



Освой программирование игр

Сайт Александра Климова



(<http://developer.alexanderklimov.ru/>)

/ Моя кошка замечательно разбирается в программировании. Стоит мне объяснить проблему ей - и все становится ясно. */*

John Robbins, Debugging Applications, Microsoft Press, 2000



(<http://feeds.feedburner.com/alexanderklimov/VJcl>)

Android (<http://developer.alexanderklimov.ru/android>)

C#/Visual Basic (<http://developer.alexanderklimov.ru/dotnet/>)

Windows Phone (<http://developer.alexanderklimov.ru/windowsphone/wp.php>)

WPF (<http://developer.alexanderklimov.ru/wpf/wpf.php>)

PHP (<http://developer.alexanderklimov.ru/php>)

Arduino (<http://developer.alexanderklimov.ru/arduino>)

Главная (<http://developer.alexanderklimov.ru/android/index.php>)

Теория (<http://developer.alexanderklimov.ru/android/theory/>)

Palette (<http://developer.alexanderklimov.ru/android/views.php>)

ListView (<http://developer.alexanderklimov.ru/android/listview/>)

Котошоп (<http://developer.alexanderklimov.ru/android/catshop/>)

Анимация (<http://developer.alexanderklimov.ru/android/animation/>)

SQLite (<http://developer.alexanderklimov.ru/android/sqlite/>)

OpenGL ES (<http://developer.alexanderklimov.ru/android/opengles/>)

Библиотеки (<http://developer.alexanderklimov.ru/android/library/>)

Игры (<http://developer.alexanderklimov.ru/android/games/>)

Wear (<http://developer.alexanderklimov.ru/android/wear/>)

Эмулятор (<http://developer.alexanderklimov.ru/android/emulator/>)

Советы (<http://developer.alexanderklimov.ru/android/tips-android.php>)

Статьи (<http://developer.alexanderklimov.ru/android/articles-android.php>)

Книги (<http://developer.alexanderklimov.ru/android/books.php>)

Java. Экспресс-курс (<http://developer.alexanderklimov.ru/android/java/java.php>)

Дизайн (<http://developer.alexanderklimov.ru/android/design/>)

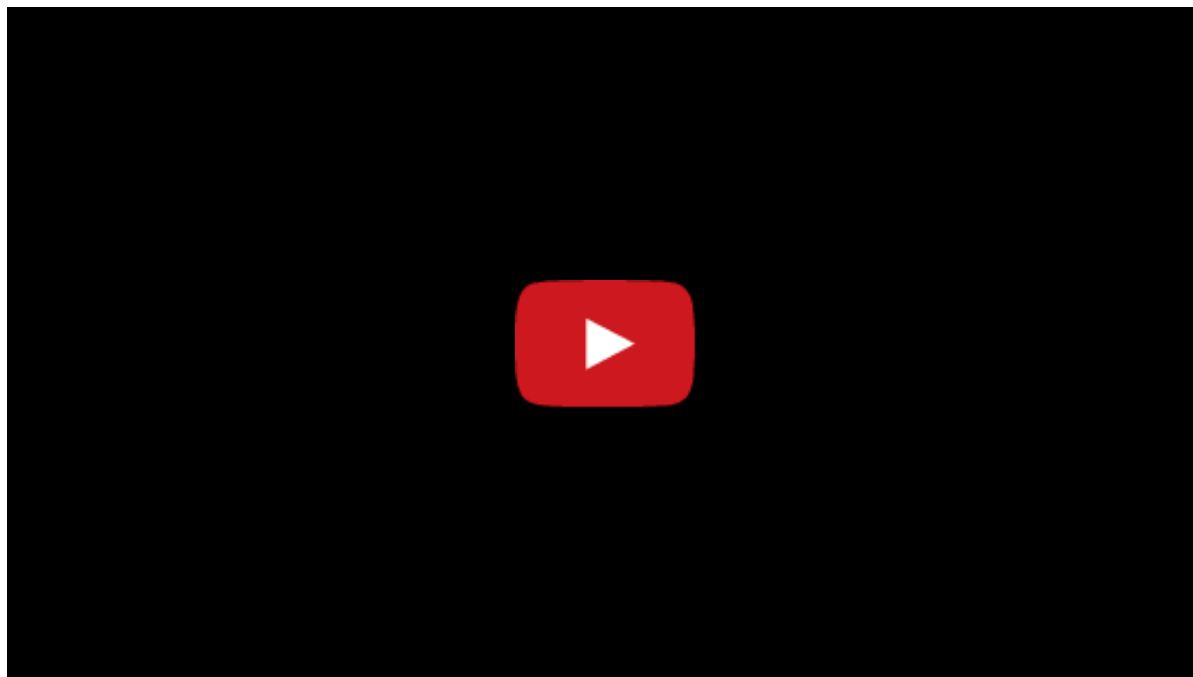
Отладка (<http://developer.alexanderklimov.ru/android/debug/>)

Open Source (<http://developer.alexanderklimov.ru/android/opensource.php>)

Полезные ресурсы (<http://developer.alexanderklimov.ru/android/links.php>)

Список с множественным выбором и собственным адаптером ArrayAdapter

В канун Рождества самые известные кошки интернета снялись в клипе на песню **Hard to be a Cat at Christmas** (Трудно быть котом в Рождество). В съёмках ролика приняли участие кошка по имени Соус Тартар, известная в интернете как Хмурый кот (Grumpy Cat), Полковник Мяу (Colonel Meow), слепой кот Оскар (Oskar the Blind Cat), косоглазая кошка Нала и кот-хипстер Гамильтон. В клипе коты смотрят на снегопад за окном, распаковывают рождественские подарки, играют с елочными украшениями и лакомятся рождественским ужином.



Напишем пример выбора лучших котиков из этого клипа.

Разметка

```

<LinearLayout 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"
    android:orientation="vertical" >

    <Button
        android:id="@+id/buttonGet"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="Ваш выбор" />

    <ListView
        android:id="@+id/listView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>

```

Создадим разметку для отдельного пункта списка (**list_item.xml**), которая будет состоять из изображения и флажка. В нашем случае все значки будут одинаковыми.

```

<?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="wrap_content"
    android:orientation="horizontal" >

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_gravity="left"
        android:src="@drawable/ic_android_cat" />

    <CheckedTextView
        android:id="@+id/checkedTextView"
        android:layout_width="match_parent"
        android:layout_height="?android:attr/listPreferredItemHeightSmall"
        android:layout_gravity="right"
        android:checkMark="?android:attr/listChoiceIndicatorMultiple"
        android:gravity="right|center_vertical"
        android:paddingLeft="?android:attr/listPreferredItemPaddingLeft"
        android:textAppearance="?android:attr/textAppearanceListItemSmall" />

</LinearLayout>

```

Подготовим небольшой массив из имён котов и передадим его адаптеру на основе **ArrayAdapter**. При нажатии на элементах списка флажок будет менять своё состояние на противоположное. Через кнопку мы можем получить информацию, какие флажки были выбраны.

```
package ru.alexanderklimov.test;

import ...

public class MainActivity extends Activity {

    ListView listView;

    private ArrayList<String> catNameList = new ArrayList<String>();

    MyArrayAdapter myArrayAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        listView = (ListView) findViewById(R.id.listView);

        initList();

        myArrayAdapter = new MyArrayAdapter(this, R.layout.list_item,
            android.R.id.text1, catNameList);

        listView.setAdapter(myArrayAdapter);
        listView.setOnItemClickListener(myOnItemClickListener);

    }

    public void onClick(View v) {
        String result = "";
        // getCheckedItems
        List<String> resultList = myArrayAdapter.getCheckedItems();
        for (int i = 0; i < resultList.size(); i++) {
            result += String.valueOf(resultList.get(i)) + "\n";
        }

        myArrayAdapter.getCheckedItemPositions().toString();
        Toast.makeText(getApplicationContext(), result, Toast.LENGTH_LONG)
            .show();

    }

    private void initList() {
```

```

        catNameList.add("Grumpy Cat");
        catNameList.add("Colonel Meow");
        catNameList.add("Oskar the Blind Cat");
        catNameList.add("Nala Cat");
        catNameList.add("Hamilton the Hipster Cat");
    }

    OnItemClickListener myOnItemClickListener = new OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
                                long id) {
            myArrayAdapter.toggleChecked(position);
        }
    };

    private class MyArrayAdapter extends ArrayAdapter<String> {

        private HashMap<Integer, Boolean> myChecked = new HashMap<Integer, Boolean>();

        public MyArrayAdapter(Context context, int resource,
                                int textViewResourceId, List<String> objects) {
            super(context, resource, textViewResourceId, objects);

            for (int i = 0; i < objects.size(); i++) {
                myChecked.put(i, false);
            }
        }

        public void toggleChecked(int position) {
            if (myChecked.get(position)) {
                myChecked.put(position, false);
            } else {
                myChecked.put(position, true);
            }

            notifyDataSetChanged();
        }

        public List<Integer> getCheckedItemPositions() {
            List<Integer> checkedItemPositions = new ArrayList<Integer>();

            for (int i = 0; i < myChecked.size(); i++) {
                if (myChecked.get(i)) {
                    (checkedItemPositions).add(i);
                }
            }

            return checkedItemPositions;
        }
    }

```

```

public List<String> getCheckedItems() {
    List<String> checkedItems = new ArrayList<String>();

    for (int i = 0; i < myChecked.size(); i++) {
        if (myChecked.get(i)) {
            (checkedItems).add(catNameList.get(i));
        }
    }

    return checkedItems;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View row = convertView;

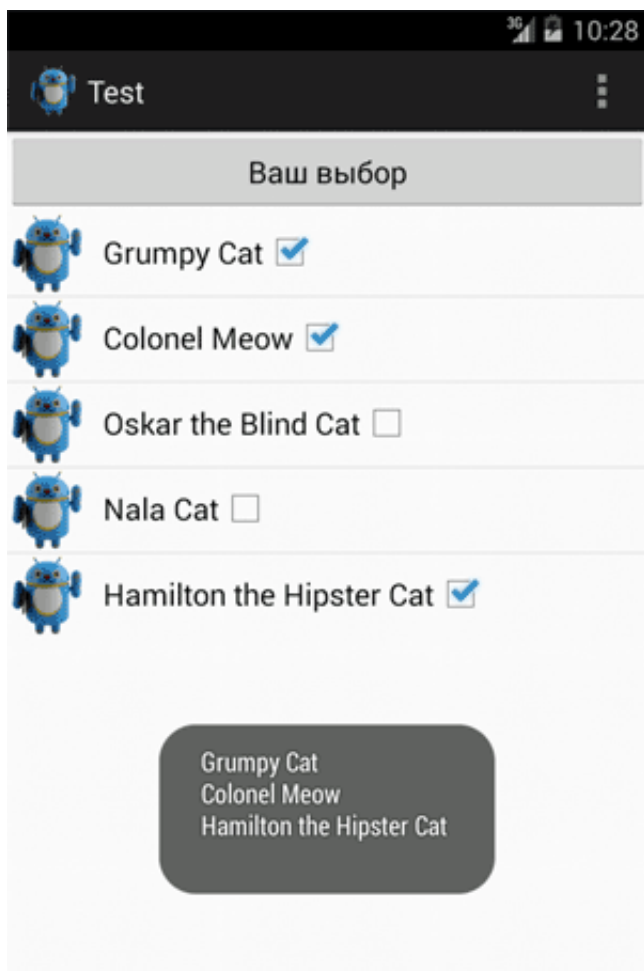
    if (row == null) {
        LayoutInflater inflater = getLayoutInflater();
        row = inflater.inflate(R.layout.list_item, parent, false);
    }

    CheckedTextView checkedTextView = (CheckedTextView) row
        .findViewById(R.id.checkedTextView);
    checkedTextView.setText(catNameList.get(position));

    Boolean checked = myChecked.get(position);
    if (checked != null) {
        checkedTextView.setChecked(checked);
    }

    return row;
}
}
}

```



Eclipse предлагает использовать **SparseBooleanArray** вместо **HashMap<Integer, Boolean>**. И к этому предупреждению стоит отнестись серьёзно, так как предлагаемый вариант требует гораздо меньше ресурсов, что для больших списков может стать важным фактором. Всё, что нужно сделать, это заменить одну строчку кода

```
private SparseBooleanArray myChecked = new SparseBooleanArray();  
// Не рекомендуется: private HashMap<Integer, Boolean> myChecked = new HashMap<Integer, Boolean>  
(());
```

Реклама

Реклама



© 2015 А.Климов (mailto:rusproject@mail.ru)  (<http://plus.google.com/109061106977829925124?prsrc=3>)



(<http://top.mail.ru/jump?from=228158>)



(<http://feeds.feedburner.com/alexanderklimov/VJcl>)

 Рекомендовать в Google

