IT ШКОЛА SAMSUNG

Модуль 4. Структуры данных

Урок 3. Адаптеры.

SAMSUNG

СПИСОК НА ЭКРАНЕ

Одним из наиболее часто используемых элементов интерфейса является **список.**
Его реализация при создании Android приложений имеет ряд особенностей: для вывода списка используется виджет **ListView**, а чтобы определить его содержание и структуру, в которой они будут храниться - свяанный с ним **адаптер**.

- Зачем адаптер?
- Чтобы сохранить оперативную память.

АДАПТЕРЫ

Базовый адаптер – **BaseAdapter**
Его потомки**:**

● **ArrayAdapter<T> -** предназначен для работы с ListView. Данные представлены в виде массива, которые размещаются в отдельных элементах TextView**;**
● **ListAdapter -** адаптер между ListView и данными. Строго говоря, это класс-интерфейс, который можно использовать и в ArrayAdapter и в SimpleAdapter и т.д**.;**
● **SpinnerAdapter -** адаптер для связки данных с элементом Spinner. Это тоже интерфейс, как ListAdapter и работает по схожему принципу**;**
● **SimpleAdapter -** адаптер, позволяющий заполнить данными список более сложной структуры, например, два текста в одной строке списка**;**
● **SimpleCursorAdapter -** дополняет ResourceCursorAdapter и создаёт компоненты TextView/ImageView из столбцов, содержащихся в курсоре. Компоненты определяются в ресурсах;
● **CursorAdapter -** предназначен для работы с ListView, предоставляет данные для списка через курсор, который должен иметь колонку с именем "_id";
● **ResourceCursorAdapter -** этот адаптер дополняет CursorAdapter и может создавать виды из ресурсов;
● **HeaderViewListAdapter -** расширенный вариант ListAdapter, когда ListView имеет заголовки**.**
● **WrapperListAdapter -** еще один адаптер для списков**.**

ПРИМЕР

Требуется вывести название двенадцати месяцев на экран устройства в виде списка.

Реализуем программу с помощью адаптера ArrayAdapter и активности ListActivity.

Использовать ListActivity удобно тогда, когда вам нужно вывести на экран исключительно список, без других элементов интерфейса.

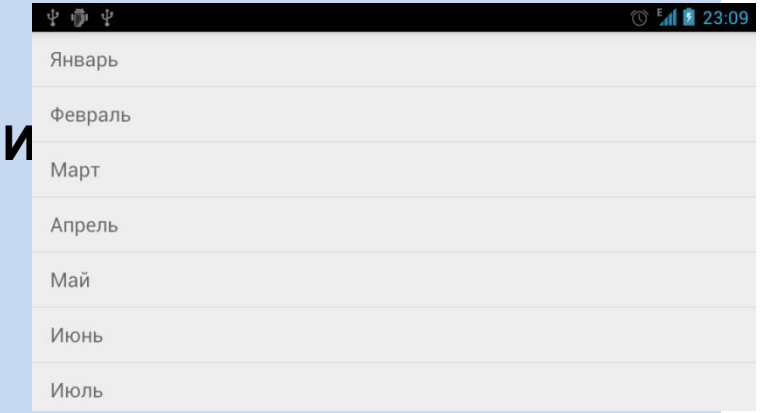
Создадим **новый проект** , родителем главной активности назначим **ListActivity.**

ПРИМЕР

```
import android.app.ListActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends ListActivity {
String[] myArr = {"Январь", "Февраль", "Март", "Апрель", "Май", "И
"Сентябрь", "Октябрь", "Ноябрь", "Декабрь"};

@Override
protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      ArrayAdapter<String> monthAdapter = new     ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, myArr);
      setListAdapter(monthAdapter);
      }
}
```



АДАПТЕР В РАБОТЕ

Не важно сколько элементов в вашем массиве (списке), который нужно вывести на экран. Будет создано столько визуальных элементов списка, сколько вмещается на экран +1. Если бы этого механизма не существовало, то для 1000 000 текстовых элементов массива системе пришлось бы создать 1000 000 экземпляров TextView для их отображения. Это повлекло бы крах приложения, так как память тут же бы закончилась.

ВЫБОР ЭЛЕМЕНТА СПИСКА

Как отследить пункт на который нажал пользователь?

Для этих целей в классе ListActivity реализован метод onListItemClick.

```
@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
String month = (String) getListAdapter().getItem(position);
Toast.makeText(this, month, Toast.LENGTH_SHORT).show();
}
```

С помощью метода getListAdapter() осуществляется доступ к адаптеру. Имея его, можно получить доступ к данным, которые выводятся на экран, в частности с помощью метода getItem().

ИЗМЕНЕНИЕ ЭЛЕМЕНТА СПИСКА

Если захочется динамически изменить элемент отображаемого массива :

```
monthArr[11] = "!TEST!";
monthAdapter.notifyDataSetInvalidated();
```

УПРАЖНЕНИЕ

Разработать приложение для отображения списка планет солнечной системы. При нажатии на элемент списка выводится описание планеты выбранного элемента.

ВЫВОД СЛОЖНОГО СПИСКА

Для списков с несколькими надписями, изображениями и контролами можно использовать адаптер SimpleAdapter.

Работу с ним можно разделить на следующие этапы:
1. Создание разметки;
2. Создание списка данных для отображения в адаптере. Элементы списка состоят из объектов типа HashMap (ключ - значение);
3. Создание массива атрибутов для сопоставления элементов списка с элементами разметки;
4. Создание массива идентификаторов разметки;
5. Создание самого адаптера.

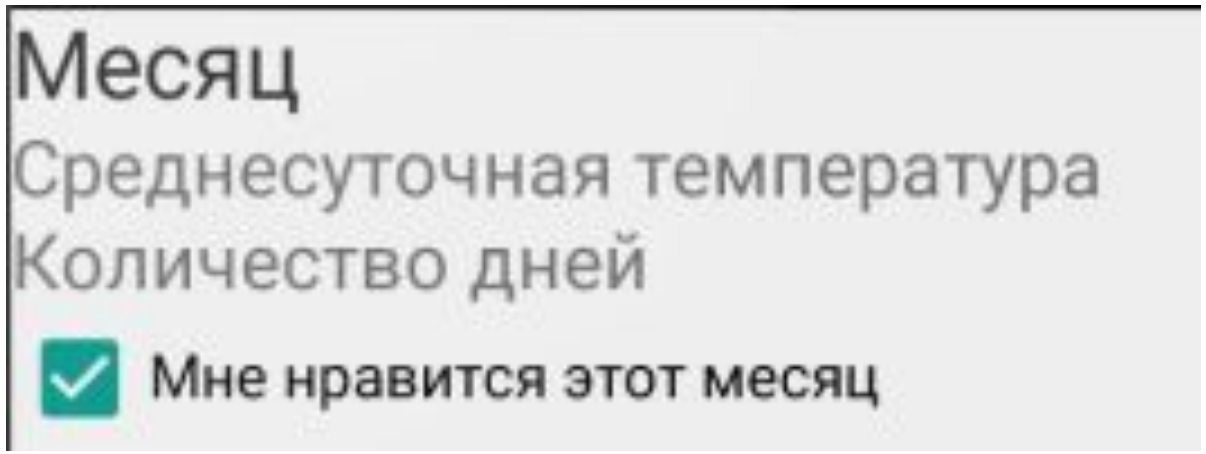
ПРИМЕР – ОСНОВНОЙ ФАЙЛ РАЗМЕТКИ

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<ListView
android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_alignParentBottom="true"
android:layout_alignParentEnd="true"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true" />
</RelativeLayout>
```

ПРИМЕР – РАЗМЕТКА ЭЛЕМЕНТА СПИСКА

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >
<ImageView
android:id="@+id/imageView"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Месяц"
android:textAppearance="?android:attr/textAppearanceLarge" />
<TextView
android:id="@+id/textView2“
android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Среднесуточная температура"
android:textAppearance="?android:attr/textAppearanceMedium" />
<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Количество дней"
android:textAppearance="?android:attr/textAppearanceMedium" />
<CheckBox
android:id="@+id/checkBox"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:checked="true"
android:text="Мне нравится этот месяц" />
</LinearLayout>
```

ПРИМЕР – ВИД ЭЛЕМЕНТА СПИСКА



ПРИМЕР – ПОДГОТОВКА ДАННЫХ

```
// Названия месяцев
String[] monthArr = { "Январь", "Февраль", "Март", "Апрель", "Май", "Июнь",
"Июль", "Август", "Сентябрь", "Октябрь", "Ноябрь", "Декабрь" };
// Среднесуточная температура
String[] tempArr = { "-12.7", "-11.3", "-4.5", "7.7", "19.3", "23.9", "23.5", "22.8", "16.0", "5.2", "-0.3", "-9.3" };
// Количество дней
String[] dayCArr = { "31", "28", "31", "30", "31", "30", "31", "31", "30", "31", "30", "31" };
@Override
protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);
                                    setContentView(R.layout.activity_main);
ArrayList<HashMap<String, Object>> data = new ArrayList<HashMap<String, Object>>(monthArr.length);
HashMap<String, Object> map; //HashMap – ключ-значение
      for (int i = 0; i < monthArr.length; i++) {
            map = new HashMap<String, Object>();
            map.put("month", monthArr[i]);
            map.put("temp", tempArr[i]);
            map.put("day", dayCArr[i]);
            map.put("like", true);
            map.put("img", R.drawable.sun);
            data.add(map);
      } }
```

ПРИМЕР

В папку res/drawable сохраняем файл sun с изображением солнышка.

Создание массива атрибутов //ключ

```
// Атрибуты элементов
String[] from = { "month", "temp", "day", "like", "img" };
```

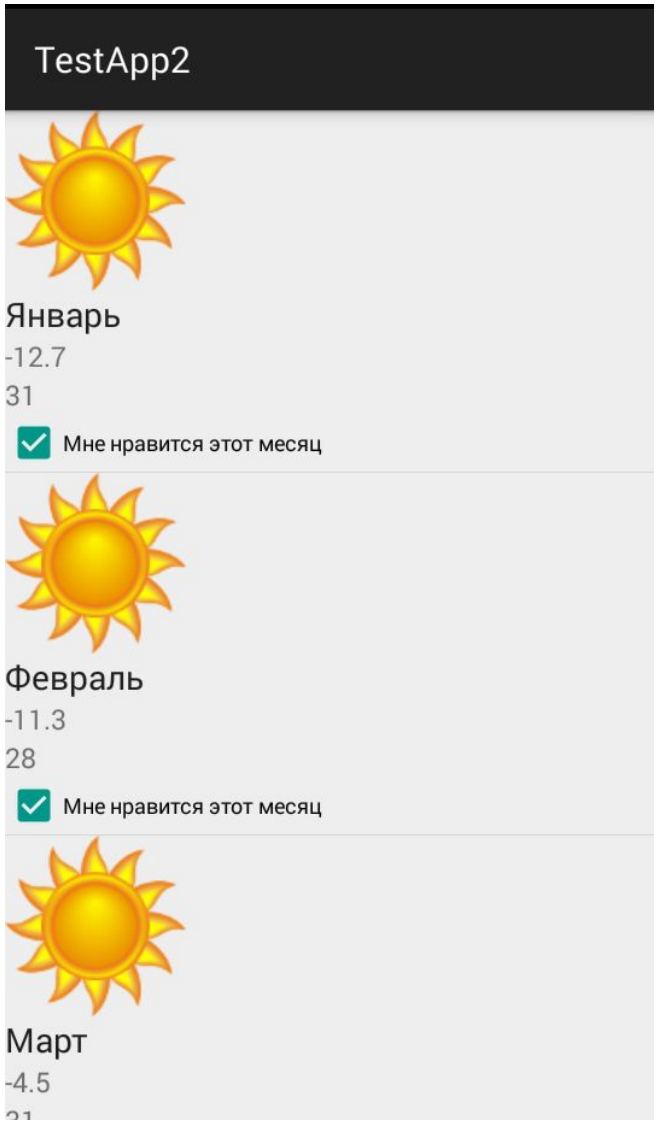
Создание массива идентификаторов разметки//значение

```
// Идентификаторы
int[] to = { R.id.textView, R.id.textView2, R.id.textView3,
R.id.checkBox, R.id.imageView };
```

ПРИМЕР

Создание адаптера с заполненными данными и назначение списку

```
// Создание адаптера
SimpleAdapter adapter = new SimpleAdapter(this, data,
R.layout.adapter_item, from, to);
ListView lv = (ListView) findViewById(R.id.listView);
lv.setAdapter(adapter);
```



СОБСТВЕННЫЙ АДАПТЕР

Возьмём под контроль ArrayAdapter, будем сами назначать, что ему показывать и где.

В новом примере адаптер будет снова показывать погоду, но на этот раз сохранять состояние CheckBox и для месяцев с отрицательной температурой будет выводиться другая картинка из файла snow. Разметка адаптера остаётся та же.

ПРИМЕР – НОВАЯ СТРУКТУРА ДАННЫХ

Для хранения параметров каждого месяца будем использовать отдельный класс MyMonth .

```
public class MyMonth {
    String month = ""; // Название месяца
    double temp = 0.; // Средняя температура
    int days = 0; // Количество дней
    boolean like = true; // Нравится месяц
}
```

Для заполнения массива месяцев (данные которого будут выводиться с помощью адаптера), в главной активности, создадим метод создания массива из объектов *MyMonth.* Конструктор активности остаётся практически без изменений.

ПРИМЕР - АКТИВНОСТЬ

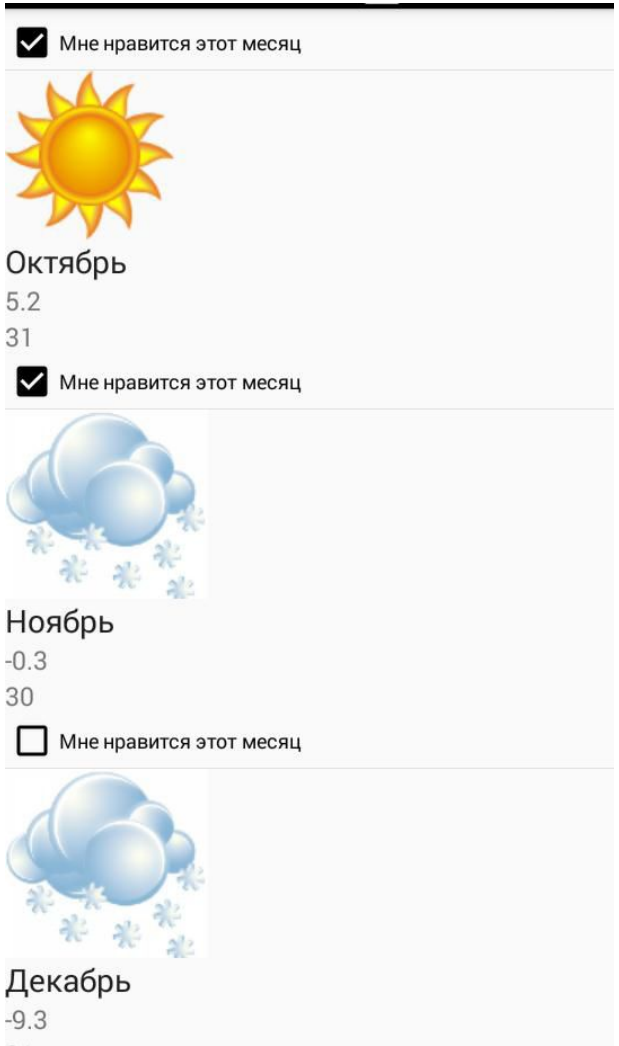
```
public class MainActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

MyMonthAdapter adapter = new MyMonthAdapter(this, makeMonth());
ListView lv = (ListView) findViewById(R.id.listView);
lv.setAdapter(adapter); }
// Метод создания массива месяцев
MyMonth[] makeMonth() {
MyMonth[] arr = new MyMonth[12];
// Названия месяцев
String[] monthArr = {"Январь", "Февраль", "Март", "Апрель", "Май", "Июнь", "Июль", "Август", "Сентябрь", "Октябрь",
"Ноябрь", "Декабрь"};
// Среднесуточная температура
double[] tempArr = {-12.7, -11.3, -4.5, 7.7, 19.3, 23.9, 23.5, 22.8, 16.0, 5.2, -0.3, -9.3};
// Количество дней
int[] dayArr = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
// Сборка месяцев
for (int i = 0; i < arr.length; i++) {
MyMonth month = new MyMonth();
month.month = monthArr[i];
month.temp = tempArr[i];
month.days = dayArr[i];
arr[i] = month;
}
return arr; } }
```

ПРИМЕР – НОВЫЙ АДАПТЕР

```
public class MyMonthAdapter extends ArrayAdapter<MyMonth> {
public MyMonthAdapter(Context context, MyMonth[] arr) {
super(context, R.layout.adapter_item, arr); }
@Override
public View getView(int position, View convertView, ViewGroup parent) {
        final MyMonth month = getItem(position);
        if (convertView == null) {
                convertView = LayoutInflater.from(getContext()).inflate(R.layout.adapter_item, null);
        }
// Заполняем адаптер
        ((TextView) convertView.findViewById(R.id.textView)).setText(month.month);
        ((TextView) convertView.findViewById(R.id.textView2)).setText(String.valueOf(month.temp));
        ((TextView) convertView.findViewById(R.id.textView3)).setText(String.valueOf(month.days));
// Выбираем картинку для месяца
        if (month.temp > 0.)
                ((ImageView) convertView.findViewById(R.id.imageView)).setImageResource(R.drawable.sun);
        else
                ((ImageView) convertView.findViewById(R.id.imageView)).setImageResource(R.drawable.snow);
        CheckBox ch = (CheckBox) convertView.findViewById(R.id.checkBox);
        ch.setChecked(month.like);
        ch.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                month.like = ((CheckBox) v).isChecked();
                }
        });
return convertView; }}
```

ПРИМЕР


Мне нравится этот месяц
Октябрь
5.2
31
Мне нравится этот месяц
Ноябрь
-0.3
30
Мне нравится этот месяц
Декабрь
-9.3

ЗАДАНИЕ

Разработайте приложение, в результате работы которого на экран будет выведена информация о всех фильмах получивших «Оскар», как лучший фильм года, с момента основания киноакадемии. Обязательно укажите название фильма, режиссера, год победы, эмблему студии. Для вывода информации используйте собственный класс наследник ArrayAdapter.

Спасибо!

SAMSUNG