Android 平台开发相关工具

1.SDK(软件开发包)

2.ADT(Eclipse 插件，安卓开发工具)

3.ADB(安卓设备调试桥)

4.DDMS(设备调试监视系统)

5.Eclipse(集成开发工具IDE)

6.Android studio

7.LogCat(日志输出)

Android 项目目录结构

1.src(自己写的java程序)

2.gen(项目自动生成的java程序，R.java,BuildConfig.java)

3.assert(项目资源目录，可以存储一些web文件等)

4.bin(存储二进制目录，例如项目的apk文件)

5.libs(类库目录，此目录可以添加第三方的一些类库文件)

6.res(资源目录):drwable,layout,values,menu,anim,...

7.AndroidManifest.xml

Anroid Common Layout相关优化

\*Hierarchy View 工具的使用

1.提高系统运行效率(性能)，改善用户体验

2.提高代码的可重用性，可维护性，从而提高开发效率，降低成本

优化方式：

1.布局的设计？(使用那种布局更合适)

2.布局的层次？(层次结构嵌套关系越浅越好)

3.布局对象个数？(对象个数越少越好)

4.布局对象加载时机(何时需要何时加载)

优化实现：

1.靠量布局设计是否合理

2.借助merge元素标签实现布局(相同的跟元素)的合并

说明merge只能作为根标签出现

3.通过include引入其他可重用的布局元素

<include layout="@layout/activity\_main" />

4.通过ViewStub延时加载(何时需要何时加载)我们需要的元素

Android Adapter Layout 概述

1.Adapter Layout 是什么？

1.View

2.ViewGroup

3.AdapterView(适配视图)

2.Adapter Layout 应用场合？

结合具体的Adapter Layout进行分析，例如ListView要以列表的形式显示多个item

3.Adapter Layout 常用类型？

1.ListView

2.GridView

3.Spinner

4.ViewPager(特殊的ViewGroup)

5......

4.Adapter Layout 构成及实现原理分析？

所以的Adapter Layout都是一种特殊的VeiwGroup，这些ViewGroup会以某种布局形式呈现

相关Item（这些Item可以理解为ViewGroup中的子元素），然后通过这些Item为用户呈现具体数

据内容。

这些ViewGroup中呈现的Item都需要借助一个称之为Adapter(适配器)的对象进行构建及组装。

Adapter在构建Item是需要：

1.Item数据(例如一个联系人信息)

2.Item布局(例如一个用于呈现联系人信息的View)

3.bind数据(将这个联系人的信息添加到Item布局对象上)

5.Adapter Layout 应用构建步骤？

1.获得或构建Adapter Layout对象(例如ListView,GridView,...)

2.构建一个Adaper(适配器)对象(new XXXAdaper(...))

3.让Adapter Layout关联适配器对象(setAdapter())

6.Adapter Layout 中常用Adapter及相关概述？

1.ListAdapter(接口)8+String

2.BaseAdapter(实现了ListAdapter的抽象类型，可以在此基础上[[[自己定义适配器]]])

3.ArrayAdapter(基于数组实现的一个简单适配器)

4.SimpleAdapter(基于List<? extends Map<String,?>>实现的一个适配器)List<? extends Map<String,?>>

5.SimpleCursorAdapter(基于数据库Cursor实现的一个适配器)

6.PagerAdapter(主要针对于ViewPager)

7......

7.核心API

ListView(AbsListView-->AdapterView-->ViewGroup-->View)

ArrayAdapter(BaseAdapter-->ListAdapter-->Adapter)

Other(List,ArrayList,String,CharSequence,...)

8.错误关注

a)java.lang.UnsupportedOperationException: addView(View, LayoutParams) is not supported in AdapterView

假如出现如上异常，首先要检测你的listview对应的布局文件中是否直接添加了其它的view子元素。

b)java.lang.NullPointerException(空指针异常)

一般要先定位出现此异常的行，然后检测此行代码中"."左边的对象是否为空，看一下在哪的赋的值。(Log跟踪一下)

c)java.lang.ClassCastException(类型转换异常);

Android Adapter Layout中适配器(Adapter)优化

1.重用适配器getView中的convertView参数对象？

1)convertView是什么？(列表项对象)

2)convertView的初始值是什么？(null)

3)convertView会在页面滑动时动态赋值？(滑出的item)

4)convertView如何实现重用？(判定convertView的值为空时在构建item)

5)convertView的应用场合？(数据量相对比较大的场合)

2.减少适配器getView方法中findViewById方法的执行次数

1)findViewById的作用？(获得item view中子元素)

2)如何减少对子元素的查找次数？(通过一个对象ViewHolder记录每个item)

3)ViewHolder(内部类)类似对象的应用场景

Android ListView 中的时间处理

1.事件源:ListView

2.事件类型

1)短按事件

2)长按事件

3)滚动事件

3.事件监听对象?(观察者对象，用于观察listview状态)

1)OnItemClickListener

2)OnItemLongClickListener

3)OnScrollListener

Android ListView 选择模式的设置

ListView的选择模式可以通过setChoiceMode方法进行设置，它的值有如下三中：

1)CHOIEC\_MODE\_NONE

2)CHOIEC\_MODE\_SINGLE

3)CHOIEC\_MODE\_MULTIPLE

当我们需要获得我们选择的值通常可以调用listView的

getCheckedItemPosition()

getCheckedItemPositions()

案例：day04\_listview\_02BaseAdapter(事件处理,选择模式)(让点击的item高亮[以不同的颜色]显示)

1)设置ListView的选择模式为单选模式

2)在适配器的getView方法中获得选择位置，然后根据位置的不同，进行view背景颜色设置

3)在listview的点击时间处理方法中刷新listview，可以调用适配器的notifyDataSetChanged();

Android ListView 中的item的排序？

ListView中item的排序在实践项目中会有很多应用场合，例如：按商品的价格或者销量进行排序。具体解决方案

一般时候借助Collections的sort方法对元素进行排序。(默认是升序排序)

分块显示，主要按照一定的规律让首字母相同的item显示到一个块中

1)分块应用场合？

1.数据量比较小(适合一次加载所有数据)

2.数据已经有序(一般是升序)

2)例如：

1.联系人信息

2.城市列表信息

3.商品品牌信息列表

3)分块应用的实现？

1.为每个item添加一个显示分块名称的TextView.

2.显示或隐藏一些item的分块名称信息。

a)显示分块中item位置最小的分块，其他都隐藏。

b)关键是如何获得每组分块中的最小位置。

1.获得没住分块的名称(标题的第一个字母)的ASCII

2.获得此大写字母在listview多个item位置中的最小位置

3.对分块内容实现右侧导航

a)目的:快速定位具体分块内容

b)手段：在分块基础上，在右侧再添加一个listview，并添加点击事件。

Android Spinner应用？

1.Spinner是什么？

1)ViewGroup

2)AdapterView

3)AbsSpinner(下拉列表)

2.Spinner应用场合

例如：选择颜色，选择学历...

3.Spinner应用的实现过程？

1)获得Spintner(findViewById)

2)构建Adapter

3)关联Adapter(setAdapter)

4.Spinner常用属性

1)android:prompt(提示标题)

2)android:spinnerMode="dialog"(显示样式)

Android ViewPager对象应用?

1.ViewPager是什么？

1)View

2)ViewGroup

2.ViewGroup应用场景

1)系统的新手指导页

2)小说分页阅读...

3.ViewPager应用的实现？(原理类似listview)

ViewPager中每个item构建都需要借助一个PagerAdapter类型的对象

其应用的实现步骤：

1)获得ViewPager对象？

a)首先在布局文件中进行声明(android.support.v4.view.ViewPager)

b)在java中实现findViewById获得ViewPager

2)构建适配器对象？

a)首先要写一个适配器直接或间接的继承PagerAdapter(抽象类)

b)构建PagerAdapter的子类类型

3)关联适配器? (setAdapter);

4)有选择性的添加监听器(OnPageChangeListener);

Android InputEvents(输入事件)

1.事件类型

1)onClick

2)onTouch

3)onKey

4)...

2.事件处理

1)事件源(在那个view控件上触发事件)

2)事件监听对象(观察者对象，用于观察或监听事件源状态的变化)

3)事件处理函数(回调函数，此函数在事件发生是会由系统回调执行)

3.Android 中事件的分发机制?

Android中任何view在处理事件首先会执行dispatchTouchEvent方法由此方法开始进行事件分发处理,

假如此view上添加了onTouch事件,那么事件分发时首先会执行OntouchListener的onTouch方法,

假如此onTouch方法返回值为true,表示事件被消费,后续其他事件不再继续处理,

假如此onTouch方法返回值为false,那么此时会调用view的onTouchEvent方法继续事件分发.

Android中在处理任何View的事件时,首先会执行这个view所在的容器(ViewGroup)的dispatchTouchEvent的方法进行事件分发,

然后才会执行其子元素的ddispatchTouchEvent方法进行事件分发.

假如不希望view的diapatchTouchEvent方法执行,可以重写view所在容器的onInterceptTouchEvent方法进行事件拦截,

此方法默认返回值为false,可以将其改为true(表示要进行拦截).

4.Android中事件分发相关方法?

1)dispatchTouchEvent(MotionEvent ev)事件分发

2)oninterceptTouchEvent(MotionEvent ev)事件拦截

3)onTouchEvent(MotionEvent ev)事件响应

Android是有Activity中的dispatchTouchEvent方法发起,然后传递到具体view

Android中的UI Components(不是view,但是可以呈现view)

Menu(菜单对象)

ActionBar(动作栏)

Dialog(对话框)

Notification(状态通知)

Toast(窗口上的吐丝通知对象)

Setting(...)

1)Menu

1.Menu是什么?

1)用于呈现view的一个UI控件对象

2)不是view

2.Menu应用场合?

以某种表现形式实现与用户的交互.

例如:可以在动作栏(ActionBar)上添加与用户交互的对象

3.Menu应用类型?

1)选项菜单(OptionMenu)

2)上下菜单(ContextMenu)

3)弹出式菜单(PopuMenu)

4)子菜单(SubMenu,每种菜单都可以创建子菜单)

4.选项菜单的事件处理?(OptionMenu)

1)创建:重写activity的onCreateOptionsMenu方法

a)res/menu/xxx.xml

b)R.menu.xxx

c)getMenuInflater().inflate(..........)

2)事件:重写activity的onOptionsItemSelected方法

a)int id=item.getItemId();

b)if(id==R.id.item01){....}

3)分组:对menu中的item可以进行分组统一处理

a)res/menu/xxx.xml中借助group标签对item进行分组

b)对item分组设置

扩展:使用java代码形式创建选项菜单的菜单项(menu.add(),menu.)

5.上下文菜单的创建及事件处理?(ContextMenu)

1)创建:onCreateContextMenu(基于view的长按事件)

2)注册:registerForContextMenu(view)

3)事件:onContextItemSelected(点击菜单项时执行)

当我们在一个view上注册了上下文菜单以后,长按此view会执行activity中的onCreateContextMenu方法,

在此方法中可以创建多个菜单项对象

其他核心API及方法:

1)AdapterContextMenuInfo(position,id,...)

2)item.getMenuInfo()

6.弹出式菜单的创建及事件:(PopuMenu)Android3.0

1)创建:参考构造方法(基于view的点击方法)

2)初始化菜单项:inflate

3)监听:setOnMenuItemClickListener

4)显示:show()

7.子菜单(每中菜单都有子菜单)(SubMenu)

创建方式参考选项菜单的子菜单.

2)ActionBar

1.ActionBar是什么?

1)UI控件(Component)

2)不是view

2.ActionBar应用场合?

1)呈现标题信息

2)程序选项菜单信息

3.ActionBar获得及相关设置?

1)获得:getActionBar()(在Activity主题设置上必须包含一个ActionBar)

2)设置:

a)隐藏或显示

a.1)hide()

a.2)show()

b)设置ActionBar的导航模式

b.1)setNavigationMode(ActionBar...)

c)设置ActionBar的自定义view

c.1)

d)设置ActionBar的zhut(Theme)

d.1)找到res/values/styles.xml

d.2)在styles.xml文件中写自己主题继承Android官方主题

d.3)在自己定义的主题中重写ActionBarStyle

d.4)让ActionBarStyle引用我们自己写的一个ActionBarStyle

d.5)我们自己写的ActionBarStyle要继承Android官方定义的ActionBarStyle样式

DrawerLayout(侧滑菜单对象)

DrawerLayout此对象实现了app中的侧滑菜单功能,此对象属于android.support.v4包中的一个布局对象,

在使用此对象是必须将此元素设置为xml布局的根元素,这个根元素内容自能有两个直接的view子元素,

其中一个是侧滑,一个是主体内容.

DrewerLayout应用步骤:

1.定义布局,设置其跟元素为android.support.v4.widget.DrewerLayout

1)定义子元素(借助此元素显示内容)

2)定义子元素(用于显示侧滑菜单内容)

说明:在定义第二个子元素时,此元素中一定要定义一个layout\_gravity属性,通过此属性指定侧滑从

那边开始,start一般表示从左侧侧滑(国家自适应,建议设置start不是left).

2.Java中初始化布局元素?

1)初始化ListView,添加点击事件

2)初始化DrawerLayout对象,添加监听事件()

Dialog

1.Dialog是什么?

1)UI Components(控件)

2)弹出式窗口(window)

2.Dialog应用场合?

以弹出窗口形式实现与用户交互

例如:

1)卸载应用是弹出的窗口

2)退出系统是提醒信息

3)弹出列表选择框

3.Dialog对象的构成分析?

1)标题按钮(图标)

2)标题(title)

3)内容(Content,...)

4)按钮(Buttons,...)

4.Dialog对象常用应用类型?

1)AlertDialog(提示对话框)

2)DatePickerDialog(日期对话框),TimePickerDialog,...

3)ProgressDialog(进度对话框)

4)自定义对话框(一般是直接或间接的继承AlertDialog)

5.AlertDialog对话框?

1.对象构建?(参考AlertDialog,Builder)

2.对象显示,销毁?(show,dismiss,cancel)

3.对象的监听?(DialogInterface,OnClickListener)

4.对象的样式?(style,...)

案例:

1)执行页面回退操作时以对话框形式给出提示.

2)以选择对话

6.自定义对话框

当Android官方给定的对话框不能满足我们需求时可以根据需要自己一个对话框，

此对话框一般要直接或间接的继承Dialog或AlertDialog然后重写相关方法（例如onCreate）。

1)构建Dialog类的子类

2)添加构造方法

3)重写相关方法(例如onCreate)

4)定义样式(theme)

扩展：通过修改activity的主题，让activity以对话框的形式呈现数据

5)扩展API?(Builder,DateUtils,SystemClock,...)

Notification对象

1.Notification是什么?

1)一个UI Component对象.

2)一个状态栏通知对象.

2.Notification应用场合?

Notification通常会与后台的服务(Service)绑定使用,当系统的后台获得一些信息以后,

以通知信息在系统的状态告诉用户.

例如:

1)你有一条短信,一个未接电话

2)你有一个系统更新

3)你有一个音乐真正播放

4)你有一个头条信息...

3.Notification对象的构成?

1)图标

2)标题

3)内容

4)...

4.Notification对象的构建,发送及销毁?

1)创建:(借助Notification或NotificationCompat中的内部类Builder创建)

a)Notification

b)Builder

c)NotificationCompat(内置一个向前兼容的Builder类型)Android4.0

说明:Notification对象在创建时必须要添加一个smallicon()方法添加一个小图标,

然后可以执行其他相关setXXX方法.

2)发送:

a)获得NotificationManager(通知管理器)

NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE);

b)发生:notify(int flags,notification)

c)销毁:cancel(int flags)

3)其他

a)通知的点击事件(PendingIntent)

setContentIntent

a)自定义Notification的布局(相关API RemoteViews)

setContent(RemoteViews)

对于自定义Notification的布局应用,添加需要事件处理可以在RemoteViews对象上添加.

Android 中UI的绘制?

1.Android UI的绘制概述?

Android中所以的可见元素都是通过一定的方法绘制处理.

2.Android UI对峙相关API?

1)Canvas(画布或画板回想)

2)Bitmap(背景位图对象)

3)Paint(画笔对象)

4)Path(路径对象)

5)...

3.Android UI绘制基本步骤?

1)构建一个背景Bitmap对象(Bitmap.createBitmap(...))

2)构建画板Canvas对象,并关联对象

3)构建画笔Paint对象

4)执行绘制动作(例如绘制一些几何图形,...)

5)显示绘制的对象(一般是将Bitmap放到ImageView上进行显示)

4.Android UI中View的自定义?

1)直接或间接继承View类型

2)重写相关方法(例如onMeasure(),onLayout(),onDraw(),dispatchDraw(),...)

Android中所以的View都是绘制出来的,其绘制过程为:

1)onMeasure(测量控件对象)测量对象的高度宽度

2)onLayout(布局控件)

3)onDraw(绘制控件,一般是非viewGroup控件重写此方法)

4)dispatchDraw(绘制子控件,例如:定做viewgroup通常会重写)

案例:

1)页面指示器的绘制?(CircleIndicator)

2)Tab指示器(RadioGroup)的自定义?

扩展:view中自定义属性的实现?

1)创建res/values/attrs.xml文件

<declare-styleable name="circleview">

<attr name="radius" format="integer" />

</declare-styleable>

2)布局文件对自定义属性赋值

a)定义命名空间

xmlns:app="http://schemas.android.com/apk/res/tarena.day09"

b)为属性复制

app:radius="16"

3)在view的构造方法中获得属性的值

TypedArray tArray = context.obtainStyledAttributes(attrs, R.styleable.circleview);

radius = tArray.getInteger(R.styleable.circleview\_radius, 10);// 没取到赋值10

tArray.recycle();

Android UI Animation

1.UI Animation是什么?

1)Android中的一组动画对象

2)Android中的一组用于丰富用户体验的API

2.UI Animation应用场合?

主要是用户改善用户体验,例如:

1)入场动画(页面进入时的动画)

2)出场动画(页面退出是的动画)

3)音乐真正播放的动画(歌词上下滚动,调动的音频,音乐正在播放,...)

4)...

3.UI Animation应用类型?

1)View Animation(视图动画)

a)Tween Animation(Alpha,scale,translate,rotate)

b)Frame Animation(res/drawable/...)

2)Property Animation(属性动画,Android3.0以后)

4.Tween Animation(补间动画)

1)基于xml方式补间动画的应用

a)定义:res/anim/xxx.xml

b)加载:Animation animation = AnimationUtils.loadAnimation(R.anim.xxx)

c)应用:view.startAnimation()

5.Frame Abimation(帧动画)

1)定义:res/values/drawable/xxx.xml

2)应用:

a)android:src="@drawable/xxx"

android:src="@drawable/frame\_01"

b)获得ImageView的drawable对象()

AnimatiorDrawable animationDrawable = (AnimatiorDrawable) imageView.getDrawable();

c)调用drawable对象的start方法

animatiorDrawable.start();

Android 属性(Property)动画

1.属性动画与view动画不同?

1)出现的时间不同(Property Animation3.0后)

2)定义的方式及API不同

res/animator/file.xml

3)访问的方式不同?

R.animator.file

4)作用对象不同?

a)view animatior直接作用于view对象

b)property animatior作用于任意对象的属性(一般也是view)

2.属性动画的应用场合?

属性动画一般应用于view animation不能实现的场合,例如直接改变某个属性的值,

或者逐渐修改对象的背景色(渐变,例如从黄色变为绿色).

3.属性对象具体定义及相关应用?

1)xml定义:res/animation/file

2)xml应用:AnimatorInflator(this,R.animator.file)

3)相关API

a)ValueAnimator(不能直接作用于对象的属性,但可以监听其值的变化)

b)ObjectAnimator(直接作用于对象的属性,可以就爱那天其值的变化)

c)AnimatorSet(实现一个属性动画集,例如多个属性动画放在一起的动画)

Android 中的数据存储概述?

1.Android 中的UI的作用是什么?

1)通过view呈现数据

2)实现于用户的交互

2.Android 中的数据来源?

1)内存(数据断电(重启,关机)则丢失)

2)外存(数据可以持久保存在手机的sdcard)

3)网络(数据可以持久存储在云端、服务器)

3.Android 中的数据格式?

1)非结构化数据(图片,音频,视频,普通文本...)

2)半结构化数据(xml)

3)结构化数据(数据库以二维表的形式对数据进行存储:SQLite)

Android 中会结合数据特点采用不同的数据读写方式.

4.Android 中的数据存储类型?

1)直接I/O存储(直接以I/O流的形式对数据进行存储)

2)偏好设置存储(以键值对Key/Value信息的形式对数据进行存储)

3)SQLite存储(以二维表的形式进行存储)

Android 中的外部存储?

1.Android 外部存储概述?

Andorid外部存储一般指外置sdcard的数据存储,外部存储的文件一般比较大,例如图片,视频,音乐,...

而且大多数是非结构化数据.

2.Android 外部存储的数据权限?

1)共有数据(一般指所以APP都可以访问到的数据)

2)私有数据(一般指本APP内部对象可以访问的数据)必须添加权限

3.Android 外部存储应用的实现及相关API?

3.1相关步骤

1)获取外置sdcard的存储状态(是否有可用的sdcard)

2)获取sdcard的存储目录

3)设置读写sdcard的权限(假如是私有数据必须设置)

4)获取外置sdcard的空间大小(尤其是对写的动作)

5)通过I/O流的方式直接读写sdcard.

3.2相关API

1)Environment(状态,共有目录)

a)getExternalStorageState()

b)isExternalStorageRemovable()

c)getExternalStorageDirectory()

d)getExternalStoragePublicDirectory()

2)Context(应用私有目录)

a)getExternalStorageDirectory

b)getExternalStoragePublicDirectory

3)StatFs(目录控件大小)

a)getBlockCount()

b)getFreeBlocks()

c)getAvailableBlocks()

d)getBlosckSize()

4)I/O(直接读写数据)

3.3相关案例?

1)例如将一个字符串写到外置sdcard的目录中

a)公有目录(所以APP都可以访问,应用卸载数据不会删除)

b)私有目录(应用在卸载同时文件和数据会被删除)

2)例如读取sdcard中的所以音乐文件(以列表形式显示)

4.Android 中的内部直接I/O存储实现步骤及相关API?

4.1相关步骤

1)获得内部存储目录(data/data/项目包/...)

2)获得目录空间大小(是否有足够空间可以存储我们的数据)

3)直接I/O读写数据?(数据为应用程序私有数据)

4.2相关API

1)Context

a)getFilesDir()

b)getCacheDir()

c)openFileOutput()

d)openFileInput()

2)StatFs

3))I/O

4.3相关案例(直接I/O读写字符串)

说明:内部存储数据在应用卸载是会被删除.

5.Android 中内部存储中的偏好设置存储的实现步骤及相关API?

5.1概述

偏好设置存储以Key/Value形式对用户的一些行为信息,系统的状态进行相关记录(例如记住用户手机号,密码,...)

5.2步骤(写数据)

1)获得SharedPreferences对象

2)获得Editor对象(负责编辑,假如是读数据不需要获得此对象)

3)通过Editor方法存储数据(暂时存在内存)

4)通过Editor对象的commit方法持久化数据(将数据写到外存)

5.3相关API

1)Context(通过此对象获得SharedPreferences对象)

2)SharedPreferences(通过此对象获得数据或者获得Editor对象)

3)Editor(通过此对象实现对数据的编辑操作)

6.Android 内部存储中的SQLite存储的实现步骤及相关API?

6.1概述

SQLite本身是一个开源数据库系统DBMS(Database),底层借助C/C++.

实现,体积非常小,非常时候于一些便携式设置,在Android和IOS系统中读默认集成了这样的一个数据库系统.

SQLite实现了对Android系统中结构化数据的更好存储,保证了数据的安全,稳定,高效.

6.2步骤

1)创建或打开数据库

2)创建对应表(假如表不存在)

3)操作表中的数据(insert,update,delete,select)

4)释放数据库资源(释放内存)

数据库(DB):用于存储数据的数据仓库(一个SQLite系统可以管理多个数据库)

表(table):数据库中最基本的存储单元(行,列),数据库所以的数据都是要存储在表中的

6.3相关API

1)Context

2)SQliteDatabase,SqliteOpenHelper,Cursor,SimpleCursorAdapter,...

3)SQL(结构化程序语言):DML,DDL,DCL

DML(数据操纵语言)

DDL(数据定义语言)

DCL(数据控制语言)

Android 中的线程应用基础?

1.Android 中的线程是什么?

线程是进程中的一个顺序的执行流,是进程中的最小执行单位,在Android中每个APP默认都会运行在一个独立的进程中,此进程

由独立的内存空间,并且可以启动多个线程并发执行进程中的多个操作(例如在下载音乐的同时,也可以听音乐),这些线程会

共享同一个进程的内存空间.

2.Android 中的线程并发?

1)宏观上的并发:多个线程好像是在同时执行.

2)微观上的并发:多个线程其实是在切换执行.

例如:(听音乐的同时可以下载操作其他内容)

3.Android 中多线程并发执行的优势,劣势?

1)优势:改善用户体验,提供系统的执行效率.

2)劣势:会增加程序员编写,调试,以及维护代码的难度.

4.Android 中线程对象的创建以启动?

Andorid 中线程的类型为Thread类型,在构建线程对象是要参考Thread类型的构造方法,启动线程要调用Thread类型的start方

法,线程一旦启动并获得了CPU就会执线程对象的run方法,假如我们自己有任务要执行,通常是要交给线程对象的run方法.

线程中常用的构造方法:

1)Thread():假如采用此方法构建线程对象,通常是构建Thread类的子类对象,并重写run方法.

2)Thread(Runnable r):假如采用此方法构建线程对象,在构造方法中需要传入一个Runnable类型的对象.

FAQ?

1)线程对象可以直接通过run方法启动吗?不可以

2)线程对象执行了start方法会立刻执行run方法吗?不确定

3)线程对象可以多次调用start方法启动吗?不可以

4)一个线程中的多个任务是顺序执行吗?是

5.Android 中线程对象状态及相关方法?

5.1线程状态

1)新建状态(new Thread(){...})

2)就绪状态(start())

3)运行状态(run()正在运行)

4)阻塞状态(sleep(),...)

5)死亡状态(run()运行结束)

5.2线程方法?

1)start()启动线程

2)run()执行任务

3)sleep()让线程休眠(此时线程会让出CPU而阻塞状态)

4)currentThread()获得当前线程

5)getName()获得线程名字/setName()

6)isAlive()判定线程是否还活着

7)setPriority()设置线程优先级,在android中一般建议使用Process类设置线程优先级Process.setThreadPriority(int i)

8)getPriority()获得线程的优先级

9)join()调用此方法的线程会优先执行,假如指定了事件此线程会优先执行指定的那么长时间

10)setDaemon()设置线程为守护线程(服务线程),一定要咋启动线程进行设置,当没有其他线程在执行是守护线程会自动终止.

11)...

Android 中的线程同步?

1.为何线程同步?

线程同步是多个线程在共享数据集上的互斥于协作(通讯)

互斥:多个线程在共享数据集上排队执行.

协作:构建于互斥基础上,是多个线程在共享数据集的通讯.

2.线程同步应用场合?

1)保证数据的安全

2)保证业务的合理

3.线程互斥的实现?

1)同步代码块

2)同步方法

a)synchronizend void method01(){}//默认对象锁额为this

b)static synchronized void method02(){}//默认对象锁"类名.class"

多个线程在并发执行时会在同步代码块或同步方法上排队执行.

案例:1)模拟多个线程卖共同的100张票 2)模拟一个计数操作

回顾:

1)StringBuffer是一个线程安全的StringBuilder?

a)在一个线程安全的环境进行字符串的连接是使用StringBuilder还是StringBuffer?StringBuffer

2)Vector是一个线程安全的ArrayList,但性能不高.

a)在线程不安全的环境假如要使用ArrayList,建议使用Collections集合将其转换线性安全的集合,调用

此synchronizedList方法实现.或者直接使用CopyOnWriteArrayList.

3)HashTable是一个线程安全的HashMap,但性能不高.

a)在线程不安全的环境假如要使用HashMap,建议借助Collections集合将其转haunted线性安全的集合,调

用此synchronizedMap方法实现.或者直接使用ConcurrentHashMap(此对象采用局部加锁算法保证数据安全)

4.线程的协作？

线程协作是多个线程在共享数据集上借助Object类中的wait,notify,notifyall方法实现线程之间的交互。

含义说明:

1)wait 表示等待，调用此方法的线程会阻塞同时让出CPU

2)notify/notifyall 表示通知，调用此方法的线程会通知具备相同锁且处于等待状态的线程。

应用说明:

1)wait,notify/notifyall必须应用在同步代码块或同步方法中

2)wait,notify/notifyall必须由对象锁(同步锁)调用。

案例：生产者,消费者应用模型。和单例模式(day13\_thread\_04(homework))

ConcurrentHashMap,ConcurrentArrayList,BlockingQueue

1)为什么要加锁?(多个线程有共享数据集，并且共享数据集上的操作又非原子操作)

a)取的数据重复(一个数据会被取多次)

b)有的值取不到(多个消费者，.....)

2)为什么要通讯?(满的时不要放了，空的时不要取了)

a)对于生产者满了则等待，没满则继续放。

b)对于消费者空了则等待，没空则继续取。

Android 中的线程消息模型?

1.FAQ?

1)线程的生命周期何时正常结束?(run方法运行结束)

2)如何让线程对象迭代执行多个任务?(让线程对象内置一个任务队列)

new Thread(){

BlockingQueue<Runnable> queue = new ArrayBlockingQueue<>(3);

public void run(){

while(true){

Runnable r = queue.take();

r.run();

}

}

}.start();

3)Android中线程的分工或者说线程的应用机制?

a)主线程负责执行UI操作(UI输入,UI事件,UI更新)

b)工作线程负责执行耗时操作(访问网络,下载,...)

这样分工的目的:尽量不要阻塞主线程.

4)Android中的线程是如何轮循执行多个任务的呢?

a)借助消息模型

2.Android 中的消息模型概述?

Android中的消息模型是线性轮循机制的一种实现,是对线程同步的一种封装,是实现线程通讯的一种手段(例如:工作

线程交给主线程一些任务执行或主线程交给工作线程一些任务去执行等).尽量实现线程之间的无阻塞(不阻塞主线程)通讯.

3.Android 中消息模型相关对象(四大核心对象)?

1)Message(消息-数据的载体,好比是一个任务)

2)MessageQueue(消息队列-多个消息对象的载体)

3)Looper(迭代器-用于迭代消息队列)

4)Handler(处理器-负责发送并处理消息)

4.Android 中消息模型伪代码实现?

class Message{

Object obj;用于存储对象类型数据

int what;表示你要做什么

...

}

class MessageQueue{

Object[] messages;

void put(Message msg){}

Message take(){}

}

class Looper{

MessageQueue msgQ;

void loop(){

while(isLoop){}

}

void cancel(){}

...

}

class Handler{

void sendMessage(Message msg){}

void handleMessage(Message msg){}

...

}

在Android应用中一个线程最多有一个消息队列,最多有一个Looper;

在Android应用中这个APP启动时,会创建一个进程,并开启一个主线程,在此主线程中会创建一个消息队列,创建一个Looper,

当我们执行APP中的任何一个UI操作时,底层都在给主线程发消息,主线程收到消息以后,会依次处理多个操作.

5.Android中消息模型应用案?

1)工作线程发消息给主线程

2)主线程发消息给主线程

3)主线程发消息给工作线程

例如:

1)listview显示工作线程加载的数据?

2)播放音乐的同时更新SeekBar?

3)连续点击两次实现页面的退出?

4)APP系统首页5秒以后进入下一个页面?(广告)

5)交给工作线程执行多个下载操作?(主线程发消息给工作线程)

FAQ?

1)主线程默认存在一个Looper,一个消息队列.

2)工作线程默认没有Looper,假如希望工资线程启动以后创建Looper可以考虑使用HandlerThread/

3)给谁发消息就获得一个与谁的Looper相关联的hander.

4)在哪个线程处理消息取决于handler关联的looper.

API?

1)Message

2)MessageQueue

3)Looper

4)Handler

5)HandlerThread

Android 中的异步任务对象?(AsyncTask)

1.AsyncTask是什么?

1)Android 中的一个异步工具类.

2)Android 中的消息模型异步对象.

此类实现了对异步逻辑的封装,有些方法运行在工作线程,有些方法运行在主线程,并且封装了这些方法的执行流程.

2.AsyncTask应用场合?

简化消息模型中handler发送及处理消息的逻辑步骤.对于一些简单的消息发送及处理可以考虑直接使用异步任务，

例如通过异步任务执行多个文件的下载操作。

3.AsyncTask具体应用?

1)编写AsyncTask类的子类类型,然后重写相关方法.

2)构建异步任务对象.

3)启动异步任务,执行异步处理逻辑.

4.AysncTask中的相关方法?

1)executeXXX(...)开启异步任务

2)doInbackgroud(...)执行耗时操作

3)onPostEcxecute(...)耗时操作执行结束,更新UI

4)onPreExecute(...)初始化操作

5)publishProgress(...)发布进度

6)onUpdateProgress(...)更新进度

7)cancel(...)尝试退出任务(true表示正在线程的进行,false表示要等任务结束)

8)isCanceled()当执行了cancel方法,此方法的放回值为true

9)...

5.AsyncTask应用相关FAQ?

1)AsyncTask对象的创建必须在主线程.

2)AsyncTask对象的启动必须在主线程.(例如执行execute方法)

3)每次执行新的任务必须重新创建AsyncTask对象

4)异步任务对象通知execute方法执行任务是顺序执行的.

5)异步任务对象通过executeOnExecutor方法执行任务,任务是并发执行的.

Android 中的线程池

1.线程池是什么?

1)内存中的一块区域.

2)此区域允许存储多个线程对象.

2.线程池的应用场合?

1)多个任务需要并发执行

2)线程对象要运行重复使用

例如:多个文件的并发下载

3.Android 中的线程池的应用及API?

1)创建线程池(开辟内存区域)

a)线程池类型(Executor:ExecutorService,ScheduledExecutorService,ThreadPoolExecutorService)

b)线程池工具类(Executors)

2)启动线程执行任务

a)调用线程池的execute方法

b)调用线程池的submit方法

3)线程池对象的其他操作(例如关闭线程池,...)

4.线程池在Android异步任务(AysncTask)中的应用?

Android 线程应用导致的内存泄露问题?OutOfMemory[OOM]

1.何为内存泄露?

当内存泄漏是一个对象不用了,但还占有着系统内存,导致内存不能及时释放,这个现象我们称之为内存泄漏.(不是溢出)

一个对象在内存中何时标识为垃圾?(此对象没有任何强引用引用者)

2.Android 线程应用可能导致的内存泄漏?

1)线程对象以非静态内部类的形式出现在Activity中.

2)线程对象强引用了Activity中的成员(例如一些view对象)

3.Android线程应用如何解决内存泄漏问题?

1)线程内部类,异步任务内部类尽量使用static修饰

2)线程内部类,异步任务内部类尽量弱引用外部类对象(WeakReference弱引用)[SoftReference软引用]

3)handler内部类尽量使用static修饰

Android 中图片的异步并发加载// Caused by: java.lang.OutOfMemoryError

1.图片的压缩(内存溢出[OOM])

2.图片的异步加载(耗时操作放在工作线程)

3.图片的并发异步加载(并发可能会有那些问题)

1)实现每个item中imageview与异步的绑定

a)构建一个AsyncDrawable对象,通过此对象绑定异步任务

b)实现imageView与AsyncDrawable对象的绑定

2)假如不需要执行加载时要考虑如何退出这个异步任务

4.图片的缓存问题(内存缓存LruCache,外存缓存)

1)为什么要缓存(已经加载过的数据劲量不要再从外存加载了)

2)内存缓存的实现(LruCache)

3)外存缓存的实现(DiskLruCache)

5.图片的显示(ImageView):在主线程执行显示动作

1)判定ImageView是否是空

2)判定返回的结果(bitmap)是否为空

3)判断imageview上是否绑定了异步任务

4)判断imageview上绑定的异步任务是否是当前的异步任务

5)显示图片

Android 核心应用组件(Component)?

1.ContentProvider内容提供者

2.Service后台服务对象

3.BroadCastReceiver广播接收者

4.Activity(Fragment)活动,UI接收对象

Android ContentProvider

1.ContentProvider 是什么?

1)Android 中核心应用组件(Component)之一

2)Android 中的内容提供者?

2.ContentProvider 应用场合?

ContentProvider 对象实现了APP内部数据的对外(可能是本进程的其他组件,也可能是跨进程的组件)共享.

例如:

1)手机联系人数据对外共享.

2)手机相册中的数据对外共享.

3)...

3.ContentProvider 应用的构成?

1)Data(要对外共享的数据)

2)ContentProvider对象(是对外共享)

3)URI(统一资源标识,用于标识具体资源)

4)ContentResolver对象(可以简单理解为ContentProvider的客户端对象)

4.ContentProvider 应用的具体实现?

1)ContentProvider端的实现(暂且理解为服务端):数据提供端

a)数据(Data)

b)编写ContentProvider(直接或间接继承ContentProvider类)

c)注册ContentProvider(AndroidManifest.xml)

2)ContentResolver端的实现(暂且理解为客户端):数据访问端

a)获得Context对象

b)获得ContentResolver对象

c)获得要访问的contentProvider端的URI

d)执行ContentResolver对象的相关方法进行数据处理.

说明:ContentResolver对象方法中的方法名于contentProvider端的对象方法基本是一致的,当我们通过contentResolver对象的

某个方法时,底层会根据方法名及参数设置访问contentProvider端对象的方法.

说明:现阶段重点掌握ContentProvider的访问流程,了解ContentResolver对象方法的构建.

5.访问Android 系统提供的ContentProvider?(重点)

1)访问系统提供的媒体信息(音频,视频,图片)

2)访问系统提供的联系人消息...

步骤:

1)获得ContentResolver

2)获得URI(Media,Phone,...)

3)执行查询(query(...))

6.CursorLoader在异步访问provider中的应用?

CursorLoader底层默认采用异步加载机制访问数据,并可以对数据提供缓存机制,以提供对数据的访问效率.假如在Activity

中要访问ContentProvider中提供的数据,建议使用CursorLoader.

CursorLoader对象的使用步骤?

1)获得LoaderManager(getLoaderManager(id))

getLoaderManager().initLoader(0, null, this);

2)初始化CursorLoader(初始化的同时注册一个回调监听对象)

3)执行监听对象中的相关方法

LoaderCallbacks<Cursor>

a)onCreateLoader(初始化Loader是执行)

b)onLoadFinished(数据加载完成以后执行)

c)onLoaderReset(Loader重置或销毁时执行)

7.ContentProvider应用FAQ?

1)ContentResolver对象应用场合?(应用数据对外共享,例如微信访问联系人数据)

2)ContentProvider对象应用优势?(提供了对数据的统一访问方式)

3)ContentProvider对象对外共享数据的格式?(数据库,文件assert,...)

4)ContentResovler对象的作用?(是ContentProvider对外的一个窗口)

5)URI是什么,在ContentProvider应用中充当的角色?(统一资源标识,用于标识ContentProvider对象对外的一个地址)

6)CursorLoader应用优势?(异步加载,防内存泄漏,数据缓存,便于用oaderManager管理)

Android Service 组件?

1.Service是什么?

1)Android中核心的组件应用之一.(生命周期方法)

2)Android中的Context(资源访问能力)

3)Android中的应用服务(此服务不能呈现view,工作于后台)

2.Service应用场合?

Service通常用于执行一些"长时间"的后台操作(这些耗时操作也是要启动工作线程的),例如下载,音乐的播放等.

FAQ?Service中启动工作线程执行长时间耗时操作相对于Activity启动工作线程执行耗时操作有什么优势?

Service服务所在的进程运行后台时生命力更加顽强.

\* Android 中进程的类型及生命力(如下由高到低)?

1)前台进程(正在于用户交互的进程):生命力最顽强,不允许被kill

a)有activity正在前台运行(正在与用户交互)

b)有通知(Notification)绑定了正在运行的service

2)可见进程(进程可见,但不可操作)

3)服务进程(有service正在后台运行,且进程不可见)

4)后台进程(后台没有service运行,前台又没有可见组件在运行)

5)空进程(没有任何组件在运行的进程)

Android 系统在内存步骤是会选择一些优先级第的进程执行kill操作.

3.Service的应用类型?

1)启动模式的Service

2)绑定模式的Service

3)混合模式的Service

4.Service应用的构建?

1)创建Service(直接或间接的继承Service)

2)注册Service(在AndroidManifest.xml中进行注册)

3)启动或绑定Service(startService,bingService,...)

4)在Service对象相关生命周期方法中执行业务.

5.Service启动模式的应用?

1)Service启动模式的相关操作:

a)创建Service,重写相关方法

b)并注册Service(AndroidManifest.xml)

c)启动Service:startService(Intent);

d)停止Service:stopService(ntent);

2)启动模式Service的生命周期方法:

a)onCreate():第一次启动是执行

b)onStartCommand():每次启动都会执行

c)onDestory():service销毁时执行

3)启动模式Service中onStartCommand方法的返回值:

a)START\_STICKY:表示被非正常kill以后会自动重启service

b)START\_STICKY\_COMPATIBILITY:是START\_STICKY的一个兼容模式,但进程被非正常kill以后不保证能重新启动service

c)START\_NOT\_STICKY:表示被非正常kill以后不会重启service

d)START\_REDELIVER\_INTENT:表示service被非正常kill以后会自动重启并重新传递intent

4)启动模式Service中onStartCommand方法的参数?

a)intent:传值

b)flags:service的启动方式,intent是否是重新传递的

c)startId:服务第几次启动的次数,stopSelf

6.IntentService在启动模式中的应用?

IntentService主要应用于启动模式,此对象在启动后会自动创建一个工作线程,我们可以在此工作线程执行耗时操作,

此耗时操作执行完成,Service会自动关闭.

在构建IntentService应用时,我们需要构建此类的一个子类类型,然后重写onHandleIntent方法(此方法运行在工作线程),

在此方法中执行我们的业务处理就可以了.(onHandleIntent运行在工作线程)

7.Service绑定模式的应用?

1)绑定模式Service概述

绑定模式Service相当于客户端(需要获得service的一端)与Service之间建立一个长连接(ServiceConnection),然后此客户端就可

以获得Service端对象,进而操作Service.

FAQ?

a)Service对象可以new创建吗?不能

b)Service对象有谁创建?底层框架

c)Activity中如何获得Service对象?通过绑定模式

2)绑定模式Service应用的构建?

a)创建Service重写生命周期方法

b)注册Service(AndroidManifest.xml)

c)绑定Service(bindService)

d)解绑Servoce(unBindService)

3)绑定模式执行需要用到的API:

a)Context

b)Intent

c)Service

d)ServiceConnection

e)IBinder(Binder)

4)Service对象的生命周期方法?

a)onCreate(Service创建时执行)

b)onBinder(Service第一次绑定时)

c)onUnbind(Service解除绑定时)

d)onDestory(Service销毁时)

FAQ?

1)假如绑定模式中想重复点击播放按钮播放音乐,如何实现?

2)绑定模式Service可能会出现内存泄漏问题吗?(可能,一般建议在哪个对象中执行的绑定,在此对象销毁之前解除绑定)

8.Service混合模式的应用?(加载广告的时候,就可以开启Service进行加载数据,从而节省时间,提供用户体验)

1)Servie混合模式的概述?

Service混合模式是既有service启动又有Service的绑定,在混合模式中只有Service执行了停止和解除绑定以后Servcie才会销毁.

2)Service混合模式应用的实现?

a)构建Service,重写生命周期方法(一般是启动和绑定的所以生命周期方法)

b)注册Service(AndroidManifest.xml)

c)启动Service,绑定Service

d)解除绑定Service,停止Service

3)Service混合模式应用的生命周期?

a)创建一次(第一次启动或绑定是创建)

b)销毁一次(执行了解除绑定和停止操作)

Android 中Service在跨进程(IPC)中的应用?

1.跨进行启动,停止Service?

1)为远程service注册action(清单配置文件)

<service

android:name="com.example.remoteserve.RemoteService"

android:enabled="true"

android:exported="true" >

<intent-filter >

<action android:name="action.REMOTE\_SERVICE" />

</intent-filter>

</service>

2)客户端借助远程Service的action执行启动和停止操作.

startService(new Intent("action.REMOTE\_SERVICE"));

stopService(new Intent("action.REMOTE\_SERVICE"));

2.跨进程绑定,解绑Service?

1)远程service注册action(清单配置文件)

2)客户端借助远程Service的action执行绑定和解绑操作.

a)在执行远程绑定是,要获得绑定的对象需要借助AIDL

b)AIDL是什么?(Android接口定义语言)

c)AIDL应用场景?(跨进程实现Service绑定通讯)

d)AIDL应用的实现?

d.1)在远程service端编写AIDL文件(于Java语法基本相同,文件民后缀aidl)

d.2)此时保存项目,项目会自动编译aidl文件,并在src/gen目录生成对应的java文件

d.3)将远程service端生成的aidl文件对应的Java代码拷贝到本地客户端(与包一起拷贝)

d.4)在执行绑定操作就可以使用aidl文件生成的Java类来获得绑定的Service

说明:假如有多个进程要跨进程绑定一个远程的Service对象,此时要考虑线程安全.

3.Messager对象在跨进程service通讯中的应用?

Message对象应用于Service的绑定模式,底层借助AIDL实现了进程间的通讯机制,我们可以借助Messager对象在绑定模式中

实现继承进程消息传递.

应用步骤:

1)远程Service端在onBind方法中返回new Messager(handler).getBinder();

2)客户端在onServiceConnection方法中

a)Messager mGer = new Messager(binder);

b)mGer.sendMessager(msg);给远程发消息

Android BroadcastReceiver

1.BroadcastReceiver是什么?

1)Android中的核心应用组件(生命周期方法)

2)Android中的广播接收者

2.BroadcastReceiver应用场合?

BroadcastReceiver是实现进程间一对多消息传递的一种非常巧妙的实现.当然也可以实现进程内部的通讯(通常会用本地进程)

常用场合?

1)手机系统启动完成会发出广播,我们应用可以接收广播.(开机自动启动应用)

2)系统在启动拨号程序以后也会发出广播,我们也可以接收广播

3)系统在接收短信或电话时会发出广播,我们也可以接收广播拦截信息(黑名单)

4)系统断网以后会向应用发出广播,我们也可以接收广播执行处理

5)系统电量不足...

3.BroadcastReceiver应用类型?

1)普通广播(没有指定接收顺序)

2)有序广播(按优先级接收广播)

4.BroadcastReceiver应用的构建?

1)编写广播接收者,直接或间接继承BroadcastReveiver.

2)注册关闭接受者对象(静态注册[xml],动态注册[java])

3)接收系统或应用发出的广播

5.BroadcastReceiver中普通广播的应用?

1)编写(extends BroadcastReceiver{})

2)注册(register)

a)静态注册(AndroidManifest.xml)

b)动态注册(Java)

6.BroadcastReceiver中广播接收者的生命周期?

BroadcastReceiver对象的生命周期由注册方式决定,静态注册每次接收[新]的广播都会创建新的对象,动态注册取决于自己何时构建,

何时解除注册.

广播接收者在接收到广播以后会执行onReceive方法,对于静态注册onReceive方法执行结束此对象的生命周期也就结束了.

7.BroadcastReveiver的有序广播的实现?

有序广播一般是按照优先级接收广播,发送广播时候需要借助sendOrderBroadCast(...),例如短信拦截,电话拦截,...

有序广播在发送时候需要指定:

1)intent(隐式意图):new Intent("action.A01");

2)permission(权限):声明权限,使用权限

声明权限:<permission android:name="app.permission.A01" />

使用权限:<uses-permission android:name="app.permission.A01" />

有序广播中接受者优先级的指定:在IntentFilter中进行配置,范围一般是-1000~+1000;

8.BroadcastReveiver应用的本地化实现?

本地化实现一般指的是进程内部广播的发送于接收,其他进程的广播接收者是收不到数据的,本地广播的实现需要借

助LocalBroadcastManager实现.

应用步骤:

1)LocalBroadcastManager对象的获得(getInstance)

2)LocalBroadcastManager对象实现广播注册

3)LocalBroadcastManager对象实现广播发送

4)LocalBroadcastManager对象实现解除注册

说明:一个进程多个组件之间假如需要使用广播建议使用本地广播.

FAQ?

1)一个广播发出者是否可以对应多个广播接收这?(可以)

2)一个广播接收者任何接收系统发出的广播?

a)注册

b)添加权限(use-permission)

3)任何让具备某些权限的广播接收者接收我们的应用广播?

a)发送广播:sendBroadcast(intent,permission)

b)声明权限:(permission)

4)如何让一个广播只在本进程内部传递?(LocalBroadcastManager)

5)广播的拦截只是针对有序的广播吗?(是)

6)任何设置广播接收者的优先级?(静态,动态)[IntentFilter]

7)广播接收者对象常用的实例方法?

a)getResultData()取得数据

b)setResultData()设置数据

c)aboutBroadCast()终止

8)动态注册广播接收者在不要的时必须解除注册吗?(是)[OOM]

Android Activity组件?

1.Activity是什么?

1)Android中的一个核心应用组件(周期方法)

2)Android中的Context(资源访问能力)

3)Android中的Controller(控制器)

2.Activity应用场合?

1)呈现view,提供界面以及与用户交互

2)访问model

3.Activity应用的构建?

1)编写(直接或间接继承Activity)

2)注册(AndroidManifest.xml)

3)启动(startActivity(...),startActivityForResult(...))

4.Activity的生命周期及相关方法?

a)onCreate(创建时执行)

b)onStart(启动时执行)

c)onResume(运行时执行)

d)onPause(暂停时执行)

e)onStop(停止时执行)

f)onDestory(销毁时执行)

g)onRestart(重新启动时执行)

5.Activity的状态保存?

1)保存的目的?(进行数据恢复)

2)何时保存?(Activity销毁之前,例如onPause,onStop,onSaveInstanceState)

3)保存到哪?(内存[Bundel],外存[SharedPreferences,SQLite])

4)何时恢复数据?(Activity运行之前,例如onCreate,onStart,onRestoreInstanceState)

横竖屏切换的时候,应该保存数据,否则有可能会出现界面错乱的情况.

Android 所在的任务栈?

Android 中所以的Activity都是存储在任务栈(又叫回退栈)中的,一个Android APP可能会存在多个任务栈,每个任务都有一个唯一的id,

可以在activity中通过getTaskId()方法获得.我们第一次启动的activity的肯定会存在任务栈的栈底,正在于用户交互的activity位于任务

栈的栈顶,在同一个任务栈中可能会存在类型相同不同对象的activity实例.

Android 的启动模式?

Activity的启动模式会影响activity对象的构建及在任务栈中的存储,通常会在清单配置文件的activity元素中借助lanuchModel属性

进行配置,其值如下:

1)standard(标准模式):每次启动activity都会创建新的实例

2)singleTop(栈顶模式):activity为栈顶时不再创建新的实例

3)singleTask(单任务模式)每个任务栈activity的实例只能有一份

4)singleInstance(单实例模式)此activity在内存只有一份且独享一个任务栈

应用场合:

1)standard:小对象,对象占用资源比较小

2)singleTop:通过service或notification启动时

3)singleTask:大对象,对象占用资源比较多(例如浏览器,系统主页)

4)singleInstance:跨进程共享(例如拨号程序,相机)

说明:

1)对于singleTop是栈顶的activity或singleTask模式的activity再次启动是虽然不会创建新的对象但会执行activity

的onNewInternt方法

2)在activity外界启动activity(例如service中)一般要在intent中添加一个标记,

例如:intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

Android 中Activity的亲族设置?

Android的每个Activity都有一个亲族值,此值需要借助清单配置文件中Activity的taskAffinity属性进行设置,此值的设置会影响

Activity对象在任务栈的存储位置.

1.Activity中亲族值应用相关规则:

1)每个任务栈都有一个亲族值,此值于栈底activity的亲族值相同

2)每个Activity都有一个亲族值,此值默认于application的亲族值相同

3)清单配置文件中application对象假如没有设置亲族值默认于根元素的package相同

2.Activity亲族值&SingleTask模式:

1)当启动一个SingleTask模式的Activity时,首先会在内存检测有没有activity亲族值相同的任务栈,假如有则直接存储,

假如没有则创建新的任务栈进行存储.

2)当通过SingleTask模式的activity启动其他非SingleInstance和SingleTask模式的activity时,这些activity默认会与

本activity存在同一个任务栈.

3.Activity亲族值&SingleInstance模式:

1)当通过SingleInstance模式的activity启动其他模式的activity时,首先检测在内存有没有亲族值相同的任务栈,假如

有则直接存储,没有则创建新的任务进行存储.

Android 中Activity在跨进程中的启动?

Activity的跨进程启动通常在不同APP之间,例如启动相机程序等.

Activity的跨进程启动方式:

1)在activity注册是添加默认分类(Category),注册隐式意图

2)通过startActivity(隐式意图对象)的方式启动activity

Android 中Activity对象的销毁?

在Android应用中假如Activity对象不用了,或者说我们的系统在执行退出操作时,我们要销毁activity,销毁所以activity,

销毁所以组件对象.

1.finish():关闭当前activity

2.finishAffinity():关闭当前activity所在的任务栈

3.Process.killProcess(int pid):关闭当前activity所在的进程(一般指当前进程)

4.killBackgroundProcesses(getPackageName());关闭本进程或本进程外的服务进程

1)ActivityManager aManager = (ActivityManager)getSystemService(Context.ACTIVITY\_SERVICE);

2)aManager.killBackgroundProcesses(getPackageName())

5.System.exit(0);退出当前进程[不推荐]

Andorid 中Activity的其他配置?

1.theme(主题)

2.icon(图标)

3.screenOrientation(横竖屏切换) android:screenOrientation="portrait" 不会随屏幕横竖切换

4.android:configChanges="orientation|screenSize"屏幕发生变化,内容发生变化,Activity不会重写创建

或者Java中重写onConfigurationChanged()方法

Android 中Activity的常用对象?

1.Activity

2.ListActivity(内置一个ListView)

3.FragmentActivity(提供了FragmentManager对象的相关应用)

Fragment对象?

1.Fragment是什么?

1)Activity中的一个模块化组件.(Anfroid3.0以后推出)

2)Activity用于定义局部UI对象的一个控件.

2.Fragment应用场合?

Fragment对象主要用于更加灵活的定制Activity中的UI对象

简化代码编写,提高UI对屏幕尺寸的适配.

3.Fragment应用构建?

1)编写fragment(直接或间接的继承fragment对象)

2)注册Fragment(静态注册,动态注册)

a)静态注册需要在activity对应的布局文件中借助fragment元素实现

b)动态注册需要借助activity中的FragmentManager对象实现

3)运行Fragment(加载,关联等fragment对象)

4.Fragment对象生命周期?

Fragment对象有独立的生命周期及生命周期方法,但其生命周期受Activity对象的生命周期影响.

...具体生命周期参考生命周期图,[先Activity在Fragment]

onArrach()

onCreate()

onCreateView

onActivityCreated()

onStart()

onResume()

onPause()

onStop()

onDestroyView()

onDestroy()

onDetach()

5.Fragment应用案例?

1)RadioGroup+Fragment (基本实现+优化实现)

2)ViewPager+Fragment （尝试实现）

a)android.support.v4.app.Fragment

b)android.support.v4.app.FragmentManager

c)MyActivity extends FragmentActivity{}

d)getSupportFragmentManager

e)FragmentPagerAdapter

Android 中Fragment于Activity之间的交互?

1.Activity中获得页面上的Fragment对象?

Fragment frg = fManager.findFragmentByTag(String tagName);

2.Fragment 中获得Activity对象?

Activity aty = getActivity();

3.Activity中传递数据到Fragment

1)调用fragment的setArguments(bundle)

2)在fragment的onCreate或onCreateView方法中掉过getArguments方法

说明：

Fragment中的数据来源,一般有两种方式:

a)被动接收(Activity传数据给Fragment)

b)主动获取(自己获取数据)

FAQ?

Fragment如何嵌套Fragment?

在Fragment中嵌套Fragment一般要通过Fragment对象中的getChildFragmentManager()方法开启事务，然后动态添加

其它Fragment.

Android Intent对象

1.Intent是什么?

1)意图对象(表示要做什么)

2)信使对象(用于传递数据)

2.Intent对象应用场合?

1)启动activity(startActivity,startActivityForResult)

2)启动,解绑,停止service()

3)发送广播(sendBroadcast)

3.Intent对象应用类型?

1)显式intent(目标明确(new Intent(context,类名.class)))

2)隐式intent(目标不明确(new Intent(String action)))根据字符串底层去查找

说明:显式intent通常应用在进程内部各组件之间,隐式intent主要应用于多进程之间.

4.Inetent对象常用属性及方法?

1)setAction(String action)

2)addCategory分类

3)setData(Uri)

4)...

5.Intent对象实现数据的传递?

1)putXXX(key,value)

2)getXXX(key)

通过intent传递数据要么实现Serializable接口,要么实现Parcelable接口.在Android中假如要Intent传递一个自定义

的一个对象，建议实现Parcelable接口，具体如何实现parcelable接口如何写代码可以参考Bundle类型

Android Application对象

1.Application是什么?

1)Android中的一个全局Context对象(资源访问能力).

2)Android中的一个全局共享对象(能够实现多组件之间数据共享).

2.Application对象应用场合?

1)访问全局资源

2)实现数据缓存

例如: 登录以后的用户信息[每次打开APP,都记录一次,临时]

3.Application对象的创建及生命周期?

每个Android项目都内置一个Application对象，假如我们需要此application对象指向我们自己定义的application,

那我们需要自己编写并注册此application.

1)编写:直接或间接的继承Application

2)注册:清单配置文件的application元素中使用name属性指定

3)应用:(例如记录一些数据)

此对象在APP启动时创建,在APP销毁时销毁,此对象是一个常驻内存的对象,不建议在此对象存储大量的数据.