**实 验 报 告**

**课程名称：** **移动应用开发**

**学 院： 计算机科学与工程学院**

**专 业：计算机科学与技术** **班 级： 5班**

**姓 名： 尹浩男 学 号：201801051827**

**年 月 日**

**山 东 科 技 大 学 教 务 处 制**

**实 验 报 告**

**页**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组 别** |  | **姓 名** | **尹浩男** | **同组实验者** |  |
| **实验项目**  **名称** | Fragment | | **实验日期** |  | |
| **教师评语** |  | | | | |
| **实验成绩：** | | | **指导教师（签名）：**  **年 月 日** | | |
| 一、实验目的  * 熟练掌握基于Fragment的Android APP的界面实现技术。  二、实验内容  * 在实验2实现的APP的基础上，使用Fragment使得APP能够根据不同分辨率的设备自动呈现单屏或双屏模式。  三、实验要求  * 实现上述的基本功能要求。在此基础上，可以进一步优化界面，也可以进一步添加其他功能。 * 实验报告提交内容：1）结合系统界面（截图）介绍APP功能；2）核心实现代码（Java或Kotlin代码）和布局文件等。  四、实验代码4.0 项目代码文件组织  4.1 PhoneTitleFragment4.1.1 功能介绍4.1.1.1手机上的效果 **主界面的布局如左图所示 采用线性布局**  **内层是recycleview 线性垂直布局很多个phoneitem**  **recycleview是使用adapter进行设置**  **将json文件内容加载 然后利用实体类Phone和fastjson将json文件内容封装成对象，然后指定PhoneAdapter的模板layout，让listView渲染每个item ，最终展示效果如左图** 4.1.1.2平板上的效果如图所示4   对于本次实验我还做了个小功能  当平板竖屏的时候和手机的显示是一样的   4.1.2 详细代码(注释有解释) class PhoneTitleFragment:Fragment() {      // 默认是手机 单界面      不是平板 两界面      private var isTwoPane = false      inner class PhoneAdapter(val phoneList: List<Phone>): RecyclerView.Adapter<PhoneAdapter.ViewHolder>(){          // 拿到资源id 方便下边函数调用          inner class ViewHolder(view: View): RecyclerView.ViewHolder(view){              val phoneTitle: TextView = view.findViewById(R.id.phoneTitle)              val phoneDetail: TextView = view.findViewById(R.id.phoneDetail)              val phoneImage:ImageView = view.findViewById(R.id.phoneImage)          }          // 做判断 如果是手机 就开启新activity 否则就重新渲染数据          override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {              val view = LayoutInflater.from(parent.context)                      .inflate(R.layout.phone\_item, parent, false)              val holder = ViewHolder(view)              holder.itemView.setOnClickListener {                  val phone = phoneList[holder.absoluteAdapterPosition]                  if(isTwoPane){                      // 双页模式                      val fragment = phoneContentFrag as PhoneContentFragment                      fragment.refresh(phone.phoneName, phone.phoneDetail,phone.imageid.toString())                  }else{                      PhoneContentActivity.actionStart(parent.context, phone.phoneName, phone.phoneDetail,phone.imageid)                  }              }              return holder          }          // 绑定数据 渲染数据          override fun onBindViewHolder(holder: ViewHolder, position: Int) {              val phone = phoneList[position]              holder.phoneTitle.text = phone.phoneName              holder.phoneDetail.text = phone.phoneDetail              holder.phoneImage.setImageResource(phone.imageid)          }          override fun getItemCount(): Int {              return phoneList.size          }      }      override fun onCreateView(          inflater: LayoutInflater,          container: ViewGroup?,          savedInstanceState: Bundle?      ): View? {          // 加载 recycleView的布局文件          return inflater.inflate(R.layout.phone\_title\_frag, container ,false)      }      // 使用adapter加载数据      override fun onActivityCreated(savedInstanceState: Bundle?) {          super.onActivityCreated(savedInstanceState)          isTwoPane = activity?.findViewById<View>(R.id.phoneTitle) != null  // 判断是手机还是平板          val layoutManager = LinearLayoutManager(activity)          phoneTitleRecyclerView.layoutManager = layoutManager          val adapter =  PhoneAdapter(getPhones())          phoneTitleRecyclerView.adapter = adapter      }      // 读取json文件中的信息 填充数据      private  fun getPhones(): List<Phone>{          //val fileContent = context?.assets?.open("phoneList\_copy.json")?.bufferedReader().use { it?.readText() }          val fileContent = this.context?.assets?.open("phoneList.json")?.bufferedReader().use { it?.readText() }          Log.e("avad",fileContent.toString())          var phoneList  =  JSON.parseArray(fileContent,Phone::class.java)          for (phone in phoneList){ // 按照名字来获取图片的资源id  这样就不会写死了              phone.imageid = getResourceByName(phone.imageName)          }          return phoneList      }      // 根据图片名字获取图片资源id      private fun getResourceByName( imageName:String):Int{          val appInfo = context?.applicationInfo          val resId = resources.getIdentifier(imageName, "drawable", appInfo?.packageName)          return resId      }  } 4.1.3 layout 主要涉及的layout有 phone\_title\_frag 里边内容就是一个RecyclerView用于展示商品  phone\_title\_frag.xml  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:orientation="vertical"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent">      <androidx.recyclerview.widget.RecyclerView          android:id="@+id/phoneTitleRecyclerView"          android:layout\_width="match\_parent"          android:layout\_height="match\_parent" />  </LinearLayout> 4.2 PhoneTitleFragment 针对平板的  响应点击事件的时候可以调用 进行刷新界面 4.2.1 详细代码 class PhoneContentFragment:Fragment() {      override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {          return inflater.inflate(R.layout.phone\_content\_frag, container, false)      }      // 针对平板的  响应点击事件的时候可以调用 进行刷新界面      fun refresh(phoneName: String, phoneDetail: String,phoneImageId:String){          contentLayoutt.visibility = View.VISIBLE          phoneTitle.text = phoneName  //  刷新标题          phoneDetaill.text = phoneDetail // 刷新内容          phoneImagee.setImageResource(phoneImageId.toInt())      }  } 4.1.2 layout代码 主要涉及的layout有phone\_content\_frag.xml  内容是详情页的布局       <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:orientation="vertical"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      >      <LinearLayout          android:visibility="invisible"          android:orientation="vertical"          android:layout\_width="match\_parent"          android:layout\_height="match\_parent"          android:id="@+id/contentLayoutt">          <TextView              android:layout\_marginTop="80dp"              android:layout\_gravity="center"              android:id="@+id/phoneTitle"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:text="phoneName"              android:textSize="20dp" />          <ImageView              android:id="@+id/phoneImagee"              android:layout\_width="300dp"              android:layout\_height="400dp"              android:layout\_gravity="center"              android:layout\_marginTop="40dp" />          <TextView              android:id="@+id/phoneDetaill"              android:layout\_width="250dp"              android:layout\_height="wrap\_content"              android:layout\_gravity="center"              android:layout\_marginTop="40dp"              android:text="phoneDetail" />      </LinearLayout>  </LinearLayout> 4.1.2.2 phone\_item.xml   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="wrap\_content">      <ImageView          android:id="@+id/phoneImage"          android:layout\_width="150dp"          android:layout\_height="150dp"          android:layout\_gravity="center\_vertical"          android:layout\_marginLeft="10dp"          android:background="@drawable/l1">      </ImageView>      <LinearLayout          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_gravity="center\_vertical"          android:orientation="vertical">          <TextView              android:id="@+id/phoneTitle"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_gravity="center\_vertical"              android:layout\_marginLeft="10dp"              android:text="手机名字"              >          </TextView>          <TextView              android:id="@+id/phoneDetail"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_gravity="center\_vertical"              android:layout\_marginLeft="10dp"              android:text="手机介绍"              >          </TextView>      </LinearLayout>  </LinearLayout> | | | | | |