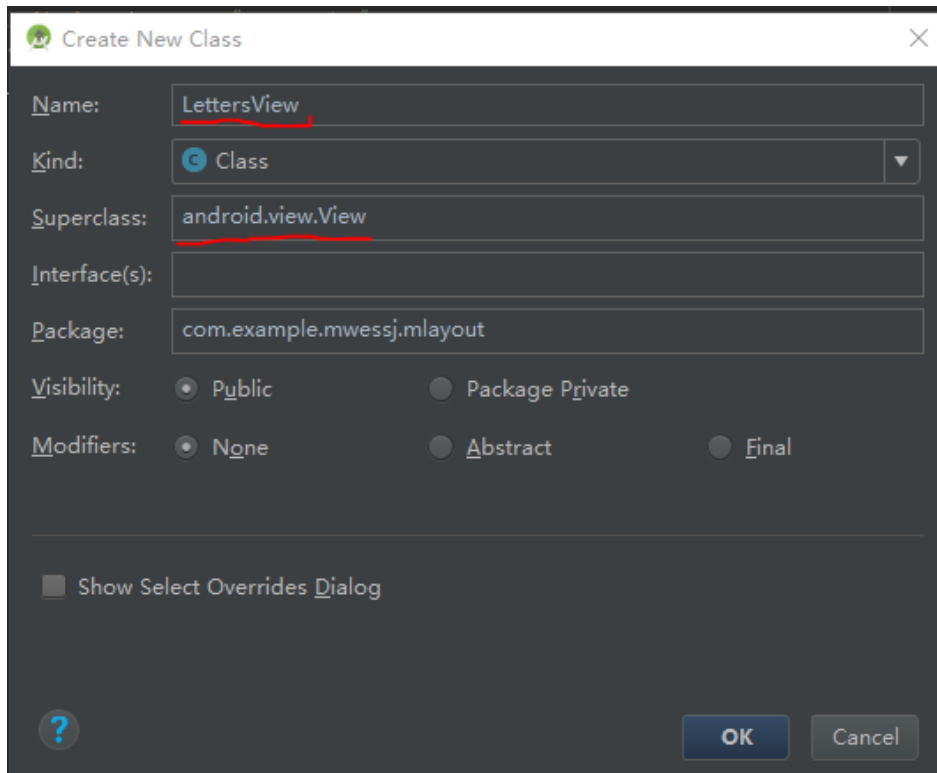


创建导航栏

1.自定义导航栏LettersView

如下图在自己的项目中创建一个Java类，起名：LettersView。继承View类



设置TAG与索引项，并创建Paint实例：

```
1 public class LettersView extends View {
2     // TAG
3     private static final String TAG = "LettersView";
4
5     // 导航栏的索引项
6     private String[] strChars = {"A", "B", "C", "D", "E", "F", "G", "H",
7     "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W",
8     "X", "Y", "Z", "#"};
9
10    // 画笔
11    private Paint mPaint;
12
13    //选中字母的下标
14    private int checkIndex;
15 }
```

设置构造函数与初始化函数：

```
1 public LettersView(Context context) {
```

```

2     super(context);
3     initView();
4 }
5
6 public LettersView(Context context, AttributeSet attrs) {
7     super(context, attrs);
8     initView();
9 }
10
11 public LettersView(Context context, AttributeSet attrs, int defStyleAttr)
12 {
13     super(context, attrs, defStyleAttr);
14     initView();
15 }
16 //初始化
17 private void initView() {
18     //实例化画笔
19     mPaint = new Paint();
20     //设置style
21     mPaint.setTypeface(Typeface.DEFAULT_BOLD);
22     //设置抗锯齿
23     mPaint.setAntiAlias(true);
24     //设置被点击索引项的初始值为-1
25     checkIndex = -1;
26 }

```

创建绘制函数onDraw():

```

1 @Override
2 protected void onDraw(Canvas canvas) {
3     super.onDraw(canvas);
4     /**
5      * 获取View的宽高用以排列索引项
6      */
7     int viewWidth = getWidth();
8     int viewHeight = getHeight();
9     int singleHeight = viewHeight / strChars.length;    //索引项的高度
10
11     //绘制字母
12     for (int i = 0; i < strChars.length; i++) {
13         //设置选中索引项的颜色与字体大小
14         if (i == checkIndex) {
15             mPaint.setColor(Color.WHITE);
16             mPaint.setTextSize(50);
17         } else {
18             mPaint.setColor(Color.BLACK);
19             mPaint.setTextSize(40);
20         }
21         /**
22          * x: 左侧空白位置的宽度
23          * y: 单个字母的高度 + 最上面的字幕空白高度

```

```

24         */
25         float lettersX = (viewWidth - mPaint.measureText(strChars[i])) /
26         2;
27         float lettersY = singleHeight * i + singleHeight;
28         //绘制
29         canvas.drawText(strChars[i], lettersX, lettersY, mPaint);
30         mPaint.reset();
31     }
32 }

```

自定义接口回调，与ListView互动

```

1 //接口回调
2 private OnLettersListViewListener onLettersListViewListener;
3
4 public OnLettersListViewListener getOnLettersListViewListener() {
5     return onLettersListViewListener;
6 }
7
8 public void setOnLettersListViewListener(OnLettersListViewListener
9     onLettersListViewListener) {
10     this.onLettersListViewListener = onLettersListViewListener;
11 }
12
13 /**
14  * 接口回调/ListView联动
15  */
16 public interface OnLettersListViewListener {
17     public void onLettersListener(String s);
18 }

```

设置触摸事件

```

1 /**
2  * 事件分发
3  *
4  * @param event
5  * @return
6  */
7 @Override
8 public boolean dispatchTouchEvent(MotionEvent event) {
9     //判断手势
10     switch (event.getAction()) {
11         case MotionEvent.ACTION_DOWN:
12         case MotionEvent.ACTION_MOVE:
13             setBackgroundResource(R.color.colorAccent);
14             //获取点击的Y坐标，以此来判断选中的字母
15             float y = event.getY();
16             Log.i(TAG, "y:" + y);
17             //第一次被选中的下标
18             int oldCheckIndex = checkIndex;
19             /**
20              * 计算对应位置的索引项

```

```

21         * strChars[当前Y / View的高度 * 字母个数]
22         */
23         int c = (int) (y / getHeight() * strChars.length);
24         Log.i(TAG, "c:" + c);
25         //判断移动
26         if (oldCheckIndex != c) {
27             //越界判定
28             if (c >= 0 && c < strChars.length) {
29                 //效果联动
30                 if (onLettersListViewListener != null) {
31
32                     onLettersListViewListener.onLettersListener(strChars[c]);
33                 }
34                 checkIndex = c;
35                 //更新View
36                 invalidate();
37             }
38             break;
39         case MotionEvent.ACTION_UP:
40             //设置透明背景
41             setBackgroundResource(android.R.color.transparent);
42             //恢复不选中
43             checkIndex = -1;
44             invalidate();
45             break;
46     }
47     return true;
48 }

```

2.修改选择城市对应的xml文件

以下xml布局文件仅供参考，根据每个人的设置会有一定的变化，切勿直接粘贴!!!

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <RelativeLayout
7         android:id="@+id/title_top"
8         android:layout_width="match_parent"
9         android:layout_height="45.0dip"
10        android:background="#ffcd2626">
11
12        <TextView
13            android:id="@+id/title_name"
14            android:layout_width="match_parent"
15            android:layout_height="match_parent"
16            android:gravity="center"
17            android:text="当前城市: 北京"

```

```

18         android:textColor="#ffffffff"
19         android:textSize="20.0dip"/>
20
21     <ImageView
22         android:id="@+id/title_back"
23         android:layout_width="45dip"
24         android:layout_height="45dip"
25         android:layout_alignParentLeft="true"
26         android:layout_centerVertical="true"
27         android:src="@drawable/base_action_bar_back_normal"/>
28
29     <ImageView
30         android:layout_width="1.0dip"
31         android:layout_height="40dip"
32         android:layout_gravity = "center"
33         android:layout_centerVertical="true"
34         android:layout_toRightOf="@id/title_back"
35         android:background="#A71717"
36         android:gravity="center"/>
37
38 </RelativeLayout>
39
40 <RelativeLayout
41     android:id="@+id/title_mid"
42     android:layout_width="match_parent"
43     android:layout_height="wrap_content"
44     android:layout_below="@id/title_top">
45
46     <com.example.mwessj.mlayout.ClearEditText
47         android:id="@+id/search_city"
48         android:layout_width="match_parent"
49         android:layout_height="45dp"
50         android:hint="搜索全国城市（中文）"
51         android:layout_margin="10dip"
52         android:drawableLeft="@drawable/magnifying_glass"
53         android:drawablePadding="8.0dip"
54         android:paddingBottom="8.0dip"
55         android:paddingLeft="10.0dip"
56         android:paddingRight="30.0dip"
57         android:paddingTop="8.0dip"
58         android:singleLine="true"
59         android:textColor="#ff000000"
60         android:textColorHint="#ffcccc"
61         android:textSize="15.0sp"
62
63         android:background="@drawable/contact_search_box_edittext_keyword_backgrou
64         nd"/>
65
66 </RelativeLayout>
67 <FrameLayout
68     android:layout_below="@id/title_mid"

```

```

67     android:layout_width="match_parent"
68     android:layout_height="match_parent">
69
70
71     <ListView
72         android:id="@+id/title_list"
73         android:layout_width="wrap_content"
74         android:layout_height="wrap_content"
75         android:background="#ffffff"/>
76
77     <com.example.mwessj.mlayout.LettersView
78         android:id="@+id/title_letters_view"
79         android:layout_width="30dp"
80         android:layout_height="match_parent"
81         android:layout_gravity="right"/>
82
83 </FrameLayout>
84
85 </RelativeLayout>

```

3.修改ListView对应的item_listview.xml文件

以下xml布局文件仅供参考，根据每个人的设置会有一定的变化，切勿直接粘贴！！

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6      <LinearLayout
7          android:layout_width="match_parent"
8          android:layout_height="wrap_content"
9          android:orientation="vertical">
10         <TextView
11             android:layout_marginLeft="10dp"
12             android:id="@+id/lettersTv"
13             android:layout_width="wrap_content"
14             android:layout_height="wrap_content"
15             android:text="A"
16             android:textColor="#ff0000ff"
17             android:textSize="20sp"/>
18         <TextView
19             android:id="@+id/cityTv"
20             android:layout_width="match_parent"
21             android:layout_height="wrap_content"
22             android:layout_weight="1"
23             android:layout_marginTop="5dip"
24             android:layout_marginLeft="15dip"
25             android:text="北京"
26             android:textSize="18sp" />
27     </LinearLayout>

```

```
28
29 </RelativeLayout>
```

4.修改自定义的adapter

添加寻址函数：

```
1 public View getView(int position, View convertView, ViewGroup parent) {
2     ViewHolder holder = null;
3     if (convertView == null){
4         convertView = mInflater.inflate(R.layout.item_listview, parent,
5         false);
6         holder = new ViewHolder();
7         holder.cityTv = (TextView) convertView.findViewById(R.id.cityTv);
8         holder.lettersTv = (TextView)
9         convertView.findViewById(R.id.lettersTv);
10        convertView.setTag(holder);
11    }else {
12        holder = (ViewHolder)convertView.getTag();
13    }
14
15    City city = mDatas.get(position);
16    int firstPosition = getNmaeForPosition(position);
17    int index = getPositionForNmae(firstPosition);
18    if (index == position){
19        holder.lettersTv.setVisibility(View.VISIBLE);
20        holder.lettersTv.setText(city.getFirstPY());
21    } else {
22        holder.lettersTv.setVisibility(View.GONE);
23    }
24    holder.cityTv.setText(city.getCity());
25
26    return convertView;
27 }
28
29 private class ViewHolder{
30     TextView lettersTv;
31     TextView cityTv;
32 }
33
34 /**
35  * 通过首字母获取该首字母要显示的第一个人的下标
36  *
37  * @param position
38  * @return
39  */
40 public int getPositionForNmae(int position) {
41     for (int i = 0; i < getCount(); i++) {
42         String letter = mDatas.get(i).getAllFirstPY();
43         //首字母显示
44         char firstChar = letter.toUpperCase().charAt(0);
```

```

43         if (firstChar == position) {
44             return i;
45         }
46     }
47     return -1;
48 }
49 /**
50  * 根据名称拿到下标
51  *
52  * @param position
53  * @return
54  */
55 private int getNmaeForPosition(int position) {
56     return mDataas.get(position).getAllFirstPY().charAt(0);
57 }

```

5.修改SelectCity.java

首先，将下列语句

```

1 public class SelectCity extends Activity implements View.OnClickListener{

```

更换为

```

1 public class SelectCity extends Activity implements
LettersView.OnLettersListViewListener, View.OnClickListener{

```

然后，实现接口OnLettersListViewListener

```

1 private LettersView mLettersView;
2 /**
3  * ListView与字母导航联动
4  *
5  * @param s
6  */
7 @Override
8 public void onLettersListener(String s) {
9     //对应的位置
10    int position = myadapter.getPositionForNmae(s.charAt(0));
11    //移动
12    mList.setSelection(position);
13 }

```

最后，修改initView函数，添加下列语句

```

1 mLettersView = (LettersView) findViewById(R.id.title_letters_view);
2 mLettersView.setOnLettersListViewListener(this);
3 //对cityList排序
4 Collections.sort(cityList, new SortUtil());

```

6.自定义排序方法

建立java类，代码如下：

```
1 package com.example.mwessj.util;
2
3 import com.example.mwessj.bean.City;
4
5 import java.util.Comparator;
6
7 public class SortUtil implements Comparator<City> {
8
9     @Override
10     public int compare(City lettersModel, City t1) {
11         return
12         lettersModel.getAllFirstPY().compareTo(t1.getAllFirstPY());
13     }
14 }
```