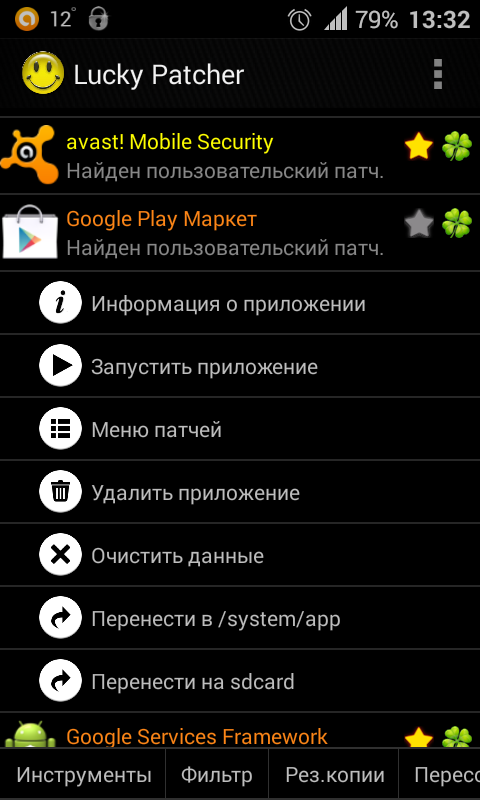
Кастомный список с меню для каждого элемента на основе ExpandableListView

Наверное многие знают приложение**Lucky Patcher**, в новых версиях которого список сделан интересно: для каждого элемента списка есть меню.

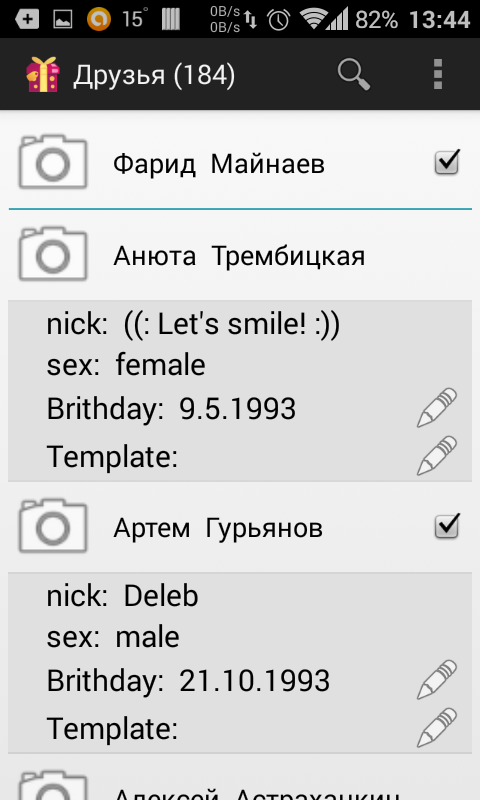


В моем приложении мне захотелось реализовать нечто подобное на примере списка друзей социальной сети.

Прочитав [*эту статью*](http://habrahabr.ru/post/147546/), где используется коллекция коллекций:

private ArrayList<ArrayList<String>> mGroups;

А так же покопавшись в [исходниках](http://androidexample.com/Custom_Expandable_ListView_Tutorial_-_Android_Example/index.php?view=article_discription&aid=107&aaid=129), я решил объединить и доработать эти примеры и вот что у меня получилось:



Приступаем к реализации. Создаем новый проект с классом ListActivity, который наследуется от **ActionBarActivity** (можно и от **Activity**). Отредактируем код нашей активности (**activity\_list.xml**):

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".ListActivity" >

<ExpandableListView

android:id="@+id/listView1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:indicatorLeft="200dp" />

</LinearLayout>

Ничего сложного — вертикальный **LinearLayout**, который содержит в себе двухуровневый список **ExpandableListView**.

Далее необходимо описать пункт списка -**item\_friend.xml**:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/friendLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal" >

<ImageView

android:id="@+id/photoFriend"

android:layout\_width="60dp"

android:layout\_height="60dp"

android:contentDescription="@string/todo"

android:src="@android:drawable/ic\_menu\_camera" />

<TextView

android:id="@+id/f\_s\_names"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="10dp"

android:layout\_weight="1"

android:text="@string/Def\_fname"

android:textAppearance="?android:attr/textAppearanceMedium" />

<ImageView

android:id="@+id/congratuated"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginRight="7dp"

android:contentDescription="@string/todo"

android:src="@android:drawable/checkbox\_on\_background"

android:visibility="gone" />

</LinearLayout>

Элемент списка содержит фотографию в **ImageView**, имя и фамилию в **TextView**, а так же еще один **ImageView**. Набор элементов может быть абсолютно любым и ограничивается лишь фантазией разработчика.

Опишем меню пункта списка — **item\_friend\_menu.xml**:

<?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:layout\_marginTop="10dp"

android:orientation="vertical"

android:background="#E0E0E0" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="25dp"

android:orientation="horizontal" >

<TextView

android:id="@+id/textNick"

style="@style/Fm"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/nick\_" />

<TextView

android:id="@+id/m\_Nick"

style="@style/Fm"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:text="@string/empty" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="25dp"

android:orientation="horizontal" >

<TextView

android:id="@+id/textSex"

style="@style/Fm"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/sex\_" />

<TextView

android:id="@+id/m\_Sex"

style="@style/Fm"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:text="@string/man" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="25dp"

android:orientation="horizontal" >

<TextView

android:id="@+id/textBdate"

style="@style/Fm"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:text="@string/brithday\_" />

<TextView

android:id="@+id/m\_Bdate"

style="@style/Fm"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginLeft="5dp"

android:layout\_weight="1"

android:text="@string/\_17\_2\_1985" />

<ImageButton

android:id="@+id/imageEditdate"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="7dp"

android:adjustViewBounds="true"

android:background="@android:color/transparent"

android:contentDescription="@string/todo"

android:scaleType="fitCenter"

android:src="@android:drawable/ic\_menu\_edit" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="25dp"

android:orientation="horizontal" >

<TextView

android:id="@+id/textTemplate"

style="@style/Fm"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:text="@string/template\_" />

<TextView

android:id="@+id/m\_Template"

style="@style/Fm"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginLeft="5dp"

android:layout\_weight="1"

android:text="@string/temp\_name" />

<ImageButton

android:id="@+id/imageSelectTemp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="7dp"

android:adjustViewBounds="true"

android:background="@android:color/transparent"

android:scaleType="fitCenter"

android:src="@android:drawable/ic\_menu\_edit"

android:contentDescription="@string/todo"/>

</LinearLayout>

</LinearLayout>

С меню элемента все аналогично — описываем то, что хотим увидеть на дисплее.  
Все необходимые XML готовы, теперь приступим к кодингу на **Java**. Нам понадобятся два вспомогательных класса, описывающих содержимое элемента списка и меню этого элемента. Сначала опишем меню в классе **FriendMenu.java**:

public class FriendMenu {

public FriendMenu() {

}

private String nick, bdate, template\_name;

private int sex;

public String getNick() {

return nick;

}

public void setNick(String nick) {

this.nick = nick;

}

public String getBdate() {

return bdate;

}

public void setBdate(String bdate) {

this.bdate = bdate;

}

public int getSex() {

return sex;

}

public void setSex(int sex) {

this.sex = sex;

}

public String getTemplate\_name() {

return template\_name;

}

public void setTemplate\_name(String template\_name) {

this.template\_name = template\_name;

}

public int getId() {

return user\_id;

}

public void setId(int id) {

this.user\_id = id;

}

}

Класс используется скорее не как меню, а как дополнительная информация об элементе списка, которую необходимо отображать по желанию пользователя.  
Здесь 4 поля — ник, день рождения, пол и поле template\_name, а так же методы **get**\* и **set\*** для работы с полями класса. В качестве поля ***sex*** можно использовать тип **boolean**, но мне пришлось использовать **int**, т.к. в списке друзей, получаемом с сервера, пол имеет тип целого числа(?).

Теперь опишем класс элемента списка **Friend.java**:

public class Friend {

public Friend() {

}

private ArrayList<FriendMenu> menu;

private Bitmap bmp;

private String text;

private boolean congratulated;

public ArrayList<FriendMenu> getMenu() {

return menu;

}

public void setMenu(ArrayList<FriendMenu> menus) {

this.menu = menus;

}

public boolean isCongratulated() {

return congratulated;

}

public void setCongratulated(boolean congratulated) {

this.congratulated = congratulated;

}

public Bitmap getBmp() {

return bmp;

}

public void setBmp(Bitmap bmp) {

this.bmp=bmp;

}

public String getText() {

return text;

}

public void setText(String text) {

this.text = text;

}

}

Поле **bmp** хранит изображение для **ImageView** элемента списка, поле **text** — имя и фамилию, поле **congratulated** отвечает за показ галочки в **ImageView**.  
В классе содиржится поле **menu** — коллекция менюшек для одного элемента списка. Помимо собственной информации внутри себя элемент 1-го уровня содержит информацию о своем меню. В моем случае **menu** будет состоять только из одного элемента.   
Здесь у вас, уважаемые читатели, наверняка возникнет замечание, что можно было бы не цеплять сразу весь **Layout** в качестве элемента списка второго уровня, а разбить его на составляющие, т.к. внутреннее содержимое примерно одинаковое (горизонтальный **LinearLayout**, внутри которого 2 **TextView** [и **ImageButton**]), а так же добавлять по мере необходимости. Согласен, у меня не оптимальное решение.

Теперь самое главное — создать свой класс, который наследуется от **BaseExpandableListAdapter** — **FriendsAdapted.java**.

public class FriendsAdapted extends BaseExpandableListAdapter {

private ArrayList<Friend> friends;

private LayoutInflater inflater;

*// каждый друг внутри себя содержит свое меню (доп. информацию).*

public FriendsAdapted(Context cont, ArrayList<Friend> list) {

inflater = LayoutInflater.from(cont);

friends = list;

}

@Override

public int getGroupCount() {

return friends.size();

}

@Override

public int getChildrenCount(int groupPosition) {

return friends.get(groupPosition).getMenu().size();

}

@Override

public Object getGroup(int groupPosition) {

return friends.get(groupPosition);

}

@Override

public Object getChild(int groupPosition, int childPosition) {

return friends.get(groupPosition).getMenu().get(childPosition);

}

@Override

public long getGroupId(int groupPosition) {

return groupPosition;

}

@Override

public long getChildId(int groupPosition, int childPosition) {

return childPosition;

}

@Override

public boolean hasStableIds() {

return true;

}

@Override

public View getGroupView(int groupPosition, boolean isExpanded,

View convertView, ViewGroup parent) {

if (convertView == null) {

convertView = inflater.inflate(R.layout.item\_friend, null);

}

*// получили группу (друга)*

Friend fr = friends.get(groupPosition);

((TextView) convertView.findViewById(R.id.f\_s\_names)).setText(fr

.getText());

*// ((ImageView)*

*// convertView.findViewById(R.id.photoFriend)).setImageBitmap(fr.getBmp());*

((ImageView) convertView.findViewById(R.id.congratuated))

.setVisibility((fr.isCongratulated() ? View.VISIBLE : View.GONE));

return convertView;

}

@Override

public View getChildView(int groupPosition, int childPosition,

boolean isLastChild, View convertView, ViewGroup parent) {

FriendMenu menu = (FriendMenu) getChild(groupPosition, childPosition);

convertView = inflater

.inflate(R.layout.item\_friend\_menu, parent, false);

((TextView) convertView.findViewById(R.id.m\_Nick)).setText(menu

.getNick());

switch (menu.getSex()) {

case 1:

((TextView) convertView.findViewById(R.id.m\_Sex))

.setText(R.string.sex\_man);

break;

case 0:

((TextView) convertView.findViewById(R.id.m\_Sex))

.setText(R.string.sex\_fem);

break;

}

((TextView) convertView.findViewById(R.id.m\_Bdate)).setText(menu

.getBdate());

((TextView) convertView.findViewById(R.id.m\_Template)).setText(menu

.getTemplate\_name());

return convertView;

}

@Override

public boolean isChildSelectable(int groupPosition, int childPosition) {

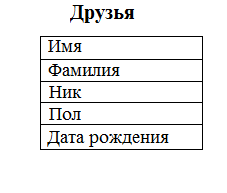
return false;

}

}

Класс содержит коллекцию друзей, а в которой каждый друг содержит свое собственное меню. Для заполнения информацией элементов первого и второго уровней необходимо переопределить методы **getGroupView** и **getChildView**соответственно. С помощью **LayoutInflater** можно подключать кастомизированные **Layout**'ы.  
Каждому **View** можно задает свои:

* Selector
* ClickListener
* LongClickListener
* и другие обработчики

Теперь осталось отобразись список. В моем приложении была предварительно заполненная база данных с таблицей друзей, поэтому список буду заполнять из нее:  
  
Код **ListActivity.java**:

public class ListActivity extends ActionBarActivity {

ExpandableListView list\_w;

static private SQLdb data;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_list);

list\_w = (ExpandableListView) findViewById(R.id.listView1);

data = new SQLdb(this, 2);

SQLiteDatabase db = data.getReadableDatabase();

Cursor c = db.rawQuery("select \* from friends", null);

ArrayList<Friend> friends = new ArrayList<Friend>();

if (c.moveToFirst()) {

for (int i = 0; i < c.getCount(); ++i) {

Friend friend = new Friend();

ArrayList<FriendMenu> menus = new ArrayList<FriendMenu>();

FriendMenu menu = new FriendMenu();

menu.setBdate(c.getString(c.getColumnIndex("bdate")));

menu.setNick(c.getString(c.getColumnIndex("nickname")));

menu.setSex(c.getInt(c.getColumnIndex("sex")));

menus.add(menu);

friend.setMenu(menus);

friend.setText(c.getString(c.getColumnIndex("fname"))

+ " " + c.getString(c.getColumnIndex("sname")));

if (i % 2 == 0) {

friend.setCongratulated(true);

}

friends.add(friend);

if (!c.moveToNext()) {

break;

}

}

} else Toast.ma