

Java
Kotlin
Дизайн
Отладка
Open Source
Полезные ресурсы

PopupMenu - Всплывающее меню

Начиная с Android 3.0, в системе появилась возможность создавать всплывающее меню, привязанное к элементу **View**. Меню реализовано в виде модального окна, которое отображается снизу от родителя меню или в другом месте, если места снизу недостаточно.

PopupMenu не нужно путать с контекстным меню. У них разные задачи, хотя поведение весьма схоже. В новых версиях Android использование всплывающих меню предпочтительнее контекстных, которые можно считать устаревшим интерфейсом.

В Android 4.0 добавили новую функциональность, чтобы работать было проще. В частности, всплывающее меню можно получить из XML-файла, используя метод **inflate(int)**, которому следует передать идентификатор ресурса меню. А до этого приходилось использовать отдельный класс **MenuInflater** с избыточным кодом.

Также появился слушатель **PopupMenu.OnDismissListener** для работы с закрытием меню. Он срабатывает либо, когда пользователь щёлкает на пункте меню и меню закрывается, либо пользователь щёлкает в другом месте экрана, и меню также закрывается.

Кстати, я обратил внимание, что всплывающее меню может быть прокручиваемым. Когда я щёлкал на изображении, то на экран выводилось только два пункта, и для третьего пункта приходилось прокручивать меню.

Создать всплывающее меню очень просто. По сути мы повторяем шаги по созданию обычного меню. Сначала в ресурсах меню создадим нужный файл:

res/menu/popupmenu.xml

```

<group android:id="@+id/menugroup1" >
    <item
        android:id="@+id/menu1"
        android:icon="@mipmap/ic_launcher"
        android:title="Popup menu item 1"/>
    <item
        android:id="@+id/menu2"
        android:title="Popup menu item 2"/>
    <item
        android:id="@+id/menu3"
        android:title="Popup menu item 3">
        <menu>
            <item
                android:id="@+id/submenu"
                android:title="Подменю"/>
        </menu>
    </item>
</group>
<group android:id="@+id/menugroup2" >
    <item
        android:id="@+id/menu4"
        android:checkable="true"
        android:checked="true"
        android:icon="@mipmap/ic_launcher"
        android:title="Popup menu item 4"/>
    <item
        android:id="@+id/menu5"
        android:title="Popup menu item 5"
        android:enabled="false"/>
    <item
        android:id="@+id/menu6"
        android:title="Popup menu item 6"/>
</group>

```

```

</menu>

```

Я постарался использовать всевозможные варианты использования атрибутов, которые отвечают за вывод флажков, доступность и т.д. Есть и другие атрибуты, о которых читай документацию.

Далее добавим на экран активности текстовую метку, кнопку и **ImageView**. При щелчке на каждом из этих компонентов мы будем выводить одинаковое всплывающее меню:

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="center_horizontal"
android:orientation="vertical">

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Показать меню" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="A TextView" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@mipmap/ic_launcher_cat" />
```

```
</LinearLayout>
```

Осталось написать код.

```

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.PopupMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button button = findViewById(R.id.button);
        TextView textView = findViewById(R.id.textView);
        ImageView imageView = findViewById(R.id.imageView);

        button.setOnClickListener(viewClickListener);
        textView.setOnClickListener(viewClickListener);
        imageView.setOnClickListener(viewClickListener);
    }

    View.OnClickListener viewClickListener = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showPopupMenu(v);
        }
    };

    private void showPopupMenu(View v) {
        PopupMenu popupMenu = new PopupMenu(this, v);
        popupMenu.inflate(R.menu.popupmenu);

        popupMenu
            .setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
                @Override
                public boolean onMenuItemClick(MenuItem item) {
                    switch (item.getItemId()) {
                        case R.id.menu1:
                            Toast.makeText(getApplicationContext(),
                                "Вы выбрали PopupMenu 1",
                                Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }
}

```

```

        "Вы выбрали PopupMenu 2",
        Toast.LENGTH_SHORT).show();
        return true;
    case R.id.menu3:
        Toast.makeText(getApplicationContext(),
            "Вы выбрали PopupMenu 3",
            Toast.LENGTH_SHORT).show();
        return true;
    default:
        return false;
    }
}

});

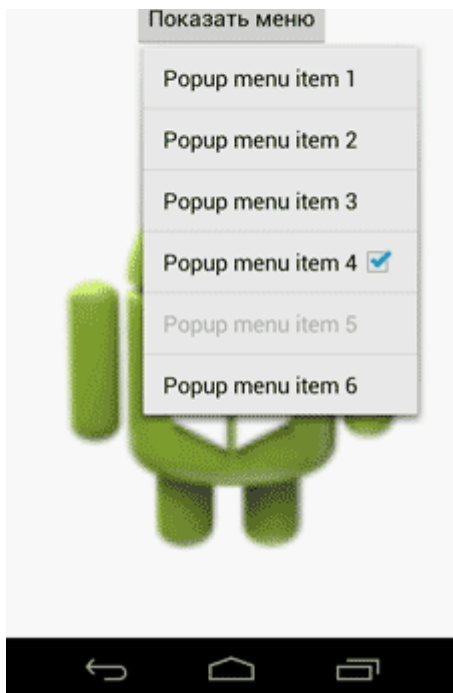
popupMenu.setOnDismissListener(new PopupMenu.OnDismissListener() {
    @Override
    public void onDismiss(PopupMenu menu) {
        Toast.makeText(getApplicationContext(), "onDismiss",
            Toast.LENGTH_SHORT).show();
    }
});
popupMenu.show();
}
}

```

Вам надо создать новый экземпляр **PopupMenu**, указав контекст активности и компонент, к которому будет привязано это меню. Далее загружаете меню из ресурсов и добавляете методы для обработки щелчков. Для отображения на экране вызывается метод **show()**.

Запустив проект, вы можете щёлкать по любому элементу на форме и увидеть всплывающее меню.

Несмотря на то, что в XML мы указали значки, в реальности они не выводятся. В интернете можно найти примеры решения проблемы через программный код. Хотелось бы увидеть поддержку из коробки.



Путаницу добавляют похожие классы **android.widget.PopupMenu**, **android.support.v7.widget.PopupMenu** и **android.support.v4.widget.PopupMenuCompat**.

Программное добавление пунктов

Повесим на новую кнопку возможность программного добавления новых пунктов меню.

```
public void onClick(View view) {
    PopupMenu popupMenu = new PopupMenu(this, view);
    // popupMenu.inflate(R.menu.popupmenu); // если добавлять к существующему меню
    popupMenu.getMenu().add(1, R.id.menu1, 1, "slot1");
    popupMenu.getMenu().add(1, R.id.menu2, 2, "slot2");
    popupMenu.show();
}
```

Дополнительное чтение

[Обсуждение статьи](#) на форуме.

Реклама



«Всё включено» в МКБ



Реклама