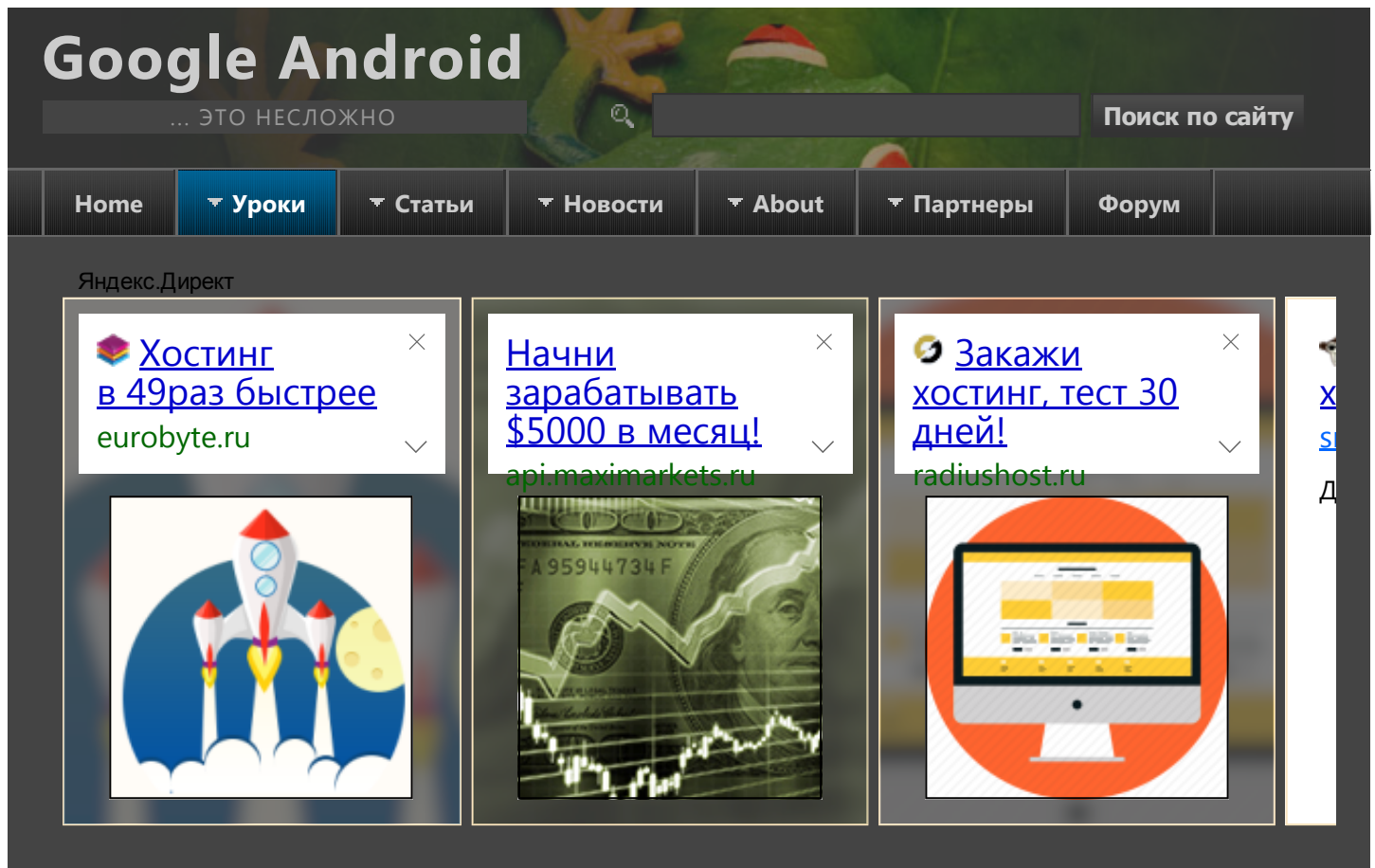




LANGUAGE



Урок 41. Используем LayoutInflater для создания списка



Присоединяйтесь к
Jabber чату
StartAndroid

ВАКАНСИИ РАЗРАБОТЧИКС

На нашем форуме
рекрутеры
периодически
пополняют [список
вакансий](#) Android
разработчиков.
Будет время -
загляните,
возможно вас
заинтересуют эти
предложения.

МАТЕРИАЛЫ ПО СМЕЖНЫМ ТЕМАМ

- [Урок 40. LayoutInflater. Учимся использовать.](#)
- [Урок 42. Список - ListView](#)
- [Урок 43. Одиночный и множественный выбор в ListView](#)
- [Урок 44. События в ListView](#)
- [Урок 45. Список-дерево ExpandableListView](#)
- [Урок 46. События ExpandableListView](#)
- [Урок 51. SimpleAdapter, добавление и удаление записей](#)
- [Урок 52. SimpleCursorAdapter, пример использования](#)
- [Урок 62. Диалоги. AlertDialog. Список](#)
- [Урок 63. Диалоги. AlertDialog. Список с одиночным выбором](#)
- [Урок 108. Android 3. ActionBar. Навигация - табы и выпадающий список](#)



Создано 12.12.2011 08:00 Автор: damager82

В этом уроке:

- делаем свой вариант списка



ПИАР-ПОДДЕРЖКА

Вы создали приложение/проект/стартап, о котором хотите рассказать? У меня к вам есть [предложение](#).

ПОДДЕРЖКА ПРОЕКТА

[Яндекс](#)

410011180491924

Alfa-Bank

5486734918678877

[WebMoney](#)

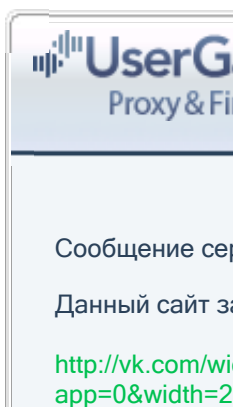
R248743991365

Z551306702056

[ePayService](#)

D434155

PayPal



На прошлом уроке мы узнали, зачем нужен класс **LayoutInflater** и сделали небольшой пример, на котором подробно рассмотрели метод **inflate** и его параметры. Для закрепления темы сделаем еще один, чуть более сложный пример.

Мы сделаем свой аналог **списка**. Для начала придумаем данные. Пусть это снова будет **штатное расписание** с именами работников, должностями и зряплатой) Т.е. каждый пункт нашего списка будет содержать три текстовых не редактируемых поля - **name**, **position**, **salary**. А пункты мы разместим в виде вертикального списка.

Для реализации нам понадобятся **два layout-файла**:

main.xml - основной экран для Activity, контейнер для пунктов списка

item.xml - экран с **FrameLayout** и тремя текстовыми полями в нем. Это будет пункт списка.

Приложение будет параллельно перебирать **три массива данных**, **создавать** для каждой тройки **View-элемент** из layout-файла **item.xml**, **заполнять** его данными и **добавлять** в вертикальный **LinearLayout** в **main.xml**.

Создадим проект:

Project name: P0411_LayoutInflaterList

Build Target: Android 2.3.3

Application name: LayoutInflaterList

Package name: ru.startandroid.develop.p0411layoutinflaterlist

Create Activity: MainActivity

Экран **main.xml**:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Staff List"
        android:layout_gravity="center_horizontal">
    </TextView>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scroll">
        <LinearLayout
            android:id="@+id/LinLayout"
```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
    </LinearLayout>
</ScrollView>
</LinearLayout>

```

ScrollView обеспечит нам **прокрутку** списка, если все пункты не влезут в экран. А в нем **LinearLayout**, в который мы будем добавлять элементы.

Экран **item.xml**:

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginTop="10dp">
    <TextView
        android:id="@+id/tvName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:layout_gravity="top/center_horizontal"
        android:textSize="24sp">
    </TextView>
    <TextView
        android:id="@+id/tvPosition"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:layout_gravity="bottom/left"
        android:layout_marginLeft="5dp">
    </TextView>
    <TextView
        android:id="@+id/tvSalary"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:layout_gravity="bottom/right"
        android:layout_marginRight="5dp">
    </TextView>
</FrameLayout>

```

FrameLayout, и три **TextView** в нем.

Кодим реализацию. **MainActivity.java**:

```
package ru.startandroid.develop.p0411layoutinflaterlist;
```



FOLLOW US ON [tw](#)

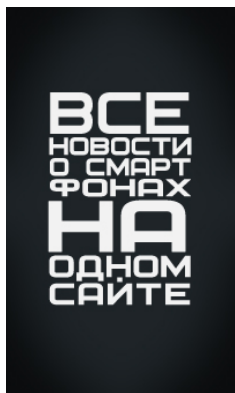
Сборник уро



Google And

... это несл

Рекламное м
свободно



Яндекс.Директ



[Выбираете](#)

ХОСТИНГ?

ТОП 10 хостинг-провайдеров на основе анализа 60 000+ отзывов о хостингах

[Российский](#)

[хостинг](#)

[Американский](#)

[хостинг](#)

[hosting-rate.net](#)



Мансардные окна VELUX

Качество, надежность, скандинавский дизайн. От 9.700 рублей за окно.

[velux.ru](#)

Полукоммер линолеум Tarkett

Tarkett Force. Низкие цены. Доставка. Укладка. СПб

[Коммерческий](#)

[линолеум](#)

[Дизайн](#)

[плитка ПВХ](#)

[Сценический](#)

[линолеум](#)

[irbis-pol.ru](#)

Адрес и телефон

```
import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.widget.LinearLayout;
import android.widget.TextView;

public class MainActivity extends Activity {

    String[] name = { "Иван", "Марья", "Петр", "Антон", "Даша", "Борис",
        "Костя", "Игорь" };
    String[] position = { "Программер", "Бухгалтер", "Программер",
        "Программер", "Бухгалтер", "Директор", "Программер", "Охранник" };
    int salary[] = { 13000, 10000, 13000, 13000, 10000, 15000, 13000, 8000 };

    int[] colors = new int[2];

    /** Called when the activity is first created. */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        colors[0] = Color.parseColor("#559966CC");
        colors[1] = Color.parseColor("#55336699");

        LinearLayout linLayout = (LinearLayout) findViewById(R.id.linLayout);

        LayoutInflater ltInflater = getLayoutInflater();

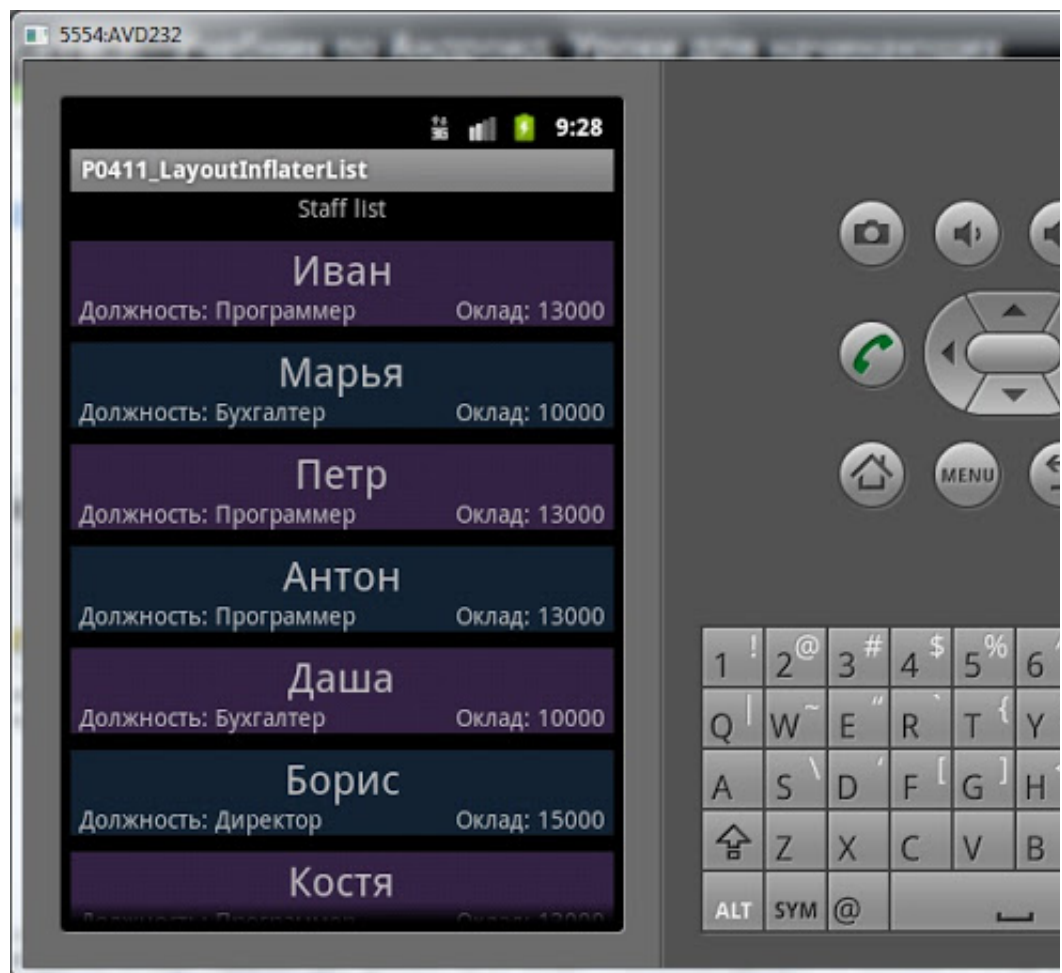
        for (int i = 0; i < name.length; i++) {
            Log.d("myLogs", "i = " + i);
            View item = ltInflater.inflate(R.layout.item, linLayout, false);
            TextView tvName = (TextView) item.findViewById(R.id.tvName);
            tvName.setText(name[i]);
            TextView tvPosition = (TextView) item.findViewById(R.id.tvPosition);
            tvPosition.setText("Должность: " + position[i]);
            TextView tvSalary = (TextView) item.findViewById(R.id.tvSalary);
            tvSalary.setText("Оклад: " + String.valueOf(salary[i]));
            item.setLayoutParams().width = LayoutParams.MATCH_PARENT;
            item.setBackgroundColor(colors[i % 2]);
            linLayout.addView(item);
        }
    }
}
```

Не так уж много нужно кода, чтобы сделать несложный список. Мы запускаем **цикл** по кол-ву элементов в массивах данных. В каждой итерации **создаем View**-элемент **item** из layout-файла **item.xml**. В нашем случае **item** - это **FrameLayout**, который содержит три **TextView**. Мы их находим в **созданном item** и **заполняем данными** из массивов.

В методе **inflate** мы указали **root** - **linLayout**, чтобы получить от него **LayoutParams** и далее использовать для настройки ширины. Также для наглядности раскрашиваем пункты методом [setBackgroundColor](#).

Обратите внимание - третий параметр **inflate** мы указали **false**. Т.е. мы **не стали сразу добавлять** создаваемый View-элемент к **linLayout**, а делаем это в конце кода методом **addView**. Этому есть объяснение. Если бы мы указали **true** - то метод добавил бы **item** к **linLayout** и вернул бы нам **linLayout**, общий для всех пунктов списка. Через **linLayout** заполнять **TextView** необходимым нам текстом было бы затруднительно. Поэтому мы получаем пункт **item** (**FrameLayout**), заполняем его **TextView** данными и только потом помещаем к остальным пунктам в **linLayout** методом **addView**.

Все сохраним и запустим:



Список удался и работает прокрутка.

Урок получился короткий, но полезный. На всякий случай хочу заметить, что это еще не классический Android-список называемый **List**. Но этот пример значительно облегчит понимание списка. Т.к. принцип схож. Для построения **List** мы также должны будем предоставлять массив данных и **layout**-файл для пунктов. Этим и займемся на следующем уроке.

Этот урок был придуман и написан на высоте 10 км., в самолете на пути в Москву. Через ряд, слева от меня сидел парень с рюкзаком **java**, распечатанными доками, ноутбуком и чего-то кодил под **Android**.

Android повсюду :)

На следующем уроке:

- используем ListView для построения списка

► [Обсудить на форуме \[140 replies\]](#)

[< Назад](#) [Вперёд >](#)

Спасибо за вашу поддержку сайта!

100 руб.

VISA

MasterCard

Яндекс



Поддержать

Перевод проекту [StartAndroid](#)

▲ TOP

При использовании материалов сайта ссылка на startandroid.ru обязательна.

T3 Framework
FAST. FLEXIBLE. POWERFUL.

Liveinternet
2
СТАТИСТИКА

Rambler
ТОП100

6 267
3 047
2 398