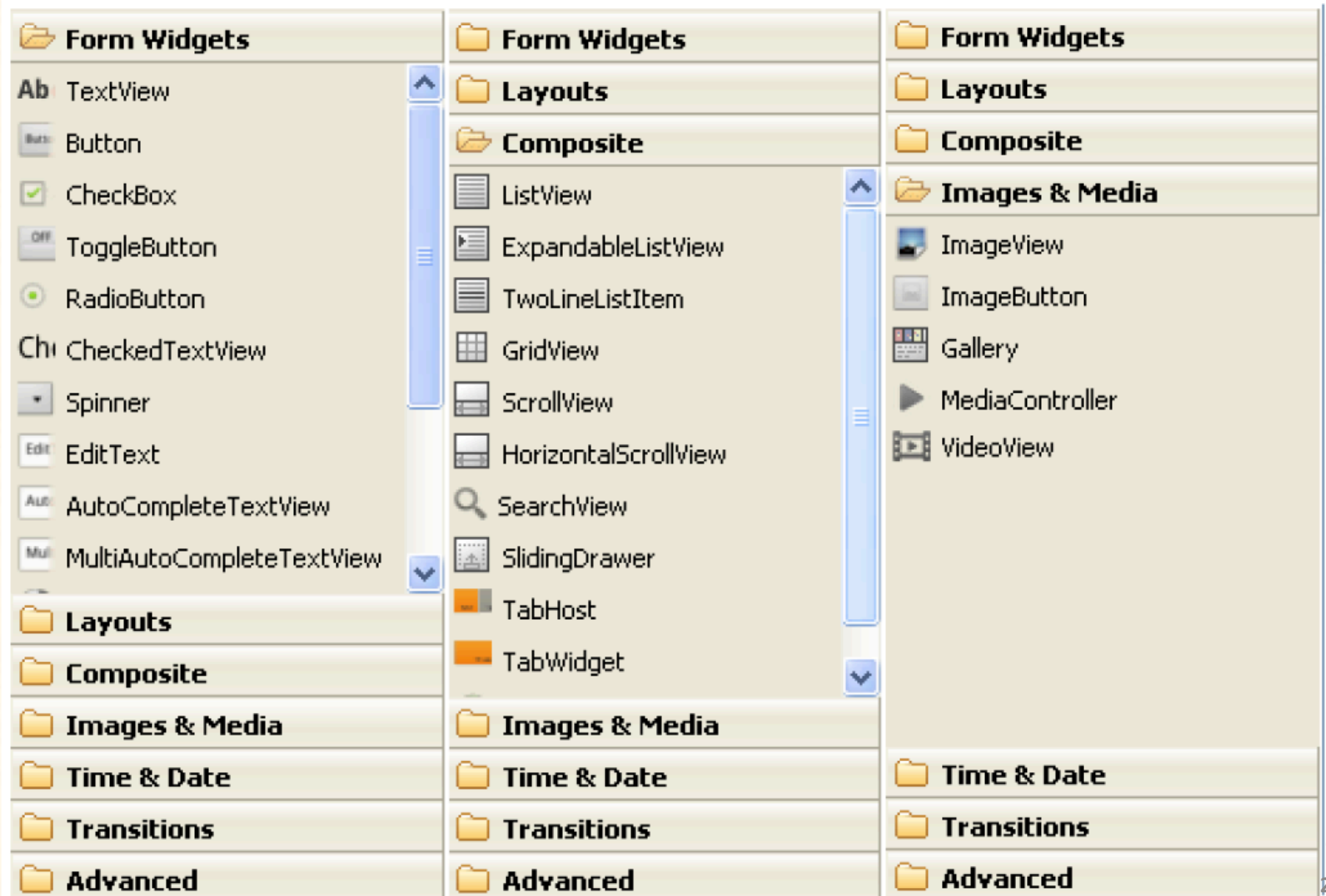


The background of the slide is a grayscale photograph of a wide, multi-tiered stone staircase. The staircase leads up a hill towards a large, ornate building with multiple stories, arched windows, and a central dome. Several people are visible walking on the stairs, providing a sense of scale. The overall atmosphere is historical and grand.

ListView in Android App

Prof. Dr.-Ing. V.Brovkov

Views Hierarchie



Project: P0421_SimpleList

<http://startandroid.ru/ru/uroki/vse-uroki-spiskom/82-urok42-spisok-listview.html>

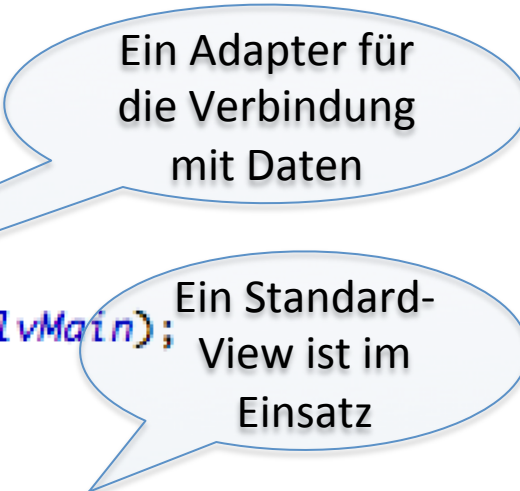
P0421_SimpleList 1

Layout.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6
7     <TextView
8         android:layout_width="fill_parent"
9         android:layout_height="wrap_content"
10        android:text="@string/hello" >
11    </TextView>
12
13    <ListView
14        android:id="@+id/lvMain"
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content" >
17    </ListView>
18
19 </LinearLayout>
```

P0421_SimpleList 2

```
8 public class MainActivity extends Activity {
9     String[] names = { "Karl-Heinz", "Linnéa", "Hans-Joachim",
10                        "Anna-Lena", "Lisa-Marie", "Anna-Maria",
11                        "Elias", "Luca", "Alexander", "Eva-Maria", "Kajetan" };
12
13     /** Called when the activity is first created. */
14     public void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.main);
17         // Find View
18         ListView lvMain = (ListView) findViewById(R.id.lvMain);
19         // Create an Adapter
20         ArrayAdapter<String> adapter =
21             new ArrayAdapter<String>(this,
22                                     android.R.layout.simple_list_item_1, names);
23         //ArrayAdapter<String> adapter =
24         //    new ArrayAdapter<String>(this, R.layout.my_list_item, names);
25         // Connect the ListView with the adapter
26         lvMain.setAdapter(adapter);
27     }
28 }
```



Ein Adapter für die Verbindung mit Daten

Ein Standard-View ist im Einsatz

P0421_SimpleList 3

- Adapter:
 - Das Adapterkonzept folgt dem MVC- (Model-View-Controller-) Muster, bei dem die Datenhaltung (im Businessmodel) von der Anzeige (in den Views) bzw. dem Zugriff über den Controller entkoppelt ist.
 - Die listenförmigen Views, zu denen ListViews und Grids gehören, zeigen mehrere Einträge auf einmal und erlauben das Durchrollen durch die Einträge.

P0421_SimpleList 4

- Einfache Adapter arbeiten mit Arrays von Objekten oder mit Listen, die eine Map – ein assoziatives Array – enthalten können.
- Das Array transportiert in jeder Zeile des Arrays genau ein Objekt. Dieses Objekt wird dann in der Regel als Zeichenkette interpretiert und in der View dargestellt.
- Ein Array kann Objekte eines Typs enthalten oder auch Objekte unterschiedlichen Typs, wobei die Umwandlung in eine Zeichenkette durch das Überschreiben der Methode toString() des Objektes durchgeführt wird.

P0421_SimpleList 6

Vordefinierte ListView Layout

Hello world!

Item 1

Sub Item 1

Item 2

Sub Item 2

Item 3

Sub Item 3

Item 4

Sub Item 4

Item 5

Sub Item 5

Item 6

Sub Item 6

Item 7

Sub Item 7



P0421_SimpleList

Hello world!

Karl-Heinz

Linnéa

Hans-Joachim

Anna-Lena

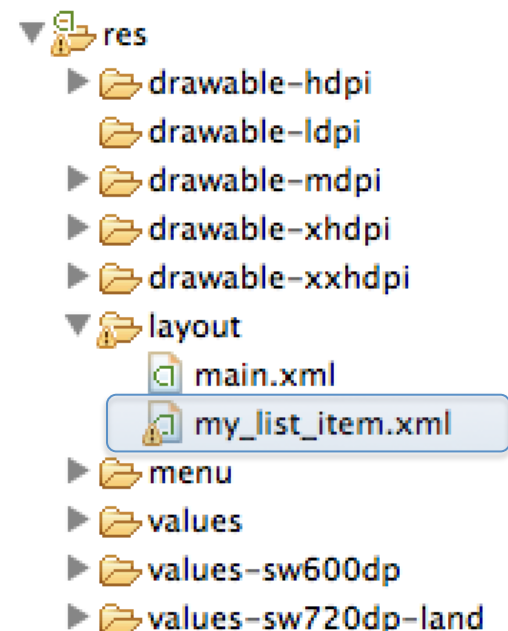
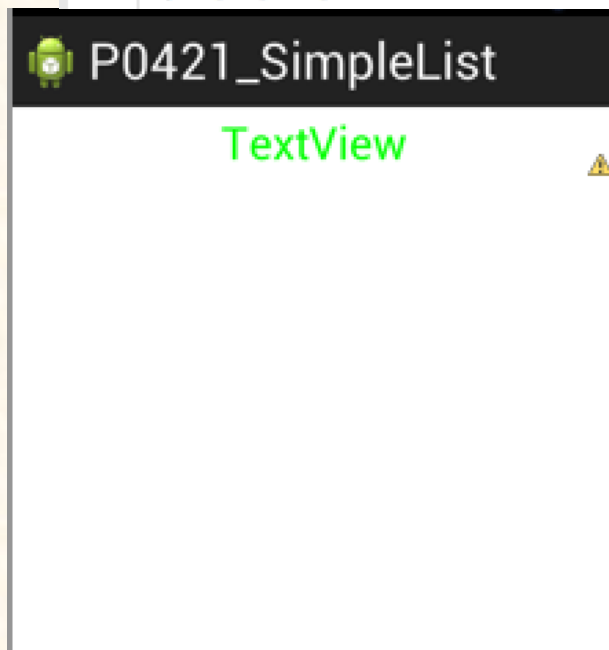
Lisa-Marie

Anna-Maria

P0421_SimpleList 7

User-definierte ListView Layout

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/textView1"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     android:gravity="center_horizontal"
7     android:padding="5dp"
8     android:text="TextView"
9     android:textColor="#00FF00"
10    android:textSize="24sp" >
11
12 </TextView>
```



P0421_SimpleList 8



P0421_SimpleList

Hello world!

Karl-Heinz

Linnéa

Hans-Joachim

Anna-Lena

Lisa-Marie

Anna-Maria

Elias

Luca

Project: P0431_SimpleListChoice

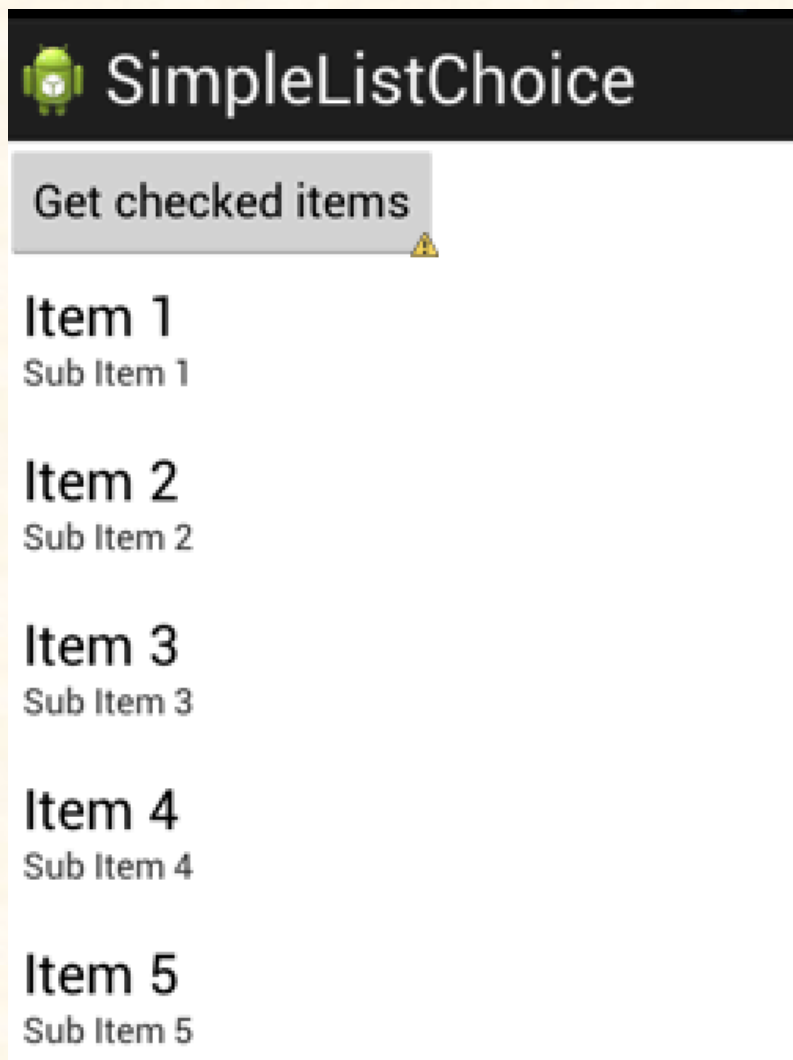
<http://startandroid.ru/ru/uroki/vse-uroki-spiskom/83-urok-43-odinochnyj-i-mnozhestvennyj-vybor-v-list.html>

P0431_SimpleListChoice 1

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6
7     <Button
8         android:id="@+id/btnChecked"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Get checked items" >
12    </Button>
13
14    <ListView
15        android:id="@+id/lvMain"
16        android:layout_width="match_parent"
17        android:layout_height="wrap_content" >
18    </ListView>
19
20 </LinearLayout>
```

P0431_SimpleListChoice 2

main.xml



strings.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <string name="hello">Hello World, MainActivity!</string>
5     <string name="app_name">SimpleListChoice</string>
6     <string name="action_settings">Settings</string>
7
8     <string-array name="names">
9         <item>Karl-Heinz</item>
10        <item>Linnéa</item>
11        <item>Hans-Joachim</item>
12        <item>Anna-Lena</item>
13        <item>Lisa-Marie</item>
14        <item>Anna-Maria</item>
15        <item>Elias</item>
16        <item>Luca</item>
17        <item>Alexander</item>
18        <item>Eva-Maria</item>
19        <item>Kajetan</item>
20    </string-array>
21
22 </resources>
```

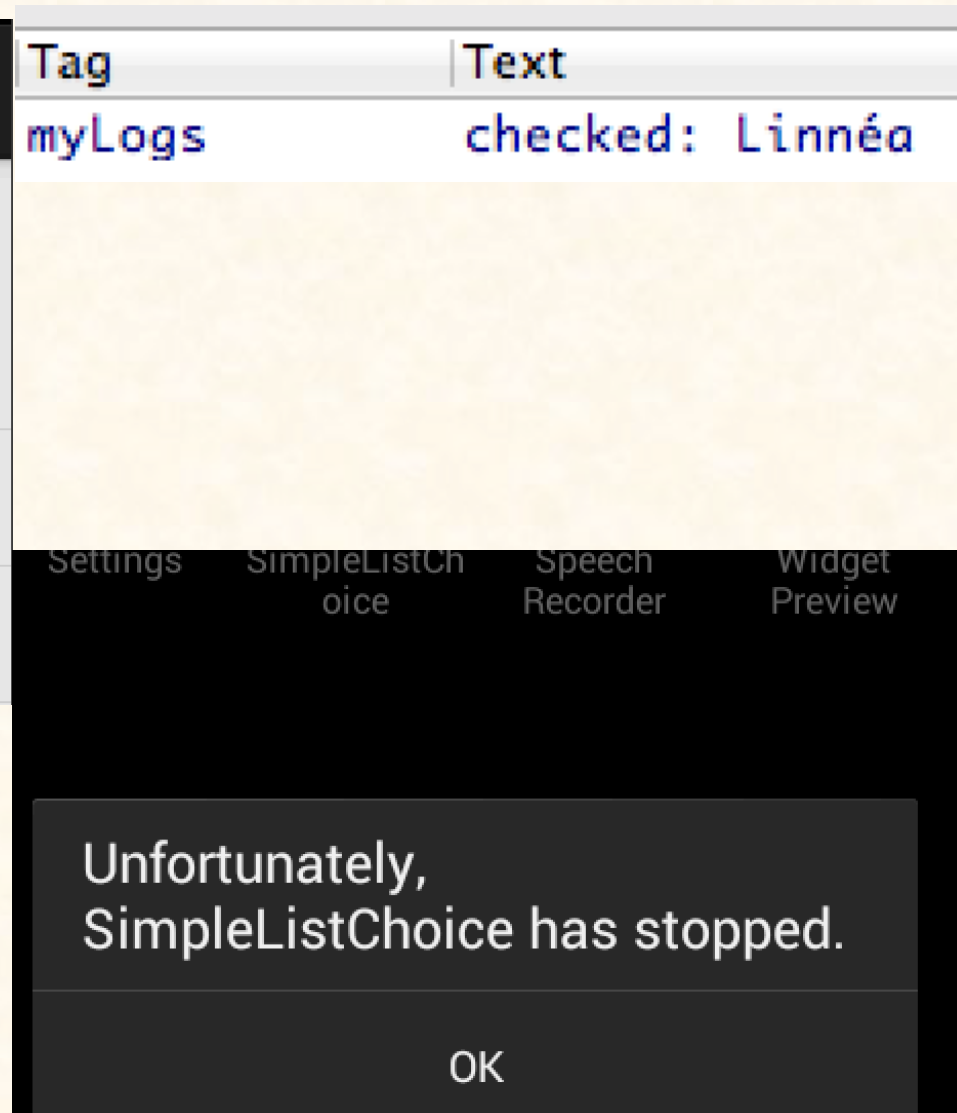
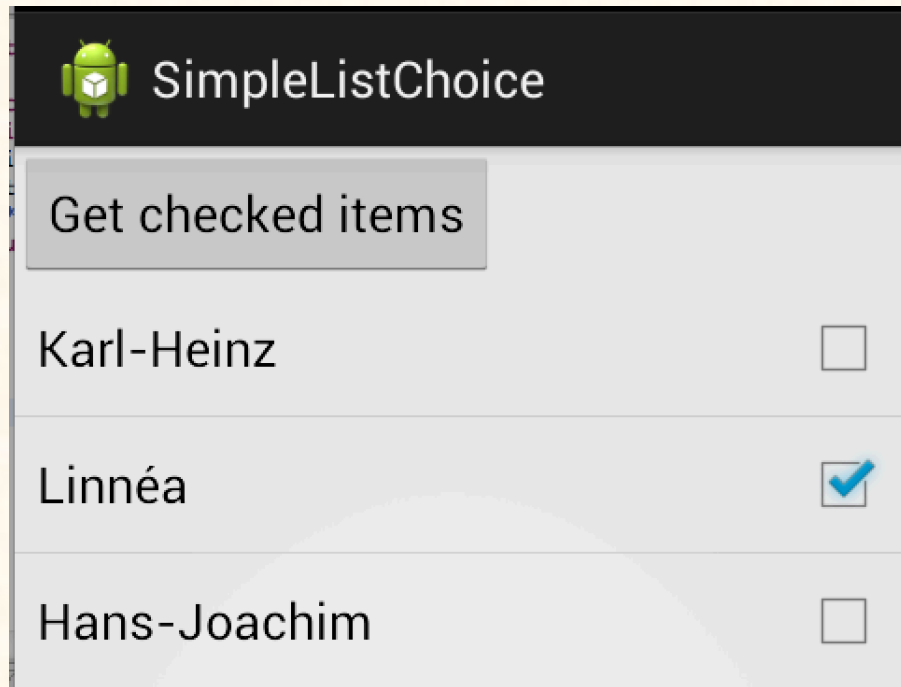
P0431_SimpleListChoice 3

```
13 public class MainActivity extends Activity implements OnClickListener {
14     final String LOG_TAG = "myLogs";
15     ListView lvMain;
16     String[] names;
17     /** Called when the activity is first created. */
18     public void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.main);
21         lvMain = (ListView) findViewById(R.id.lvMain);
22         // устанавливаем режим выбора пунктов списка
23         lvMain.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
24         //lvMain.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
25         // Создаем адаптер, используя массив из файла ресурсов
26         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
27             this, R.array.names,
28             //android.R.layout.simple_list_item_single_choice);
29             android.R.layout.simple_list_item_multiple_choice);
30         lvMain.setAdapter(adapter);
31         Button btnChecked = (Button) findViewById(R.id.btnChecked);
32         btnChecked.setOnClickListener(this);
33         // получаем массив из файла ресурсов
34         names = getResources().getStringArray(R.array.names);
35     }
36
37     public void onClick(View arg0) {
38
39     }
```

P0431_SimpleListChoice 4

```
13 public class MainActivity extends Activity implements OnClickListener {
14     final String LOG_TAG = "myLogs";
15     ListView lvMain;
16     String[] names;
17     /** Called when the activity is first created. */
18     public void onCreate(Bundle savedInstanceState) {
36
37     public void onClick(View arg0) {
38         // пишем в лог выделенный элемент
39         Log.d(LOG_TAG, "checked: " + names[lvMain.getCheckedItemPosition()]);
40         //Log.d(LOG_TAG, "checked: ");
41         //SparseBooleanArray sbArray = lvMain.getCheckedItemPositions();
42         //for (int i = 0; i < sbArray.size(); i++) {
43         //    int key = sbArray.keyAt(i);
44         //    if (sbArray.get(key))
45         //        Log.d(LOG_TAG, names[key]);
46         //}
47     }
48 }
```


P0431_SimpleListChoice 5




P0431_SimpleListChoice 6

```
13 public class MainActivity extends Activity implements OnClickListener {
14     final String LOG_TAG = "myLogs";
15     ListView lvMain;
16     String[] names;
17     /** Called when the activity is first created. */
18     public void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.main);
21         lvMain = (ListView) findViewById(R.id.lvMain);
22         // устанавливаем режим выбора пунктов списка
23         //lvMain.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
24         lvMain.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
25         // Создаем адаптер, используя массив из файла ресурсов
26         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
27             this, R.array.names,
28             //android.R.layout.simple_list_item_single_choice);
29             android.R.layout.simple_list_item_multiple_choice);
30         lvMain.setAdapter(adapter);
31         Button btnChecked = (Button) findViewById(R.id.btnChecked);
32         btnChecked.setOnClickListener(this);
33         // получаем массив из файла ресурсов
34         names = getResources().getStringArray(R.array.names);
35     }
36
37     public void onClick(View arg0) {
38
39     }
```

P0431_SimpleListChoice 7

```
13 public class MainActivity extends Activity implements OnClickListener {
14     final String LOG_TAG = "myLogs";
15     ListView lvMain;
16     String[] names;
17     /** Called when the activity is first created. */
18     public void onCreate(Bundle savedInstanceState) {
36
37     public void onClick(View arg0) {
38         // пишем в лог выделенный элемент
39         //Log.d(LOG_TAG, "checked: " + names[lvMain.getCheckedItemPosition()]);
40         Log.d(LOG_TAG, "checked: ");
41         SparseBooleanArray sbArray = lvMain.getCheckedItemPositions();
42         for (int i = 0; i < sbArray.size(); i++) {
43             int key = sbArray.keyAt(i);
44             if (sbArray.get(key))
45                 Log.d(LOG_TAG, names[key]);
46         }
47     }
48 }
49
```

P0431_SimpleListChoice 8

 SimpleListChoice

Get checked items

Karl-Heinz	<input checked="" type="checkbox"/>
Linnéa	<input checked="" type="checkbox"/>
Hans-Joachim	<input checked="" type="checkbox"/>
Anna-Lena	<input checked="" type="checkbox"/>
Lisa-Marie	<input type="checkbox"/>
Anna-Maria	<input type="checkbox"/>

Tag	Text
myLogs	checked:
myLogs	Karl-Heinz
myLogs	Linnéa
myLogs	Hans-Joachim
myLogs	Anna-Lena

Project: P0441_SimpleListEvents

<http://startandroid.ru/ru/uroki/vse-uroki-spiskom/85-urok-44-sobytaja-v-listview.html>

P0441_SimpleListEvents 1

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6
7     <ListView
8         android:id="@+id/lvMain"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content" >
11    </ListView>
12
13 </LinearLayout>
```



P0441_SimpleListEven

Item 1

Sub Item 1

Item 2

Sub Item 2

Item 3

Sub Item 3

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="app_name">P0441_SimpleListEvents</string>
4     <string name="action_settings">Settings</string>
5     <string name="hello_world">Hello world!</string>
6     <string-array name="names">
7         <item>Karl-Heinz</item>
8         <item>Linnéa</item>
9         <item>Hans-Joachim</item>
10        <item>Anna-Lena</item>
11        <item>Lisa-Marie</item>
12        <item>Anna-Maria</item>
13        <item>Elias</item>
14        <item>Luca</item>
15        <item>Alexander</item>
16        <item>Eva-Maria</item>
17        <item>Kajetan</item>
18    </string-array>
19 </resources>
```

P0441_SimpleListEvents 2

```
15 public class MainActivity extends Activity {
16     final String LOG_TAG = "myLogs";
17     ListView lvMain;
18     /** Called when the activity is first created. */
19     public void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.main);
22         lvMain = (ListView) findViewById(R.id.lvMain);
23         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
24             this, R.array.names, android.R.layout.simple_list_item_1);
25         lvMain.setAdapter(adapter);
26         lvMain.setOnItemClickListener(new OnItemClickListener() {
27             public void onItemClick(AdapterView<?> parent, View view,
28                 int position, long id) {
29                 Log.d(LOG_TAG, "itemClick: position = " + position + ", id = "
30                     + id);
31             }
32         });
33         lvMain.setOnItemClickListener(new OnItemSelectedListener() {
34             public void onItemSelected(AdapterView<?> parent, View view,
35                 int position, long id) {
36                 Log.d(LOG_TAG, "itemSelected: position = " + position + ", id = "
37                     + id);
38             }
39         });
40         lvMain.setOnScrollListener(new OnScrollListener() {
41             public void onScrollStateChanged(AdapterView<?> parent, int state) {
42                 Log.d(LOG_TAG, "onScrollStateChanged: state = " + state);
43             }
44             public void onScroll(AbsListView view, int firstVisibleItem,
45                 int visibleItemCount, int totalItemCount) {
46                 Log.d(LOG_TAG, "onScroll: firstVisibleItem = " + firstVisibleItem
47                     + ", visibleItemCount = " + visibleItemCount + ", totalItemCount = "
48                     + totalItemCount);
49             }
50         });
51     }
52 }
53 }
54 }
```


P0441_SimpleListEvents 3

```
33= lvMain.setOnItemSelectedListener(new OnItemSelectedListener() {
34=     public void onItemSelected(AdapterView<?> parent, View view,
35=         int position, long id) {
36=         Log.d(LOG_TAG, "itemSelect: position = " + position + ", id = " + id);
37=     }
38=     public void onNothingSelected(AdapterView<?> parent) {
39=         Log.d(LOG_TAG, "itemSelect: nothing");
40=     }
41= });
42= lvMain.setOnScrollListener(new OnScrollListener() {
43=     public void onScrollStateChanged(AbsListView view, int scrollState) {
44=         Log.d(LOG_TAG, "scrollState = " + scrollState);
45=     }
46=     public void onScroll(AbsListView view, int firstVisibleItem,
47=         int visibleItemCount, int totalItemCount) {
48=         Log.d(LOG_TAG, "scroll: firstVisibleItem = " + firstVisibleItem
49=             + ", visibleItemCount" + visibleItemCount
50=             + ", totalItemCount" + totalItemCount);
51=     }
52= });
```

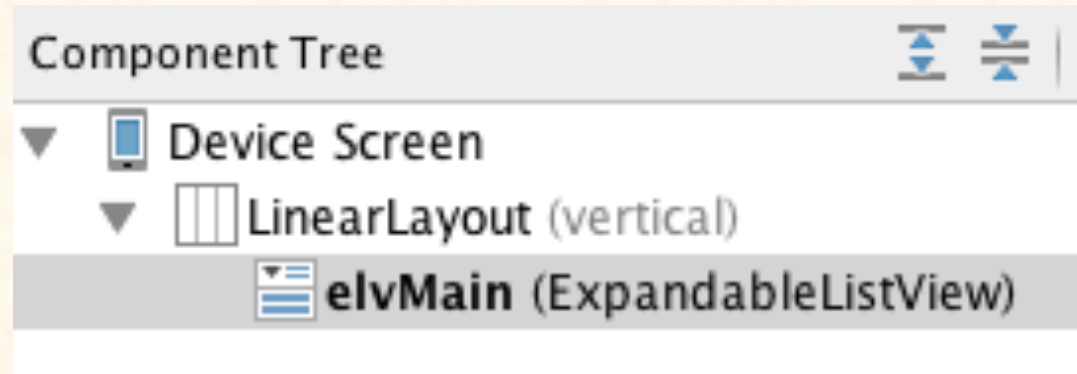
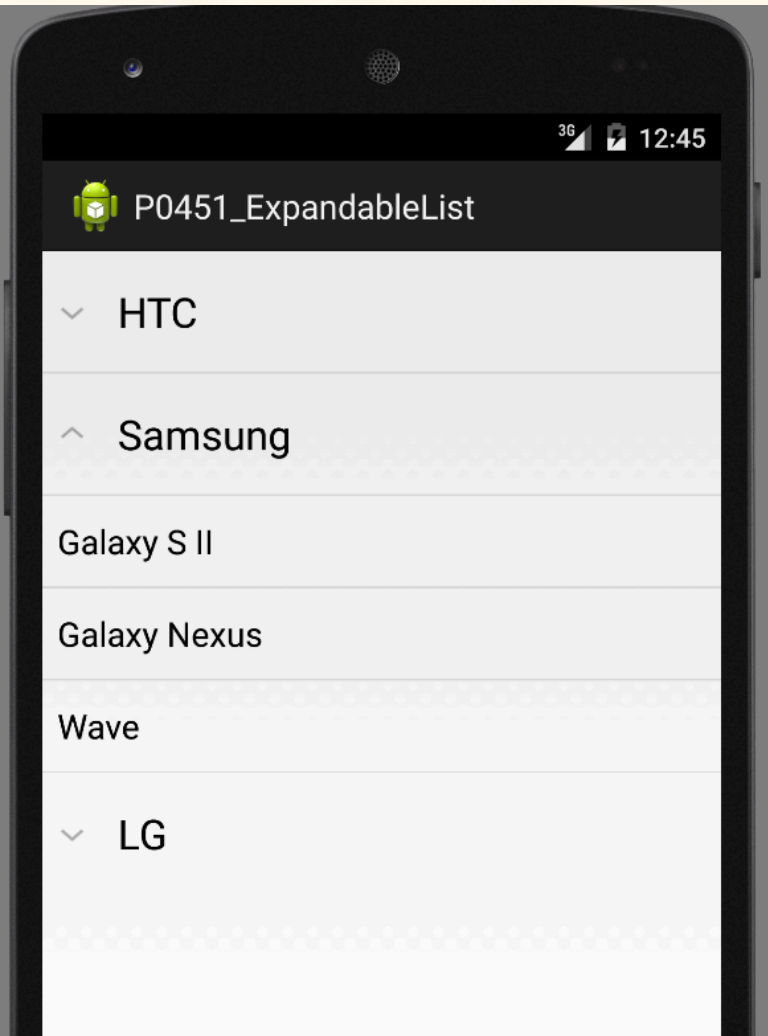
P0441_SimpleListEvents 4

```
myLogs scroll: firstVisibleItem = 0, visibleItemCount11, totalItemCount11  
myLogs scroll: firstVisibleItem = 0, visibleItemCount11, totalItemCount11  
myLogs scroll: firstVisibleItem = 0, visibleItemCount11, totalItemCount11  
myLogs scroll: firstVisibleItem = 0, visibleItemCount11, totalItemCount11  
myLogs scrollState = 0
```


```
myLogs itemClick: position = 5, id = 5
```

P0451_ExpandableList

- Project Beispiel: Expandable List View



P0451_ExpandableList

```
12  public class MainActivity extends Activity {
13     // Group names (Company)
14     String[] groups = new String[] {"HTC", "Samsung", "LG"};
15     // Element names (Phones)
16     String[] phonesHTC = new String[] {"HTC One M8s", "HTC One mini 2",
17     "HTC Desire 620", "HTC Desire EYE"};
18     String[] phonesSams = new String[] {"Galaxy Note5", "Galaxy On7", "Galaxy S6"};
19     String[] phonesLG = new String[] {"LG Y70 Spirit", "LG Pada", "LG Leon", "LG Joy"};
20     // Group collection
21     ArrayList<Map<String, String>> groupData;
22     // Component collection for a single group
23     ArrayList<Map<String, String>> childDataItem;
24     // common collection for element collections
25     ArrayList<ArrayList<Map<String, String>>> childData;
26     // result must be: childData = ArrayList<childDataItem>
27     // attribute list for a group or an element
28     Map<String, String> m;
29     ExpandableListView elvMain;
```

Map

A Map is a data structure consisting of a set of keys and values in which each key is mapped to a single value. The class of the objects used as keys is declared when the Map is declared, as is the class of the corresponding values.

ArrayList is an implementation of [List](#), backed by an array. All optional operations including adding, removing, and replacing elements are supported. All elements are permitted, including null.

P0451_ExpandableList

```
30  /** Called when the activity is first created. */
31  public void onCreate(Bundle savedInstanceState) {
32      super.onCreate(savedInstanceState);
33      setContentView(R.layout.main);
34      // fill a group collection from a group name array
35      groupData = new ArrayList<Map<String, String>>();
36      for (String group : groups) {
37          // fill an attribute list for each group
38          m = new HashMap<String, String>();
39          m.put("groupName", group); // company name
40          groupData.add(m);
41      }
42      // group attribute list for read
43      String groupFrom[] = new String[] {"groupName"};
44      // ID view-element list for a group attributes
45      int groupTo[] = new int[] {android.R.id.text1};
46      // a collection of element collections
47      childData = new ArrayList<ArrayList<Map<String, String>>>();
48      // create a first-group element collection
49      childDataItem = new ArrayList<Map<String, String>>();
50      // fill an attribute list for each element
51      for (String phone : phonesHTC) {
52          m = new HashMap<String, String>();
53          m.put("phoneName", phone); // phone name
54          childDataItem.add(m);
55      }
```


P0451_ExpandableList

```
56 // add the collection to the collection of collections
57 childData.add(childDataItem);
58 // create the second group collection of elements
59 childDataItem = new ArrayList<Map<String, String>>();
60 for (String phone : phonesSams) {
61     m = new HashMap<String, String>();
62     m.put("phoneName", phone);
63     childDataItem.add(m);
64 }
65 childData.add(childDataItem);
66 // create the third group collection of elements
67 childDataItem = new ArrayList<Map<String, String>>();
68 for (String phone : phonesLG) {
69     m = new HashMap<String, String>();
70     m.put("phoneName", phone);
71     childDataItem.add(m);
72 }
73 childData.add(childDataItem);
```

P0451_ExpandableList

```
79 SimpleExpandableListAdapter adapter = new SimpleExpandableListAdapter(  
80     this,  
81     groupData,  
82     android.R.layout.simple_expandable_list_item_1,  
83     groupFrom,  
84     groupTo,  
85     childData,  
86     android.R.layout.simple_list_item_1,  
87     childFrom,  
88     childTo);  
89 elvMain = (ExpandableListView) findViewById(R.id.elvMain);  
90 elvMain.setAdapter(adapter);  
91 }  
92 }
```

SimpleExpandableListAdapter

extends [BaseExpandableListAdapter](#)

An easy adapter to map static data to group and child views defined in an XML file. You can separately specify the data backing the group as a List of Maps. Each entry in the ArrayList corresponds to one group in the expandable list. The Maps contain the data for each row.

P0451_ExpandableList

- Ein Tip: ein Event-Listener mit

```
public boolean onChildClick(android.widget.ExpandableListView parent,  
    View v, int groupPosition, int childPosition, long id) {  
  
    String yourText = v.getText().toString(); // Get the text from the view and put it in a string  
    // use string as you need to  
}
```

ListView

- An advanced Tutorial:

Using lists in Android (ListView) – Tutorial

<http://www.vogella.com/tutorials/AndroidListView/article.html>

Table of Contents

1. Android and Lists <ul style="list-style-type: none">1.1. Using lists in Android1.2. Views for handling lists1.3. Possible input types for lists1.4. Adapters1.5. Filtering and sorting1.6. Data updates in the adapter1.7. Listener	5. Exercise: Using ListView and ListActivity 6. Exercise: ListActivity with own layout 7. Tutorial: Implementing your own adapter 8. ListViews and performance <ul style="list-style-type: none">8.1. Motivation8.2. Time consuming operations8.3. Avoiding layout inflation and object creation8.4. Holder Pattern8.5. Example	17. SimpleCursorAdapter 18. Additional Open Source libraries 19. Support free vogella tutorials <ul style="list-style-type: none">19.1. Thank you19.2. Questions and Discussion
2. Default adapter <ul style="list-style-type: none">2.1. Default platform adapter2.2. Using ArrayAdapter2.3. ListView example with ArrayAdapter	9. Storing the selection of a view 10. Contextual action mode for ListViews 11. Implementing undo for an action <ul style="list-style-type: none">11.1. When should you offer an undo action?11.2. Example	20. Links and Literature <ul style="list-style-type: none">20.1. Source Code20.2. ListView Resources20.3. vogella Resources
3. Custom adapter implementations <ul style="list-style-type: none">3.1. Developing a custom adapter3.2. Preparing a row for the list3.3. Example for a custom adapter3.4. Updating the data model from the adapter	12. Performance Optimization 13. Tutorial: How to display two items in a ListView 14. Selecting multiple items in the ListView <ul style="list-style-type: none">14.1. Interaction between the model and Listview14.2. Tutorial: Domain Model and Rows interaction	
4. ListActivity and ListFragment <ul style="list-style-type: none">4.1. Default container for using ListView4.2. ListActivity and custom layout4.3. Placeholder for an empty list	15. Implementing an expandable ListView <ul style="list-style-type: none">15.1. ExpandableListView15.2. ExpandableListView example 16. Tutorial: Miscellaneous <ul style="list-style-type: none">16.1. Adding a longclick listener to the list items16.2. Header and Footer	

ListView

Fragen?