设计模式

本组项目为跳蚤市场，在开发过程中遇到了listview中的item界面与我们需要的界面差距较大，所以引入了适配器设计模式。

1. 为何选择适配器模式

在开发过程中，由于系统中现有类与我们所需不相配，所以引入适配器模式是必然的。而且，引入适配器模式，可以调用自己的功能，可以更好的扩展系统功能。

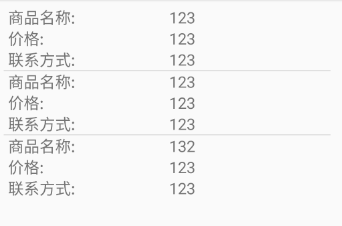
1. 适配器模式的实现

对于listview的问题，我们通过增加Adapter，改变getview()的方法，使得item界面能与我们所需要的界面相适应，具体实现如下：  
**public class** CommodityAdapter **extends** ArrayAdapter<CommodityInfo>{  
 **public** CommodityAdapter(Context context, **int** resource, List<CommodityInfo> objects){  
 **super**(context,resource,objects);  
 }  
 **public** View getView(**int** position, View convertView, ViewGroup parent)  
 {  
 **final** CommodityInfo commodityInfo = getItem(position);  
  
 View oneCommodityView = LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_commodity\_info***,parent,**false**);  
  
 TextView textView1 = (TextView) oneCommodityView.findViewById(R.id.***commodity\_name***);  
 TextView textView2 = (TextView) oneCommodityView.findViewById(R.id.***commodity\_value***);  
 TextView textView3 = (TextView) oneCommodityView.findViewById(R.id.***commodity\_tel***);  
  
 **final** String gName=commodityInfo.getName();  
 **final** String gValue=commodityInfo.getValue();  
 **final** String gTel=commodityInfo.getTel();  
 textView1.setText(gName);  
 textView2.setText(gValue);  
 textView3.setText(gTel);  
  
 oneCommodityView.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(getContext(), CommodityInfoActivity.**class**);  
 */\*intent.putExtra("name",gName);  
 intent.putExtra("value",gValue);  
 intent.putExtra("tel",gTel);\*/* getContext().startActivity(intent);  
 }  
 });  
  
 **return** oneCommodityView;  
 }  
}

1. 取得的效果

通过适配器模式的加入，我们实现了item界面的变化，具体效果如下：

原界面：

新界面：

组员：文天资、杨万纬、于鹄杰、袁其钢、王建森、谢鼎