程序代码：

1. 简述如何实现Activity的跳转，并写出主要代码。

Public class Activity1 extends Activity{

Private Button btn = null;

setContentView(R.layout.main);//显示相应的布局文件

Btn = (Button)findViewId(R.id.btn);//根据控件ID获取控件

btn.setOnClickListener(new ButtonListener());//将监听器捆绑到控件上

}

//加一个监听器，实现一个接口

Class ButtonListener implements OnClickListener{

Public void onClick(View v){

Intent intent = new Intent();//生成一个Intent对象

Intent.setClass(Activity1.this,Activity2.class);//调用Intent的setClass方法，实现使用一个Activity调用另外一个Activity

Activity1.this.startActivity(intent);

}

}

(2)public class SecondActivity extends Activity {

private TextView tv;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

//设置当前的Activity的界面布局

setContentView(R.layout.activity\_second);

//获得Activity传递的Intent

Intent intent = this.getIntent();

//获得Intent传递的Bundl对象

Bundle bundle = intent.getExtras();

tv = (TextView)findViewById(R.id.TextView1);

//获取Bundle对象中key为"useName"所对应的字符串

String str= bundle.getString ("useName");

//设置TextView的文本

tv.setText(str);

}

}

1. Service

(1)public class MainActivity extends Activity {

// 声明Button

private Button startBtn, stopBtn, bindBtn, unbindBtn;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// 设置当前布局视图

setContentView(R.layout.activity\_main);

// 实例化Button

startBtn = (Button) findViewById(R.id.startButton01);

stopBtn = (Button) findViewById(R.id.stopButton02);

bindBtn = (Button) findViewById(R.id.bindButton03);

unbindBtn = (Button) findViewById(R.id.unbindButton04);

// 添加监听器

startBtn.setOnClickListener(startListener);

stopBtn.setOnClickListener(stopListener);

bindBtn.setOnClickListener(bindListener);

unbindBtn.setOnClickListener(unBindListener); }

// 启动Service监听器

private OnClickListener startListener = new OnClickListener() {

@Override

public void onClick(View v) {

// 创建Intent

Intent intent = new Intent();

// 设置Action属性

intent.setAction("com.dh.MY\_SERVICE");

// 启动该Service

startService(intent);

} };

// 停止Service监听器

private OnClickListener stopListener = new OnClickListener() {

public void onClick(View v) {

// 创建Intent

Intent intent = new Intent();

// 设置Action属性

intent.setAction("com.dh.MY\_SERVICE");

// 启动该Service

stopService(intent);

} };

// 连接对象

private ServiceConnection conn = new ServiceConnection() {

//成功绑定Service，此回调方法被执行

public void onServiceConnected(ComponentName name, IBinder service) {

Log.i("SERVICE", "连接成功！");

Toast.makeText(MainActivity.this, "连接成功！", Toast.LENGTH\_LONG).show();

//获取MyService返回的Binder对象并强制转换为MyBinder

MyBinder binder=(MyBinder) service;

//需要通过此binder获取所绑定的MyService对象

MyService myService=binder.getService();

//通过获取到的myService对象调用MyService中的成员变量或方法

myService.countShow(); }

public void onServiceDisconnected(ComponentName name) {

Log.i("SERVICE", "断开连接！");

Toast.makeText(MainActivity.this, "断开连接！", Toast.LENGTH\_LONG).show();

} };

// 綁定Service监听器

private OnClickListener bindListener = new OnClickListener() {

public void onClick(View v) {

// 创建Intent

Intent intent = new Intent();

// 设置Action属性

intent.setAction("com.dh.MY\_SERVICE");

// 绑定Service

bindService(intent, conn, Service.BIND\_AUTO\_CREATE); } };

// 解除绑定Service监听器

private OnClickListener unBindListener = new OnClickListener() {

public void onClick(View v) {

// 创建Intent

Intent intent = new Intent();

// 设置Action属性

intent.setAction("com.dh.MY\_SERVICE");

// 解除绑定Service

unbindService(conn);

}};}

(2)public class MyService extends Service {

// 可以返回null，通常返回一个Binder子类

public IBinder onBind(Intent intent) {

Log.i("SERVICE", "onBind..............");

Toast.makeText(MyService.this, "onBind..............",

Toast.LENGTH\_LONG).show();

return new MyBinder();

}

// Service创建时调用

public void onCreate() {

Log.i("SERVICE", "onCreate..............");

Toast.makeText(MyService.this, "onCreate..............",

Toast.LENGTH\_LONG).show();

}

// 当客户端调用startService()方法启动Service时，该方法被调用

public void onStart(Intent intent, int startId) {

Log.i("SERVICE", "onStart..............");

Toast.makeText(MyService.this, "onStart..............",

Toast.LENGTH\_LONG).show();

}

// 当Service不再使用时调用

public void onDestroy() {

Log.i("SERVICE", "onDestroy..............");

Toast.makeText(MyService.this, "onDestroy..............",

Toast.LENGTH\_LONG).show(); }

public class MyBinder extends Binder {

public MyService getService() {

return MyService.this; } }

// 该Service提供了一个示例方法

public void countShow() {

Toast.makeText(MyService.this, "countShow()方法被调用", Toast.LENGTH\_LONG).show(); } }

5、Broadcast

public class Broadcast extends Activity {

public static final String BROADCAST\_ACTION = "com.china.ui.NEW\_LIFEFORM";

private Receiver receiver;

private IntentFilter filter;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//\*\*\*\*\*\*注册自定义动态广播消息\*\*\*\*\*\*\*

//创建IntentFilter

filter = new IntentFilter();

//设置IntentFilter的行为

filter.addAction(BROADCAST\_ACTION);

//设置广播接收者优先级，优先级高的先接收到广播

filter.setPriority(1000);

//创建Receiver

receiver = new Receiver();

//注册自定义动态广播消息,动态将receiver注册到系统中

registerReceiver(receiver,filter);

final Intent intent = new Intent();

intent.setAction(BROADCAST\_ACTION);

intent.putExtra("name", "science");

Button button = (Button) findViewById(R.id.sendBroadcastIntent);

button.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

//有序广播发送

sendOrderedBroadcast(intent, null);

}

});

}

protected void onDestroy() {

super.onDestroy();

unregisterReceiver(receiver);

}

}

1. Receiver：

public class Receiver extends BroadcastReceiver {

public void onReceive(Context context, Intent intent) {

CharSequence string = intent.getExtras().getString("name");

Toast.makeText(context, string, Toast.LENGTH\_LONG).show();

abortBroadcast(); //Receiver1接收到广播后中断广播

}

}

6、SharedPreference：

public class MainActivity extends Activity {

Button button1,button2;

EditText edittext;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

edittext=(EditText)findViewById(R.id.edittext);

button1=(Button)findViewById(R.id.button1);

button2=(Button)findViewById(R.id.button2);

ButtonListener bl=new ButtonListener();

button1.setOnClickListener(bl);

button2.setOnClickListener(bl);

}

class ButtonListener implements OnClickListener{

public void onClick(View v){

switch(v.getId()){

case R.id.button1:

//步骤1：获取输入值

String password = edittext.getText().toString().trim();

//步骤2:使用getSharedPreferences创建一个SharedPreferences对象；lock表示要写入的XML文件名

SharedPreferences sp=getSharedPreferences("lock", MODE\_WORLD\_READABLE);

//步骤3:使用SharedPreferences.edit()生成SharedPreferences编辑对象，通过该对象将数据放入到SharedPreferences对象中；

Editor editor = sp.edit();

//步骤4:使用putString()将获取的数据放入SharedPreference中，该值通过key键mima引用；

editor.putString("mima", password);

//步骤5：提交数据，将SharedPreference保存到XML文件中

editor.commit();

Toast.makeText(getApplicationContext(), "口令设置成功", Toast.LENGTH\_LONG).show();

break;

case R.id.button2:

//步骤1：创建一个SharedPreferences接口对象

SharedPreferences read = getSharedPreferences("lock", 2);

//步骤2：获取文件中的值

String value = read.getString("mima", "");

Toast.makeText(getApplicationContext(), "口令为："+value, Toast.LENGTH\_LONG).show();

break;

}

}

}