# 1.什么是Java虚拟机？为什么 Java被称作是“平台无关的编程语言”？

Java 虚拟机是一个可以执行 Java 字节码的虚拟机进程。Java 源文件被编译成能被 Java 虚拟

机执行的字节码文件。

Java 被设计成允许应用程序可以运行在任意的平台，而不需要程序员为每一个平台单独重写

或者是重新编译。Java 虚拟机让这个变为可能，因为它知道底层硬件平台的指令长度和其他

特性。

# 2.JDK和JRE的区别是什么？

Java 运行时环境(JRE)是将要执行 Java 程序的 Java 虚拟机。它同时也包含了执行 applet 需要

的浏览器插件。Java 开发工具包(JDK)是完整的 Java 软件开发包，包含了 JRE，编译器和其他

的工具(比如：JavaDoc，Java 调试器)，可以让开发者开发、编译、执行 Java 应用程序。

# 3.线程的创建方法

线程的创建分为两种方法：1、继承Thread类创建线程 2、实现 Runnable接口创建线程

# 4.Java支持的数据类型有哪些？什么是自动拆装箱？

Java 语言支持的 8 中基本数据类型是：

byte

short

int

long

float

double

boolean

char

自动装箱是 Java 编译器在基本数据类型和对应的对象包装类型之间做的一个转化。比如：

把 int 转化成 Integer，double 转化成 double，等等。反之就是自动拆箱。

5.Java中的方法覆盖(Overriding) 和方法重载(Overloading) 是什么意思？

Java 中的方法重载发生在同一个类里面两个或者是多个方法的方法名相同但是参数不同的

情况。与此相对，方法覆盖是说子类重新定义了父类的方法。方法覆盖必须有相同的方法名，

参数列表和返回类型。覆盖者可能不会限制它所覆盖的方法的访问。

# 6.Java中，什么是构造函数？什么是构造函数重载？

当新对象被创建的时候，构造函数会被调用。每一个类都有构造函数。在程序员没有给类提

供构造函数的情况下，Java 编译器会为这个类创建一个默认的构造函数。

Java 中构造函数重载和方法重载很相似。可以为一个类创建多个构造函数。每一个构造函数必须有它自己唯一的参数列表。

# 7.什么是值传递和引用传递？

对象被值传递，意味着传递了对象的一个副本。因此，就算是改变了对象副本，也不会影响

源对象的值。

对象被引用传递，意味着传递的并不是实际的对象，而是对象的引用。因此，外部对引用对

象所做的改变会反映到所有的对象上。

# 8.同步方法和同步代码块的区别是什么？

在 Java 语言中，每一个对象有一把锁。线程可以使用 synchronized 关键字来获取对象上的锁。

synchronized 关键字可应用在方法级别(粗粒度锁)或者是代码块级别(细粒度锁)。

# 9.Java集合类框架的基本接口有哪些？

Java 集合类提供了一套设计良好的支持对一组对象进行操作的接口和类。Java 集合类里面最基本的接口有：

Collection：代表一组对象，每一个对象都是它的子元素。

Set：不包含重复元素的 Collection。

List：有顺序的 collection，并且可以包含重复元素。

Map：可以把键(key)映射到值(value)的对象，键不能重复。

# 10.ArrayList和和LinkedList有什么区别？

ArrayList 和 LinkedList 都实现了 List 接口，他们有以下的不同点：

ArrayList 是基于索引的数据接口，它的底层是数组。它可以以 O(1)时间复杂度对元素进行随机访问。与此对应，LinkedList 是以元素列表的形式存储它的数据，每一个元素都和它的前一个和后一个元素链接在一起，在这种情况下，查找某个元素的时间复杂度是 O(n)。相对于 ArrayList，LinkedList 的插入，添加，删除操作速度更快，因为当元素被添加到集合任

意位置的时候，不需要像数组那样重新计算大小或者是更新索引。

LinkedList 比 ArrayList 更占内存，因为 LinkedList 为每一个节点存储了两个引用，一个指向前一个元素，一个指向下一个元素。也可以参考 ArrayList vs. LinkedList。

# 11.异常处理的时候，finally代码块的重要性是什么？

无论是否抛出异常，finally 代码块总是会被执行。就算是没有 catch 语句同时又抛出异常的情况下，finally 代码块仍然会被执行。最后要说的是，finally 代码块主要用来释放资源，比如：I/O 缓冲区，数据库连接。

# 12.finally代码块和finalize()方法有什么区别？

无论是否抛出异常，finally 代码块都会执行，它主要是用来释放应用占用的资源。finalize()

方法是 Object 类的一个 protected方法，它是在对象被垃圾回收之前由 Java 虚拟机来调用的。

# 13.Android 中常用的布局都有哪些

FrameLayout

RelativeLayout

LinearLayout

AbsoluteLayout

TableLayout

# 14.ListView 如何提高其效率？

① 复用 ConvertView

② 自定义静态类 ViewHolder

③ 使用分页加载

④ 使用 WeakRefrence 引用 ImageView 对象

# 15.Intent 传递数据时，可以传递哪些类型数据？

Intent 可以传递的数据类型非常的丰富，java 的基本数据类型和 String 以及他们的数组形式都可以，除此之

外还可以传递实现了 Serializable 和 Parcelable 接口的对象。

# 16.Handler 机制

Android 中主线程也叫 UI 线程，那么从名字上我们也知道主线程主要是用来创建、更新 UI 的，而其他耗时操作，

比如网络访问，或者文件处理，多媒体处理等都需要在子线程中操作，之所以在子线程中操作是为了保证 UI 的流畅程

度，手机显示的刷新频率是 60Hz，也就是一秒钟刷新 60 次，每 16.67 毫秒刷新一次，为了不丢帧，那么主线程处理

代码最好不要超过 16 毫秒。当子线程处理完数据后，为了防止 UI 处理逻辑的混乱，Android 只允许主线程修改 UI，

那么这时候就需要 Handler 来充当子线程和主线程之间的桥梁了。

我们通常将 Handler 声明在 Activity 中，然后覆写 Handler 中的 handleMessage 方法,当子线程调用

handler.sendMessage()方法后 handleMessage 方法就会在主线程中执行。

这里面除了 Handler、Message 外还有隐藏的 Looper 和 MessageQueue 对象。

在主线程中 Android 默认已经调用了 Looper.preper()方法，调用该方法的目的是在 Looper 中创建

MessageQueue 成员变量并把 Looper 对象绑定到当前线程中。当调用 Handler 的 sendMessage（对象）方法的时

候就将 Message 对象添加到了 Looper 创建的 MessageQueue 队列中，同时给 Message 指定了 target 对象，其实

这个 target 对象就是 Handler 对象。主线程默认执行了 Looper.looper（）方法，该方法从 Looper 的成员变量

MessageQueue 中取出 Message，然后调用 Message 的 target 对象的 handleMessage()方法。这样就完成了整个

消息机制。

# 17.事件分发机制

2.1 事件分发中的 onTouch 和 onTouchEvent 有什么区别，又该如何使用？

这两个方法都是在 View 的 dispatchTouchEvent 中调用的，onTouch 优先于 onTouchEvent 执行。如果在

onTouch 方法中通过返回 true 将事件消费掉，onTouchEvent 将不会再执行。另外需要注意的是，onTouch 能够得到执行需要两个前提条件，第一 mOnTouchListener 的值不能为空，第二当

前点击的控件必须是 enable 的。因此如果你有一个控件是非 enable 的，那么给它注册 onTouch 事件将永远得不到

执行。 对于这一类控件， 如果我们想要监听它的 touch 事件， 就必须通过在该控件中重写 onTouchEvent 方法来实现。

# 18.AsyncTask 如何使用

AsyncTask 用 于 处 理 异 步 任 务 ， 该 类 是 一 个 抽 象 的 泛 型 类 。 类 的 签 名 如 下 ： public abstract class

AsyncTask<Params, Progress, Result>。

三种泛型类型分别代表“启动任务执行的输入参数”、“后台任务执行的进度”、“后台计算结果的类型”。在

特定场合下，并不是所有类型都被使用，如果没有被使用，可以用 java.lang.Void 类型代替。

一个异步任务的执行一般包括以下几个步骤：

1.execute(Params... params)，执行一个异步任务，需要我们在代码中调用此方法，触发异步任务的执行。

2.onPreExecute()，在 execute(Params... params)被调用后立即执行，一般用来在执行后台任务前对 UI 做一些

标记。

3.doInBackground(Params... params)，在 onPreExecute()完成后立即执行，用于执行较为费时的操作，此方法

将接收输入参数和返回计算结果。在执行过程中可以调用 publishProgress(Progress... values)来更新进度信息。

4.onProgressUpdate(Progress... values)，在调用 publishProgress(Progress... values)时，此方法被执行，直

接将进度信息更新到 UI 组件上。

5.onPostExecute(Result result)，当后台操作结束时，此方法将会被调用，计算结果将做为参数传递到此方法中，

直接将结果显示到 UI 组件上。

在使用的时候，有几点需要格外注意：

1.异步任务的实例必须在 UI 线程中创建。

2.execute(Params... params)方法必须在 UI 线程中调用。

3. 不 要 手 动 调 用 onPreExecute() ， doInBackground(Params... params) ， onProgressUpdate(Progress...

values)，onPostExecute(Result result)这几个方法。

4.不能在 doInBackground(Params... params)中更改 UI 组件的信息。

5.一个任务实例只能执行一次，如果执行第二次将会抛出异常。

# 19.Looper 、 Message 、 Handler 的关系

当我们的Android应用程序的进程一创建的时候，系统就给这个进程提供了

一个Looper，Looper是一个死循环，它内部维护这个一个消息队列。Looper不

停地从消息队列中取消息（Message），取到消息就发送给了Handler，最后Han

dler根据接收到的消息去修改UI。Handler的sendMessage方法就是将消息添加到

消息队列中。

# 20.ANR 异常是什么？

简介Application NoResponse：应用程序无响应。在主线程中，是不允许执行耗时的操作的，如果主线程阻塞的时间大于6秒，就很有可能出现anr异常。主线程，要完成界面的更新，事件的处理，窗体显示的回调，所以如果主线程阻塞时间较长，就不能很好的处理以上比重要的事情，那么Android有一个机制，就是如果他发现消息队列中有很多消息，主线程没法响应的话，他就会抛出anr异常。所以，比较耗时的操作都必须要交给子线程。

# 21.Activity 之间如何跳转？

Activity之间的跳转分为2种：

 显式跳转：在可以引用到那个类,

并且可以引用到那个类的字节码时可以使用。一般用于自己程序的内部

。显式跳转不可以跳转到其他程序的页面中。

 隐式跳转：可以在当前程序跳转到另一个程序的页面。隐式跳转

不需要引用到那个类,但是必须得知道那个界面的动作(action)和信息(cat

egory)。

 Activity之间通过Intent进行通信。Intent即意图，用于描述一个页

面的信息,同时也是一个数据的载体。

# 22.Activity 生命周期

Activity有三种状态：

1、当它在屏幕前台时,响应用户操作的Activity, 它是激活或运行状态

2、当它上面有另外一个Activity，使它失去了焦点但仍然对用户可见时,

它处于暂停状态。

3、当它完全被另一个Activity覆盖时则处于停止状态。

当Activity从一种状态转变到另一种状态时，会调用其生命周期方法。Activ

ity一共有七个生命周期方法，具体如下表所示。

方法名 说明

void onCreate() 设置布局以及进行初始化操作

void onStart() 可见, 但不可交互

void onRestart() 调用onStart()

void onResume() 可见, 可交互

黑马Android课程笔记 就业服务部出品

void onPause() 部分可见, 不可交互

void onStop() 完全不可见

void onDestroy() 销毁

23.说说你对Service 的理解

很多情况下，一些与用户很少需要产生交互的应用程序，我们一般让它们在后台运行就行了，而且在它们运行期间我们仍然能运行其他的应用。为了处理这种后台进程，Android引入了Service的概念。

 Service在Android中是一种长生命周期的组件，它不实现任何用户界面,是一个没有界面的组件，由其他组件开启。

 Service长期在后台运行, 执行不关乎界面的一些操作比如:网易新闻服务,每隔1分钟去服务查看是否有最新新闻。Service和Thread有点相似,但是使用Thread不安全, 不严谨 Service和其他组件一样，都是运行在主线程中，因此不能用它来做耗时的操作。

# 24.Android 中的进程

Android 中进程的种类

进程优先级由高到低，依次为：

 Foreground process 前台进程

 Visible process 可视进程, 可以看见, 但不可以交互.

 Service process 服务进程

 Background process 后台进程

 Empty process 空进程(当程序退出时, 进程没有被销毁,而是变成了空进程)

# 25.如何使用 Service

 创建Service，定义类继承Service，AndroidManifest.xml中定义<service>

 开启Service，在其他组件中调用startService方法startService(intent);（在activity中可以直接调用），会执行onstartCommand(生命周期方法)(onStart()以过期的方法)，如果发现service还没有，就会先创建执行oncreate。

onCreate只会执行一次，服务一旦被创建出来，就不会再执行onCreate方法，以后再去开启服务只会执行onStartCommand。

 停止Service，调用stopService方法，会执行onDestory()（生命周期方法）。服务也是运行在主线程中，耗时操作需要放到子线程中，不然会出现anrsetForegound(true)(直接在service里调用)，为了避免服务被杀死，我们可以指定这个服务为前台进程。

服务在执行生命周期方法时，也是一个前台进程。广播接收者也一样，在执行生命周期方法onReceive时，也是一个前台进程。

# 26.说说你对ContentProvider 理解

ContentProvider即内容提供者，是Android的四大组件之一。内容提供者是应用程序之间共享数据的接口，Android系统将这种机制应用到方方面面。

比如：联系人Provider专为不同应用程序提供联系人数据；设置Provider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。当应用继承ContentProvider类，并重写该类用于提供数据和存储数据的方法，就可以向其他应用共享其数据。虽然使用其他方法也可以对外共享数据，但数据访问方式会因数据存储的方式而不同，如：采用文件方式对外共享数据，需要进行文件操作读写数据；采用SharedPreferences共享数据，需要使用SharedPreferences API读写数据。而使用ContentProvider共享数据的好处是统一了数据访问方式。内容提供者将数据封装，只暴露出我们希望提供给其他程序的数据。内容提供者中数据更改可被监听。

# 27.什么是 Fragment

Fragment（碎片）就是小型的Activity，它是在Android3.0时出现的。Fragment是表现Activity中UI的一个行为或者一部分。可以把fragment想象成activity的一个模块化区域，有它自己的生命周期，接收属于它自己的输入事件，并且可以在activity运行期间添加和删除（有点像一个可以在不同的activity中重用的“子Activity”）。Fragment必须被嵌入到一个activity中。它们的生命周期直接受其宿主activity的生命周期影响。当一个activity正在运行时，就可以独立地操作每一个Fragment，比如添加或删除它们。Fragment可以定义自己的布局、生命周期回调方法，因此可以将fragment重用到多个activity中，因此可以根据不同的屏幕尺寸或者使用场合改变fragment组合。

# 28.如何创建一个 Fragment

1、 为Fragment定义一个布局

2、 定义类继承Fragment

3、 重写类中的onCreateView方法，返回一个View对象作为当前Fragment的布局。fragment第一次绘制它的用户界面的时候，系统会调用onCreateView()方法。为了绘制fragment的UI，此方法必须返回一个作为fragment布局的根的view。如果fragment不提供UI，可以返回null。

# 29.Fragment 的生命周期

Fragment的生命周期和activity生命周期很像，其生命周期方法如下所示。

 onAttach：绑定到activity

 onCreate：创建fragment

 onCreateView： 创建fragment的布局

 onActivityCreated： activity创建完成后

 onStart： 可见, 不可交互

 onResume： 可见, 可交互

 onPause： 部分可见, 不可交互

 onStop：不可见

 onDestroyView： 销毁fragment的view对象

 onDestroy： fragment销毁了

onDetach： 从activity解绑了

# 30.Padding和Margin区别？

Padding 控件内部内容距离控件边界的距离，或者该布局的边界与布局内容的距离(内边距)

Margin 控件与外部控件的距离，控件与其所在布局边界的距离(外边距)

Gravity指布局或者控件内的内容相对于布局或控件本身的位置  
Layout\_gravity用来控件或布局相对于其父布局的位置

# 31.三级缓存的概念

实现分别是以内存--文件--网络的缓存模式，使用主要在图片上，当获取的时候我们首先在内存上进行查找，主要通过lrucache进行键值对的获取bitmap加载，如果内存上没有我们访问本地的文件(本应用绑定的cache文件)如果有先加载到内存中，放入lrucache，然后进行加载，如果本地和内存都没有我们访问网络获取到图片，先存入文件，再存入内存进行显示。

# 32.内存泄露和内存溢出的区别

内存溢出 out of memory，是指程序在申请内存时，没有足够的内存空间供其使用，出现out of memory；比如：我要存储一个long型的，用int型来收。就造成了内存溢出。如加载一张很大的bitmap。

内存泄露 memory leak，是指程序在申请内存后，无法释放已申请的内存空间，一次内存泄露危害可以忽略，但内存泄露堆积后果很严重，无论多少内存,迟早会被占光。比如申请了一个integer,但给它存了long才能存下的数，那就是内存溢出。Cursor没有及时关闭。通常都是大资源没有关闭。

# 33.线程池的概念

通过Exectors这个类获取一个进程池，可以使用fixed线程池来获取一个可以调整大小的线程池，也可以获取一个按照某些事件固定执行操作的线程，线程池维护的是子线程，我们使用线程池目的是为了线程的复用，这里我们只要实例化了ExectorService(线程池)对象，可以把线程Runnable对象放入就可以每次新开启都是复用线程池内部的线程，而不是实例化一个新的线程对象。

具体为ExecutorService(线程池对象)调用execute(Runnable)执行该子线程

# 34.如何获取assets文件夹下的文件

ContextWrapper下的getAssets获取AssetsManager对象，调用open(fileName)打开Assets文件下的指定文件这个返回了一个流对象，我们可以通过该流对象进行操作

# 35.简述常用的设计模式

单例模式、工厂模式，代理模式，观察者模式，适配器模式

1.单例模式，单例模式就是在一个应用中，我们可能通常会同时使用该对象，同时该对象是唯一的，例如，人的个体都是唯一的，音乐播放器中播放器是唯一的，所以我们引入了单例，使用的时候分为饿汉和懒汉式，饿汉就是不管使用与否都进行创建，如果频繁访问，这个可以一定程度上减少线程同步的缺陷，懒汉就是需要的时候才进行创建，这里如果访问不频繁，使用较为符合，但是如果频繁访问，可能就要引入了线程同步，但是依旧会出现死锁。

工厂模式分为简单工厂和抽象工厂，我们通常只要给入需要使用什么对象的匹配就可以生成一个新的对象。

2.代理模式，可能对于一个很复杂的对象，我们访问的时候如果创建很复杂，这里使用代理可以很好的调用，而且代理模式中我们使用在AIDL上，以完成Binder的代理，这样可以完成服务和Activity的交互。

3.观察者模式:使用在ContentObserver，我们根据具体的变化，然后进行相应的操作，如果我们每次数据库一变化，就去查看无疑很费资源，我们只要观察，只要一变化才修改，平时不去查看数据库。

4.适配器模式:MVC模式的一种，我们使用把数据显示在自己的视图上，实现了内容的分开管理

# 36.栈和队列的区别

栈和队列都是数据存储的底层，我们使用这俩个可以进行数据的存储，但是栈是先进后出的结构，而队列则是先进先出的顺序

# 37.软引用，弱引用，强引用

1.强引用：只要某个对象有强引用与之关联，JVM必定不会回收这个对象，即使在内存不足的情况下，JVM宁愿抛出OOM错误也不会回收这种现象。

2.软引用：对于软引用关联的对象，只有在内存不足的时候JVM才会回收该对象，这一点可以很好的来解决OOM的问题，很适合用来实现缓存。比如网页缓存，图片缓存。

3.弱引用：当JVM进行垃圾回收时，无论内存是否充足，都会回收被弱引用关联的对象

# 38.安卓中的启动模式？

Standard(标准):activity实例一定创建，具体就是onCreate一定会执行。

SingleTop(单顶):任务栈中，顶部相同不走onCreate方法，适用于刷新页面的

SingleTask(单任务):一个任务栈中，只有一个任务，onCreate只走一次，如果想再次调回该页面，把任务栈上的其他页面实例全部弹出。

SingleInstance(单实例):类似于共享，在其他任务栈中，onCreate方法只走一次。

# 39.Android四大组件以及作用

Android有四大组件：Activity、Service、Broadcast Receiver、Content Provider。

Activity :应用程序中，一个Activity通常就是一个单独的屏幕，它上面可以显示一些控件也可以监听并处理用户的事件做出响应。Activity之间通过Intent进行通信。

Service 服务:一个Service 是一段长生命周期的，没有用户界面的程序，可以用来开发如监控类程序。

BroadcastReceive广播接收器:你的应用可以使用它对外部事件进行过滤只对感兴趣的外部事件(如当电话呼入时，或者数据网络可用时)进行接收并做出响应。广播接收器没有用户界面。然而，它们可以启动一个activity或serice 来响应它们收到的信息。

Content Provider内容提供者 :主要用于多个应用间数据共享。这些数据可以存储在文件系统中或SQLite数据库。

# 40.Android中数据存储的方式

1、使用SharedPreferences存储数据；

2、文件存储数据；

3、SQLite数据库存储数据；

4、使用ContentProvider存储数据；

5、网络存储数据；

# 41.谈谈你对安卓的系统架构的理解

应用程序层   java语言  应用程序开发

应用程序框架层  java语言  OS定制  framework层开发

系统运行库层   C C++ 实现  so库

Linux内核层

android系统架构分从下往上为linux 内核层、运行库、应用程序框架层、和应用程序层

# 42.点9 patch的使用

可以进行图片的填充和写值，同时，这个图片中上部可以选择一个区域作为水平拉伸的区域，其他不变，同理左边的选为垂直拉伸的区域

而右边和下边选择的区域是选择文本编辑和显示的位置

工具使用的位置为sdk\tools\draw9patch.bat

# 43.安卓清单文件中包含哪些信息

(1)包名,版本号package,versionCode,versionName

(2)用户的相关信息包括

用户权限:程序中用到的权限都要在这里列出,uses-permission

用户Sd卡的信息，包括最低版本号及目标sdk版本号

(3)应用程序application,里面包括:<meta-data>及四大组件<activity\><service\><receiver\><provider\>

# 44.throw和throws有什么区别？

throw 关键字用来在程序中明确的抛出异常，相反，throws 语句用来表明方法不能处理的异常。每一个方法都必须要指定哪些异常不能处理，所以方法的调用者才能够确保处理可能发生的异常，多个异常是用逗号分隔的。

# 45.都使用过哪些自定义控件

1 pull2RefreshListView

2 LazyViewPager

3 SlidingMenu

4 SmoothProgressBar

5 自定义组合控件

6 ToggleButton

7 自定义吐司

。。。。

# 46.都使用过哪些第三方SDK、平台

1 EventBus（事件处理）

2 xUtils（网络、图片、ORM）

3 JPush（推送平台）

4 友盟（统计平台）

5 有米（优米）（广告平台）

6 百度地图

7 bmob（服务器平台、短信验证、邮箱验证、第三方支付）

8 阿里云 OSS（云存储）

9 ShareSDK（分享平台、第三方登录）

10 ImageLoader（图片加载器，支持多种缓存，避免 OOM）

。。。

# 47.Android 四大著名图片处理框架

1.Universal ImageLoader 是很早开源的图片缓存，在早期被很多应用使用。

2.Picasso 是 Square 开源的项目，且他的主导者是 JakeWharton，所以广为人知。

3.Glide 是 Google 员工的开源项目，被一些 GoogleApp 使用，在去年的 Google I/O 上被推荐，不

过目前国内资料不多。

4.Fresco 是 Facebook 在今年上半年开源的图片缓存，主要特点包括：

(1) 两个内存缓存加上 Native 缓存构成了三级缓存

(2) 支持流式，可以类似网页上模糊渐进式显示图片

(3) 对多帧动画图片支持更好，如 Gif、WebP

鉴于 Fresco 还没发布正式的 1.0 版本，同时一直没太多时间熟悉 Fresco 源码，后面对比不包括

Fresco，以后有时间再加入对比。

# 48.Service 里面可以弹吐司么？

可以的。弹吐司有个条件就是得有一个 Context 上下文，而 Service 本身就是 Context 的子类，因此在 Service里面弹吐司是完全可以的。比如我们在 Service 中完成下载任务后可以弹一个吐司通知用户。

# 49.Fragment 的如何向下兼容

Fragment是在Android3.0才推出的，若想在3.0的低版本下使用Fragment，则需要执行下面2步：

1、 把所有Fragment和FragmentManager改成support-v4包下的类

2、 把Activity的继承改为FragmentActivity(support-v4包下的)

# 50.Intent传递数据时，可以传递哪些类型数据？

Serializable、Charsequence、Parcelable、Bundle