字体单位应用(**D**)

A.dp B.dip 像素 C.px **D.sp**

32. 下列属于 Activity 的状态是(**ABC**) [多选]

A.运行状态

B 暂停状态

C 停止状态

D 睡眠状态

33. 下面属于 View 的子类的是(**CD**) [多选]

A Activity

B Service

C ViewGroup

D TextView

34. 在 main.xml 中，定义一个组件时，有两个属性必须写(**AB**) [多选]

A android:layout_width

B android:layout_height

C android:id="@+id/start"

D android:text

35. 关于主题的说法，正确的是(**ABC**) [多选]

A 它是属性集合

B 它可以在程序中来设置

C 它通常用于一个 Activity 或所有 Activity 上

D 它可以用于单个 TextView 上

二、填空题

- 1、 activity 一般会重载 7 个方法用来维护其生命周期，除了 onCreate(),onStart(),onDestory() 外还有 onrestart,onresume,onpause,onstop。
- 2、 元素中 layout_width 的取值有 wrap_content、match parent。
- 3、 一个 Activity 就是一个 可视化的界面 或者看成是控件的容器。
- 4、 onPause()方法在 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，也是对运行时数据的现场保护。
- 5、 onDestroy 是 activity 被干掉前最后一个被调用方法，当调用 finish 方法或者系统为了节省空间将它暂时性的干掉时候调用。
- 6、 定义 LinearLayout 水平方向布局时至少设置的三个属性：android:orientation，android:layout width 和 android:layout height
- 7、 Android 当中基本的所有的 UI 都是由 (view) 或者其子类实现的
- 8、 Android4.0 中提供了 5 种布局管理器：FrameLayout、LinearLayout、RelativeLayout、TableLayout 和 GridLayout。

三、简答题

1、 简述 Activity 的生命周期

一个 activity 有三个基本的状态：

当 activity 在前台运行时（在 activity 当前任务的堆栈顶），为活动或者运行状态。这时 activity 会响应用户的操作。

当 activity 失去焦点但是对用户仍然可见时为 **paused** 暂停状态。此时，别的 activity 在他的上面，透明或者备有被全部覆盖。所以其中一些暂停的 activity 也可以被显示。一个暂停的 activity 是处于活动状态的（他维护着所有的状态保存着信息，并且依然附着在窗口管理器）。

如果一个 activity 完全被另一个 activity 所掩盖那他会处于 **stop** 状态。但仍然保存着原来的状态和信息。然而，如果别的地方需要更多的内存而且这个 activity 仍一直处于隐藏状态，那么系统有可能会杀死他的进程。

如果一个 activity 是暂停或者停止状态，系统可以清理他们占用的内存，或者调用 **finish()** 方法，或者直接结束他的进程。当他再次显示给用户时，会完全的重新运行并且加载以前所

存储的信息。

2、 阐述 `android:padding` 和 `android:layout_margin` 的异同？

(1) 不同点：`android:padding` 是站在父 View 的角度描述问题，它规定它里面的内容必须与这个父 View 边界的距离。`android:layout_margin` 则是站在自己角度描述问题，规定自己和其他（上下左右）的 View 之间的距离如同一级只有一个 View，那么他的效果基本上就和 `padding` 一样。

(2) 相同点：如同一级只有一个 View，那么 `android:layout_margin` 的效果基本上就和 `android:padding` 一样

3、 随着 **Activity** 的创建和销毁，可能会经历哪四种状态？

活动状态：当前的 **Activity**，位于 **Activity** 栈顶，用户可见，并且可以获得焦点

暂停状态：失去焦点的 **Activity**，依然可见，即使在内存低的情况下，也不会被系统杀死

停止状态：该 **Activity** 被其他 **Activity** 所覆盖，不可见，但是仍然保存所有的状态和信息。当内存低的情况下，它将要被系统杀死

销毁状态：该 **Activity** 结束，或 **Activity** 所在的 Dalvik 进程结束

5、 使用资源包括哪几种方法？

在代码里使用资源类；

在代码里调用资源类并实例化；

从其他资源类里引用

6、 现行布局中属性 **orientation** 的作用是什么？

用于设置布局管理器内组件的排列方式，其可选的值为 `horizontal` 和 `vertical`，默认值为 `vertical`

7、 **Android** 中实现事件处理的步骤？

(1) 将事件监听器注册到事件源

(2) 触发事件源上的事件

(3) 生成事件对象

(4) 触发事件监听器，事件被作为参数传入事件处理器

(5) 调用事件处理器做出响应

第三章 Fragment 和图形控件

一、选择题

1. 使进度条变横向的系统样式是(A)

A. `@android:style/Widget.ProgressBar.Horizontal`

B. `@android:style/ProgressBar.Horizontal`

C. `@style/Widget.ProgressBar.Horizontal`

D. `@style/ProgressBar.Horizontal`

2. `setOnTouchEvent` 设置返回值为 `true` 和 `false` 有何区别? (C)

A、没有区别，都能对事件进行监听

B、设置为 `true` 时 只能在移动时获得一次监听事件，`false` 则可以多次

C、返回 `true` 表示这个消息已经被处理结束，后续的 `handler` 不再接收到这个消息

D、设置为 `false` 是，在处理一次监听事件后，系统将抛弃该次事件

3. 下列哪个可做 `EditText` 编辑框的提示信息(D)

A `android:inputType` B `android:text`

C `android:digits` D `android:hint`

4. 在 `Activity` 中需要找到一个 `id` 是 `bookName` 的 `TextView` 组件,下面哪种语句写法是正确的(C)

A、`TextView tv = this.findViewById(R.id.bookName);`

B、`TextView tv = (TextView)this.findViewById(R.id.code);`

C、`TextView tv = (TextView)this.findViewById(R.id.bookName);`

D、`TextView tv=(TextView)this.findViewById(R.string.bookName)`

5. 如果将一个 `TextView` 的 `android:layout_height` 属性值设置为 `wrap_content` 那么该组件将是以下哪种显示效果(D)

A、该文本域的宽度将填充父容器宽度

B、该文本域的宽度仅占据该组件的实际宽度

C、该文本域的高度将填充父容器高度

D、该文本域的高度仅占据该组件的实际高度

6. 给一个 **TextView** 设置红色字体 应该使用以下哪种写法(**A**)
- A、`setTextColor(0xffff0000);`
B、`setColor("0xffff0000")`
C、`setTextColor("0xffff0000")`
D、`setColor("red")`
7. 下列哪个布局和组件是在开发 **AppWidget** 窗口小部件时不可以使用的(**B**)
- A,ImageView **B,WebView** C ,GridView D ,Button
8. 对于 **AlertDialog** 的描述不正确的是(**A**)
- A,使用 new 关键字创建 AlertDialog 的实例**
B,对话框的显示需要调用 show 方法
C,setPositiveButton 方法是用来加确定按钮的
D,setNegativeButton 方法是用来加取消按钮的
9. 如果需要捕捉某个组件的事件，需要为该组件创建(**B**)
- (A)属性 **(B)监听器** (C)方法 (D)工程
10. 如果需要捕捉某个组件的事件，需要为该组件创建(**D**)
- A、TextView B、EditText C、ImageView **D、AutoCompleteTextView**
11. 下列说法错误的是(**D**)
- A、Button 是普通按钮组件，除此外还有其他的按钮组件
B、TextView 是显示文本的组件，TextView 是 EditText 的父类
C、EditText 是编辑文本的组件，可以使用 EditText 输入特定的字符
D、ImageView 是显示图片的组件，可以通过设置显示局部图片

二、填空题

- 1、 设置 TextView 字体的属性是： `(android:typeface)`
- 2、 创建只显示文本的 Toast 对象时建议使用 `makeText` 方法
- 3、 可以做 EditText 编辑框的提示信息是 `(android:hint)`
- 4、 使用 RadiusButton 时， 要想实现互斥的选择需要用的组件是 **RadioGroup**
- 5、 给 ListView 设置适配器的方法是 (`setAdapter ()`)
- 6、 为了使 android 适应不同分辨率机型，布局时字体单位应用 (`sp`)，像素单位应用

(sp) 和 (dip)

- 7、 layout 布局文件的命名不能出现字母 (大写)
- 8、 Android 的事件处理机制有两种：一种是基于回调机制的；另一种是基于监听接口
- 9、 使用 RadioButton 时，要想实现互斥的选择需要用的组件是？ RadioGroup
- 10、进度条中 style 属性是设置进度条大小格式
- 11、多选框被选择事件:setonCheckchangeListener

第四章 菜单和动作条

一、选择题

1. 在 Android 中使用 Menu 时可能需要重写的方法有(AC)。

- A、onCreateOptionsMenu() B、onCreateMenu()
C、onOptionsItemSelected() D、onItemSelected()

2. 创建 Menu 需要重写的方法是(C)

- A、onOptionsCreateMenu (Menu menu) B、onOptionsCreateMenu (MenuItem menu)
C、onCreateOptionsMenu(Menu menu) D、onCreateOptionsMenu(MenuItem menu)

3. 创建子菜单的方法是(B)

- A、add B、addSubMenu C、createSubMenu D、createMenu

4. 处理菜单项单击事件的方法不包含(D)

- A、使用 onOptionsItemSelected(MenuItem item)响应
B、使用 onOptionsItemSelected(int featureId ,MenuItem item) 响应
C、使用 onOptionsItemSelected(MenuItem item) 响应
D、使用 onCreateOptionsMenu(Menu menu)响应

5. 上下文菜单与其他菜单不同的是(B)

- A、上下文菜单项上的单击事件可以使用 onOptionsItemSelected 方法来响应
B、上下文菜单必须注册到指定的 view 上才能显示
C、上下文菜单的菜单项可以添加，可以删除
D、上下文菜单的菜单项可以有子项

6. 在 android 中使用 Menu 时可能需要重写的方法有(A)

- A、onCreateOptionsMenu()、onOptionsItemSelected()
B、onCreateMenu()、onOptionsItemSelected()
C、onOptionsItemSelected()、onItemSelected()
D、onItemSelected()

二、填空题

- 1、处理菜单项点击事件的方法有 `onOptionsItemSelected()`、`onOptionsItemSelected(int featureId, MenuItem item)` 和 `onMenuItemClick(MenuItem item)`
- 2、处理菜单项被点击事件的方法有 `setonMenuItemClickListener` 监听
- 3、创建子菜单的方法 `addSubMenu()`

三、简答题

1、Android 应用程序的菜单有哪几种？

Android 应用程序的菜单有 4 种：

选择菜单 (Option Menu)、上下文菜单 (Context Menu)、弹出菜单 (Popup Menu)
子菜单 (Sub Menu)

第五章 发送和接收消息

一、选择题

1. Android 中下列属于 Intent 的作用的是(C)

- A、实现应用程序间的数据共享
- B、是一段长的生命周期，没有用户界面的程序，可以保持应用在后台运行，而不会因为切换页面而消失

C、可以实现界面间的切换，可以包含动作和动作数据，连接四大组件的纽带

D、处理一个应用程序整体性的工作

2. 下面在 AndroidManifest.xml 文件中注册 BroadcastReceiver 方式正确的(A)

A、

```
<receiver android:name="NewBroad">
    <intent-filter>
        <action
            android:name="android.provider.action.NewBroad"/>
        <action>
    </intent-filter>
</receiver>
```

B、

```
<receiver android:name="NewBroad">
    <intent-filter>
        android:name="android.provider.action.NewBroad"/>
    </intent-filter>
</receiver>
```

C、

```
<receiver android:name="NewBroad">
    <action
        android:name="android.provider.action.NewBroad"/>
    <action>
</receiver>
```

D、

```
<intent-filter>
    <receiver android:name="NewBroad">
        <action>
            android:name="android.provider.action.NewBroad"/>
        <action>
    </receiver>
</intent-filter>
```

3. 激活 Activity 的方法是(C)

- A . runActivity()
- B . goActivity()
- C . startActivity()
- D . startActivityForIn()

4. 关于 Intent 对象说法错误的是(D)
- A . 在 android 中, Intent 对象是用来传递信息的
 - B . Intent 对象可以把值传递给广播或 Activity
 - C . 利用 Intent 传值时, 可以传递一部分值类型
 - D . 利用 Intent 传值时, 它的 key 值可以是对象
5. 关于广播的作用, 正确的说法是(ABCD)[多选]
- A 它是用接收系统发布的一些消息的
 - B 它可以帮助 service 修改用户界面
 - C 它可以启动一个 Activity
 - D 它可以启动一个 Service
6. 意图可分为(AB)[多选]
- A 显式意图
 - B 隐式意图
 - C 组件意图
 - D 类意图
7. Intent 传递数据时, 下列的数据类型哪些可以被传递(ABCD)[多选]
- A、Serializable
 - B、charsequence
 - C、Parcelable
 - D、Bundle
8. 下列说法错误的是(B)
- A、Intent 起着 一个媒体中介的作用, 专门提供组件互相调用的相关信息, 实现调用者与被调用者之间的解耦
 - B、Intent 可以传递 View 对象
 - C、对于直接 Intent, Android 不需要去做解析, 因为目标组件已经很明确, Android 需要解析的是那些间接 Intent, 通过解析, 将 Intent 映射给可以处理此 Intent 的 Activity、IntentReceiver 或 Service
 - D、通过 Intent 可以删除程序
9. Android 项目启动时最先加载的是 AndroidManifest.xml 文件, 如果有多个 Activity, 以下哪个属性决定了该 Activity 最先被加载? (C)
- (A) android.intent.action.LAUNCH
 - (B) android.intent.action.ACTIVITY (没有)
 - (C) android.intent.action.MAIN
 - (D) android.intent.action.VIEW
10. 关于 BroadcastReceiver 的说法不正确的是(B)
- A,是用来接收广播 Intent 的
 - B,一个广播 Intent 只能被一个订阅了此广播的 BroadcastReceiver 所接收
 - C,对有序广播系统会根据接收者声明的优先级别按顺序逐个执行接收者
 - D,接收者声明的优先级别在<intent-filter>的 android:priority 属性中声明 数值越大优先

级别越高

二、填空题

- 1、 当应用程序中某广播在 `AndroidMainifest.xml` 进行注册后，即使该应用程序关闭后，也可以接受操作系统发出的广播信息。
- 2、 广播 broadcast 分为 **无序广播** 和 **有序广播**

三、简答题

1、 列举通过 **Intent** 传递不同数据类型的方法（至少 5 个）？

- ①Intent.addCategory(String category)
- ②Intetn.addFlags(int flags)
- ③Intent.equals(Object o)
- ④Intent.fillIn(Intent other,int flags)
- ⑤Intent.filterEquals(Intent other)

2、 简述 Activity 和 Intent、IntentFilter 的作用，以及它们之间的关系

Activity 作用：向用户呈现操作界面，与用户交互

Intent 作用：启动另一个 Activity，传递数据

IntentFilter 作用：可以帮助 Activity 跨应用调用，可以对 Activity 进行功能分组

3、 注册广播有几种方式，这些方式有何优缺点？请谈谈 Android 引入广播机制的用意。

1.常驻型广播（静态注册）

常驻型广播，当应用程序关闭了，如果有广播信息来，写的广播接收器同样的能接收到，它的注册方式就是在应用程序的 `AndroidManifast.xml` 中进行注册。这种方式可以理解为通过清单文件注册的广播是交给操作系统去处理的。示例代码如下：

<!--广播注册、name 里面填写广播类的路径-->

```
<receiver android:name=".SmsBroadCastReceiver">
```

```
    <intent-filter android:priority="20">
```

```
        <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
```

```
    </intent-filter>
```

```
</receiver>
```

使用场景：开机自启动的实现

优点：不受应用生命周期的影响，常驻

缺点：常驻会耗费 cpu、电量等资源

2.非常驻型广播（动态注册）

非常驻型广播，当应用程序结束了，广播自然就没有了，比如在 Activity 中的 onCreate 或者 onResume 中注册广播接收者，在 onDestroy 中注销广播接收者。这样广播接收者就一个非常驻型的了，这种注册方式也叫动态注册。这种方式可以理解为通过代码注册的广播是和注册者关联在一起的。示例代码如下：

//定义广播

```
class SmsReceiver extends BroadcastReceiver{
```

```
    @Override
```

```
    public void onReceive(Context context, Intent intent) {
```

```
        //doSomething
```

```
    }
```

```
}
```

//注册广播

```
private SmsReceiver smsReceiver;
```

```
smsReceiver=new SmsReceiver();
```

```
IntentFilter intentFilter=new IntentFilter("android.provider.Telephony.SMS_RECEIVED");
```

```
registerReceiver(smsReceiver,intentFilter);
```

//发送广播

```
Intent intent=new Intent();
```

```
intent.setAction("android.provider.Telephony.SMS_RECEIVED");
```

```
sendBroadcast(intent);
```

使用场景：更新 UI 等及时性通信操作

优点：在 Android 的广播机制中，动态注册的优先级高于静态注册的优先级，因此在必要情况下，我们需要动态注册广播接收器。取消注册后，不再占用资源

缺点：程序退出，注销广播后，便无法进行广播监听

3.引入广播的意义

Android 广播机制最大的特点是只管发送，不管有没有接收，也不管接收方怎么处理数据，这是它与普通方法的区别。事实上，Android 广播机制是为了解决跨进程通讯。而这一点是直接调用方法所不能实现的。例如，电量变化，很多应用要根据这个改变来作出相应操作，而在最初设计时，我们是不可能知道所有这些应用的，也就无法进行逐个实例化对象，不能直接调用方法，这时广播机制的作用就体现出来了。

由于 android 系统中应用程序之间不能共享内存，因此，在不同应用程序之间交互数据（跨进程通讯）就稍微麻烦一些。Android 广播有两种类型，一种是程序内的，一种程序间的。程序内的广播体现不出其真正意义。而程序间的广播，比如，通知时区改变、电池电量低、拍摄了一张照片或者用户改变了语言选项，这些才是其真正的用途。Android 广播一般用于一个程序告诉另一个程序，某些信息改变了，发生变化了，需要作出相关的应对，并不管有哪些程序接收到这些信息。

a: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 为什么要有那 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。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，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。

b: 程序间互通消息(例如在自己的应用程序内监听系统来电)

c: 效率上(参考 UDP 的广播协议在局域网的方便性)

d: 设计模式上(反转控制的一种应用，类似监听者模式)

4、 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？请说明原由。

通过直接发送 Uri 把参数带过去，或者通过 manifest 里的 intentfilter 里的 data 属性

5、 activity 间通过 Intent 传递数据，有没有最小限制？数据怎么传输？

没有最小限制，最小可以发一个空字符串，最大还可以传递图片（bitmap 序列化）

实现步骤：

- 1、Activity1 需要构造一个 Intent，这个 Intent 用于告诉系统，我们要做"查看"动作
- 2、intent 可调用 putExtra 来存放想要传递的数据
- 3、然后调用 setClass，设置 Activity1 和欲调用的组件 Activity2
- 4、最后调用 startActivity 将构造的 Intent 传入，系统会根据此 Intent 中的描述，到 Activity1

中找到满足此 Intent 要求的 Activity,

5、系统会调用找到的 Activity2 最终传入 Intent

6、在 Activity2 中可使用 getIntent 来获取传递的 Intent, 并通过获取数据的方法来获取数据

代码示例:

```
Intent intent = new Intent(); // Activity1
intent.putExtra("one", num1);
intent.putExtra("two", num2);
intent.setClass(FirstActivity.this, SecondActivity.class);
startActivity(intent);
```

```
Intent intent = getIntent(); //Activity2
String num1 = intent.getStringExtra("one");
String num2 = intent.getStringExtra("two");
int ret = Integer.parseInt(num1) + Integer.parseInt(num2);
result.setText(ret+"");
```

第六章 多任务和服务

一、选择题

1. Android 关于 service 生命周期的 onCreate()和 onStart()说法正确的是(AD)
 - A、当第一次启动的时候先后调用 onCreate()和 onStart()方法
 - B、当第一次启动的时候只会调用 onCreate()方法
 - C、如果 service 已经启动，将先后调用 onCreate()和 onStart()方法
 - D、如果 service 已经启动，只会执行 onStart()方法，不再执行 onCreate()方法
2. 我们都知道 Handler 是线程与 Activity 通信的桥梁，如果线程处理不当，你的机器就会变得越慢，那么线程销毁的方法是(A)
 - A、onDestroy()
 - B、onClear()
 - C、onFinish()
 - D、onStop()
3. 在 Activity 中，如何获取 service 对象(A)
 - A、可以通过直接实例化得到。
 - B、可以通过绑定得到。
 - C、通过 startService()
 - D、通过 getService()获取。
4. service 中如何实现更改 Activity 界面元素(B)
 - A、通过把当前 activity 对象传递给 service 对象。
 - B、通过向 Activity 发送广播。
 - C、通过 Context 对象更改 Activity 界面元素
 - D、可以在 service 中，调用 Activity 的方法实现更改界面元素。
5. 关于 android 进程，说法不正确的是(C)
 - A、组件运行所在的进程，是由 androidmanifest.xml 决定，它可以指定该组件运行于哪个进程。
 - B、当急需内存时，android 会决定优先关闭那些空闲的进程
 - C、背景进程是不为用户所见的 Activity，但是还会有可能被用户看到，所以它不能被杀死
 - D、可视进程一般不会不被系统所杀死
6. 关于线程说法不正确的是(B)
 - A、在 android 中，我们可以在主线程中，创建一个新的线程
 - B、在创建的新线程中，它可以操作 UI 组件

- C . 新线程可以和 Handler 共同使用
- D . 创建的 Handler 对象, 它隶属于创建它的线程
7. 关于 Handler 的说法正确的是(AB)[多选]
- A.它是实现不同线程间通信的一种机制 B.它避免了新线程操作 UI 组件
- C.它采用栈的方式来组织任务的 D.它可以属于一个新的线程
8. 下列关于 Service 的描述, 正确的是(D)
- A . Servie 主要负责一些耗时比较长的操作, 这说明 Service 会运行在独立的子线程中
- B . 每次调用 Context 类中的 StartService()方法后都会新建一个 Service 实例
- C . 每次启动一个服务时候都会先后调用 onCreate()和 onStart()方法
- D . 当调用了 ConText 类中的 StopService()方法后, Service 中的 onDestroy()方法会自动回调
9. 下列关于 Service 的描述, 不正确的是(B)
- (A)Service 是 Android 系统的后台服务组件, 适用于开发无界面、长时间运行的应用功能
- (B)Activity 比 Service 的优先级高, 不会轻易被 Android 系统终止
- (C)Service 有两种不同的使用方式, 一种是以启动方式使用 Service, 另一种是以绑定方式使用 Service
- (D)每个服务都继承自 Service 基类
10. 在 AsyncTask 中下列哪个方法是负责执行那些很耗时的后台计算工作的(C)
- A,run B,execute C,doinBackground D,onPostExecute
11. 使用 AIDL 完成远程 service 方法调用下列说法正确的是(BCD)
- A, aidl 对应的接口名称不能与 aidl 文件名相同
- B, aidl 的文件的内容类似 java 代码
- C, 创建一个 Service (服务), 在服务的 onBind(Intent intent)方法中返回实现了 aidl 接口的对象
- D,aidl 对应的接口的方法前面不能加访问权限修饰符
12. Timer 的实例为 mTimer,mTimer.schedule(mTimerTask, 1000, 2000);方法表示(C)

A, 每间隔1000秒后执行 mTimerTaskB,每间隔2000秒执行一次 mTimerTask

C,mTimerTask 是一个线程类 D,schedule 方法的第二个参数也可以是 Date

13. 下列哪些语句关于内存回收的说明是正确的? (B)

A、 程序员必须创建一个线程来释放内存 **B、 内存回收程序负责释放无用内存**

C、 内存回收程序允许程序员直接释放内存 D、 内存回收程序可以在指定的时间释放内存对象

二、填空题

1. Android 中 service 的实现方法是: **startservice** 和 **bindservice**。
2. 使用 **startService()**方法启动服务后, 调用者和服务间没有关联, 即使调用者退出了, 服务仍然进行。
3. android 系统 4 个应用程序之一, 主要用于后台运行和跨进程访问的是 **_service_**

三、简答题

1、 Android 根据应用程序的组件以及组件当前运行状态将所有的进程按重要性程度从高到低划分为几个?

可以划分为 5 个:

前台进程、可见进程、服务进程、后台进程、空进程

1、用户当前操作所必须的进程

(1) 拥有一个正在与用户交互的 Activity。

(2) 拥有一个 Service, 这个 Service 绑定了某一个正在与用户交互的 Activity。

(3) 拥有一个前台 Service。

(4) 拥有一个 Service 且它正在执行生命周期回调方法。

(5) 拥有一个 BroadcastsReceiver 且这个 BroadcastsReceiver 正在执行 onReceive 方法。

2、没有任何前台组件, 但仍会影响用户在屏幕上所见内容的进程

(1) 拥有一个可见当不可与用户交互的 Activity。

(2) 拥有一个 Service, 这个 Service 绑定了一个可见当不可与用户交互的 Activity。

3、由 startServer () 方法启动的 Service 进程

虽然不直接和用户所见内容关联, 但会执行一些用户关心的操作, 例如后台播放音乐、下载数据等, 若系统不足以维持前台进程和可见进程, 才会牺牲服务进程空间

4、包含用户不可见的 Activity 的进程, 这些进程对用户体验没有直接影响, 可以随时在任意时间终止它们, 以回收内存资源。系统通过 LRU(最近最少使用)列表进行多个后台进程的管理, 确保最近使用的 Activity 最后被终止

5、进程不含有任何应用组件, 该进程主要作用是缓存, 以改善在此进程中运行组件的启动时间。系统会经常终止此类进程

2、 startService()和 bindService()启动服务的区别:

服务不能自己运行, 需要通过调用startService()或bindService()方法启动服务。这两个方法都可以启动Service, 但是它们的使用场合有所不同。

1. 使用startService()方法启用服务,调用者与服务之间没有关连,即使调用者退出了,服务仍然运行。使用bindService()方法启用服务,调用者与服务绑定在了一起,调用者一旦退出,服务也就终止。

2. 采用startService()方法启动服务,在服务未被创建时,系统会先调用服务的onCreate()方法,接着调用onStart()方法。如果调用startService()方法前服务已经被创建,多次调用startService()方法并不会导致多次创建服务,但会导致多次调用onStart()方法。采用startService()方法启动的服务,只能调用Context.stopService()方法结束服务,服务结束时会调用onDestroy()方法。

3. bindService()方法启动服务,在服务未被创建时,系统会先调用服务的onCreate()方法,接着调用onBind()方法。这个时候调用者和服务绑定在一起,调用者退出了,系统就会先调用服务的onUnbind()方法,接着调用onDestroy()方法。如果调用bindService()方法前服务已经被绑定,多次调用bindService()方法并不会导致多次创建服务及绑定(也就是说onCreate()和onBind()方法并不会被多次调用)。如果调用者希望与正在绑定的服务解除绑定,可以调用unbindService()方法,调用该方法也会导致系统调用服务的onUnbind()->onDestroy()方法。

3、 handler 机制的原理

1、 handler 封装消息的发送(主要包括消息发送给谁)

主线程通过 Handler.handleMessage()读取消息队列中的消息

工作线程通过方法发送消息到主线程的消息队列

Handler.sendMessage()

Handler.post()

2、Looper——消息封装的载体。(1)内部包含一个MessageQueue,所有的Handler发送的消息都走向这个消息队列;(2)Looper.looper方法,就是一个死循环,不断地从MessageQueue取消息,如果有消息就处理消息,没有消息就阻塞。

Looper.myLooper():得到当前线程的Looper对象

Looper.getMainLooper():可以获得当前进程的主线程的Looper对象

Looper.prepare():创建消息队列

Looper.loop():进入消息循环

3、MessageQueue,一个消息队列,添加消息,处理消息

4、Message:线程间通讯的消息载体。Message充当消息封装的功能,里面可以存放任何你想要传递的消息。

5、handler内部与Looper关联,handler->Looper->MessageQueue,handler发送消息就是向MessageQueue队列发送消息。

总结:handler负责发送消息,Looper负责接收handler发送的消息,并把消息回传给handler自己。

4、 如何启用 Service, 如何停用 Service。

一. 步骤

第一步:继承Service类

```
public class SMSService extends Service { }
```

第二步:在AndroidManifest.xml文件中的<application>节点里对服务进行配置:

```
<service android:name=".DemoService" />
```

二. Context.startService()和Context.bindService

服务不能自己运行,需要通过调用Context.startService()或Context.bindService()方法启动服务。这两个方法都可

以启动Service，但是它们的使用场合有所不同。

1. 使用startService()方法启用服务，调用者与服务之间没有关连，即使调用者退出了，服务仍然运行。

使用bindService()方法启用服务，调用者与服务绑定在了一起，调用者一旦退出，服务也就终止。

2. 采用Context.startService()方法启动服务，在服务未被创建时，系统会先调用服务的onCreate()方法，

接着调用onStart()方法。如果调用startService()方法前服务已经被创建，多次调用startService()方法并不会导致多次创建服务，但会导致多次调用onStart()方法。

采用startService()方法启动的服务，只能调用Context.stopService()方法结束服务，服务结束时调用onDestroy()方法。

3. 采用Context.bindService()方法启动服务，在服务未被创建时，系统会先调用服务的onCreate()方法，

接着调用onBind()方法。这个时候调用者和服务绑定在一起，调用者退出了，系统就会先调用服务的onUnbind()方法，接着调用onDestroy()方法。如果调用bindService()方法前服务已经被绑定，多次调用bindService()方法并不会导致多次创建服务及绑定(也就是说onCreate()和onBind()方法并不会被多次调用)。如果调用者希望与正在绑定的服务解除绑定，可以调用unbindService()方法，调用该方法也会导致系统调用服务的onUnbind()->onDestroy()方法。

1. Service常用生命周期回调方法如下：

onCreate() 该方法在服务被创建时调用，该方法只会被调用一次，无论调用多少次startService()或bindService()方法，服务也只被创建一次。 onDestroy()该方法在服务被终止时调用。

2. Context.startService()启动Service有关的生命周期方法

onStart() 只有采用Context.startService()方法启动服务时才会回调该方法。该方法在服务开始运行时被调用。

多次调用startService()方法尽管不会多次创建服务，但onStart()方法会被多次调用。

3. Context.bindService()启动Service有关的生命周期方法

onBind() 只有采用Context.bindService()方法启动服务时才会回调该方法。该方法在调用者与服务绑定时被调用，

当调用者与服务已经绑定，多次调用Context.bindService()方法并不会导致该方法被多次调用。

onUnbind() 只有采用Context.bindService()方法启动服务时才会回调该方法。该方法在调用者与服务解除绑定时被调用。

备注：

1. 采用startService()启动服务

```
Intent intent = new Intent(DemoActivity.this, DemoService.class);
startService(intent);
```

2. Context.bindService()启动

```
Intent intent = new Intent(DemoActivity.this, DemoService.class);
bindService(intent, conn, Context.BIND_AUTO_CREATE);
```

```
//unbindService(conn); //解除绑定
```

5、 IntentService 有何优点?

答: IntentService 的好处

- * Activity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

6、 什么是 Service, Service 有哪些状态?

Services是一个没有任何UI能够在后台长时间运行的应用程序组件

Service是android 系统中的四大组件之一(Activity、Service、BroadcastReceiver、ContentProvider), 它跟Activity的级别差不多, 但不能自己运行只能后台运行, 并且可以和其他组件进行交互。

Started 一个activity调用startService() 来启动它

Bound 一个activity调用bindService() 来绑定它

第七章 数据存储

一、选择题

1. 在 Android 中使用 SQLiteOpenHelper 这个辅助类时，可以生成一个数据库，并可以对数据库版本进行管理的方法可以是(AB)

A、getWritableDatabase()

B、getReadableDatabase()

C、getDatabase()

D、getAbleDatabase()

2. 关于 ContentValues 类说法正确的是(D)

A、他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是任意类型，而值都是基本类型

B、他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名，可以为空，而值都是 String 类型

C、他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是 String 类型，而值也是 String 类型

D、他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是 String 类型，而值都是基本类型

3. 关于 Sqlite 数据库，正确的说法(ABD) [多选]

A SqliteOpenHelper 类主要是用来创建数据库和更新数据库

B SQLiteDatabase 类是用来操作数据库的

C 在每次调用 SQLiteDatabase 的 getWritableDatabase()方法时，会执行 SQLiteOpenHelper 的 onCreate 方法。

D 当数据库版本发生变化时，可以自动更新数据库结构

4. 在使用 SQLiteOpenHelper 这个类时，它的哪一个方法是用来实现版本升级之用的

(D)

A . onCreate()

B . onCreate()

C . onUpdate()

D . onUpgrade()

5. android 中文件操作模式中表示只能被本应用使用，写入文件会覆盖的是(D)

A, MODE_APPEND

B, MODE_WORLD_READABLE

C, MODE_WORLD_WRITEABLE

D, MODE_PRIVATE

6. 下列对 **SharedPreferences** 存、取文件的说法中不正确的是 (C)
- A,属于移动存储解决方案 B,sharePreferences 处理的就是 key-value 对
C,读取 xml 文件的路径是/sdcard/shared_prefix D,信息的保存格式是 xml
7. 数据源如果为 **sqlite** 数据库中查出的信息最适合的适配器为 (B)
- A.SimpleAdapter B.SimpleCursorAdapter
C.ArrayAdapter D.ListAdapter
8. 下列哪个可做 **Android** 数据存储 (A)
- A、SQLite B、MySQL C、Oracle D、DB2
9. 在手机开发中常用的数据库是 (A)
- A,sqlite B,Oracle C,Sql Server D,Db2
10. 下列哪个是 **SQLite** 下的命令 (C)
- A、shell B、push C、quit D、keytool

二、填空题

- 1、 android 的数据存储的方式 sharedpreference, 文件, SQLite,contentprovider, 网络。
- 2、 SQLite 支持五种数据类型包括 TEXT、NULL、BLOB、REAL、INTEGER。
- 3、 Android 平台提供了 2D, 3D 的图形支持, SQLite 数据库支持, 并且集成了浏览器
- 4、 Android 中文件操作模式中表示只能被本应用使用的是 MODE_PRIVATE 模式
- 5、 在手机开发中常用的数据库是? SQLite
- 6、 读取文件内容的首要方法是: openFileInput()

三、简答题

1、 简述 **Android** 的数据存储方式。

(1)使用 **SharedPreferences** 存储数据它是 **Android** 提供的用来存储一些简单配置信息的一种机制, 采用了 XML 格式将数据存储到设备中 e 只能在同一个包内使用, 不能在不同的包之间使用。

(2)文件存储数据, 文件存储方式是一种较常用的方法, 在 **Android** 中读取 / 写入文件的方法, 与 Java 中实现 I/O 的程序是完全一样的, 提供了 **openFileInput()** 和 **openFileOutput()**方法来读取设备上的文件。

(3)**SQLite** 数据库存储数据, **SQLite** 是 **Android** 所带的一个标准的数据库, 它支持 SQL 语句, 它是一个轻量级的嵌入式数据库。

(4)使用 **ContentProvider** 存储数据, 主要用千应用程序之间进行数据交换, 从而能够让其他的应用保存或读取此 **Content Provider** 的各种数据类型。

(5)网络存储数据, 通过网络上提供给我们的存储空间来上传 (存储) 和下载 (获取) 我们存储在网络空间中的数据信息。

2、简述创建或打开一个 SQLite 数据库的方法有哪些？

`openDataBase`、`openOrCreateDataBase`

3、如何将打开 `res aw` 目录中的数据库文件？

解答：在 Android 中不能直接打开 `res aw` 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。

4、SharedPreferences 本质是什么？以什么形式存在于 APP 中？

`SharedPreferences` 是 `android.content.SharedPreferences` 包内的一个类，它主要是用来共享程序数据的。

`SharedPreferences` 是以键值对 (key-value) 的方式来存储数据，它可以存储 `booleans`, `floats`, `ints`, `longs`, and `strings` 这些基本的数据类型。

`SharedPreferences` 存储的数据是持久性的，一旦通过其存储数据，数据就一直保留着，即便是你的 APP 完全停止运行，从内存中完全退出，这些数据也会被保留下来，

下一次你再进入 APP 时，从 `SharedPreferences` 中，你仍然可以读出上一次写入的数据。从这个角度来讲，`SharedPreferences` 存储数据的方式和 xml 文件类似。

但是，如果把 APP 卸载了，下次安装时，`SharedPreferences` 中的数据会被清除，这点和 xml 文件是不一样的。

获取 `SharedPreferences` 对象的方法有两种：

`getSharedPreferences()`，这种方法用于 APP 中有多个 `SharedPreferences` 文件，通过名字来获取 `SharedPreferences` 对象；

`getPreferences()`，这种方法用于 Activity 中只有一个 `SharedPreferences` 文件时，因为只有一个，所以就不必提供名字。

实际应用时，第一种方法用的会更多。

向 `SharedPreferences` 中写数据：

通过 `edit()` 方法获取 `SharedPreferences.Editor` 对象；

调用 `putXXX()` 方法，放值，(XXX 为基本数据类型)；

调用 `commit()` 方法提交放入的值。

第八章 多媒体和触摸事件

一、选择题

1. 下面属于 **android** 的动画分类的有(**AB**)

A、Tween B、Frame C、Draw D、Animation

2. 以下属于调用摄像头硬件的权限的是：(**A**)

A.<uses-permission android:name="android.permission.CAMERA"/>

B.<uses-permission

android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />

C.<uses-permission

android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

D.<uses-permission android:name="android.permission.INTERNET"/>

二、简答题

1、 音频播放

(1)从源文件中播放： 1) 在项目的 res/raw 文件夹下放置一个 Android 支持的文件， 如 MP3; 2) 创建一个 MediaPlayer 实例， 可以使用 MediaPlayer 的静态方法 create() 来完成； 3) 调用 start()方法开始播放，调用 pause()暂停播放， 调用 stop()方法停止播放， 如果要重复播放， 在调用 start()方法前， 必须调用 reset()方法和 prepare()方法

(2) 从文件系统中播放： 1) 实例化一个 MediaPlayer 2) 调用 setDataSource()方法设置播放文件路径 3)先调用 prepare(), 然后调用 start()方法播放

3)从网络中播放： 方法一 1) 创建 URI 实例； 2) 根据一个 MediaPlayer 实例， 使用 MediaPlayer 的静态方法 create(), 通过传递 URI 参数完成： 3)调用 start()方法开始播放
方法二 1)实例化一个 MediaPlayer: 2) 调用 setDataSource()方法设置想要播放文件的路径，网络上的可用路径； 3) 先调用 prepare(), 然后调用 start() 播放

2、 绘制一张 K线图，需要重写 View 的哪些方法？

有三个构造方法（一个参数、两个参数、三个参数），其中两个参数的构造方法必须有。

onDraw()方法必须有，是用来绘制 View 图像的

如果要改变 View 的大小，需要重写 onMeasure()方法。

如果要改变 View 在父控件中的位置，需要重写 onLayout()方法

执行顺序 onMeasure()==> onLayout()==> onDraw()

第九章 内容提供者

一、选择题

1. 关于 **ContentValues** 类说法正确的是(A)

A、他和 **Hashtable** 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的，名是 **String** 类型，而值都是基本类型

B、他和 **Hashtable** 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是任意类型，而值都是基本类型

C、他和 **Hashtable** 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名，可以为空，而值都是 **String** 类型

D、他和 **Hashtable** 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是 **String** 类型，而值也是 **String** 类型

二、解答题

1、请介绍下 **ContentProvider** 是如何实现数据共享的？

创建一个属于你自己的 **Content provider** 或者将你的数据添加到一个已经存在的 **Content provider** 中，前提是有相同数据类型并且有写入 **Content provider** 的权限。

2、什么是 **Content Provider**？

内容提供者是应用程序之间共享数据的接口，

A content provider can store and retrieve data making it accessible to all application

比如：联系人提供者专为不同应用程序提供联系人数据；设置提供者专为不同应用程序提供系统配置信息，包括内置的设置应用程序等。

7 下列说法哪一个不正确 (1.3分)

- A、 Java程序经编译后会产生字节码
- B、 java虚拟机加载java文件
- C、 javac是编译的命令
- D、 javadoc是文档化的命令

☐ A ☐ B ☐ C ☐ D

9 使用HttpURLConnection的Get方式请求数据时,下列哪个属性是必须设置的 (1.3分)

- A、 connection.setDoOutput(true)
- B、 connection.connect()
- C、 connection.setRequestMethod("POST")
- D、 connection.setDoInput(true)

☐ A ☐ B ☐ C ☐ D

11 进度条中哪个属性是设置进度条大小格式的() (1.3分)

- A、 android:secondaryProgress
- B、 android:progress
- C、 android:max
- D、 style

☐ A ☐ B ☐ C ☒ D

15 在android中使用Menu时不需要重写的方法有 (1.3分)

- A、 onCreateOptionsMenu()
- B、 onCreateMenu()
- C、 onOptionsItemSelected()
- D、 onOptionsItemSelected ()

☐ A ☒ B ☐ C ☐ D

19 下列关于如何使用Notification,不对的是() (1.3分)

- A、 notification需要NotificatinManager来管理
- B、 使用NotificationManager的notify方法显示notification消息
- C、 在显示Notification时可以设置通知时的默认发声,震动等
- D、 Notification中有方法可以清除消息

22 android 关于service生命周期的onCreate()和onStart()说法正确的是() (1.3分)

- A、 当第一次启动的时候先后调用onStart()和onCreate()方法
- B、 当第一次启动的时候只会调用onCreate()方法
- C、 如果service已经启动,将先后调用onCreate()和onStart()方法
- D、 如果service已经启动,只会执行onStart()方法,不在执行onCreate()方法

23 下列哪种不是Android的存储方式?

(1.3分)

- A、 form
- B、 SharedPreferences
- C、 SQLite
- D、 ContentProvider

☒ A ☐ B ☐ C ☐ D

28 下列不是手机操作系统的是() (1.3分)

- A、 Android
- B、 Window Mobile
- C、 Apple iPhone IOS
- D、 Windows Vista

☐ A ☐ B ☐ C ☒ D

20.关于 AndroidManifest.xml 文件, 以下描述错误的选项有哪些

- A、 在所有的元素中只有<manifest>和<application>是必需的,且只能出现一次
- B、 处于同一层次的元素,不能随意打乱顺序
- C、 元素属性一般都是可选的,但是有些属性是必须设置的
- D、 对可选的属性,即使不写,也有默认的数值项说明

2 Intent传递数据时,下列的数据类型哪些可以被传递 (3.0分)

A、 Serializable

B、 charsequence

C、 Parcelable

D、 Bundle

☒ A ☒ B ☒ C ☒ D

android中使用SQLiteOpenHelper这个辅助类时,可以生成一个数据库,并可以对数据库版本进行管理的方法可以是?

- ☐ A、 getWritableDatabase()
- ☐ B、 getReadableDatabase()
- ☐ C、 getDatabase()
- ☐ D、 getAbleDatabase()

答案: AB

8 android 中service的实现方法是:_____和_____。(3.0分)

第1空:

start ...

第2空:

bind ...

你如何评价 Android 系统? 优缺点?

优点: 开放性: 开发的平台允许任何移动终端厂商加入到 Android 联盟中来, 有丰富的软件, 硬件。

挣脱运营商的束缚; 接入 google 服务。

缺点: 硬件、屏幕太多难适配; 安全和隐私, 信息被 google 监视

6 页面上现有ProgressBar控件progressBar,请用书写线程以10秒的时间完成其进度 显示工作。(1.5分)

     上传附件  

上一题

当前第50题/共 56 题

下一题

交卷