要求：自定义广播，系统广播，有序广播
1、Android中的广播主要分为两种类型：标准广播和有序广播
未命名图片.png 发 出 一 条 广 播 
广 播 接 收 器 [ 
广 播 接 收 器 2 
广 播 接 收 器 3 
标准广播：一种完全异步执行的广播，在广播发出之后，所有的广播接收器几乎都会在同一时刻接收到这条广播消息（没有任何先后顺序）
未命名图片.png 发 出 一 条 广 播 
广 播 接 收 器 丨 
可 将 广 播 截 断 
广 播 接 收 器 2 
可 将 广 播 截 断 
广 播 接 收 器 3 
有序广播：一种同步执行的广播，在广播发出之后，同一时刻只会有一个广播接收器能够收到这条广播消息，当这个广播接收器中的逻辑执行完毕后，广播才会继续传递（有先后顺序）
2、接收系统广播
若要接收广播，则需要使用广播接收器。广播接收器可以自由地对自己感兴趣的广播进行注册，这样当有相应的广播发出时，广播接收器就能够收到该广播，并在内部处理相应的逻辑
  在AndroidManifest.xml中注册——静态注册
注册广播的方式：在代码中注册——动态注册

创建广播接收器：新建一个类（继承自BroadcastReceiver）>>>重写父类的onReceive（）方法

动态注册：（注册的逻辑写在onCreate（）方法中，必须在程序启动之后才能接收到广播）
intentFilter = new IntentFilter();//创建一个intentFilter实例
intentFilter.addAction("Android.net.conn.CONNECTIVITY_CHANGE");//广播接收器想要监听什么广播，就在这里添加相应的action
networkChangeReceiver = new NetworkChangeReceiver();//创建一个NetworkChangeReceiver实例
registerReceiver(networkChangeReceiver,intentFilter);//进行注册
//取消注册广播接收器
@Override
Protected void onDestroy(){//动态注册的广播接收器一定都要取消注册
super.onDestroy();
unregisterReceiver(networkChangeReceiver);
}
//重写父类onReceive（）方法
Class NetworkChangeReceiver extends BroadcastReceiver{
@Override
  
@Override
Publicvoid onReceive(Contextcontext,Intentintent){
ConnectivityManagerconnectionManager=(ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connectionManager.getActiveNetworkInfo();//得到NetworkInfo的实例
if(networkInfo != null && networkInfo.isAvailable()){
Toast.makeText(context,"networkavailable",Toast.LENGTH_SHORT).show();
}else{
Toast.makeText(context,"networkisunavailable",Toast.LENGTH_SHORT).show(); 
}
//声明权限
<uses-permission  android:name="android.permission.ACCESS_NETWORK_STATE"/>

静态注册：（在程序在未启动的情况下就接收广播）
Public  class  BootCompleteReceiver  extends  BroadcastReceiver{
@Override
Public  void  onReceive(Contextcontext,Intentintent){
Toast.makeText(context,"BootComplete",Toast.LENGTH_LONG).show();
}   //只是在方法中使用Toast弹出一段提示信息
//静态广播接收器注册
<receiver
android:name=".BootCompleteReceiver"
android:enabled="true"
android:exported="true">
<intent-filter>
<action  android:name="android.intent.action.BOOT_COMPLETED"/>
</intent-filter>     //添加相应的action
//声明权限
<uses-permissionandroid:name="android.permission.RECEIVE_BOOT_COMPLETED"/>

广播接收器主要是创建一条状态栏通知，或者启动一个服务。

3、发送自定义广播
在发送广播之前，需要先定义一个广播接收器来准备接收此广播。
①新建一个广播接收器
public  class  MyBroadcastReceiver  extends  BroadcastReceiver{
@Override
public  void  onReceive(Contextcontext,Intentintent){
Toast.makeText(context,"received in MyBroadcastReceiver",Toast.LENGTH_LONG).show();
}
}
②在AndroidMainfest.xml文件中修改
<intent-filter>
<action android:name="com.example.broadcasttest.MY_BROADCAST"/>    //让MyBroadcastReceiver 接收广播
</intent-filter>
③加入一个button，设置点击事件
Button  button = (Button) findViewById(R.id.button);
button.setOnClickListener(newView.OnClickListener(){
@Override
public  void  onClick(Viewview){
Intent  intent = new Intent("com.example.broadcasttest.MY_BROADCAST");//要发送的广播
sendBroadcast(intent);//将广播发送出去
}  

￼
  

4、发送有序广播
应用程序发出的广播是可以被其他的应用程序接收到的！！！
sendBroadcast(intent);//发送标准广播 

send  Ordered  Broadcast(intent,null);//发送有序广播

设定广播接收器的先后顺序，前面的广播接收器可以将广播截断，以阻止其继续传播
>>>>>>>>>>>>>>快捷键 Ctrl+Shfit+/ 快捷键进行注释>>>>>>>>>>>>>>>
<intent-filter  android:priority="100">    <!--给广播接收器设置优先级-->

abortBroadcast();//表示将这条广播截断，后面的广播接收器将无法再接收到这条广播

5.使用本地广播
本地广播机制，发出的广播只能够在应用程序的内部进行传递，并且广播接收器也只能接收来自本应用程序发出的广播（安全性问题）
使用一个LocalBroadcastManager对广播进行管理 >>>提供发送广播和注册广播接收器的方法
localBroadcastManager=LocalBroadcastManager.getInstance(this);//获取实例

Intent  intent = new ntent("com.example.broadcasttest.LOCAL_BROADCAST");
localBroadcastManager.sendBroadcast(intent);//发送本地广播

intentFilter.addAction("com.example.broadcasttest.LOCAL_BROADCAST");//广播接收器想要监听什么广播，就在这里添加相应的action
localReceiver = new LocalReceiver();
localBroadcastManager.registerReceiver(localReceiver,intentFilter);//注册本地广播监听器

localBroadcastManager.unregisterReceiver(localReceiver);

classLocalReceiverextendsBroadcastReceiver{
@Override
publicvoidonReceive(Contextcontext,Intentintent){
Toast.makeText(context,"receivedlocalbroadcast",Toast.LENGTH_SHORT).show();
}
}

本地广播无法通过静态注册的方式来接收

例子：实现强制下线功能
1、一种是新建一个广播接收器（此时需要在AndroidManifest.xml文件里修改action），另一种是新建一个类，继承自BroadcastReceiver，并重写父类的onReceiver（）方法
墨迹绘图
类似于动态注册广播接收器
以及发送广播的过程
