1. Activity本质是什么
2. Activity生命周期
3. 一个Activity生命周期

可见状态：onCreate -> onStart -> onResume

不可见状态：onPause -> onStop

销毁状态：onDestory

1. 多个Activity交互的生命期

A Activity打开B Activity

第一步：首先打开A Activity

A Activity onCreate

A Activity onStart

A Activity onResume

第二步：打开B Activity

A Activity onPause

B Activity onCreate

B Activity onStart

B Activity onResume

A Activity onStop

第三步：B Activity点击Back键

B Activity onPause

A Activity onRestart

A Activity onStart

A Activity onResume

B Activity onStop

B Activity onDestroy

第四步：A Activity点击Back键

A Activity onPause

A Activity onStop

A Activity onDestroy

1. Activity生命周期交互设计思想

为什么要暂停当前显示的Activity？

为什么要先执行要打开的Activity的onCreate，onStart，onResume方法呢？

在打开新Acticity的时候，为什么不先执行当前Acticity的onStop呢？

1. Activity的横竖屏切换

竖屏：

onCreate

onStart

onResume

切换成横屏：

onPause

（如果要保存状态，则在onSaveInstanceState（Bundle outState）里面保存状态， 如：outState.putString（”NAME”, “zhaoqy”）; ）

onStop

onDestroy

onCreate（获取保存的状态值： saveInstanceState.getString(“NAME”);）

onStart

onResume

1. 生命周期应用场景

音乐播放：

在res中新建一个raw文件夹存放响应的音频文件

在class中新建一个MediaPlayer 的对象mediaPlayer：

private MediaPlayer mediaPlayer；

在onCreate()方法中写mediaPlayer = MediaPlayer.create(this, R.raw.sing);

mediaPlayer.start();

在activity跳转以后停止播放， 并记录下播放的位置

在onPause中

if(mediaPlayer.isPlaying()){

mediaPlayer.pause();

position = mediaPlayer.getCurrentPosition();

}

//在重新播放的时候获得之前保存的位置  
在onResume中mediaPlayer.seekTo(positon);

1. Activity的启动方式
2. 直接启动
3. 匿名启动
4. Activity之间的数据交互
5. 如何启动系统的Activity

启动系统Activity：浏览器

Intent intent = new Intent();

intent.setAction(Intent.ACTION\_VIEW);

Uri uri = Uri.parse(“http://www.imooc.com”);

intent.setData(uri);

startActivity(intent);

启动系统Activity：相册

Intent intent = new Intent();

intent.setAction(Intent.ACTION\_GET\_CONTENT);

intent.setType(“image/\*”);

startActivity(intent);

启动系统Activity：发送短信

Intent intent = new Intent();

intent.setAction(Intent.ACTION\_SEND);

intent.setType(“text/plain”);

intent.putExtra(Intent.EXTRA\_TEXT, “I am a boy”);

startActivity(intent);

启动系统Activity：拨打电话

Intent intent = new Intent();

intent.setAction(Intent.ACTION\_VIEW);

Uri uri = Uri.parse(“tel:123456789”);

intent.setData(uri);

startActivity(intent);

1. Activity启动模式

Android:launcherMode—— standard、singleTop、singleTask、singleInstance

1. Activity之间的数据传递
2. 传递基本类型：

传递数据

（直接使用Intent进行传递）

Intent intent = new Intent(ThreeActivity.this, FourActivity.class);

intent.putExtra(“name”, “zhaoqy”);

intent.putExtra(“age”, 23);

startActivity(intent);

或者

（使用bundle进行传递）

Intent intent = new Intent(ThreeActivity.this, FourActivity.class);

Bundle bundle = new Bundle();

bundle.putString(“name”, “zhaoqy”);

bundle.putInt(“age”, 23);

intent.putExtra(bundle);

startActivity(intent);

获取数据

Intent intent = getIntent();

If(intent != null){

String name = intent.getStringExtra(“name”);

int age = intent.getIntExtra(“age”, 0);

}

1. 传递一个对象：

public class Person implements Serializable{

private int age;

private String name;

private String address;

public Person(int age, String name, String address){

this.age = age;

this.name = name;

this.address = address;

}

public String toString(){

return “[name=”+name+“age=”+age+“address=”+address;

}

}

传递数据

Intent intent = new Intent(ThreeActivity.this, FourActivity.class);

Person person = new Person(23, “zhaoqy”, “shenzhen”);

Bundle bundle = new Bundle();

Bundle.putSerializable(“person”, person);

intent.putExtras(bundle);

startActivity(intent);

获取数据

Intent intent = getIntent();

If(intent != null){

Person person = (Person)intent.getSerializableExtra(“peson”);

textView.setText(person.toString);

}

1. 传递bitmap：

传递数据

Intent intent = new Intent(ThreeActivity.this, FourActivity.class);

Bundle bundle = new Bundle();

Bitmap bitmap = BitmapFactory.decodeResource(getResource, R.drawable.ic\_;ouncher);

Bundle.putParcelable(“bitmap”, bitmap);

intent.putExtras(bundle);

startActivity(intent);

获取数据

Intent intent = getIntent();

If(intent != null){

Bitmap bitmap = intent.getParcelableExtra(“bitmap”);

imageView.setImageBitmap(bitmap);

}